SIGMOD Officers, Committees, and Awardees

Chair	Vice-Chair	Secretary/Treasurer
Donald Kossmann Systems Group ETH Zürich Cab F 73 8092 Zuerich SWITZERLAND +41 44 632 29 40 <donaldk AT inf.ethz.ch>	Anastasia Ailamaki School of Computer and Communication Sciences, EPFL EPFL/IC/IIF/DIAS Station 14, CH-1015 Lausanne SWITZERLAND +41 21 693 75 64 <natassa AT epfl.ch>	Magdalena Balazinska Computer Science & Engineering University of Washington Box 352350 Seattle, WA USA +1 206-616-1069 <magda AT cs.washington.edu>

SIGMOD Executive Committee:

Anastasia Ailamaki, Magdalena Balazinska, K. Selçuk Candan, Curtis Dyreson, Donald Kossmann, Yannis Ioannidis, Richard Hull, and Ioana Manolescu.

Advisory Board:

Raghu Ramakrishnan (Chair), Yahoo! Research, <First8CharsOfLastName AT yahoo-inc.com>,
Amr El Abbadi, Serge Abiteboul, Ricardo Baeza-Yates, Phil Bernstein, Elisa Bertino, Mike Carey, Surajit Chaudhuri, Christos Faloutsos, Alon Halevy, Joe Hellerstein, Renée Miller, C. Mohan, Beng-Chin Ooi, Z. Meral Ozsoyoglu, Sunita Sarawagi, Min Wang, and Gerhard Weikum.

SIGMOD Information Director:

Curtis Dyreson, Utah State University, <curtis.dyreson AT usu.edu>

Associate Information Directors:

Manfred Jeusfeld, Georgia Koutrika, Michael Ley, Wim Martens, Mirella Moro, Rachel Pottinger, Altigran Soares da Silva, and Jun Yang.

SIGMOD Record Editor-in-Chief:

Ioana Manolescu, Inria Saclay—Île-de-France, <ioana.manolescu AT inria.fr>

SIGMOD Record Associate Editors:

Denilson Barbosa, Pablo Barceló, Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, Rada Chirkova, Anish Das Sarma, Glenn Paulley, Alkis Simitsis, Nesime Tatbul, and Marianne Winslett.

SIGMOD Conference Coordinator:

K. Selçuk Candan, Arizona State University <candan AT asu.edu>

PODS Executive Committee: Rick Hull (chair), <hull AT research.ibm.com>, Michael Benedikt, Wenfei Fan, Maurizio Lenzerini, Jan Paradaens, and Thomas Schwentick.

Sister Society Liaisons:

Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment).

Awards Committee:

Masaru Kitsuregawa (chair, University of Tokyo, <kitsure AT tk1.iis.u-tokyo.ac.jp>), Rakesh Agrawal, Elisa Bertino, Umesh Dayal, and Maurizio Lenzerini.

Jim Gray Doctoral Dissertation Award Committee:

Johannes Gehrke (Co-chair), Cornell Univ.; Beng Chin Ooi (Co-chair), National Univ. of Singapore, Alfons Kemper, Hank Korth, Alberto Laender, Boon Thau Loo, Timos Sellis, and Kyu-Young Whang.

[Last updated : September 24th, 2013]

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Formerly known as the "SIGMOD Innovations Award", it now honors Dr. E. F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)
David DeWitt (1995)
Serge Abiteboul (1998)
Rudolf Bayer (2001)
Ronald Fagin (2004)
Jennifer Widom (2007)
Umeshwar Dayal (2010)
Stefano Ceri (2013)

Jim Gray (1993)
C. Mohan (1996)
Hector Garcia-Molina (1999)
Patricia Selinger (2002)
Michael Carey (2005)
Moshe Y. Vardi (2008)
Surajit Chaudhuri (2011)

Philip Bernstein (1994)
David Maier (1997)
Rakesh Agrawal (2000)
Don Chamberlin (2003)
Jeffrey D. Ullman (2006)
Masaru Kitsuregawa (2009)
Bruce Lindsay (2012)

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)
Jeffrey Ullman (1996)
Raghu Ramakrishnan (1999)
Daniel Rosenkrantz (2001)
Surajit Chaudhuri (2004)
Hans-Jörg Schek (2007)
David Lomet (2010)
H.V. Jagadish (2013)

Gio Wiederhold (1995)
Avi Silberschatz (1997)
Michael Carey (2000)
Richard Snodgrass (2002)
Hongjun Lu (2005)
Klaus R. Dittrich (2008)
Gerhard Weikum (2011)

Yahiko Kambayashi (1995)
Won Kim (1998)
Laura Haas (2000)
Michael Ley (2003)
Tamer Özsu (2006)
Beng Chin Ooi (2009)
Marianne Winslett (2012)

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **2006** *Winner*: Gerome Miklau, University of Washington. *Runners-up*: Marcelo Arenas and Yanlei Diao.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley. *Honorable Mentions*: Xifeng Yan and Martin Theobald.
- **2008** *Winner*: Ariel Fuxman, University of Toronto. *Honorable Mentions*: Cong Yu and Nilesh Dalvi.
- **2009** *Winner*: Daniel Abadi, MIT. *Honorable Mentions*: Bee-Chung Chen and Ashwin Machanavajjhala.
- **2010** *Winner:* Christopher Ré, University of Washington. *Honorable Mentions*: Soumyadeb Mitra and Fabian Suchanek.
- **2011** *Winner*: Stratos Idreos, Centrum Wiskunde & Informatica. *Honorable Mentions*: Todd Green and Karl Schnaitterz.
- **2012** *Winner*: Ryan Johnson, Carnegie Mellon University. *Honorable Mention*: Bogdan Alexe.
- **2013** *Winner*: Sudipto Das, University of California – Santa Barbara. *Honorable Mention*: Herodotos Herodotou and Wenchao Zhou.

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated : September 24th, 2013]

Editor's Notes

Welcome to the December 2013 issue of the ACM SIGMOD Record!

First, congratulations to the newly recognized ACM fellows which are also prominent members of our scientific community: Charu C. Aggarwal, Stefano Ceri, Peter Haas, Matthias Jarke, Timos Sellis, Dennis Shasha, Kyuseok Shim, Val Tannen, and Limsoon Wong!

The issue opens with a Database Principles article by Ngo, Ré and Ruda, on new developments in the theory of join algorithms. While the general perception may be that the important things about join processing are already well-known, the authors make a very solid case against this perception. Specifically, they present a novel unifying framework of two recent results on the algorithmic complexity of join query evaluation; these optimal algorithms are not the ones taught in textbooks over the last twenty to thirty years or so, although a single commercial system already implemented one of them! Two main ideas lie behind the novel optimal algorithms: first, exploit together the query shape and statistics on the data (whereas most previously proposed optimization frameworks used either one or the other); second, process all the joins simultaneously, as opposed to ordering them and performing them one by one. I am thrilled to have this paper published in the Record, as it provides a great introduction to a very recent family of contributions on which many future works will likely be based.

The Surveys column features three articles. First, Bordawekar, Blainey and Apte consider the currently hot area of research and development generally termed "data analytics", outline the main algorithmic techniques involved, and study how each can be mapped effectively for execution on a parallel infrastructure. The authors identify a set of popular analytical models and identify for each suitable parallelization strategies for large-scale data management workloads.
The second survey, by Li, Wang, Li and Gao, focuses on the problem of approximate (similarity) joins on XML data sets. Such joins are useful, for instance, when integrating heterogeneous data sets of tree-structured data. In such contexts, approximate joins are needed, and to avoid performing too many comparisons, lower bounds on the distance between two trees are necessary. The survey outlines three families of edit distances, namely string-based, histogram-based and binary branch distance. The authors show how such distances can be combined, and also compare their associated computational costs.
The survey by Felix Naumann focuses on the topic of profiling data, seen broadly as deriving or extracting metadata under the form of statistics or other information, out of a dataset. Data profiling is recognized as an important area given the abundance of new data sets, which must be integrated in existing or novel applications. The survey identifies data profiling application areas, such as data integration, analytics, or scientific data management; it outlines previous works in the area and also discusses novel trends such as incremental data profiling, profiling on novel architectures etc.

The Distinguished Profiles in Databases column features an interview with Anand Rajaraman from WallMart Labs. Read it to find out how one co-authors the same year a SIGMOD and a VLDB paper that would 10 years later each get the respective Test-of-Time/10 years best paper award, leave the PhD program to create the Junglee start-up, come back to finish the PhD without a scholarship but paying the registration costs out of the pocket, be close to buying an obscure start-up named Google (without buying it in the end), co-head @WalmartLabs, and now invest in and advise Sillicon Valley startups!

In the Research Centers column, Christodoulakis, Garofalakis, Petrakis, Deligiannakis, Samoladas, Ioannou, Papapetrou and Sotiriadis describe ongoing data management research at the Technical University of Crete. Within the Software Technology and Network Applications (SoftNet) lab, research areas include stream monitoring, data as a service in a cloud context, and uncertain/probabilistic data managsvement. The Intelligent Systems lab studies architectures and tools for cloud-based deployment of

complex software applications, in particular for healthcare services. Finally, the Distributed Multimedia Information Systems and Applications (MUSIC) laboratory investigates semantic-based integration and interoperability of digital contents, for application contexts such as digital libraries, natural history, or cultural heritage.

The issue features three event reports. Darmont and Pedersen report on the First International Workshop on Cloud Intelligence (Cloud-I 2012), held in conjunction with the VLDB 2012 Conference in Istanbul, Turkey. The report sessions focused on topics such as data analytics, security and privacy in the cloud.
The report on the Second International Worskhop on Energy Data Management (EnDM 2013) is by Pedersen and Lehner. The workshop was held in conjunction with EDBT 2013 in Genova; it featured works on representing, extracting and visualizing energy data, energy forecasts in local distribution networks etc.
The last workshop report is from the 2nd workshop on Scalable Workflow Enactment Engines and Technology (SWEET) held in 2013 next to the SIGMOD conference. Sroka, Hidders and Missier outline the keynotes (by prof. Paul Watson from Newcastle University and Jelena Pjesivac-Grbovic from Google), and the papers on topics such as simulating execution on heterogeneous hardware, workflow scheduling, or user steering within e.g. scientific workflows.

The issue closes with two call for contributions for the ACM SIGIR conference, respectively, for the new ACM e-Energy conference, both to be held in 2014.

On behalf of the SIGMOD Record Editorial board, I wish all our community a happy and prosperous 2014, full of success and joy!

Your submissions to the Record are welcome via the submission site:

http://sigmod.hosting.acm.org/record

Prior to submitting, be sure to peruse the Editorial Policy on the SIGMOD Record's Web site (http://www.sigmod.org/publications/sigmod-record/sigmod-record-editorial-policy).

Ioana Manolescu

December 2013

Past SIGMOD Record Editors:

Harrison R. Morse (1969)
Daniel O'Connell (1971 – 1973)
Randall Rustin (1974-1975)
Douglas S. Kerr (1976-1978)
Thomas J. Cook (1981 – 1983)
Jon D. Clark (1984 – 1985)
Margaret H. Dunham (1986 – 1988)
Arie Segev (1989 – 1995)
Jennifer Widom (1995 – 1996)
Michael Franklin (1996 – 2000)
Ling Liu (2000 – 2004)
Mario Nascimento (2005 – 2007)
Alexandros Labrinidis (2007 – 2009)

Skew Strikes Back: New Developments in the Theory of Join Algorithms*

Hung Q. Ngo
University at Buffalo, SUNY
hungngo@buffalo.edu

Christopher Ré
Stanford University
chrismre@cs.stanford.edu

Atri Rudra
University at Buffalo, SUNY
atri@buffalo.edu

Evaluating the relational join is one of the central algorithmic and most well-studied problems in database systems. A staggering number of variants have been considered including Block-Nested loop join, Hash-Join, Grace, Sort-merge (see Grafe [17] for a survey, and [4, 7, 24] for discussions of more modern issues). Commercial database engines use finely tuned join heuristics that take into account a wide variety of factors including the selectivity of various predicates, memory, IO, etc. This study of join queries notwithstanding, the textbook description of join processing is *suboptimal*. This survey describes recent results on join algorithms that have provable worst-case optimality runtime guarantees. We survey recent work and provide a simpler and unified description of these algorithms that we hope is useful for theory-minded readers, algorithm designers, and systems implementors.

Much of this progress can be understood by thinking about a simple join evaluation problem that we illustrate with the so-called *triangle query*, a query that has become increasingly popular in the last decade with the advent of social networks, biological motifs, and graph databases [36, 37]

> *Suppose that one is given a graph with N edges, how many distinct triangles can there be in the graph?*

A first bound is to say that there are at most N edges, and hence at most $O(N^3)$ triangles. A bit more thought suggests that every triangle is indexed by any two of its sides and hence there at most $O(N^2)$ triangles. However, the correct, tight, and non-trivial asymptotic is $O(N^{3/2})$. An example of the questions considered in this survey is how do we list all the triangles in time $O(N^{3/2})$? Such an algorithm can be shown to have a worst-case optimal running time. In contrast, traditional databases evaluate joins pairwise, and as has been noted by several authors, this forces them to run in time $\Omega(N^2)$ on some instance of the triangle query. This survey gives an overview of recent developments that establish such non-trivial bounds for *all* join queries and algorithms that meet these bounds, which we call worst-case optimal join algorithms.

Estimates on the output size of join have been known since the 1990s, thanks to the work of Friedgut and Kahn [11] in the context of bounding the number of occurrences of a given small hypergraph inside a large hypergraph. More recently and more generally, tight estimates for the natural join problem were derived by Grohe-Marx [20] and Atserias-Grohe-Marx [2] (henceforth AGM). In fact, similar bounds can be traced back to the 1940s in geometry, where it was known as the famous Loomis-Whitney inequality [26]. The most general geometric bound is by Bollobás-Thomason in the 1990s [5]. We proved (with Porat) that AGM and the discrete version of Bollobás-Thomason are *equivalent* [29], and so the connection between these areas is deep.

Connections of join size to arcane geometric bounds may reasonably lead a practitioner to believe that the cause of suboptimality is a mysterious force wholly unknown to them—but it is not; it is the old enemy of the database optimizer: skew. We hope to highlight two conceptual messages with this survey:

- The main ideas of the algorithms presented here are a theoretically optimal way of avoiding skew – something database practitioners have been fighting with for decades. We mathematically justify a simple yet effective technique to cope with skew called the "power of two choices."

- The second idea is a challenge to the database dogma of doing *"one join at a time,"* as is done in traditional database systems. We show that there are

*Database Principles Column. Column editor: Pablo Barcelo, Department of Computer Science, University of Chile. E-mail: pbarcelo@dcc.uchile.cl.

HQN's work is partly supported by NSF grant CCF-1319402 and a gift from Logicblox. CR's work on this project is generously supported by NSF CAREER Award under No. IIS-1353606, NSF award under No. CCF-1356918, the ONR under awards No. N000141210041 and No. N000141310129, Sloan Research Fellowship, Oracle, and Google. AR's work is partly supported by NSF CAREER Award CCF-0844796, NSF grant CCF-1319402 and a gift from Logicblox.

classes of queries for which *any* join-project plan is destined to be slower than the best possible run time by a polynomial factor *in the data size*.

Outline of the Survey. We begin with a short (and necessarily incomplete) history of join processing with a focus on recent history. In Section 1, we describe how these new join algorithms work for the triangle query. In Section 2, we describe how to use the new size bounds for join queries as well as conjunctive queries with simple functional dependencies. In Section 3, we provide new simplified proofs of these bounds and join algorithms. Finally, we describe two open questions in Section 4. We recall some background knowledge in the appendix. For lack of space some details are deferred to the full version of this survey [30].

A Brief History of Join Processing

Conjunctive query evaluation in general and join query evaluation in particular have a very long history and deep connections to logic and constraint satisfaction [6, 8, 10, 14, 16, 25, 31, 38]. Most of the join algorithms with provable performance guarantees work for specific classes of queries.[1] As we describe, there are two major approaches for join processing: using *structural information of the query* and *using cardinality information*. As we explain, the AGM bounds are exciting because they bring together both types of information.

The Structural Approaches. On the theoretical side, many algorithms use some structural property of the query such as *acyclicity* or *bounded "width."* For example, when the query is acyclic, the classic algorithm of Yannakakis [42] runs in time essentially linear in the input plus output size. A query is acyclic if and only if it has a *join tree*, which can be constructed using the textbook *GYO-reduction* [18, 43].

Subsequent works further expand the classes of queries that can be evaluated in polynomial time. These works define progressively more general notions of "width" for a query, which intuitively measures how far a query is from being acyclic. Roughly, these results state that if the corresponding notion of "width" is bounded by a constant, then the query is "tractable," i.e. there is a polynomial time algorithm to evaluate it. For example, Gyssens et al. [21,22] showed that queries with bounded "degree of acyclicity" are tractable. Then came *query width* (qw) from Chekuri and Rajaraman [8], *hypertree width* and *generalized hypertree width* (ghw) from Gottlob et al. [15,34]. These are related to the *treewidth* (tw) of a query's hypergraph, rooted in Robertson and Seymour on graph minors [33]. Acyclic queries are exactly those with $\mathrm{qw} = 1$.

Cardinality-based Approaches . Width only tells half of the story, as was wonderfully articulated in Scarcello's SIGMOD Record paper [34]:

> *decomposition methods focus "only" on structural features, while they completely disregard "quantitative" aspects of the query, that may dramatically affect the query-evaluation time.*

Said another way, the width approach disregards the input relation sizes and summarizes them in a single number, N. As a result, the run time of these structural approaches is $O(N^{w+1} \log N)$, where N is the input size and w is the corresponding width measure. On the other hand, commercial RDBMSs seem to place little emphasis on the structural property of the query and tremendous emphasis on the cardinality side of join processing. Commercial databases often process a join query by breaking a complex multiway join into a series of pairwise joins; an approach first described in the seminal System R, Selinger-style optimizer from the 1970 [35]. However, throwing away this structural information comes at a cost: *any* join-project plan is destined to be slower than the best possible run time by a polynomial factor *in the data size*.

Bridging This Gap. A major recent result from AGM [2, 20] is the key to bridging this gap: AGM derived a *tight* bound on the output size of a join query as a function of individual input relation sizes *and* a much finer notion of "width". The AGM bound leads to the notion of *fractional query number* and eventually *fractional hypertree width* (fhw) which is strictly more general than all of the above width notions [28]. To summarize, for the same query, it can be shown that

$$\mathrm{fhw} \leqslant \mathrm{ghw} \leqslant \mathrm{qw} \leqslant \mathrm{tw} + 1,$$

and the join-project algorithm from AGM runs in time $O(N^{\mathrm{fhw}+1} \log N)$. AGM's bound is sharp enough to take into account cardinality information, and they can be *much* better when the input relation sizes vary. The bound takes into account *both* the input relation statistics *and* the structural properties of the query. The question is whether it is possible and how to turn the bound into join algorithms, with runtime $O(N^{\mathrm{fwh}})$ and much better when input relations do not have the same size. (These size bounds were extended to more general conjunctive queries by Gottlob et al. [13].)

The first such worst-case optimal join algorithm was designed by the authors (and Porat) in 2012 [29]. Soon after, an algorithm (with a simpler description) with the

[1] Throughout this survey, we will measure the run time of join algorithms in terms of the input data, assuming the input query has constant size; this is known as the *data complexity* measure, which is standard in database theory [38].

same optimality guarantee was presented, called "Leapfrog Triejoin" [39]. Remarkably this algorithm was already implemented in a commercial database system *before* its optimality guarantees were discovered. A key idea in the algorithms is handling skew in a theoretically optimal way, and uses many of the same techniques that database management systems have used for decades heuristically [9, 40, 41]

A technical contribution of this survey is to describe the algorithms from [29] and [39] and their analyses in one unifying (and simplified) framework. In particular, we make the observation that these join algorithms are in fact special cases of a *single* join algorithm. This result is new and serves to explain the common link between these join algorithms. We also illustrate some unexpected connections with geometry, which we believe are interesting in their own right and may be the basis for further theoretical development.

1. MUCH ADO ABOUT THE TRIANGLE

We begin with the triangle query

$$Q_\triangle = R(A,B) \bowtie S(B,C) \bowtie T(A,C).$$

The above query is the simplest cyclic query and is rich enough to illustrate most of the ideas in the new join algorithms.[2] We first describe the traditional way to evaluate this query and how skew impacts this query. We then develop two closely related algorithmic ideas allowing us to mitigate the impact of skew in these examples; they are the key ideas behind the recent join processing algorithms.

1.1 Why traditional join plans are suboptimal

The textbook way to evaluate any join query, including $Q_\triangle$, is to determine the best pair-wise join plan [32, Ch. 15]. Figure 1 illustrates three plans that a conventional RDBMS would use for this query. For example, the first plan is to compute the intermediate join $P = R \bowtie T$ and then compute $P \bowtie S$ as the final output.

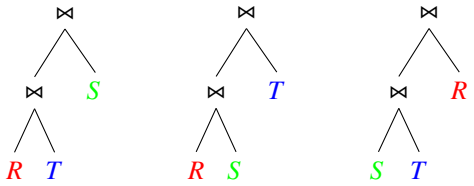


Figure 1: The three pair-wise join plans for $Q_\triangle$.

[2]This query can be used to list all triangles in a given graph $G = (V,E)$, if we set R, S and T to consist of all pairs (u,v) and (v,u) for which uv is an edge. Due to symmetry, each triangle in G will be listed 6 times in the join.

We next construct a family of instances for which any of the above three join plans must run in time $\Omega(N^2)$ because the intermediate relation P is too large. Let $m \geqslant 1$ be a positive integer. The instance family is illustrated in Figure 2, where the domains of the attributes A, B and C are $\{a_0, a_1, \ldots, a_m\}$, $\{b_0, b_1, \ldots, b_m\}$, and $\{c_0, c_1, \ldots, c_m\}$ respectively. In Figure 2, the unfilled circles denote the values a_0, b_0 and c_0 respectively while the black circles denote the rest of the values.

For this instance each relation has $N = 2m + 1$ tuples and $|Q_\triangle| = 3m + 1$; however, any pair-wise join has size $m^2 + m$. Thus, for large m, any of the three join plans will take $\Omega(N^2)$ time. In fact, it can be shown that even if we allow projections in addition to joins, the $\Omega(N^2)$ bound still holds. (See Lemma 3.2.) By contrast, the two algorithms shown in the next section run in time $O(N)$, which is optimal because the output itself has $\Omega(N)$ tuples!

1.2 Algorithm 1: The Power of Two Choices

Inspecting the bad example above, one can see a root cause for the large intermediate relation: a_0 has "high degree" or in the terminology to follow it is *heavy*. In other words, it is an example of *skew*. To cope with skew, we shall take a strategy often employed in database systems: we deal with nodes of high and low skew using different join techniques [9, 41]. The first goal then is to understand when a value has high skew. To shorten notations, for each a_i define

$$Q_\triangle[a_i] := \pi_{B,C}(\sigma_{A=a_i}(Q_\triangle)).$$

We will call a_i *heavy* if $|\sigma_{A=a_i}(R \bowtie T)| \geqslant |Q_\triangle[a_i]|$. In other words, the value a_i is *heavy* if its contribution to the size of intermediate relation $R \bowtie T$ is *greater* than its contribution to the size of the output. Since

$$|\sigma_{A=a_i}(R \bowtie T)| = |\sigma_{A=a_i}R| \cdot |\sigma_{A=a_i}T|,$$

we can easily compute the left hand side of the above inequality from an appropriate index of the input relations. Of course, we do not know $|Q_\triangle[a_i]|$ until after we have computed $Q_\triangle$. However, note that we always have $Q_\triangle[a_i] \subseteq S$. Thus, we will use $|S|$ as a proxy for $|Q_\triangle[a_i]|$. The two choices come from the following two ways of computing $Q_\triangle[a_i]$:

(i) Compute $\sigma_{A=a_i}(R) \bowtie \sigma_{A=a_i}(T)$ and filter the results by probing against S or

(ii) Consider each tuple in $(b,c) \in S$ and check if $(a_i,b) \in R$ and $(a_i,c) \in T$.

We pick option (i) when a_i is light (low skew) and pick option (ii) when a_i is heavy (high skew).

Example 1. Let us work through the motivating example from Figure 2. When we compute $Q_\triangle[a_0]$, we

$$R = \{a_0\} \times \{b_0, \ldots, b_m\} \cup \{a_0, \ldots, a_m\} \times \{b_0\}$$

$$S = \{b_0\} \times \{c_0, \ldots, c_m\} \cup \{b_0, \ldots, b_m\} \times \{c_0\}$$

$$T = \{a_0\} \times \{c_0, \ldots, c_m\} \cup \{a_0, \ldots, a_m\} \times \{c_0\}$$

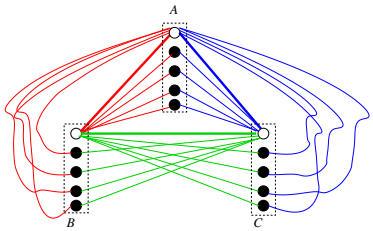


Figure 2: Counter-example for join-project only plans for the triangles (left) and an illustration for $m = 4$ (right). The pairs connected by the red/green/blue edges form the tuples in the relations $R/S/T$ respectively. Note that the in this case each relation has $N = 2m + 1 = 9$ tuples and there are $3m + 1 = 13$ output tuples in $Q_\triangle$. Any pair-wise join however has size $m^2 + m = 20$.

realize that a_0 is heavy and hence, we use option (ii) above. Since here we just scan tuples in S, computing $Q_\triangle[a_0]$ takes $O(m)$ time. On the other hand, when we want to compute $Q_\triangle[a_i]$ for $i \geqslant 1$, we realize that these a_i's are light and so we take option (i). In these cases $|\sigma_{A=a_i}R| = |\sigma_{A=a_i}T| = 1$ and hence the algorithm runs in time $O(1)$. As there are m such light a_i's, the algorithm overall takes $O(m)$ each on the heavy and light vertices and thus $O(m) = O(N)$ overall which is the best possible since the output size is $\Theta(N)$.

Algorithm and Analysis. Algorithm 1 fully specifies how to compute $Q_\triangle$ using the above idea of two choices. Given that the relations R, S, and T are already indexed appropriately, computing L in line 2 can easily be done in time $O(\min\{|R|, |T|\})$ using sort-merge join. (We assume input relations are already sorted and this runtime does not count this one-time pre-processing cost.) Then, for each $a \in L$, the body of the for loop from line 4 to line 11 clearly takes time in the order of

$$\min\left(|\sigma_{A=a}R| \cdot |\sigma_{A=a}T|, |S|\right),$$

thanks to the power of two choices! Thus, the overall time spent by the algorithm is up to constant factors

$$\sum_{a \in L} \min\left(|\sigma_{A=a}R| \cdot |\sigma_{A=a}T|, |S|\right). \tag{1}$$

We bound the sum above by using two inequalities. The first is the simple observation that for any $x, y \geqslant 0$

$$\min(x, y) \leqslant \sqrt{xy}. \tag{2}$$

The second is the famous Cauchy-Schwarz inequality[3]:

$$\sum_{a \in L} x_a \cdot y_a \leqslant \sqrt{\sum_{a \in L} x_a^2} \cdot \sqrt{\sum_{a \in L} y_a^2}, \tag{3}$$

where $(x_a)_{a \in L}$ and $(y_a)_{a \in L}$ are vectors of real values. Applying (2) to (1), we obtain

$$\sum_{a \in L} \sqrt{|\sigma_{A=a}R| \cdot |\sigma_{A=a}T| \cdot |S|} \tag{4}$$

$$= \sqrt{|S|} \cdot \sum_{a \in L} \sqrt{|\sigma_{A=a}R|} \cdot \sqrt{|\sigma_{A=a}T|} \tag{5}$$

$$\leqslant \sqrt{|S|} \cdot \sqrt{\sum_{a \in L} |\sigma_{A=a}R|} \cdot \sqrt{\sum_{a \in L} |\sigma_{A=a}T|}$$

$$\leqslant \sqrt{|S|} \cdot \sqrt{\sum_{a \in \pi_A(R)} |\sigma_{A=a}R|} \cdot \sqrt{\sum_{a \in \pi_A(T)} |\sigma_{A=a}T|}$$

$$= \sqrt{|S|} \cdot \sqrt{|R|} \cdot \sqrt{|T|}.$$

If $|R| = |S| = |T| = N$, then the above is $O(N^{3/2})$ as claimed in the introduction. We will generalize the above algorithm beyond triangles to general join queries in Section 3. Before that, we present a second algorithm that has exactly the same worst-case run-time and a similar analysis to illustrate the recursive structure of the generic worst-case join algorithm described in Section 3.

1.3 Algorithm 2: Delaying the Computation

Now we present a second way to compute $Q_\triangle[a_i]$ that differentiates between heavy and light values $a_i \in A$ in a different way. We don't try to estimate the heaviness of a_i right off the bat. Algorithm 2 "looks deeper" into what pairs (b, c) can go along with a_i in the output by computing c for each candidate b.

Algorithm 2 works as follows. By computing the intersection $\pi_B(\sigma_{A=a_i}(R)) \cap \pi_B(S)$, we only look at the candidates b that can possibly participate with a_i in the output (a_i, b, c). Then, the candidate set for c is $\pi_C(\sigma_{B=b}(S)) \cap \pi_C(\sigma_{A=a_i}(T))$. When a_i is really skewed toward the heavy side, the candidates b and then c help gradually reduce the skew toward building up the final solution $Q_\triangle$.

Example 2. Let us now see how delaying computation works on the bad example. As we have observed in using the power of two choices, computing the intersection

[3]The inner product of two vectors is at most the product of their length.

Algorithm 1 Computing $Q_\triangle$ with power of two choices.

Input: $R(A,B), S(B,C), T(A,C)$ in sorted order

```
1: Q△ ← ∅
2: L ← πA(R) ∩ πA(T)
3: For each a ∈ L do
4:     If |σA=aR| · |σA=aT| ⩾ |S| then
5:         For each (b,c) ∈ S do
6:             If (a,b) ∈ R and (a,c) ∈ T then
7:                 Add (a,b,c) to Q△
8:     else
9:         For each b ∈ πB(σA=aR) ∧ c ∈ πC(σA=aT)
   do
10:            If (b,c) ∈ S then
11:                Add (a,b,c) to Q△
12: Return Q
```

Algorithm 2 Computing $Q_\triangle$ by delaying computation.

Input: $R(A,B), S(B,C), T(A,C)$ in sorted order

```
1: Q ← ∅
2: LA ← πA(R) ∩ πA(T)
3: For each a ∈ LA do
4:     L_B^a ← πB(σA=a(R)) ∩ πB(S)
5:     For each b ∈ L_B^a do
6:         L_C^{a,b} ← πC(σB=b(S)) ∩ πC(σA=a(T))
7:         For each c ∈ L_C^{a,b} do
8:             Add (a,b,c) to Q
9: Return Q
```

of two sorted sets takes time at most the *minimum* of the two sizes. Sort-merge join has this runtime guarantee, because its inputs are already sorted. Note that the sort-merge join algorithm also makes use of the power of two choices idea implicitly to deal with skew. If one set represents high skew, having very large size, and the other set has very small size, then their intersection using sort-merge join only takes time proportional to the smaller size.

For a_0, we consider all $b \in \{b_0, b_1, \ldots, b_m\}$. When $b = b_0$, we have

$$\pi_C(\sigma_{B=b_0} S) = \pi_C(\sigma_{A=a_0} T) = \{c_0, \ldots, c_m\},$$

so we output the $m+1$ triangles in total time $O(m)$. For the pairs (a_0, b_i) when $i \geqslant 1$, we have $|\sigma_{B=b_i} S| = 1$ and hence we spend $O(1)$ time on each such pair, for a total of $O(m)$ overall.

Now consider a_i for $i \geqslant 1$. In this case, $b = b_0$ is the only candidate. Further, for (a_i, b_0), we have $|\sigma_{A=a_i} T| = 1$, so we can handle each such a_i in $O(1)$ time leading to an overall run time of $O(m)$. Thus on this bad example Algorithm 2 runs in $O(N)$ time.

We present the full analysis of Algorithm 2 in [30]: its worst-case runtime is exactly the same as that of Algorithm 1. What is remarkable is that both of these algorithms follow exactly the same recursive structure and they are special cases of a generic worst-case optimal join algorithm.

2. A USER'S GUIDE TO THE AGM BOUND

We now describe one way to generalize the bound of the output size of a join (mirroring the $O(N^{3/2})$ bound we saw for the triangle query) and illustrate its use with a few examples.

2.1 AGM Bound

To state the AGM bound, we need some notation. The natural join problem can be defined as follows. We are given a collection of m relations. Each relation is over a collection of attributes. We use $\mathcal{V}$ to denote the set of attributes; let $n = |\mathcal{V}|$. The join query Q is modeled as a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where for each hyperedge $F \in \mathcal{E}$ there is a relation R_F on attribute set F. Figure 3 shows several example join queries, their associated hypergraphs, and illustrates the bounds below.

Atserias-Grohe-Marx [2] and Grohe-Marx [20] proved the following remarkable inequality, which shall be referred to as the *AGM's inequality* henceforth. Let $\mathbf{x} = (x_F)_{F \in \mathcal{E}}$ be *any* point in the following polyhedron:

$$\left\{ \mathbf{x} \mid \sum_{F : v \in F} x_F \geqslant 1, \forall v \in \mathcal{V}, \mathbf{x} \geqslant \mathbf{0} \right\}.$$

Such a point $\mathbf{x}$ is called a *fractional edge cover* of the hypergraph $\mathcal{H}$. Then, AGM's inequality states that the join size can be bounded by

$$|Q| = |\bowtie_{F \in \mathcal{E}} R_F| \leqslant \prod_{F \in \mathcal{E}} |R_F|^{x_F}. \tag{6}$$

2.2 Example Bounds

We now illustrate the AGM bound on some specific join queries. We begin with the triangle query $Q_\triangle$. In this case the corresponding hypergraph $\mathcal{H}$ is as in the left part of Figure 3. We consider two covers (which are also marked in Figure 3). The first one is $x_R = x_T = x_S = \frac{1}{2}$. This is a valid cover since the required inequalities are satisfied for every vertex. For example, for vertex C, the two edges incident on it are S and T and we have $x_S + x_T = 1 \geqslant 1$ as required. In this case the bound (6) states that

$$|Q_\triangle| \leqslant \sqrt{|R| \cdot |S| \cdot |T|}. \tag{7}$$

Another valid cover is $x_R = x_T = 1$ and $x_S = 0$ (this cover is also marked in Figure 3). This is a valid cover, e.g. since for C we have $x_S + x_T = 1 \geqslant 1$ and for vertex

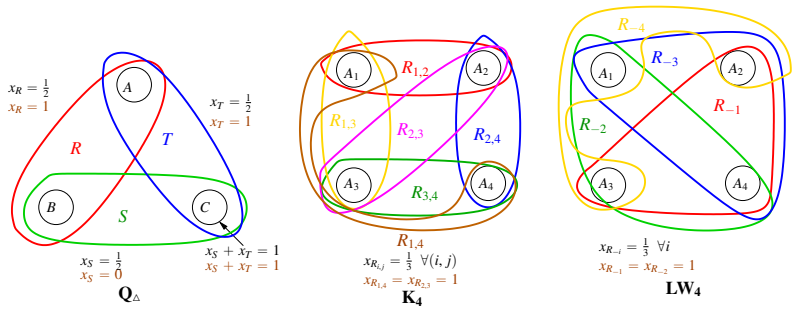


Figure 3: A handful of queries and their covers.

