SIGMOD Officers, Committees, and Awardees

Chair

Yannis Ioannidis
University of Athens
Department of Informatics
Panepistimioupolis, Informatics Bldg
157 84 Ilissia, Athens
HELLAS
+30 210 727 5224
<yannis AT di.uoa.gr>

Vice-Chair

Christian S. Jensen
Department of Computer Science
Aarhus University
Åbogade 34
DK-8200 Århus N
DENMARK
+45 99 40 89 00
<csj AT cs.aau.dk >

Secretary/Treasurer

Alexandros Labrinidis
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260-9161
PA 15260-9161
USA
+1 412 624 8843
<labrinid AT cs.pitt.edu>

SIGMOD Executive Committee:

Sihem Amer-Yahia, Curtis Dyreson, Christian S. Jensen, Yannis Ioannidis, Alexandros Labrinidis, Maurizio Lenzerini, Ioana Manolescu, Lisa Singh, Raghu Ramakrishnan, and Jeffrey Xu Yu.

Advisory Board:

Raghu Ramakrishnan (Chair), Yahoo! Research, <First8CharsOfLastName AT yahoo-inc.com>,
Amr El Abbadi, Serge Abiteboul, Rakesh Agrawal, Anastasia Ailamaki, Ricardo Baeza-Yates, Phil Bernstein, Elisa Bertino, Mike Carey, Surajit Chaudhuri, Christos Faloutsos, Alon Halevy, Joe Hellerstein, Masaru Kitsuregawa, Donald Kossmann, Renée Miller, C. Mohan, Beng-Chin Ooi, 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:

Yanif Ahmad, Denilson Barbosa, Pablo Barceló, Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, 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:

Rakesh Agrawal, Elisa Bertino, Umesh Dayal, Masaru Kitsuregawa (chair, University of Tokyo, <kitsure AT tk1.iis.u-tokyo.ac.jp>) 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 : March 21st, 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. Until 2003, this award was known as the "SIGMOD Innovations Award." In 2004, SIGMOD, with the unanimous approval of ACM Council, decided to rename the award to honor 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)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)	Moshe Y. Vardi (2008)	Masaru Kitsuregawa (2009)
Umeshwar Dayal (2010)	Surajit Chaudhuri (2011)	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)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)	Klaus R. Dittrich (2008)	Beng Chin Ooi (2009)
David Lomet (2010)	Gerhard Weikum (2011)	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, University of Toronto; Yanlei Diao, University of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley. *Honorable Mentions*: Xifeng Yan, University of Indiana at Urbana Champaign; Martin Theobald, Saarland University
- **2008** *Winner*: Ariel Fuxman, University of Toronto. *Honorable Mentions*: Cong Yu, University of Michigan; Nilesh Dalvi, University of Washington.
- **2009** *Winner*: Daniel Abadi, MIT. *Honorable Mentions*: Bee-Chung Chen, University of Wisconsin at Madison; Ashwin Machanavajjhala, Cornell University.
- **2010** *Winner:* Christopher Ré, University of Washington. *Honorable Mentions*: Soumyadeb Mitra, University of Illinois, Urbana-Champaign; Fabian Suchanek, Max-Planck Institute for Informatics.
- **2011** *Winner*: Stratos Idreos, Centrum Wiskunde & Informatica. *Honorable Mentions*: Todd Green, University of Pennsylvania; Karl Schnaitter, University of California in Santa Cruz.
- **2012** *Winner*: Ryan Johnson, Carnegie Mellon University. *Honorable Mention*: Bogdan Alexe, University of California in Santa Cruz.

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

[Last updated : December 18th, 2012]

Editor's Notes

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

The article opening this issue is an outline of research around "The Continuous Distributed Monitoring Model" by Cormode. The setting considered is one where multiple distributed sites collaborate in monitoring some (potentially distributed) phenomenon and must implement, together, a specific computation on their monitoring result, such as counting events, or detecting "unusual" activity according to a given (un)usual activity profile. The key is to accurately capture the interesting information, for instance, record each event exactly once, or detect each unusual pattern, with as little communication effort as possible among the participating sites. The survey outlines existing works on variants of the problem such as distributed countdown (the sites must sum up their observations to detect the arrival of a given number of events) and entropy monitoring (where the problem is to detect changes in the entropy of the distributed system under observation), and other extensions, while also pointing to the connection with related areas of work and outlining possible advances to be made in future research.

The survey by Schomm, Schall and Vossen focuses on the hot topic of data marketplaces, defined as places where anyone (or at least a large number of users) can connect to upload and/or obtain datasets, for free or for a fee. The authors identify twelve categories according to which existing data marketplaces can be classified, and present the distribution of 46 data marketplaces surveyed as of mid-2012 according to these categories. Depending on the core product (=data) they peddle, marketplaces can be split in several classes, such as crawlers that are services to be invoked to obtain data from a given target crawl site, search engines over specific existing databases, raw data vendors, tagging services (which enrich a dataset given as input with annotations based on an ontology (semantic model) etc. The analysis is also based on other dimensions such as the data pricing model, data formats, and the maturity of the marketplace platforms.

In the Distinguished Profiles column, we have the pleasure of reading Hank Korth' reflections on his long successful career as a researcher at Bell Labs and Panasonic, and a professor and deparment head at LeHigh University among others. The Korth-Silberschatz (later Korth-Silberschatz-Sudarshan) database book has reached its sixth edition in print, which is at least a reason why most if not all of our readers already know one of Hank's works very well! Hank also shares in the interview thoughts on his experience with transferring research results to the industry – a topic obviously non-trivial even for a large company such as Bell Labs, teaching computer science to non-CS majors, playing ultimate Frisbee with the undergraduate students, seizing career opportunities and much more.

The Research Centers column features two research center presentations. The first one, by Chakrabarti, Ramakrishnan, Ramamrithan, Sarawagi and Sudarshan, is from the IIT Bombay. The paper outlines the group's research areas, featuring an interesting mix of database, information retrieval and machine learning. The research topics comprise graph querying, keyword search, entity annotation, Web tables, working with imprecise data, continuous queries and streams. The second one, by Stonebraker, Madden and Dubey, is from the Intel "Big Data" science and technology center, created in 2012. The report outlines the center's vision of the current state of "Big Data" technology and the problems currently open, which in the authors' view include complex analytics on big-volume data, and integrating high-velocity data with a large persistent state.

Two workshop reports are included in this issue. Pedersen, Lehner and Hackebroich report on the works of the Energy Data Management (EnDM) 2012 workshop, held in conjunction with EDBT 2012. The workshop focused on the data management issues raised by intelligent distribution networks, e.g., electricity networks using smart meters, performance and energy reduction—aware file placement, smart

grids, database techniques for energy environmental impact assessment, and gaming techniques to encourage users to adopt energy saving behaviors. The Cloud Data Management (CloudDB) workshop, held next to ACM CIKM 2012, is described in the report by Meng, Wang and Silberstein. The workshop has covered topics such as OLTP benchmarking in the cloud, data analytics, scaling out social applications, privacy and security, energy-efficient clouds, as well as more specific performance-oriented techniques for large-scale data management in the cloud.

The Call for Participation to the ACM SIGMOD/PODS yearly conference closes this issue. Looking forward to see you all in New York!

Your contributions to the Record are welcome via the RECESS submission site (http://db.cs.pitt.edu/recess). 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

March 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)

The Continuous Distributed Monitoring Model*

Graham Cormode
AT&T Labs—Research
graham@research.att.com

ABSTRACT

In the model of continuous distributed monitoring, a number of observers each see a stream of observations. Their goal is to work together to compute a function of the union of their observations. This can be as simple as counting the total number of observations, or more complex non-linear functions such as tracking the entropy of the induced distribution. Assuming that it is too costly to simply centralize all the observations, it becomes quite challenging to design solutions which provide a good approximation to the current answer, while bounding the communication cost of the observers, and their other resources such as their space usage. This survey introduces this model, and describe a selection results in this setting, from the simple counting problem to a variety of other functions that have been studied.

1. INTRODUCTION

The model of continuous, distributed monitoring is a quite natural one, which arose only in the early years of the 21st century. It abstracts an increasingly common situation: a number of observers are making observations, and wish to work together to compute a function of the combination of all their observations. This abstract description can be applied to a number of settings:

- Network elements within the network of a large ISP are observing local usage of links, and wish to work together to compute functions which determine the overall health of the network.
- Many sensors have been deployed in the field, with the aim of collecting environmental information, and need to cooperate to track global changes in this data.
- A large social network monitors the usage of many compute nodes in data centers spread around the world, and wants to coordinate this information to track shifts in usage patterns and detect any unusual events, possibly indicative of an attack or exploit.

Each of these examples maps naturally onto the outline above: the network elements, sensors and compute nodes respectively play the part of the observers, who want to collaborate in the computation.

There are various "trivial" solutions to these problems. Studying the drawbacks of these helps us to identify the properties to optimize. A first approach is to simply have all the observers send all their observations to a single, centralized location. For cases where the flow of new observations is sufficiently slow, then indeed this is a satisfactory solution. However, in the above scenarios, this places an intolerable burden on the underlying network. For example, in the ISP example, the number of observations may be equivalent to the total number of packets traveling on a link: generating this much extra traffic on the network for the purpose of health monitoring will quickly contribute to the ill-health of the network!

A second approach is to perform "periodic polling": at some fixed interval, say every five minutes, or once an hour, a central monitor polls each observer for information about their observations since the last poll, and collates these together to get a snapshot of the current status. Again, in some situations, this will suffice. Indeed, many network protocols, such as the Simple Network Management Protocol (SNMP) operate on exactly this basis. Still, often this too is insufficient. Firstly, we require that the information needed can be summarized compactly. For example, SNMP allows the reporting of the total amount of traffic (measured in packets or bytes) processed by a network element within a given time window. Quantities like sums and counts of observations, therefore, fit naturally within this setting. However, when the objective is a more complex function, like measuring some non-linear function of all the (distributed) observations, or detecting when some complex event has occurred, it is less clear how periodic polling can operate.

The other limitation of periodic polling is the careful balance needed in setting the frequency of the polling event. Set the gap too narrow, and again the network becomes overloaded with data which may be of limited

*An earlier version of this survey was published in the Proceedings of the Workshop on Algorithms and Models for Distributed Event Processing, 2011

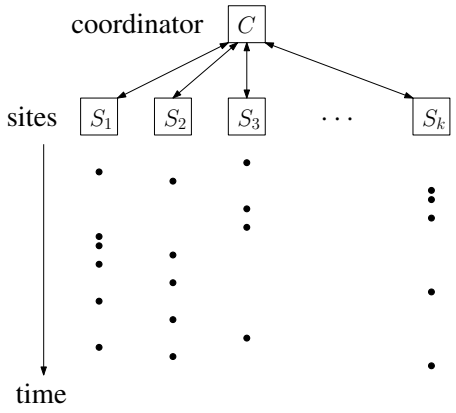

Figure 1: Continuous Distributed Monitoring model

usefulness. But set the gap too large, and the delay between an important event occurring and it being detected by the protocol may become too large.

In continuous distributed monitoring, we aim to address all these concerns. The central idea is to incur minimal communication when there is nothing important being observed, but at the same time to enable rapid (near-instantaneous) updates when necessary.

There has been considerable research effort in this area since its inception. Progress has been made by considering sets of fundamental functions, and describing protocols which provide strong guarantees on the accuracy of the monitoring, while incurring low costs, in the form of communication required, and computational overhead and storage needed by the observers.

Outline. The rest of this survey proceeds as follows. First, we formalize the model, and define the key cost measures. Then we begin by considering a seemingly simple problem in this setting, the problem of counting a fixed number of events, in Section 2. Section 3 considers monitoring the information theoretic concept of entropy, which varies non-monotonically as the number of events increase. We then describe a very general approach to problems in this model via the "geometric approach", in Section 4. Section 5 considers how to maintain a random sample, of either the entire data, or only a recent selection. In Section 6, we outline the history of the model and other results in this area, while Section 7 presents some concluding remarks and open problems.

1.1 Formalizing the model

In total we have k observers (or sites), indexed $S_1, \ldots S_k$. Each observer sees a stream of observations. Typically, each individual observation is quite simple, but in aggregate these define a complex whole. For example, in a communication network, each event might be the arrival of a packet at a router. The description of each event is quite simple: the destination and payload size, say. But the overall distribution of traffic to different destinations observed by multiple routers is very large and complex.

We treat the observations as items $A = a_1, a_2, \ldots a_n$, such that each observation is seen by exactly one observer. There is also a central site, or coordinator, C, who can communicate directly with each observer. For simplicity, we do not allow communication between observers (this can be achieved by sending messages through the coordinator), and we assume each message has unit cost. Varying these assumptions leads to different cost models, some of which are studied in the works described in Section 6. The goal of the monitoring is for the coordinator to continually track some function $f(A)$ over the complete set of observations.

In this survey, we see several different cases of this problem. In 'threshold monitoring', the goal is to determine whether $f(A)$ is above or below a threshold τ. For example, we may want to know when the total network traffic in the last hour exceeds a given amount; or when the entropy of this traffic distribution exceeds a given bound. In 'value monitoring', the goal is to provide an estimate $\hat{f}(A)$ of $f(A)$, such that the difference $|\hat{f}(A) - f(A)|$ is bounded. In the network example, this corresponds to providing an approximate value of the total network traffic; or of the entropy of the traffic distribution. In 'set monitoring', the goal is to provide a set of values which satisfy some property. This could be a uniform sample of the input items, or an approximate top-k (e.g. the top-k most popular destinations in the network).

Figure 1 gives a schematic of the model: communication is between the coordinator and the k different sites. New observations are made over time, which prompts more communication between the parties.

1.2 Comparison to Other Models

There are several other models of computation over data which may be rapidly arriving or distributed. Here, we identify some common models, and outline the key differences.

Communication Complexity. The model of communication complexity focuses on the case where there are two parties, Alice who holds input x and Bob who holds input y, and they wish to work together to compute $f(x, y)$ for some fixed function f [29]. The most important difference between this model and the continuous distributed monitoring case is that the inputs x and y are fixed for communication complexity, whereas in our case, they are allowed to vary. Moreover, it turns out that the main focus of communication complexity is providing lower bounds or impossibility results for various functions, whereas in continuous distributed monitoring, there has been most interest in providing protocols with low communication costs. However, the mod-

els are closely related: techniques from communication complexity have been used to show lower bounds for problems in continuous distributed monitoring [40, 39].

The Data Streaming Model. In the streaming model, a single observer sees a large stream of events, and must keep a sublinear amount of information in order to approximate a desired function f [32]. This omits the key feature of the continuous distributed model, the fact that multiple distributed observers need to compute a function of all their inputs combined. While each observer in our model sees a stream of inputs, the model does not insist that they use sublinear space—rather, the space used by each observer is an additional property of any given protocol. However, it is often desirable that the observers use small space, and techniques from stream processing are therefore useful to help achieve this.

Distributed Computation. Clearly, the continuous distributed model is a special case within the general area of distributed computation. The focus on continually maintaining a function of evolving input distinguishes it from the general case. There are other models within distributed computation, such as the Distributed Streams Model [20, 21] or the Massive, Unordered Data model [17]. These capture the emphasis on distributed streams of data, but focus on a one-time computation, rather than continually tracking a function.

2. THE COUNTDOWN PROBLEM

We begin with a seemingly simple problem which nevertheless admits some fairly sophisticated solutions. In the *countdown problem*, each observer sees some events (non-overlapping, so each event is seen by only one observer), and we wish to determine when a total of τ events have been seen. This is an instance of threshold monitoring. This abstract problem captures many natural settings: we want to raise an alert when more than τ unusual network events have been seen; report when more than 10,000 vehicles have crossed a highway; or identify the 1,000,000th customer; and so on. A trivial solution has each observer send a bit for each event they observe, which uses $O(\tau)$ communication. We aim to considerably improve over this baseline.

A first approach. A smarter approach takes advantage of the fact that there have to be many events at each site before the threshold τ can have been reached. A necessary condition is that at least one of the k sites must observe τ/k events before the threshold can be reached. This leads to a relatively simple scheme (derived from [28]): Each site begins with an initial upper bound value of τ/k, and begins to observe events. Whenever its local count n_i exceeds this upper bound, it informs the coordinator, which collects n_i from each observer, and the n_is are reset to zero. From these, we can determine the current "slack": the difference S between the current count N and the threshold τ, i.e. $S = \tau - N$. This slack can then be redistributed to the observers, so each site now enforces an upper bound of S/k on n_i. Each iteration reduces the slack by a factor of $(1-1/k)$. When the slack (initially τ) reaches k, the observers can switch to reporting every event. The number of slack updates is then

$$\log_{1/(1-\frac{1}{k})}\left(\frac{\tau}{k}\right) = \frac{\log(\frac{\tau}{k})}{\log(\frac{1}{1-\frac{1}{k}})} = O\big(k \log \frac{\tau}{k}\big)$$

The total communication is $O(k^2 \log \tau/k)$, since each update causes communication of $O(k)$.

A quadratic improvement. The step of updating every node whenever one node reports that it has exceeded its current local threshold is somewhat wasteful. This can be improved on by tolerating more updates before a global communication is triggered. This idea was introduced in [10], and we follow the simplified version described in [11].

Now the protocol operates over $\lceil \log(\tau/k) \rceil$ rounds. In the jth round, each observer sends a message to the coordinator when its local count n_i reaches $\lfloor 2^{-j}\tau/k \rfloor$, and then subtracts this amount from n_i. So, in the first round, this bound is $\lfloor \tau/2k \rfloor$. In the jth round, the coordinator waits until it has received k messages, at which point the round is terminated, and the coordinator alerts each site to begin the $j+1$th round, causing the bound to approximately halve. This continues until the bound reaches 1, when each site reports each event when it occurs. Observe now that the communication in each round is more "balanced": the sites send a total of k messages, and the coordinator sends k messages (to inform each site that the new round has begun). Each of these messages can be constant size. Thus, the total communication is $O(k \log \tau/k)$: a factor k improvement over the prior approach.

It also follows immediately that protocol is correct: in any round, the total "unreported" count is at most

$$k\lfloor \tau 2^{-j}/k \rfloor \leq \tau/2^j,$$

while the "reported" count is at most

$$\sum_{i=1}^{j-1} k\lfloor \tau 2^{-i}/k \rfloor \leq \tau \sum_{i=1}^{j-1} 2^{-i} \leq \tau(1-2^{-j}).$$

Hence, the total count never exceeds τ until the final round, when every event is reported directly.

Approximate Countdown. We can improve on the cost of this protocol if we are prepared to tolerate some imprecision in the result. Specifically, we consider protocols which approximate the answer. To approximate, we introduce a parameter ϵ, and ask that the coordinator can determine that the true count is below $(1-\epsilon)\tau$

or above τ; when the true count is in between, then the coordinator can indicate either state.

The protocol is almost identical, but now we terminate when the bound on the unreported count reaches $\epsilon\tau$. The number of rounds is reduced to $\log 1/\epsilon$. This removes τ from the bounds, and makes the total cost of the protocol $O(k \log 1/\epsilon)$ communication.

Countdown lower bounds. We might ask if we can improve further on this result. For deterministic solutions, the answer is no: this bound is tight. This was shown formally in [10]. The intuition is natural: consider the perspective of a single observer, who witnesses a number of events. When this number is substantial enough, it could be part of a global trend, and so must be reported in case they push the total count above the threshold τ. At the same time, it might just be a local phenomenon, in which case any communication does not change the overall answer. Since the observer cannot distinguish these two cases unless it receives a message from the coordinator, then it is forced to communicate. Based on this argument, it is possible to show that the total amount of communication is at least $\Omega(k \log \tau/k)$.

Randomized Countdown Protocol. We can give tighter bounds if we allow both randomization and approximation. Allowing randomization means that we let the protocol have a small probability of giving an erroneous answer at some point in its operation.

The randomized protocol operates as follows, based on a constant c determined by the analysis. Each site observes events, and after collecting a "bundle" $\epsilon^2\tau/(ck)$ of observations, it decides whether to send a message to the coordinator. With probability $1/k$ it sends a message, but with probability $1 - 1/k$, it stays silent. The coordinator declares that enough events have been seen once it has received $c(1/\epsilon^2 - 1/2\epsilon)$ messages. The idea here is that there will be enough opportunities to send messages that with high probability the coordinator will not declare too early or too late. We omit a full analysis here: it can be shown that the amount of communication from sites is $O(1/\epsilon^2)$, and the coordinator is unlikely to declare that the threshold τ has been passed before the true count reaches $(1 - \epsilon)\tau$. Note that this omits the cost to initiate and terminate the protocol, which involves alerting all k sites.

Non-monotonic counts. The approaches outlined for the countdown problem rely critically on the fact that the function being monitored was monotonic: the number of events kept increasing. The non-monotone case is more complex. In general, the count might increase and decrease a lot while close to zero, forcing a lot of communication even for approximate, randomized protocols. However, in cases when there is some randomness in the update streams — for example, when they follow a random walk, or the arrivals are randomly permuted — then stronger guarantees can be provided [31].

3. MONITORING ENTROPY

We next consider monitoring the entropy function from Information Theory. Consider the case where the observers are now witnessing events in the form of arrivals of different items. These arrivals generate an empirical probability distribution (recording the relative proportion of each different item observed), which we can compute the entropy of.

Entropy. Suppose that f_i denotes the number of occurrences of item i observed across the whole system, and m denotes the total number of items (so $m = \sum_i f_i$). Then the empirical probability of i is just f_i/m, and the entropy H of the distribution is given by

$$H = \sum_i \frac{f_i}{m} \log \frac{m}{f_i}$$

The entropy H is an important metric on the distribution: if all f_is are about equal, then the entropy is high, while if most f_is are small and only one is significant, then the entropy is low. It has been argued that changes in entropy are an important indicator of changes in behavior in distributed systems and networks [30].

Entropy Protocol Outline. Arackaparambil *et al.* design a protocol to monitor entropy in the continuous distributed monitoring model [2]. Specifically, they design an approximate protocol, which determines whether the current entropy H is above a given boundary τ, or below $(1 - \epsilon)\tau$. The overall protocol is quite straightforward: the key step is an invocation of an approximate protocol for the countdown problem from Section 2. The protocol proceeds in a number of rounds. In the first round, each site sends every item it receives directly to the coordinator, until some constant number (say, 100) of items have been observed across all sites. This is because the entropy can change quickly in this initial stage. In each subsequent round i, the coordinator computes a parameter τ_i, and runs an instance of the approximate countdown protocol for threshold τ_i, with a constant approximation factor $\epsilon = \frac{1}{2}$. When this protocol terminates, the coordinator contacts each site, which sends a description of its current distribution. The coordinator combines these to estimate the current entropy, and uses this to compute the parameter τ_{i+1} for the next round.

The analysis relies on a basic property of the entropy function: the change in entropy between two points is bounded in terms of the number of new observations. Specifically, if the number of observations at the first point is m, and there are n new arrivals, the change in entropy is at most $\frac{n}{m} \log(2m)$ [2]. Thus, since we know the entropy at the end of round i, and we wish

to know if it changes by at most $\epsilon\tau/2$ (the minimum change needed to change the output of the coordinator), we can set $\tau_{i+1} = \frac{\epsilon\tau m}{2\log(2m)}$, where m is the total number of observations made at the end of round i. Given an upper bound N on the total number of observations, we can ensure that m_i, the total number of observations at the end of round i, satisfies

$$m_{i+1} = m_i + \tau_{i+1} = m_i\left(1 + \frac{\epsilon\tau}{2\log(2m_i)}\right) \geq m_i\left(1 + \frac{\epsilon\tau}{2\log(2N)}\right)$$

and hence the number of rounds to reach N observations is $O(\frac{1}{\epsilon\tau}\log^2 N)$ (provided $\log N \geq \tau\epsilon$).