A, we have $x_R + x_T = 2 \geqslant 1$ as required. For this cover, bound (6) gives

$$|Q_\triangle| \leqslant |R| \cdot |T|. \tag{8}$$

These two bounds can be better in different scenarios. E.g. when $|R| = |S| = |T| = N$, then (7) gives an upper bound of $N^{3/2}$ (which is the tight answer) while (8) gives a bound of N^2, which is worse. However, if $|R| = |T| = 1$ and $|S| = N$, then (7) gives a bound of $\sqrt{N}$, which has a lot of slack; while (8) gives a bound of 1, which is tight.

For another class of examples, consider the "clique" query. In this case there are $n \geqslant 3$ attributes and $m = \binom{n}{2}$ relations: one $R_{i,j}$ for every $i < j \in [n]$: we will call this query K_n. Note that K_3 is $Q_\triangle$. The middle part of Figure 3 draws the K_4 query. We highlight one cover: $x_{R_{i,j}} = \frac{1}{n-1}$ for every $i < j \in [n]$. This is a valid cover since every attribute is contained in $n-1$ relations. Further, in this case (6) gives a bound of $\sqrt[n-1]{\prod_{i<j} |R_{i,j}|}$, which simplifies to $N^{n/2}$ for the case when every relation has size N.

Finally, we consider the Loomis-Whitney LW_n queries. In this case there are n attributes and there are $m = n$ relations. In particular, for every $i \in [n]$ there is a relation $R_{-i} = R_{[n]\setminus\{i\}}$. Note that LW_3 is $Q_\triangle$. See the right of Figure 3 for the LW_4 query. We highlight one cover: $x_{R_{i,j}} = \frac{1}{n-1}$ for every $i < j \in [n]$. This is a valid cover since every attribute is contained in $n-1$ relations. Further, in this case (6) gives a bound of $\sqrt[n-1]{\prod_i |R_{-i}|}$, which simplifies to $N^{1+\frac{1}{n-1}}$ for the case when every relation has size N. Note that this bound approaches N as n becomes larger.

2.3 The Tightest AGM Bound

As we just saw, the optimal edge cover for the AGM bound depends on the relation sizes. To minimize the right hand side of (6), we can solve the following linear program:

$$\begin{aligned} \min \quad & \sum_{F \in \mathcal{E}} (\log_2 |R_F|) \cdot x_F \\ \text{s.t.} \quad & \sum_{F : v \in F} x_F \geqslant 1, v \in \mathcal{V} \\ & \mathbf{x} \geqslant \mathbf{0} \end{aligned}$$

Implicitly, the objective function above depends on the database instance $\mathcal{D}$ on which the query is applied. Let $\rho^*(Q, \mathcal{D})$ denote the optimal objective value to the above linear program. We refer to $\rho^*(Q, \mathcal{D})$ as the *fractional edge cover number* of the query Q with respect to the database instance $\mathcal{D}$, following Grohe [19]. The AGM's inequality can be summarized simply by $|Q| \leqslant 2^{\rho^*(Q,\mathcal{D})}$.

2.4 Applying AGM bound on conjunctive queries with simple functional dependencies

Thus far we have been describing bounds and algorithms for natural join queries. A super-class of natural join queries is called *conjunctive queries*. A conjunctive query is a query of the form

$$C = R_0(\bar{X}_0) \leftarrow R_1(\bar{X}_1) \wedge \cdots \wedge R_m(\bar{X}_m)$$

where $\{R_1, \ldots, R_m\}$ is a multi-set of relation symbols, i.e. some relation might occur more than once in the query, $\bar{X}_0, \ldots, \bar{X}_m$ are tuples of variables, and each variable occurring in the query's head $R(\bar{X}_0)$ must also occur in the body. It is important to note that the same variable might occur more than once in the same tuple $\bar{X}_i$.

We will use $\text{vars}(C)$ to denote the set of all variables occurring in C. Note that $\bar{X}_0 \subseteq \text{vars}(C)$ and it is entirely possible for $\bar{X}_0$ to be *empty* (Boolean conjunctive query). For example, the following are conjunctive queries:

$$\begin{aligned} R_0(WXYZ) &\leftarrow S(WXY) \wedge S(WWW) \wedge T(YZ) \\ R_0(Z) &\leftarrow S(WXY) \wedge S(WWW) \wedge T(YZ). \end{aligned}$$

The former query is a *full conjunctive query* because the head atom contains all the query's variables.

Following Gottlob, Lee, Valiant, and Valiant (henceforth GLVV) [12, 13], we also know that the AGM bound can be extended to general conjunctive queries even with

simple functional dependencies.[4] In this survey, our presentation closely follows Grohe's presentation of GLVV [19].

To illustrate what can go "wrong" when we are moving from natural join queries to conjunctive queries, let us consider a few example conjunctive queries, introducing one issue at a time. In all examples below, relations are assumed to have the same size N.

Example 3 (Projection)**.** Consider

$$C_1 = R_0(W) \leftarrow R(WX) \wedge S(WY) \wedge T(WZ).$$

In the (natural) join query, $R(WX) \wedge S(WY) \wedge T(WZ)$ AGM bound gives N^3; but because $R_0(W) \subseteq \pi_W(R) \bowtie \pi_W(S) \bowtie \pi_W(T)$, AGM bound can be adapted to the instance restricted only to the output variables yielding an upper bound of N on the output size.

Example 4 (Repeated variables)**.** Consider the query

$$C_2 = R_0(WY) \leftarrow R(WW) \wedge S(WY) \wedge T(YY).$$

This is a full conjunctive query as all variables appear in the head atom R_0. In this case, we can replace $R(WW)$ and $T(YY)$ by keeping only tuples $(t_1, t_2) \in R$ for which $t_1 = t_2$ and tuples $(t_1, t_2) \in T$ for which $t_1 = t_2$; essentially, we turn the query into a natural join query of the form $R'(W) \wedge S(WY) \wedge T'(Y)$. For this query, $x_{R'} = x_{T'} = 0$ and $x_S = 1$ is a fractional cover and thus by AGM bound N is an upperbound on the output size.

Example 5 (Introducing the chase)**.** Consider the query

$$C_3 = R_0(WXY) \leftarrow R(WX) \wedge R(WW) \wedge S(XY).$$

Without additional information, the best bound we can get for this query is $O(N^2)$: we can easily turn it into a natural join query of the form $R(WX) \wedge R'(W) \wedge S(XY)$, where R' is obtained from R by keeping all tuples $(t_1, t_2) \in R$ for which $t_1 = t_2$. Then, $(x_R, x_{R'}, x_S)$ is a fractional edge cover for this query if and only if $x_R + x_{R'} \geqslant 1$ (to cover W), $x_R + x_S \geqslant 1$ (to cover X), $x_S \geqslant 1$ (to cover Y); So, $x_S = x_{R'} = 1$ and $x_R = 0$ is a fractional cover, yielding the $O(N^2)$ bound. Furthermore, it is easy to construct input instances for which the output size is $\Omega(N^2)$:

$$\begin{aligned} R &= \{(i,i) \mid i \in [N/2]\} \bigcup \{(i,0) \mid i \in [N/2]\} \\ S &= \{(0,j) \mid j \in [N]\}. \end{aligned}$$

Every tuple $(i, 0, j)$ for $i \in [N/2]$, $j \in [N]$ is in the output.

Next, suppose we have an additional piece of information that the first attribute in relation R is its *key*, i.e. if (t_1, t_2) and (t_1, t_2') are in R, then $t_2 = t_2'$. Then we can significantly reduce the output size bound because we can infer the following about the output tuples: (w, x, y) is an output tuple iff (w, x) and (w, w) are in R, and (x, y) are in S. The functional dependency tells us that $x = w$. Hence, the query is equivalent to $C_3' = R_0(WY) \leftarrow R(WW) \wedge S(WY)$. The AGM bound for this (natural) join query is N. The transformation from C_3 to C_3' we just described is, of course, the famous *chase* operation [1,3,27], which is much more powerful than what is conveyed by this example.

Example 6 (Taking advantage of FDs)**.** Consider the following query

$$C_4 = R_0(XY_1, \ldots, Y_k, Z) \leftarrow \bigwedge_{i=1}^{k} R_i(XY_i) \wedge \bigwedge_{i=1}^{k} S_i(Y_i Z).$$

First, without any functional dependency, AGM bound gives N^k for this query, because the fractional cover constraints are

$$\begin{aligned} \sum_{i=1}^{k} x_{R_i} &\geqslant 1 \text{ (cover } X) \\ x_{R_i} + x_{S_i} &\geqslant 1 \text{ (cover } Y_i) \; i \in [k] \\ \sum_{i=1}^{k} x_{S_i} &\geqslant 1 \text{ (cover } Z). \end{aligned}$$

The AGM bound is $N^{\sum_i (x_{R_i} + x_{S_i})} \geqslant N^k$.

Second, suppose we know $k+1$ functional dependencies: each of the first attributes of relations $R_1, \ldots, R_k$ is a key for the corresponding relation, and the first attribute of S_1 is its key. Then, we have the following sets of functional dependencies: $X \rightarrow Y_i$, $i \in [k]$, and $Y_1 \rightarrow Z$. Now, construct a fictitious relation $R'(X, Y_1, \ldots, Y_k, Z)$ as follows: $(x, y_1, \ldots, y_k, z) \in R'$ iff $(x, y_i) \in R_i$ for all $i \in [k]$ and $(y_1, z) \in S_1$. Then, obviously $|R'| \leqslant N$. More importantly, the output does not change if we add R' to the body query C_4 to obtain a new conjunctive query C_4'. However, this time we can set $x_{R'} = 1$ and all other variables in the fractional cover to be 0 and obtain an upper bound of N.

We present a more formal treatment of the steps needed to convert a conjunctive query with simple functional dependencies to a join query in [30].

3. WORST-CASE-OPTIMAL ALGORITHMS

We first show how to analyze the upper bound that proves AGM and from which we develop a generalized join algorithm that captures both algorithms from Ngo-Porat-Ré-Rudra [29] (henceforth NPRR) and Leapfrog Triejoin [39]. Then, we describe the limitation of any join-project plan.

[4] GLVV also have fascinating bounds for the general functional dependency and composite keys cases, and characterization of treewidth-preserving queries; both of those topics are beyond the scope of this survey, in part because they require different machinery from what we have developed thus far.

Henceforth, we need the following notation. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be any hypergraph and $I \subseteq \mathcal{V}$ be an arbitrary subset of vertices of $\mathcal{H}$. Then, we define

$$\mathcal{E}_I := \{F \in \mathcal{E} \mid F \cap I \neq \emptyset\}.$$

Example 7. For the query $Q_\triangle$ from Section 1, we have $\mathcal{H}_\triangle = (\mathcal{V}_\triangle, \mathcal{E}_\triangle)$, where

$$\begin{aligned} \mathcal{V}_\triangle &= \{A, B, C\}, \\ \mathcal{E}_\triangle &= \big\{\{A, B\}, \{B, C\}, \{A, C\}\big\}. \end{aligned}$$

Let $I_1 = \{A\}$ and $I_2 = \{A, B\}$, then $\mathcal{E}_{I_1} = \{\{A, B\}, \{A, C\}\}$, and $\mathcal{E}_{I_2} = \mathcal{E}_\triangle$.

3.1 A proof of the AGM bound

We prove the AGM inequality in two steps: a query decomposition lemma, and then a succinct inductive proof, which we then use to develop a generic worst-case optimal join algorithm.

3.1.1 The query decomposition lemma

Ngo-Porat-Ré-Rudra [29] gave an inductive proof of AGM bound (6) using Hölder inequality. (AGM proved the bound using an entropy based argument: see [30] for more details.) The proof has an inductive structure leading naturally to recursive join algorithms. NPRR's strategy is a generalization of the strategy in [5] to prove the Bollobás-Thomason inequality, shown in [29] to be *equivalent* to AGM's bound.

Implicit in NPRR is the following key lemma, which will be crucial in proving bounds on general join queries (as well as proving upper bounds on the runtime of the new join algorithms).

Lemma 3.1 (Query decomposition lemma)**.** *Let $Q = \bowtie_{F\in\mathcal{E}} R_F$ be a natural join query represented by a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, and $\mathbf{x}$ be any fractional edge cover for $\mathcal{H}$. Let $\mathcal{V} = I \uplus J$ be an arbitrary partition of $\mathcal{V}$ such that $1 \leqslant |I| < |\mathcal{V}|$; and,*

$$L = \bowtie_{F\in\mathcal{E}_I} \pi_I(R_F).$$

Then,

$$\sum_{\mathbf{t}_I \in L} \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F} \leqslant \prod_{F\in\mathcal{E}} |R_F|^{x_F}. \tag{9}$$

Before we prove the lemma above, we outline how we have already used the lemma above specialized to $Q_\triangle$ in Section 1 to bound the runtime of Algorithm 1. We use the lemma with $\mathbf{x} = (1/2, 1/2, 1/2)$, which is a valid fractional edge cover for $\mathcal{H}_\triangle$.

For Algorithm 1 we use Lemma 3.1 with $I = \{A\}, J = \{B, C\}$. Note that L in Lemma 3.1 is the same as

$$\pi_A(R) \bowtie \pi_A(T) = \pi_A(R) \cap \pi_A(T),$$

i.e. this L is exactly the same as the L in Algorithm 1. We now consider the left hand side (LHS) in (9). Note that we have $\mathcal{E}_J = \big\{\{A, B\}, \{B, C\}, \{A, C\}\big\}$. Thus, the LHS is the same as

$$\begin{aligned} &\sum_{a\in L} \sqrt{|R \ltimes (a)|} \cdot \sqrt{|T \ltimes (a)|} \cdot \sqrt{|S \ltimes (a)|} \\ &= \sum_{a\in L} \sqrt{|\sigma_{A=a}R|} \cdot \sqrt{|\sigma_{A=a}T|} \cdot \sqrt{|S|}. \end{aligned}$$

Note that the last expression is exactly the same as (4), which is at most $\sqrt{|R| \cdot |S| \cdot |T|}$ by Lemma 3.1. This was shown in Section 1.

Proof of Lemma 3.1. The plan is to "unroll" the sum of products on the left hand side using Hölder inequality as follows. Let $j \in I$ be an arbitrary attribute. Define

$$\begin{aligned} I' &= I - \{j\} \\ J' &= J \cup \{j\} \\ L' &= \bowtie_{F\in\mathcal{E}_{I'}} \pi_{I'}(R_F). \end{aligned}$$

We will show that

$$\sum_{\mathbf{t}_I \in L} \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F} \leqslant \sum_{\mathbf{t}_{I'} \in L'} \prod_{F\in\mathcal{E}_{J'}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F}. \tag{10}$$

Then, by repeated applications of (10) we will bring I' down to empty and the right hand side is that of (9).

To prove (10) we write $\mathbf{t}_I = (\mathbf{t}_{I'}, t_j)$ for some $\mathbf{t}_{I'} \in L'$ and decompose a sum over L to a double sum over L' and t_j, where the second sum is only over t_j for which $(\mathbf{t}_{I'}, t_j) \in L$.

$$\begin{aligned} &\sum_{\mathbf{t}_I \in L} \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F} \\ &= \sum_{\mathbf{t}_{I'} \in L'} \sum_{t_j} \prod_{F\in\mathcal{E}_J} |R_F \ltimes (\mathbf{t}_{I'}, t_j)|^{x_F} \\ &= \sum_{\mathbf{t}_{I'} \in L'} \sum_{t_j} \left(\prod_{F\in\mathcal{E}_J} |R_F \ltimes (\mathbf{t}_{I'}, t_j)|^{x_F}\right) \cdot \left(\prod_{F\in\mathcal{E}_{J'}-\mathcal{E}_J} 1^{x_F}\right) \\ &= \sum_{\mathbf{t}_{I'} \in L'} \sum_{t_j} \prod_{F\in\mathcal{E}_{J'}} |R_F \ltimes (\mathbf{t}_{I'}, t_j)|^{x_F} \\ &= \sum_{\mathbf{t}_{I'} \in L'} \prod_{F\in\mathcal{E}_{J'}-\mathcal{E}_{\{j\}}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F} \sum_{t_j} \prod_{F\in\mathcal{E}_{\{j\}}} |R_F \ltimes (\mathbf{t}_{I'}, t_j)|^{x_F} \\ &\leqslant \sum_{\mathbf{t}_{I'} \in L'} \prod_{F\in\mathcal{E}_{J'}-\mathcal{E}_{\{j\}}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F} \prod_{F\in\mathcal{E}_{\{j\}}} \left(\sum_{t_j} |R_F \ltimes (\mathbf{t}_{I'}, t_j)|\right)^{x_F} \\ &\leqslant \sum_{\mathbf{t}_{I'} \in L'} \prod_{F\in\mathcal{E}_{J'}-\mathcal{E}_{\{j\}}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F} \prod_{F\in\mathcal{E}_{\{j\}}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F} \\ &= \sum_{\mathbf{t}_{I'} \in L'} \prod_{F\in\mathcal{E}_{J'}} |R_F \ltimes \mathbf{t}_{I'}|^{x_F}. \end{aligned}$$

In the above, the third equality follows from fact that $F \subseteq I' \cup \{j\}$ for any $F \in \mathcal{E}_{J'} - \mathcal{E}_J$. The first inequality is an application of Hölder inequality, which holds because $\sum_{F\in\mathcal{E}_{\{j\}}} x_F \geqslant 1$. The second inequality holds since the sum is only over t_j for which $(\mathbf{t}_{I'}, t_j) \in L$. □

It is quite remarkable that from the query decomposition lemma, we can prove AGM inequality (6), and describe and analyze two join algorithms succinctly.

3.1.2 *An inductive proof of* AGM *inequality*

Base case. In the base case $|\mathcal{V}| = 1$, we are computing the join of $|\mathcal{E}|$ unary relations. Let $\mathbf{x} = (x_F)_{F\in\mathcal{E}}$ be a fractional edge cover for this instance. Then,

$$\begin{aligned} |\bowtie_{F\in\mathcal{E}} R_F| &\leqslant \min_{F\in\mathcal{E}} |R_F| \leqslant \left(\min_{F\in\mathcal{E}} |R_F|\right)^{\sum_{F\in\mathcal{E}} x_F} \\ &= \prod_{F\in\mathcal{E}} \left(\min_{F\in\mathcal{E}} |R_F|\right)^{x_F} \leqslant \prod_{F\in\mathcal{E}} |R_F|^{x_F}. \end{aligned}$$

Inductive step. Now, assume $n = |\mathcal{V}| \geqslant 2$. Let $\mathcal{V} = I \uplus J$ be any partition of $\mathcal{V}$ such that $1 \leqslant |I| < |\mathcal{V}|$. Define $L = \bowtie_{F\in\mathcal{E}_I} \pi_I(R_F)$ as in Lemma 3.1. For each tuple $\mathbf{t}_I \in L$ we define a new join query

$$Q[\mathbf{t}_I] := \bowtie_{F\in\mathcal{E}_J} \pi_J(R_F \ltimes \mathbf{t}_I).$$

Then, obviously we can write the original query Q as

$$Q = \bigcup_{\mathbf{t}_I\in L} (\{\mathbf{t}_I\} \times Q[\mathbf{t}_I]). \tag{11}$$

The vector $(x_F)_{F\in\mathcal{E}_J}$ is a fractional edge cover for the hypergraph of $Q[\mathbf{t}_I]$. Hence, the inductive hypothesis gives us

$$|Q[\mathbf{t}_I]| \leqslant \prod_{F\in\mathcal{E}_J} |\pi_J(R_F \ltimes \mathbf{t}_I)|^{x_F} = \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F}. \tag{12}$$

From (11), (12), and (9) we obtain AGM inequality:

$$|Q| = \sum_{\mathbf{t}_I\in L} |Q[\mathbf{t}_I]| \leqslant \prod_{F\in\mathcal{E}} |R_F|^{x_F}.$$

3.2 Worst-case optimal join algorithms

From the proof of Lemma 3.1 and the query decomposition (11), it is straightforward to design a class of recursive join algorithms which is optimal in the worst case: see Algorithm 3. On the surface it seems that Algorithm 3 does not deal with skew explicitly. However, the algorithm deals with skew implicitly and this is visible in the *analysis* of the algorithm.

A mild assumption which is not very crucial is to pre-index all the relations so that the inputs to the subqueries $Q[\mathbf{t}_I]$ can readily be available when the time comes to compute it. Both NPRR and Leapfrog Triejoin algorithms do this by fixing a global attribute order and build a B-tree-like index structure for each input relation consistent with this global attribute order. A hash-based indexing structure can also be used to remove a log-factor from the final run time. We will not delve on this point here, except to emphasize the fact that we do not include the linear time pre-processing step in the final runtime expression.

Algorithm 3 Generic-Join($\bowtie_{F\in\mathcal{E}} R_F$)

Input: Query Q, hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$
Input: Input relations already indexed

```
Q ← ∅
If |V| = 1 then
    return ⋂_{F∈E} R_F
Pick I arbitrarily such that 1 ⩽ |I| < |V|
L ← Generic-Join(⋈_{F∈E_I} π_I(R_F))
For every t_I ∈ L do
    Q[t_I] ← Generic-Join(⋈_{F∈E_J} π_J(R_F ⋉ t_I))
    Q ← Q ∪ {t_I} × Q[t_I]
Return Q
```

Given the indices, when $|\mathcal{V}| = 1$ computing $\bigcap_{F\in\mathcal{E}} R_F$ can easily be done in time

$$\tilde{O}(m \min |R_F|) = \tilde{O}(m \prod_{F\in\mathcal{E}} |R_F|^{x_F}).$$

To attain this run time, an m-way sort merge can be performed. The power of m choices is implicitly applied: some relations R_F might be skewed having extremely large size, but the intersection can still be computed in time proportional to the smallest relation size. (Again, here we assume that the input is already presorted.) Then, given this base-case runtime guarantee, we can show by induction that the overall runtime of Algorithm 3 is $\tilde{O}(mn \prod_{F\in\mathcal{E}} |R_F|^{x_F})$, where $\tilde{O}$ hides a potential log-factor of the input size. This is because, by induction the time it takes to compute L is $\tilde{O}(m|I| \prod_{F\in\mathcal{E}_I} |R_F|^{x_F})$, and the time it takes to compute $Q[\mathbf{t}_I]$ is

$$\tilde{O}\left(m(n - |I|) \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F}\right)$$

Hence, from Lemma 3.1, the total run time is $\tilde{O}$ of

$$\begin{aligned} & m|I| \prod_{F\in\mathcal{E}_I} |R_F|^{x_F} + m(n - |I|) \sum_{\mathbf{t}_I\in L} \prod_{F\in\mathcal{E}_J} |R_F \ltimes \mathbf{t}_I|^{x_F} \\ \leqslant\ & m|I| \prod_{F\in\mathcal{E}_I} |R_F|^{x_F} + m(n - |I|) \prod_{F\in\mathcal{E}} |R_F|^{x_F} \\ \leqslant\ & mn \prod_{F\in\mathcal{E}} |R_F|^{x_F}. \end{aligned}$$

The NPRR algorithm is an instantiation of Algorithm 3 where it picks $J \in \mathcal{E}$, $I = \mathcal{V} - J$, and solves the subqueries $Q[\mathbf{t}_I]$ in a different way, making explicit use of the power of two choices idea. Since $J \in \mathcal{E}$, we write

$$Q[\mathbf{t}_I] = R_J \bowtie (\bowtie_{F\in\mathcal{E}_J - \{J\}} \pi_J(R_F \ltimes \mathbf{t}_I)).$$

Now, if $x_J \geqslant 1$ then we solve for $Q[\mathbf{t}_I]$ by checking for every tuple in R_J whether it can be part of $Q[\mathbf{t}_I]$. The

run time is $\tilde{O}$ of

$$(n-|I|)|R_J| \leqslant (n-|I|)\prod_{F\in\mathcal{E}_J}|R_F \ltimes \mathbf{t}_I|^{x_F}.$$

When $x_J < 1$, we will make use of an extremely simple observation: for any real numbers $p, q \geqslant 0$ and $z \in [0,1]$, $\min\{p,q\} \leqslant p^z q^{1-z}$ (note that (2) is the special case of $z = 1/2$). In particular, define

$$\begin{aligned} p &= |R_J| \\ q &= \prod_{F\in\mathcal{E}_J-\{J\}}|\pi_J(R_F \ltimes \mathbf{t}_I)|^{\frac{x_F}{1-x_J}} \end{aligned}$$

Then,

$$\begin{aligned} \min\{p,q\} &\leqslant |R_J|^{x_J}\prod_{F\in\mathcal{E}_J-\{J\}}|\pi_J(R_F \ltimes \mathbf{t}_I)|^{x_F} \\ &= \prod_{F\in\mathcal{E}_J}|R_F \ltimes \mathbf{t}_I|^{x_F}. \end{aligned}$$

From there, when $x_J < 1$ and $p \leqslant q$, we go through each tuple in R_J and check as in the case $x_J \geqslant 1$. And when $p > q$, we solve the subquery $\bowtie_{F\in\mathcal{E}_J-\{J\}}\pi_J(R_F \ltimes \mathbf{t}_I)$ first using $\left(\frac{x_F}{1-x_J}\right)_{F\in\mathcal{E}_J-\{J\}}$ as its fractional edge cover; and then checking for each tuple in the result whether it is in R_J. In either case, the run time is $\tilde{O}(\min\{p,q\})$, which along with the observation above completes the proof.

Next we outline how Algorithm 1 is Algorithm 3 with the above modification for NPRR for the triangle query $Q_\triangle$. In particular, we will use $\mathbf{x} = (1/2, 1/2, 1/2)$ and $I = \{A\}$. Note that this choice of I implies that $J = \{B, C\}$, which means in Step 5 Algorithm 3 computes

$$L = \pi_A(R) \bowtie \pi_A(T) = \pi_A(R) \cap \pi_A(T),$$

which is exactly the same L as in Algorithm 1. Thus, in the remaining part of Algorithm 3 one would cycle through all $a \in L$ (as one does in Algorithm 1). In particular, by the discussion above, since $x_S = 1/2 < 1$, we will try the best of two choices. In particular, we have

$$\begin{aligned} \bowtie_{F\in\mathcal{E}_J-\{J\}}\pi_J(R_F \ltimes (a)) &= \pi_B(\sigma_{A=a}R) \times \pi_C(\sigma_{A=a}T), \\ p &= |S|, \\ q &= |\sigma_{A=a}R| \cdot |\sigma_{A=a}T|. \end{aligned}$$

Hence, the NPRR algorithm described exactly matches Algorithm 1.

The Leapfrog Triejoin algorithm [39] is an instantiation of Algorithm 3 where $\mathcal{V} = [n]$ and $I = \{1, \ldots, n-1\}$ (or equivalently $I = \{1\}$!). Next, we outline how Algorithm 2 is Algorithm 3 with $I = \{A, B\}$ when specialized to $Q_\triangle$. Consider the run of Algorithm 3 on $\mathcal{H}_\triangle$, and the first time Step 4 is executed. The call to Generic-Join in Step 5 returns $L = \{(a,b) | a \in L_A, b \in L_B^a\}$, where L_A and L_B^a are as defined in Algorithm 2. The rest of Algorithm 3 is to do the following for every $(a,b) \in L$. $Q[(a,b)]$ is computed by the recursive call to Algorithm 3 to obtain $\{(a,b)\} \times L_C^{a,b}$, where

$$L_C^{a,b} = \pi_C(\sigma_{B=b}(S)) \bowtie \pi_C(\sigma_{A=a}(T)),$$

exactly as was done in Algorithm 2. Finally, we get back to L in Step 5 being as claimed above. Note that the recursive call of Algorithm 3 is on the query $Q_{\bowtie} = R \bowtie \pi_B(S) \bowtie \pi_A(T)$. The claim follows by picking $I = \{A\}$ in Step 4 when Algorithm 3 is run on $Q_{\bowtie}$ (and tracing through rest of Algorithm 3).

3.3 On the limitation of any join-project plan

AGM proved that there are classes of queries for which join-only plans are significantly worse than their join-project plan. In particular, they showed that for every $M, N \in \mathbb{N}$, there is a query Q of size at least M and a database $\mathcal{D}$ of size at least N such that $2^{\rho^*(Q,\mathcal{D})} \leqslant N^2$ and every join-only plan runs in time at least $N^{\frac{1}{5}\log_2|Q|}$.

NPRR continued with the story and noted that for the class of LW_n queries from Section 2.2 every join-project plan runs in time polynomially worse than the AGM bound. The proof of the following lemma can be found in [30].

Lemma 3.2. *Let $n \geqslant 2$ be an arbitrary integer. For any LW-query Q with corresponding hypergraph $\mathcal{H} = ([n], \binom{[n]}{n-1})$, and any positive integer $N \geqslant 2$, there exist n relations R_i, $i \in [n]$ such that $|R_i| = N, \forall i \in [n]$, the attribute set for R_i is $[n] - \{i\}$, and that* any *join-project plan for Q on these relations has run-time at least $\Omega(N^2/n^2)$.*

Note that both the traditional join-tree-based algorithms and AGM's algorithm are join-project plans. Consequently, they run in time asymptotically worse than the best AGM bound for this instance, which is

$$|\bowtie_{i=1}^n R_i| \leqslant \prod_{i=1}^n |R_i|^{1/(n-1)} = N^{1+1/(n-1)}.$$

On the other hand, both algorithms described in Section 3.2 take $O(N^{1+1/(n-1)})$-time because their run times match the AGM bound. In fact, the NPRR algorithm in Section 3.2 can be shown to run in linear data-complexity time $O(n^2N)$ for this query [29].

4. OPEN QUESTIONS

We conclude this survey with two open questions: one for systems researchers and one for theoreticians:

1. A natural question to ask is whether the algorithmic ideas that were presented in this survey can gain runtime efficiency in databases systems. This is an intriguing open question: on one hand we have shown asymptotic improvements in join algorithms, but on the other there are several decades

of engineering refinements and research contributions in the traditional dogma.

2. Worst-case results may only give us information about pathological instances. Thus, there is a natural push toward more refined measures of complexity. For example, current complexity measures are too weak to explain why indexes are used or give insight into the average case. For example, could one design an adaptive join algorithm whose run time is somehow dictated by the "difficulty" of the input instance (instead of the input size as in the currently known results)?

5. REFERENCES

[1] Aho, A. V., Beeri, C., and Ullman, J. D. The theory of joins in relational databases. *ACM Trans. Database Syst. 4*, 3 (1979), 297–314.

[2] Atserias, A., Grohe, M., and Marx, D. Size bounds and query plans for relational joins. *SIAM J. Comput. 42*, 4 (2013), 1737–1767.

[3] Beeri, C., and Vardi, M. Y. A proof procedure for data dependencies. *J. ACM 31*, 4 (1984), 718–741.

[4] Blanas, S., Li, Y., and Patel, J. M. Design and evaluation of main memory hash join algorithms for multi-core CPUs. In *SIGMOD* (2011), ACM, pp. 37–48.

[5] Bollobás, B., and Thomason, A. Projections of bodies and hereditary properties of hypergraphs. *Bull. London Math. Soc. 27*, 5 (1995), 417–424.

[6] Chandra, A. K., and Merlin, P. M. Optimal implementation of conjunctive queries in relational data bases. In *STOC* (1977), J. E. Hopcroft, E. P. Friedman, and M. A. Harrison, Eds., ACM, pp. 77–90.

[7] Chaudhuri, S. An overview of query optimization in relational systems. In *PODS* (1998), ACM, pp. 34–43.

[8] Chekuri, C., and Rajaraman, A. Conjunctive query containment revisited. *Theor. Comput. Sci. 239*, 2 (2000), 211–229.

[9] DeWitt, D. J., Naughton, J. F., Schneider, D. A., and Seshadri, S. Practical skew handling in parallel joins. In *Proceedings of the 18th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1992), VLDB '92, Morgan Kaufmann Publishers Inc., pp. 27–40.

[10] Fagin, R. Degrees of acyclicity for hypergraphs and relational database schemes. *J. ACM 30*, 3 (1983), 514–550.

[11] Friedgut, E., and Kahn, J. On the number of copies of one hypergraph in another. *Israel J. Math. 105* (1998), 251–256.

[12] Gottlob, G., Lee, S. T., and Valiant, G. Size and treewidth bounds for conjunctive queries. In *PODS* (2009), J. Paredaens and J. Su, Eds., ACM, pp. 45–54.

[13] Gottlob, G., Lee, S. T., Valiant, G., and Valiant, P. Size and treewidth bounds for conjunctive queries. *J. ACM 59*, 3 (2012), 16.

[14] Gottlob, G., Leone, N., and Scarcello, F. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci. 64*, 3 (2002), 579–627.

[15] Gottlob, G., Leone, N., and Scarcello, F. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *J. Comput. Syst. Sci. 66*, 4 (2003), 775–808.

[16] Gottlob, G., Miklós, Z., and Schwentick, T. Generalized hypertree decompositions: NP-hardness and tractable variants. *J. ACM 56*, 6 (2009).

[17] Graefe, G. Query evaluation techniques for large databases. *ACM Computing Surveys 25*, 2 (June 1993), 73–170.

[18] Graham, M. H. On the universal relation, 1980. Tech. Report.

[19] Grohe, M. Bounds and algorithms for joins via fractional edge covers. In *In Search of Elegance in the Theory and Practice of Computation* (2013), V. Tannen, L. Wong, L. Libkin, W. Fan, W.-C. Tan, and M. P. Fourman, Eds., vol. 8000 of *Lecture Notes in Computer Science*, Springer, pp. 321–338.

[20] Grohe, M., and Marx, D. Constraint solving via fractional edge covers. In *SODA* (2006), ACM Press, pp. 289–298.

[21] Gyssens, M., Jeavons, P., and Cohen, D. A. Decomposing constraint satisfaction problems using database techniques. *Artif. Intell. 66*, 1 (1994), 57–89.

[22] Gyssens, M., and Paredaens, J. A decomposition methodology for cyclic databases. In *Advances in Data Base Theory* (1982), pp. 85–122.

[23] Hardy, G. H., Littlewood, J. E., and Pólya, G. *Inequalities*. Cambridge University Press, Cambridge, 1988. Reprint of the 1952 edition.

[24] Kim, C., Kaldewey, T., Lee, V. W., Sedlar, E., Nguyen, A. D., Satish, N., Chhugani, J., Di Blas, A., and Dubey, P. Sort vs. hash revisited: fast join implementation on modern multi-core CPUs. *Proc. VLDB Endow. 2*, 2 (Aug. 2009), 1378–1389.

[25] Kolaitis, P. G., and Vardi, M. Y. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci. 61*, 2 (2000), 302–332.

[26] Loomis, L. H., and Whitney, H. An inequality related to the isoperimetric inequality. *Bull. Amer.*

Math. Soc 55 (1949), 961–962.

[27] Maier, D., Mendelzon, A. O., and Sagiv, Y. Testing implications of data dependencies. *ACM Trans. Database Syst. 4*, 4 (Dec. 1979), 455–469.

[28] Marx, D. Approximating fractional hypertree width. *ACM Trans. Algorithms 6*, 2 (Apr. 2010), 29:1–29:17.

[29] Ngo, H. Q., Porat, E., Ré, C., and Rudra, A. Worst-case optimal join algorithms: [extended abstract]. In *PODS* (2012), pp. 37–48.

[30] Ngo, H. Q., Re, C., and Rudra, A. Skew Strikes Back: New Developments in the Theory of Join Algorithms. *ArXiv e-prints* (Oct. 2013).

[31] Papadimitriou, C. H., and Yannakakis, M. On the complexity of database queries. In *PODS* (1997), A. O. Mendelzon and Z. M. Özsoyoglu, Eds., ACM Press, pp. 12–19.

[32] Ramakrishnan, R., and Gehrke, J. *Database Management Systems*, 3 ed. McGraw-Hill, Inc., New York, NY, USA, 2003.

[33] Robertson, N., and Seymour, P. D. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms 7*, 3 (1986), 309–322.

[34] Scarcello, F. Query answering exploiting structural properties. *SIGMOD Record 34*, 3 (2005), 91–99.

[35] Selinger, P. G., Astrahan, M. M., Chamberlin, D. D., Lorie, R. A., and Price, T. G. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1979), SIGMOD '79, ACM, pp. 23–34.

[36] Suri, S., and Vassilvitskii, S. Counting triangles and the curse of the last reducer. In *WWW* (2011), pp. 607–614.

[37] Tsourakakis, C. E. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM* (2008), IEEE Computer Society, pp. 608–617.

[38] Vardi, M. Y. The complexity of relational query languages (extended abstract). In *STOC* (1982), H. R. Lewis, B. B. Simons, W. A. Burkhard, and L. H. Landweber, Eds., ACM, pp. 137–146.

[39] Veldhuizen, T. L. Leapfrog Triejoin: a worst-case optimal join algorithm. In *ICDT* (2014). To appear.

[40] Walton, C. B., Dale, A. G., and Jenevein, R. M. A taxonomy and performance model of data skew effects in parallel joins. In *Proceedings of the 17th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1991), VLDB '91, Morgan Kaufmann Publishers Inc., pp. 537–548.

[41] Xu, Y., Kostamaa, P., Zhou, X., and Chen, L. Handling data skew in parallel joins in shared-nothing systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2008), SIGMOD '08, ACM, pp. 1043–1052.

[42] Yannakakis, M. Algorithms for acyclic database schemes. In *VLDB* (1981), IEEE Computer Society, pp. 82–94.

[43] Yu, C., and Ozsoyoglu, M. On determining tree-query membership of a distributed query. *Informatica 22*, 3 (1984), 261–282.

APPENDIX

The following form of Hölder's inequality (also historically attributed to Jensen) can be found in any standard text on inequalities. The reader is referred to the classic book "Inequalities" by Hardy, Littlewood, and Pólya [23] (Theorem 22 on page 29).

Lemma .1 (Hölder inequality)**.** *Let m, n be positive integers. Let $y_1, \ldots, y_n$ be non-negative real numbers such that $y_1 + \cdots + y_n \geqslant 1$. Let $a_{ij} \geqslant 0$ be non-negative real numbers, for $i \in [m]$ and $j \in [n]$. With the convention $0^0 = 0$, we have:*

$$\sum_{i=1}^{m} \prod_{j=1}^{n} a_{ij}^{y_j} \leqslant \prod_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} \right)^{y_j}. \tag{13}$$

Analyzing Analytics

Rajesh Bordawekar
IBM Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
bordaw@us.ibm.com

Bob Blainey
IBM Toronto Software Lab
8200 Warden Avenue
Markham, Ontario L6G 1C7
blainey@ca.ibm.com

Chidanand Apte
IBM Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
apte@us.ibm.com

ABSTRACT

Many organizations today are faced with the challenge of processing and distilling information from huge and growing collections of data. Such organizations are increasingly deploying sophisticated mathematical algorithms to model the behavior of their business processes to discover correlations in the data, to predict trends and ultimately drive decisions to optimize their operations. These techniques, are known collectively as *analytics*, and draw upon multiple disciplines, including statistics, quantitative analysis, data mining, and machine learning.

In this survey paper, we identify some of the key techniques employed in analytics both to serve as an introduction for the non-specialist and to explore the opportunity for greater optimizations for parallelization and acceleration using commodity and specialized multi-core processors. We are interested in isolating and documenting repeated patterns in analytical algorithms, data structures and data types, and in understanding how these could be most effectively mapped onto parallel infrastructure. To this end, we focus on analytical models that can be executed using different algorithms. For most major model types, we study implementations of key algorithms to determine common computational and runtime patterns. We then use this information to characterize and recommend suitable parallelization strategies for these algorithms, specifically when used in data management workloads.

1. ANALYTICS AT YOUR SERVICE

From streaming news updates on smart-phones, to instant messages on micro-blogging sites, to posts on social network sites, we are all being overwhelmed by massive amounts of data [35, 29]. Access to such a large amount of diverse data can be of tremendous value if useful *information* can be extracted and applied rapidly and accurately to a problem at hand. For instance, we could contact all of our *nearby* friends for a dinner at a local *mutually agreeable* and *well-reviewed* restaurant that has advertised *discounts* and *table availability* for *that night*; but finding and organizing all that information in a short period of time is very challenging. Similar opportunities exist for businesses and governments, but the volume, variety and velocity of data can be far greater. This process of identifying, extracting, processing, and integrating *information* from *raw data*, and then applying it to solve a problem is broadly referred to as *analytics*.

Table 1 presents a sample of analytic applications from different domains, along with their functional characteristics. As this table illustrates, many services that we take for granted and use extensively in everyday life would not be possible without analytics. For example, social networking applications such as Facebook, Twitter, and LinkedIn encode social relationships as graphs and use graph algorithms to identify hidden patterns (e.g., finding common friends). Other popular applications like Google Maps, Yelp or FourSquare combine location and social relationship information to answer complex spatial queries (e.g., finding the nearest restaurant of a particular cuisine that your friends like). Usage of analytics has substantially improved the capabilities and performance of gaming systems as demonstrated by the recent win of IBM's Watson/DeepQA intelligent question-answer system over human participants in the Jeopardy challenge [31]. The declining cost of computing and storage and the availability of such infrastructure in *cloud* environments has enabled organizations of any size to deploy advanced analytics and to package those analytic applications for broad usage by consumers.

While consumer analytical solutions may help us all to better organize or enrich our personal lives, the analytic process is also becoming a critical capability and competitive differentiator for modern businesses, governments and other organizations. In the current environment, organizations need to make on-time, informed decisions to succeed. Given the globalized economy, many businesses have sup-

Application (Domain)	Principal Goals	Key Functional Characteristics
Netflix and Pandora [3, 19] (Consumer)	Video and music recommendation	Analyzing structured and unstructured data, Personalized recommendations
Yelp and FourSquare (Consumer)	Integrated geographical analytics	Spatial queries/ranking, Streaming and persistent data
DeepQA (Watson) [12] (HealthCare/Consumer)	Intelligent question-answer (Q/A) System	Real-time natural language, Unstructured data processing, Artificial intelligence techniques for result ranking
Telecom churn analysis [28] (Telecom)	Analysis of call-data records	Graph modeling of call records, Large graph dataset, Connected component identification
Fraud analytics (Insurance/HealthCare)	Detection of suspicious behavior	Identification of abnormal patterns Real-time data analysis over streaming and persistent data
Cognos consumer insight [30] Twitter sentiment [13] (Hospitality)	Sentiment/Trend analysis	Processing large corpus of text documents, Extraction transformation, Text indexing, Entity extraction
UPS [1], Airline scheduling [20] (Logistics)	Transportation routing	Mathematical programming solutions for transportation
Integrated supply Chain (Resource Planning)	Maximize end-to-end efficiencies	Mathematical solutions for optimizing under multiple constraints
Salesforce.com (Marketing)	Customer data analytics	Reporting, Text search, Multi-tenant support, Automated price determination, Recommendation
Quantitative Trading (Finance)	Identify trading opportunities	Identification of patterns in high-speed data Statistical modeling of financial instruments
Moody's, Fitch, and S&P [7, 6] (Finance)	Financial credit rating	Statistical analysis of large historical data
Amazon retail analysis (Retail)	End-to-end retail management	Analysis over large persistent and transactional data, Integration with logistics and customer information
Energy trading (Energy)	Determining prices	Processing large time-series data, Integrated stochastic models for generation, storage and transmission
Splunk [32] (Enterprise)	System management analysis	Text analysis of system logs, Large data sets
Flickr and Twitter, Facebook and Linkedin (Consumer/Enterprise)	Social network analysis	Graph modeling of relations, Massive graph datasets, Graph analytics, Multi-media annotations and indexing
Voice of customer analytics [4] (Enterprise)	Analyzing customer voice records	Natural language processing, Text entity extraction
Workforce Analytics (Enterprise)	Intelligent staffing	Human resource matching, Intelligent work assignments
Genomics (Medical/BioInformatics)	Genome analysis, sequencing, and assembly	Analysis of large text sequences, Processing of large graphs
fMRI analysis (Medical)	Analyzing synaptic activities	Graph modeling, Graph analytics
Facial recognition [33] (Government)	Biometric classification	Analysis and matching of 2-/3-D images, Large data sets
Predictive policing [21] (Government)	Crime prediction	Spatial and temporal analytics of iamges and streams

Table 1: Well-known analytics solutions and their key characteristics

ply chains and customers that span multiple continents. In the public sector, citizens are demanding more access to services and information than ever before. Huge improvements in communication infrastructure have resulted in wide-spread use of online commerce and a boom in smart, connected mobile devices. More and more organizations are run around the clock, across multiple geographies and time zones and those organizations are being *instrumented* to an unprecedented degree. This has resulted in a deluge of data that can be studied to harvest valuable information and make better decisions. In many cases, these large volumes of data must be processed rapidly in order to make timely decisions. Consequently, many organizations have employed analytics to help them decide what kind of data they should collect, how this data should be analyzed to glean key information, and how this information should be used for achieving their organizational goals. Examples of such techniques can be found in almost any sector of the economy, including financial services [7, 6], government [33, 14], healthcare, retail [28, 24], manufacturing, logistics [1, 20], hospitality, and eCommerce [8, 9].

1.1 Characterizing Analytics Workloads

The distinguishing feature of an analytics application is the use of mathematical formulations for modeling and processing the raw data, and for applying the extracted information [34]. These techniques include statistical approaches, numerical linear algebraic methods, graph algorithms, relational operators, and string algorithms. In practice, an analytics application uses multiple formulations, each with unique functional and runtime characteristics (Table 1). Further, depending on the functional and runtime constraints, the same application can use different algorithms. While many of the applications process a large volume of data, the type of data processed varies considerably. Internet search engines process unstructured text documents as input, while retail analytics operate on structured data stored in relational databases. Some applications such as Google Maps, Yelp, or Netflix use both structured and unstructured data. The velocity of data also differs substantially across analytics applications. Search engines process read-only historical data whereas retail analytics process both historical and transactional data. Other applications, such as the monitoring of medical instruments, work exclusively on real-time or streaming data. Depending on the mathematical formulation, the volume and velocity of data and the expected I/O access patterns, the data structures and algorithms used by analytical applications vary considerably. These data structures include vectors, matrices, graphs, trees, relational tables, lists, hash-based structures, and binary objects. They can be further tuned to support in-memory, out-of-core, or streaming execution of the associated algorithm. Thus, analytics applications are characterized by diverse requirements, but share a common focus on the application of advanced mathematical modelling, typically on large data sets.

1.2 Systems Implications

Although analytics applications have come of age, they have not yet received significant attention from the data management and systems communities. It is important to understand systems implications of the analytics applications, not only because of their diverse and demanding requirements, but also, because systems architecture is currently undergoing a series of disruptive changes. Wide-spread use of technologies such as multi-core processors, specialized co-processors or accelerators, flash memory-based solid state drives (SSDs), and high-speed networks has created new optimization opportunities. More advanced technologies such as phase-change memory are on the horizon and could be game-changers in the way data is stored and analyzed.

In spite of these trends, currently there is limited usage of such technologies in the analytics domain. Even in the current implementations, it is often difficult for analytics solution developers to fine-tune system parameters, both in hardware and software, to address specific performance problems. Naive usage of modern technologies often leads to unbalanced solutions that further increase optimization complexity. Thus, to ensure effective utilization of system resources: CPU, memory, networking, and storage, it is necessary to evaluate analytics workloads in a holistic manner.

1.3 Our Study

In this paper, we aim to understand the application of modern systems technologies to optimizing analytics workloads by exploring the interplay between overall system design, core algorithms, software (e.g., compilers, operating system), and hardware (e.g., networking, storage, and processors). Specifically, we are interested in isolating repeated patterns in analytical applications, algorithms, data structures, and data types, and using them to make informed decisions on systems design. Over the past two years, we have been examining the functional flow of a variety of analytical workloads across multiple domains (Table 1), and as a result of this exer-

cise, we have identified a set of commonly-used analytical models, called *analytics exemplars* [5]. We believe that these exemplars represent the essence of analytical workloads and can be used as a toolkit for performing exploratory systems design for the analytics domain. We use these exemplars to illustrate that analytics applications benefit greatly from holistically co-designed software and hardware solutions and demonstrate this approach using the Netezza [11] appliance as an example. In spite of the recent efforts in integrating analytics components into database systems, a lot of work still needs to be done [25, 15, 11], in particular, for accelerating analytics workloads within the context of database systems. We hope this study acts as a *call to action* for researchers to focus future data management and systems research on analytics.

2. ANATOMY OF ANALYTICS WORKLOADS

To motivate the study of analytics workloads, we first describe in detail a recent noteworthy analytics application: the Watson intelligent question/answer (Q/A) system [12].

2.1 The Watson DeepQA System

Watson is a computer system developed to play the Jeopardy! game-show against human participants [31]. Waston's goals are to correctly interpret the input natural language questions, accurately predict answers to the input questions and finally, intelligently choose the input topics and the wager amounts to maximize the gains. Watson is designed as an open-domain Q/A system using the DeepQA system, a probabilistic evidence-based software architecture whose core computational principle is to assume and pursue multiple interpretations of the input question, to generate many plausible answers or hypotheses and to collect and evaluate many different competing evidence paths that might support or refute those hypotheses through a broad search of large volumes of content.

This process is accomplished using multiple stages: the first, question analysis and decomposition stage parses the input question and analyzes it to detect any semantic entities like names or dates. The analysis also identifies any relations in the question using pattern-based or statistical approaches. Next, using this information, a keyword-based primary search is performed over a varied set of sources, such as natural language documents, relational databases and knowledge bases, and a set of supporting passages (initial evidence) is identified. This is followed by the candidate (hypothesis) generation phase which

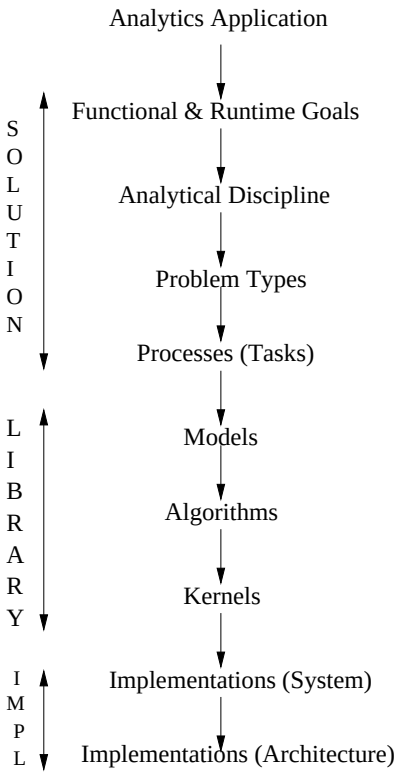


Figure 1: Simplified functional flow of business analytics applications

uses rule-based heuristics to select a set of candidates that are likely to be the answers to the input question. The next step, Hypothesis and Evidence Scoring, for each evidence-hypothesis pair, applies different algorithms that dissect and analyze the evidence along different *dimensions of evidence* such as time, geography, popularity, passage support, and source reliability. The end result of this stage is a ranked list of candidate answers, each with a confidence score indicating the degree to which the answer is believed correct, along with links back to the evidence. Finally, these evidence features are combined and weighted by a logistic regression to produce the final confidence score that determines the successful candidate (i.e. the correct answer). In addition to finding correct answers, Watson needs to master the strategies to select the clues to it's advantage and bet the appropriate amount for any given situation. The DeepQA system models different scenarios of the Jeopardy! game using different simulation approaches (e.g., Monte Carlo techniques) and uses the acquired insights to maximize Watson's winning chances by guiding topic selection, answering decisions and wager selections.

2.2 Functional Flow of Analytics Applications

The Watson system displays many traits that are common across analytics applications. They all have one or more functional goals. These goals are accomplished by one or more multi-stage processes, where each stage is an independent analytical component. To study the complex interactions between these components, it is useful to examine the functional flow of an analytics application from the customer usage to implementation stages. As Figure 1 illustrates, execution of an analytics application can be partitioned into three main phases: (1) solution, (2) library, and (3) implementation.

2.2.1 The Solution Phase

The solution phase is end-user focused and customized to to satisfy user's functional goals, which can be one of the following: prediction, prescription, reporting, recommendation, quantitative analysis, simulation, pattern matching, or alerting[1]. For example, Watson's key functional goals are: *pattern matching* for input question analysis, *prediction* for choosing answers, and *simulation* for wager and clue selection. Usually, any functional goal needs to be achieved under certain runtime constraints, e.g., calculations to be completed within a fixed time period, processing very large datasets or large volumes of data over streams, supporting batch or ad-hoc queries, or supporting a large number of concurrent users. For example, for a given clue, the Watson system is expected to find an answer before any of the human participants in the quiz. To achieve the functional and runtime goals of an application, the analytical solution leverages well-known analytical disciplines such as machine learning, data mining, statistics, business intelligence, and numerical analysis. Specifically, for a given analytical problem, the solution chooses appropriate problem types from these disciplines to build processes. Examples of analytic problem types include supervised and unsupervised learning, optimization, structured and unstructured data analysis, inferential and descriptive statistics, and modeling and simulation.

Table 2 presents a set of analytics applications along with their functional goals and the analytic problem types used to achieve these goals. As illustrated in Table 2, in many cases, a functional goal can be achieved by using more than one problem types. The choice of the problem type to be used depends on many factors that include runtime constraints, underlying software and hardware infrastructure, etc. For example, customer churn analysis is a technique for predicting the customers that are most likely to leave the current service provider (retail, telecom or financial) for a competitor. This analysis can use one of the three problem types: inferential statistics, supervised learning or unstructured data analysis. One approach models individual customer's behavior using various parameters such as duration of service, user transaction history, etc. These parameters are then fed either to a statistical model such as regression or to a supervised learning model such as a decision tree, to predict if a customer is likely to defect [22]. The second approach, models behavior of a customer based on her interactions with other customers. This strategy is commonly used in the telecom sector, where customer calling patterns are used to model subscriber relationships as a graph. This unstructured graph can then be analyzed to identify subscriber groups and their influential leaders: usually the active and well-connected subscribers. These leaders can then be targeted for marketing campaigns to reduce defection in the members of her group [23].

2.2.2 The Library Phase

The library component is usually designed to be portable and broadly applicable across multiple analytic solutions (e.g., the DeepQA runtime that powers the Watson system). A library usually provides implementations of specific models of the common problem types. For example, an unsupervised learning problem can be solved using one of many models including associative mining, classification, or clustering [16]. Each model can, in turn, use one or more algorithms for its implementation. For instance, the associative rule mining model can be implemented using the different associative rule mining or decision tree algorithms. Similarly, classification can be implemented using nearest-neighbor, neural network, or naive Bayes algorithms. It should be noted that in practice, the separation between models and algorithms is not strict and many times, an algorithm can be used for supporting more than one models. For instance, neural networks can be used for clustering or classification.

2.2.3 The Implementation Phase

Finally, depending on how the problem is formulated, each algorithm uses specific data structures and kernels. For example, many algorithms formulate the problem using dense or sparse matrices and invoke kernels like matrix-matrix and matrix-vector multiplication, matrix factorization, and linear system solvers. These kernels are sometimes

[1]We have expanded the classification proposed by Davenport et al. [8, 9].

Analytical applications	Functional goals	Problem types
Supply chain management, Product scheduling, Logistics, Routing, Workforce management	Prescription	Optimization
Revenue prediction, Disease spread prediction, Semiconductor yield analysis, Predictive policing	Prediction	Unsupervised/Supervised learning Descriptive/Inferential statistics
Retail sales analysis, Financial reporting, Budgeting, System management analysis, Social network analysis	Reporting	Structured/Unstructured data analysis
VLSI sensitivity analysis, Insurance risk modeling, Credit risk analysis, Physics/Biology simulations, Games	Simulation	Modeling and simulation Descriptive/Inferential statistics
Topic/Sentiment analysis, Computational chemistry, Document management, Searching, Bio-informatics	Pattern matching	Structured/Unstructured data analysis Unsupervised/Supervised learning
Cross-sale analysis, Customer retention, Music/Video, Restaurant recommendation, Intrusion detection	Recommendation	Unsupervised/Supervised Learning Structured/Unstructured data analysis
Web-traffic analysis, Fraud detection, Geological Sensor networks, Geographical analytics (Maps)	Alerting	Descriptive/Inferential statistics Unsupervised/Supervised learning
Customer relationship analysis, Weather forecasting Econometrics, Computational finance	Quantitative analysis	Descriptive/Inferential statistics Unsupervised/Supervised learning

Table 2: Examples of analytics applications, associated functional goals, and analytical problem types

optimized for the underlying system architecture, in form of libraries such as IBM ESSL [17] or Intel MKL [18]. Any kernel implementation can be characterized according to how it manages parallel execution, if at all, and how it manages data and maps it to the system memory and I/O architecture. Many parallel kernels can use shared or distributed memory parallelism. In particular, if the algorithm is embarrassingly parallel, requires large data, and the kernel is executing on a distributed system, it can often use the MapReduce approach [10]. At the lowest level, the kernel implementation can often exploit hardware-specific features such as short-vector data parallelism (SIMD) or task parallelism on multi-core CPUs, massive data parallelism on GPUs, and application-specific parallelism using Field Programmable Gate Arrays (FPGAs).

3. ANALYTICS EXEMPLARS

Given the wide variety of algorithmic and system alternatives for executing analytics applications, it is difficult for solution developers to make the right choices to address specific performance issues. To alleviate this problem, we have analyzed the functional flow (Figure 1) of a wide set of key applications across multiple analytics domains and have isolated repeated patterns in analytical applications, algorithms, data structures, and data types. We have been using this information to optimize analytic applications and libraries for modern systems and in some cases, specialize our processor and system designs to better suit analytic applications.

Towards this goal, we have identified a set of widely-used analytical models that capture the most important computation and data access patterns of the analytics applications that we have studied [5, 27]. These models, referred to as *Analytics Exemplars*, cover the prevalent analytical problem types and each exemplar can be used to address one or more functional goals. Table 3 presents the list of thirteen exemplars, along with target functional goals and key algorithms used for implementing these exemplars.

3.1 Key Algorithms

As Table 3 illustrates, each exemplar can be implemented by one or more distinct algorithms. Some of the algorithms can be used for implementing more than one exemplars, e.g., the Naive Bayes algorithm can be used in text analytics and for general clustering purposes. Each algorithm, depending on the runtime constraints, i.e., whether the application data can fit into main memory or not, can use a variety of algorithmic kernels (Figure 1). For more details on the algorithms and their implementations, the reader is referred to [5, 36].

3.2 Computational Patterns

Table 4 presents a summary of computational patterns, key data types, data structures and functions used by algorithms for each exemplar. As Table 4 illustrates, while different exemplars demonstrate distinct computational and runtime characteristics, they also exhibit key similirities. Broadly, the analytic exemplars can be classified into two classes: the first class exploits linear-algebraic formulations and the second uses non-numeric data structures (e.g., hash tables, trees, bit-vectors, etc.). Exemplars belonging to the first class, e.g., Math-

Model Exemplar (Problem type)	Functional goals	Key algorithms
Regression analysis (Inferential statistics)	Prediction Quantitative Analysis	Linear, Non-linear, Logistic Probit regression
Clustering (Supervised learning)	Recommendation, Prediction, Reporting	K-Means and Hierarchical clustering EM Clustering, Naive Bayes
Nearest-neighbor search (Unsupervised learning)	Prediction, Recommendation	K-d, Ball, and Metric trees, Approx. Nearest-neighbor Locality-sensitive Hashing, Kohonen networks
Association rule mining (Unsupervised learning)	Recommendation	Apriori, Partition, FP-Growth, Eclat and MaxClique, Decision trees
Neural networks (Supervised learning)	Prediction Pattern matching	Single- and Multi-level perceptrons, RBF, Recurrent, and Kohonen networks
Support Vector Machines (Supervised learning)	Prediction Pattern matching	SVMs with Linear, Polynomial, RBF, Sigmoid, and String kernels
Decision tree learning (Supervised learning)	Prediction Recommendation	ID3/C4.5, CART, CHAID, QUEST
Time series processing (Data analysis)	Pattern matching, Alerting	Trend, Seasonality, Spectral analysis, ARIMA, Exponential smoothing
Text analytics (Data analysis)	Pattern matching Reporting	Naive Bayes classifier, Latent semantic analysis, String-kernel SVMs, Non-negative matrix factorization
Monte Carlo methods (Modeling and simulation)	Simulation Quantitative analysis	Markov-chain, Quasi-Monte Carlo methods
Mathematical programming (Optimization)	Prescription Quantitative analysis	Primal-dual interior point, Branch & Bound methods, Traveling salesman, A* algorithm, Quadratic programming
On-line analytical processing (Structured data analysis)	Reporting Prediction	Group-By, Slice_and_Dice, Pivoting, Rollup and Drill-down, Cube
Graph analytics (Unstructured data analysis)	Pattern matching Recommendation	Eigenvector Centrality, Routing, Coloring, Searching and flow algorithms, Clique and motif finding

Table 3: Analytics exemplar models, along with problem types and key application domains

ematical Programming, Regression Analysis, and Neural Networks, operate primarily on matrices and vectors. Matrices are either sparse or dense, and are used in various linear algebraic kernels like the matrix multiplication, inversion, transpose, and factorization. The second class, which includes clustering, nearest-neighbor search, associative rule mining, decision tree learning, use data structures like hash-tables, queues, graphs, and trees, and operate on them using set-oriented, probabilistic, graph-traversal, or dynamic programming algorithms. Exemplars like mathematical programming, text analytics, and graph analytics can use either of these approaches. The analytic exemplars use a variety of types, such as integers, strings, bit-vector, and single and double precision floats, to represent the application data. This information is then processed using different functions that compare, transform, and modify input data. Examples of common analytic functions include various distance functions (e.g., Euclidian), kernel functions (e.g., Linear, Sigmoid), aggregation functions (e.g., Sum), and Smoothing functions (e.g., correlation). These functions, in turn, make use of intrinsic library functions such as `log`, `sine` or `sqrt`.

3.3 Runtime Characteristics

Table 5 summarizes the runtime characteristics of the analytics exemplars. The key distinguishing feature of analytics applications is that they usually process input data in read-only mode. The input data can be scalar, structured with one or more dimensions, or unstructured, and is usually read from files, streams or relational tables in the binary or text format. In most cases, the input data is large, which requires analytics applications to store and process data from disk. Notable exceptions to this pattern are Monte Carlo Methods and Mathematical Programming, which are inherently in-memory as they operate on small input data. The results of analysis are usually smaller than the input data. Only two exemplars, association rule mining and on-line analytical processing (OLAP) generate larger output. Finally, analytics applications can involve one or more stages (real-time execution can be considered to have only one stage), where each stage invokes the corresponding algorithm in an iterative or non-iterative manner. For the iterative workloads, for the same input data size, the running time can vary depending on the precision required in the results.

4. SYSTEM IMPLICATIONS

Given the varied computational and runtime characteristics of the analytics exemplars, it is clear that a single systems solution for different analytics ap-

Model Exemplar	Computational pattern	Key data types, Data structures, and Functions
Regression Analysis	Matrix inversion, LU decomposition Transpose, Factorization	Double-precision and Complex data Sparse/Dense matrices, Vectors
Clustering	Metric-based iterative convergence	Height-balanced tree, Graph, Distance functions, `log` function
Nearest-Neighbor Search	Non-iterative distance calculations Singular value decomposition, Hashing	Higher-dimensional data structures, Hash tables, Distance functions
Rule Mining	Set intersections, Unions, and Counting	Hash-tree, Prefix trees, Bit vectors
Neural Networks	Iterative Feedback networks Matrix multiplication, Inversion, Factorization	Sparse/dense matrices, Vectors, Double-precision/Complex data Smoothing functions
Support Vector Machines	Factorization, Matrix multiplication	Double-precision Sparse matrices, Vectors Kernel functions (e.g., Linear)
Decision Trees	Dynamic programming Recursive Tree Operations	Integers, Double-precision, Trees, Vectors, `log` function
Time Series Processing	Smoothing via averaging, Correlation Fourier and Wavelet transforms	Integers, Single-/Double-precision, Dense matrices Vectors, Distance and Smoothing functions
Text Analytics	Parsing, Bayesian modeling, String matching Hashing, Singular value decomposition Matrix multiplication, Transpose, Factorization	Integers, Single-/Double-precision, Strings Sparse matrices, Vectors, Inverse indexes, String functions, Distance functions
Monte Carlo Methods	Random number generators Polynomial evaluation, Interpolation	Double-precision, Bit vectors Bit-level operations, `log`, `sqrt` functions
Mathematical Programming	Matrix multiplication, Inversion, Factorization Dynamic programming, Greedy algorithms, Backtracking-based search	Integers, Double-precision, Sparse Matrices, Vectors, Trees, Graphs
On-line Analytical Processing	Grouping and ordering Aggregation over hierarchies	Prefix trees, Relational tables, OLAP Operators Sorting, Ordering, Aggregation operators
Graph Analytics	Graph traversal, Eigensolvers, Matrix-vector, Matrix-matrix multiplication, Factorization	Integer, Single-/Double-precision, Adjacency Lists Trees, Queues, Dense/Sparse matrices

Table 4: Computational characteristics of the analytics exemplars

Model Exemplar	Execution characteristics		Input-Output characteristics	
	Methodology	Memory Issues	**(Read-only)** Input Data	Output Data
Regression Analysis	Iterative	In-memory Disk-based	Large historical Structured	Small Scalar
Clustering	Iterative	In-memory Disk-based	Large historical Unstructured or structured	Small scalar Unstructured or structured
Nearest-Neighbor Search	Non-iterative	In-memory	Large historical Structured	Small Scalar or structured
Association Rule Mining	Iterative Non-iterative	In-memory Disk-based	Large historical Structured	**Larger** Structured
Neural Networks	Iterative Two Stages	In-memory Disk-based	Large Structured	Small Scalar
Support Vector Machines	Iterative Two Stages	In-memory Disk-based	Large Structured	Small Scalar
Decision Tree Learning	Iterative Two Stages	In-memory Disk-based	Large Structured & Unstructured	Small Scalar
Time Series Processing	Non-iterative Real-time	In-memory	High volume streaming Structured or unstructured	Small scalar or streaming Structured or unstructured
Text Analytics	Iterative Non-iterative	In-memory Disk-based	Large historical or streaming Structured or unstructured	Large or small Structured or unstructured
Monte Carlo s Methods	Iterative	In-memory	Small Scalar	Large Scalar
Mathematical Programming	Iterative	In-memory	Small Scalar	Small Scalar
On-line Analytical Processing (OLAP)	Non-iterative	In-memory Disk-based	Large historical Structured	**Larger** Structured
Graph Analytics	Iterative	In-memory Disk-based	Large historical Unstructured	Small Scalar or unstructured

Table 5: Runtime characteristics of the analytics exemplars

plications would be sub-optimal. As Tables 4 and 5 demonstrate, each exemplar has a unique set of computational and runtime features, and ideally, every exemplar would get a system tailor-made to match its requirements. However, we have also observed that different analytic exemplars share many computational and runtime features. Therefore, for a systems designer, the challenge is to customize analytics systems using as many re-usable software and hardware components as possible.

4.1 System Acceleration Opportunities

Table 6 describes system opportunities for accelerating analytics exemplars. Based on the computational and runtime characteristics described in Tables 4 and 5, we first identify key bottlenecks in the execution of analytic exemplars, namely compute-bound, memory-bound, and I/O bound (which covers both disk and network data traffic). As Table 6 illustrates, a majority of the analytics exemplars are compute bound in the in-memory mode and I/O-bound when in the disk-based mode. The compute-bound exemplars can benefit from traditional task-based parallelization approaches on multi-core processors, as well as by hardware-based acceleration via SIMD instructions or using GPUs. When used in the disk-based scenarios, these exemplars can improve their I/O performance by using solid state drives or data compression. Some of the analytics exemplars are memory-bound due to their reliance on algorithms that traverse large in-memory data structures such as trees or sparse matrices. For these exemplars, a better memory sub-system, with faster, larger, and deeper memory hierarchies, would be most beneficial. Once the memory accesses are optimized, these exemplars can also benefit from traditional computational acceleration techniques. Finally, some of the exemplars exhibit unique computational patterns (e.g., bit-level manipulations, pattern matching, or string processing) which could be accelerated using special-purpose processors such as FPGAs or by introducing new instructions in general-purpose processors. In most cases, the exemplars can be accelerated using commodity hardware components (e.g., multi-core processors, GPUs or SSDs). These hardware components can be then used to optimize re-usable software kernel functions (e.g., numerical linear algebra, distance functions, etc.), which themselves can be parallelized by a variety of parallelization techniques such as task parallelism, distributed-memory message-passing parellelism or MapReduce [26, 2]. These functions can be used as a basis of specialized implementations of the exemplars. Such hardware-software co-design enables optimized analytics solutions that can balance customization and commoditization.

4.2 The Netezza Example

An example of hardware-software co-design for database workloads is the Netezza data warehouse and analytics appliance [11]. The Netezza appliance supports both SQL-based OLAP and analytics queries. Netezza uses a combination of FPGA-based acceleration and customized software to optimize data-intensive mixed database and analytics workloads with concurrent queries from thousands of users. The Netezza system uses two key principles to achieve scalable performance: (1) Reduce unnecessary data traffic by moving processing closer to the data, and (2) Use parallelization techniques to improve the processing costs. A Netezza appliance is a distributed-memory system with a host server connected to a cluster of independent servers called the snippet blades (S-Blades). A Netezza host first compiles a query using a cost-based query optimizer that uses the data and query statistics, along with disk, processing, and networking costs to generate plans that minimize disk I/O and data movement. The query compiler generates executable code segments, called snippets which are executed in parallel by S-blades. Each S-blade is a self-contained system with multiple multi-core CPUs, FPGAs, gigabytes of memory, and a local disk subsystem. For a snippet, the S-Blade first reads the data from disks into memory using a technique to reduce disk scans. The data streams are then processed by FPGAs at wire speed. In a majority of cases, the FPGAs filter data from the original stream, and only a tiny fraction is sent to the S-Blade CPUs for further processing. The FPGAs can also execute some additional functions which include decompression, concurrency control, projections, and restrictions. The CPUs then execute either database operations like sort, join, or aggregation or core mathematical kernels of analytics applications on the filtered data streams. Results from the snippet executions are then combined to compute the final result. The Netezza architecture also supports key data mining and machine learning algorithms on numerical data (e.g., matrices) stored in relational tables.