The communication cost in each round is $O(kD)$, where D is an upper bound on the number of distinct items observed at each of the k sites. When D becomes large, we can instead communicate compact *sketches* of the distribution, which allow us to estimate a function (in this case, entropy) of the combination of the inputs. There are randomized sketches which provide $(1 \pm \epsilon)$ approximation of the entropy using a data structure of size $\tilde{O}(\frac{1}{\epsilon^2})$, where the $\tilde{O}$ notation suppresses logarithmic factors [18, 22].

Lower bounds for entropy monitoring. Lower bounds for this problem can be generating by defining a set of possible inputs chosen so that any individual site cannot tell which case it is in, and so is forced to communicate to resolve this uncertainty. This leads to a deterministic lower bound of $\Omega(k\epsilon^{-1/2}\log(\epsilon N/k))$ and a randomized lower bound of $\Omega(\epsilon^{-1/2}\log(\epsilon N/k))$ [2]. Note that the above protocol is essentially deterministic, and so the stronger bound applies to this case. Recently, Woodruff and Zhang showed stronger lower bounds of $\Omega(k/\epsilon^2)$ for entropy when the input may include arrivals and departures of items [39].

4. THE GEOMETRIC APPROACH

The two results discussed so far considered specific problems (countdown and entropy), and provided tailored protocols based on exploiting specific properties of each function. It is natural to ask whether there are general purpose techniques for generating protocols in this model. The "geometric approach", due to Sharfman, Schuster and Keren aims to do exactly this [35]. The basic idea is to take any desired function, f, and break down the testing of whether $f(x) > \tau$ or $f(x) \leq \tau$ into conditions which can be checked locally, even though x represents the global state of the system. The central result relies on a neat geometric fact, that the area of a convex hull of a set of points can be fully covered by a set of spheres, one sphere incident on each point.

4.1 Formal Description

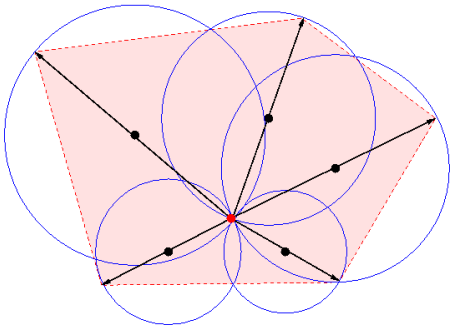

Figure 2: Current estimate e (central red dot), drift vectors Δv_i (arrows out of e), convex hull (dotted outline) and enclosing balls

Preliminaries. Each stream observed at each site is assumed to define a current d dimensional vector v_i. In the countdown case, each v_i was simply the local count; in the entropy case it was the local frequency distribution. With each site we associate a weight λ_i such that these weights sum to 1, i.e. $\sum_{i=1}^k \lambda_i = 1$. These weights might reflect the number of observations at each site, so in this case $\lambda_i = n_i / \sum_{i=1}^k n_i$. Or they may simply be uniform, i.e. $\lambda_i = 1/k$ for all i. Initially, assume that these weights are fixed and known to all nodes.

The weighted combination of all local vectors v_i gives the global vector $v = \sum_{i=1}^k \lambda_i v_i$. The instance of the threshold monitoring problem is then to determine whether $f(v) \leq \tau$ or $f(v) > \tau$, for a fixed function f and threshold τ. For example, we can map the countdown problem into this setting: here, we set $\lambda_i = 1/k$, each v_i is the single dimensional quantity $\langle n_i \rangle$ (number of event observations at site i), and $f(v) = \|v\|_1$. We set τ here to be $1/k$ times the desired threshold. In other words, v is the mean of the event counts at each site, and we want to alert when this mean exceeds a threshold that implies that the total count is above the global threshold.

Protocol Description. At any moment during the protocol, each site has previously informed the coordinator of some prior state of its local vector, v_i'. So the coordinator knows v_i', but not the current state v_i. Based on this knowledge, the coordinator has an estimated global vector $e = \sum_{i=1}^k \lambda_i v_i$. Clearly, if the local vectors v_i move too far from their last reported value v_i', it is possible that the τ threshold may be violated. Therefore, each site monitors its *drift* from its last reported value, as $\Delta v_i = v_i - v_i'$. Thus we can write the current global vector, v, in terms of the current estimate e and the drift vectors:

$$v = \sum_{i=1}^k \lambda_i v_i = \sum_{i=1}^k \lambda_i (e + \Delta v_i) = e + \sum_{i=1}^k \lambda_i \Delta v_i$$

Observe that this is a convex combination of drift vec-

tors. Therefore, the current global vector v is guaranteed to lie somewhere within the convex hull of the drift vectors v_i around e. Figure 2 shows an example in $d = 2$ dimensions, with five drift vectors emanating from an estimate e, and their convex hull. The current value must lie somewhere within this shaded region.

To transform the global condition into a local one, we place a ball on each local drift vector, of radius $\frac{1}{2}\|\Delta v_i\|_2$ and centered at $e + \frac{1}{2}\Delta v_i$. This is illustrated in Figure 2. It can be shown that the union of all these balls entirely covers the convex hull of drift vectors [35]. Thus, we reduce the problem of monitoring the global vector to the local problem of each site monitoring the ball of its drift vector.

Specifically, given the function f, we can partition the space into two sets: X, which is those points x for which $f(x) \leq \tau$, and $\bar{X}$, which is those for which $f(x) > \tau$. The basic protocol is now quite simple: each site monitors its drift vector Δv_i, and checks with each new observation if the ball given by $e + \frac{1}{2}\Delta v_i$ is *monochromatic*, i.e. all points in the ball fall in the same set (X or $\bar{X}$). If this is not the case, then the site communicates to the coordinator. The coordinator then collects the current vectors v_i from each site to compute a new estimate e, which resets all drift vectors to 0. From the above discussion of convex hulls, it is clear that when all balls are monochromatic in the same set (X or $\bar{X}$), then v must also be in the same set, and so the coordinator knows the correct state.

4.2 Extensions to the Geometric Approach

There are several extensions and variations of this basic geometric monitoring scheme which are able to reduce the cost, and avoid some bad cases.

Local Resolution via slack. Whenever a local drift vector creates a non-monotone ball, it causes communication with all sites, to collect their current vectors and distribute the new estimate. This global communication can be be postponed by the coordinator, who can introduce additional "slack", in the form of offset vectors. That is, the coordinator can contact a small number of sites, and allocate a set of vectors δ_i chosen so that the balls for $\Delta v_i + \delta_i$ are now monochromatic, and $\sum_{i=1}^{k} \delta_i = 0$. This idea is discussed in detail in [35]; similar concepts arose earlier, e.g. in work on tracking top-k of frequency distributions [3].

Approximate Thresholds. The version of the protocol described is for an exact version. We can reduce the cost by relaxing this requirement, and introducing an ϵ tolerance around τ. Applying this, when $f(v) < \tau$, we define the sets X and $\bar{X}$ as before, but when we are above the threshold, we define the sets based on $f(x) < (1-\epsilon)\tau$. This gives more room for the balls to grow, and prevents constant communication when the current value of $f(v)$ is close to τ.

Affine Transformations and Reference Vectors. The use of spherical balls is a natural one, but it is not the only choice. In [36], the authors observe that one can perform any affine transformation on the input, without changing the region covered by the convex hull. In some cases, the resulting ellipsoids can more tightly conform to the convex hull than spheres would. In the same work, the authors discuss replacing the estimate e with a difference reference vector. This can reduce communication by providing a larger "safe area" for the drift vectors to occupy.

Making Predictions. The concept of "prediction" was introduced by Cormode, Garofalakis, Muthukrishnan and Rastogi [9]. The idea is that if items are continually arriving at approximately even rates, then each site can share a simple prediction model of where its distribution will be at any given point in time, rather than relying on a static historical snapshot. Recent work has combined this idea with the geometric approach, and shown that this can be very effective in reducing the cost of monitoring [19].

5. SAMPLING

So far we have concentrated on the case of threshold monitoring: tracking which side of a threshold τ a given function f is on. This is actually quite a general task. For example, we might instead want to monitor the value of f, so that we always have an approximation to its value (value monitoring). But this can be modeled as multiple instances of the threshold monitoring task, for thresholds $1, (1+\epsilon), (1+\epsilon)^2 \ldots$. Tracking all these in parallel can be done by running $O(\frac{1}{\epsilon}\log T)$ instances of the threshold monitoring solution in parallel, where T is maximum value of the function. Although this $1/\epsilon$ factor is large enough to make it worthwhile designing new solutions for value monitoring problems, the techniques and approaches that have been used for value monitoring and threshold monitoring are quite similar.

There are some other monitoring tasks which do not fit either the threshold monitoring or value monitoring paradigms, and instead require us to track the members of a set (set monitoring). For example, we might want to extract information such as which are the k most frequently observed items across all the event streams [3]. In this section, we describe a basic task: to draw a uniform sample from the different event streams, based on the results from [11, 38]. We describe two variations: where we want to sample over all the events ever observed (the infinite window case), and where we want a sample only over the more recent events (the sliding window case).

5.1 Infinite Window

Recall the set-up: we have k distributed sites, each of which is observing events occurring at arbitrary and varying rates. We wish to compute a sample of size s of these events. First, we consider drawing a sample without replacement. The basic idea is to sample across all sites with the same probability p. All sampled items are sent to the coordinator to form a collection, from which s items can be extracted uniformly. Periodically, the coordinator may tell a site to reduce its local p value, and will also prune its collection. We want to bound the resources taken for this process, in terms of the amount of communication, and space needed by the participants.

A simple protocol is as follows [38]: each site i maintains a local p_i, initially 1. For each item that arrives at a site, a random value $0 \leq u \leq 1$ is choosen. If $u \leq p_i$, the item is forwarded to the coordinator, which returns an updated p value to use as the new p_i. The coordinator maintains a set of k items and their corresponding u values, and when a site sends a new item, the coordinator returns the current k'th smallest u value it has seen so far. The correctness of this process follows immediately from the description: the coordinator correctly maintains the k items achieving the k smallest (random) u values across the input, which gives a uniform random sample.

The analysis is a little more involved, but can be done by relating the cost of this simple protocol to a slightly more complex one that keeps a fixed sampling rate p across all sites, which is periodically decreased by a constant factor [11, 38]. The communication cost can then be bounded (with high probability) as $O(k \log_{k/s} n + s \log n)$. One can show a matching lower bound by arguing that this many different items should appear in a random sample over the course of the protocol [11].

The protocol can also be extended to sample with replacement. A trivial solution just runs the above protocol with $s = 1$ in parallel s times over. However, this blows up the costs by a factor of s. Instead, it is possible to take this idea, but to keep all instances of the protocol sampling at the same rate, thus reducing the communication from the coordinator. Analyzing this process allows us to argue that communication of this protocol is bounded by $O((k + s \log s) \log n)$.

5.2 Sliding Windows

A natural variation of continuous distributed monitoring problems is when we do not want to track events across an unbounded history, but rather to see only the impact of recent events. For example, in a network we may only want to include events which have happened within the last hour; in a sensor network, we may only want to track a window of 1 million recent events, and so on. A naive solution would just be to pick a fixed interval—say, 1 hour—and restart the protocol afresh at multiples of this interval. This has the benefit of simplicity, but means that we re-enter a 'start-up' phase every time the protocol restarts, and so we lose information and history around this time. Instead, we describe an approach that is almost as simple as this naive solution, but which provides a sample of an exact sliding window.

A Tale of Two Windows. The key insight needed to generate the solution is due to Braverman, Ostrovsky and Zaniolo [4], who observed that any sliding window can be decomposed into two pieces, relative to a fixed point in time: a growing window as new items arrive after the fixed point, and a shrinking or expiring window of items from before the fixed point. Suppose we want to maintain a sample of items drawn from the last W global arrivals. To draw a sample uniform from these W, we want to take all unexpired sampled items from the expiring window, and make up the shortfall by sampling from those in the growing window. A simple probability calculation shows that this does indeed provide us a uniform sample from the most recent W arrivals.

To implement this idea, we can run an instance of the countdown protocol to count off every W arrivals. We can also run an instance of the above sliding window protocol for drawing a sample beginning at every multiple of W arrivals, which we halt when W further items have arrived. The only additional information needed is that the coordinator needs to know when an item sampled in the expiring window has expired. This can be done by starting a fresh instance of the countdown problem for every sampled item (and terminating this when the item is ejected from the coordinator's collection). This gives the coordinator exactly what is needed to perform the above sampling process: drawing unexpired items from the expiring window, and making up the shortfall from the growing window. The cost of this protocol now grows as $O(ks \log(W/s))$ per window, but this is unavoidable: [11] shows that any protocol for this problem must incur $\Omega(ks \log(W/ks))$ cost.

6. OTHER RELATED WORK

The idea of continuous distributed monitoring is a natural one, and as such it has arisen independently in different areas, under different labels. An early form was as 'Reactive Monitoring' in the networking world. Here, Dilman and Raz introduced a problem that was essentially a variant of the countdown problem, and provided some solutions based on distributing slack amongst the observers [16]. The notion of testing whether a function had exceeded a global threshold appeared under the name of "distributed triggers", and was motivated by Jain *et al.* in a workshop paper [26].

The continuous distributed model has attracted most attention in the data management community. Early work

by Olston, Jiang and Widom focused on tracking a function over single values which could vary up and down, such as monitoring their sum [34]. Here, some uncertainty can be tolerated, so they introduce a natural "filter" approach, which assigns a local filter to each site so that if the current value is within the filter, it does not need to be reported. When a site's value falls outside its filter, the current value is reported, and the filter is re-centered on this value. Over time, some filters can be widened and others narrowed so that the total uncertainty remains bounded, but more slack is allocated to values that are less stable.

A similar approach was used by Babcock and Olston to report the top-k items from a distribution [3]. Again, some tolerance for approximate answers is necessary to avoid communicating every change. The central idea is to choose a set of "adjustment factors" for each item at each site, so that the local distribution after adjustment appears identical to the global distribution. Each site monitors its (adjusted) distribution, and reports if the local (adjusted) top-k changes. In this case, a costly 'rebalancing' stage is invoked.

The use of "predictions" was applied to complex functions such as join sizes (or equivalently, the inner product of large vectors) by Cormode and Garofalakis [7]. Here, the idea was to operate predominantly in "sketch space": a random linear transformation of the input down to low-dimensional vectors. Due to the linearity of the sketch transformation, a prediction based on linear or quadratic growth in different dimensions could be captured by a sketch of the (first order or second order) difference between past values, which in turn is the appropriate difference of sketches. Violations of predictions can be detected by testing the deviation between the actual and predicted sketches.

Huang *et al.* worked on tracking spectral properties of distributed data, where each time step adds a new row to a matrix of observations from different observers. The quantity of interest to be monitored here was the residual energy of the signal after removing the projections along the principal components [25]. Other work studied anomaly detection, where an anomaly occurs when the number of events exceeds an expected rate, over any historical window [24]. This can be seen as a variant of the countdown problem where there is a background process which depletes the number of observed events at a uniform rate. A different approach to this problem is due to Jain *et al.* [27], who consider optimizing slack allocation within a hierarchical network topology, and robustness within a dynamic network (nodes dying, or new nodes joining).

Many other specific functions have been studied in this model, including monitoring the cardinality of set expressions [15] tracking the (large) number of distinct elements observed [12], tracking clusterings of points in a metric space [13], sparse approximation of signals [33], and conditional entropy [1].

The continuous distributed model has also been studied from a more theoretical perspective. [10] revisited various fundamental functions: F_0 (number of distinct elements), F_1 (count/countdown) and F_2 (self-join size or Euclidean norm), and gave the first or improved worst-case bounds for these problems, as well as the first lower bounds. Woodruff and Zhang provided strong lower bounds for a variety of such foundational problems, based on the hardness of a number of primitive problems in communication complexity [39].

Yi and Zhang proposed improved bounds for tracking quantiles and heavy hitters [40]. Specifically, they show how both problems can be solved with total communication $O(k/\epsilon \log n)$ to provide ϵ-approximate results over streams of total length n. Chan *et al.* study the same problems in the context of time-based sliding windows, where only recent events are counted [5]. Cormode and Yi observed that the 'two window' approach used for sampling can also be applied to simplify the analysis, and achieve improved bounds [14].

7. CONCLUDING REMARKS

This survey has aimed to give a flavour of the line of work in continuous distributed monitoring, by highlighting a few problems and approaches, and identifying the breadth of other related work. For a different perspective (with, admittedly, a similar authorial tone), there are surveys and tutorials [8, 6].

Since those prior surveys, there has certainly been progress made in this area. In particular, additional problems have been studied; more robust bounds—both upper and lower bounds—have been proved on the communication costs, as well as other costs such as space; variant models have been introduced, such as sliding windows and the online-tracking model; and a broader set of researchers have worked on related problems (see, for example, the LIFT project, `lift-eu.org/`).

At the same time, many questions posed previously have yet to be fully addressed. Next, I outline two quite different directions for this area that are capable of generating interesting and important results.

Systems for Continuous Distributed Monitoring. While there has been considerable progress on developing protocols and techniques for continuous distributed monitoring, these have yet to translate to practical implementations. There have been several prototype studies of protocols in the works introducing them, which have indicated the potential for orders of magnitude savings in the amount of communication incurred. However, as far as I am aware, these have not translated to widespread adoption of these ideas, or incorporation into standard

protocols. Moreover, these trials have tended to be in simulated environments on recorded data streams, rather than "live" tests. Possibly the lack of uptake of these methods is due to a lack of urgency for the problems considered. While orders of magnitude saving may be possible, if the overhead of centralization, or the delay of polling is considered acceptable, then there is no requirement to implement a more complex monitoring solution. In other words, attention needs to focus on settings where the naive solutions do place an intolerable burden on the network. One interesting example arises in Massively Multi-player Online Role Playing Games (MMORPGs). Here, it has been argued that distributed monitoring of quantities (such as the health scores of players and enemies) would benefit from smarter solutions [23].

In terms of open problems, the basic challenge is to first develop libraries of code, and then evolve these into general purpose systems, so that they can be easily adopted by programmers and data owners. Or, there should exist systems for distributed monitoring which are as accessible and general purpose as traditional centralized database management systems. It remains to determine what classes of functions such tools should support. Should they be based on a collection of "typical" functions (such as the countdown and entropy monitoring problems), or adopt the more generic geometric monitoring approach? Should there be a general purpose, high level query language for flexibly specifying monitoring problems?

A Deeper Theory of Continuous Distributed Monitoring. In recent years, there have been theoretical results shown for problems in continuous distributed monitoring. For the first time, strong upper bounds on the amount of communication of certain protocols have been shown, when previously only heuristic results were known. In some cases these are complemented by lower bounds, sometimes matching or almost matching. Nevertheless, it seems that a richer notion of continuous communication complexity is called for.

There are several powerful results in the literature which could potentially be extended. The famed Slepian-Wolf theorem [37] captures the case where there are correlated sources. They can encode their outputs to allow correct decoding, while using a total amount of communication proportional to the joint entropy. We can cast this "distributed source coding" as a special case of continuous distributed monitoring, where the target is the streams. Then, what we seek is a generalization of Slepian-Wolf, that will capture a function of multiple inputs, rather than just the identity function. This could also take advantage of correlations over time as well as space.

Acknowledgments.

I thank S. Muthukrishnan and Ke Yi for many comments and suggestions.

8. REFERENCES

[1] C. Arackaparambil, S. Bratus, J. Brody, and A. Shubina. Distributed monitoring of conditional entropy for anomaly detection in streams. In *IPDPS Workshops*, 2010.

[2] C. Arackaparambil, J. Brody, and A. Chakrabarti. Functional monitoring without monotonicity. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2009.

[3] B. Babcock and C. Olston. Distributed top-k monitoring. In *ACM SIGMOD International Conference on Management of Data*, 2003.

[4] V. Braverman, R. Ostrovsky, and C. Zaniolo. Optimal sampling from sliding windows. In *ACM Principles of Database Systems*, 2009.

[5] H.-L. Chan, T.-W. Lam, L.-K. Lee, and H.-F. Ting. Continuous monitoring of distributed data streams over a time-based sliding window. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2010.

[6] G. Cormode and M. Garofalakis. Efficient strategies for continuous distributed tracking tasks. *IEEE Data Engineering Bulletin*, 28(1):33–39, March 2005.

[7] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *International Conference on Very Large Data Bases*, 2005.

[8] G. Cormode and M. Garofalakis. Streaming in a connected world: Querying and tracking distributed data streams. In *ACM SIGMOD International Conference on Management of Data*, 2007.

[9] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In *ACM SIGMOD International Conference on Management of Data*, 2005.

[10] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed, functional monitoring. In *ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[11] G. Cormode, S. Muthukrishnan, K. Yi, and Q. Zhang. Continuous sampling from distributed streams. *J. ACM*, 59(2):25, 2012.

[12] G. Cormode, S. Muthukrishnan, and W. Zhuang. What's different: Distributed, continuous monitoring of duplicate resilient aggregates on data streams. In *IEEE International Conference on Data Engineering*, 2006.

[13] G. Cormode, S. Muthukrishnan, and W. Zhuang. Conquering the divide: Continuous clustering of distributed data streams. In *IEEE International Conference on Data Engineering*, 2007.

[14] G. Cormode and K. Yi. Tracking distributed aggregates over time-based sliding windows. In *ACM Conference on Principles of Distributed Computing (PODC)*, 2011.

[15] A. Das, S. Ganguly, M. Garofalakis, and R. Rastogi. Distributed set-expression cardinality estimation. In *International Conference on Very Large Data Bases*, 2004.

[16] M. Dilman and D. Raz. Efficient reactive monitoring. In *IEEE INFOCOMM*, 2001.

[17] J. Feldman, S. Muthukrishnan, A. Sidiropoulos, C. Stein, and Z. Svitkina. On distributing symmetric streaming computations. In *ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[18] S. Ganguly and B. Lakshminath. Estimating entropy over data streams. In *European Symposium on Algorithms (ESA)*, 2006.

[19] N. Giatrakos, A. Deligiannakis, M. N. Garofalakis, I. Sharfman, and A. Schuster. Prediction-based geometric monitoring over distributed data streams. In *ACM SIGMOD International Conference on Management of Data*, pages 265–276, 2012.

[20] P. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 281–290, 2001.

[21] P. Gibbons and S. Tirthapura. Distributed streams algorithms for sliding windows. In *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2002.

[22] N. J. A. Harvey, J. Nelson, and K. Onak. Sketching and streaming entropy via approximation theory. In *IEEE Conference on Foundations of Computer Science*, 2008.

[23] K. Heffner and G. Malecha. Design and implementation of generalized functional monitoring, 2009. `http://www.people.fas.harvard.edu/~gmalecha/proj/funkymon.pdf`,

[24] L. Huang, M. N. Garofalakis, A. D. Joseph, and N. Taft. Communication-efficient tracking of distributed cumulative triggers. In *ICDCS*, 2007.

[25] L. Huang, X. Nguyen, M. Garofalakis, J. Hellerstein, A. D. Joseph, M. Jordan, and N. Taft. Communication-efficient online detection of network-wide anomalies. In *IEEE INFOCOMM*, 2007.

[26] A. Jain, J. Hellerstein, S. Ratnasamy, and D. Wetherall. A wakeup call for internet monitoring systems: The case for distributed triggers. In *Proceedings of the 3rd Workshop on Hot Topics in Networks (Hotnets)*, 2004.

[27] N. Jain, M. Dahlin, Y. Zhang, D. Kit, P. Mahajan, and P. Yalagandula. STAR: Self-tuning aggregation for scalable monitoring. In *International Conference on Very Large Data Bases*, 2007.

[28] R. Keralapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *ACM SIGMOD International Conference on Management of Data*, 2006.

[29] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[30] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM*, 2005.