A key lesson learned from the design of Netezza has been the huge value of specializing system design for analytics. Orders of magnitude improvements in efficiency can be achieved by carefully analyzing the system requirements and innovating using a collaborative software-hardware design methodology. As analytics applications become more main-

Model Exemplar	Bottleneck	Acceleration requirements and opportunities
Regression Analysis Clustering Nearest-Neighbor Search Neural Networks Support Vector Machines	Compute-bound I/O-bound	Shared- and Distributed-memory task parallelism Data parallelism via SIMD or GPUs Faster I/O using solid state drives
Association Rule Mining	I/O-bound	Shared-memory task parallelism Faster I/O using solid state drives Faster bit operations or tree traversals via FPGAs
Decision Tree Learning	Memory-bound	Larger and deeper memory hierarchies Data parallelism via SIMD
Time Series Processing	Compute-bound Memory-bound	Shared- and Distributed-memory task parallelism Data parallelism via SIMD or GPUs High-bandwidth, low-latency memory subsystem Pattern matching via FPGA
Text Analytics	Memory-bound I/O-bound	Shared- and Distributed-memory task parallelism Data parallelism via SIMD or GPUs Larger and deeper memory hierarchies Faster I/O via solid state drives Pattern matching and string processing via FPGA
Monte Carlo Methods	Compute-bound	Shared- and Distributed-memory task parallelism Data parallelism via SIMD or GPUs Faster bit manipulations using FPGAs or ASICs
Mathematical Programming	Compute-bound	Shared-memory task parallelism Massive data-parallelism via GPUs Larger and deeper memory hierarchies Search-tree traversals via FPGAs
On-line Analytical Processing	Memory-bound I/O-bound	Shared- and Distributed-memory task parallelism Data parallelism via SIMD or GPUs Larger and deeper memory hierarchies Pattern Matching via FPGAs, Faster I/O using solid state drives
Graph Analytics	Memory-bound	Shared-memory task parallelism Larger and deeper memory hierarchies

Table 6: Opportunities for parallelizing and accelerating analytics exemplars

stream, future database systems need to be designed in an integrated manner to support both the classical and analytics workoads.

5. SUMMARY

In this survey paper and the accompanying research report [5], we have reviewed the growing field of analytics that uses mathematical formulations to solve business and consumer problems. We have identified some of the key techniques employed in analytics, called *analytics exemplars*, both to serve as an introduction for the non-specialist, and to explore the opportunity for greater optimization for parallel computer architectures, and systems software. We hope this work spurs follow-on work on analyzing and optimizing analytics workloads.

6. REFERENCES

[1] A. P. Armacost, C. Barnhart, K. A. Ware, and A. M. Wilson. UPS Optimizes Its Air Network. *Interfaces*, 34(1), January-February 2004.

[2] R. Bekkerman, M. Bilenko, and J. Langford, editors. *Scaling Up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press, 2011. To appear.

[3] R. M. Bell, Y. Koren, and C. Volinsky. All Together Now: A Perspective on the Netflix Prize. *Chance*, 23(1):24–29, 2010.

[4] I. Bhattacharya, S. Godbole, A. Gupta, A. Verma, J. Achtermann, and K. English. Enabling analysts in managed services for CRM analytics. In *In Procs. of the 2009 ACM KDD Intl. Conf. on Knowledge and Data Discovery*, 2009.

[5] R. Bordawekar, B. Blainey, C. Apte, and M. McRoberts. Analyzing analytics, part 1: A survey of business analytics models and algorithms. Technical Report RC25186, IBM T. J. Watson Research Center, July 2011.

[6] R. Cantor, F. Packer, and K. Cole. Split ratings and the pricing of credit risk. Technical Report Research Paper No. 9711, Federal Reserve Bank of New York, March 1997.

[7] P. Crosbie and J. Bohn. Modeling default risk, December 2003. Moody's KMV.

[8] T. Davenport and J. Harris. *Competing on Analytics, The New Science of Winning*. Harvard Business School Press, 2007.

[9] T. Davenport, J. Harris, and R. Morison. *Analytics at Work, Smarter Decisions, Better Results*. Harvard Business School Press, 2010.

[10] J. Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. *Communications of the ACM*, 53(1), 2010.

[11] D. J. Feldman. Netezza Performance Architecture. Keynote Presentation at the DaMoN'13 Workshop.

[12] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 59(Fall), 2010.

[13] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, Computer Science Department, Stanford University, December 2009.

[14] E. Goode. Sending the police before there's a crime. The New York Times, August 16, 2011.

[15] P. Grosse, W. Lehner, T. Weichert, F. Faerber, and W.-S. Li. Bridging Two Worlds with RICE: Integrating R into the SAP In-Memory Computing Engine. In *Proc. of the VLDB Endowment*, September 2011.

[16] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2006.

[17] IBM Corp. Engineering and Scientific Subroutine Library (ESSL) and Parallel ESSL. `www.ibm.com/systems/software/essl`.

[18] Intel Inc. Intel Math Kernel Library. `software.intel.com`.

[19] J. Joyce. Pandora and the Music Genome Project. *Scientific Computing*, 23(10):40–41, September 2006.

[20] M. Lohatepanont and C. Barnhart. Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. *Transportation Science*, 38(1), February 2004.

[21] G. Mohler, M. Short, P. Brantingham, F. Schoenberg, and G. Tita. Self-exciting point process modeling of crime. *Journal of American Statistical Association*, 106(493), March 2011.

[22] T. Mutanen. Customer churn analysis- a case study. Technical Report VTT-R-01184-06, Technical Research Centre of Finland, 2006.

[23] A. A. Nanavati, S. Gurumurthy, G. Das, D. Chakraborty, K. Dasgupta, S. Mukherjea, and A. Joshi. On the structural properties of massive telecom call graphs: Findings and implications. In *Proc. of the Conference on*

Information and Knowledge Management (CIKM'06), pages 435–444, November 2006.

[24] E. W. T. Ngai, L. Xiu, and D. C. K. Chau. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36:2592–2602, 2009.

[25] Oracle Inc. Oracle Advanced Analytics, 2013. Oracle Data Sheet.

[26] A. Rajaraman and J. Ullman. *Mining Massive Datasets*. Cambridge University Press, 2010. Free version available at `infolab.stanford.edu/~ullman/mmds.html`.

[27] K. Rexer, H. Allen, and P. Gearan. 2010 data miner survey summary. In *Procs. of the Predictive Analytics World*, October 2010.

[28] Y. Richter, E. Yom-Tov, and N. Slonim. Predicting customer churn in mobile networks through analysis of social groups. In *Procs. of the SIAM International Conference on Data Mining, SDM 2010*, pages 732–741, 2010.

[29] Science Special Issue. Dealing with data. *Science*, 331(6018), February 2011.

[30] V. Sindhwani, A. Ghoting, E. Ting, and R. Lawrence. Extracting insights from social media with large-scale matrix approximations. *IBM Journal of Research and Development*, 55(5):9:1–9:13, Sept-Oct 2011.

[31] Sony Pictures Inc. Jeopardy! The IBM Challenge. `www.jeopardy.com/minisites/watson`.

[32] Splunk Inc. Splunk Tutorial, 2011. `www.splunk.com`.

[33] A. Suman. Automated face recognition, applications within law enforcement: Market and technology review, October 2006.

[34] The Economist. Algorithms: Business by numbers. Print Edition, 13th September, 2007.

[35] The Economist. Data, data everywhere. Print Edition, 25th February, 2010.

[36] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, H. David J, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge Information Systems*, 14:1–37, 2008.

A Survey on Tree Edit Distance Lower Bound Estimation Techniques for Similarity Join on XML Data

Fei Li
Harbin Institute of Technology
lifei@umich.edu

Hongzhi Wang*
Harbin Institute of Technology
wangzh@hit.edu.cn

Jianzhong Li
Harbin Institute of Technology
lijzh@hit.edu.cn

Hong Gao
Harbin Institute of Technology
honggao@hit.edu.cn

ABSTRACT

When integrating tree-structured data from autonomous and heterogeneous sources, exact joins often fail for the same object may be represented differently. Approximate join techniques are often used, in which similar trees are considered describing the same real-world object. A commonly accepted metric to evaluate tree similarity is the tree edit distance. While yielding good results, this metric is computationally complex, thus has limited benefit for large databases. To make the join process efficient, many previous works take filtering and refinement mechanisms. They provide lower bounds for the tree edit distance in order to reduce unnecessary calculations. This work explores some widely accepted filtering and refinement based methods, and combines them to form multi-level filters. Experimental results indicate that string-based lower bounds are tighter yet more computationally complex than set-based lower bounds, and multi-level filters provide the tightest lower bound efficiently.

1. INTRODUCTION

For the ability to represent data from heterogeneous sources, XML is widely used for web data representation and exchange. For its flexibility, data representing the same object may not be exactly the same. For duplication detection and data integration, approximate join techniques are in demand. That is, similar XML fragments are joined for they are considered as representing the same real-world object.

XML fragments are often modeled as ordered labeled trees. Tree edit distance is a widely used metric to evaluate the similarity between trees [17]. The tree edit distance is the minimum number of node insertions, deletions, or relabels to transform one tree to another [1]. Two trees are considered as a *similar tree pair* if their tree edit distance is below a predefined threshold. It is effective but computationally expensive. Many researches have been performed to improve the efficiency [22, 14, 7, 8]. Unfortunately, the time complexity is still at least $O(n^3)$, where n is the tree size. When there are large numbers of trees and the trees are huge, the join process needs a lot of time.

Filtering and refinement mechanisms are often used to overcome this problem. The main idea is to compute lower bounds for tree edit distances and filter out *dissimilar tree pairs* without computing their exact tree edit distances. Since lower bounds are much easier to compute than the exact value, the overall efficiency is improved significantly.

To our knowledge, existing lower bounds are computed based on transformation. Trees are transformed into other data structures whose distances serve as lower bounds to tree edit distance. String is a relatively simple data structure which contains order for structure as well as content information in each entry. In [10], XML documents are transformed into their corresponding preorder and postorder traversal sequences. Then the string edit distance is used as the lower bound of the tree edit distance. This method has high filter quality but relatively low efficiency. Set (multi-set) is even simpler than string. In [12], three kinds of histograms are proposed based on the node height (leaf height), node degree, and node label, respectively, to compute relatively rough lower bounds. In the method of binary branch [21], trees are transformed into binary branch sets and the binary branch distance between these sets is used to compute the lower bound of the tree edit distance. These two set-based methods are

*corresponding author

[1]In this article, we mainly discuss unit cost tree edit distance, in which all operations have the same cost.

very efficient but can not provide the lower bound as tight as the string-based methods do.

Since all the lower bounds of tree edit distance are definitely lower than or equal to the exact tree edit distance, these methods can be combined to give tighter lower bounds. The maximum value of all the lower bounds in different methods serves as the tightest lower bound. Instead of computing all the lower bounds independently, a multi-level filtering mechanism can be applied. Efficient lower bounds are computed first to wipe some dissimilar tree pairs out. Then more expensive yet tighter lower bound are computed only for the remaining tree pairs. After all the lower bound methods are applied, tree edit distance is compute for the remaining tree pairs. While having the same filtering quality, multi-level filter is conducted more efficiently than computing all lower bounds independently and choosing the highest one.

Contributions: This paper presents a comparative study of these filtering and refinement methods for tree similarity join. We implement the string-based lower bounds [10], Histogram [12], and binary branch distance [21] respectively to test the bound tightness and computational efficiency. From the comparisons, each of these three methods has special benefits. As a result, they could be combined to form a multi-level filter to achieve tighter lower bound efficiently. Such a combined mechanism could be more effective and efficient than each single one.

The rest of the paper is organized as follows. In Section 2, related work is discussed. In Section 3, some background knowledge is introduced. Three widely accepted methods in computing the lower bound of tree edit distance are described in detail in Section 4. We analyze the properties of each method in Section 5. The combined strategy is discussed in Section 6. We test the efficiency and effectiveness of each method experimentally in Section 7. The conclusions are drawn in Section 8.

2. RELATED WORK

Approximate joining techniques for trees are often based on similarity evaluation. A well-known distance function for trees is the tree edit distance. To describe time complexity, we use n, l, and h to denote the number of nodes, leaves, and the height of a tree, respectively. [17] presented the first algorithm for computing tree edit distance in time $O(n^2l^4)$. [22] improved this result to $O(n^2min^2(l,h))$ running time with $O(n^4)$ in the worst case. [14] improved it to $O(n^3 \log n)$. Both [22] and [14] achieved their improvements based on closely related dynamic programming, presenting different ways to compute only a subset of relevant subproblems. [7] presented a different approach based on the results of fast matrix multiplication and give an algorithm with time complexity $O(n^{3.5})$ in the worst case. A recent development is by [8] which compute the tree edit distance in time $O(n^3)$.

Obviously, the tree edit distance computation is expensive and does not scale for large trees in massive data-sets. Therefore, many previous works take the filtering and refinement mechanisms to accelerate the similarity join process. In the filtering step, many pairs of dissimilar trees are filtered out. In the refinement step, tree edit distance is only computed for the remaining tree pairs. The overall join process is accelerated since fewer tree edit distances need to be computed directly.

To the best of our knowledge, existing filtering and refinement methods are based on transformation. Trees are transformed into simpler data structures whose distance is lower than the tree edit distance but much easier to compute. String is a relatively simple data structure that contains order for structure information as well as content information in each entry. In [10], XML documents are transformed into their corresponding preorder (or postorder) traversal sequences. Then the string edit distance between two sequences serves as the lower bound of their tree edit distance. [2, 1] use half of the string edit distance between Euler traversals as the lower bound of the tree edit distance. However, this lower bound is often lower than the maximum of the two lower bounds (provided by their preorder traversal sequences and postorder traversal sequences, respectively) proposed in [10], thus cannot be tighter lower bounds in most cases. The Euler traversal is twice as long as the preorder (or postorder) traversal, which would cause 4 times in running time. So we use [10] to represent string based lower bounds.

Set (multi-set) is a even simpler data structure. In Histogram [12], three kinds of histograms are proposed based on the node height, node degree, and node label, respectively, to compute rough lower bounds for tree edit distance. Another set-based method is Binary branch [21]. In that method, trees are first transformed into binary trees and then into sets. The binary branch distance between these sets is then used to compute the lower bound of the tree edit distance.

Recently, some works adopted different distance functions to evaluate the similarity between trees directly. *pq*-gram distance is first proposed to evaluate the distance between ordered trees directly [4].

In [3], the *pq*-gram method is extended to evaluate the similarity between unordered trees. Recently in [18], each tree is transformed into a set of pivots and the Jaccard Coefficient between two sets of pivots are used to approximate the tree edit distance. As is shown in [18], for unordered trees, their method approximates tree edit distance more accurately than *pq*-gram. In the case of ordered trees, their matching quality is lower than that using *pq*-gram [11]. These methods are proposed to evaluate the similarity between trees directly. Although some of them approximate tree edit distance well, they do not have any guarantee of being lower bounds to tree edit distance. Hence we do not discuss these methods in this paper. Later in [5], the *pq*-gram distance is modified to serve as a lower bound to the fanout-weighted tree edit distance, but not to the widely used unit cost tree edit distance or general case. We do not consider it in this paper.

3. PRELIMINARY DEFINITIONS

Definition 1. (Tree Edit Distance). *Given a pair of trees T_1 and T_2, the tree edit distance between them is the minimum cost of a series of tree edit operations to transform one into another. The three standard tree edit operations [17] includes:*

1. *relabeling (changing the label) a node v.*
2. *deleting a node v (and moving all the children of v to v's parent).*
3. *inserting a node v to w (and moving a contiguous sequence of w's children under v).*

In order to determine the distance between trees, a cost model must be defined. In this paper, we discuss the unit cost model: the cost of each standard operation is 1. We use the symbol $TD(T_1, T_2)$ to denote the unit tree edit distance between T_1 and T_2.

Definition 2. (Approximate Join on Trees). *]Given two tree sets, F_1 and F_2, the Join between F_1 and F_2 on Tree Edit Distance is the set $\{(T_i, T_j)|(T_i, T_j) \in F_1 \times F_2, TD(T_i, T_j) \leq \tau\}$, where τ is a predefined threshold.*

Example 1. *Figure 1 shows two tree sets $F_1 = \{T_{11}, T_{12}\}$ and $F_2 = \{T_{21}, T_{22}\}$. Suppose the predefined threshold is 2. Only $TD(T_{11}, T_{21})$ and $TD(T_{12}, T_{22})$ are lower or equal to that threshold. Then the tree edit distance join on them is $\{(T_{11}, T_{21}), (T_{12}, T_{22})\}$.*

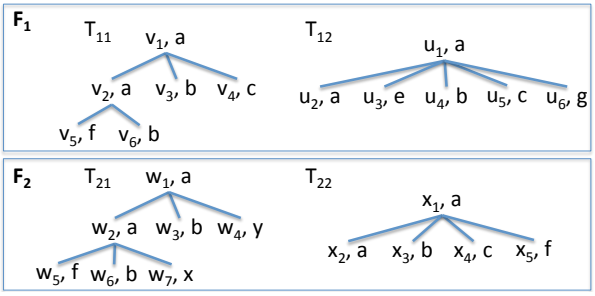


Figure 1: Approximate Tree Matching

Join based on tree edit distance is effective but computationally expensive. To accelerate the efficiency, filtering and refinement mechanisms are often used. That is, if the lower bound of the tree edit distance is above the predefined threshold, that tree pair must be dissimilar and can be safely eliminated. Since lower bounds are much easier to be computed than the tree edit distance, the whole join efficiency is improved significantly.

4. LOWER BOUNDS FOR TREE EDIT DISTANCE

In this section, we introduce three commonly accepted methods for computing the lower bounds of tree edit distance: string-based lower bound [10], histogram [12] and binary branch distance [21].

4.1 String-based Lower Bound

Let T be an ordered labeled tree, where $pre(T)$ and $post(T)$ are the preorder and postorder traversals of T, respectively. Both $pre(T)$ and $post(T)$ are viewed as strings. With $ed(s_1, s_2)$ denoting the edit distance between two strings, the relationship between the unit tree edit distance and the string edit distance is shown as follows:

$$ed(pre(T_1), pre(T_2)) \leq TD(T_1, T_2)$$

$$ed(post(T_1), post(T_2)) \leq TD(T_1, T_2)$$

Example 2. *Figure 2 shows the preorder and postorder of T_{12} and T_{21} in Figure 1. Suppose the predefined threshold is 2. Since $ed(pre(T_{12}), pre(T_{21}))$ = 4 and $ed(post(T_{12}), post(T_{21}))$ = 6, $TD(T_{12}, T_{21})$ is at least 6. So T_{12} and T_{21} are definitely dissimilar.*

String edit distance is computed in time $O(n^2)$, where n is the tree size. This is much faster than computing the tree edit distance. However, when the trees in the databases is too large, the computation of the string edit distance is also costly. Vectors and sets (bags) are data structures even simpler

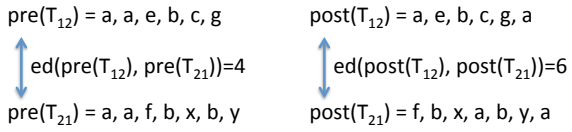


Figure 2: String-based Lower Bound

than strings. So many following researches transform trees into vectors or sets to estimate the lower bound faster [12, 21, 5].

4.2 Histogram

Histogram was firstly proposed in [12] to compute lower bounds for the tree edit distance efficiently. In their method, three kinds of histograms (leaf distance histogram, degree histogram, and label histogram) are developed. The basic idea of all these methods is to transform trees into vectors and use the L_1 distance between these vectors to estimate the lower bounds.

4.2.1 Leaf Distance Histogram

The height of the nodes in a tree is an important structural property. Leaf distance histogram defines the height from the leaves to the root as follows:

DEFINITION 3. (LEAF DISTANCE). *The leaf distance $d_l(v)$ of a node v is the maximum length of a path from v to any leaf node in the subtree rooted at v.*

DEFINITION 4. (LEAF DISTANCE HISTOGRAM). *The leaf distance histogram $h_l(T)$ of a tree T is a vector of length $k = 1 + height(T)$ where the value of any entry $i \in 0, ..., k$ is the number of nodes that share the leaf distance i, i.e. $h_l(T)[i] = |v \in T, d_l(v) = i|$.*

THEOREM 1. *For any two trees T_1 and T_2, the L_1-distance of the leaf distance histogram is a lower bound of the edit distance between T_1 and T_2 [12]. That is:*

$$L_1(h_l(T_1), h_l(T_2)) \leq TD(T_1, T_2).$$

EXAMPLE 3. *Figure 3 shows the leaf distance histogram of T_{12} and T_{21} in Figure 1. Take T_{21} as an example. T_{21} has 5 nodes with leaf distance 5 (w_3, w_4, w_5, w_6, w_7), 1 node with leaf distance 1 (w_2), and 1 node with leaf distance 2 (w_1). So the leaf distance histogram of T_{21} is (5, 1, 1). Suppose the predefined threshold is 2. If we use leaf distance histogram to compute a lower bound, which is $L_1(h_l(T_{12}), h_l(T_{21})) = 1$, we can not filter this dissimilar tree pair off.*

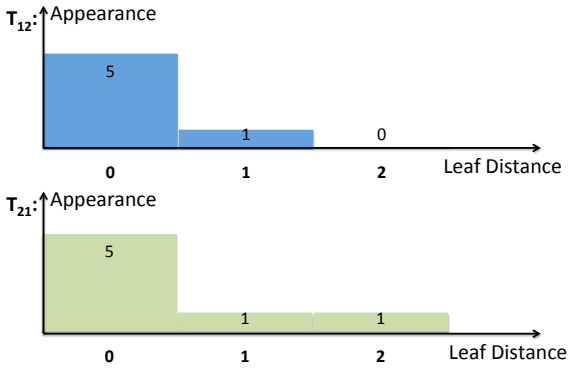


Figure 3: Leaf Distance Histogram

4.2.2 Degree Histogram

The degrees of the nodes are another structural property. The degree histogram uses the information of node degrees and gives a rough lower bound for the unit tree edit distance.

DEFINITION 5. (DEGREE HISTOGRAM). *The degree histogram $h_d(T)$ of a tree T is a vector with length $k = 1 + degree_{max}(T)$ where the value of any entry $i \in 0, ..., k$ is the number of nodes that share the degree i, i.e. $h_d(T)[i] = |v \in T, degree(v) = i|$.*

THEOREM 2. *$L_1(h_d(T_1), h_d(T_2))/3$ provide a lower bound for the edit distance between two trees T_1 and T_2 [12]. That is:*

$$\frac{L_1(h_d(T_1), h_d(T_2))}{3} \leq TD(T_1, T_2).$$

EXAMPLE 4. *Figure 4 shows the degree histogram of T_{12} and T_{21} in Figure 1. Take T_{21} as an example. T_{21} has 5 nodes in 0 degree (w_3, w_4, w_5, w_6, w_7), 2 nodes in 3 degree (w_1, w_2). So the degree histogram of T_{21} is (5, 0, 0, 2, 0, 0). Suppose the predefined threshold is 2. Since $\frac{L_1(h_d(T_{12}), h_d(T_{21}))}{3} = 3/3 = 1$, we can only tell that the tree edit distance between T_{12} and T_{21} is at least 1. In this example, similar to leaf distance histogram, degree histogram cannot filter this tree pair off either.*

4.2.3 Label Histogram

Apart from the structure information, the content features, which are stored as tree labels, can also be used to distinguish dissimilar trees. Intuitively, if two trees share many labels, they are very likely to be similar.

DEFINITION 6. (LABEL HISTOGRAM). *The label histogram $h_{lab}(T)$ of a tree T is a (multi-)set consists of all the node labels in T.*

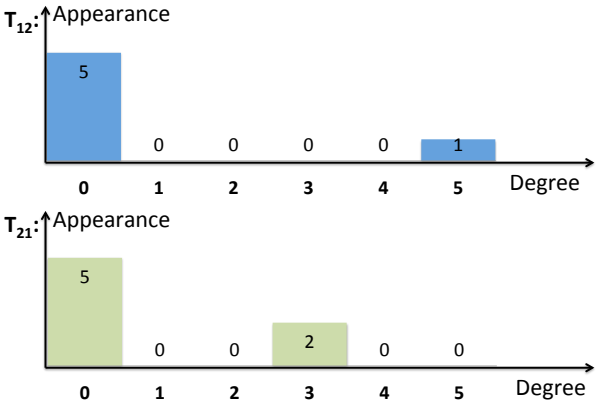


Figure 4: Degree Histogram

THEOREM 3. *For two trees T_1 and T_2 [12]:*

$$\frac{L_1(h_{lab}(T_1), h_{lab}(T_2))}{2} \leq TD(T_1, T_2).$$

EXAMPLE 5. *The label histogram of T_{12} in Figure 1 is {a, a, e, b, c, g}, while the histogram of T_{21} is {a, a, b, y, f, b, x}. Suppose the predefined threshold is 2. The lower bound provided by label histogram is $|h_{lab}(T_{12}) \bigcup h_{lab}(T_{21})$ - $h_{lab}(T_{12}) \bigcap h_{lab}(T_{21})|/2 = 4$, which is higher than the threshold. In this example, label histogram successfully filters this tree pair off.*

Label histogram can effectively filter the tree pairs whose node labels are very different. And, in most cases, label histogram can provide a much tighter lower bound than leaf distance histogram and degree histogram.

For a pair of trees with n nodes, height h and degree d, the length of their leaf distance histogram and degree histogram is $h + 1$ and $d + 1$, respectively. Thus their L_1 distance can be computed in time $O(h)$ and $O(d)$. Also, the size of each label histogram is n, thus the symmetric difference between them can be computed in time $O(n \log n)$. Furthermore, in the case of similarity join two tree sets, the efficiency can be further enhanced when applying some well-known techniques (e.g., sort merge and hash join) to avoid nested loop. So all the three kinds of histograms can give rough lower bounds and wipe out some dissimilar trees very efficiently.

4.3 Binary Branch Distance

Leaf distance histogram and degree histogram consider only structural information while the label histogram considers only content information. Thus they can only give relatively rough lower bounds. Binary branch [21] is a set-based method which considers both structure and content information at the same time.

There is a natural correspondence between a tree and its binary tree. For each node in a tree, its left most child (if any) becomes it left child in its binary tree while its right sibling (if any) becomes its right child in its binary tree. In this paper, we use the symbol $B(T)$ to denote the binary tree transformed from T.

EXAMPLE 6. *Figure 5 shows binary trees $B(T_{12})$ and $B(T_{21})$ of T_{12} and T_{21}, respectively. Note that we further transform each binary tree into a full binary tree by adding dummy nodes (labeled $*$).*

DEFINITION 7. (BINARY BRANCH). *Let B be a binary tree. $\forall u \in B$ has a binary branch $Br(u)$ composed by u and its two children.*

DEFINITION 8. (BINARY BRANCH VECTOR). *A binary branch vector $BRV(T)$ of a tree T is a vector $(b_1, b_2, ..., b_{|B|})$, with each element b_i representing the number of occurrences of the ith binary branch. $|B|$ is the size of the binary branch space of the dataset.*

DEFINITION 9 (BINARY BRANCH DISTANCE). *Let $BRV(T_1)$=$(b_1, b_2, ..., b_{|B|})$, $BRV(T_2) = (b'_1, b'_2, ..., b'_{|B|})$ be the binary branch vectors of tree T_1 and T_2, respectively. Their binary branch distance is $BDist(T_1, T_2) = \Sigma_{i=1}^{B} |b_i - b'_i|$.*

THEOREM 4. *For any two trees T_1 and T_2 [21]:*

$$\frac{BDist(T_1, T_2)}{5} \leq TD(T_1, T_2).$$

EXAMPLE 7. *Figure 6 shows all binary branches of T_{12} and T_{21}. Their corresponding binary branch vectors are shown in Figure 7. The binary branch distance between the two binary branch vectors is 11. Thus the estimate lower bound is 3.*

5. ANALYSIS

5.1 Running Time

In the previous section, we describe altogether 6 lower bound functions: two string-based lower bounds and four set-based lower bounds. For string-based lower bounds, the computation of string edit distance needs potentially quadratic time. The latter 4 distance function compute the L_1 distance between vectors, which is equal to compute the symmetric difference between the (multi-)sets of the entries in these vectors. The symmetric difference between the (multi-)sets can be computed in time $O(n \log n)$, which is much faster than the $O(n^2)$

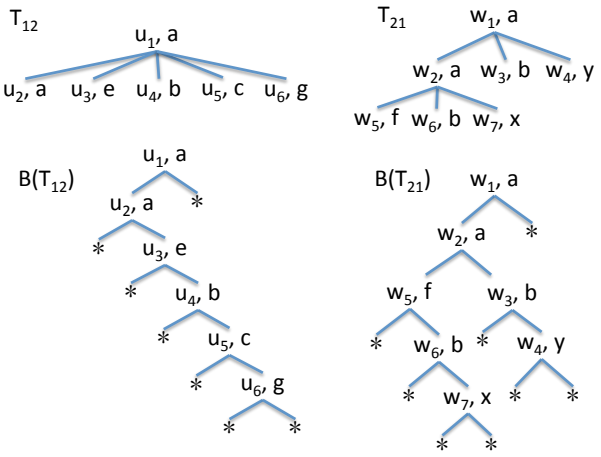


Figure 5: Binary Tree of T_{12} and T_{21}

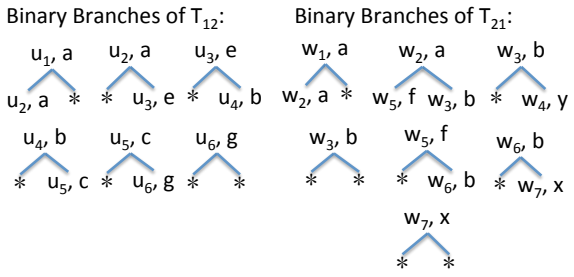


Figure 6: Binary Branches of T_{12} and T_{21}

of computing the string edit distance. When applying well-known techniques (e.g., sort merge and hash join), all the lower bounds between each tree pair in $F_1 \times F_2$ can be computed without nested-loop. This makes the filtering process using set-based method much more efficient than that based on strings. Here we take the label histogram distance as an example to discuss the efficiency of set-based join methods.

Suppose that F_1 and F_2 are two sets of XML fragments. The goal of filtering process is to find all the tree pairs in $F_1 \times F_2$ with lower bound within the threshold τ. Algorithm 1 describes the filtering process. All the trees are firstly transformed into node-sets (multi-sets). Then we merge all the node-sets transformed from trees in F_i into $List_i$ (line 3). Note that the two $List$s are lists sorted by the $label - value$ of each node (to be brief, all alphanumeric labels are converted to number labels method [13]). The size of each node-set in F_i is computed and stored in the $List$s (line 4). We check for each node label in which pairs of trees it appears and count the number of node labels that each tree pair shares (line 5-6). That number equals to the size of the intersection of a pair of node-sets. The

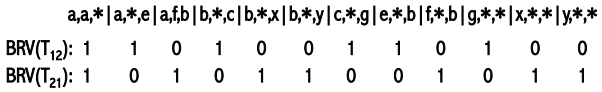


Figure 7: Binary Branch Vectors of T_{12} and T_{21}

sum of the size of two trees minus twice the size of their intersection equals to the size of symmetric difference(line 7). Then all the lower bounds of edit distance between in $F_1 \times F_2$ are computed without nested-loop (line 7). The tree pairs with lower bound lower than τ are returned (line 7).

Algorithm 1 Filtering Algorithm for Set or Vector Based Methods

Input: F_1, F_2, τ
Output: $CandidateTreePairs$

1: **for all** trees in F_i **do**
2: **for** all the node labels in this tree **do**
3: $List_i = List_i \uplus (tID_i, \text{label-value}, count_i)$
4: **end for**
5: **end for**
6: $\Gamma_{tID_i, SUM(count_i) \rightarrow size_i}(List_i)$
7: $List' = List_1 \bowtie List_2$
8: $List'' = \Gamma_{tId_1, tId_2, sum(min(count_1, count_2)) \rightarrow \cap}(List')$
9: $candidate \leftarrow \pi_{tId_1, tId_2}(\sigma_{\frac{size_1 + size_2 - 2*\cap}{2} \leq \tau}(List''))$
10: **return** $candidate$

To be brief, it is supposed that the two XML sets have N trees for each and all the trees have n nodes. To analyze the time complexity, we summary the filtering algorithm algorithm to two steps:

1. All the trees are transformed to their corresponding node sets.

2. Sort-merge and hash join is applied to the sets and the tree pairs with lower bound distance lower than τ are returned.

In the first step, since each tree can be transformed to its corresponding label set in time $O(n)$, the running time in the first step is $O(Nn)$. In the second step, the diversity of the trees would affect the running time. In the best case, when no tree pair shares any element, the run time in this step is the time of merging all sets into $List_1$ and $List_2$. That is $O(Nnlog(Nn))$. In the worst case, when all the transformed sets are exactly the same. Each element in one List would match N tuples in the other List. Thus the run time is $O(Nnlog(Nn) + N^2n)$. From our experiments on various real-world data sets, the running time in this step is usually close to the best case. Therefore, the average time complex-

Methods	Worst Case	Average
String-based	N^2n^2	N^2n^2
Leaf Distance Histogram	$Nh*log(Nh) + N^2h$	$Nh*log(Nh)$
Degree Histogram	$Nd*log(Nd) + N^2d$	$Nd*log(Nd)$
Label Histogram	$Nn*log(Nn) + N^2n$	$Nn*log(Nn)$
Binary Branch	$Nn*log(Nn) + N^2n$	$Nn*log(Nn)$

Figure 8: Time Complexity of each Method (N denotes the number of trees in each data source, n denotes the number of nodes in each tree, h denotes the height of a tree, and d denotes the highest fanout of a tree).

ity of set-based filtering algorithm can be estimated as $O(Nnlog(Nn))$.

The time complexity of each method is summarized in Figure 8.

5.2 Tightness of each Lower Bound

Since different lower bound functions are suitable in different cases, it is hard to analyze the overall tightness of each lower bound theoretically. Intuitively, leaf distance histogram and degree histogram give very rough lower bounds, while label histogram and binary branch provide much tighter lower bounds. Except in some extreme examples, the string-based lower bounds are much tighter than set-based lower bound functions. Here, we analyze the tightness of each lower bound function using extreme examples.

5.2.1 Leaf Distance Histogram and Degree Histogram

Leaf distance histogram (degree histogram) can only detect the differences in leaf distance (degree) information between trees but entirely disregard the label information. As long as the leaf distance (degree) information between trees are similar, leaf distance histogram (degree histogram) cannot detect the distance. Here we illustrate this point in two examples.

EXAMPLE 8. *In Figure 9, T_1 and T_2 are very different trees, but their leaf distance histograms are exactly the same. Both of them have 5 nodes (v_4, v_5, v_6, v_7, v_8 in T_1 and w_4, w_5, w_6, w_7, w_8 in T_2) at leaf height 0, 2 nodes (v_2, v_3 in T_1 and w_2, w_3 in T_2) at leaf height 1, and one node (v_1 in T_1 and w_1 in T_2) at leaf height 2. Leaf distance histograms fail to detect the differences between these two trees, thus cannot provide tight lower bounds in this case. In Figure 10, T_3 and T_4 are also very different trees. Using degree histogram, their label and structural differences cannot be detected at all, since the two trees have exactly the same degree histograms: one node (v_1 in T_3 and w_3 in T_4) has 3 children, one node (v_2 in T_3 and w_1 in T_4) has 2 children, and other five nodes do not have children.*

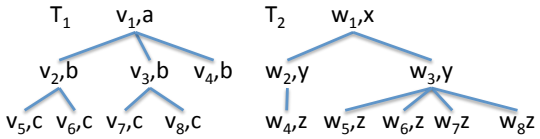


Figure 9: Mismatch using Leaf Distance Histogram

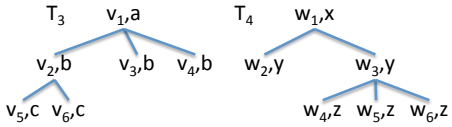


Figure 10: Mismatch using Degree Histogram

5.2.2 Label Histogram

Since the label histogram only considers the label information of trees, it cannot work well when most changes are structural changes.

EXAMPLE 9. *In Figure 11, T_5 and T_6 are very different trees. But their histograms are exactly the same since they share the same label set. So in this case, label histograms fail to provide a tight lower bound.*

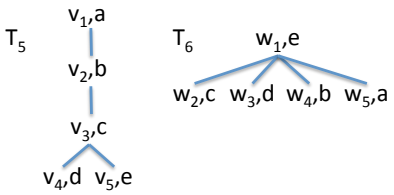


Figure 11: Mismatch using Label Histogram

5.2.3 Binary Branch

Although both label and structure information are considered, the lower bounds provided by Binary Branch is also rough.

THEOREM 5. *Let T_1 and T_2 be two trees with n and m nodes, respectively. The lower bound distance between T_1 and T_2 provided by the binary branch is at most $0.2(n+m)$.*

PROOF. The number of binary branch of a tree equals to the tree size. The binary branch distance between them is at most $n+m$ (only in the case

that the two trees share no binary branch). Thus the lower bound distance provide by binary branch is at most $0.2(n+m)$. □

In many cases, especially when the predefined threshold is above $0.2(m+n)$, the method binary branch cannot filter out any tree pairs. Now we analyze the provided lower bounds in different cases. A small change in a node would affect all its binary branches. Consider the ratio between binary branch distance and number of changed nodes. This ratio would be higher in the case of many small changes than a big change (a subtree move or deletion). Thus the lower bounds provided by binary branch would be relatively tighter in the cases when a lot of small differences exist between trees.

5.2.4 String-based Lower Bound

String is a data structure which contains order for structure as well as content information in each entry. Although computationally more complex than sets, in most cases, string-based lower bounds are tight. Here we illustrate this point intuitively. Since the discussed trees are ordered trees, the child order information is important to identify similar trees. In both preorder and postorder traversal, the child order information is fully contained. Also, the label of each tree node appears exactly once, which describes the label information properly. The hierarchical information of trees is the most difficult information to describe. To describe the hierarchical information, string-based lower bounds use two kinds of traversals: preorder traversal and postorder traversal, in which each node is visited before (after) all its children are visited. Since the maximum value of the two string edit distance is chosen as the lower bound, the lower bound is not accurate only when neither of the two traversals can properly describe the hierarchical information.

Also, string-based lower bound is always no worse than that provided by label histogram. That is because label differences cost the same in both string-based method and label histogram while string-based lower bound also detects some structure differences.

6. COMBINING FILTERING METHODS

Since lower bounds are definitely lower than the exact tree edit distance, the lower bound which has the maximum value is the one closest to the exact value. This inspires us to use different methods to compute different lower bounds and use the maximum lower bound as the final lower bound.

Definition 10. (Combined Distance Function). *Let $D = d_i$ (i from 1 to n) be a set of lower bound distance functions. The combined distance function d_c is defined as the maximum of the component functions:*

$$d_C(T_1, T_2) = max\{d_i(T_1, T_2) | 1 < i < n\}.$$

Also, a tree pair is definitely dissimilar if any of its lower bounds is higher than the threshold. In the case of approximate join two tree sets, we can first use some very efficient yet rough lower bound functions to wipe out a large part of dissimilar tree pairs and then use a slower yet more accurate filtering method to further filter the remained tree pairs.

Definition 11. (Multi-level filtering). *Let F_1 and F_2 be two tree sets, τ be the threshold, $D = d_i$ (i from 1 to n) be a list of lower bound distance function (or combined distance function), C_i be tree pairs remained after the ith filtering. The result of multi-level filtering method using D is C_n where C_0=$F_1 \times F_2$, $C_i = \{t | t \in C_{i-1}, d_i(t) \leq \tau\}$.*

The filtering effect of using multi-level filtering method is equal to using all the methods one by one, while saving much of the overall running time. Since set-based lower bounds functions are computed more efficiently. All set-based functions are combined to form a set-based combined distance function, which serves as the first round of filter. Two string-based functions $ed(pre(T_1), pre(T_2))$ and $ed(post(T_1), post(T_2))$ are used as the second and third distance function, respectively. The overall efficiency is enhanced, since most dissimiar tree pairs are wiped out in the first round.

7. EXPERIMENTS

In this section, we test the efficiency and effectiveness of all the reviewed lower bound functions and the combined lower bound functions. All of our experiments were performed on a PC with Intel Core Duo 2GHz, 1GB main memory and 250GB hard disk. The OS is Windows XP Professional. We implemented our experiments using CodeBlocks.

We use four real-world data sets ranging from apartment data (street), bioinformatics (Swissprot), linguistics (Treebank), and bibliography (DBLP).

- **Street:** We use the application data from the Municipality of Bozen. The scene is that the Office wants to integrate the apartment data stored in two databases and display that information on a map. The data is hierarchically organized, in which the root of the tree is the street name, the children of the street name are the house numbers, the children of house

numbers are the entrance numbers, and the children of entrance numbers are the apartment numbers. We choose subsets from them which has 100 trees for each. The tree size is from 50 to 200. We denote the two sources as R and L.

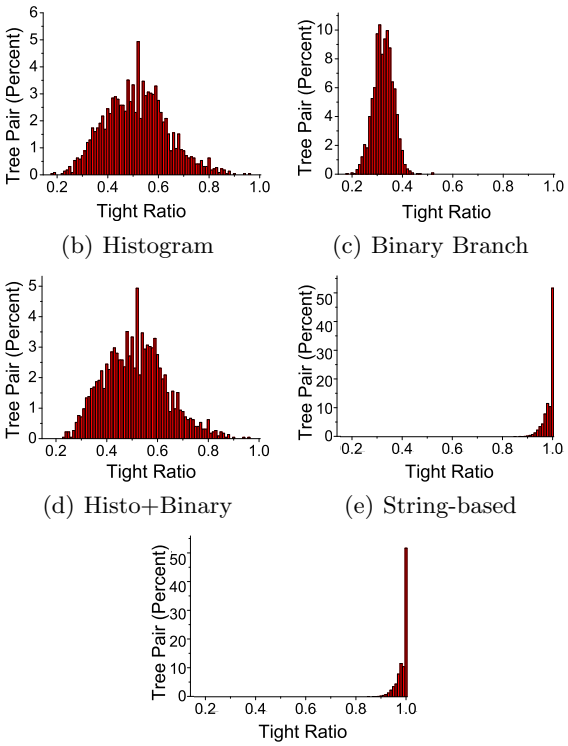
- **SwissProt:** SwissProt is a database which describe protein sequence. Each SwissProt document contains trees with about 100 nodes and about 4 depth on average. The documents in SwissProt show high degree of similarity for they share large numbers of labels.
- **TreeBank:** TreeBank is a database storing parts of speech tagged English sentences. Its documents have deep recursive structure (about 50 nodes and about 7 depth on average).
- **DBLP:** DBLP is a bibliography database that consists of large numbers of small and flat documents (about 15 nodes and 2 depth on average).

7.1 Tightness of each Lower Bound

In this section, we test the tightness of each lower bound function. For two trees T_1 and T_2, $TD(T_1, T_2)$ is the exact tree edit distance, while $d_i(T_1, T_2)$ is the lower bound provided by ith method. We use *tight ratio* ($tr = \frac{d_i(T_1,T_2)}{TD(T_1,T_2)}$) to evaluate the tightness for each lower bound. The closer the tight ratio is to 1, the tighter the lower bound is. The average tight ration for each database is shown in Figure 12(a). The detailed tight ratios for the street database are shown in Figure 12(b) to Figure 12(f). Binary branch serves as the roughest lower bound distance while histogram often gives much tighter lower bounds. String-based lower bounds always provide the most accurate lower bounds. The combination of histogram and binary branch performs slightly better than that of histogram, while the combination of all the lower bounds slightly outperforms string-based lower bounds.

	Histogram	Binary	Histo+Binary	String	All Methods
Street	0.502	0.310	0.503	0.983	0.983
SwissProt	0.795	0.344	0.795	0.998	0.998
TreeBank	0.748	0.325	0.748	0.996	0.996
DBLP	0.894	0.414	0.894	1.000	1.000

(a) Average Tightness in each Database



(b) Histogram (c) Binary Branch (d) Histo+Binary (e) String-based (f) All the Methods

Figure 12: Tightness of each Lower Bound Function

7.2 Filter Quality

In this section, we test the filtering quality provided by each method. For two tree sets F_1 and F_2 and a threshold τ, the goal of the filtering process is to wipe out as many dissimilar tree pairs as possible. In this section, we test the size of remaining tree sets provided by each method. We also compute the size of exact result by computing the exact edit distance for the smallest remaining set. The closer the size of remaining set to the result size, the higher filter quality is. We compute the size of remaining set for different thresholds from $0.05(m+n)$ to $0.5(m+n)$ in all the databases, where m and n are the size of the two current trees. The average size of remaining set is shown in Figure 13. The sizes of remaining set under different thresholds are shown in Figure 14. The result is that binary branch has the lowest filter quality while histogram works much better. String-based lower bounds always give the highest filter quality and nearly filter out all the dissimilar tree pairs. The combination of histogram and binary branch performs slightly better than histogram while the combination of all the lower bounds works slightly better string-base lower bounds.

7.3 Computing Efficiency

As we analyzed in Section 5, the string-based lower bounds are relatively costly, while set-based lower bounds can be computed efficiently. Also, the multi-level filter, which is at least as effective as

	Histogram	Binary	Histo+Binary	String	All Methods	Result
Street	1441	1808	1440	543	543	526
SwissProt	220	649	220	108	108	107
TreeBank	288	658	288	105	105	105
DBLP	223	681	223	136	136	136

Figure 13: Average Filter Quality in each Database.

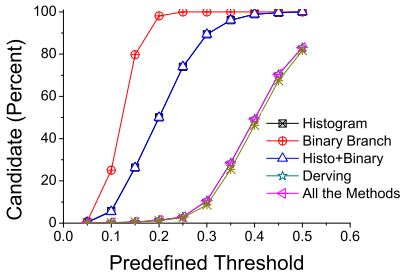


Figure 14: Detail Filter Quality in Street Database.

string-based methods since the latter is included by the former, benefits high efficiency. In this section, we test the efficiency of each individual method and the multi-level filter.

We use the Swissprot, Treebank, DBLP and street databases to test the efficiency. We set the threshold $\tau = 0.1(m+n)$ (m and n are the size of the two current trees) and increase the number of trees in each database to test the filter time. In the multi-level filter, all set-based lower bounds are used in the first round while the $ed(pre(T_1), pre(T_2))$ and $ed(post(T_1), post(T_2))$ serve as the second and third filters. The results are shown in figure 15(a) - figure 15(d). Histogram and binary branch have much higher efficiency than string-based lower bounds. The multi-level filter also outperforms string-based lower bounds significantly in efficiency.

8. CONCLUSION

In this paper, we have compared and analyzed the performance of string-based lower bounds, histogram, and binary branch for giving the lower bound to tree edit distance. String-based lower bounds is the tightest and thus have the highest filter quality. Although relatively rough, the lower bounds provided by histogram and binary branch can be computed very efficiently. We also combine these methods to form multi-level filters to get tight lower bound efficiently. Experiment results confirm the analytical results.

9. ACKNOWLEDGEMENTS

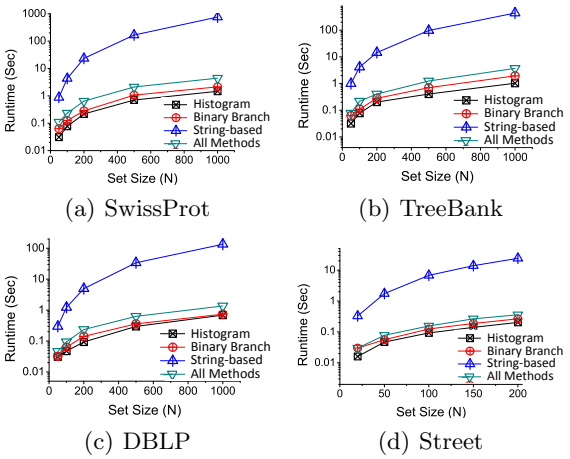


Figure 15: Filter Efficiency

Many thanks to Michael H. Böhlen [4, 3, 5] for his source code and test data. He has been a great help in this research. This paper was partially supported by NGFR 973 grant 2012CB316200, NSFC grant 61003046, 61111130189, 61133002 and NGFR 863 grant 2012AA011004. Doctoral Fund of Ministry of Education of China (No. 20102302120054). Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), Ministry of Education (No.KF2011003). the Fundamental Research Funds for the Central Universities(No. HIT. NSRIF. 2013064).

10. REFERENCES

[1] Tatsuya Akutsu. A relation between edit distance for ordered trees and edit distance for euler strings. *Inf. Process. Lett.*, 100(3):105–109, 2006.

[2] Tatsuya Akutsu, Daiji Fukagawa, and Atsuhiro Takasu. Approximating tree edit distance through string edit distance. In *ISAAC*, pages 90–99, 2006.

[3] Nikolaus Augsten, Michael H. Böhlen, Curtis E. Dyreson, and Johann Gamper. Approximate joins for data-centric XML. In *ICDE*, pages 814–823, 2008.

[4] Nikolaus Augsten, Michael H. Böhlen, and Johann Gamper. Approximate matching of hierarchical data using *pq*-grams. In *VLDB*, pages 301–312, 2005.

[5] Nikolaus Augsten, Michael H. Böhlen, and Johann Gamper. The *pq*-gram distance between ordered labeled trees. *ACM Trans. Database Syst.*, 35(1): 1–36, 2010.

[6] Philip Bille. A survey on tree edit distance and related problems. *Theor. Comput. Sci.*,

337(1-3):217–239, 2005.

[7] Weimin Chen. New algorithm for ordered tree-to-tree correction problem. *J. Algorithms*, 40(2):135–158, 2001.

[8] Erik D. Demaine, Shay Mozes, Benjamin Rossman, and Oren Weimann. An optimal decomposition algorithm for tree edit distance. In *ICALP*, pages 146–157, 2007.

[9] Minos N. Garofalakis and Amit Kumar. XML stream processing using tree-edit distance embeddings. *ACM Trans. Database Syst.*, 30(1):279–332, 2005.

[10] Sudipto Guha, H. V. Jagadish, Nick Koudas, Divesh Srivastava, and Ting Yu. Approximate XML joins. In *SIGMOD Conference*, pages 287–298, 2002.

[11] Fei Li, Hongzhi Wang, Cheng Zhang, Liang Hao, Jianzhong Li, and Hong Gao. Approximate joins for XML using g-string. In *XSym*, pages 3–17, 2010.

[12] Karin Kailing, Hans-Peter Kriegel, Stefan Schönauer, and Thomas Seidl. Efficient similarity search for hierarchical data in large databases. In *EDBT*, pages 676–693, 2004.

[13] Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, 1987.

[14] Philip N. Klein. Computing the edit-distance between unrooted ordered trees. In *ESA*, pages 91–102, 1998.

[15] Tetsuji Kuboyama. *Matching and Learning in Trees*. Doctoral Dissertation. The University of Tokyo, 2007.

[16] Bruce A. Shapiro and Kaizhong Zhang. Comparing multiple RNA secondary structures using tree comparisons. *Computer Applications in the Biosciences*, 6(4):309–318, 1990.

[17] Kuo-Chung Tai. The tree-to-tree correction problem. *J. ACM*, 26(3):422–433, 1979.

[18] Shirish Tatikonda and Srinivasan Parthasarathy. Hashing Tree-Structured Data: Methods and Applications. In *ICDE*, pages 429-440, 2010.

[19] Gabriel Valiente. An efficient bottom-up distance between trees. In *SPIRE*, pages 212–219, 2001.

[20] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[21] Rui Yang, Panos Kalnis, and Anthony K. H. Tung. Similarity evaluation on tree-structured data. In *SIGMOD Conference*, pages 754–765, 2005.

[22] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, 1989.

Data Profiling Revisited

Felix Naumann*
Qatar Computing Research Institute (QCRI), Doha, Qatar
fnaumann@qf.org.qa

ABSTRACT

Data profiling comprises a broad range of methods to efficiently analyze a given data set. In a typical scenario, which mirrors the capabilities of commercial data profiling tools, tables of a relational database are scanned to derive metadata, such as data types and value patterns, completeness and uniqueness of columns, keys and foreign keys, and occasionally functional dependencies and association rules. Individual research projects have proposed several additional profiling tasks, such as the discovery of inclusion dependencies or conditional functional dependencies.

Data profiling deserves a fresh look for two reasons: First, the area itself is neither established nor defined in any principled way, despite significant research activity on individual parts in the past. Second, more and more data beyond the traditional relational databases are being created and beg to be profiled. The article proposes new research directions and challenges, including interactive and incremental profiling and profiling heterogeneous and non-relational data.

1. DATA PROFILING

"*Data profiling is the process of examining the data available in an existing data source [...] and collecting statistics and information about that data.*"[1] Profiling data is an important and frequent activity of any IT professional and researcher. We can safely assume that any reader of this article has engaged in the activity of data profiling, at least by eye-balling spreadsheets, database tables, XML files, etc. Possibly more advanced techniques were used, such as key-word-searching in data sets, sorting, writing structured queries, or even using dedicated data profiling tools. While the importance of data profiling is undoubtedly high, and while efficiently and effectively profiling is an enormously difficult challenge, it has yet to be established as a research area in its own right. We focus our discussion on relational data, the predominant format of traditional data profiling methods, but we do regard data profiling for other data models in a separate section.

Data profiling encompasses a vast array of methods to examine data sets and produce metadata. Among the simpler results are statistics, such as the number of null values and distinct values in a column, its data type, or the most frequent patterns of its values. Metadata that are more difficult to compute usually involve multiple columns, such as inclusion dependencies or functional dependencies. More advanced techniques detect approximate properties or conditional properties of the data set at hand. To allow focus, the broad field of data mining is deliberately omitted from the discussion here, as justified below. Obviously, all such discovered metadata refer only to the given data instance and cannot be used to derive with certainty schematic/semantic properties, such as primary keys or foreign key relationships. Figure 1 shows a classification of data profiling tasks. The tasks for "single sources" correspond to state-of-the-art in tooling and research (see Section 2), while the tasks for "multiple sources" reflect new research directions for data profiling (see Section 5).

Systematic data profiling, i.e., profiling beyond the occasional exploratory SQL query or spreadsheet browsing, is usually performed by dedicated tools or components, such as IBM's Information Analyzer, Microsoft's SQL Server Integration Services (SSIS), or Informatica's Data Explorer. Their approaches all follow the same general procedure: A user specifies the data to be profiled and selects the types of metadata to be generated. Next, the tool computes in batch the metadata using SQL queries and/or specialized algorithms. Depending on the volume of the data and the selected profiling results, this step can last minutes to hours. The results are usually displayed in a vast collec-

*On leave from Hasso Plattner Institute, Potsdam, Germany (naumann@hpi.uni-potsdam.de).
[1]Wikipedia on "Data Profiling", 2/2013

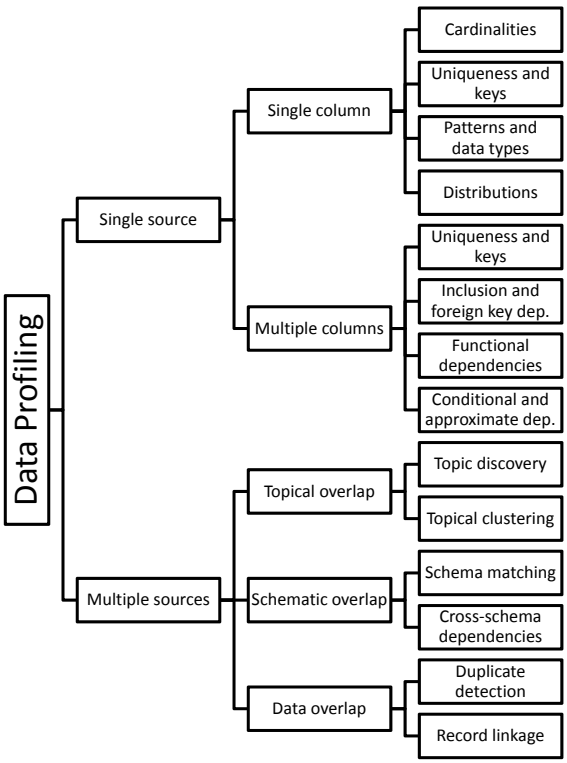


Figure 1: A classification of data profiling tasks

tion of tabs, tables, charts, and other visualizations to be explored by the user. Typically, discoveries can then be translated into constraints or rules that are then enforced in a subsequent cleansing/integration phase. For instance, after discovering that the most frequent pattern for phone numbers is `(ddd)ddd-dddd`, this pattern can be promoted to the *rule* that all phone numbers must be formatted accordingly. Most cleansing tools can then either transform differently formatted numbers or at least mark them as violations.

Use cases for profiling. The need to profile a new or unfamiliar set of data arises in many situations, in general to prepare for some subsequent task.

Query optimization. Basic profiling is performed by most database management systems to support query optimization with statistics about tables and columns. These profiling results can be used to estimate the selectivity of operators and ultimately the cost of a query plan.

Data cleansing. Probably the most typical use case is profiling data to prepare a data cleansing process. Profiling reveals data errors, such as inconsistent formatting within a column, missing values, or outliers. Profiling results can also be used to measure and monitor the general quality of a data set, for instance by determining the number of records that do not conform to previously established constraints.

Data integration. Often the data sets to be integrated are somewhat unfamiliar and the integration expert wants to explore the data sets first: How large is it? What data types are needed? What are the semantics of columns and tables? Are there dependencies between tables and among databases, etc.? The vast abundance of (linked) *open data* and the desire and potential to integrate them with enterprise data has amplified this need.

Scientific data management. The management of data that is gathered during scientific experiments or observations has created additional motivation for efficient and effective data profiling: When importing raw data, e.g., from scientific experiments or extracted from the Web, into a DBMS, it is often necessary and useful to profile the data and then devise an adequate schema.

Data analytics. Almost any statistical analysis or data mining run is preceded by a profiling step to help the analyst understand the data at hand and appropriately configure tools, such as SPSS or Weka. Pyle describes detailed steps of analyzing and subsequently preparing data for data mining [38].

Knowledge about data types, keys, foreign keys, and other constraints supports data modeling and helps keep data consistent, improves query optimization, and reaps all the other benefits of structured data management. Other research efforts have mentioned query formulation and indexing [42], scientific discovery [26], and database reverse engineering [35] as further motivation for data profiling.

Time to revisit. Recent trends in the database field have added challenges but also opportunities for data profiling. First, under the *big data* umbrella, industry and research have turned their attention to data that they do not own or have not made use of yet. Data profiling can help assess which data might be useful and reveals the yet unknown characteristics of such new data: before exposing an infrastructure to Twitter's firehose it might be worthwhile to know about properties of the data one is receiving; before downloading significant parts of the linked data cloud, some prior

sense of the integration effort is needed; before augmenting a warehouse with text mining results an understanding of their quality is required. Leading researchers have recently noted "*If we just have a bunch of data sets in a repository, it is unlikely anyone will ever be able to find, let alone reuse, any of this data. With adequate metadata, there is some hope, but even so, challenges will remain* [...]" [4].

Second, much of the data that shall be exploited is of non-traditional type for data profiling, i.e., non-relational (e.g., linked open data), non-structured (e.g., tweets and blogs), and heterogeneous (e.g., open government data). And it is often truly "big", both in terms of schema, rendering algorithms that are exponential in the number of schema elements infeasible, and in terms of data, rendering main-memory based methods infeasible. Existing profiling methods are not adequate to handle that kind of data: Either they do not scale well (e.g., dependency discovery), or there simply are no methods yet (e.g., incremental profiling, profiling multiple data sets, profiling textual attributes).

Third, different and new data management architectures and frameworks have emerged, including distributed systems, key-value stores, multi-core- or main-memory-based servers, column-oriented layouts, streaming input, etc. These new premises provide interesting opportunities as we discuss later.

Profiling challenges. Data profiling, even in a traditional relational setting, is non-trivial for three reasons: First, the results of data profiling are *computationally complex* to discover. For instance, discovering key candidates or dependencies usually involves some sorting step for each considered column. Second, the *discovery-aspect* of the profiling task demands the verification of complex constraints on all columns and combinations of columns in a database. And thus also the solution-space of uniqueness-, inclusion dependency-, or functional dependency-discovery is exponential in the number of attributes. Third, profiling is often performed on data sets that may not fit into main memory.

Various tools and algorithms have tackled these challenges in different ways. First, many rely on the capabilities of an underlying DBMS, as many profiling tasks can be expressed as SQL queries. Second, many have developed innovative ways to handle the individual challenges, for instance using indexing schemes, parallel processing, and reusing intermediate results. Third, several methods have been proposed that deliver only approximate results for various profiling tasks, for instance by profiling samples. Finally, users are asked to narrow down the discovery process to certain columns or tables. For instance, there are tools that verify inclusion dependencies on user-suggested pairs of columns, but that cannot automatically check inclusion between all pairs of columns or column sets.

The following section elaborates these traditional data profiling tasks and gives a brief overview of known approaches. Sections 3 – 6 are the main contributions of this article by defining and motivating new research perspectives for data profiling. These areas include interactive profiling (users can act upon profiling results and re-profile efficiently), incremental profiling (profiling results are incrementally updated as new data arrives), profiling heterogeneous data and multiple sources simultaneously, profiling non-relational data (XML and RDF), and profiling on different architectures (column stores, key-value stores, etc.).

This article is not intended to be a survey of existing approaches, though there is certainly a need for such, nor is it a formal framework for future data profiling developments. Rather, it strives to spark interest in this research area and to assemble a wide range of research challenges.

2. STATE OF THE ART

While the introduction mentions current industrial profiling tools, this section discusses current research directions. In its basic form, data profiling is about analyzing data values of a single column, summarized as "traditional data profiling". More advanced techniques detect relationships among columns of one or more tables, which we discuss as "dependency detection". Finally, we distinguish data profiling from the broad field of "data mining", which we deliberately exclude from further discussion.

Traditional data profiling. The most basic form of data profiling is the analysis of individual columns in a given table. Typically, generated metadata comprises various counts, such as the number of values, the number of unique values, and the number of non-null values. These metadata are often part of the basic statistics gathered by DBMS. Mannino et al. give a much-cited survey on statistics collection and its relationship to database optimization [32]. In addition to the basic counts, the maximum and minimum values are discovered and the data type is derived (usually restricted to string vs. numeric vs. date). Slightly more advanced techniques create histograms of value distributions, for instance to optimize range-queries [37], and identify typical patterns in the data values in the form

of regular expressions [40]. Data profiling tools display such results and can suggest some actions, such as declaring a column with only unique values a key-candidate or suggesting to enforce the most frequent patterns.

Dependency detection. Dependencies are metadata that describe relationships among columns. The difficulties are twofold: First, *pairs* of columns or column-sets must be regarded, and second, the chance existence of a dependency in the data at hand does not imply that this dependency is *meaningful*.

The most frequent real-world use-case is the discovery of foreign keys [30, 41] with the help of inclusion dependencies [6, 33]. Current data profiling tools often avoid checking all combinations of columns, but rather ask the user to suggest a candidate key/foreign-key pair to verify. Another form of dependency, which is also relevant for data quality, is the functional dependency (FD). Again, much research has been performed to automatically detect FDs [26, 45].

Both types of dependencies can be relaxed in two ways. First, *conditional dependencies* need to hold only for tuples that fulfill the condition. Conditional inclusion dependencies (CINDs) were proposed for data cleaning and contextual schema matching [11]. Different aspects of CIND discovery have been addressed in [5, 17, 22, 34]. Conditional functional dependencies (CFDs) were introduced in [20] for data cleaning. Algorithms for discovering CFDs are also proposed in [14, 21]. Second, *approximate dependencies* need to hold only for a certain percentage of the data – they are not guaranteed to hold for the entire relation. Such dependencies are often discovered using sampling [27] or other summarization techniques [16].

Finally, algorithms for the discovery of columns and column combinations with only unique values (which is strictly speaking a constraint and not a dependency) have been proposed in [2, 42].

To reiterate our motivation: There are various individual techniques for various individual profiling tasks. What is lacking even for the state-of-the-art is a unified view of data profiling as a field and a unifying framework of its tasks.

Data mining. Rahm and Do distinguish data profiling from data mining by the number of columns that are examined: "Data profiling focusses on the instance analysis of individual attributes. [...] Data mining helps discover specific data patterns in large data sets, e.g., relationships holding between several attributes" [39]. Yet, a different distinction is more useful to separate the different use cases: Data profiling gathers technical metadata to support data management, while data mining and data analytics discovers non-obvious results to support business management. In this way, data profiling results are information about columns and column sets, while data mining results are information about rows or row sets (clustering, summarization, association rules, etc.).

Of course such a distinction is not strict. Some data mining technology does express information about columns, such as feature selection methods for sets of values within a column [7] or regression techniques to characterize columns [13]. Yet with the distinction above, we concentrate on data profiling and put aside the broad area of data mining, which has already received unifying treatment in numerous text books and surveys.

3. INTERACTIVE DATA PROFILING

Data profiling research has yet hardly recognized that data profiling is an inherently user-oriented task. In most cases, the produced metadata is consumed directly by the user or it is at least regarded by a user before put to use in some application, such as schema design or data cleansing. We suggest the involvement of the user already during the algorithmic part of data profiling, hence "interactive profiling".

Online profiling. Despite many optimization efforts, data profiling might last longer than a user is willing to wait in front of a screen with nothing to look at. Online profiling shows intermediate results as they are created. However, simply hooking the graphical interface into existing algorithms is usually not sufficient: Data that is sorted by some attribute or has a skewed order yields misleading intermediate results. Solutions might be approximate or sampling-based methods, whose results gracefully improve as more computation is invested. Naturally, such intermediate results do not reflect the properties of the entire data set. Thus, some form of confidence, along with a progress indicator, can be shown to allow an early interpretation of the results.

Apart from entertaining users during computation, an advantage of online profiling is that the user may abort the profiling run altogether. For instance, a user might decide early on that the data set is not interesting (or clean) enough for the task at hand.

Profiling on queries and views. In many cases, data profiling is performed with the purpose of cleaning the data or the schema to some extent, for instance, to be able to insert it into a data warehouse or to integrate it with some other data set. However, each cleansing step changes the data, and thus implicitly also the metadata produced by profiling. In general, after each cleansing step a new profiling run should be performed. For instance, only after cleaning up zip codes does the functional dependence with the city values become apparent. Or only after deduplication does the uniqueness of email addresses reveal itself.

A modern profiling system should be able to allow users to virtually interact with the data and re-compute profiling results. For instance, the profiling system might show a 96% uniqueness for a certain column. The user might recognize that indeed the attribute should be completely unique and is in fact a key. Without performing the actual cleansing, a user might want to virtually declare the column to be a key and re-perform profiling on this virtually cleansed data. Only then a foreign key for this attribute might be recognized.

In short, a user might want to act upon profiling results in an ad-hoc fashion without going through the entire cleansing and profiling loop, but remain within the profiling tool context and perform cleansing and re-profiling only on a virtually cleansed view. When satisfied, the virtual cleansing can of course be materialized. A key enabling technology for this kind of interaction is the ability to efficiently re-perform profiling on slightly changed data, as discussed in the next section. In the same manner, profiling results can be efficiently achieved on query results: While calculating the query result, profiling results can be generated on the side, thus showing a user not only the result itself, but also the nature of that data. Faceted search provides similar features in that a user is presented with cardinalities based on the chosen filters.

For all suggestions above, new algorithms and data structures are needed to enhance the user experience of data profiling.

4. INCREMENTAL DATA PROFILING

A data set is hardly ever fixed: Transactional data is appended to frequently, analytics-oriented data sets experience periodic updates (typically daily), and large data sets available on the web data are updated every few weeks or months. Data profiling methods should be able to efficiently handle such moving targets, in particular without re-profiling the entire data set.

Incremental profiling. An obvious, but yet under-examined extension to data profiling is to reuse earlier profiling results to speed-up computation on changed data. I.e., the profiling system is provided with a data set and with knowledge of its delta compared to a previous version, and it has stored any intermediate or final profiling results on that previous version. In the simplest cases, profiling metadata can be calculated associatively (e.g., sum, count, equi-width histograms), in some cases some intermediate metadata can help (e.g., sum and count for average, indexes for value patterns), and finally in some cases a complete recalculation might be necessary (e.g., median or clustering).

There is already some research on performing individual profiling tasks incrementally. For instance, the AD-Miner algorithm allows an incremental update of functional dependency information [19]. Fan et al. focus on the area of conditional functional dependencies and also consider incremental updates [20]. The area of data mining, on the other hand, has seen much related work, for instance on association rule mining and other data mining applications [24].

Continuous profiling. While for incremental profiling we assumed periodic updates (or periodic profiling runs), a further use case is to update profiling results while (transactional) data is created or updated. If the profiling results can be expressed as a query, and if they shall be performed only on a temporal window of the data, this use case can be served by data stream management systems [23]. If this is not the case, continuous profiling methods need to be developed, whose results can be displayed in a dashboard. Of particular importance is to find a good tradeoff between recency, accuracy, and resource consumption. Use cases for continuous profiling include internet traffic monitoring or the profiling of incoming search queries.

Multi-measure profiling. Each profiling algorithm has its own scheme of running through the data and collecting or aggregating whatever information is needed. Realizing that multiple types of profiling metadata shall be collected, it is likely that many of these runs can be combined. Thus, in a manner similar to multi-query-optimization, there is a high potential for efficiency gains, in particular wrt. I/O cost. While such potential is already realized in commercial systems, it has not yet been investigated for the more complex tasks that are not covered by these tools.

5. PROFILING HETEROGENEOUS DATA

While typical profiling tasks assume a single, largely homogeneous database or even only a single table, there are many use cases in which a combined profiling of multiple, heterogeneous data sets is needed. In particular when integrating data it is useful to learn about the *common properties* of participating data sets. From profiling one can learn about their integrability, i.e., how well their data and schemata fit together, and learn in advance the properties of the integrated data set. Even profiling a single source that stores data for multiple or many domains, such as DBpedia or Freebase, can profit from techniques that profile heterogeneous data.