[31] Z. Liu, B. Radunovic, and M. Vojnovic. Continuous distributed counting for non-monotonic streams. In *ACM Principles of Database Systems*, pages 307–318, 2012.

[32] S. Muthukrishnan. Data streams: Algorithms and applications. In *ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[33] S. Muthukrishnan. Some algorithmic problems and results in compressed sensing. In *Allerton Conference*, 2006.

[34] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *ACM SIGMOD International Conference on Management of Data*, 2003.

[35] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. In *ACM SIGMOD International Conference on Management of Data*, 2006.

[36] I. Sharfman, A. Schuster, and D. Keren. Shape sensitive geometric monitoring. In *ACM Principles of Database Systems*, 2008.

[37] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on on Information Theory*, 19:471–480, 1973.

[38] S. Tirthapura and D. P. Woodruff. Optimal random sampling from distributed streams revisited. In *DISC*, 2011.

[39] D. P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *ACM Symposium on Theory of Computing*, pages 941–960, 2012.

[40] K. Yi and Q. Zhang. Optimal tracking of distributed heavy hitters and quantiles. In *ACM Principles of Database Systems*, 2009.

Marketplaces for Data: An Initial Survey

Fabian Schomm[1] Florian Stahl[1] Gottfried Vossen[1,2]

[1]European Research Center for Information Systems (ERCIS)
University of Muenster, Germany
firstname.lastname@uni-muenster.de

[2]Waikato Management School
The University of Waikato, New Zealand

ABSTRACT

Data is becoming more and more of a commodity, so that it is not surprising that data has reached the status of tradable goods. An increasing number of data providers is recognizing this and is consequently setting up platforms for selling, buying, or trading data. We identify several categories and dimensions of data marketplaces and data vendors and provide a snapshot of the situation as of Summer 2012.

1. INTRODUCTION

Today information is one of the crucial driving factors for most businesses. Only if high quality information is available, correct decisions (i. e., decisions in the interest of company revenues) can be made on a rational and well-founded basis. Despite the sheer quantities of data available on the Web, such information is not always easy to find, and data marketplaces, surveyed in this paper, are one of several recent developments to remedy this situation.

Shortly after the arrival of the Web in the early 1990s a new category of professionals emerged who took on the function of information intermediaries. To these intermediaries search task could be given, who would then search the Web correspondingly (for a fee) and return the results found. In 1998 the term *data marketplace* was probably first used by Armstrong and Durfee [1], who modeled trading of information between digital libraries, focusing on the motivation and behavior of participants and identifying factors that affect cooperations in a network.

Thanks to advances in technology, but also to the vast amount of data available nowadays, numerous new forms of marketplaces for data have emerged. A modern information intermediary or information marketplace in our understanding is a platform through which data can be purchased or sold. Commonly, they process, sell, and re-sell data available on the Web. By doing that, these platforms can provide added value in numerous ways. First, some data may be hard to find and scattered across numerous websites. A data vendor that aggregates these single datasets into a bigger and more refined one performs a service that makes it easier for customers or end-users to access relevant data. Secondly, datasets from different providers often have different access mechanisms and formats. Therefore, offering one single mechanism to access data in a consistent format can save time and money for customers.

This has also been realized by information providers who seek commercialization of their data. In accordance with that, it can be observed that evermore suppliers of data emerge. Aggregating and curating this data into accessible and understandable datasets is a business opportunity with high potential, driven by the over-supply of data.

While there have been small, not primarily scientific surveys of data marketplaces ([7, 10, 11]) and research on specific data marketplaces such as the Windows Azure Marketplace [9] and others (e. g., [12]), there is—to our knowledge—to date no comprehensive survey and comparison of multiple data marketplaces and data vendors. Therefore, we have conducted such a survey, including a total of 46 suppliers of data. The study was conducted from April to July 2012[1] with the aim of identifying categories and dimensions of data marketplaces as well as vendors of data in order to build a taxonomy for data marketplaces.

[1]The list of companies surveyed can be found at `http://dbis-group.uni-muenster.de/temporary_downloads/SurveyList.pdf`, and we are happy to provide the full data of the survey upon request. However, because data marketplaces are a very vivid field and change fast, it has to be pointed out that Kasabi went out of business since the survey was taken.

Surveying the current state of affairs in this field can be seen as the first step in analyzing and understanding this emerging market. We plan on repeating this study annually in order to gain further insight about what has changed, which competitors have been successful or not and why, which models and practices have proven themselves, etc. Researching the market and its developments can not only help understanding the market dynamics but also can give valuable insights into the emergence or application of new technologies and, thus, present new research opportunities.

The remainder of this paper is organized as follows: First, the survey approach will be described in Section 2. Then we present our findings, i. e., groupings and categorizations in Section 3. Section 4 gives an overview of related work that has been conducted in this area. The paper is concluded by summarizing our findings in Section 5.

2. METHODOLOGY AND APPROACH

In this section, we first elaborate on what we consider to be a data market or data vendor. Then we explain how the survey was conducted, using an iterative approach for both collecting data suppliers and deriving categories in Section 2.2. Section 2.3 discusses limitations of the method applied.

2.1 Data Marketplaces and Data Vendors

In the context of this work we have analyzed data vendors and data marketplaces. In order to restrict the potentially vast amount of companies, we have focused on companies offering either a platform for trading data (e. g., datamarket.com), raw data in any form (e. g., www.data.gov), or data enrichment tools (e. g., attensity.com). In order to gain a comparable set of data vendors, we have chosen to focus on vendors that offer online Web services. This implies that we have excluded offline products for data cleansing or data fusion and similar tasks.

We define a *data marketplace* as a platform on which anybody (or at least a great number of potentially registered clients) can upload and maintain data sets. Access to and use of the data is regulated through varying licensing models.

A data vendor has data and offers it to others, either for a given fee or free of charge. However, it is not important how vendors obtain this data, and many ways are common, e. g., aggregation from freely available sources, generation using proprietary methods, or buying from other vendors. It is important to note that a data vendor can offer its data either on its own or through a data marketplace as described above. Conversely, it is also possible that a data marketplace operator sells data and thus takes on the role of a vendor.

In our understanding, data marketplaces and data vendors have evolved from traditional Web crawlers and search engines as they all provide users with data. That is why we chose to also include crawlers and search engines that were comparable. Additionally, we also looked at data enrichment services that take input from the user and enhance it in some way, e. g., by analyzing or tagging it. Seeing how these services face the same data curation challenges as data marketplaces do, we allowed them into this survey.

2.2 Data Acquisition and Approach

The initial set of vendors consisted of well-known suppliers we found in previous research [14]. From this starting point, keywords were derived that were then used for a broader online search, which in turn revealed a more comprehensive set of different products and services.

We came up with a set of twelve dimensions along which the vendors considered can be categorized. As not all dimensions are measureable, and the dimensions are grouped into objective and subjective dimensions to clarify where our own opinion has influenced the results. Table 1 shows the dimensions that we used, the categories that constitute this dimension as well as the questions we asked to conduct this survey.

The values in our approach are strictly Boolean. An offering either fulfills the criteria for a certain dimension category or it does not. However, categories are not mutually exclusive in most cases. This means that, e. g., one offering can fall into multiple categories, have multiple pricing models, or provide multiple ways for data access. Some dimensions (e. g., maturity), however, are mutually exclusive. Where this is the case, it will be stated explicitly in the dimension description in Section 3.

The facts about the data vendors were gathered by means of a Web search. As every vendor or marketplace has a website, this publicly available information was used to determine how to categorize each vendor. After having done that with the initial set of vendors, it was checked how many entries a category had to justify its existence. When a category had only few entries, a new Web search for more data suppliers falling into that category was started in order to make sure no important vendors were omitted. If more companies were found, the list was extended iteratively, and the new companies were analyzed regarding the other dimensions. How-

Table 1: Set of dimensions.

	Dimension	Categories	Question to be answered
objective	Type	Web Crawler, Customizable Crawler, Search Engine, Pure Data Vendor, Complex Data Vendor, Matching Vendor, Enrichment Tagging, Enrichment Sentiment, Enrichment Analysis, Data Market Place	What is the type of the core offering?
	Time Frame	Static/Factual, Up To Date	Is the data static or real-time?
	Domain	All, Finance/Economy, Bio Medicine, Social Media, Geo Data, Address Data	What is the data about?
	Data Origin	Internet, Self-Generated, User, Community, Government, Authority	Where does the data come from? Who is the author?
	Pricing Model	Free, Freemium, Pay-Per-Use, Flat Rate	Is the offer free, pay-per-use or usable with a flat rate?
	Data Access	API, Download, Specialized Software, Web Interface	What technical means are offered to access the data?
	Data Output	XML, CSV/XLS, JSON, RDF, Report	In what way is the data formatted for the user?
	Language	English, German, More	What is the language of the website? Does it differ from the language of the data?
	Target Audience	Business, Customer	Towards whom is the product geared?
subjective	Trustworthiness	Low, Medium, High	How trustworthy is the vendor? Can the original data source be tracked or verified?
	Size of Vendor	Startup, Medium, Big, Global Player	How big is the vendor?
	Maturity	Research Project, Beta, Medium, High	Is the product still in beta or already established?

ever, if no more companies were found, the category definitions were reconsidered and updated.

2.3 Limitations

The information we used was taken directly from the website of each vendor. This may limit the accuracy of our findings in some cases, where the description of a product exceeds the actual functionality. Verifying that every product fulfills its own description is a task that goes beyond the purpose of this survey. Random samples, however, indicate that the descriptions commonly match the services provided. Nevertheless, there are also cases where the information provided on a vendor's website was not sufficient to categorize all dimensions. This was particularly the case for B2B vendors, which only reveal their pricing models upon request. We chose to leave these dimensions out than to speculate about their value. As a result, however, the numbers of these dimensions are minimally skewed.

The market of data vendors and data market places is highly active, i. e., new actors emerge and others disappear, and the market as such is growing rapidly. Therefore, it cannot be guaranteed that this study is fully exhaustive with regard to the number of vendors in the market. That said, we are confident that during our observation period from April to July 2012 we have obtained a representative sample that allows for a meaningful analysis. Furthermore, it has to be stated that data trading channels are not necessarily made public. This means that we are aware of the fact that a certain amount of data is traded directly between (large) corporations or within a certain ecosystem (such as social networks) *without* the use of intermediaries. It is obvious that it is impossible to investigate those forms of data trading using our Web survey approach.

3. FINDINGS

As stated in the previous section, the following twelve dimensions have been examined: *Type, Time Frame, Domain, Data Origin, Pricing Model, Data Access, Data Output, Language, Target Audience, Trustworthiness, Size of Vendor,* and *Maturity.* To structure these dimension we have categorized them into objective and subjective measures, i. e., whether the classification within each dimension can be easily verified or whether the classification is down to the researcher's judgement.

3.1 Objective Dimensions

3.1.1 Type

The first dimension *type* is used to classify vendors based on what their core product is. In order to form a common understanding of the different categories these are explained below:

- (Focused) Web Crawler: Services that are specifically designed to crawl a particular website or set of websites. These are always bound to one domain, e. g., spinn3r is a service that is specialized on indexing the blogosphere.
- Customizable Crawler: General purpose crawlers that can be set up by the customer to crawl

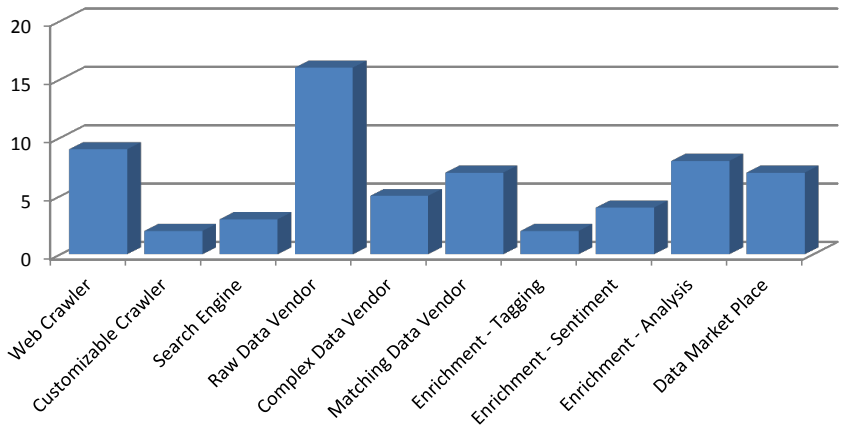

Figure 1: Number of vendors for each Type.

any website and search for arbitrary content. For example, 80legs offers such a service, in which customers can define regular expressions to crawl a set of sites.

- Search Engine: Services that offer their content via an interface similar to a search engine. Customers specify combinations of keywords as input and the search engine produces output relevant to that input. FactForge is such a search engine that represents an interface to the Linking Open Data cloud.

- Raw Data Vendor: This category comprises vendors that offer raw data, most often in the form of tables or lists. For example, Factual offers lists of restaurants, hotels, and other points of interest.

- Complex Data Vendor: These vendors offer data that is the result of some kind of analysis process. For example, The Stock Sonar provides information about current stock prices as well as indicators on how individual shares might develop in the near future.

- Matching Data Vendor: Vendors that offer the matching of input data against some other database. These vendors most often operate in domains where a customer does not want a complete dataset, but rather needs the data they already have corrected or verified, e. g., address data. Companies like AddressDoctor are specialized in this area.

- Enrichment – Tagging: This category describes services that enrich a given input (mostly text, but other forms are also possible) through means of tags. This enables customers to make more use of their data. Calais for example creates metadata for content submitted using natural language processing.

- Enrichment – Sentiment: With the proliferation of social media websites on the internet, a multitude of vendors has emerged that specialized on what is commonly referred to as sentiment analysis [15]. Given the name of a brand or a product, these services try to capture and analyze the sentiment of people towards that subject. This kind of service is, for example, offered by Salesforce under the name Radian6.

- Enrichment – Analysis: The data offered is enriched with analysis results obtained through various means, like comparisons with historical data or forecasts. Attensity Analyze is one of such services, offering customer analytics across multiple channels.

- Data Market Place: These services allow customers to both buy and sell data by providing the infrastructure needed for such transactions. A prime example for this type of vendor is Microsoft's Windows Azure Marketplace.

Figure 1 shows how many vendors fall into which category. It has to be kept in mind, though, that these categories are not mutually exclusive and one vendor can fulfill the criteria of multiple categories. Also, it should be noted that this histogram only shows a distribution over our sample and does not represent the entire market. This is owing to the fact that (as stated in the Section 2) we have intentionally excluded offline providers and tools.

3.1.2 Time Frame

The time frame dimension captures the temporal context of the data. We distinguish two categories in this dimension:

- Static/Factual: Data is valid and relevant for a long period of time and does not change abruptly, i. e., population numbers, geographical coordinates, etc.
- Up To Date: Data is important shortly after its creation and loses its relevance quickly, i. e., current stock prices, weather data, or social media entries.

As evident from Figure 2, we found that static data (32 offerings) was offered more often than up-to-date data (23 offerings). Some vendors offer data from both these categories. For example, Data.gov offers real-time data about worldwide earthquakes for the past 7 days as well as a dataset containing information on the total calories of commonly eaten foods. However, we found that only less than 20% (9 offerings) of the surveyed vendors offer both static and up to date information. This suggests that generally data vendors tend to specialize in either of the two options.

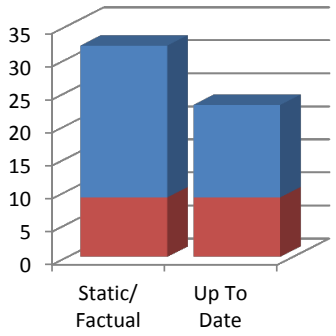

Figure 2: Number of vendors for Time Frame.

3.1.3 Domain

The dimension *domain* describes what the actual data is about. While most domain names are self-explanatory, domain *any* deserves clarification. This domain was used to classify vendors whose offers are not restricted and could incorporate arbitrary domains. For example, the Windows Azure Marketplace is not focused on a specific domain, which means that all different kinds of data can be found there. Whilst other domains were not mutually exclusive (i. e., a vendor could supply more than one domain), vendors serving any domain did not count towards explicit domains. The results are shown in Figure 3.

It is obvious that the *any* domain is by far the biggest group. An explanation for this is that data market places, search engines, and customizable crawlers do indeed serve any domain, depending on what customers choose to upload or search for. Given that they account for more than a fourth of all companies under investigation, the peak in *any* is not surprising. The other domains have a lower number of vendors, because they are more specialized. Furthermore, we have observed that the geo data (7) and address data (8) domains have a significant overlap (6), which can be explained by their obviously close relationship. Companies like AggData specialize in providing high-quality data about customers and their locations, so they fit into both categories. Address and geo data are, however, not the same, as evidenced for example by CustomLists.net, who offer only address data for marketing purposes.

3.1.4 Data Origin

The origin of data describes where it comes from. We have identified six different categories in this dimension:

- Internet: The data is pulled directly from a publicly and freely available online resource.
- Self-Generated: Vendors have means of generating data on their own, i. e., manual curation of a specific dataset or calculating forecasts based on patented methods.
- User: Users have to provide an input before they can obtain any data, i. e., address data offerings that return the address for a given name.
- Community: Based on a wiki-like principle, these vendors obtain and maintain their data in a very open fashion. The restrictions as to who can participate and contribute are usually rather low.
- Government: Governments capture and process huge amounts of data and have recently begun to make this data publicly available.
- Authority: Authorities in a domain are entities which are the main provider of data, i. e., the stock market for stock prices or the postal offices for address data.

In our survey the most popular origin category was the Internet. Almost 50% of all vendors receive their data from an online source. Another category

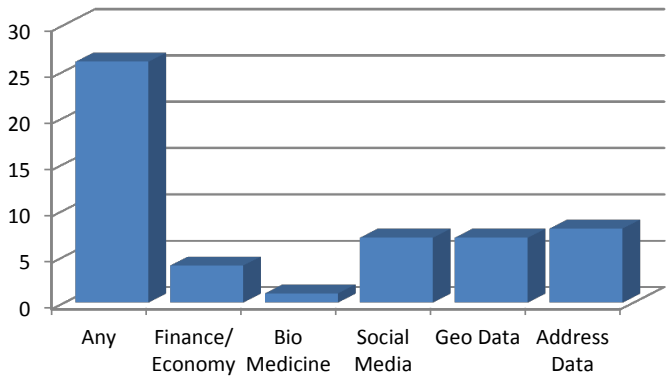

Figure 3: Number of vendors for each Domain.

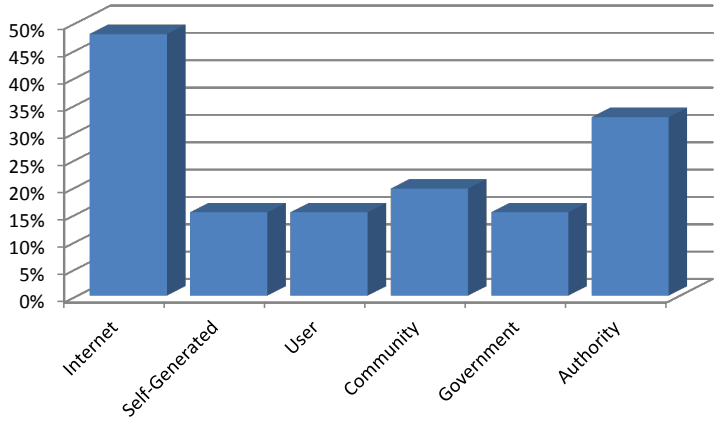

Figure 4: Data origin distribution.

with a large number of vendors was *authority*: 32% obtain their data from authoritative sources. For example, Intelligent Search Technology claims that their address verification service is certified by the U.S. Postal Service. The main advantage of these offers is that the data is usually of high correctness, completeness, and credibility. This also holds for the *government* category, into which fell 15% of vendors. The categories *self-generated* and *community* are matched by 15% and 19%, respectively. The problem with self-generated data is that there is no transparency in the data sourcing process. For example, CustomLists.net does not reveal where they get their data from, which might raise concerns regarding credibility or correctness. Lastly, category *user* with 15% is a special case because it cannot stand on its own, i. e., every vendor classified into this category also gathered data from another source. This is inherent to the definition of this category, according to which users submit their data and receive it back with additional annotations for which a vendor needs additional data sources. These facts are illustrated in Figure 4.

3.1.5 Pricing Model

Pricing models are very important to understanding how exactly the different vendors set up their business models. Four main pricing models could be found; the number of vendors for each model is illustrated in Figure 5. A verbal explanation of the pricing models is provided by the following list:

- Free: These services can be used at no charge. Reasons for offering a service for free are, among others, that it is only a beta test or research project, the vendor is a public authority funded by tax money, or simply interested in attracting more customers. For example, Data.gov is free as it is a website of the U.S. government. Vendors in this category do not count towards one of the following categories.
- Freemium: As a portmanteau combining free and premium, this pricing model offers a limited access at no cost with the possibility of an update to a fee-based premium access. Freemium models are always combined with at least one of the following two payment models.

- Pay-Per-Use: Customers are billed based on how much they use the respective service. This manifests mostly in the form of x$ per thousand API calls.
- Flat Rate: After paying a fixed amount of money, customers can make unlimited use of the service for a limited time, mostly a month or a year.

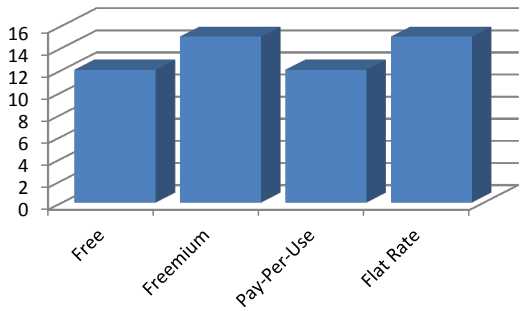

Figure 5: Number of vendors for each Pricing Model.

An example for the combination of the Freemium and Pay-Per-Use model is Factual.com. Their API may be called up 10,000 times per day for free. Any additional calls have to be paid for. The CloudMade Data Market Place, on the other hand, combines Freemium with Flat Rates by offering free trials for their datasets and unlimited access for an annual fee.

3.1.6 Data Access

The *data access* dimension describes through which means end-users receive their data from vendors. The main categories identified and presented in Figure 6 are:

- API: An API (application programming interface) is used to provide a language- and platform-independent programmatic access to data over the Internet.
- Download: Traditional download of files is the easiest way to access a data set, because anyone can use such a service with only a Web browser.
- Specialized Software: Some vendors have implemented a specialized software client to connect with their Web service. While this approach does have downsides (implementation and maintenance expense, dependency issues, etc.), there are some scenarios in which the concept is worthwhile, for example, providing the customer with an easy-to-use graphical user interface as an out-of-the-box solution that needs no further customization, or granting access to real-time streams of data.
- Web Interface: In a Web interface, the data is displayed to the customer directly on a website.

The flexibility and modularity of APIs have made these the most popular of all access methods. More than 70% of all vendors offer an API. However, less than 30% of all vendors have an API as their only way to access data. Most vendors offer an API next to other methods. For example, Web interfaces or file downloads are used to give previews of the dataset, to make it easier and more accessible for the customer to see what the actual data looks like, e. g., Factual.com has an extensive Web frontend that renders tables or geodata. The concept of specialized software does not seem to stand very well on its own. Out of all investigated vendors, only three use specialized software as the only way of data access. For example, MeaningMine provides the user with a dashboard-like interface that shows graphs and important numbers. However, this approach lacks flexibility, because customers are restricted in the way they can use the data by the functionality of the provided software. Nevertheless, most customers who want data do not want any restrictions on how they can access and process the data. From a theoretical point of view, it seems to be the best approach for a vendor to offer all the aforementioned means of access to his data, because that allows customers to choose their preferred way of access. However, we have not found a single vendor that does so, which is probably due to the high cost associated with creating such a broad offering.