Degrees of heterogeneity. Heterogeneity in data sets can appear at many different levels and in many different degrees of severity. Data profiling methods can be used to uncover these heterogeneities and possibly provide hints on how to overcome them.

Heterogeneity is traditionally divided into syntactic heterogeneity, structural heterogeneity, and semantic heterogeneity [36]. Discovering syntactic heterogeneity, in the context of data profiling, is precisely what traditional profiling aims at, e.g., finding inconsistent formatting. Next, structural heterogeneity appears in the form of unmatched schemata and differently structured information. Such problems are only partly addressed by traditional profiling, e.g., by discovery schema information, such as types, keys, or foreign keys. Finally, semantic heterogeneity addresses the underlying and possibly mismatched meaning of the data. For data profiling we interpret it as the discovery of semantical overlap of the data and their domain(s).

Data profiling for integration. Our focus here is on profiling tasks to discover structural and semantic heterogeneity, arguing that structural profiling seeks information about the schema and semantic profiling seeks information about the data. Both serve to assess the *integrability* of data sets, and thus also indicate the necessary integration effort, which is vital to project planning. The integration effort might be expressed in terms of similarity, but also in terms of man-months or in terms of which tools are needed.

An important issue in integrated information systems, irrelevant for single databases, is the *schematic similarity*, i.e., the degree to which their schemata complement each other and the degree to which they overlap. There is an obvious relation to schema matching techniques, which aim at automatically finding correspondences between schema elements [18]. Already Smith et al. have recognized that schema matching techniques often play the role of profiling tools [43]: Rather than using them to derive schema mappings and perform data transformation, they play roles that have a more informative character, such as assessment of project feasibility or the identification of integration targets. However, the mere matching of schema elements might not suffice as a profiling-for-integration result: Additional information on the structure of the values of the matching columns can provide further details about the integration difficulty.

After determining schematic overlap, a next step is to determine *data overlap*, i.e., the (estimated) number of real-world objects that are represented in both data sets, or that are represented multiple times in a single data set. Such multiple representations are typically identified using entity matching methods (aka. record linkage, entity resolution, duplicate detection, and many other names) [15]. However, estimating the number of matches without actually performing the matching on the entire data set is an open problem. If used to determine the integration effort, it is additionally important to know how diverse such matching records are represented, i.e., how difficult it is to devise good similarity measures and find appropriate thresholds.

Topical profiling. When profiling yet unknown data from a large pool of sources, it is necessary to recognize the topic or domain covered by the source. One recently proposed use case for such source discovery is situational BI where warehouse data is complemented with data from openly available sources [3, 31]. Examples for such sources are the set of linked open data sources (`linkeddata.org`) or tables gleaned from the web: "Data on the Web reflects every topic in existence, and topic boundaries are not always clear." [12]

Topical profiling should be able to match a data set to a given set of topics or domains. Given two data sets, it should be able to determine topical overlap between them. There is already initial work on topical profiling for traditional databases in the iDisc system [44], which matches tables to topics or clusters them by topic, and for web data [8], which discovers frequent patterns of concepts and aggregates them to topics.

6. DATA PROFILING ON OTHER ARCHITECTURES

Most current data profiling methods and tools assume data to be stored in relational form on a

single-node database. However, much interesting data nowadays resides in data stores of different architecture and in various (non-relational) models and formats. If these architectures are more amenable to data profiling tasks, they might even warrant copying data for the purpose of profiling.

Storage architectures. Of all modern hardware architectures, columnar storage seems the most promising for many data profiling tasks, which often are inherently column-oriented: Analyzing individual columns for patterns, data types, uniqueness, etc. involves reading only the data of that column and thus matches precisely the sweet-spot of columns stores [1]. This advantage may dwindle when analyzing column-combinations, for instance to discover functional dependencies, but even then one can avoid reading entire rows of data.

As data profiling includes many different tasks on many tables and columns, a promising research avenue is the use of many cores, GPUs, or distributed environments for parallelization. Parallelization can occur at different levels: A comprehensive profiling run might distribute individual, independent profiling tasks to different nodes (task parallelism). Another approach is to partition data for a single profiling task (data parallelism). As most profiling tasks are not associative, in the sense that profiling results for subsets of column-values cannot be aggregated to overall results, horizontal partitioning is usually not useful or at least raises some coordination overhead. For instance, uniqueness within each partition of a column does not imply uniqueness of the entire column, but communicating the sets of distinct values is sufficient. Finally, task parallelism can again be applied to finer-grained tasks, such as sorting or hashing, that form the basic building blocks of many profiling algorithms.

Further challenges arise when performing data profiling on key-value stores: Typically, the values contain some structured data, without enforced schemata. Thus, even defining the expected results on such "soft schema" values is a challenge, and a first step must involve schema profiling as described in Section 5.

To systematically evaluate different methods and architectures for the various data profiling tasks, a corresponding *data profiling benchmark* is needed. It must define (i) a set of tasks, (ii) data on which the tasks shall be executed, and (iii) measures to evaluate efficiency. For (i) the first (single-source) subtree of Figure 1 can serve as an initial set of tasks. Arguably, the most difficult part of establishing a benchmark is to (ii) provide data that closely mirrors real-world situations. Given a schema and a set of constraints (uniqueness, data types, FDs, INDs, patterns, etc.) it is not trivial to create a valid database instance. If in addition some dirtiness, i.e., violations to constraints, are to be inserted, or if conditional dependencies are needed, the task becomes even more daunting. The measures for (iii) need to be carefully selected, in particular if they are to go beyond traditional measures of response time and cost efficiency and include the evaluation of approximate results. Finally, the benchmark should be able to evaluate not only entire profiling systems but also methods for individual tasks.

Types of data. Data comes not only in relational form, but also in tree or graph shapes, such as XML and RDF data. A first step is to adapt traditional profiling tasks to those models. An example is ProLOD, which profiles linked open data delivered as RDF triples [10]. A further challenge arises from the sheer size of many RDF data sets, so profiling computation must be distributed [9]. In addition, such data models demand new, data model-specific profiling tasks, such as maximum tree depth or average node-degree.

Structured data is often intermingled with unstructured, textual data, for instance in product information or user profiles on the web. The field of linguistics knows various measures to characterize a text from simple measures, such as average sentence length, to complex measures, such as vocabulary richness [25] as visualized in [29]. Thus, data profiling might be extended to text profiling and possibly to methods that jointly profile both data and text. A discussion on the large area of text mining is omitted, for the same reasons data mining was omitted from this article.

7. AN OUTLOOK

This article points out the potentials and the needs of modern data profiling – there is yet much principled research to do. A planned first step is to develop a general framework for data profiling, which classifies and formalizes profiling tasks, shows its amenability for a range of use cases, and provides a means to compare various techniques both in their abilities and their efficiency.

At the same time, this article shall serve as a "call to arms" for database researchers to develop more efficient and more advanced profiling techniques, in particular for the fast growing areas of "big data" and "linked data", both of which have attracted

great interest by industry, but both of which have proven that data is difficult to grasp and use effectively. Data profiling can bridge this gap by showing what the data sets are about, how well they fit the data environment at hand, and what steps are needed to make use of them.

Several research areas were deliberately omitted in this article, in particular data mining and text mining, as reasoned above, but also data visualization: Because data profiling targets users, effectively visualizing the profiling results is of utmost importance. A suggestion for such a visual data profiling tool is the Profiler system [28]. A strong cooperation between the database community, which produces the data and metadata to be visualized, and the visualization community, which enables users to understand and make use of the data, is needed.

Acknowledgments. Discussions and collaboration with Ziawasch Abedjan, Jana Bauckmann, Christoph Böhm, and Frank Kaufer inspired this article.

8. REFERENCES

[1] D. J. Abadi. Column stores for wide and sparse data. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, pages 292–297, Asilomar, CA, 2007.

[2] Z. Abedjan and F. Naumann. Advancing the discovery of unique column combinations. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 1565–1570, Glasgow, UK, 2011.

[3] A. Abelló, J. Darmont, L. Etcheverry, M. Golfarelli, J.-N. Mazón, F. Naumann, T. B. Pedersen, S. Rizzi, J. Trujillo, P. Vassiliadis, and G. Vossen. Fusion Cubes: Towards self-service business intelligence. *Data Warehousing and Mining (IJDWM)*, in press, 2013.

[4] D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Han, H. V. Jagadish, A. Labrinidis, S. Madden, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, K. Ross, C. Shahabi, D. Suciu, S. Vaithyanathan, and J. Widom. Challenges and opportunities with Big Data. Technical report, Computing Community Consortium, `http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf`, 2012.

[5] J. Bauckmann, Z. Abedjan, H. Müller, U. Leser, and F. Naumann. Discovering conditional inclusion dependencies. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 2094–2098, Maui, HI, 2012.

[6] J. Bauckmann, U. Leser, F. Naumann, and V. Tietz. Efficiently detecting inclusion dependencies. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1448–1450, Istanbul, Turkey, 2007.

[7] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Proceedings of the Conference on Advanced Information Systems Engineering (CAiSE)*, pages 452–466, Toronto, Canada, 2002.

[8] C. Böhm, G. Kasneci, and F. Naumann. Latent topics in graph-structured data. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 2663–2666, Maui, HI, 2012.

[9] C. Böhm, J. Lorey, and F. Naumann. Creating voiD descriptions for web-scale data. *Journal of Web Semantics*, 9(3):339–345, 2011.

[10] C. Böhm, F. Naumann, Z. Abedjan, D. Fenz, T. Grütze, D. Hefenbrock, M. Pohl, and D. Sonnabend. Profiling linked open data with ProLOD. In *Proceedings of the International Workshop on New Trends in Information Integration (NTII)*, pages 175–178, Long Beach, CA, 2010.

[11] L. Bravo, W. Fan, and S. Ma. Extending dependencies with conditions. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 243–254, Vienna, Austria, 2007.

[12] M. J. Cafarella, A. Halevy, and J. Madhavan. Structured data on the web. *Communications of the ACM*, 54(2):72–79, 2011.

[13] S. Chaudhuri, U. Dayal, and V. Ganti. Data management technology for decision support systems. *Advances in Computers*, 62:293–326, 2004.

[14] F. Chiang and R. J. Miller. Discovering data quality rules. *Proceedings of the VLDB Endowment*, 1:1166–1177, 2008.

[15] P. Christen. *Data Matching*. Springer Verlag, Berlin – Heidelberg – New York, 2012.

[16] G. Cormode, M. N. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3):1–294, 2012.

[17] O. Curé. Conditional inclusion dependencies for data cleansing: Discovery and violation

detection issues. In *Proceedings of the International Workshop on Quality in Databases (QDB)*, Lyon, France, 2009.

[18] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer Verlag, Berlin – Heidelberg – New York, 2007.

[19] S. M. Fakhrahmad, M. H. Sadreddini, and M. Z. Jahromi. AD-Miner: A new incremental method for discovery of minimal approximate dependencies using logical operations. *Intelligent Data Analysis*, 12(6):607–619, 2008.

[20] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. *ACM Transactions on Database Systems (TODS)*, 33(2):1–48, 2008.

[21] W. Fan, F. Geerts, J. Li, and M. Xiong. Discovering conditional functional dependencies. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 23(4):683–698, 2011.

[22] L. Golab, F. Korn, and D. Srivastava. Efficient and effective analysis of data quality using pattern tableaux. *IEEE Data Engineering Bulletin*, 34(3):26–33, 2011.

[23] L. Golab and M. T. Özsu. *Data Stream Management*. Morgan Claypool Publishers, 2010.

[24] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[25] D. I. Holmes. Authorship attribution. *Computers and the Humanities*, 28:87–106, 1994.

[26] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, 42:100–111, 1999.

[27] I. F. Ilyas, V. Markl, P. J. Haas, P. Brown, and A. Aboulnaga. CORDS: Automatic discovery of correlations and soft functional dependencies. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 647–658, Paris, France, 2004.

[28] S. Kandel, R. Parikh, A. Paepcke, J. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of Advanced Visual Interfaces (AVI)*, pages 547–554, Capri, Italy, 2012.

[29] D. A. Keim and D. Oelke. Literature fingerprinting: A new method for visual literary analysis. In *Proceedings of Visual Analytics Science and Technology (VAST)*, pages 115 –122, Sacramento, CA, 2007.

[30] S. Lopes, J.-M. Petit, and F. Toumani. Discovering interesting inclusion dependencies: application to logical database tuning. *Information Systems*, 27(1):1–19, 2002.

[31] A. Löser, F. Hueske, and V. Markl. Situational business intelligence. In *Proceedings Business Intelligence for the Real-Time Enterprise (BIRTE)*, pages 1–11, Auckland, New Zealand, 2008.

[32] M. V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3):191–221, 1988.

[33] F. D. Marchi, S. Lopes, and J.-M. Petit. Efficient algorithms for mining inclusion dependencies. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 464–476, Prague, Czech Republic, 2002.

[34] F. D. Marchi, S. Lopes, and J.-M. Petit. Unary and n-ary inclusion dependency discovery in relational databases. *Journal of Intelligent Information Systems*, 32:53–73, 2009.

[35] V. M. Markowitz and J. A. Makowsky. Identifying extended entity-relationship object structures in relational schemas. *IEEE Transactions on Software Engineering*, 16(8):777–790, 1990.

[36] T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice-Hall, 2nd edition, 1999.

[37] V. Poosala, P. J. Haas, Y. E. Ioannidis, and E. J. Shekita. Improved histograms for selectivity estimation of range predicates. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 294–305, Montreal, Canada, 1996.

[38] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[39] E. Rahm and H.-H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.

[40] V. Raman and J. M. Hellerstein. Potters Wheel: An interactive data cleaning system. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 381–390, Rome, Italy, 2001.

[41] A. Rostin, O. Albrecht, J. Bauckmann, F. Naumann, and U. Leser. A machine

learning approach to foreign key discovery. In *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB)*, Providence, RI, 2009.

[42] Y. Sismanis, P. Brown, P. J. Haas, and B. Reinwald. GORDIAN: Efficient and scalable discovery of composite keys. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 691–702, Seoul, Korea, 2006.

[43] K. P. Smith, M. Morse, P. Mork, M. H. Li, A. Rosenthal, M. D. Allen, and L. Seligman. The role of schema matching in large enterprises. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA, 2009.

[44] W. Wu, B. Reinwald, Y. Sismanis, and R. Manjrekar. Discovering topical structures of databases. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1019–1030, Vancouver, Canada, 2008.

[45] H. Yao and H. J. Hamilton. Mining functional dependencies from data. *Data Mining and Knowledge Discovery*, 16(2):197–219, 2008.

Anand Rajaraman Speaks Out on Startups and Social Data

by Marianne Winslett and Vanessa Braganholo

Anand Rajaraman
http://anand.typepad.com/datawocky/anand-rajaraman.html

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are in Phoenix, site of the 2012 SIGMOD and PODS conference. I have here with me Anand Rajaraman, who is an entrepreneur from the database research community. Anand was a cofounder of the data integration company Junglee, the semantic search company Kosmix, and the venture capital fund Cambrian Ventures. After Amazon acquired Junglee, Anand served as Director of Technology for Amazon.com. After Walmart acquired Kosmix, Anand became the senior vice president and co-head of @WalmartLabs. After leaving Walmart in 2012, Anand continues to invest in, mentor, and advise several Silicon Valley startups. He has a VLDB 10 Year Best Paper Award[1] and a SIGMOD Test of Time Award[2]. His PhD is from Stanford University. So, Anand, welcome!

Your two 10-year best paper awards are both for papers that you wrote in 1996, which was also the last year that you published a research paper! What happened?

[1] Querying Heterogeneous Information Sources using Source Descriptions. Alon Halevy, Anand Rajaraman, and Joann J. Ordille, 1996.
[2] Implementing Data Cubes Efficiently. Venky Harinarayan, Anand Rajaraman, and Jeffrey Ullman, 1996.

Sometimes people have these years they call *annus mirabilis* (miraculous years). So 1996 was my *annus mirabilis*. In 1996 I did two streams of research, which ended up with these best paper awards, one at SIGMOD and one at VLDB. But at the same time, I also came up with the idea for my first company, Junglee, together with some other students at Stanford. That ended up being the year that I took a leave of absence from the PhD program at Stanford to start my first company, Junglee, and after that, I never published any refereed research paper. (I misspoke: it turns out I do have a couple of research papers after 1996. But they have been sporadic).

How did your advisor feel about that?

My advisor was Jeff Ullman at Stanford, and you know, I really credit Jeff with everything that has happened to me since that time. We'd been doing all this research work on how to do data integration by combining all these enterprise databases; it was a big project at Stanford called TSIMMIS. A bunch of us had this idea that you could take some of these ideas, but not the exact technologies, and apply them to something new that was coming up called the World Wide Web. As we thought about it, it became clear to us that the right way to pursue this was not as a research project, but as a company, as a Silicon Valley startup. And we were also highly inspired by meeting the Yahoo! founders, who at around the same time had left Stanford and started a company as well. So I was kind of in two minds. Should I leave Stanford and start this company? Here I was, I had just published these papers, it looked like my research career was finally going to take off, and at the same time, I had this idea for a company, which I thought was a truly interesting idea that could change the world. So what should I do? So, I spoke to the person who I thought had the most insight and this happened to be Jeff Ullman. And Jeff said: "You know what, if you truly believe in this idea, then go make the company happen. Building a great product that many, many people use is far more impactful than writing a thesis, so just go ahead and make it happen".

It's never been so easy to collect data than it has been now, so never take the data as a given.

If I were cynical, I should ask you if he had shares in the company.

Well, that's a good question! At the time Jeff gave me the advice, he had absolutely no shares in the company or any interest in the company. Later on, as it turned out, as we got further along in the company, we actually added Jeff to the Board of Directors at Junglee, but this was much later. You know, maybe several months later after this conversation.

So, it seems that, in the intro, I should have said, "If Anand had a PhD, it would have been from Stanford?"

Actually, I do have a PhD, and it is from Stanford. But here's the story. I took a leave of absence in 1996, as I said, to start this company, Junglee. In 1998, Amazon.com acquired Junglee, so I went to Amazon and did a bunch of interesting things. But most importantly, I worked around how to get third party merchants selling on Amazon. The Amazon Marketplace that you see today is what the Junglee team did at Amazon. Then around 2000, I had this feeling that there was something incomplete, I needed to get closure on this whole PhD thing. So I came back to

Stanford in the year 2000, spent a year, and wrote up my thesis. But here's one thing that is still a sore point. When I came back, Jeff Ullman told me that he wouldn't actually give me a scholarship to finish my PhD. He said, "You know you've got to pay your own fees". So I did that.

That's so inconsiderate!

(Anand laughs)

I would say that among the successful founders of companies, you'd be in the minority in the fact that you came back and finished that degree. What was motivating you that other people didn't feel? Like the Google guys, they didn't come back and finish.

I guess they were far more successful than I was. If you look at the people who actually started companies that kind of took off, and then they stayed for a long time at those companies, they actually haven't ended up coming back to finish their PhDs. In my case, two things happened. One is that my company got acquired within a relatively short time after I started it, and so my research was still fresh, so I could come back and complete my thesis, so that was good. And the other interesting thing that happened was that I really wanted to finish this, so I just did it.

What will e-commerce be like five years from now?

Do you remember the time before e-commerce when you actually had to go to the stores to shop? Then e-commerce sort of happened in the early 90's, and there was a huge change in the way people shopped, right? So the way we shopped changed fundamentally with e-commerce, and a fundamental change, as fundamental as that change, is just happening now to the way we shop. That's kind of driven by two factors. One is social, and the other is mobile. These days, more and more shoppers are carrying smartphones, and they use these smart phones, not necessarily to make phone calls, or to check the weather, but also to compare prices, and find where to buy products. We spend more and more time on social media, and what our friends say about what products they buy and so on deeply influences our purchase behavior. So because of social and mobile, e-commerce is going to change fundamentally, and it is going to be as big a transformation as e-commerce was.

There are two distinct worlds today. There is the world of e-commerce, and there is the world of retail commerce, where you shop offline. Because of mobile, these two worlds are going to merge together. The distinction between what we call e-commerce and what we call retail is going to go away. And it is going to be one seamless customer experience. Customers won't care or won't even know sometimes that they are shopping online or offline. For example, you could go to a shop, see that the product that you want is out of stock, and order it online and have it shipped to your home. Do you call that retail or do you call that e-commerce, right? Or you could go online and have a product shipped to your nearest store and you can go pick it up there, now is that retail, or is it e-commerce? So all kinds of interesting combinations will come into play that will completely blur the line between e-commerce and retail, and this whole category of e-commerce is going to go away, there is just going to be commerce.

Now I'm confused, because the two examples you gave already exist. For example, if you shop at Talbot's and what you want isn't there, they do that and have it shipped, and the reverse direction also works. So where's the new angle?

Right, so, all these are trends that are starting to happen. There are early experiments in these things by a few online retailers. But these will become the new reality over time. The mobile will be an incredibly important part of the retail experience. Today, there are a lot of things, for example, when we shop online, there is a lot of stuff that we take for granted. For example, we read reviews, and we see what other people have said, and so on. Yet, when we go into a store, we have none of those things. We just see shelves of products, right? So if you think about the first generation of e-commerce, it was all about taking the products that were in the store, and bringing them to the web. The second generation, now of commerce, is going to be taking all the information about products that's online and bringing them into the store through the mobile phone, and then using your social identity to connect the rest of your behavior with your shopping behavior.

One of the most quoted examples from e-commerce is Amazon's feature of what you should read based on what other people similar to you have read. I was at Amazon at the time when they launched that, and it is truly a brilliant feature. If you think about it, the only information that Amazon, or any other e-commerce site has access to right now is your shopping behavior on that site. Yet, there is so much of our life beyond what we spend at any one website. And that behavior has more and more been captured in social media streams like Facebook and Twitter. So if you can combine the information that's in Facebook and Twitter about us together with the information that the retailer has, and deliver all those recommendations and the better search experience through mobile, that's going to be truly revolutionary.

[…] the most successful uses of big data […] use all the data to answer the questions. They don't ever throw away the data, they are kind of "model light and data rich".

So speaking as an introvert here, how will that make my life better? Except, for example, maybe if I am buying a car, or some other mega-purchase? If I'm buying socks, how's that going to make my life better?

Well, I'll give you an example that happened to me: I work out and I run, and my feet blister easily, so I needed to find socks that would not blister. So I asked my friends, and they told me, "we also run, and these are the socks to buy". Now, it would be nice when you are in a store to ask your friends right from there: "which socks should I pick up"? These are the kinds of things that you might find interesting, for instance. How do you connect with your friends when you are in the store, how do you leverage recommendations, how do you leverage the wisdom of your friends as well as the whole community when you are shopping, in a better way? How do you get personalized recommendations? For example, let's say you're traveling somewhere, and you just happened to go into a store that has the right

guidebook for where you are traveling. Well it might be interesting for you to get an alert to your phone saying, "hey, you know, the product you are looking for is right here".

Speaking of Amazon, where did Amazon's Mechanical Turk come from?

That's an interesting story. I told you that I left Amazon around the year 2000 and came back to Stanford to complete my PhD. At the same time as I was working on my PhD, together with another Junglee cofounder from Stanford, Venky Harinarayan, we started what we called an idea incubator, called Cambrian Explosion, which is an arm of Cambrian Ventures, a venture capital firm. With Cambrian Explosion, we were interested in coming up with new ideas that could potentially become interesting business. And one of the ideas that we were playing around with at the time was this idea of how we combine humans and machines to complete interesting tasks. What we observed (this was around the year 2000) is that computers are great at doing some things, but there are some things that computers are terrible at doing that humans do effortlessly, like image recognition and things like this. So we thought that if we could combine humans and computers, and create what we call hybrid human-machine computation, we could solve a wider area of problems. So we sort of came up with this idea, and found a couple of entrepreneurs, who were in fact willing to take this idea forward. We wrote up a patent called Hybrid Human-Machine Computation, filed it in 2000, and started a company to take the idea forward.

Our idea at the time was we could build software that would enable companies to write systems combining humans and machines in interesting ways. So we had these two founders of this company who were going to do this, and they were talking to a whole bunch of potential customers to see whether they could use humans and machines together to solve interesting problems and so on, and we were getting some interest. But, as it turns out, just around this time, 9/11 happened, and companies stopped trying to do new things. Kind of, the bottom fell out of innovation around that time. And so, it sort of became apparent to us that this company that we had started around hybrid human-machine computation wasn't going anywhere. The two entrepreneurs with whom we were working on that came up with a different idea they got more passionate about.

So here we were: we were sitting on this idea that we thought had potential, but we had no people to take it forward. This was when we had a chat with Jeff Bezos. Incidentally, when we left Amazon, Jeff Bezos wanted to stay engaged with us, and was in fact, the biggest investor in our venture capital firm, Cambrian Ventures. When we told him about this idea about hybrid human-machine computation, he got incredibly excited. He said "look, I'd like to take this idea forward. Why don't you guys sell me this patent?" So we sold him the patent on hybrid human-machine computation, and that became the basis for Amazon Mechanical Turk. So the name "Amazon Mechanical Turk" is entirely Jeff Bezos's. We had nothing to do with it. Jeff's genius in this was to take that idea, and combine it with the idea of a market place. It was sort of saying that you could have this marketplace of humans, and you could create these tasks and you could put it out there, so that was his thinking. And then Amazon executed very well on the idea, and it became quite successful. So that was our contribution to Amazon Mechanical Turk.

You have claimed that more data almost always beats better algorithms. Why is that?

You know, we live in a world where there's more and more digital data that's being created. And usually people pull out statistics about how data is growing at 50% year over year. But my favorite quote on this is from Eric Schmidt who said that every 2 days now, we create as much data as was created from the dawn of civilization until 2003. That's a huge amount of digital data that's being created. When I think about how to solve difficult problems, I always think about how do I leverage all this data to solve that difficult problem. Now, if you think about data driven applications today, most of them follow a certain paradigm. You sort of create your favorite machine learning model, whether that's support vector machines, or regression, or whatever it is, and then you use all this big data as training data to train this algorithm. Then, once you have the algorithm, which is the trained model, the parameterized model, you through away all the data, and then you just ask the questions directly to the model. What a waste! Because you've thrown away all this data, and you've tried to capture everything, all the intelligence, in this model.

It's a well-known phenomenon that as you keep throwing more and more training data at a given machine learning model, the precision-recall performance of the model saturates at a certain point. At this point, if you want to get better at prediction, the only thing you can do is to make the model more complex by adding more features. But the problem is, the more complex you make the model, the more likely you are to be wrong. Just because the world is a fundamentally complex and a changing place, and all this complexity in the model probably means the world has diverged away from the model over time. So, if I think of the most successful uses of big data, like Amazon's recommendations, which is an example of collaborative filtering, or Google search, which I think is the best data driven application out there, both of these applications use all the data to answer the questions. They don't ever throw away the data; they are kind of "model light and data rich". I think that that's the right paradigm to think about how to leverage big data. Never throw it away once you've trained a model, keep it around all the time and use all of it to do every task, and come up with light thin models that are like icing on top of the data rather than try to replace the data by a model.

… we live in a world of big data, and there's never been a better time for startups around the idea of data.

What about things like smoothing that help you model the data that doesn't yet exist.

That is a very good point. One of the things that you run into, especially with high dimensional data sets, is the sparsity problem. When you try to find nearest neighbors in high dimensional data, if you have a certain number of data points, and the dimensionality of your data cell increases, then they, on average, get further and further away, so finding nearest neighbors becomes harder and harder. In my experience, one of the best ways I've found of dealing with this is through dimensionality reduction, to the extent possible, and then to just keep getting more data. Throwing more and more data into this mix. I think smoothing is a way of compensating for the lack of data, but we are transitioning from a data poor world into a data rich world. So while compensating for lack of data is interesting, I think we should be thinking about how to leverage all this extra data that's coming online.

In Google's case, don't they use hundreds of features, isn't that very high dimensional already?

I am not entirely familiar with the details of the technology behind Google search. I am sure they use hundreds and hundreds of features, but the key is that the data is fundamentally the lever, and the algorithms are the fulcrums, it's not the other way around. They don't talk about training data, the index is not the training data, the index is the data, and it answers every question.

Well, how can a database researcher know when the payoff is in collecting more data, and when to focus on modeling the part they haven't seen? I mean, the fatter the tail, the more you'll never see, to how do you know whether you should work on a model or work on getting more data?

I think it depends on the problem you're solving. So there's definitely no "one size fits all". But the one thing that I would say is that it's never been easier to get more data. So the way I like to phrase it, is don't ever take the data as a given when approaching a problem. I teach students in the data mining class at Stanford as well, and many students tend to approach the data as a given. The data is never a given. You can always collect more data. It's never been more easy to collect data than it has been now, so never take the data as a given. Always look for complimentary data sets, or additional data sets. I think time spent doing that is usually more rewarding than time spent designing more complex algorithms.

We talked a lot about startups. Do you have any advice for database researchers who would like to have a startup?

Well, they should just come talk to me! Seriously, what I mean is, you know, we live in a world of big data, and there's never been a better time for startups around the idea of data. If there is any database researcher who wants to start a company, the time is now, there's no time like the present. And I am happy to sort of talk to any of them, and help figure out how to take it forward. But, I think there are huge opportunities in the area, specifically around big data, in the infrastructure layer. And there is another trend that I'm sort of starting to see merge around fast data, which is data that's big but data that's moving faster and its real time. For this, there are opportunities in the infrastructure layer, in the algorithm layer and in the application layer, so there's huge opportunities, and now's a great time to be doing startups.

Well, these startups are usually West coast US, what about for all the people in our audience who live in other parts of the world?

Move to Silicon Valley! It worked for Mark Zuckerberg.

So, proximity is key?

Well, I think it's not necessarily about proximity. I think Silicon Valley has a great ecosystem that helps startups succeed.

What about Bangalore?

You know, I do see some interesting startups in Bangalore, I was just in Bangalore about a month ago, and I met with some very interesting startups there.

Beijing?

I have not been to Beijing, so it is hard for me to tell.

Maybe in time, there will be places other than Silicon Valley.

It is quite possible. And I know Silicon Alley in the New York area is immerging as an interesting startup hub as well. But I've found there is no place to beat Silicon Valley.

Right now you are at @WalmartLabs. What's that extra "at" there for?

Sure, you know how on Twitter and on Facebook when you want to address someone, you put an @ in front of their name? It's sort of a handle. So we built @WalmartLabs in the same sense, because @WalmartLabs is all about combining social into commerce, so we thought we'd sort of make a point by putting the @ in front of our name. And that also happens to be our handle, so that you might want to follow that handle on Twitter.

What are you guys doing with social media?

We are doing experiments on how is commerce best done using social media. For example, one of the experiments that we've done is something called Shopycat[3]. This is a Facebook App that we launched for the last holiday season, and what this Facebook App does is that it sort of takes the pain out of gift giving. So in the holiday season, we all want to give gifts. We have so many people in our lives, and we want to give them thoughtful gifts, not just a gift card. You want to give a thoughtful gift, and a gift that you think they are interested in. But how do we keep track of all that? Well it so happens that we tell our friends on Facebook everything that we're doing. And there's a set of information in there to figure out your hobbies, your interests, and so on. So what Shopycat does is for each of your friends, it figures out what their hobbies and interests are, combines them with a giant gifting catalog, and comes up with interesting gift suggestions for each of them. For example, you might find out that one of your friends is into hiking and the other is into running, and you can give them a different pair of shoes, hiking shoes or running shoes. And if you have a younger relative, you can find out that she's into the Hunger Games, and you can get her some Hunger Games memorabilia.

I think this data about human beings […] is going to create a revolution that's as fundamental or more fundamental than the industrial revolution.

Is it true that you were offered a chance to buy Google and turned it down?

[3] https://www.facebook.com/Shopycat

That's an interesting story. Remember, this is back in the year 1998, when the company that we had cofounded, Junglee, was in the process of being acquired by Amazon. So we had sort of agreed to be acquired by Amazon, but the deal had not closed yet, and around the same time, Sergey and Larry were getting started with Google. But they hadn't quite figured out how to make it a big company or whether it was going to be a big company even at that point in time – that was back in 1998. So it so happened that Sergey's advisor is also Jeff Ullman, who's my advisor. And Jeff connected us to Sergey and Larry and then he mentioned they were trying to figure out what to do, and perhaps Junglee might be interested in acquiring the company. The search technology at that time was relevant to what we were doing, you know, we were doing product search, they had some web search, maybe there was some synergy and so on. So it seemed very interesting to us. The problem for us is that we were in the process of being acquired by Amazon, so when you are in the process of being acquired yourself, you can hardly go around acquiring other companies. So that is why we couldn't do it at that time. Interestingly, there was another incident in the year 2000, or maybe in the year 1999, when we were at Amazon and we were talking to Jeff Bezos, and we were seeing Google starting to take off. This was the early days, but we could see the potential, and we convinced Jeff Bezos that Amazon should acquire Google. So Jeff Bezos sent me down, together with a couple other people to visit Google headquarters, which were in Palo Alto, and try to buy them. We were authorized to offer up to 300 Million dollars to buy Google, but when we met Sergey and Larry, they wouldn't budge for anything less than a billion dollars. So that didn't happen either.

Has being married to a fashion designer improved your fashion sense?

What do you think? (He laughs.) No, seriously, it is great fun being married to a fashion designer, because that's very remote from database technology, as you can imagine. And it gives you a different perspective on life. You know, I especially like the fashion shows and being able to go back stage during the fashion shows, and all that stuff. It's just different. And the parties are a lot more fun!

The distinction between what we call e-commerce and what we call retail is going to go away.

Maybe you can get some invitations for some members of our community! This cross-fertilization is probably good.

I would be happy to!

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

You know, I would personally get my hands dirty and play with big data more than I am. At @WalmartLabs, we've set up this giant, big fast data cluster, with many many nodes. There is lots of interesting data analysis going on that combine Walmart's data with Twitter and Facebook and so on. Fascinating stuff. And I wish I had the time to actually go do some of that myself. Unfortunately, when you get to a point when you are managing a large organization like

this, you tend to play more of an advisory role to the people who are actually doing the really fun stuff. So I wish I had more time to do some of that stuff myself.

If you could change one thing about yourself as a computer science researcher, what would it be?

You know the one thing that I would love to do more is to actually spend more time being a computer science researcher. My career, as you mentioned, has been in startups and in venture capital. When you are a venture capitalist it turns out you can actually spend a lot of time doing interesting stuff because there's not much to do otherwise. But when you are running a startup, or when you are working for a company, you don't have that much time to do real research. I try to spend as much time as possible at Stanford, in fact, I teach a class there on data mining. And I dearly love interacting with students, and I wish I could do more of that, and do more computer science research, and come to more conferences like SIGMOD and interact with the great people here. You know, I find it so refreshing to be able to do that, I wish I had more time to do that.

If you had that time, would you work on big data and social media, or would you pick a different topic, something different from your current day job?

I definitely think social media and social data is something really huge. The way I think about this is the following. If you go back a few hundred years, to the 16th century, there was this guy called Tycho Brahe and he observed the heavens and he jotted down the positions of the moons of Jupiter and all these things in a big book, and that I think was the first real database. And it lead to wonderful things, like Kepler's laws of planetary motion, and Newton's equations, and it lead, indirectly, to the industrial revolution, which changed the way we live. Now, if you think about all the advances that have happened in Physics, and in various other fields, that have actually been transformative for the world, many of them have started from physical observations of phenomena that are in the cosmos and all around us. What's been lacking until now, when we wonder about the laws of cosmos, and the laws of physics, we lack a fundamental understanding about human beings and human societies, and what makes us tick. And for the first time in our lives, due to social data, we have more data about human beings than ever before. I think this data about human beings is actually more valuable than the data about the cosmos, and it is going to create a revolution that's as fundamental or more fundamental than the industrial revolution. And if I can in some way play a small part in that, that's what would give me the greatest pleasure.

That is really exciting. So we have a lot of young readers who may be reading what you say and inspired by it, and then the next question in their mind would be how do I get access to this incredible data set? So how do they, how can they do that if they don't work for Facebook, Twitter, etc.?

That's right. One of the nice things about social platforms is that, for example, Facebook has a platform where you can create a Facebook App, and if you can get people to install your Facebook App, then you get access to their data. So I would encourage people to start creating Facebook Apps that are useful for people to use, and then that gives them access to data of people, which they can then use. So that is one way of gathering data on Facebook. On Twitter,

you can license Twitter data, or you can license them on relatively cheap terms. And I would highly encourage pretty much every university to go get cracking on that.

Well, thanks very much for talking with me today.

Thank you, it has been a pleasure, Marianne.

Data Management Research at the Technical University of Crete

Stavros Christodoulakis, Minos Garofalakis, Euripides G.M. Petrakis, Antonios Deligiannakis, Vasilis Samoladas, Ekaterini Ioannou, Odysseas Papapetrou, Stelios Sotiriadis
School of Electronic and Computer Engineering, Technical University of Crete

1. INTRODUCTION

The Technical University of Crete (TUC, www.tuc.gr) founded in 1977 in Chania, Crete is the youngest of the two technical universities in Greece (the other being the National Technical University of Athens). The purpose of this state institution is to provide high-quality undergraduate as well as graduate studies in modern engineering fields demanded by the Greek and international job market, to conduct research in cutting edge technologies as well as to develop links with the Greek and European industry. Today, the Technical University of Crete comprises five Engineering Schools (Electronic and Computer Engineering, Production Engineering and Management, Mineral Resources Engineering, Environmental Engineering, and Architecture). The School of Electronic and Computer Engineering (ECE) at TUC (www.ece.tuc.gr) has achieved an excellent reputation for its research and teaching internationally. The department accepts about 150 undergraduate students each year and employs 28 full-time faculty members. More than 75% of the ECE faculty members have obtained their Ph.D. degrees in top-rated foreign Universities, and several held academic or research positions abroad for many years prior to joining TUC. Faculty credits include multiple best paper awards at the ACM and IEEE Society level, professional recognition in terms of associate editor and technical committee member appointments, and leadership in conference organization here and abroad. Many TUC ECE graduates have pursued graduate studies at TUC and abroad. Their ranks include faculty members at top-rated North-American and European Universities, researchers at University, Government, and Industrial Research Labs, and successful professional engineers across Greece and Europe.

Data-management research at TUC revolves around a broad and diverse range of topics, ranging from fundamental algorithmic techniques (e.g., for managing streaming and probabilistic data) and tools for big-data analytics, to cloud database architectures, digital libraries, and the semantic web. In this short article, we present an overview of some recent and ongoing data-management research efforts at TUC-ECE. We structure our discussion by grouping our research activities under each of the three main data-management research labs at TUC-ECE: The Software Technology and Network Applications (SoftNet) Lab (www.softnet.tuc.gr, headed by Prof. Minos Garofalakis), the Intelligent Systems Lab (www.intelligence.tuc.gr, headed by Prof. Euripides G.M. Petrakis), and the Distributed Multimedia Information Systems and Applications (MUSIC) Lab (www.music.tuc.gr, headed by Prof. Stavros Christodoulakis).

2. SOFTNET LAB

Continuous Monitoring of Distributed Streaming Data. Large-scale stream processing applications rely on continuous, event-driven monitoring, that is, real-time tracking of measurements and events, rather than one-shot answers to sporadic queries. Furthermore, the vast majority of these applications are inherently distributed, with several remote monitor sites observing their local, high-speed data streams and exchanging information over a communication network. This distribution of the data naturally implies critical communication constraints that typically prohibit centralizing all the streaming data, due to either the huge volume of the data (e.g., in IP-network monitoring), or power and bandwidth restrictions (e.g., in wireless sensornets). Finally, an important requirement of large-scale event monitoring is the effective support for tracking complex, holistic queries that provide a global view of the data by combining and correlating information across the collection of remote monitor sites. Monitoring the precise value of such holistic queries without continuously centralizing all the data at first seems hopeless. Given the prohibitive cost of data centralization, it is clear that realizing sophisticated, large-scale distributed data-stream analysis tools must rely on novel algorithmic paradigms for processing local streams of data *in situ* (i.e., locally at the sites where the data is observed). This, of course, implies the need for intelligently decomposing a (possibly complex) global data-analysis and monitoring query into a collection of "safe" local queries that can be tracked independently at each

site (without communication), while guaranteeing correctness for the global monitoring operation. This decomposition process can enable truly distributed, event-driven processing of real-time streaming data, using a push-based paradigm, where sites monitor their local queries and communicate only when some local query constraints are violated. Nevertheless, effectively decomposing a complex, holistic query over the global collections of streams into such local constraints is far from straightforward, especially in the case of non-linear queries (e.g., joins).

A useful tool for monitoring complex non-linear queries over distributed streams is the recently proposed geometric approach [15, 8]. In a nutshell, the geometric method enables the monitoring of complex non-linear functions expressed over the *average* of data vectors maintained at distributed sites. The monitoring is made possible by having each site monitor a geometric condition over the *domain* where the average vector lies, rather than monitoring the range of the function. These local geometric constraints are designed to guarantee that, if the monitored condition on the global function is violated, then at least one of the local constraints must be violated, that is, at least one of the remote sites will fire a *local violation*. Thus, no global violation can go undetected.

Our recent work, in the context of the LIFT EU-FET Open project (`www.lift-eu.org`), builds on the geometric framework in order to solve a variety of complex distributed stream monitoring problems, including: Detecting outliers in sensor networks by monitoring the pair-wise similarities (which can be expressed as a wide range of functions, including, for example, L_k norms, cosine similarity, and Extended Jaccard coefficient) of neighboring sensor nodes [3]; efficiently monitoring complex functions by combining the use of *prediction models* with the geometric approach [6]; monitoring sliding-window queries by efficiently summarizing streaming data over sliding windows with probabilistic accuracy guarantees [14]; enriching the geometric approach with *sketch synopses* [4] to efficiently track a broad class of complex queries (including, general inner products, self-join sizes and range aggregates) over massive, high-dimensional distributed data streams with provable guarantees [5]; monitoring continuous fragmented skyline queries over distributed data streams [13]; and, proposing novel techniques for defining improved *safe zones* (i.e., safe regions of the domain for local data vectors) for distributed monitoring problems [7]

Our ongoing work in the area of (centralized and distributed) data-stream management focuses on novel extensions of the technology and tools to handle the challenges of (1) large-scale Complex Event Processing (CEP) systems (in the context of the upcoming FERARI EU-STREP project), (2) massive brain data analytics (in the context of the EU FET-Flagship Human Brain Project, `www.humanbrainproject.eu`), and (3) new, elastic software/hardware architectures for effective data-stream analytics (in the context of the upcoming QualiMaster EU-STREP project).

Data-as-a-Service (DaaS) in Microcloud Federations. Collecting, storing, and processing public web-size data, such as the web graph and public data from social networks, has for long being an exclusive privilege of a few large companies world-wide that have the capacity to construct and maintain huge server farms. Towards enabling small and medium companies to perform management and mining tasks on data of such magnitude, we have recently, in the context of the LEADS EU-STREP project (`www.leads-project.eu`), started exploring an innovative cloud model, called Data-as-a-Service (DaaS). The model enables companies to use shared cloud resources for storing and accessing public and private data, and for performing arbitrary processing tasks on this data. The targeted infrastructure in our case is an elastic set of distributed microclouds, combined to create the illusion of a large unified cloud.

The considered scenario has several key benefits compared to traditional in-house solutions. First and foremost, companies can share the acquisition (e.g., crawling) and storage cost of the public data. Results of common processing tasks, such as the PageRank scores of web-pages or the influence factors of users in social networks, can also be shared across platform users. Second, companies can use a pay-as-you-go charge model, without requiring upfront investment. This enables small companies to test innovative, high-risk, ideas, without a substantial investment. Last, sharing of the infrastructure reduces the idle time of the participating nodes, promoting green computing and reducing the platform's running cost.

The model also comes with a novel set of challenges. Probably the key concerns for companies are the correctness of the data and results, and the privacy of sensitive data. Therefore, in a recent paper we have considered the problem of verifying the correctness and freshness of query results on data streams, necessary in the existence of malicious or misbehaving nodes in the network [12]. Our solution induces a very small overhead, and is readily applicable to generic cloud setups. In the same context, we recently started investigating the problem of data analytics on private, encrypted, data in the cloud. Our recent results (working paper) show that many of the queries necessary for powerful data analytics can in fact be executed without information leakage, directly in the cloud.

The physical distribution of the individual clouds in the considered infrastructure offers many optimization

opportunities. Data is partitioned on servers distributed across the world, each one with different (possibly even varying) computational capacity and charging policies. By controlling the data placement and replication, the location of the processing tasks, as well as the network interaction between the nodes, we can drastically reduce the running cost of the platform. Keeping this in mind, we are now developing novel distributed algorithms for frequent data processing tasks over both streaming and stored data, which rely on approximation and on in situ processing. Preliminary results on maintenance of PageRank scores for the web graph show that these techniques substantially reduce the cost without noticeable impact on quality (working paper).

Uncertain/Probabilistic Data Management. We are also working on developing techniques for efficient management of uncertain data, originating, for example, from information extraction and resolution processes. The majority of the work on this topic is done in the context of the HeisenData project (`heisendata.softnet.tuc.gr`), which aims at extending the traditional relational table store with support for a broad class of statistical models and probabilistic-reasoning tools.

Part of our research focus involves efficient query processing over data extracted from unstructured sources [23, 22, 24]. For instance, the possible extractions can be represented using the Conditional Random Field (CRF) statistical model [18], and inference over such a CRF provides the final extraction results. The system presented in [24], is a probabilistic framework that allows performing such extraction tasks. It uses a linear-chain CRF with the Viterbi inference algorithm and query processing returns the maximum-likelihood extraction results. The in-database implementation of extraction tasks, as introduced in [18], improves the quality of query results as well as the efficiency of query processing since it enables the incorporation of several optimizations. The approach in [23] considers additional inference algorithms, such as variations of the sampling-based Markov chain Monte Carlo. It also proposes mechanisms for choosing the most suitable inference algorithm based on the given data, model, and query. We are currently working on further improving quality and efficiency by combining additional extraction and database activities, including coreference, canonicalization and optimizations using algebraic equivalences. Providing such a system requires addressing several challenges, such as efficiently executing the inference process on the potentially large graphical models that will be created. To enable large-scale probabilistic data analysis, we have recently developed a novel MapReduce algorithm. It efficiently executes exact inference on large graphs by taking advantage of the parallel nature of exact inference both structurally and computationally.

With respect to management of data from entity resolution methods, we are considering probabilistic unmerged duplicates specifying which objects can be merged. More specifically, we proposed an entity-join operator that allows expressing joins between the tables containing unmerged duplicates with other tables from the database. The focus is on analytics that allow users to express aggregation and iceberg queries over the massive collection of "possible resolution worlds". Processing is based on a novel indexing structure that allows efficient access to the resolution-related information and a set of techniques for evaluating complex queries that include qualifiers for retrieving analytical and summarized information and moving towards a higher level of detail. An extension of this work is to reduce the time required for query processing by considering also a set of apriori merges along with the on-the-fly merges created during query processing.

3. INTELLIGENT SYSTEMS LAB

During the latest years we have witnessed the rapid growth of cloud computing that delivers leased services to everyday Internet users. Various provisioning models have been defined to separate accessibility and content of services. A fundamental taxonomy includes infrastructural services (Infrastructure as a Service-IaaS), software based services (Software as a Service-SaaS) and development platforms (Platform as a Service-PaaS). Initially, IaaS is related to active and virtualized services that scale dynamically while SaaS refers to applications that are already hosted in cloud datacenters. At last, PaaS could be seen as an outgrowth of SaaS, wherein applications are available in a development platform environment where users could implement their own cloud based solutions. Cloud computing has been proven to be a novel approach, for many cases, regarding to the minimization of operational and monitoring costs while it increases elasticity. However in the healthcare domain, in particular, there are standards, regulations and recommendations (e.g., national legislation, ISOs and security standards). These stress severe restrictions for data transfer, storage, aggregation and analysis. For the case of cloud computing a typical requirement is that services presumed to be stored in remote datacenters, while the data storage happens, as well, remotely. This raises obstacles and the utilization of cloud capabilities by healthcare domain seems challenging.

We are taking part in the Future Internet Public-Private Partnership (FI-PPP, `www.fi-ppp.eu`) of the EU (a European programme for Internet-enabled innovation). The ongoing phase 2 of FI-PPP includes 5 use case trials, one of which is the FI-Star project (`www.fi-star.eu`) that aims at establishing early trials in the Health Care domain building on Future Internet (FI) technology lever-

aging on the outcomes of FI-PPP Phase 1. FI-STAR adopts a fundamentally different, "reverse" cloud approach that is bringing the software to the data, rather than bringing the data to the software [19] as a means of developing Future Internet (FI) applications. At a glance, cloud services (SaaS) could reach the local user infrastructure and utilized in an on demand model while services are deployed locally. This highlights new challenges in the area of designing FI applications for sensitive domains like the e-health domain where data security and confidentiality raise obstacles on data processing (e.g., data may not migrate to a public cloud and need to be processed locally).

Within the FI-Star project, we focus on exploiting a SaaS cloud model for deploying an architecture solution [17] that resorts to FI-WARE . FI-WARE is a core software platform that eases creation of innovative Future Internet (FI) applications by offering reusable modules named as Generic Enablers (GEs). Generic Enablers (GEs) are considered as software units (the core building blocks of a cloud application) that offer various functionalities along with protocols and interfaces for operation and communication.

Particular emphasis is given to interoperability and portability of cloud services. This drives from the need of translating services across different cloud providers and the need for service discovery within each cloud platform. Thus, providers could allow services to communicate and interoperate. Service interoperability refers to the interoperation of services across multiple clouds using a common management API while, system portability defines the ability of cloud services to be deployed on other cloud service of a lower service model (for instance a SaaS to be integrated in a PaaS). Our work is also closely related with the case of portability wherein an application needs to be ported in the cloud, and this is particularly interesting for utilizing the legacy applications and systems. We are motivated from the exploration of semantic annotated descriptions in order to assist cloud porting in terms of service automate operations as well as assign rich content and relationships. Practically, this requires definition of the service (for instance using service descriptions) in order to be translated and be compatible with a cloud.

One of our research aims is to explore ontology-based solutions, for representing cloud services specific concepts, attributes and relationships. In recent work we have presented an analysis to define the requirements for interoperability and portability in various cloud deployment models. A future task is to design service descriptions of GEs and automate the service discovery process. This will offer significant advantages for locating appropriate GEs that integrate FI applications.

Recent work includes developments in our lab's state-of-the-art cloud setting (termed Intellicloud, `cloud.intellicloud.tuc.gr`) that will offer several kinds of provider services for FI application development including GEs. Intellicloud is based on Open Cloud Computing Interface (OCCI, `occi-wg.org`) standard that is an open specification and API for cloud offerings and it is aligned with FI-WARE. OCCI promotes developments in the area of interoperability and portability, areas that we perform our research. Currently, Intellicloud offers IaaS and PaaS services (that integrate GEs) to researchers for experimentation and implementation of FI applications (e.g., healthcare services)

In many instances, healthcare services have been developed based on the IoT paradigm that enables devices to be represented in an Internet like structure. In FI-STAR, provider cloud services will manage and upgrade functionalities of GEs and will be deployed in a customize way that matched the health care use case constraints. Fundamentally, this includes the cloud management for supervision, underlying infrastructure, the utilization of various IoT devices for data collection, provision of APIs (e.g., tools for data analytics) and communication among interfaces. This in combination with edge computing expands clouds functionality by allowing business logic and process management to happen at the actual source similar to a distributed computing fashion. This characterizes an alternative view of clouds where services can reach user premises and utilized directly in users personal IoT devices. We also focus on the exploration of healthcare provisioning models and approaches. For instance we have designed decision support system for patients suffering from Bipolar Disorder. Thus, the adaptation of performed work by utilizing emerging technologies highlights an area of our new challenges.

We vision that edge computing could offer cloud capabilities for remote data storage and management, while local data processing will facilitate a self-adaptive environment for data extraction and analysis. In such solution, legacy or on-the-fly developments will need to be imported to the cloud infrastructure and to interoperate in both local and remote clouds. This means that users software and APIs will need to communicate successfully and understand the new system constraints, a case that highlights cloud portability.

4. MUSIC LAB

Semantic Web Interoperability. The dominant standard for information exchange in the web today is XML and many important international standard have been expressed in XML and its derivatives. The emerging Semantic Web (SW) world however is based on different models and languages. We investigate methodologies for bridging the gap between the Semantic Web and the

XML world. A survey and comparison of recent technologies and standards in the XML and SW world as well as the data integration and data exchange systems between the two appears in [1]. We have developed the XS2OWL Framework ([1]) which provides a transformation model for automatic and accurate expression of the XML Schema Semantics in OWL Syntax. In addition it allows it allows the transformation of the XML data in RDF format and vice versa. The current version of XS2OWL exploits the OWL 2 semantics (like identity constraints) and supports the new XML constructs introduced in XML Schema 1.1. We have also developed the SPARQL2XQuery Framework for the translation of SPARQL to XQuery [2]. Although several systems offer SPARQL end points over relational data there are no systems supporting XML data. The Framework provides a formal model for the expression of mappings from OWL to XML Schema and a generic method for SPARQL to XQuery translation, thus providing an important part of ontology based integration involving XML resources in the Linked data environment.

Driven by the fact that Semantic Web is comprised of distributed, diverse (in terms of schema adopted) and in some cases overlapping RDF datasets, we are investigating generic frameworks supporting query answering in federated architectures. To this end, the SPARQL–RW Framework [9] provides a formal method and implementation for SPARQL query rewriting with respect to a set of predefined mappings between ontology schemas. The supported mapping model has been formally described using Description Logics and allows the definition of a rich set of mapping types. Our Framework is proved to provide semantics preserving queries to the nodes.

Finally, important international standards such as MPEG-7 for multimedia content descriptions do not describe a formal mechanism for the systematic integration of domain knowledge in the MPEG-7 descriptions. We have developed a formal model for domain knowledge representation within MPEG-7 [20]. The model allows the systematic integration of domain knowledge in MPEG-7 descriptions using only MPEG-7 constructs thus maintaining interoperability with existing MPEG-7 software.

Meta Data Management, Interoperability and Linked Data Publishing in Museum Digital Libraries. In addition to web presence today museums are also very interested in interoperability with generic or domain specific large international metadata publishers as well as publishing their data in the semantic web world. In the context of the Natural Europe project we have developed methodologies, software architectures and systems to support Natural history museums for their web presence, their interoperability with major international metadata providers and search engines and for publishing their data as Linked Data ([10], [16]). The systems have been installed and used in six important European Natural History museums.

Each museum node is provided with a Multimedia Authoring Tool (MMAT), a Cultural heritage Object (CHO) repository and with a Vocabulary Server facilitating the complete metadata management lifecycle ingestion, maintenance, curation and dissemination of CHOs. The infrastructure also supports the migration of legacy metadata migration into the node. The application profile of CHOs has been created through an iterative process with the museum domain experts and it is a superset of the Europeana Semantic Elements (ESE) metadata format, thus providing a direct interoperability with the central European Digital library (Europeana). The CHO repository manages both content and metadata and adopts the OAIS Reference Model for ingestion, maintenance and dissemination of information packages. The Vocabulary Server supports any taxonomic classification that the museum may use. The ingested taxonomies follow the SKOS format which is a leading international standard based on the Semantic Web principles for representations of Thesauri, Taxonomies and other types of controlled vocabularies. The controlled vocabularies provide strong support for the curation, indexing, retrieval, autocomplete functionality, etc.

For Natural history an important vocabulary is is the Catalogue of life (CoL) which contains 1.4 millions of species and their relationships. We have expressed the taxonomy of CoL to SKOS using the CoL annual checklist and a D2R server. An Access Module provides a number of services that allow selective harvesting of metadata from external entities through an OAI-PMH interface. The museums can be seen individually or through a federation. The Access Module is used to harvest the metadata to the federal node, to Europeana, as well as to establishing connections with major biodiversity networks such as GBIF and BIOCASE. The BIOCASE network is based on a very involved Schema (ABCD Schema) which describes nearly 1200 different concepts. We developed in cooperation with museum experts mappings between the ABCD schema concepts, and wrappers to be used by BIOCASE to access the XML databases of natural Europe (the BIOCASE wrappers assume relational dbms underneath). The wrappers developed follow a layered architecture so that they can be easily adapted for other XML data sources.

To support the Semantic Web presence of each museum individually we have described in OWL the CHO Application profile of Natural Europe. The resulting Natural Europe ontology references well known ontologies/schemas (like DC, FOAF, Geonames, SKOS) and has been aligned with the Europeana Data model (EDM) supporting interoperability with the Europeana Seman-

tic Layer. The publication process involves establishing links to the external RDF data sets, conversion of the XML data to RDF, maintenance, publishing and dissemination of RDF data. The Semantic Infrastructure allows highly expressive queries combining knowledge from distributed resources like 'find photos of endangered species of genus "Bufo" in neighboring countries of Greece' which combines information from Natural Europe, DBpedia, CoL/Uniprot, and Geonames.

Management of Mobile Multimedia Nature Observations using Crowd Sourcing. Several scientific fields (biodiversity, biology, agriculture, etc.) would greatly benefit if informed users with interest in the domain could contribute with their observations to the knowledge in the domain. This need arises from the fact that the number of scientists and the available funding in certain domains are very limited with respect to the real needs, We are developing a Software Framework [21] that supports communities with common interests in nature to capture and share multimedia observations of nature objects or events using mobile devices. The observations are automatically associated with contextual objects (such as GPS objects, pictures, sensor data) and they can be visualized in a faceted manner on top of 2D or 3D maps. The observations are managed by a multimedia management system, and annotated by the same and/or other users with common interests. Multimedia observations of nature objects or events can be annotated by multimedia annotations that are complex resources. Annotations made by the crowd support the knowledge distillation and data provenance.

Collaborative Environments for Instructional Design. We are investigating the design of collaborative environments that allow instructional designers and educators to develop educational templates and scenarios that can be used in different educational contexts. We have developed such a tool, Octopus [11], in the context of EU projects. Octopus is compatible with IMS LD Level A, while hiding its complexity from its user interfaces. It supports a wide range of collaboration and interoperability features, and extensive usability tests have been used to improve its interfaces.

5. ADDITIONAL AUTHORS

Additional authors: N. Giatrakos, N. Gioldasis, P. Arapi, N. Moumoutzis, K. Stravoskoufos, A. Preventis, C. Tsinaraki, K. Makris, G. Skevakis, M. Mylonakis, I. Trohatou, V. Kalokyri, and V. Vazaios.

6. REFERENCES

[1] N. Bikakis, C. Tsinaraki, N. Gioldasis, I. Stavrakantonakis, and S. Christodoulakis. The XML and Semantic Web Worlds: Technologies, Interoperability and Integration: A Survey of the State of the Art. In *Semantic Hyper/Multimedia Adaptation: Schemes and Applications*. 2012.

[2] N. Bikakis, C. Tsinaraki, I. Stavrakantonakis, N. Gioldasis, and S. Christodoulakis. The SPARQL2XQuery Interoperability Framework. *WWW Journal*, 2013.

[3] S. Burdakis and A. Deligiannakis. Detecting outliers in sensor networks using the geometric approach. In *IEEE ICDE*, 2012.

[4] G. Cormode, M. Garofalakis, P. J. Haas, and C. M. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3), 2012.

[5] M. Garofalakis, D. Keren, and V. Samoladas. Sketch-based geometric monitoring of distributed stream queries. *PVLDB*, 6(10), 2013.

[6] N. Giatrakos, A. Deligiannakis, M. N. Garofalakis, I. Sharfman, and A. Schuster. Prediction-based geometric monitoring over distributed data streams. In *ACM SIGMOD*, 2012.

[7] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, and A. Deligiannakis. Geometric monitoring of heterogeneous streams. *IEEE TKDE*, 2013.

[8] D. Keren, I. Sharfman, A. Schuster, and A. Livne. Shape sensitive geometric monitoring. *IEEE TKDE*, 24(8), 2012.

[9] K. Makris, N. Bikakis, N. Gioldasis, and S. Christodoulakis. SPARQL-RW: Transparent Query Access over Mapped RDF Data Sources. In *EDBT*, 2012.

[10] K. Makris, G. Skevakis, V. Kalokyri, P. Arapi, and S. Christodoulakis. Metadata Management and Interoperability Support for Natural History Museums. In *TPDL*, 2013.

[11] M. Mylonakis, P. Arapi, N. Moumoutzis, S. Christodoulakis, and M. Ampatzaki. Octopus: A Collaborative Environment Supporting the Development of Effective Instructional Design. In *ICEEE*, 2013.

[12] S. Papadopoulos, G. Cormode, A. Deligiannakis, and M. Garofalakis. Lightweight authentication of linear algebraic queries on data streams. In *ACM SIGMOD*, 2013.

[13] O. Papapetrou and M. Garofalakis. Continuous fragmented skylines over distributed streams. In *IEEE ICDE*, 2014.

[14] O. Papapetrou, M. N. Garofalakis, and A. Deligiannakis. Sketch-based querying of distributed sliding-window data streams. *PVLDB*, 5(10), 2012.

[15] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM TODS*, 32(4), 2007.

[16] G. Skevakis, K. Makris, P. Arapi, and S. Christodoulakis. Elevating Natural History Museums' Cultural Collections to the Linked Data Cloud. In *SDA*, 2013.

[17] S. Sotiriadis, G. Petrakis, S. Covaci, P. Zampognaro, E. Georga, and C. Thuemmler. An architecture for designing Future Internet (FI) applications in sensitive domains: Expressing the Software to data paradigm by utilizing hybrid cloud technology. In *IEEE BIBE*, 2013.

[18] C. A. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4), 2012.

[19] C. Thuemmler, J. Mueller, S. Covaci, T. Magedanz, S. de Panfilis, T. Jell, A. Schneider, and A. Gavras. Applying the Software-to-Data Paradigm in Next Generation E-Health Hybrid Clouds. In *ITNG*, 2013.

[20] C. Tsinaraki and S. Christodoulakis. Domain Knowledge Representation in Semantic MPEG-7 Descriptions. In *The Handbook of MPEG applications: Standards in Practice*.

[21] C. Tsinaraki, G. Skevakis, I. Trochatou, and S. Christodoulakis. MoM-NOCS: Management of Mobile Multimedia Nature Observations using Crowd Sourcing. In *MoMM*, 2013.

[22] D. Z. Wang, M. J. Franklin, M. N. Garofalakis, and J. M. Hellerstein. Querying probabilistic information extraction. *PVLDB*, 3(1), 2010.

[23] D. Z. Wang, M. J. Franklin, M. N. Garofalakis, J. M. Hellerstein, and M. L. Wick. Hybrid in-database inference for declarative information extraction. In *ACM SIGMOD*, 2011.

[24] D. Z. Wang, E. Michelakis, M. J. Franklin, M. N. Garofalakis, and J. M. Hellerstein. Probabilistic declarative information extraction. In *IEEE ICDE*, 2010.

Report on the First International Workshop on Cloud Intelligence (Cloud-I 2012)

Jérôme Darmont
Université de Lyon (ERIC Lyon 2)
5 avenue Pierre Mendès-France
F-69676 Bron Cedex – France
jerome.darmont@univ-lyon2.fr

Torben Bach Pedersen
Aalborg University (Daisy)
Selma Lagerløfs Vej 300
DK-9220 Aalborg Ø – Denmark
tbp@cs.aau.dk

1. INTRODUCTION

Business intelligence (BI) is a broad field related to integrating, storing and analyzing data to help decision-makers in many domains (from business to administration, health, and environment) make better decisions using analytics methods include reporting, on-line analytical processing (OLAP), and data mining.

With the increasing success of cloud computing, cloud business intelligence "as a service" offerings have arisen, both from cloud start-ups and major BI industry vendors. Beyond porting BI features into the cloud, which already implies numerous issues (e.g., BigData/NoSQL database modeling and storage, data localization, data marketplaces, security and privacy, performance, cost and usage models...), this trend also poses new, broader challenges for making data analytics available to small and medium-size enterprises (SMEs), non-governmental organizations, web communities (e.g., supported by social networks), and even the average citizen. This vision requires new integration and deployment models. For example, some deployments would benefit from an integrated database of private and open data.

Thus, Cloud Intelligence is not only a current technological and research challenge, but also an important societal stake, since people increasingly demand open data (e.g., the Spanish *indignados*), which they possibly mix with private data, and analyze with tools with advanced collaborative features, enabling users to share and re-use business intelligence concepts and analysis results world-wide.

The First International Workshop on Cloud Intelligence (Cloud-I 2012) [1] was held in conjunction with VLDB 2012 in Istanbul, Turkey on August 31, with the aim of becoming an interdisciplinary, regular exchange forum for researchers, industry and practitioners, as well as all potential users of Cloud Intelligence. This full-day event brought together researchers and engineers from academia and industry to discuss and exchange ideas related to BI and the cloud. The workshop featured one industrial keynote, three research sessions, and a panel.

The topics of the accepted papers spanned a number of exciting topics within Cloud Intelligence, including (in no particular order) RDF triple stores for the cloud, secure and private data sharing and analytics outsourcing in the cloud, MapReduce-based computations, and domain-specific cloud-based BI solutions.

2. INDUSTRIAL KEYNOTE

The keynote entitled "Analytic Lessons: in the Cloud, about the Cloud" was given by Dr. Morten Middelfart, CTO of TARGIT, Europe's largest pure-play developer of business intelligence products and a top 15 international vendor. With a single quote in mind: "The journey to courageous leadership, where organizations compete at a new level is: eliminate fear and trust computing as partner in a high-performance team.", Dr. Middelfart shared his experience about designing two different approaches to cloud-based deployment of analytics and business intelligence, namely an analyst specialist platform and a social platform. The analyst specialist platform helps model and share data, and has proven particularly useful in the analysis of large amounts of streaming unstructured data; aka Big Data. In the social networking approach, users can friend, share, analyze and discuss datasets. So far, the analyst platform has been the most popular. However, Dr. Middelfart finally discussed the trending behavior of the social platform and its current and potentially game-changing impact on the industry, as analytics shifts from being inside-out to embracing entire industries from the outside and in.

3. RESEARCH PAPERS

3.1 Session 1: Data and Knowledge Management

The position paper entitled "Towards a Hybrid Row-Column Database for a Cloud-based Medical Data Management System", by Baraa Mohamad, Laurent d'Orazio and Le Gruenwald, pinpoints the challenges in integrating high-volume, heterogeneous medical data in the form of DICOM files in the cloud. A novel hybrid "row-column", two-level database architecture is pro-

posed, where mandatory/frequently used attributes and attributes frequently accessed together are stored in a row-oriented database, and optional/private attributes are stored in a column-oriented database. This architecture is easy to use, extensible, efficient and allows ad-hoc queries over DICOM files, while benefiting from the elasticity, billing by use, and scalability of the cloud.

Yasin Silva, Jason Reed and Lisa Tsosie, in "MapReduce-based Similarity Join for Metric Spaces", study cloud-based similarity joins (a sparsely studied issue up till now). They propose a MapReduce-based algorithm called MR-SimJoin that efficiently partitions and distributes the data until the subsets are small enough to be processed in a single node. MR-SimJoin is general enough to be used with data that lies in any metric space, thus it can be used with multiple data types and distance functions. It is implemented in Hadoop and has good execution time and scalability properties.

Roshan Punnoose, Adina Crainiceanu and David Rapp propose "Rya: A Scalable RDF Triple Store For The Clouds". This scalable RDF data management system uses Accumulo, a Google Bigtable variant. Storage methods, indexing schemes and query processing techniques allow to scale to billions of triples across multiple nodes, while providing fast and easy access to the data through conventional query mechanisms such as SPARQL. Performance evaluations show that Rya outperforms existing distributed RDF solutions in most cases.

3.2 Session 2: Data Analytics

The position paper entitled "Integrity Verification of Cloud-hosted Data Analytics Computations", by Wendy Wang, introduces efficient and practical integrity verification techniques that check whether an untrusted cloud returns correct results of outsourced data analytics computations including a large class of machine learning and data mining methods. Verication techniques work for both non-collusive and collusive malicious workers in MapReduce.

Thanh Binh Nguyen, Fabian Wagner and Wolfgang Schöpp, in "Cloud Business Intelligent Services of well-established modeling tools to explore the synergies and interactions among climate change, air quality objectives", design a Cloud-based Business Intelligent Application Framework that includes a set of services grouped into Data warehousing Services and Business Intelligent Services. The former are used to specify the GAINS (Greenhouse Gas – Air Pollution Interactions and Synergies) data warehouse, while the latter help publish key data of scientific analysis in a transparent manner.

In their position paper "On Saying "Enough Already!" in MapReduce", Christos Doulkeridis and Kjetil Nørvåg criticize the brute force approach of MapReduce, which leads to performing redundant work, especially in the case of top-k queries. Different techniques that allow the efficient processing of top-k queries without exhaustively accessing input data are investigated. Various individual approaches and combinations of such approaches are proposed to provide the first steps towards integrating efficient top-k processing in MapReduce.

3.3 Session 3: Security and Privacy

Bharath Samanthula, Gerry Howser, Yousef Elmehdwi and Sanjay Madria, in the paper entitled "An Efficient and Secure Data Sharing Framework using Homomorphic Encryption in the Cloud", propose an efficient and Secure Data Sharing (SDS) framework using homomorphic encryption and proxy re-encryption schemes that prevents the leakage of unauthorized data when a revoked user rejoins the system. This framework is generic and secure under the security definition of Secure Multi-Party Computation (SMC). Any additive homomorphic encryption and proxy re-encryption scheme can be used. In addition, the underlying Secure Data Sharing (SDS) framework features a new solution based on data distribution to prevent information leakage in the case of collusion between users and Cloud Service Providers.

Mehdi Bentounsi, Salima Benbernou, Mikhail Atallah and Cheikh Deme present "Anonyfrag: An Anonymization-Based Approach For Privacy-Preserving BPaaS", which is an anonymization-based approach to preserve the client business activity while sharing process fragments between organizations on the cloud, i.e., when using on demand applications as Business Process as a Service through multi-tenant cloud platforms.