3.1.7 Data Output

This dimension shows the format in which data can be obtained. To us, the most reasonable set of categories in this dimension is the following:

- XML: Being both human- and machine-readable, the Extensible Markup Language is a widely established standard for data transfer and representation.
- CSV/XLS: Most structured data is laid out in a tabular way, so it makes sense to wrap it into a table file format. We do not distinguish between CSV and XLS and other table file formats, because the main differences between them, like formatting and embedding, do not apply when you are showing raw data
- JSON: The JavaScript Object Notation is similar to XML and is also used as a data transfer

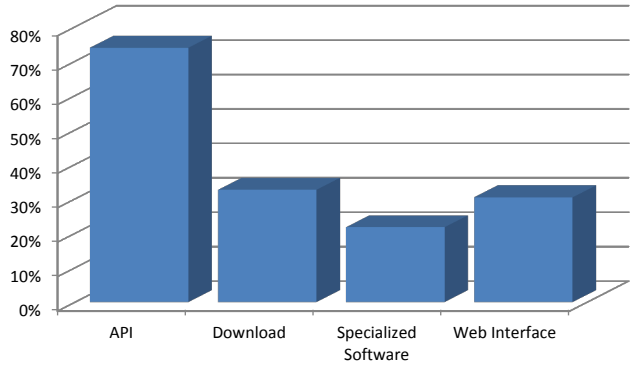

Figure 6: Data Access distribution.

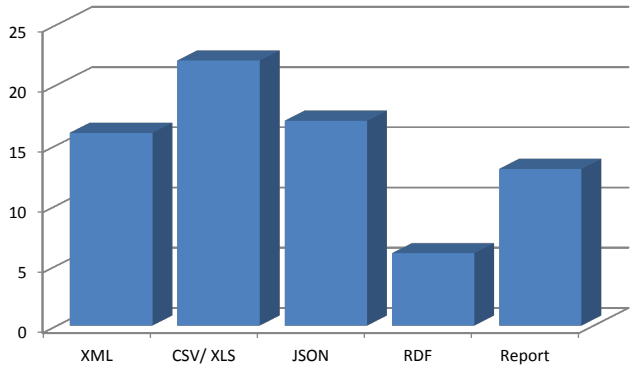

Figure 7: Number of vendors per Data Output category.

format. Data is represented as text in key-values pairs.

- RDF: The Resource Description Framework is a method to describe and model information. It uses subject-predicate-object triplets to make statements about resources. Due to its graph data model, it is a good choice for data that is inherently graph-shaped.

- Report: When data is preprocessed, aggregated and "prettified" in some way, we declared the output as a report. The main difference in this category is that the customer does not have insight into the underlying raw data. Also visual reports in the form of MS Excel spreadsheet classified for this category.

The most popular category in the output dimension shown in Figure 7 is CSV/XLS. With 22 vendors, almost half of all vendors considered offer the possibility to receive their data as a raw table. However, only six of those vendors have CSV/XLS as their only output format. Most vendors also offer either an XML (10) or a JSON (6) interface, some even both (3). This is consistent with the observation from the previous dimension, that an API is the most popular way of data access. An API usually produces XML or JSON output. Offering many ways to access data is a key feature of a data marketplace, because it broadens the range of possible users. DataMarket.com therefore supports all aforementioned output categories except RDF. Other competitors, however, do not provide all these different access mechanisms. The Infochimps Data Marketplace favors JSON over XML for their API. It remains to be seen what further implications this technical limitation may have.

3.1.8 Language

We have focused on the English and German languages because of personal language skills. Thus, further differentiations in this dimension were not possible. Therefore, any additional languages we encountered were aggregated into a third category called *more*. Although English is a dominant language on the Internet, we would be happy to cooperate with other researchers with other language skills in a future edition of the survey.

The analysis of language distinguishes between the language of the website and the language of the

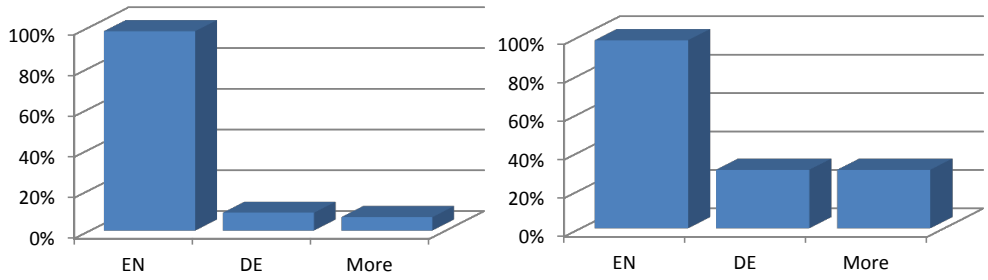

Figure 8: Language of websites (left) and data (right).

data offered. A visual representation of the results is shown in Figure 8. Nearly all investigated vendors (98%) run an English-language website. For the majority, English is also the only language available (89%). Only some companies run a multilingual website (9% German; 7% More). These tend to be the bigger player with a global strategy, like Microsoft or LexisNexis. This picture changes when looking at the language of the data itself. We observed that again 98% offered English Language Data, but about 30% offered German data and almost 20% of the vendors also offered data in other languages.

We have seen that English is the dominant language for both websites and data. This is not surprising because the market for data has a global scope and English seems to be the best suited language for that. However, there is also a demand for local data in the corresponding language, which is suggested by the amount of vendors that offer such data.

3.1.9 *Target Audience*

The last objective dimension is concerned with the target audience. Here, we have investigated towards whom offerings are tailored. As is evident from Figure 9, there are only two categories in this dimension, *business* and *customer*. Providing data for another company in a B2B fashion is the most logical application area of data vending. Specialized vendors focus on their respective domain, e. g., CustomLists.net targets business users while Wolfram Alpha is aimed more at private users. The more general vendors, especially those operating in the *any* domain like Kasabi or Windows Azure Marketplace, target their offer at all audiences. Out of all vendors in this research, 87% offered data in a business context, 41% sold data relevant for end consumers, and 28% had data that could be of use for both groups.

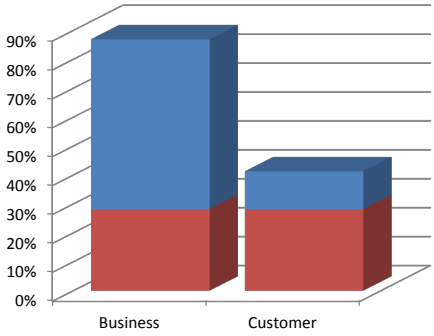

Figure 9: Number of vendors by Target Audience.

3.2 Subjective Dimensions

3.2.1 *Trustworthiness*

This dimension indicates how trustworthy the data of a vendor is, depending on the origin of the data as well as on how it is processed. For instance, data that come from a community could have a lower trustworthiness than data that is sourced from an authority. In other words, data from a postal operator as offered by, e. g., AddressDoctor is more likely to be correct than an aggregation of online sources. However, there are also other cases where a collective of anonymous authors produce data that is verifiably correct and therefore trustworthy, e. g., Wikipedia. Whether more trust is put in a single authority of a domain or in a crowd of people depends on the application context and one's personal attitude. Nevertheless, this dimension is not quantifiable and, thus, the results are subjectively biased.

As depicted in Figure 10, we have found that 54% of all vendors have a high trustworthiness. Among those are vendors that carefully select the data they offer in a transparent and comprehensible way. Also, authorities and governments as explained in Section 3.1.4 all exhibit a high trustworthiness. The category *medium* is populated by around 33% of all examined vendors. The main indicator for their

classification that they seem to be trustworthy is based on the descriptions, but this could not be verified in any way, e. g., because they do not explicitly state their data sources or explain their analytical methods. The lowest degree of trustworthiness applied to only 22% of all vendors. Typical vendors in this category are those that do not even claim to deliver correct or complete data, like web crawlers (e. g., 80legs) or community-supplied websites (e. g., Freebase).

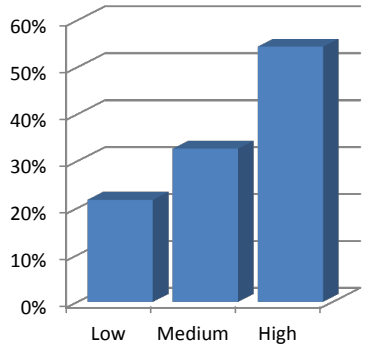

Figure 10: Trustworthiness distribution.

Note that the overlap between the three categories stems from the fact that one vendor can offer multiple datasets from different sources, like Kasabi or Infochimps. In such a case, we have assigned all possible levels of trustworthiness. Furthermore, while it is intuitive that high trustworthiness is good, it is not necessarily the case that a low trustworthiness is bad. There are scenarios in which incomplete data is sufficient for a rough estimation, or data with a high trustworthiness is not available (e. g., social media analysis). This leads us to the conclusion that vendors with all levels of trustworthiness are likely to co-exist in the future, because they fulfill different demands.

3.2.2 *Size of Vendor*

While some might argue that the size of a vendor is quantifiable (e. g., using the number of employees or its revenue), and thus, an objective dimension, it is difficult to find reliable figures that would support such an analysis. We therefore took the presentation of the offering as a foundation for a classification with the following four categories:

- Startup: Companies that are newly created and that have only a small number of people involved are usually referred to as startups; examples include Uberblic or QuantBench. These are often funded by investors, as they do not yet have a positive cash flow from the very beginning.
- Medium: Leaving the beta stage, gaining experience and maturity, and not being dependent on investors anymore are the key characteristics that set medium-sized companies apart from startups. Examples include eXelate or Spinn3r.
- Big: Companies that are well-established and have more than one product in their offering range are considered big, e. g., Infochimps or LexisNexis. While there is no sharp dividing line between medium-sized and big companies, we still felt that separating the two in different groups yields more accuracy for the analysis.
- Global Player: In this category fell only the biggest companies out there, like Yahoo!, Microsoft, IBM, etc.

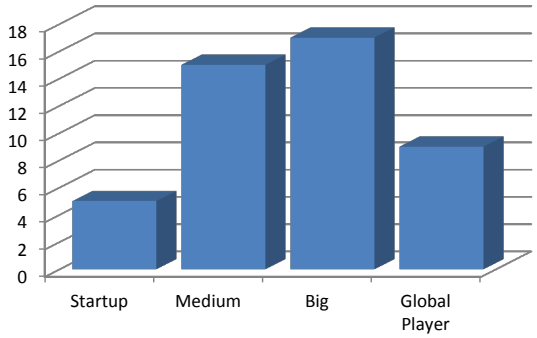

Figure 11: Number of vendors by size.

Note that in this dimension, the categories are mutually exclusive. Figure 11 shows the number of vendors for each size. It can be seen that the number of startups is the lowest. This could indicate that the market for data is not easy to enter. The number of global players also seems rather low, but it has to be kept in mind that these vendors have the potential to quickly seize huge market shares, because they usually have experienced people and extensive capital. The majority of vendors is either medium-sized or big.

3.2.3 *Maturity*

The maturity of all offerings has been classified into the following four categories, which are mutually exclusive:

- Research Project: These offerings are usually not for profit and can therefore be used free of charge. They are mainly executed as a proof-of-concept. Examples include Goolap or IBM Cognos Many Eyes.

- Beta: A beta product is still in development and has not been fully launched yet. Nevertheless, we have also seen offerings in beta phase that already demanded a usage-fee, like Semantics3.
- Medium: This category classified products that were already out of beta, but were still not as highly developed as other products, such as BuzzData or CloudMade Data Market Place.
- High: Full-fledged products that implement all intended features and are ready for use in an operational environment. For example, the Windows Azure Marketplace seems to be relatively advanced in this sense.

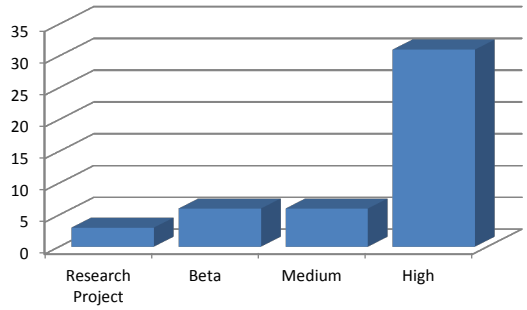

Figure 12: Maturity of vendors.

Evaluating the numbers presented in Figure 12 shows that only 3 research projects, 6 betas and 6 medium-matured offerings could be identified. The remaining 31 offerings can all be classified as having a high maturity. This observation can also serve as an explanation to the previous finding of a low number of startups. When there are already established vendors with mature projects, the space for new companies to enter the market is relatively small.

4. RELATED WORK

In [6] a general discussion of data services can be found. Starting from a general data service architecture, the authors examine concepts and example products for *service-enabling data stores, integrated data services,* and *cloud data services.* Further, they highlight technical challenges, such as transactions an updates to data structures underlying the service, as well as predict emerging trends, such as convergence and cloud integration.

Ge et al. [8] studied electronic marketplaces but restricted themselves to Web sites where users can ask questions (e. g., Askjeeves.com), which are then answered by other users or experts. Furthermore, they only described five websites and focused rather on business models than on surveying marketplace properties.

Regarding data markets as we defined them in Section 2.1 surveys have only been done on a (much) smaller scale, not disclosing any methodology, and only in textual form. For instance, Strata [7] describe characteristics of the four (according to them) most mature data markets Factual, Infochimps, DataMarket, and Windows Azure Data Marketplace, which we also examined in this study.

Similarly, Miller interviewed 10 providers of data marketplaces or data related services in a series of Podcasts [10]. However, he only provides the interviews in a rather unprocessed form, i. e., as audio files, which makes it difficult to access and aggregate the contained information. Later, he published a report [11] on data marketplaces and their business models, in which he identified common functionalities that data marketplaces offer, elaborated on potential business models and makes some rather general predictions such as increasing competition and a wider choice of data and sources.

Furthermore, there have been investigations into particular market places, for instance on Kasabi [12], which went out of business in the meantime. It was described as a "web-based information marketplace" and stored data using the Resource Description Framework (RDF) with the goal of bridging the gap between data publishers and application developers by providing a platform that allows hosting of and searching for data. It was designed after the linked data paradigm originally outlined by Tim Berners-Lee. The basic idea of linked data is to publish data in a structured way that allows for linkage to data sets. An overview of this concept, the technical principles and its applications can be found in [4]. A survey about the current usage of these dataset is given by [13] and actual trends are outlined in [3].

In the course of the Linked Open Data (LOD) movement, FactForge emerged as a publicly available service that is meant to "provide an easy point of entry for would-be consumers of Linked Data" [2]. It was built with the intention to facilitate access to the LOD cloud of data by integrating the major datasets into one view.

A different approach is pursued by the developers of Freebase. They try to create what they call a "collaboratively created graph database for structuring human knowledge" [5]. The collaboration aspect is inspired by Wikipedia and based on the idea that data quality improves when lots of people refine datasets. They employ a graph database, because it depends less on a rigid schema and is more flexible.

The authors state explicitly that they want to allow conflicting and contradictory types and properties to exist simultaneously in order to "reflect users' differing opinions and understanding" [5].

DBPedia is a different project that shares many similarities with Freebase. They both aim at extracting structured data and making it available in RDF. However, DBPedia focuses on Wikipedia as its only source, and also does not allow direct editing of data.

Microsoft's contribution to the market is Windows Azure Marketplace [9] and has been launched in 2010. It is designed to make the sharing of data as well as applications an easy process for both consumers and providers of data. The key features are global reach through a central platform, unified billing and access mechanism, high data quality, and easy integration with other Microsoft products. Unique to Windows Azure Marketplace is the way in which datasets and applications are combined. Providers of data can go beyond selling their raw data, and bundle it with applications that are designed specifically for this dataset. Customers can purchase these bundles directly and have a working out-of-the-box solution without any additional implementation effort.

That said, there is—to our knowledge—no survey that investigates data marketplaces in such a comprehensive manner as we have done.

5. CONCLUSION & FUTURE WORK

In this study we have presented an initial overview of data vendors and marketplaces for data. Utilizing an iterative approach we have derived dimensions along which data providers can be classified and grouped. We have then presented a survey drawing a preliminary picture of the current data vendor landscape. Our survey gives an overview of the current market situation and shows which categories are currently underrepresented and which ones can be particularly interesting for practitioners. However, it is too early to make reliable statements about where data marketplaces are heading. That is why we plan on repeating this survey on an annual basis to re-evaluate the individual vendors and extending the study with a development section. We believe that a comparison over time will allow for assessing which models and practices stand the test of time. Also, technical trends can then be deduced from market observations and give valuable insights to researchers.

This study has focused on the *provider* view of data marketplaces, which have emerged because it has by now been recognized that and how data can be monetized. It will also be interesting to observe *buyers* of data and analyze their perception of these new offerings, where a distinction between private and professional customers is likely to be appropriate. Here, it will over time be possible to determine who is spending money for what kind of data, and we expect that certain domains will be more attractive than others. For example, a variety of current activities in the healthcare domain (e.g., taltioni.fi, ensembl.org, patientslikeme.com, or cancerresearchuk.org) indicates a high attractiveness for data markets. This will be a subject of future research.

Acknowledgment

We like to thank the anonymous reviewers of the ACM SIGMOD Record for their helpful comments.

6. REFERENCES

[1] A.A. Armstrong and E.H. Durfee. Mixing and memory: emergent cooperation in an information marketplace. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 34 –41, jul 1998.

[2] B. Bishop, A. Kiryakov, D. Ognyanov, I. Peikov, Z. Tashev, and R. Velkov. FactForge: a fast track to the web of data. *Semantic Web*, 2(2):157–166, April 2011.

[3] C. Bizer. The Emerging Web of Linked Data. *IEEE Intelligent Systems*, 24(5):87–92, 2009.

[4] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–2, 2009.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008. ACM.

[6] Michael J. Carey, Nicola Onose, and Michalis Petropoulos. Data services. *Commun. ACM*, 55(6):86–97, June 2012.

[7] Edd Dumbill, 2012. `http://strata.oreilly.com/2012/03/data-markets-survey.html`.

[8] W. Ge, M. Rothenberger, and E. Chen. A Model for an Electronic Information Marketplace. *Australasian Journal of Information Systems*, 13(1), 2005.

[9] Microsoft White Paper. Windows Azure Marketplace, 2011. `http://go.microsoft.com/fwlink/?LinkID=201129&clcid=0x409`.

[10] Paul Miller, 2012. `http://cloudofdata.com/category/podcast/data-market-chat/`.

[11] Paul Miller, 2012. `http://pro.gigaom.com/2012/08/data-markets-in-search-of-new-business-models/`.

[12] K. Möller and L. Dodds. The Kasabi Information Marketplace. In *21nd World Wide Web Conference, Lyon, France*, 2012.

[13] K. Möller, M. Hausenblas, R. Cyganiak, and S. Handschuh. Learning from Linked Open Data Usage: Patterns & Metrics. In *Web Science Conference*, 2010.

[14] A. Muschalle, F. Stahl, Löser, and G. Vossen. Pricing Approaches for Data Markets. In *to appear in 6th International Workshop on Business Intelligence for the Real Time Enterprise (BIRTE)*, 2012.

[15] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

Hank Korth Speaks Out on two-career issues, why not to write a book in the beginning of your career, and more

by Marianne Winslett and Vanessa Braganholo

Hank Korth
http://www.cse.lehigh.edu/~korth

Welcome to ACM SIGMOD Record Series of Interviews with Distinguished Members of the Database Community. I'm Marianne Winslett, and today we are in Providence, site of the 2009 SIGMOD/PODS conference. I have here with me Hank Korth, who is the Wieseman Professor and Chair of the Department of Computer Science and Engineering at Lehigh University[1]*. Before joining Lehigh, Hank was the Director of Database Principles Research at Bell Labs, a vice president of Panasonic Technologies, a professor at the University of Texas at Austin, and a research staff member at IBM TJ Watson. Hank's research interests lie in database systems, especially transaction processing. Hank is on the editorial board of ACM TODS. He is an ACM Fellow and an IEEE Fellow. His PhD is from Princeton. So Hank, welcome!*

[1] Hank Korth served as Department chair in Lehigh from 2003 to 2009.

During your time at Bell Labs, the company underwent a lot of surgery. What was it like working there?

I started at Bell Labs about a month before AT&T had, what they called, trivestiture. This is where AT&T split themselves up into a new AT&T, not the current one, but a new one then, what became Lucent, and NCR. So, there I was, a new employee at Bell Labs, wondering what is this all about. "Is this an excuse to get rid of research? What's going to happen?" It took a while to work it out. As we all know now, AT&T had its own research lab, AT&T Labs, which is still there, alive and well. Lucent inherited the Bell Labs name that still continues, although today in a much different form. But then came the telecom boom, when Bell Labs was a truly wonderful place to be in research.

So, you were there during that golden period?

Exactly. It turned out to be the perfect time.

At Bell Labs, you were involved with technology transfer. It is hard for many big companies to get their research results out into products. So, how did tech transfer work at Bell Labs?

Historically, tech transfer at Bell Labs was not very successful. For one thing, in the monopoly days, it was less necessary. Lucent being a profit making company, it became more important. But I really did not get involved in tech transfer initially. We were initially involved in building a new research center under Avi Silberschatz. I built, with Avi's substantial help, a very nice database group. Then, a bit later on, Lucent, flushed with the telecom boom, bought at telecommunication billing company, called Kenan Systems.

The founder of that company, who then came into Lucent to head the software products group to be affiliated with Bell Labs, designed a special organization whose goal was to take mature research projects in Bell Labs, and put a wrapper around them consisting of a development team as well as a business team. A separate organization housed this until first customer introduction. Then the whole package, not just the technology, but now a product, a marketing team, customers, the whole works, would be transferred into the business. I was on the leadership team of that organization, which put me in a very interesting spot, because I was a department head at Bell Labs, plus involved in this. We got a number of very nice ideas productized, that started off wonderfully, but of course, then the telecommunication bust hit us, and we never really got to bring that whole model completely to fruition. But it was a great time, and I think a very good model for tech transfer, because what you are able to do then is isolate the group that is trying to make the technology something real; treat it as something special in the company, so you can attract the most dynamic and interesting people to the group. Then for the true tech transfer, you now have the full organization in place, and people who appreciate the value of the technology, but also appreciate the value of the business end of the technology. I think where most companies fail in tech transfer is that you get technologists who think marketing is trivial, and you get marketers with a superficial understanding of the technology. You wind up with a horrible mismatch, and people unhappy on both sides.

Your database textbook with Avi Silberschatz and Sudarshan, Database System Concepts, is about to have its sixth edition, which to me sounds enormous!

It is. If I think about all the words that we've written over the years, I wonder if I would have started. But, all kidding aside there, the initial idea, way back in the early '80s, was that there was a gap between very practical database books and very theoretical ones. Avi Silberschatz and I (Sudarshan wasn't in the picture at that point) were looking to fit somewhere in between. I did that as an assistant professor, something that I certainly would not recommend to new faculty getting started. But I did that because I had a unique opportunity. Silberschatz had a successful operating systems book, still successful, by the way, *Operating System Concepts*. He wanted to do a database book, and asked if I was interested. I did not think I would necessarily get the opportunity again to co-author with an experienced and successful person like Avi, and he is just a great person to work with. We had already started working on some research together. And so I decided "okay, if I don't get tenure, I don't get tenure, but here's this opportunity, I'm going to take it". So despite the book, I did manage to get tenure, but I really have to emphasize that's not the best strategy for that. Books do not count in the computer science tenure process. Not in any place that I know of.

What about your time in Panasonic? Panasonic isn't a big name in the database community. What made you go there?