4. PANEL

Finally, Jérôme Darmont, Torben Bach Pedersen and Morten Middelfart launched a panel discussion themed "Cloud Intelligence: What is REALLY New?" to sort out what is new and not so new in cloud business intelligence as a service.

Torben Bach Pedersen defined three new things in cloud intelligence: elasticity, including the ability to bring in new data sources; a bottom-up, user-driven approach (in opposition to a top-down, enterprise-driven approach); and the fundamentally new economic model needed for cloud intelligence (pay-as-you-go instead of large prior investment).

Jérôme Darmont stressed out that security issues were even more critical in the cloud. Although some of these issues are inherited from classical distributed architecture, some directly relate to the new framework of the cloud, with privacy being of premium importance. Moreover, the social aspect of cloud intelligence involves sharing analysis results without necessarily disclosing source data.

Morten Middelfart eventually discussed the challenges about interpretation, bias, and completeness of external data gathered from the Web. Cloud intelligence presents an entirely new era of analytically founded strategic thinking, but on the other hand, it elevates the need for user understanding of the "truth behind the chart". A rich discussion ensued with the audience, the conclusions of which are included in the next section.

5. DISCUSSION AND OUTLOOK

The lessons that can be drawn from the workshop fall along several directions. First, when comparing the wide range of themes within cloud intelligence, e.g., as outlined in the call for papers and the panel discussions,

with the papers that actually appeared in the workshop, it is clear that the presented papers mainly focus on rather specific, mostly technical, issues. These are more precisely data management architectures and systems for cloud platforms, MapReduce-based algorithms for specific problems, and issues related to privacy, security, and integrity in the more technical sense. As a side note, the non-accepted papers also mainly fell in these areas. The only outlier to this pattern is the paper on cloud business intelligent services that mainly focus on the new user-oriented functionality enabled by cloud deployment. These issues were also covered in some of the panel statements.

Second, we can look at for which topics no papers appeared. One such issue is elasticity in the wider sense of the word. Another important "missing" set of topics relates to the social aspects of cloud intelligence, e.g., sharing results and new collaborative bottom-up approaches for building BI systems in the cloud. This leads to a demand for exploring new ways of using analytics enabled by the new opportunities in the cloud. However, these opportunities will only be used if the delivered results are backed up by work on truth and trust in the more intuitive sense of the word. Finally, new economic models for pay-as-you-go cloud intelligence will have to be developed. A long discussion on this topic concluded that typical web economic models like online ads and app stores were not well suited for this scenario. Micro-payment models had a better fit, but conflicted with the need for enterprises to above all have *predictable* costs.

We can thus conclude that there is a large demand for further research within cloud intelligence. As a first facilitating activity, the two best papers have been invited to submit extended versions to a special issue/section of *Information Systems* which also has an *open* call for papers[1]. We thus encourage the readers of SIGMOD Record to submit papers on cloud intelligence topics. Next, we hope that the workshop is just the first in a hopefully long series, and we certainly hope to hold the workshop again in 2013 and beyond.

6. ACKNOWLEDGEMENTS

The Cloud-I Chairs would like to thank all the authors of submitted papers for their interest in the workshop and the high quality of the submitted papers. We would also like to thank all the referees (both PC members and external reviewers) for their careful and dedicated work, both during the reviewing and the discussion phases. Working in cooperation with this Program Committee has been both an honor and a pleasure. Finally, we would like to express our gratitude to the members of the Organizing Committee of VLDB 2012, especially the Workshop Chairs James Joshi, Hakan Ferhatosmanoglu, and Andreas Wombacher, for their support in organizing this workshop.

[1] http://eric.univ-lyon2.fr/cloud-i/?p=269

7. REFERENCES

[1] J. Darmont and T. B. Pedersen, editors. *1st International Workshop on Cloud Intelligence (colocated with VLDB 2012), Cloud-I '12, Istanbul, Turkey, August 31, 2012*. ACM, 2012. http://dl.acm.org/citation.cfm?id=2347673.

Report on the Second International Workshop on Energy Data Management (EnDM 2013)

Torben Bach Pedersen
Center for Data-intensive Systems (Daisy)
Aalborg University
9220 Aalborg, Denmark
tbp@cs.aau.dk

Wolfgang Lehner
Database Technology Group
Technische Universität Dresden
01062 Dresden, Germany
{wolfgang.lehner}@tu-dresden.de

1. INTRODUCTION

The energy sector is in transition–being forced to rethink the current practice and apply data-management based IT solutions to provide a scalable and sustainable supply and distribution of energy. Novel challenges range from renewable energy production over energy distribution and monitoring to controlling and moving energy consumption. Huge amounts of "Big Energy Data," i.e., data from smart meters, new renewable energy sources (RES–such as wind, solar, hydro, thermal, etc), novel distributions mechanisms (Smart Grid), and novel types of consumers and devices, e.g., electric cars, are being collected and must be managed and analyzed to yield their potential.

Energy is at the top of the worldwide political agenda. For example, The European Union has stated the "20-20-20 goals" (20% renewable energy, 20% better energy efficiency, and 20% CO2 reduction by 2020). Even more ambitious goals are set for 2030 and 2050. This situation is reflected in research funding schemes such as the EU Horizon 2020 Framework program as well as national programs. Increasingly, such programs include joint calls involving both energy and IT partners. Data management is at the heart of this development, as witnessed by the following story headlines from key players: "The Smart Grid Data Deluge" (O'Reilly Radar); "Big data for the Smart Grid" (theenergycollective); "The Coming Smart Grid Data Surge" (SmartGridNews.com).

Thus, data management within the energy domain becomes increasingly important. The International Workshop on Energy Data Management (EnDM) focuses on conceptual and system architecture issues related to the management of very large-scale data sets specifically in the context of the energy domain. The overall goal of the EmDM workshop is a) to bridge the gap between domain experts and data management scientists and b) to create awareness of this emerging and very challenging application area. For the workshop's research program, the organizers especially try to attract contributions that push the envelope towards novel schemes for large-scale data processing with special focus on energy data management.

The Second International Workshop on Energy Data Management (EnDM'13)[1] was held in conjunction with EDBT 2013 in Genova, Italy, on March 22, 2013. This half-day event brought together researchers and engineers from academia and industry to discuss and exchange ideas related to energy data management and related topics. The workshop featured one industrial keynote, five research papers, and finished off with a panel/roundtable discussion. The accepted papers spanned a number of exciting topics within energy data management, including (in no particular order) representation of smart meter data, ontologies for emissions trading, and forecasting of renewable energy production. Two papers concerned the important topic of capturing and managing flexible energy demands, specifically the visualization of flexible energy demand objects and the extraction of consumption flexibilities from consumer consumption time series. The workshop proceedings have been published in a joint volume of all EDBT/ICDT 2013 workshops [1].

2. INDUSTRIAL KEYNOTE

The keynote was given by Data Warehouse Architect Jens Otto Sørensen from the Danish Transmission System Operator (TSO), Energinet.dk, and was entitled "The Danish DataHub Solution." The keynote first described the un-bundling and liberalization which has taken place in the Danish electricity market over the past decade. This process has led to a number of new problems, including the lack of separation between grid companies and electricity suppliers, competitive market barriers, (too) varying quality of readings and master data, no overview of errors and delays in transactions and data exchange, and the inability of lack of (corporate) customers to get sufficient overview of their electricity consumption. These problems led to new market regulations in order to solve them. However, in order to implement these regulations there was a need for a common place to exchange detailed energy data among all the market players. This so-called "DataHub" has now been implemented and entered into production[2]. The talk explained the benefits of the solutions, including a generally improved data management, technocratic un-bundling, lowering market entry barriers, improved efficiency, facilitating new products and services, and better market integration. The talk also outlines the desirable properties for such a solution, includ-

[1]http://endm2013.endm.org

[2]DataHub web page

ing (market) transparency, seamlessness, real-time operation, decoupling of (business) processes, full "transaction time flexibility" (rolling back committed transactions, processing transactions in the past and the future), and low technical entry barriers. As of March 2013, the system processes around 2 million inbound and 2 million outbound transactions per day, contained in around 800,000 messages, the largest of which was 44MB(!). The keynote summarized the lessons learned, which included the never-ending importance of data quality, the non-trivial involvement of IT vendors, and the significant technical and organizational challenges of communicating with 130+ organizations using 25+ IT systems. However, an even bigger challenge was to get everyone involved to understand the new business requirements (new timelines and regulations). Finally, the keynote discussed the taken design decisions and concluded that would have been different if only technical, and not also political, considerations had to be taken. For example, the political requirements meant that EDIFACT (in addition to XML) formats were allowed and the master data ownership was distributed to 78 local grid companies rather than a single central authority.

3. RESEARCH PAPERS

The first paper "Symbolic Representation of Smart Meter Data" by Tri Kurniawan Wijaya, Julien Eberle, and Karl Aberer focused on the topic of smart meter data analytics, which allows utility companies to analyze of smart meter data in real-time to understand customer behavior. However, the data volumes are very large, leading to performance problems, and detailed meter data is furthermore a potential privacy breach. Thus, the paper instead proposes to generalize the detailed readings into symbolic units that reduces both the volume and privacy risks significantly, while still allowing interesting analyses, e.g., data mining, to be performed on the symbolic data. A number of experiments on real-world data showed the feasibility of this very interesting proposal.

The next paper "Visualizing Complex Energy Planning Objects With Inherent Flexibilities" by Laurynas Siksnys and Dalia Kaulakiene focused on visualization of smart grid data. Specifically, it considered visualizing objects capturing the inherent flexibilities in (intended) electricity consumption and production, so-called "flex-offers." The paper first presented the planning and control activities involved in balancing demand and supply, which are made harder by increasing rates of (non-schedulable) renewable energy, faced by current energy companies. The paper then presented its OLAP-inspired approach to navigate and explore flex-offers, including several specific visualizations and a histogram-based technique for the visualization. The paper finished by outlining the research challenges ahead in visualizing energy flexibilities.

The paper by Umberto Ciorba, Antonio De Nicola, Stefano La Malfa, Tiziano Pignatelli, Vittorio Rosato, and Maria Luisa Villani called "Towards Ontological Foundations of Knowledge related to the Emissions Trading System" discussed the European Union's Emissions Trading System (EST). It first analyzed some of the EST-related challenges that can be handled by ICT systems. A significant challenge in this area is the need for a precise understanding of the area, in the form of a common and formalized model, i.e., an ontology of the area, for which the paper presented the first step. The paper then discussed the ontological foundations for the development of ontologies related to the EST, a first example and a vision for practical implementation, and the associated challenges encountered with their development.

The fourth paper "Optimized Renewable Energy Forecasting in Local Distribution Networks" by Robert Ulbricht, Ulrike Fischer, Wolfgang Lehner, and Hilko Donker considered the role of forecasting in integrating renewable energy sources (RES) into local energy distribution networks. Since RES are not controllable, it is essential to be able to accurately forecast the supply delivered by RES. However, a number of challenges exist, including the wide variety of RES installations, and the non-availability of fine-grained metering data. The paper presents a generalized optimization approach for determining the best forecasting strategy for a given scenario, including the choice of forecasting model, forecasting granularity (single RES installation or aggregated view), and model parameters. The approach is tested on real-world data and directions for future research are given.

The final paper by Dalia Kaulakiene, Laurynas Siksnys and Yoann Pitarch was called "Towards the Automated Extraction of Flexibilities from Electricity Time Series." Like the second paper, it also concerned the topic of flexibilities in energy consumption and demand, but from a different perspective. Specifically, the paper considered how to derive the available flexibility in the energy consumption of a given customer based only on metering data. The paper presented a number of approaches, ranging from basic to advanced, and requiring various amounts of background knowledge, e.g., knowledge of appliances or usage frequencies. Some of the approaches have been implemented in a software tool used in the simulation trials of the EU project MIRABEL. The paper rounded off by presenting a number of directions for future research.

4. ROUNDTABLE/PANEL

The workshop finished off with a panel/roundtable discussion on Research Challenges for Energy Data Management. The workshop organizers first suggested some important topics. First, there is currently a lack of common definitions of data and information concepts within the area, e.g., community-wide agreed-upon standard ontologies specifying common concepts. Second, there is a lack of standardization of the units of the technical architecture within smart grid systems, e.g., which types of layers exist, and what the nodes at each layer does. Further challenges include optimized forecasting and prediction techniques, seamless integration of past, present and future data, and developing scalable and robust data management techniques tailor-made for en-

ergy data management systems. In this context, the domain of energy data management is a driving force to build robust solutions combining data-intensive applications (classical analytical workloads on large datasets) and compute-intensive applications (simulations, numerical optimizations etc).

In more general terms, energy data management systems are a prime example of massively distributed systems managing large amounts of data in real-time while operating vital societal infrastructures. Thus, techniques developed within energy data management will have further applications in other demanding application domains. This impact will embrace a variety of different areas in database and information systems research. For example, on the one side, domain-specific modeling techniques can be adapted to suit other application areas as well. On the other side, optimizations at the system architecture layer are required to deal with massive amounts of time series data and allow flexible aggregation and sampling techniques. Since time series are relevant for many other domains as well, the technological impact sparked by energy data management will help to push the envelope of sophisticated data management techniques in general. In the long term, we also consider the domain of energy data management as one of the most prominent use-case of cyber-physical systems (CPS, see cyberphysicalsystems.org/) to seamlessly combine activities within the real and virtual world by an omnipresent monitoring and activity triggering mediation layer.

The roundtable discussion added further perspectives. It was mentioned that electricity consumers will change their behavior if the incentives are right, e.g., in a case from Florida, consumers changed their use significantly in return for less blackouts. In general, financial incentives is not enough, one must also look into "earthsaver points" and friendly competition with peers and neighbors. Another upcoming issue is charging electric vehicles (EVs), which can in some areas at some times exceed the available capacity. Thus, intelligent approaches for handling such flexible demand are needed. Here, a lot can be gained from analyzing and understanding user behavior, all the way down to the person and device levels. However, there is a lack of good datasets for this, also due to privacy concerns. However, open datasets would be a significant asset in this area.

5. DISCUSSION AND OUTLOOK

Summing up, if we first look at the topics of the presented papers, we note that they span a wide range of topics ranging from smart meter data representation and use, ontologies for emissions trading, forecasting of renewable energy production, and managing flexible energy demand. Compared to the first workshop, the important issue of privacy of energy data was now addressed. The papers are generally the result of interdisciplinary collaborations, including contributions from several areas within computer science.

Next, when looking at the topics which occurred in the Call for Papers, but not within the accepted (or submitted) papers, we see that more systems-oriented topics such as data processing architectures, partitioning, caching, and replication schemes, query languages and query processing, robustness aspects are not covered. We believe this is not because the topics are not important, but rather due to the fact that energy data management is still new, and thus most systems are still in the development phase. While most papers are based on small case studies, only the keynote described large industrial case studies of already running systems. We again attribute this to the fact that smart grids are still in development.

Summing up, we conclude that there is a lot of interesting work going on in the area of energy data management, with many remaining challenges to be met. This supports the need for venues that focus on this issue. The EnDM workshop series will continue at EDBT 2014 in Athens where the 3nd International Workshop on Energy Data Management will be held on March 28, 2014[3]. For the 3rd edition of the workshop, it is the intention to organize a special issue of a journal for extended versions of the best papers.

6. ACKNOWLEDGEMENTS

The EnDM Chairs would like to thank all the authors of submitted papers for their interest in the workshop and the high quality of the submitted papers. We would also like to thank the distinguished PC members for their careful and dedicated work, both during the reviewing and the discussion phases. Finally, we would like to thank the Organizing Committee of EBDT/ICDT 2013, especially the General Chair Giovanna Guerrini, the Workshop Chair Kai-Uwe Sattler, and the Proceedings Chair Anastasios Gounaris. for their support in organizing the EnDM 2013 workshop.

7. REFERENCES

[1] G. Guerrini, editor. *Joint 2013 EDBT/ICDT Conferences, EDBT/ICDT '13, Genoa, Italy, March 22, 2013, Workshop Proceedings*. ACM, 2013.

[3]http://endm2014.endm.org

Report from the second workshop on Scalable Workflow Enactment Engines and Technology (SWEET'13)

Jacek Sroka
University of Warsaw, Poland
j.sroka@mimuw.edu.pl

Jan Hidders
TUDelft, The Netherlands
a.j.h.hidders@tudelft.nl

Paolo Missier
Newcastle University, UK
paolo.missier@ncl.ac.uk

ABSTRACT

This report summarizes the Second International Workshop on Scalable Workflow Enactment Engines and Technologies (SWEET'13). This workshop was held in conjunction with the 2013 SIGMOD conference in New York, NY, USA on June 23th, 2013. The goal of the workshop was to bring together researchers and practitioners to explore the state of the art in workflow-based programming for data-intensive applications, and the potential of cloud-based computing in this area. The program featured 4 paper presentations and two very well attended invited talks by Prof. Paul Watson, Newcastle University, UK and Dr Jelena Pjesivac-Grbovic from Google, Inc.

1. INTRODUCTION

The SWEET workshop is aimed at exploring the cross-over between languages and models for parallel data processing, and traditional workflow technology, primarily on a cloud infrastructure and for data-intensive applications. The next generation of these systems is increasingly capable of dealing with changing circumstances. Rather than efficiently running off-line a specific workflow on a predictable cloud-based data processing back-end, they now have to deal with dynamic behavior such as real-time data-analysis during the execution, user-interference with the computation while executing and changes in the computational efficiency or network structure of the heterogeneous execution back-end. These developments were reflected in the contributions of this edition of the workshop. For example, the first paper presents the *DynamicCloudSim* system for simulating the effects of certain resource allocation and scheduling strategies in dynamic cloud-based distributed architectures where the efficiency of the different services may change in time. The second paper presents STAFiLOS, a STreAm FLOw Scheduler, which allows the stream-based execution of a workflow with dynamic input streams on top of a conventional workflow execution engine. The third paper introduces OSIRIS-SR, a distributed peer-to-peer workflow execution framework that allows workflows to be efficiently and reliably executed on a dynamic networks of cooperating nodes. The final paper gives an overview of user-steering in HPC workflows, where users can dynamically interact with the execution of a workflow for purposes such as analysis and debugging.

Next to the presented papers, the workshop featured two invited talks: the first by prof. Paul Watson from Newcastle University, UK on *Realizing the Potential of the Cloud for Workflow: Scalability, Security and Reproducibility*, and the second talk by Jelena Pjesivac-Grbovic from Google on *The Google Cloud Platform* and giving an overview of the various distributed data processing frameworks offered and developed by Google.

Details of the papers, keynotes and tutorials are available on the workshop web-site[1], and the proceedings are published on the ACM DL [1]. The rest of the report provides a summary of the contributions, and is structured along the distinction in scope and purpose introduced above.

2. PAPER PRESENTATIONS

DynamicCloudSim: Simulating Heterogeneity in Computational Clouds

In this paper from the Humboldt-University in Berlin, Marc Nicolas Bux, on behalf of Ulf Leser, presented the *DynamicCloudSim* system. It extends the popular framework CloudSim [3] for sim-

[1] http://sites.google.com/site/sweetworkshop2013

ulating resource provisioning and scheduling algorithms on cloud computing infrastructure with the aspect of instabilities that are common to shared cloud infrastructures. There are several factors of instability that are taken into account. The first is inhomogeneity in the performance of computational resources observed and measured for example by Dejun et al. [6], Jackson et al. [8] and Schad et al. [12]. A second factor is uncertainty in, and dynamic changes to, the performance of VMs due to sharing of common resources with other VMs and users, as for instance reported by Dejun et al. [6]. A third and final factor are straggler VMs [16] and failures during task execution for which programmers need to be prepared for, especially in massively parallel applications [13]. Accounting for those types of instabilities makes simulations of cloud applications more reliable which is important for cost planning. It is also the first step for evaluating novel approaches towards resource allocation and task scheduling on distributed architectures. DynamicCloudSim has been tested on scientific workflow scenarios, yet it is still to be verified against traces of workflow execution on actual cloud infrastructure.

A Continuous Workflow Scheduling Framework

Panayiotis Neophytou from the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA, on behalf of Panos Chrysanthis and Alexandros Labrinidis argued that both scientific and business WfMS can be extended to support data stream semantics to enable monitoring applications. For this goal the authors designed and implemented an integrated workflow scheduling framework STAFiLOS, a STreAm FLOw Scheduler, a Continuous Workflows framework within their CONFLuEnCE [10] engine built on top of the Kepler system [9]. STAFiLOS supports the implementation of different scheduling policies. It was evaluated based on the Linear Road Benchmark [2] — the standard benchmark for stream processing system — and compared against Kepler's own Thread-Based director.

OSIRIS-SR – A Scalable yet Reliable Distributed Workflow Execution Engine

Nenad Stojnic from the Department of Mathematics and Computer Science, University of Basel, Switzerland, presented this paper on behalf of the second author Heiko Schuldt. It introduces OSIRIS-SR (Open Service Infrastructure for Reliable and Integrated process Support – Safety Ring) a true peer-to-peer workflow execution engine, which is an extension of the OSIRIS system [14, 15]. In contrast to other workflow engines, here the workflow orchestration itself is distributed across a set of cooperating nodes. This is done by means of *mini workflow engines* on each node, which together form the *OSIRIS-SR layer*. This results in higher scalability and reliability. To protect against network or node failures OSIRIS-SR uses a scalable self-organizing and self-healing *node monitor* overlay, called the Safety Ring. Its members supervise the non-member nodes currently in charge of service invocation and also provide a scalable and reliable metadata storage. The presented evaluation results show that the Safety Ring-based failure handling and transactional migration at instance level comes with a only minor and affordable impact on the overall performance.

User-Steering on HPC Workflows: State of the Art and Future Directions

Daniel de Oliveira from Fluminense Federal University, Niterói, Brazil summarized the state-of-the-art and the main challenges in supporting user-steering in HPC workflows. The other authors are Marta Mattoso, Jonas Dias, Kary Ocaña, Flavio Costa, Felipe Horta, Vítor Sousa and Igor Araújo from COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil and Eduardo Ogasawara from Federal Center for Technological Education, Rio de Janeiro, Brazil. User-steering of workflows refers here to the run-time interference of users with the execution of a workflow. This can mean for example stopping the workflow, analysing intermediate results, changing parameters or even the structure of the workflow, and finally letting the execution continue. User-steering is a big step towards more dynamic workflows and fully supporting the exploratory nature of Science and the dynamic process involved in scientific analysis [7]. Based on the motivation from domains of Bioinformatics and the Gas & Oil domain and based on the previous experience such as [11] three main issues were formulated and discussed: (i) monitoring of execution, (ii) data analysis at runtime, and (iii) dynamic interference in the execution. This division guided the discussion of the state-of-the-art in workflow steering. The conclusion was that in (i) the desired features are already tackled by existing support for querying provenance at runtime [11, 5] and scientific event notification [4]. For (ii) several open challenges were discussed as data staging, big data *in situ* analysis, decision-support tools, dynamic workflow engines, parameter slice exploration and experiment optimization.

3. KEYNOTES

Realizing the Potential of the Cloud for Workflow: Scalability, Security and Reproducibility

The workshop started with a first invited talk by Prof. Paul Watson from Newcastle University, UK who's focus were cloud-based workflow systems. The talk discussed the opportunities offered by cloud computing in overcoming some of the limitations of service-based approaches to workflows enactment. Specifically, three areas were discussed where synergies can be found between workflow technology and cloud computing, namely scalability, security, and reproducibility. Exploiting such synergies, however, requires a radical redesign of the workflow management system, rather than simply a porting of existing implementations to the cloud. An example of such design is e-Science Central, an open-source workflow platform developed by the Information Management group at Newcastle University. This WFMS runs natively on multiple public cloud infrastructures, and is aimed at supporting workflows which are deployed over hundred of cloud nodes and with a running time that is measured in weeks, in areas such as chemical engineering (QSAR) and activity recognition for medical applications.

The Google Cloud Platform

The workshop finished with a second invited talk by Dr Jelena Pjesivac-Grbovic from Google, Inc. on The Google Cloud Platform. Jelena Pjesivac-Grbovic is a senior software engineer in Systems Infrastructure at Google, focusing on distributed data processing frameworks. The Google Cloud Platform is a collection of services offered by Google that allows external users to build their applications and run their computations on top of the Google infrastructure. It consists of the following parts: *App Engine* which allows developers to create apps that are easy to manage and scale, *Cloud Datastore* which offers a schema-less, non-relational datastore with built-in query support, *CloudSQL* which lets developers run MySQL databases in Google Cloud, *Compute Engine* which can run large-scale computing workloads on Linux virtual machines, *Cloud Storage* for storing, accessing and managing data, *BigQuery* for interactive analysis of datasets with billions of rows, *Prediction API* for applying machine learning and finally *Translation API* for automatic translation into other languages.

In this talk Jelena focused on the parts of Google Cloud Platform for executing large-scale data-intensive workflows, which are Google App Engine, Google Compute Engine, Google Cloud Storage and Google Big Query. Two use-cases were presented where these services where used in concert to do large scale data collection and processing. The first was to collect and present a queryable visual interface to show the positions of all the ships in the world. The data would be collected by loggers into Cloud Storage, and the interface was built using Cloud Storage and BigQuery. The second use case was a data sensing lab for collecting and analyzing data during the 2012 Google I/O event by "mote" robots and sensors that monitored participants and environmental parameters such as temperature, pressure, humidity, quality of air, light and RF noise. This resulted in more than 10 GB of data per 20 seconds, which was collected using App Engine and Cloud Datastore, and subsequently analyzed using Cloud Storage and BigQuery with the possible additional use of R and Hadoop. Both presented use cases aimed at showing the practicality and scalability of data-intensive workflows built upon the different presented services of the Google Cloud Platform.

4. CONCLUSION

The presentations and tutorials at SWEET 2013 provided an overview of current developments and emerging issues in the area of dynamic workflow execution by which we mean here the type of workflow execution where during the execution there are changes in the input of the workflow, the specification of the workflow or the distributed backend. These proceedings show that although much has been achieved in this area to make large-scale data-intensive computing more robust and practical, there is still much left for further research. Specifically the following issues and topics were raised and discussed during the workshop:

User-friendliness and Workflow Design Assistance: One of the goals of data-intensive workflow systems is to make big-data computing platforms more usable for non-programmers. However, their user-interfaces are up to now still fairly technical and not giving much assistance with designing effective workflows for certain tasks. The interface could for example recommend certain components or patterns, based on a task description, or it could detect anti-patterns that signal an incorrect or inefficient workflow.

Heterogeneous data-processing workflows: In practice workflows often have to process different types of data from different sources to produce the final result. The data sources may differ in data complexity and retrieval speed, but also in whether

the data is retrieved in big chunks or as a stream of small chunks. At the same time the components in the workflow may also be very different, some may be simple arithmetical operations while others are complex database queries. All this makes it harder to efficiently schedule and execute the workflow, and requires additional research.

Realistic performance models for computational clouds: The efficient scheduling and optimisation of data-intensive workflows, both dynamically and statically, depends to a large extent on having realistic and reliable models for estimating the cost of a schedule or evaluation plan. Research is needed into which types of computational clouds that currently are offered have which types of performance characteristics, and how reliably their behavior can be predicted for the purpose of optimization.

As can be seen from this list, there is no lack of research challenges for the future editions of the SWEET workshop, and interesting papers investigating them are therefore to be expected.

Acknowledgements: We would like to thank the PC members, keynote speakers, authors, local workshop organizers and attendees for making SWEET 2013 a successful workshop. We also express our great appreciation for the support from Google Inc.

5. REFERENCES

[1] *SWEET '13: Proceedings of the 2nd ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*, New York, NY, USA, 2013. ACM.

[2] Arvind Arasu, Mitch Cherniack, Eduardo Galvez, David Maier, Anurag S. Maskey, Esther Ryvkina, Michael Stonebraker, and Richard Tibbetts. Linear road: a stream data management benchmark. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 480–491. VLDB Endowment, 2004.

[3] Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose, and Rajkumar Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *CoRR*, abs/0903.2525, 2009.

[4] Flavio Costa, Vítor Silva, Daniel de Oliveira, Kary Ocaña, Eduardo Ogasawara, Jonas Dias, and Marta Mattoso. Capturing and querying workflow runtime provenance with prov: a practical approach. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, EDBT '13, pages 282–289, New York, NY, USA, 2013. ACM.

[5] D. de Oliveira, E. Ogasawara, F. Baião, and M. Mattoso. Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 378–385, 2010.

[6] Jiang Dejun, Guillaume Pierre, and Chi-Hung Chi. Ec2 performance analysis for resource provisioning of service-oriented applications. In *Proceedings of the 2009 international conference on Service-oriented computing*, ICSOC/ServiceWave'09, pages 197–207, Berlin, Heidelberg, 2009. Springer-Verlag.

[7] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. *Computer*, 40(12):24–32, 2007.

[8] Keith R. Jackson, Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. Performance analysis of high performance computing applications on the amazon web services cloud. In *Cloud Computing, Second International Conference, CloudCom 2010, November 30 - December 3, 2010, Indianapolis, Indiana, USA, Proceedings*, pages 159–168, 2010.

[9] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system: Research articles. *Concurr. Comput. : Pract. Exper.*, 18(10):1039–1065, August 2006.

[10] Panayiotis Neophytou, Panos K. Chrysanthis, and Alexandros Labrinidis. Confluence: Implementation and application design. In Dimitrios Georgakopoulos and James B. D. Joshi, editors, *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2011, Orlando, FL, USA, 15-18 October, 2011*, pages 181–190, 2011.

[11] Eduardo S. Ogasawara, Daniel de Oliveira, Patrick Valduriez, Jonas Dias, Fabio Porto, and Marta Mattoso. An algebraic approach for data-centric scientific workflows. *PVLDB*, 4(12):1328–1339, 2011.

[12] Jörg Schad, Jens Dittrich, and Jorge-Arnulfo Quiané-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proc. VLDB Endow.*,

3(1-2):460–471, September 2010.

[13] Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *Proceedings of the International Conference on Dependable Systems and Networks*, DSN '06, pages 249–258, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Christoph Schuler, Heiko Schuldt, Can Türker, Roger Weber, and Hans-Jörg Schek. Peer-to-peer execution of (transactional) processes. *Int. J. Cooperative Inf. Syst.*, 14(4):377–406, 2005.

[15] Christoph Schuler, Can Turker, Hans-Jorg Schek, Roger Weber, and Heiko Schuldt. Scalable peer-to-peer process management. *International Journal of Business Process Integration and Management*, 1(2):129–142, 2006.

[16] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, and Ion Stoica. Improving mapreduce performance in heterogeneous environments. In *Proceedings of the 8th USENIX conference on Operating systems design and implementation*, OSDI'08, pages 29–42, Berkeley, CA, USA, 2008. USENIX Association.

ACM SIGIR 2014 JULY 6 – 11, 2014

THE 37TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE

Planning is well underway for the 37th Annual ACM SIGIR Conference, to be held on the Gold Coast, Queensland from Sunday 6 – Friday 11 July 2014.

SIGIR is the major international forum for the presentation of new research results and for the demonstration of new systems and techniques in the broad field of information retrieval. Next year's conference will feature 6 days of papers, posters, demonstrations, tutorials and workshops focused on research and development in the area of informational retrieval, also known as search.

The Conference and Program Chairs are now inviting all those working in areas related to information retrieval to submit original papers related to any aspect of information retrieval theory and foundation, techniques and application. A list of key submission dates, relevant paper topics, submission guidelines and instructions are now available on the official SIGIR 2014 Conference website: http://sigir.org/sigir2014/callforpapers.php. Abstract submission closes 20 January 2014.

In addition, to a full scientific program the conference presents delegates with the perfect networking opportunity, bringing together several hundred researchers, academic faculty, students and industry leaders from around the world.

SIGIR 2014 will take place at one of Australia's premier tourist destinations, the Gold Coast. From the iconic Surfers Paradise beach, to the sophisticated dining precincts of Broadbeach and out to the lush, green Hinterland, there is a new experience waiting for you at every turn on the Gold Coast. Theme parks, world-renowned beaches, shopping and almost year-round sunshine are just a few reasons why delegates will enjoy this vibrant coastal city.

From the SIGIR Conference Organising committee we hope to see you on the Gold Coast in 2014 for the 37th Annual ACM SIGIR Conference.

Call for Papers
ACM e-Energy 2014
Cambridge, UK, June 11-13 2014

Computing and communication technologies impact energy systems in two distinct ways. The exponential growth in deployment of these technologies has made them large-scale energy consumers. Therefore, new architectures, technologies and systems are being developed and deployed to make computing and networked system more energy efficient.

Additionally, and perhaps more importantly, these technologies are at the center of the on-going revolution in next generation "smart" and sustainable energy systems. They measure, monitor and control energy systems such as the smart grid; inform and shape human demand; aid in the prediction, deployment, storage and control of energy resources; and determine how utilities, generators, regulators, and consumers measure, analyze, and collectively control system elements.

The fifth International Conference on Future Energy Systems (ACM e-Energy), to be held in Cambridge UK in June 2014, aims to be the premier venue for researchers working in the broad areas of computing and communication for smart energy systems (including the smart grid), and in energy-efficient computing and communication systems. By bringing together researchers in a high-quality single-track conference with significant opportunities for individual and small-group interaction, it will serve as a major forum for presentations and discussions that will shape the future of this area.

We solicit high-quality papers in the area of computing and communication for the Smart Grid and energy-efficient computing and communications. We welcome submissions describing theoretical advances as well as system design, implementation and experimentation. ACM e-Energy is committed to a fair, timely, and thorough review process providing authors of submitted papers with sound and detailed feedback.

Relevant topics for the conference include, but are not limited to the following:

- Advances in monitoring and control of smart homes and buildings
- Sensing, monitoring, control, and management of energy systems
- Energy-efficient computing and communication, including energy-efficient data centers
  The impact of storage integration on the smart grid
- Electric Vehicle monitoring and control
- Distribution and transmission network control techniques
- Microgrid and distributed generation management and control
- Modeling, control, and architectures for renewable energy generation resources
- Smart grid communication architectures and protocols
- Privacy and security of smart grid infrastructure

- Innovative pricing and incentives for demand-side management
- Novel technologies to enhance reliability and robustness of energy systems
- HCI for energy monitoring, management, and awareness
- User studies and behavioral change enabled by computing and communication technologies
- Data analytics for the smart grid and energy-efficient systems
- Modeling, management and control of variability and uncertainty in energy supply and demand

Two type of contributions are solicited:

Full papers, up to 12 pages in ACM double-column format, should present original theoretical and/or experimental research in any of the areas listed above that has not been previously published, accepted for publication, or is not currently under review by another conference or journal.

Poster/demo descriptions, up to 2 pages in ACM double-column format showcasing works in progress. Accepted posters/demos will be presented at the conference. Topics of interest are the same as the research topics listed above. Preference will be given to posters/demos where the primary contribution is from one or more students.

Full submission details can be found at the conference website:

http://conferences.sigcomm.org/eenergy/2014

Important Dates:
- January 15, 2014: paper submission deadline
- March 21, 2014: Author notification
- April 7, 2014: Camera ready papers due
- June 11-13, 2014: ACM e-Energy conference, Cambridge, UK

Organizing Committee:
- General Chairs: Jon Crowcroft, Richard Penty (U. Cambridge, UK)
- TPC Co-chairs: Jean-Yves Leboudec (EFPL), Prashant Shenoy (U. Massachusetts)