Well, I was at the University of Texas at the time. I had tenure, my wife finished her PhD, so we had a two-career issue to solve. Obviously, there were lots of possibilities, but the intersection of possibilities in computer science, and possibilities in the pharmaceutical industry, aren't exactly that large. The New Jersey and Philadelphia area is wonderful for the pharmaceutical business, and it turned out that there was this great opportunity. Well, Panasonic wasn't exactly a household name in computing, but this was a time, in the early '90's, where the Japanese economy had been booming, and the major Japanese companies were looking to have a research presence in the United States. And so, some visionaries within Panasonic had the idea of setting up an information technology lab. It was located in Princeton, it was headed by Dick Lipton, and among the people involved were some people from the database community: Hector Garcia-Molina, Rafael Alonso, in particular; and from the OS community, Kai Li. It was a really wonderful group. Then we added a number of others in databases – there was Daniel Barbará, as well as myself.

Some years from now […] there will be nothing on paper, and that can change the way we do reviews, the way we evaluate publications, and the way we evaluate people for promotion.

This looked like an opportunity to combine the kinds of things we were doing in database research with consumer electronics. One of the things that we paid a lot of attention to was the concept of mobile computing, which was very new then! In fact, some of the things that we

talked about were clearly way too early. I remember Rafael Alonso talking about having the web in the palm of your hand. Okay, but why would anybody want the web in the palm of your hand? Well, it is now a joke, but that we were thinking about those issues back then, and this would have been in the range of 1991-94, made it a truly wonderful time. Wonderful until the Japanese economy went down, and the economics of having these labs began to make less sense for Panasonic. So the lab got redirected towards a more year term applied set of objectives, which then led to many of us moving to other places -- in my case, it was Bell Labs.

With Francois Bancilhon and Won Kim, you wrote a 1985 VLDB paper called "A Model of CAD Transactions". That paper won a 1995 VLDB award as the most influential paper from the proceedings of ten years ago. What was that paper about?

At the time we were working on this in 1985, there was a lot of concern about how to manage databases that contained not traditional data processing data, but design data. The issue there is that, if you are working on a design, you don't simply read a bank balance, add an amount, and put it back into the database. Instead, you read some part of the design, and work on it for some period of time, which could be a few minutes, but also could be a few days. Then you check it back into the database. Transactions in that realm are long duration transactions. And so, we need to have different ideas of how to manage concurrency, and recovery, and define what a transaction will be in that process. So, we were thinking about issues clearly related to long-duration nested transactions, and other ideas along those lines. They were all happening at the same time. Those were the ideas we talked about in that paper.

You've been the department chair at Lehigh for a long time. What have been your goals there?

I came to Lehigh as Bell Labs was becoming a less interesting place to be than it had been, because of the changes going on with the telecom bust, which was followed by the dot com bust, and it looked like a good time to move back to academia. Lehigh was an interesting challenge. The computer science department there (technically Computer Science and Engineering) was formed in 2001. So, I was coming in, basically, at the beginning, and while there were some faculty members there, it was a time to try to mold the department into something that could be a good research department, while maintaining Lehigh's traditional strong focus on undergraduate education. I did quite a bit of hiring, which seems in some ways to have been the story of my life. I did that for the database group at Bell Labs. Then at Lehigh I had to look much more broadly than databases, to have a department with sufficient breadth to be able to cover computer science from a curricular point of view, but also to get groups of people with sufficiently similar research interests, that we have a strong research activity.

I am now done being chair. Today is my second day as not being chair anymore, but in the six and a half years I hired more than half the faculty in the department, and I am really pleased with the results. We have four NSF CAREER award winners in the department, and a department with 16 tenure/tenure track faculty. I think that is quite good. Also, I am very pleased with my first hire, Dan Lopresti, who had been at Panasonic with me, he was at Bell Labs, and he's going to be my successor now as Chair. I know he is going to do a wonderful job. So here the challenges were, reaching broadly, not just in areas of computer science that I know well, but in areas that I perhaps don't know as well as I should. But still, I needed to try to lead the

department in bringing the right people together, and also to try to maintain the sense of community that I found when I was there. That was one of the things that attracted me to Lehigh. It was a wonderful group of people. There was none of the factions that you see in so many departments, but rather a real strong sense that we all want to create a good department. And we are still that way. It has been very happy place to be.

You said that the focus is on undergraduate education. Did you have time to do research?

Anybody who is department chair, you know, there's advice that you get that of teaching, research, and administration – one of them has to go. If the department chair lets administration go, that can be problematic. Because I was coming from industry, I thought it was very important that I do some teaching, both on the graduate and undergraduate level, so that I had a sense of what the Lehigh community was about. We were looking to strengthen our graduate programs, strengthen our research, but I also wanted to understand the Lehigh undergraduate traditions. One of the things that I ended up teaching was intro, which I'll admit, was a tremendous amount of work, but I have a lot of fun doing it. I do get a kick out of students taking their interactive webpage and sending it home to mom by email, and just feeling so proud of themselves, that they've done that.

If you enjoy what you are doing, you are going to be much more productive, and much more

So, that is how you introduce computer science, through webpages, and Java Script, or something?

Well, that's the beginning. The main focus of the course is sufficient Java programming to get onto the second course, as we do our undergraduate curriculum initially in Java, and then switch to C and C++. But, part of the course is to talk about the breadth of computer science that is really a whole lot more than programming. It is much more about design, creativity, and interactivity. All the students have web experience coming in, maybe not web development, but certainly web usage. And when they can actually create something that they can click on, and their browser opens up and runs their code, they love it.

Are there textbooks that take this Intro to CS via let's start at the web?

There are a number of books that do that, some better than others. But we also, of course, need a Java book. So, I pick one of these general books, and then a Java book that we continue to use for the second course. We use both of those, but I tend to do things more my way than any book's way, in part because my audience is different than the typical intro course.

How are they different?

Our engineering students have to take the engineering curriculum, and don't take my course. So I have our arts and science students, and students who are computer science and business majors.

The Computer Science and Business program is an accredited CS degree, and business degree altogether in one. The only major like that anywhere. I have a lot of students that have some degree of business interest, and I have some experience with that (we've talked about my work with technology transfer). So, I try to bring some of that perspective into the class as well. But that's in no textbook.

The VLDB conference recently backed away from double-blind reviewing, while ACM Transactions on Database Systems (TODS) recently took it up. What do you think about double-blind reviewing?

It's far from perfect, and it is very easy to complain about it. But I think the fundamental thing that drove the discussion in TODS is that there are real problems in terms of perceptions of unfairness, and maybe some actual unfairness in the process that we were using, where everything was totally open to the reviewers. Even if it is just the perception, I think it is important to try to be as fair and as open as possible in the process. I think TODS has put together a very nice compromise there, because it is not absolute blind reviewing – when some unblinding becomes necessary in the process, the associate editors have some discretion to try to do the right thing. So I think it works well there. In a conference setting, I think it is more problematic, because there is a deadline by which the decisions have to be reached, and that creates a little bit more pressure in the review process.

Well, you guys have a deadline too, because you do this fast turn-around, faster than conferences, in fact.

True, but the deadline we are talking about there is a 5 month deadline from submission to decision. Conferences have a long turn-around time in part because there is the whole delay between the decisions on the paper, and when the conference actually happens. So, the reviewers have the paper in hand for a longer time in the TODS process than they do in a conference process. And that gives us the ability to do a careful, thorough review to I think a much greater degree than a conference can do.

What do you think should be the purpose of our conferences?

Conferences traditionally have been a way to get ideas out, get them out quickly, and be interactive. Workshops now seem to be taking on that role more than conferences. In part, I think that's because we have neglected journals as a discipline, and as a result, conferences became the primary way that we measure people for tenure and promotion in the field. So conferences have, therefore, necessarily become ever more strictly refereed, ever more competitive, and as a result, I think, the tone of conferences has changed a bit, and not necessarily for the better. We may want, at some point, to rethink our whole publication process, and recognize that some years from now, and probably not that many years from now, we're going to have all of our publications electronic: journal, conference proceedings. There will be nothing on paper, and that can change the way we do reviews, the way we evaluate publications, and the way we evaluate people for promotion. And then, I'm hoping that conferences can take on more of the flavor of a workshop: more interactivity, more half-baked ideas, half of which are no good, but the ones that

are good will then be much more valuable and much more stimulative than statements of things that are already done, packaged, finished, and presented.

"Being fully electronic is likely to change the nature of how we evaluate people". Why does being electronic change that?

The first thing it changes is how we publish. Right now, our publications are structured by the clock, because conferences occur at a certain time, and that drives that process. Journals come out in print in certain intervals, and have limits. There is a limit on how many pages we can put in a journal. We have that in TODS. Any journal I've been involved with has some kind of limit. In the electronic domain, there is no need for page limits, except as mandated by quality and readability. There is also no limit on the number of papers we can have, or when we can have them. There is no need to have, let's say, four publications of a journal per year, or one instance of a conference per year. These things can happen whenever something is good to put in there. And there can be some push notification for people who want to subscribe: "there's a new TODS paper, go look at it". Having that kind of dynamic mode of publication, we can now define the venues that we want to treat as the ones that count in terms of tenure and promotion. And that can be done independently of how we disseminate ideas quickly. So there are new possibilities there. I think it would be worthwhile for the community to think about what the end point ought to be, and then to back up from that, and to say how do we get from where we are to where we are going.

As a community, are our experimental results meaningful, believable, statistically valid, and repeatable?

[…] don't write a book, but I did. Try to stay focused on an area and establish yourself, but I didn't.

I am sure many of them are, but certainly, not all. But not always for, necessarily, bad reasons. If I look at some of the work that I've done lately, looking at architecture-aware algorithms for multicore, and cache awareness, when you move from one platform to another, the results are going to be remarkably different. If you simulate, rather than do it on a real machine, the results are different. Not necessarily wrong, but different. I think what SIGMOD is trying to do with repeatability is a good step in that direction. I would also like to see us focus more carefully on proper statistical methodology in our results. Sometimes it seems that we run an experiment, we like the results, and we write the paper. As opposed to stepping back and saying "Did we do a good experimental design?"; "Are there other experiments we can do?"; "Are our results truly statistically valid?". Having a spouse in the pharmaceutical business and seeing what they have to go through to convince the FDA to approve a drug, and the types of studies they have to do, we are not at their standards. Perhaps we don't have to be. But I would like to see a better delineation between careful statistical studies than some sense of definitive pieces of work, and perhaps, more cursory experimental work, that is there to stimulate new ideas. The way we have things structured now, where our papers seem to have an experimental section that has to be there, whether there really needs to be one or not, we're trying to be somewhere in between, and I think uncomfortably so.

You mentioned multicore architectures. Is that something that you've been working on?

I would like to see us focus more carefully on proper statistical methodology in our results. Sometimes it seems that we run an experiment, we like the results, and we write the paper.

Well, earlier, we were talking about being chair, and we went off on my discussion of teaching. One of the things that I've done less of in the past six and a half years as chair is research. I'd been very focused on that up to that point. I put a lot of effort into department building, and into the undergraduate curriculum. But having great undergraduates at Lehigh, I managed to do some work with an undergraduate on multicore architectures and their impact on database algorithms. That was a very nice experience. The student stayed on to do a Master's, and then went on to do PhD work at the University of Wisconsin, Madison. I've gotten really interested in how the changing computer architecture is going to have to change the way we think about database systems. We seem to be at a point where rethinking the engine could truly be the right thing to do. We talked about that in the Claremont Report, we're switching now towards at least thinking and maybe going to a column-oriented approach rather than a row-oriented approach. We have to think about parallelism… We have transactions, and if I'm going to be running thousands of transactions, I've got all the parallelism that I need. But to have sufficiently little conflict among transactions to get that parallelism, you then run into trouble with cache footprint. And so we clearly do have to start thinking about parallelizing the engine at various levels. I think that going forward, that is going to be a truly critical thing for us to address.

For our younger readers, can you explain about the cache footprint issue?

In the old days, all we cared about was disk and main memory. If I brought data from disk into main memory, I could do whatever I wanted with it, and it was effectively free, because disk was so slow. Main memory now looks like disk to a processor, or worse, because processors are so fast, and memory is relatively slow. And so, processors have multiple levels of higher speed storage called cache. The idea is that just as we brought data in from disk to memory for efficiency purposes, and tried to optimize that, we now need to think about how we optimize bringing data from memory into these caches. We have a similar issue, although, there are differences in it. While a database system has control over what it brings into memory, the hardware controls how cache is managed. So in some sense, the database system is pushing on a string, and doing cache management requires a different kind of cleverness. As a result, it requires a bit more heuristics, it is a bit less repeatable, but clearly, it is something we will have to think about. But it's not just cache. It's also the fact that we all have core duos, so we have a couple processors. We are going to be having hundreds of processors, and on the server's side, it is probably going to be many hundreds of processors. So we have this dua problem of keeping all

the many cores busy, but yet not requiring so much data that we wind up having cache misses, and now our cores are no longer busy, but simply sitting there waiting on cache misses.

Some people have argued that column stores are good for better L2 cache utilization, so would you agree with that?

I think that is a distinct possibility. I haven't seen the actual data yet to support that, but intuitively, what you want to be able to do is not have to bring in large rows. Most of which contain information that you either don't need at all, or don't need for this particular operation. Since caches are relatively small, you don't want to pollute them with stuff that you don't need, because it won't be there anymore by the time you may actually need it.

You commuted to Bell Labs from Pennsylvania for a long time. How long was your commute?

It was only 70 miles. I had mentioned earlier the two career issues, so my wife works west of Philadelphia, we live just north of Philadelphia, and when I went to Panasonic, that was about 45 miles. Bell Labs was longer, but I was able to telecommute once in a while. It was a wonderful commute: beautiful drive, a covered bridge, something like one-and-a-half-lane highways, deer, woods, streams. On the other hand, when you want to get back and forth, it does take a lot of time.

How long?

I did it once in an hour and twenty minutes. I don't think you would have wanted to have been a passenger on that ride. But I always treated it like a 2 hour commute from a scheduling point of view, so I wouldn't feel pressure to have to be there in a hurry and perhaps take unnecessary chances.

That meant your whole day was 3 hours shorter… Unless, did you find something useful you could do during that long drive?

Well, I'm supposed to say the nice intellectual things… ESPN radio and rock and roll. Sorry, I can't be inspiring there! What I did by trying to telecommute a couple of days a week was that I amortized my commute time over more days. For a lot of what I was doing, telecommuting was as effective as being there. While working with my database group, we sent email to the person in the next office! So, I could be home just as well. Also, the various business units of Lucent are all over New Jersey and elsewhere as well. We had interacted with people in the Chicago area, and some people in Britain for a while, and so we did a lot by teleconference, which I could do just fine from home. So I tried to schedule things strategically. Of course, when I moved to Lehigh, and then became department chair, I did need to be there every day.

How far away is that?

It is 35 miles. So I'd say I cut my commute in half, but it is not entirely fair because I have a little bit more traffic to deal with. I managed to make all that work. It is one of the compromises of the home and career balance, one that I think, overall, I'm very happy with.

You have two kids, or two teens. Are they following in your footsteps?

Thankfully, no. They have never had strong interest in computing. I'm not entirely sure why, although I love a story back from the preschool days, where the preschool teacher asked "what do your parents do?". "Mommy works with computers, and goes to meetings". Okay, well true, she does statistical analyses for a drug company. "Daddy goes to meetings". I knew something was wrong at that point. So, my daughter, while not being interested in computing, is very interested in mathematics. I was an undergrad math major at Williams, so you could say that's in my footsteps, but I've not played up by being a math major. Whether she will go into mathematics for a career or not remains to be seen, but she is strongly interested in math, and the sciences. So, in some sense, yes, but differently[2].

Do you have any words of advice for fledgling or midcareer database researchers?

I mentioned earlier, don't write a book, but I did. Try to stay focused on an area and establish yourself, but I didn't. I worked in transactions, I worked in relational database theory, I worked in object-oriented databases. So a lot of the traditional advice, I've not followed. And so I think, giving advice that I can be more credible giving, would be, take advantage of the opportunities that are there. They are going to be different for each individual. Opportunities don't necessarily come back, and they don't necessarily show up at the optimal time or when you want them. And so, take advantage of it, if it hurts you in some ways, it's going to help you in others. And enjoy what you're doing. If you enjoy what you are doing, you are going to be much more productive, and much more successful.

Among all your past research, do you have a favorite piece of work?

Oh, that's a tough one, because I have worked in so many different areas! I guess if I have to pick one thing, I gotta pick my one JACM paper, even though that goes way back to a 1983 issue of JACM.

What was that paper about?

That paper was on my work on transaction processing where I took the type of multi-granularity locking used in System R, and extended it to have an arbitrary set of fundamental operations, instead of just read and write. Something that I think is still relevant in many ways, and may actually become more relevant if things like transactional memory start to come into being in a wide-spread way, where you can truly have multiple levels of atomicity, and then looking down at the lower levels, those are like harder operations.

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

Go back to playing ultimate Frisbee with the students again, like I did at the University of Texas.

[2] She's graduating in May 2013 from Harvey Mudd with a degree in engineering.

It's not too late, now that you're not department head, you can go out with those undergrads.

I don't have the knees for it anymore. My kids outrun me. My son outruns me by about a factor of two. So, I think I've reached a certain age.

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

I think I would find a way to give myself more time to write code, and build real systems. I started off on the theoretical side, working under Jeff Ullman. Jim Gray taught me about database systems as systems. He is certainly one of my greatest mentors there. I have always been somewhere between theory and systems, bouncing back and forth between them. I really wish I could have had more time to get involved in system building, and actually be deeply involved in the actual construction of a major system.

I remember Rafael Alonso talking about having the web in the palm of your hand. But why would anybody want the web in the palm of your hand? Well, it is now a joke...

Do you mean sort of on a Dewitt model?

Well, that's a bit of an extreme, perhaps. But that is what I am referring to, yes.

In one of these interviews, he said that his students won't let him touch the code[3]*! So you think these people are out there coding up all this stuff, but maybe they actually aren't.*

Maybe not, but certainly I think I have been less hands on personally than I would enjoy. The reason I went into computer science originally was that I enjoyed programming. Then I found out that it looks more like mathematics. Well, that's okay, I was a math major, I can go with that too.

Kind of funny that way, isn't it?

I have been looking for the balance ever since.

Well, thank you very much for talking with me today.

Thank you, it was a pleasure talking to you.

[3] David DeWitt speaks out: on rethinking the CS curriculum, why the database community should be proud, why query optimization doesn't work, how supercomputing funding is sometimes very poorly spent, how he's not a good coder and isn't smart enough to do DB theory, and more. SIGMOD RECORD 31(2): 50-62, 2002.

Data-based Research at IIT Bombay

Soumen Chakrabarti Ganesh Ramakrishnan
Krithi Ramamritham Sunita Sarawagi S. Sudarshan
Indian Institute of Technology Bombay
Mumbai 400076 India
{soumen,sunita,ganesh,krithi,sudarsha}@cse.iitb.ac.in

1. OVERVIEW

The Indian Institute of Technology (IIT) Bombay has a history of research and development in the area of databases, dating back to the early 1980s. D. B. Phatak and N. L. Sarda were among the first faculty members at IIT Bombay to work in the area of database systems. This was a period when the financial sector of India, headquartered primarily in Bombay (now renamed Mumbai) saw a spurt in computerization, and IIT Bombay faculty played a leading role as consultants for database implementations in these companies. Research in the area of databases began in the early 1980s, but increased greatly from the early 1990s, with the hiring of several faculty including S. Seshadri, S. Sudarshan, and later Krithi Ramamritham, who moved to IIT Bombay from U. Mass. Amherst in the early to mid 1990s. With the hiring of Sunita Sarawagi and Soumen Chakrabarti in the late 1990s, there was a significant broadening, with the group no longer being just a database group, but rather a much broader data management group, with interests in information retrieval, and data mining. More recently Ganesh Ramakrishnan joined our group, further increasing its strengths in information retrieval and data mining.

The number of PhD students increased from around 1 or 2 enrolled at a time in the early 1990s, to about 12 to 15 students at a time in recent years. While this number is much better than earlier, and is increasing rapidly, it is still small by most standards. However, our master's and bachelor's students have compensated for the shortage of PhD students, and have made very significant contributions to our research efforts, with well over three fourths of our papers having such students as coauthors.

Today, the group covers a diverse range of interests, which you can see from the different research projects showcased in this article. In the following sections, we outline the major research projects of the group. We wrap up the article with a summary of other contributions to the community, by group members.

For more information about the group, please visit:
http://www.cse.iitb.ac.in/infolab

2. RANKING IN GRAPH DATA MODELS

Graph data models are ubiquitous in semistructured search. Modeling a data graph as an electrical network, or equivalently, as a Markovian "random surfer" process, is widely used in applications that need to characterize some notion of graph proximity. Edges in most data graphs have rich semantics. The probability of a random surfer exiting a node through an outbound edge should be influenced by the semantics of the edge. In applications, relative edge conductances used to be tuned by trial and error, from domain knowledge. We gave several formulations [2, 1] to automatically learn relative edge conductances from pairwise preferences between nodes.

Random walks in weighted graphs is a form of personalized PageRank. In applications, the edge conductances are known offline, but the teleport vector is a function of the query. For large graphs, a global PageRank computation in response to every new teleport is prohibitive. We [13] proposed a query workload-driven framework to carefully choose a small number of basis nodes where certain indices are built, so that provably correct top-k PageRank nodes can be reported within a time that, in practice, remained essentially *constant* even as the graph scaled substantially.

3. LARGE-SCALE WEB ENTITY ANNOTATION AND INDEXING

A recent study by a search company revealed that up to 40% of Web queries pertain to a named entity. Search engines have been steadily supplementing the "ten blue links" with structured knowledge about entities and relationships. Toward this end, we undertook to annotate token spans in a large scale Web corpus (500 million pages) with disambiguated mentions of any Wikipedia entity (at that time, numbering between 2 and 3 million). Note that we are not interested in determining the broad type of an entity, such as person, place, time or event. We are interested in pinpointing the exact entity that has been mentioned, in the face of aliases ("John Smith") and other ambiguity ("apple").

We proposed [31] and demonstrated that collective disambiguation of mentions on a page can improve accuracy. E.g., "Michael Jordan" on a page by itself is more often about the basketball player than the Berkeley professor, and "Stuart Russell" is often the actor, not the Berkeley professor. But a document that mentions *both* "Michael Jordan" and "Stuart Russell" is almost certainly describing the Berkeley professors.

Another challenge was the design of practical data structures for fast disambiguation. Part of the statistical disambiguation model above consists of a map from a key consisting of a phrase ID p (where a phrase may be "Michael Jordan" or "big Apple"), a candidate entity ID e (the basketball player, the Berkeley professor, New York City, Apple Computer Corporation — a standard and unique URN in Wikipedia), and a context feature f (e.g., the words *variational*, *league*, or *iOS*), to a value w that is a trained model weight. The number of keys is large but the key space (p, e, f) is exceedingly sparse: each of p, e, f runs into tens of millions of distinct values. In our modest 500-million page, 2-million entity prototype, there were 700 million (p, e, f) keys, which would have cost us over 20GB of RAM using conventional Java hash maps. Lossy maps as used in signed or hash kernels showed considerably poorer accuracy than our disambiguator with lossless model weights. We designed a new workload-driven lossless compression scheme [12] that traded space for access time, cutting the $(p, e, f) \rightarrow w$ map down to only 1.2GB of RAM. At the same time, our annotator was orders of magnitude faster than other public annotators.

4. CONSENSUS-BASED QUANTITY AND ENTITY SEARCH

While commerce verticals serve plenty of quantitative information in response to Web queries, they do not cover many quantity information needs. E.g., What is the typical battery life on an iPad while playing games? What is the driving time between Mumbai and Pune? How far should I relocate a raccoon? What is the typical production budget of French art movies? Unlike record extraction to precise schema, snippets that may carry the desired information are matched very noisily, and extracting the quantity (with unit) is also nontrivial. If we plot the snippet match score against the candidate quantity embedded in the snippet, the poor reliability of snippet scores shows up as substantial vertical spread of both relevant and irrelevant snippets. However, a new signal comes to our rescue [5]: if there is palpable consensus on the answer, the relevant snippets appear in narrow vertical bands in the 2d plot. We designed new, practical quantity-ranking functions for this class of queries, and their accuracy was substantially better than prior art.

Similar issues of detecting consensus arise in regular entity search as well. Based on our 500 million-page Web corpus with over 8 billion indexed entity annotations, we have built an entity search engine [14] based on discriminative learning-to-rank models. Our system has a semi-structured query model where types from Wikipedia or YAGO can be specified, e.g., find entities belonging to `WikiCategory: Austrian_Physicist` who played the violin. However, users will generally not be familiar with the hundreds of thousands of types registered in Wikipedia, YAGO, or Freebase. They will ask the usual "telegraphic" queries (e.g., `austria physicist violin`) and expect that the search engine will divine their intention. We proposed [44] generative and discriminative approaches to simultaneously explore multiple interpretations of the query and score response entities, effectively aggregating over these interpretations to rank entities. The resulting ranking accuracy is significantly better than trying to interpret the target type first and then execute a structured query.

5. KEYWORD QUERIES ON STRUCTURED DATA

In recent years there has been a good deal of research in the area of keyword search on structured and semi-structured data, including our own work on the BANKS system [6, 29]. Most of this body of work has a significant limitation in the context of enterprise data, since it ignores the application code that has often been carefully designed to present data in a meaningful fashion to users. In recent work [40], we considered how to perform keyword search on enterprise applications. Such applications provide a number of forms that can take parameters; parameters may be explicit, or implicit such as the identifier of the user. In the context of such applications, the goal of keyword search is as follows: given a set of keywords, retrieve forms along with corresponding parameter values, such that result of each retrieved form executed on the corresponding retrieved parameter values will contain the specified keywords. For example, suppose a university ERP system includes a form that takes a roll number as parameter, and returns information about the student with that roll number. Given a keyword query "Krithi CS" our system would return (amongst other results) the above form, along with roll numbers for which the form result includes the keywords "Krithi" and "CS".

Some earlier work in this area was based on creating keyword indices on form results, but there are problems in maintaining such indices in the face of updates. In contrast, in our work we introduced techniques based on creating "inverted SQL queries" from the SQL queries in the forms. Unlike earlier work, our techniques do not require any special purpose indices, and instead make use of standard text indices supported by most database systems. We have implemented our techniques and show that keyword search can run at reasonable speeds even on large databases with a significant number of forms.

We have also been extending the BANKS system to support querying on annotated textual data; in this context, an annotation marks a particular word or phrase (e.g. Gandhi) with the entity that it (likely) refers to (e.g. the Indian leader M. K. Gandhi). Queries on such annotated textual data can search for entities (entity queries) or entities which satisfy specified relationships (entity-relationship queries). We have built a prototype that works on Wikipedia data (described in [4]), and are currently extending it to work on annotated Web crawl data.

6. WORLD WIDE TABLES (WWT)

The Web today comprises of billion of semi-structured objects such as tables and lists that have been universally accepted as an idiom for expressing relational data even for human consumption. Usually, these are considerably higher quality than completely unstructured free-format text. The goal of the WWT project is to explore methods to exploit tables and lists on the Web for various query-driven structure extraction tasks. We have explored the following challenges in the context of this project.

Extracting structure from lists.

The user poses a query by providing a few seed multi-attribute rows in a table, and expects as answers more rows of the same type. We assemble such tables on-the-fly from the few seed rows by aligning, segmenting, and consolidating information from raw lists on the Web. We deploy statistical methods such as Semi-Markov Conditional Random Fields (CRFs) [42] for this task. Our setup differs from conventional deployment of Semi-CRFs in two ways: First, we do not possess any explicitly labeled data for training purposes. Our only supervision is in terms of a few seed structured records. In [24] we report how to carefully label unstructured list items using the seed record set. Second, we need to train multiple extractors, one for each list source. We exploit the fact that the lists often enjoy partial content overlap to jointly train the model so as to ensure agreement in the labels of overlapping content [25].

Answering Column Keywords Queries.

The user poses a query consisting of keywords describing each column in a table he expects to see in the answer: example "university" and "motto". WWT returns a multi-column table in response to such queries by exploiting the millions of existing tables on the Web. We represented this task as a graphical model that jointly maps all tables by incorporating diverse sources of clues spanning matches in different parts of the table, corpus-wide co-occurrence statistics, and content overlap across table columns. We defined a novel query segmentation model for matching keywords to table columns, and a robust mechanism of exploiting content overlap across table columns. We designed an efficient inference algorithms based on bipartite matching and constrained graph cuts to solve the joint labeling task. More details of this work can be found in [37].

Annotating Web tables to an Ontology.

Most web tables are not organized on any formal, uniform schema; consequently Web search cannot take advantage of these high-quality sources of relational information. In the presence of a popular On-

tology, such as Wikipedia and its derivatives like Yago, we enrich raw Web tables by linking them with the Ontology. We developed new machine learning techniques to annotate table cells with entities that they likely mention, table columns with types from which entities are drawn for cells in the column, and relations that pairs of table columns seek to express. We proposed [32] a new graphical model for making all these labeling decisions for each table simultaneously, rather than make separate local decisions for entities, types and relations. These annotations are invaluable for retrieving tables based on column keyword queries and integrating information across isolated tables.

Current status of the project.

The WWT system currently runs over a crawl of 36 million tables and lists extracted from an offline crawl of 0.5 billion Web pages. The project was started in 2008 and after five years it is nearing completion. It provided us an exciting platform to carry out research on large-scale information extraction and integration while grappling with the noise and redundancy that is so prevelant over Web data. More information is available at the project page at `http://www.cse.iitb.ac.in/ sunita/wwt`.

7. AGGREGATING IMPRECISE DATA

The goal of this project is to develop statistical learning models for extracting precise aggregate statistics over sets of instances. We highlight three diverse facets for this task:

Aggregating Unstructured Crowd Data.

There is increasing interest in tapping the wisdom of the crowd for a wide variety of information needs. Our focus is on extracting reliable aggregate information from a set of unstructured textual inputs provided by the crowd. For example, given a set of user comments on an article, identify the fraction of supporters of the articles. Existing method are based on aggregating per-instance predictions from a classifier. Given the inherent inaccuracy of learning models, the question we ask is: can we obtain more accurate summaries through alternative paradigms? We have developed a method based on Kernel Mean Matching that provides more accurate estimates, that are also very efficient to deploy in practice.

Estimating classifier accuracy.

A practical problem facing many industrial deployments of imprecise prediction models is calibrating the accuracy of the model on large unlabeled data. Let $C(\mathbf{x})$ denote a model and D the unlabeled dataset. A conventional method of evaluating the accuracy of $C(\mathbf{x})$ on D is to manually select a labeled set L out of D, invoke $C(\mathbf{x})$ on each instance in L and aggregate the individual errors to calculate the accuracy of $C(\mathbf{x})$. When D is large in comparison to L, the accuracy measured this way is unlikely to be a reliable estimate of the classifier's real performance on the deployment data D. We propose [30] a method based on stratified sampling for estimating accuracy and select instances for labeling in a loop. For stratifying data we develop a novel strategy of learning r bit hash functions to preserve similarity in accuracy values.

Top-K count queries over duplicates.

Suppose we are interested in finding the K most frequently mentioned entities in a dataset containing many noisily duplicated entities. We show how to dedup on the fly only the part of the data actually needed for the answer — a requirement in massive and rapidly evolving sources where batch deduplication is not feasible. We propose a novel method of successively collapsing and pruning records which yield an order of magnitude reduction in running time compared to deduplicating the entire data first. We also show how to return multiple high scoring answers to handle situations where it is impossible to resolve if two records are indeed duplicates of each other [43].

8. STATISTICAL RELATIONAL LEARNING

The emerging area of statistical relational learning (SRL) is characterised by a number of distinct strands of research. Especially prominent is research concerned with the construction of parametric and non-parametric models from data that consist of multiple relations. To a first approximation, this is the kind of data that can be stored in multiple tables of a relational database, although it can get more complicated than this. Interest in this form of modelling is driven by at least two different trends: (1) The data are no longer simply values of known (pre-defined) features, but are in the form of observations of several inter-related variables. In such situations, it is often impractical to pre-define all kinds of features that may be of interest; and human expertise may not be available to define new sets of features for each new situation and modelling task. (2) Data is now available in very large quantities, largely due to advances in automation and the low-cost of secondary storage.

Research in SRL is concerned principally with different ways of representing the relational information; techniques of combining these representations with the calculus of probability; estimation of parameters of distributions and inference to yield probabilities with model predictions. The field is currently at an early stage, and shows great potential for the modelling of complex systems.

In the older, but related, field of Inductive Logic Programming (ILP), there has been substantial work of an engineering flavour specifically aimed at combining relational and statistical modelling that may be directly applicable to the kinds of data described here. This research consists of the use of ILP systems—programs that are capable of learning relations in first-order logic— as a tool for discovering useful features for subsequent use by any standard statistical model.The case for this form of data analysis is that the discovery of relational features must necessarily require some form of first-order learning, of which ILP systems are an instance.

Arguments in-principle aside, there are several reports in the literature that augmenting any existing features with ILP-discovered relational features can substantially improve the predictive power of a statistical model. While this approach is simple, and there appears to be experimental evidence to suggest it is effective, much still needs to be done to scale these up to meet modern data and model requirements. This includes the abilities to discover features using very large datasets stored in secondary memory; from relational data arriving in a streaming manner; and from data which do not conform easily to expected patterns. The SRL research at IIT Bombay is concerned with scaling-up relational feature-discovery methods to an industrial strength. Specifically, research undertaken are in the following areas:

Conceptual. The purpose here is to investigate conceptual ways in which relational structures can be discovered efficiently and effectively from extremely large search spaces. By this we mean: constraints that can be imposed on features and structures without serious loss of expressive power; transformations of the feature-discovery problem to other tasks for which efficient algorithms are known; optimisation formulations that can be solved efficiently, learning and inferencing with structured output spaces and so on. We have pursued the following strategies: (a) pose the problem as a discrete optimisation problem and solve it heuristically [28, 15, 48, 39, 49], (b) pose the problem as a continuous (often convex) opti-

misation problem with sparsity inducing regularizers and solves it optimally [27, 36] and (c) study restrictions on the space of relational features and investigate empirically whether it is acceptable for a relational learner to examine a more restricted space of features than that actually necessary for the full statistical model [41, 35, 33] We have also looked at heuristics for speeding up inference algorithms in relational settings [34].

Application. The purpose here is to investigate the applicability of feature-based techniques to data analysis tasks of constructing discriminatory and generative models for problems such as information extraction and disambiguation [33, 15, 48, 39, 49, 50].

9. EXECUTING CONTINUOUS QUERIES OVER DATA AGGREGATORS

Queries on web data, very often, involve distributed data sources. If the data items are dynamic, query results need to be refreshed continuously to avoid the risk of results becoming stale. We have developed techniques for executing continuous aggregation queries over data aggregators with the minimum number of refresh messages between the data sources [19], the aggregators and the client [20], leading to significant improvement in the utilization of network and computational resources.

In continuous query applications, usually the user is not interested in all the updates: a user either specifies an incoherency bound, where the user can tolerate some bounded inaccuracy in the query results; or a selection condition (e.g., threshold), such that the user is interested only if the query value satisfies the condition. Data aggregators can pull the data values from data sources or the data sources can push the values to an aggregator. Different data aggregators can execute either different sets of queries; or divide the queries such that different aggregators execute sub-queries of the individual queries [45]. Further, the aggregation functions used in the query can either be linear functions like SUM and AVG, or non-linear functions such as ratio, MAX, non-linear polynomial [21], etc. Although there exist various point solutions to the problem, as far as we know, our work is the first to cover all the dimensions [23].

As an example of the cases where threshold values are used to limit the number of refresh messages, consider ratio threshold queries (RTQs) where the user is interested in knowing whether the ratio of two aggregations has crossed a user specified threshold [22]. Such queries are executed by assigning conditions for individual data sources so that the data sources refresh the data values only if the assigned conditions are violated. We assign these source conditions such that no violation of the client threshold condition is missed while minimizing the number of message exchanges between the data sources and the data aggregator. Using performance evaluation we have shown that our method of assigning the source conditions results in up to an order of magnitude fewer refresh messages compared to the methods proposed in literature.

10. EXECUTING CATEGORY BASED QUERIES OVER DYNAMIC DATA

In order to harness the information present in unstructured dynamic data such as blog posts, forum postings, etc. we have been developing specialized techniques which given a keyword query, find the top-K categories related to the query. Example categories returned by such a category based search system for a keyword query say '911' would include: Information about September 11 attacks, Information about emergency services in US, Information about cars (Porsche 911), Information about 900 AD, etc. Thus category based search systems are able to provide a macroscopic view of the information present in the data [7].

There are multiple dimensions to the problem. The first is the nature of the query execution. In certain domains where users want to track the evolving information present in the dynamic data, users would want to be updated continuously of the changes in the top-K categories for their keyword queries. For such domains, we continuously report the top-K categories for the user query [9, 10]. On the other hand, in domains where the user wants to perform 'data exploration' a user would be satisfied with point queries where the top-K categories are reported only at the time when the query is executed [8]. Another dimension is the nature of the accuracy requirement. Users may not be interested in finding the exact answers to their top-K keyword queries but may be satisfied with say 80% accurate results. In others, it might be imperative that the system provides answers without any infidelity. Based on the above dimensions, we have built and evaluated systems that tackle the following types of queries over dynamic data: Category Based Exact Point Queries, Category Based Exact Continuous Queries, Category Based Incoherency Bounded Continuous Queries and Category Based Incoherency Bounded Point Queries. This work is one of the first to explore the problems associated with executing category based queries over dynamic data. We have evaluated them using real world data which show the superior performance of our techniques as compared to known alternatives.

11. REAL TIME TOPIC DETECTION OVER UNSTRUCTURED DATA STREAMS

Existing techniques for discovering emerging topics from a microblog stream in real time (such as Twitter trending topics), have several lacunae; extant graph based event detection techniques are not practical in microblog settings due to their complexity; and conventional techniques, which have been developed for blogs, webpages, etc., involving the use of keyword search, are only useful for finding information about known events. Our techniques discover topics that are unraveling in microblog message streams in real time so that such topics can be reported as soon as they occur[3].

Let S_i represent a set of keywords (potentially spread over multiple messages) from a unique user *i* in a time window that spans from time $(t - \delta.w)$ to current time *t*, where δ represents unit time called *quantum* and $\delta.w$ is the length of the time window. Let S_w^t=$\{S_1 \ldots S_m\}$ be a set of keywords sent by *m* unique users in the microblog stream in a given time window. S_w^t contains the messages from a sliding window (of size $\delta.w$) over the message stream. Time unit δ denotes the fixed rate at which the window is moved forward. We represent all the keywords, after removing stop words, appearing in the messages in the current window as nodes in an undirected graph Correlated Keyword Graph (CKG). CKG is a dynamic graph whose state at time *t*, is $G^t = (V^t, E^t)$ where V^t is the subset of keywords appearing in message set S_w^t. Thus, two keywords are said to be temporally correlated *iff* they appear in V^t and are said to be spatially correlated if they have an edge between them in E^t. An edge links two keywords *iff* they both appear in a keyword set S_i belonging to a user *i*.

Hence, in [3], we model the problem as discovering dense clusters in highly dynamic graph CKG. Half-quasi cliques are considered as clusters of interest as this leads to good precision and recall of discovered events. In order to find clusters, we propose and exploit a novel graph property which we call *short-cycle property*. Further we present a novel ranking function to identify the important events. We show that globally consistent ranking of events can be achieved by exploiting local properties of clusters.

12. HOLISTIC QUERY OPTIMIZATION

Queries, or calls to stored procedures/user-defined functions are often invoked multiple times, either from within a loop in an application program, or from the where/select clause of an outer query. When the invoked query or function involves database access, a naive implementation can result in very poor performance, due to random I/O. Query decorrelation addresses this problem in the special case of nested sub-queries, but is not applicable otherwise. This problem is traditionally addressed by manually rewriting the application to make it set-oriented, by creating a batch of parameters, and by rewriting the query/procedure to work on the batch instead of one parameter at a time. Such manual rewriting is time-consuming and error prone.

We have been working on an approach which we call holistic query optimization, which combines program analysis and rewriting of imperative application programs, with the rewriting of database queries, with the goal of optimizing the database accesses of an application. Our early work in this area (Guravannavar and Sudarshan [26]) introduced a technique for program analysis and rewriting to automatically replace queries inside loops by batched versions of the queries. In [18], we explored how to achieve similar effects using asynchronous query submission, which is useful when batching cannot be done, e.g. for Web service calls. We have built a tool called DBridge for holistic optimization of Java programs using the JDBC framework [17]. In Ramachandra and Sudarshan [38] we explored to how introduce query result prefetching into complex procedure call sequences, with minimal program rewriting. Such prefetching could be very useful for optimizing programs written using Object-Relational Mapping frameworks such as Hibernate. Performance studies of our system show significant reduction in cost due to our techniques.

13. XDATA: GENERATING TEST DATA TO KILL SQL QUERY MUTATIONS

Complex SQL queries are widely used today, but it is difficult to check if a complex query has been written correctly. Formal verification based on comparing a specification with an implementation is not applicable, since SQL queries are essentially a specification without any implementation. Queries are usually checked by running them on sample datasets and checking that the correct result is returned; there is no guarantee that all possible errors are detected.

In this project, we address the problem of test data generation for checking correctness of SQL queries, based on the query mutation approach for modeling errors. In this approach, errors are modeled as small changes ("mutations") to an intended query, and the goal is to generate test data that can distinguish between the original query and the mutant, i.e, kill the mutant, In [46], we focus on test data generation to kill a particular on a class of join/outer-join mutations, comparison operator mutations, and aggregation operation mutations, which are a common cause of error. To minimize human effort in testing, our techniques generate a test suite containing small and intuitive test datasets. The number of datasets generated, is linear in the size of the query, although the number of mutations in the class we consider is exponential. Under certain assumptions on constraints and query constructs, the test suite we generate is complete for a subclass of mutations that we define, i.e., it kills all non-equivalent mutations in this subclass. We have subsequently extended this work to handle more SQL constructs and types of mutations; we note that completeness in terms of killing mutants is not guaranteed for some of the extensions, but the tool is useful for testing, even without guaranteeing completeness.

We are currently building a tool to help in grading student SQL assignments [16] which works as follows. The tool first generates test datasets from a correct sample answer to an SQL assignment; for each student query, the tool compares the result of the student query with the result of the correct query, on each of these datasets. A mismatch on any of the dataset is evidence of an error in the student query. We ran a large number of actual student SQL queries through our system, and found that our system outperformed both TAs who had corrected the assignments manually, and two sample datasets which had been used to check for correctness of the queries. We believe such a tool could be of great value to instructors of database courses, and plan to open-source the tool.

14. CONTRIBUTIONS TO THE COMMUNITY

In addition to the above research-oriented activities, our efforts have resulted in several artifacts, of potential benefit to the community at large:

- S. Sarawagi's software for segmenting and cleaning Indian addresses, deployed in banks and other financial institutions.
- Conditional Random Fields based information extraction package, developed by S. Sarawagi and placed in open source[1].
- Source code for S. Sudarshan's PYRO multi-query optimizer and the BANKS system for keyword search are available on request.
- In the spirit of Weka, G. Ramakrishnan has developed a relational learning (RL) workbench called BET (Background + Examples = Theories) implemented in Java[2]. The objective of BET is to shorten the learning curve of users (including novices) and to facilitate speedy development of new RL systems as well as quick integration of existing ILP systems.
- K. Ramamritham and his colleagues have developed and deployed *almost All QUestions Answered* (aAQUA [3]), a multilingual, multimedia agricultural portal that lets rural farmers and agribusiness employees ask and answer questions online. The system incorporates novel database systems and information retrieval techniques, including intelligent caching and offline access with intermittent synchronization.
- S. Sudarshan has coauthored a textbook on database systems, which is widely used all over the world [47].
- Chakrabarti's book, Mining the Web [11], published in 2002, continues to be among the standard texts used in graduate courses worldwide on Web crawling, indexing, search and hyperlink analysis. A second edition is in preparation.

15. REFERENCES

[1] A. Agarwal and S. Chakrabarti. Learning random walks to rank nodes in graphs. In *ICML*, 2007.

[2] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *SIGKDD Conf.*, pages 14–23, 2006.

[3] M. K. Agarwal, K. Ramamritham, and M. Bhide. Real time discovery of dense clusters in highly dynamic graphs: Identifying real world events in highly dynamic environments. In *VLDB'12: Proc. of 38th Intl. Conf. on Very Large Data Bases*, 2012.

[1] http://crf.sf.net

[2] http://www.cse.iitb.ac.in/~bet/

[3] http://www.aaqua.org

[4] A. Agrawal, S. Sudarshan, A. Sahoo, A. Sandalwala, and P. Jaiswal. Entity ranking and relationship queries using an extended graph model. In *Intl. Conf. on Management of Data (COMAD)*, 2012.

[5] S. Banerjee, S. Chakrabarti, and G. Ramakrishnan. Learning to rank for quantity consensus queries. In *SIGIR Conf.*, 2009.

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *ICDE*, 2002.

[7] M. .Bhide, P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, and P. Shenoy. Adaptive push pull: Disseminating dynamic web data. *IEEE Transactions on Computers*, 51:652–668, 2002.

[8] M. Bhide, K.Ramamritham, and P.Roy. Keyword search over dynamic categorized information. In *Intl. Conf. on Data Engineering*, 2009.

[9] M. Bhide and K. Ramamritham. Category based infidelity bounded queries over unstructured data streams. In *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[10] M. Bhide, K. Ramamritham, and M. Agrawal. Efficient execution of continuous incoherency bounded queries over multi-source streaming data. In *Intl. Conf. on Distributed Computing Systems*, 2007.

[11] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kauffman, 2002.

[12] S. Chakrabarti, S. Kasturi, B. Balakrishnan, G. Ramakrishnan, and R. Saraf. Compressed data structures for annotated web search. In *WWW Conf.*, pages 121–130, 2012.

[13] S. Chakrabarti, A. Pathak, and M. Gupta. Index design and query processing for graph conductance search. *VLDB Journal*, 2010.

[14] S. Chakrabarti, D. Sane, and G. Ramakrishnan. Web-scale entity-relation search architecture (poster). In *WWW Conf.*, pages 21–22, 2011.

[15] A. Chalamalla, S. Negi, L. V. Subramaniam, and G. Ramakrishnan. Identification of class specific discourse patterns. In *CIKM*, pages 1193–1202, 2008.

[16] B. Chandra, B. Chawda, S. Shah, and S. Sudarshan. Extending XData to kill SQL query mutants in the wild. Unpublished manuscript, 2012.

[17] M. Chavan, R. Guravannavar, K. Ramachandra, and S. Sudarshan. DBridge: A program rewrite tool for set-oriented query execution (demo). In *ICDE*, pages 1284–1287, 2011.

[18] M. Chavan, R. Guravannavar, K. Ramachandra, and S. Sudarshan. Program transformations for asynchronous query submission. In *ICDE*, pages 375–386, 2011.

[19] R. Gupta, A. Puri, and K. Ramamritham. Executing incoherency bounded continuous queries at web data aggregators. In *WWW '05: Proc. of the 16th Intl. Conf. on World Wide Web*, Chiba, Japan, 2005.

[20] R. Gupta and K. Ramamritham. Optimized query planning of continuous aggregation queries in dynamic data dissemination networks. In *WWW '07: Proc. of the 16th Intl. Conf. on World Wide Web*, pages 321–330, Banff, Alberta, Canada, 2007.

[21] R. Gupta and K. Ramamritham. Optimized query planning of continuous aggregation queries over a network of data aggregators. In *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[22] R. Gupta, K. Ramamritham, and M. Mohania. Executing ratio threshold queries over distributed data sources. In *ICDE '10: Proc. of 26th IEEE Intl. Conf. on Data Engineering*, 2010.

[23] R. Gupta, K. Ramamritham. Scalable Execution of Continuous Aggregation Queries over Web Data. In *IEEE Internet Computing* 16(1): 43-51 2012.

[24] R. Gupta and S. Sarawagi. Answering table augmentation queries from unstructured lists on the web. In *PVLDB*, 2009.

[25] R. Gupta and S. Sarawagi. Joint training for open-domain extraction on the web: Exploiting overlap when supervision is limited. In *WSDM*, 2011.

[26] R. Guravannavar and S. Sudarshan. Rewriting procedures for batched bindings. *PVLDB*, 1(1):1107–1123, 2008.

[27] P. Jawanpuria, J. S. Nath, and G. Ramakrishnan. Efficient rule ensemble learning using hierarchical kernels. In *ICML*, pages 161–168, 2011.

[28] S. Joshi, G. Ramakrishnan, and A. Srinivasan. Feature construction using theory-guided sampling and randomised search. In *ILP*, pages 140–157, 2008.

[29] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, 2005.

[30] N. Katariya, A. Iyer, and S. Sarawagi. Active evaluation of classifiers on large datasets. In *ICDM (***Runner-up for Best paper award***)*, 2012.

[31] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in Web text. In *SIGKDD Conf.*, pages 457–466, 2009.

[32] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. In *VLDB*, 2010.

[33] A. Nagesh, G. Ramakrishnan, L. Chiticariu, R. Krishnamurthy, A. Dharkar, and P. Bhattacharyya. Towards efficient named-entity rule induction for customizability. In *EMNLP-CoNLL*, pages 128–138, 2012.

[34] N. Nair, A. Govindan, C. Jayaraman, K. TVS, and G. Ramakrishnan. Pruning search space for weighted first order horn clause satisfiability. In *ILP*, 2010.

[35] N. Nair, A. Nagesh, and G. Ramakrishnan. Probing the space of optimal markov logic networks for sequence labeling. In *ILP*, 2012.

[36] N. Nair, A. Saha, G. Ramakrishnan, and S. Krishnaswamy. Rule ensemble learning using hierarchical kernels in structured output spaces. In *AAAI*, 2012.

[37] R. Pimplikar and S. Sarawagi. Answering table queries on the web using column keywords. In *Proc. of the 36th Int'l Conf. on Very Large Databases (VLDB)*, 2012.

[38] K. Ramachandra and S. Sudarshan. Holistic optimization by prefetching query results. In *SIGMOD*, pages 133–144, 2012.

[39] G. Ramakrishnan, S. Joshi, S. Balakrishnan, and A. Srinivasan. Using ilp to construct features for information extraction from semi-structured text. In *ILP*, pages 211–224, 2007.

[40] A. Ramesh, S. Sudarshan, P. Joshi, and M. N. Gaonkar. Keyword search on form results. *VLDB J.*, 22(1):99–123, 2013.

[41] A. Saha, A. Srinivasan, and G. Ramakrishnan. What kinds of relational features are useful for statistical learning? In *ILP*, 2012.

[42] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *NIPS*, 2004.

[43] S. Sarawagi, V. S. Deshpande, and S. Kasliwal. Efficient top-k count queries over imprecise duplicates. In *EDBT*, 2009.

[44] U. Sawant and S. Chakrabarti. Learning joint query interpretation and response ranking. In *WWW Conf.*, Brazil, 2013.

[45] S. Shah, K. Ramamritham, and P. J. Shenoy. Maintaining coherency of dynamic data in cooperating repositories. In *VLDB*, pages 526–537. Morgan Kaufmann, 2002.

[46] S. Shah, S. Sudarshan, S. Kajbaje, S. Patidar, B. P. Gupta, and D. Vira. Generating test data for killing SQL mutants: A constraint-based approach. In *ICDE*, pages 1175–1186, 2011.

[47] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill, 6th edition, 2010.

[48] L. Specia, A. Srinivasan, S. Joshi, G. Ramakrishnan, and M. das Graças Volpe Nunes. An investigation into feature construction to assist word sense disambiguation. *Machine Learning*, 76(1):109–136, 2009.

[49] L. Specia, A. Srinivasan, G. Ramakrishnan, and M. das Graças Volpe Nunes. Word sense disambiguation using inductive logic programming. In *ILP*, pages 409–423, 2006.

[50] A. Srinivasan and G. Ramakrishnan. Parameter screening and optimisation for ilp using designed experiments. *Journal of Machine Learning Research*, 12:627–662, 2011.

Intel "Big Data" Science and Technology Center Vision and Execution Plan

Michael Stonebraker, Sam Madden, Pradeep Dubey
stonebraker@csail.mit.edu, madden@csail.mit.edu, pradeep.dubey@intel.com
http://istc-bigdata.org

Abstract

Intel has moved to a collaboration model with universities consisting of "Science and Technology Centers" (ISTCs). These are located at a "hub" university with participation from other universities, contain embedded Intel personnel, and are focused on some research theme. Intel held a national competition for a 5th Science and Technology center in 2012 and selected a proposal from M.I.T. with a theme of "Big Data". This paper presents the big data vision of this technology center and the execution plan for the first few years.

1. Introduction

Others have categorized "big data" in three ways:

Big volume. The size of the database is too large to manage with current tools.

Big velocity. The data is arriving too fast for software to cope. This has been labeled as "drinking from a fire hose".

Big variety. The data is coming from too many disparate sources, and there is a massive data integration problem.

The members of the ISTC big data team believe that the "big volume" category must be subdivided into two components. Some users want to run conventional SQL analytics on massive data sets. In our opinion, this market is well served by the commercial data warehouse vendors, who are adept at managing peta-scale databases on multi-hundred node server farms, with on-line replication and failover and full transaction support. In fact, we know of a dozen or so installations at this scale from multiple vendors. Hence, we do not see the necessity of a research initiative in this area.

In contrast, we see an emerging need for complex analytics (machine learning, data clustering, predictive modeling, data categorization, etc.) on massive data sets. This market is not well served by the data warehouse crowd. In fact, most of the underlying algorithms are expressed as sequences of linear algebra operations on matrices. Hence, the relational model of data is a poor fit to this class of problems. Obviously, there is a need to mix complex analysis operations with data management (filtering, joins, etc.). As such, statistical packages provide only part of the needed functionality. Therefore, we are performing a major initiative in this area, as will be discussed in Section 2.

In the "big velocity" space, we require a corresponding subdivision. Some applications, for example electronic trading, must consume a "fire hose", looking for complex patterns in the stream of data. For example, if the trading engine thinks stocks A, B and C are correlated, then it would look for movement in any two of the three, and then trade the third based on the expected correlation. Complex event processing (CEP) engines are focused on this use case. The DBMS research community focused on this application area during the last decade, (see for example [8] and [9]), and it is not clear that a major new initiative is warranted in this area.

The second "big velocity" problem is processing the fire hose as before, but dealing with updating a persistent state, rather than searching for patterns. Maintaining the state of massively multiplayer Internet games is an example of this second use case. Here, applications look more like very high performance On-Line Transaction Processing (OLTP) problems. We know of many areas (e.g., maintaining leader boards, ad placement on web pages, maintaining real time risk exposure in trading engines) where there is considerable pain because of high velocity input. In Section 3, we indicate our initiative in this area.

In both areas, we propose end-to-end research programs, which span visualization technology, DBMS technology, and algorithm development, as will be explained in Sections 2 and 3 of this paper. Obviously, Intel is interested in the implications of big data on CPU and storage architectures; hence, we have research tasks in both areas dealing with hardware implications. These will be covered in Section 4.

We believe the "big variety" problem to be exceedingly important. In fact, some of us are active in this area [10]. In the future this ISTC will likely expand its scope into this facet of big data. Also, there are several aspects of "big data" that we are not addressing, largely because other Intel ISTCs are addressing them. These issues include security and privacy, being addressed by the ISTC at Berkeley, and cloud-oriented aspects of big data being tackled by the ISTC at CMU.

2. Complex Analytics – Big Volume

2.1 Motivation

We are motivated by four real world problems, which we briefly describe in this section.

Problem 1: Earth Science and satellite imagery. On the ISTC team are two researchers (Jim Frew and Bill Howe) who deal with Earth Science research using satel-

lite imagery. Frew uses such imagery to analyze snow depth in the Sierra Nevada Mountains to help manage water in the state of California, while Howe works with an oceanography group looking at water quality issues in Puget Sound. Both groups want to do transformations on massive amounts of imagery (MODIS in their case) and then browse the resulting data sets. The requirement is a scalable visualization system connected to a scalable database holding many terabytes of MODIS data.

Problem 2: Medical records and ICU data. Another ISTC member (Peter Szolovits) is interested in predictive modeling of medical data. We have access to 1600 patient days of intensive care unit (ICU) monitoring data, along with corresponding patient records. The goal is predictive modeling of medical events (for example code blues), so early intervention can be taken. This task requires complex prediction on a sea of data. Additionally, we have teamed with the Massachusetts General Hospital (MGH) and have access to their cancer patient database, which contains treatments and test results for all MGH cancer patients for the last 20 years. We are exploring a range of questions from modeling the relationship between cost and outcomes to the effectiveness of early screening and preventative medicine programs like mammograms and flu shots.

Problem 3: Large industrial machine maintenance. We are still looking for a partner in this area, but can describe the task as follows. Consider a complex piece of machinery (jet engine, helicopter, chemical plant, agricultural combine, etc.) Appropriate companies have the entire maintenance history on each piece of equipment and often real time monitoring data. The goal is to predict unscheduled maintenance problems, so they can be dealt with during a previous scheduled maintenance event. Problem 3 is the same kind of predictive modeling as Problem 2, but in a different domain.

Problem 4: Graph data. In the semantic web, Twitter, Facebook, and many science communities, there is a preponderance of graph data. What is needed is complex analytics on very large graphs. As an example, consider finding the average distance between any two humans in the Facebook graph. Other operations include minimum cut sets, reachability, etc.

We have (or expect to have) large amounts of data in each area on a server at MIT. In aggregate, we hope to have about 500 terabytes (0.5 petabytes) of data under management. Although others might suggest storing 0.5 petabytes in a file system, we believe that is the wrong approach to massive data sets. Database Management Systems (DBMSs) offer a range of useful services not addressed by file systems, including a schema (to control data semantics), a query language (to access subsets of data), sophisticated access control (on data granules), data consistency services (integrity control and transactions), compression (to reduce storage space), and indexing (to speed query performance). In our opinion, everybody with a big data problem should use DBMS technology. Hence, the file system is merely the storage layer used by the DBMS.

All of the sample problems in Section 2.1 are array or graph data and are ill-suited to the relational model. As a result we are studying two other options. First, we are exploring array databases, such as SciDB [11][1], as an obvious representation for much of the above data. In addition, we are exploring how to manage graph data. Our options include building a native graph DBMS, simulating graph data as sparse arrays in an array DBMS and a commercial graph DBMSs, such as Neo4J. We are also exploring visualization of large data sets and efficient algorithms for complex analysis.

2.2. Array Databases

We are investigating a variety of issues surrounding array DBMSs, including the following.

Array query languages. The success of the relational model has been helped immensely by a standard notation for queries and updates (SQL). In fact, other technologies, for example object-oriented databases and the entire "NoSQL" movement, have been hampered by a lack of standards. Some of us (Stan Zdonik and David Maier) are working on a standard query language for array data. Our approach is to define a standard abstract algebra of the semantics of operations (e.g. join, restriction, etc.). Then we plan to solicit agreement from the popular array-based DBMSs, including SciDB, Rasdaman and SciQL. Once we have agreement on the meaning of basic operations, we can move on to formalizing an SQL-like notation for arrays. A necessity for such an array query language (AQL) is to operate on arrays with integer dimensions (the standard ones in programming languages) as well as ones with dimensions of other data types (say latitude and longitude). As such the data model must allow arbitrary dimensions of user-defined data types, along with cell values that can be arbitrary vectors. It is also possible that we will extend our work to cover arrays with cell values that are complex data types. Our initial efforts are detailed at http://www.xldb.org/arrayql/.

Another standardization effort builds on the universality of the R statistical environment. R includes computation and visualization, as well as statistics. Hence, it is widely used as a programming and execution model for scientific computation. One of the pet peeves of many R users is the absence of scalability and data management functionality. Hence, we have built an extension of R that allows it to perform scalable execution by passing commands to an array database backend. This system is described in [12].

[1] One goal of our work in the ISTC is to release all code under an MIT or BSD open-source license. Because SciDB is GPL, our implementations do not make use of any SciDB code, and are designed to be able to operate independently from it.

Physical layout of array data in storage. There has been considerable research on the best ways of allocating relational data to storage blocks. However, most commercial RDBMSs allow a table to be sequenced, and that makes it a one-dimensional array. General arrays, on the other hand, can have multiple dimensions, and this allow more opportunities for storage optimization than do tables or sequenced tables. Hence, the best way to "chunk" multi-dimensional arrays onto storage blocks is a question we are working on. This problem gets more interesting if arrays are sparse and have a multitude of holes (nulls). Even harder is the case where arrays are sparse and the empty cells are skewed. For example imagine a two dimensional array with a cell value for every person in the United States. Obviously the density of people in Manhattan is 5 orders of magnitude higher than that in Montana.

We are investigating fixed size chunking systems, which would define a "stride" in one or more dimensions as the size of a "chunk". Such a layout makes query processing straight-forward, but will have problems with sparse and skewed data. On the other hand, we are also considering hierarchically decomposable systems based on splitting chunks, for example using quad trees. This will support skewed data using a regular, but variable size chunking system. Finally, we could also use a hierarchical chunk splitting scheme, for example based on R-trees, whereby all chunks become variable in size and irregular. The more flexible schemes deal with sparseness and skew more effectively, but make processing of joins more difficult. Lastly, we are investigating a scheme to group fixed-size chunks into "super chunks". Our initial results are presented in [13].

No overwrite and versioning of data. In many scientific applications it is important to be able to go back to earlier versions and compare the results of computations. Our approach is to add an extra dimension onto all arrays, which records wall clock time. Then, updates to array data merely add cells in this extra dimension. Hence, arrays have a dimension that grows without bound. This, of course, makes chunking strategies even more challenging, and our initial work in this direction is presented in [1][14].

Seamless on-line reprovisioning. A goal of all DBMSs is to support dynamic reprovisioning. In other words, if a data base is currently allotted X nodes, and more horsepower is needed, then the software should be able to add another Y nodes of storage and processing and then seamlessly move to utilizing all X + Y nodes for storage and processing. There have been extendible techniques developed for record data (e.g., Chord) as well as ones that make no attempt to organize the data for fast access (e.g., Hadoop). We are starting an effort to do the same thing for the chunk-oriented data we see in array-based systems.

Query optimizers. Optimization of SQL commands for relational data has been investigated for years, and appears to be well understood. There are well known strategies for performing joins of tables spread across multiple nodes in a computer system. However, array DBMSs present additional challenges. For example, if two arrays are joined using equality on all of the dimensions, then a straightforward chunk-to-chunk join can be performed. This generalizes the standard merge-sort used in relational systems. In addition, an array system must also perform joins where the join predicate entails matching cell values as well as ones that have a mix of cell values and dimension values. Just as with storage optimization, array systems present a more complex challenge than the simpler relational systems that have preceded array DBMSs.

Provenance. In most of our applications in Section 2.1, there is the possibility of incorrect data. Hence, whenever a result is calculated, a user should be able to trace the derivation of data, if he believes the result to be suspect. We have built an elaborate system that does exactly that, exploiting the semantics of relational and array operators to be able to efficiently work backwards, using a notion of *fine-grained provenance* [7]. We are also currently investigating visualization and other tools to help users understand data quality [20].

2.3. Matrix Calculations

Many big data analytic applications will need to combine data management with linear algebra in the same query, for example, finding the covariance between the historical times series of all pairs of stocks that have a market capitalization over $1B. This is a filter operation (at most an operation that is linear in the matrix size) followed by a covariance computation (cubic in the array size). Obviously, the "high pole in the tent" is the matrix calculation underneath covariance. There is a 10^5 difference (or more) in performance between coding such an operation in Python and in carefully optimized C++. Performance differences can be even greater when considering parallel implementations of such operations, and building efficient implementations can require many man-years of labor. Since there exist carefully optimized implementations of array-parallel operations (e.g., ScaLAPACK for dense arrays and ARPack for sparse arrays), we believe it makes sense for a DBMS to reuse to these libraries (as user-defined functions) whenever possible. Using optimized matrix code should move composite queries to be less dramatically CPU bound.

However, resource management is a problem in this hybrid world, because both the database system and ScaLAPACK are trying to be elastic and take advantage of otherwise idle resources. However, each system uses resources as though it has full control of the system, and does not access memory or disk in a way that is "friendly" to the other system. Hence, we are working on a meta-resource manager to mediate the resource demands of each system.

2.4. Graph Data

It is also clear that RDBMSs are a poor fit for graph data, although Facebook has continued to make them work for their problem. We are working on a number of different tasks in this area.

First, Carlos Guestrin has written a graph processing system (GraphLab) supported by a custom processing engine. This engine, which started as a single node main-memory system, has been extended to support distributed main memory and independently to support a single node disk environment. To truly scale, it must be further extended to distributed rotating storage. This will be the performance baseline, against which any other engine can be compared. One thrust is to implement graphs as sparse matrices. One of our team members (Jeremy Kepner) has a graph-processing engine supported by sparse arrays [24], which we plan to test against the baseline on a GraphLab benchmark.

To complement this activity, we are also working on a graph-specific storage system. We plan to compare these systems, to see if the above sparse matrix simulation of graphs is competitive with a native graph engine. We will also bake off GraphLab on top of Hadoop (and perhaps Pregel). We are skeptical that any Hadoop-based scheme will be competitive.

2.5. Visualization

The traditional form-based user interface (UI) technology is mostly useless in the problem domains of Section 2.1. Instead one needs a visualization system. Our focus is on scalability issues, not on the pixel representation on the screen. For example, MODIS users want an array browser to look through the gridded data that results from domain-specific transformations. Pointed at California, such a browser would overwhelm a conventional screen with data cells (in other words, the screen would be painted black). Instead, middleware software should perform resolution reduction to deliver to the visualization system an understandable amount of data. Our initial system that leverages query optimizer prediction of result sizes is described in [21]. We are working on a much more elaborate system, and are also working on predictive middleware to do intelligent prefetching and caching [22]. In parallel we are also investigating client side caching and how two optimization systems can work together [25].

2.6. Scalable Algorithms

We are working on several new, scalable algorithms, including a new, parallel streaming implementation of the widely used Locality Sensitive Hashing [2] and a new, scalable language for scientific computation called Julia.

PLSH: The goal of the PLSH (Parallel Locality Sensitive Hashing) project is to extend the widely used idea of Locality Sensitive Hashing to run in parallel on Intel multi-core chips, distributable across several machines, and to support streaming updates as new data arrives and is hashed. We are planning to deploy it this spring on a collection of 1 billion tweets, looking at applications ranging from finding pairs of users who tweet about similar things to hash tags that a given user should follow.

Julia: Julia is a high-level, high-performance dynamic programming language for technical computing, with syntax that is familiar to users of other technical computing environments. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library. In addition, the Julia developer community is contributing a number of external packages through Julia's built-in package manager at a rapid pace. Julia programs are organized around multiple dispatch; by defining functions and overloading them for different combinations of argument types, which can also be user-defined. We plan to integrate Julia with SciDB, so that SciDB applications can be written in Julia.

3. "Big State - Little Pattern" High Velocity Problems

As noted above, the second major thrust of our research is in high throughput processing of operations over large amount of state.

3.1. Motivation

High velocity data means drinking from a fire hose, using online transaction processing (OLTP). Obviously, the only way to do this is with a parallel OLTP engine with very high node performance. Our thinking in this area is motivated by our work in [3], which showed that traditional RDBMSs suffered from high overhead, specifically in the implementation of dynamic locking, write-ahead logging, buffer pool management and multi-threading. Only perhaps 10% of the cycles contributed to useful work; the rest goes into the overhead associated with the above four issues. Clearly, one must remove all four of the above sources of overhead to perform dramatically better than traditional systems. Based on these criteria, we designed the H-Store OLTP-oriented DBMS a few years ago (see http://hstore.cs.brown.edu). It solved the four problems by eliminating the buffer pool, executing transactions in timestamp order, implementing command logging rather than data logging, and dividing main memory among the various cores, so there is no multi-threading. H-Store has been shown to be about two orders of magnitude faster than traditional RDBMSs on TPC-C [3]. However, there are substantial issues remaining, as we discuss below.

3.2. "Anti-caching"

We are working on relaxing the requirement that all H-Store data fit in the collective main memory of the allocated nodes. H-Store does not work well in this situation, as the only option is to allow the virtual memory manager on the underlying OS to page data to disk, which is extremely slow. Instead, we are investigating "anti-caching".

When memory is nearly exhausted, we package up the least-used ("coldest") tuples and write them to disk, together with a map of their location. As a result, the most used ("hottest") data resides in main memory and the cold data is on disk (but in main memory format). H-Store has been modified to make a "pre-pass" for any command to ensure needed tuples are in main memory. If not, they are

fetched, and the transaction is delayed until all needed data is main memory resident. Then, the command is executed normally. We have worked out eviction policies, fetch policies, and disk rearrangement policies for this model and have benchmarked it against a traditional RDBMS (MySQL). Additionally, we have benchmarked our system against MySQL with a Memcached main memory cache. Our system is dramatically faster than either system on almost all workloads, and H-Store degrades very slowly as the database becomes larger and larger. A paper on this effort has been submitted [16].

3.3. Concurrency Control

With the advent of interest in high performance main memory transactional data bases and the realization that traditional record-level locking is too slow to be used, there have been numerous ideas for high performance concurrency control, including deterministic time stamp ordering with speculative execution (H-Store), deterministic scheduling via pre-resolving conflicts [17], and multi-version concurrency control (NuoDB, Hekatron). We plan to study these algorithms to see if there are workloads on which one or another is preferred. Such studies were popular in the 1980's for disk-based DBMSs [18].

3.4. Integration of OLTP and Stream Processing

In the past, some of us have worked on complex event processing (CEP) engines. We have built the StreamSQL engine [9] as well as high performance pattern matching systems [4][5][6]. In effect, these are query processing engines that maintain a main memory state (the current partial satisfaction of temporal matches). There is much commonality between a CEP engine and an OLTP engine like H-Store. Each has a set of metadata catalogs, an execution engine, and the need for services such as high availability and crash recovery. As such, it would make perfect sense to combine a CEP engine with an OLTP engine. The composite would have broader applicability as well as allowing the sharing of quite a bit of functionality. Hence, we plan to start a project in this area.

4. Implications of "Big Data" on Computer Architecture

Both "big volume" and "big velocity" have implications for computer architecture as we explain below.

4.1. Big Volume Issues

At the heart of complex analytics lie algorithms with high computational complexity. Additionally data access patterns are often highly irregular, as in simple breadth first search of very large social network graphs. Addressing the architectural needs of a big-data compute platform is therefore quite challenging. Our immediate goal is to assess the new Intel® Xeon Phi™ chips for their capabilities in an end-to-end system, composed of both Intel® Xeon Phi™ and Intel® Xeon® chips.

Our first cut is to run the data management code on the Intel® Xeon® chips and ScaLAPACK on the Intel® Xeon Phi™ chips. The result should be a dramatic speedup in the matrix calculations. Opinions abound as to what the "high pole" will be in this configuration. Clearly, the matrix calculations will be improved significantly, which may result in an I/O bound or network bound composite system.

To support this work, we require a standard benchmark, which can be run on various hardware configurations and on other DBMSs and stat packages. We have developed a genomics benchmark [19] and are in the process of running it on hardware configurations, ranging from low end server clusters to the Stampede supercomputer at the University of Texas, which has thousands of nodes, each composed of Intel® Xeon Phi™ and Intel® Xeon® chips. In addition, we plan to test a variety of solutions capable of executing combined DBMS and statistics workloads.

This work could have substantial ramifications for the design of future high performance computers. Many existing supercomputers have a compute cluster, which is distinct from a companion file system cluster. Instead, we are proposing a much tighter integration of computation and storage management. Also, one can vary the computational resources of nodes by varying the ratio of Intel® Xeon Phi™ boards to Intel® Xeon® boards.

The genomics benchmark noted above has two instantiations, one is a dense array of genome values, while the other is an array of popular genomic sequences (SNPs). Since humans possess only some of these sequences, the array is quite sparse. Our benchmark requires covariance, biclustering and linear regression on such arrays. Optimizing dense array calculations for Intel® Xeon Phi™ is being done by Jack Dongarra, while others in the Intel Lab in Santa Clara are optimizing these operations for sparse matrices.

Additionally, we are working on using GPUs (including the Intel® Xeon Phi™) to efficiently render visualizations of massive scale data, using techniques such as transparency, heat maps, and other techniques to aggregate together many data points and present them most effectively. As a part of this effort, we are looking at pushing some kinds of common data filtering and processing operators into these types of co-processors.

Lastly, fixed function hardware can deliver orders of magnitude improvement in energy efficiency with respect to its programmable counterpart. We foresee opportunities for such acceleration for repeatedly used primitives like, data compression and basic operations on repeatedly used core data structures like a binary tree. This task is aimed at proper identification and abstraction of these functions such that hardware cost is minimized and the ease/portability of software development is not compromised.

4.2. Big Velocity Issues

There are three projects we are investigating in the big velocity realm. The first concerns thread movement, the second deals with flash memory as a replacement for disk, while the third concerns making main memory persistent.

We believe that there is an opportunity for hardware and/or operating system support for moving threads among the CPUs in a cluster. We are working on the details of doing this ultra-efficiently, possibly using new hardware we are developing [23]. If thread movement can be made efficient enough, then we need to revisit the standard DBMS scheme of "move the query to the data". Specifically, current H-Store execution decomposes a command into a tree of operations divided into phases. During each phase, a sub-command is executed at each of perhaps several modes and then reshuffling of data is performed. This strategy is reasonable when the cost of thread migration is expensive.

Cheap thread migration allows us to rethink query execution. In particular, one could have a collection of threads that move from node to node, exchanging synchronization and data messages when necessary. Moreover, a different execution scheme might allow other concurrency control schemes or in "tilting the playing field" toward one or another of the known schemes.

Many enterprises are currently investigating flash memory as a persistence mechanism to replace slower rotating magnetic storage. The primary reason is to improve the performance of secondary storage. Our second project is to explore the use of flash in H-Store. This can be performed in two different ways. It is a "drop on" to replace the disk in our anti-caching system with flash memory that is block addressable. However, our anti-caching system would rather have byte addressable flash system so finer granularity objects could be moved back and forth. We could even try putting the whole data base on flash; thereby using flash as a main memory replacement. We plan to address the performance of all of these configurations on a standard benchmark.

Looking further into the future, our third task is to explore the potential for emerging non-volatile (persistent), byte-addressable memory technologies, such as *phase change memory* (PCM). This technology offers DRAM-like access speed while being non-volatile, without the huge energy overhead and performance degradation of disks. We expect this technology to be better suited for main memory replacement than flash, which would eliminate the need for elaborate recovery schemes when power to DRAM is lost. Intel will provide us with a PCM emulator through which we can test the implications of this technology, both in a conventional H-Store setting as well as in an anti-caching setting. A paper on the performance of these memory systems is in preparation.

5. Summary

This paper has described the newest Intel ISTC focused on big data. As we explained, we are working on both big volume and big velocity issues, leaving big variety as a future topic. Our approach is to develop and leverage DBMS technology, as opposed to file systems. In all cases there are significant implications to the design of future computer systems. For more information about the Intel Science and Technology Center in Big Data, visit our website at http://istc-bigdata.org.

References

[1] A. Seering, P. Cudre-Mauroux, S. Madden, and M. Stonebraker. Efficient Versioning for Scientific Array Databases. In *ICDE* 2012.

[2] A. Gionis, P. Indyk, and R. Motwani. Similarity Search in High Dimensions via Hashing. In *VLDB* 1999.

[3] S. Harizopoulos, D. Abadi, S. Madden, and M. Stonebraker. OLTP Through the Looking Glass, And What We Found There. In *SIGMOD* 2008.

[4] Y. Mei, and S. Madden. ZStream: A Cost-based Query Processor for Adaptively Detecting Composite Events. In *SIGMOD* 2009.

[5] R. Newton, L. Girod, M. Craig, S. Madden, and G. Morrisett. Design and Evaluation of a Compiler for Embedded Stream Programs. In *LCTES* 2008.

[6] W. Thies, M. Karczmarek, and S. Amarasinghe. StreamIt: A Language for Streaming Applications. In *ICCC* 2002.

[7] E. Wu, S. Madden, and M. Stonebraker. SubZero: a Fine-Grained Lineage System for Scientific Databases. In *ICDE* 2013.

[8] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. Manku, C. Olston, J. Rosenstein, and R. Varma. Query Processing Resource Management and Approximation in a Data Stream Management System. In *CIDR* 2003.

[9] D. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. The Design of the Borealis Stream Processing Engine. In *CIDR* 2005.

[10] M. Stonebraker, D. Bruckner, I. Ilyas, G. Beskales, M. Cherniack, S. Zdonik, A. Pagan, and S. Xu. Data Curation at Scale: The Data Tamer System. In *CIDR* 2013.

[11] M. Stonebraker. The Architecture of SciDB. In *SSDBM* 2011.

[12] P. Leyshock. Agrios: A Hybrid Approach to Scalable Data Analysis Systems. In *XLDB* 2012.

[13] E. Soroush, M. Balazinska, and D. Wang. ArrayStore: A Storage Manager for Complex Parallel Array Processing. In SIGMOD 2011.

[14] E. Soroush, and M. Balazinska. Time Travel in Scientific Array Databases. In *ICDE* 2013.

[15] N. Malviya, S. Madden, and M. Stonebraker. Rethinking Main Memory OLTP Recovery. (*submitted for publication*)

[16] J. DeBrabant, A. Pavlo, M. Stonebraker, S. Tu, and S. Zdonik. The Traditional Wisdom is all Wrong. (*submitted for publication*)

[17] A. Thomson, T. Diamond, S. Weng, K. Ren, P. Shao, and D. Abadi. Calvin: Fast Distributed Transactions for Partitioned Database Systems. In *SIGMOD* 2012.

[18] M. Carey, and M. Stonebraker. The Performance of Concurrency Control Algorithms for Database Management Systems. In *VLDB* 1984.

[19] M. Vartek, and R. Taft. A DBMS Benchmark for Complex Analytics. (*in preparation*)

[20] E. Wu, S. Madden, and M. Stonebraker. A Demonstration of DBWipes: Clean as You Query. In *VLDB* 2012.

[21] L. Battle. Resolution Reduction to Augment Visualizations. (*submitted for publication*).

[22] J. DeBrabant, L. Battle, U. Çetintemel, M. Stonebraker, and S. Zdonik. Caching and Prefetching to Support Massive Data Visualization. (*in preparation*).

[23] M. Lis, K. Shim, M. Cho, O. Khan, and S. Devadas. Directoryless Shared Memory Coherence Using Execution Migration. In *ICPDC* 2011.

[24] J. Kepner, W. Arcand, W. Bergeron, N. Bliss, R. Bond, C. Byun, G. Condon, K. Gregson, M. Hubbell, J. Kurz, A. McCabe, P. Michaleas, A. Prout, A. Reuther, A. Rosa, and C. Yee. Dynamic Distributed Dimensional Data Model (D4M) Database and Computation System. In *ICASSP* 2012.

[25] Z. Liu, B. Jiang, and J. Heer. ImMens: Real-Time Visual Querying of Big Data. (*submitted for publication*)

Report on the First International Workshop on Energy Data Management (EnDM 2012)

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

Wolfgang Lehner
Database Technology Group,
Dresden University of Technology
01062 Dresden, Germany
{firstname.lastname}@tu-dresden.de

Gregor Hackenbroich
Data Management and Analytics
SAP Research, SAP AG
01187 Dresden, Germany
gregor.hackenbroich@sap.com

1. INTRODUCTION

The energy sector is one of the most active application domains being forced to re-think the current practice and apply data-management based IT solutions to provide a scalable and sustainable supply and distribution of energy. Challenges range from energy production by seamlessly incorporating renewable energy resources over energy distribution and monitoring to controlling energy consumption. Decisions are based on huge amounts of empirically collected data from smart meters, new energy sources (increasingly RES - renewable energy sources such as wind, solar, hydro, thermal, etc), new distributions mechanisms (Smart Grid), and new types of consumers and devices, e.g., electric cars.

Energy is at the top of the worldwide political agenda, e.g., due to global warming concerns and recent nuclear accidents. Ambitious goals for reductions of energy consumption and CO2 emissions have been formulated, e.g., the EU 20-20-20 goals (20% renewable energy, 20% better energy efficiency, and 20% CO2 reduction by 2020), with much more ambitious goals set for 2030 and 2050. This situation is reflected by increasing attention in research funding schemes such as the EU 7th Framework program as well as national programs. A recent trend in these programs is 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).

There is thus a need for focusing on data management within the energy domain. 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 to bridge the gap between domain experts and data management scientists on the one hand. Additionally, the workshop's goal is to create awareness of this upcoming and very challenging application area. For the workshop's research program, the organizers solicited contributions that push the envelope towards novel schemes for large-scale data processing with special focus on energy data management.

The First International Workshop on Energy Data Management (EnDM'12)[1] was held in conjunction with EDBT 2012 in Berlin, Germany on March 30, 2012. 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 discussion. The accepted papers spanned a number of exciting topics within energy data management, including (in no particular order) smart grid architectures, smart grid specific data management challenges, and the use of gamification in active demand response, as well as related issues such as energy efficient file access and energy environmental impact data management. The proceedings of the workshop was published in a joint volume of all EDBT/ICDT 2012 workshops [1].

2. INDUSTRIAL KEYNOTE

The keynote was given by Dr. Kevin Brown, Chief Architect for Informix Dynamic Server at IBM, and was entitled "*The Massive Data Challenge - A unique approach to handling smart meter data with a hybrid database*". The talk first outlined the massive challenges related to efficient management of very large amounts of electricity meter data. It then went on to describe the Informix Benchmark for Meter Data Management including the specifics of the captured data and the query workload to process over it. The talk introduced a specific instance of this benchmark, the so-called "100 Million Meter Benchmark" [2], which simulates 100 million meters being read every 15 minutes. The talk also delved into the specifics of the Informix TimeSeries extension, including its optimized physical data storage and loading strategies, as well discussing how it could be used to handle the Meter Data Benchmark much more effieciently than traditional RDBMSes. The solution was shown to handle both the benchmark and seveal real-world cases from US energy companines much more efficiently than traditional solutions. In addition to providing this exciting technical contribution, Dr. Brown provided his valuable industry perspective on the remaining papers in the workshop.

[1] http://endm2012.endm.org

3. RESEARCH PAPERS

The paper by Masaru Iritani and Haruo Yokota entitled "*Effects on performance and energy reduction by file relocation based on file-access correlations*" considered the energy-efficient placement and relocation of files across a set of distributed hard disk drives (HDDs) with the goal of reducing the energy consumption of the drives while keeping the access performance of the system at the same level. Previous approaches have mainly located frequently accessed files together on a few drives in order to enable spin-down of the remaining drives, but this causes significant energy consumption for spin-up of these drives when accessing infrequently accessed data, especially when some files that tend to be used together are placed on many different drives. The paper goes further by proposing a novel method called PLECO (Placement of files for Latency and Energy Consumption Optimization). This method tries to locate correlated files on the same drive, and thus both reduce power consumption further while also improving the system performance. The simulated evaluation of PLECO indicates that it can reduce both the energy consumption and the access latency by up to 32% and 92%, respectively, compared with a baseline system.

The next paper was by Benjamin Bertin, Vasile-Marian Scuturici, Emmanuel Risler, and Jean-Marie Pinon and was entitled "*A semantic approach to life cycle assessment applied on energy environmental impact data management.*" The paper concerned semantic web-based modeling of lifecycle assessment for energy environmental impact. Specifically, the paper focused on the life cycle inventory stage of life cycle assessment, which decomposes a life cycle into its individual economic activities. Modeling this is complex due to the large amounts of elementary processes and interdependency links. The paper proposes a semantic approach for the modelling of life cycle inventory databases which in comparison with earlier work offers a more comprehensible model. The model is explained and illustrated with life cycle inventory data for the U.S. electricity production.

The paper by Matthias Boehm, Lars Dannecker, Andreas Doms, Erik Dovgan, Bogdan Filipic, Ulrike Fischer, Wolfgang Lehner, Torben Bach Pedersen, Yoann Pitarch, Laurynas Siksnys, and Tea Tusar called "*Data management in the MIRABEL smart grid system*" focused on the data management challenges of a specific approach to the smart grid. The motivation for the paper is that Renewable Energy Sources (RES) are becoming increasingly important to reduce greenhouse gas emissions and will take up a much larger share of the energy production. This leads to a number of challenges such as balancing energy supply and demand since RES cannot be scheduled. The paper addresses the balancing challenge by specifically presenting the MIRABEL project and its Energy Data Management System (EDMS) which uses the flexibilities available in the electricity demand, e.g., dishwasher, electric vehicles, etc., to efficiently balance energy demand and supply. The major novel concept of MIRABEL are so-called *flex-offers* that explicitly capture intended energy use and the flexibilities in time, amount, and price that are associated with it. A MIRABEL-based EDMS will eventually consist of millions of heterogeneous nodes, each incorporating a number of advanced components such as flex-offer aggregation, forecasting, scheduling, and negotiation. The paper describes each of these components and their interaction while focusing on the data managent challenges that arise. The challenges include effective aggregation of flexibilities, tight integration of forecasting, both for functionality and forecasted data, the interplay between aggregation, forecasting, and scheduling, and the monetization of the flexibilities. The experimental results show that the proposed EDMS is indeed feasible.

The next paper by Benjamin Gnauk, Lars Dannecker, and Martin Hahmann, entitled "*Leveraging gamification in demand dispatch systems*," focused on how to involve energy consumers more actively in so-called demand-side management techniques to help optimize the grid's efficiency and a better utilize renewable energy sources. The paper focuses on so-called demand dispatch systems, where consumers must proactively communicate their flexibilities. A standard incentive is monetary compensation, but this is often not enough to motivate the individual consumer for a sustainable participation. The approach proposed by the paper instead uses *gamification* as a motivational framework. Here, well-known game mechanics instruments, e.g., point awards and leaderboards, are used to engage the consumers. The paper explains the special scoring system used and how it is combined with aspects of social competition in a user interface that helps consumer define and management their flexible energy demands. The paper reports on an initial user study which shows that the user acceptance is high and that the system can potentially engage many consumers.

Finally, the paper by Daniel Rech and Andreas Harth called "*Towards a decentralised hierarchical architecture for smart grids*" considered the technical architecture for smart grids. Specifically, the paper presented a hierarchical distributed communication and control architecture for Smart Grids. The topology of the proposed architecture accomodates the decentralised nature and large sizes of smart grid systems by having multiple layers in order that ensures both robust and flexible data access and resource allocation. The paper describes a specific use scenario with a number of different smart grid actors, and further develops an architecture for this scenario based on the Linked Data principles known from the semantic web area. Further, the authors propose a simple language that can express allocation constraints. They also map the resource allocation problem into a constraint satisfaction problem. The paper finally provides initial experimental results within the tasks of decentralised data access and resource allocation for smart grids.

4. DISCUSSION AND OUTLOOK

At the end of the workshop, a lively discussion took place among the participants, the conclusion of which are included in this section along with some post-workshop reflections.

If we first look at the topics of the presented papers, we note that they span a very wide range of topics, ranging from low-level technical issues within data management and communication, over conceptual level modeling, to the integration of user interaction aspects. The papers also cover both energy systems and the energy consumption of IT systems themselves. This is a reflection of the fact that the journey smart grid is long and requires tight collaboration between many different areas not just within computer science itself, but also including inter-disciplinary collaborations with other sciences.

Next, when looking at the topics which occcured in the Call for Papers, but not within the accepted (or submitted) papers, we see that topics such as data security and privacy and data mining techniques for energy data are missing. 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 more pressing issues must be solved before considering such topics. While most papers are based on small case studies, there were no papers describing large industrial case studies of already running systems. We again attribute this to the fact that smart grids are still in development.

The workshop discussions identified a number of issues that must be resolved in order to better unify and leverage the many concurrent research activities within energy data management. The first such issue was the lack of common definitions of data and information concepts within the area, e.g., community-wide agreed-upon standard ontologies specifying common concepts. Another issue was the 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. Such standards already exist at the business level of the energy sector, e.g., for standardizing the different types of actors in smart grid setups, but not yet at the more technical levels.

As the final words, we can safely conclude that there is a large demand for further work in the area of energy data management, including a need for venues that focus on this issue. The EnDM workshop series will continue at EDBT 2013 in Genoa where the 2nd International Workshop on Energy Data Management will be held on March 22, 2013[2]. For the 2nd edition of the workshop, it is the intention for organizing a special issue of a journal for extended versions of the best papers.

5. 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 2012, especially the Workshop Chair Divesh Shrivastava and the Proceedings Chair Ismail Ari. for their support in organizing the EnDM 2012 workshop.

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

6. REFERENCES

[1] D. Srivastava and I. Ari, editors. *Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, March 30, 2012*. ACM, 2012.

[2] Unprecedented performance and scalability demonstrated for meter data management. Technical report, IBM Corporation, Online at http://www.ibm.com/developerworks/forums/thread.jspa?messageID=14685713, 2011.

Report on the Fourth International Workshop on Cloud Data Management (CloudDB 2012)

Xiaofeng Meng
Renmin University of China
Beijing, China
xfmeng@ruc.edu.cn

Fusheng Wang
Emory University
Atlanta, USA
fusheng.wang@emory.edu

Adam Silberstein
Trifacta Inc.
San Francisco, USA
aesilberstein@yahoo.com

1. INTRODUCTION

The workshop series on Cloud Data Management (CloudDB) has been held successfully in the past four years [4, 2, 3, 5]. CloudDB serves as a premier forum for researchers and practitioners to present research results and share ideas and progress in the area of data management within cloud computing infrastructure.

Technology advances in communications, computation, and storage result in huge collections of data, capturing information of value to business, science, government, and society. Data volumes are currently growing faster than Moore's law. Looking forward, the exponential growth is not likely to stop. The huge volumes of data pose major infrastructure challenges, including data storage at Petabyte scale, massively parallel query execution, facilities for analytical processing, and online query processing. Meanwhile, the rise of large data centers and cluster computers has created a new business model, cloud-based computing, where businesses and individuals can rent storage and computing capacity, rather than making the large capital investments needed to construct and provision large-scale computer installations. Cloud-based data storage and management is a rapidly expanding business. Whilst these emerging services have reduced the cost of data storage and delivery by several orders of magnitude, there is significant complexity involved in ensuring that large data services can scale when needed to ensure consistent and reliable operations under peak loads. Cloud-based environment has the technical requirements to manage data center virtualization, lower cost and boost reliability by consolidating systems on the cloud.

CloudDB brings together researchers and practitioners in cloud computing and data-intensive system design, programming, parallel algorithms, data management, scientific applications, and information-based applications to maximize performance, minimize cost and improve the scale of their endeavors.

The fourth ACM international workshop on cloud data management was held in Hawaii, USA on October 29, 2012, co-located with the ACM 21st Conference on Information and Knowledge Management (CIKM) [1]. The call for papers attracted a wide range of submissions on query optimization, data security and privacy, big data analytics, and system development. The program committee accepted seven papers from fourteen submissions by authors from Asia, Europe, North America and South America. In addition, the program included three keynote speakers from leading cloud computing researchers.

2. KEYNOTE TALKS

The keynote talks covered topics on OLTP benchmarking in the cloud, data analytics in the cloud, and challenges in enabling social applications at scale.

The first keynote talk was delivered by Carlo Curino from Microsoft on *Benchmarking OLTP/Web Databases in the Cloud: the OLTP-Bench*. The speaker shared the experience in building several ad-hoc benchmarking infrastructures for various research projects targeting several OLTP DBMSs, ranging from traditional relational databases, main-memory distributed systems, and cloud-based scalable architectures. OLTP-Bench is capable of controlling transaction rate, mixture, and workload skew dynamically during the execution of an experiment, thus allowing the user to simulate a multitude of practical scenarios that are typically hard to test (e.g., time-evolving access skew). OLTP-Bench also provides ten workloads derived from synthetic micro benchmarks, popular benchmarks and real world applications.

The second keynote talk *Large Scale Data Analytics on Clouds* was delivered by Geoffrey Fox from Indiana University. The speaker summarized major issues affecting the use of clouds to support data science, and discussed the major characteristics between cloud and traditional high performance computing systems. While cloud is on-demand driven and provides scalable elastic services, traditional supercomputers can achieve high performance through large scale highly parallelized jobs. Thus, when analyzing large scale data, the characteristics of different categories of applications should be considered. These include map-only applications, MapReduce applications, classic MPI applications, and

iterative MapReduce based applications. To achieve high performance, the speaker discussed the mapping of different applications to HPC and Cloud systems.

Ashwin Machanavajjhala from Duke University presented in his keynote talk *Challenges in Enabling Social Application at Scale* on major challenges associated with social network data and new applications for social discovery and engagement. He summarized three applications that should be considered properly to solve the data management and privacy issues: i) Feed Following, or the problem of delivering highly personalized feeds based on content generated by one's friends; ii) Social Coordination, or the problem of jointly planning and coordinating on a task, and iii) Social Recommendations, or recommending objects and people based on one's social connections. He also shared his experience on working with these challenges in data management and privacy research.

3. RESEARCH PAPER PRESENTATIONS

3.1 Workload-aware Processing

Computation throughput maximization, efficient resource scheduling, and query optimization are critical components for cloud computing. These are covered by three papers from different perspectives: load balancing through automatically imbalance detecting and mitigating, elastic query processing to maximize the efficiency of providers' environment, and statistical based estimation of the data in the cloud to support query optimization in the cloud.

Cloud data stores achieve high scalability and elasticity by partitioning data across a large number of servers. These stores must detect and cope with load imbalance. Markus Klems et al. develop a cloud datastore load balancer in the paper *The Yahoo! Cloud Datastore Load Balancer*. The load balancer is called *Yak*, which now provides load balancing for Yahoo!'s cloud storage system *Sherpa*. The authors describe the key design principles for Yak: understanding the goal, measurable, simple, extensible and configurable, conservative and knowing the limit. Based on the design principles, Yak defines and monitors load metrics to detect imbalance, and provides a set of rules that decide when to invoke load balancing actions. When hotspots are detected, Yak automatically balances the load by migrating tablets from the overloaded servers, and also by splitting data into new tables.

The paper *Towards Non-Intrusive Elastic Query Processing in the Cloud* by Ticiana Coelho Da Silva et al. focuses on taking full advantages of the potential flexibility of cloud computing systems. One major benefit of cloud computing is the elasticity which enables the systems to provide and remove resources according to the applications needs in real-time. They develop a non-intrusive approach that monitors the performance of relational DBMSs in a cloud infrastructure, and automatically makes decisions to maximize the efficiency of providers' environment while still satisfying "service level agreements". The workflow of their approach contains four modules: the *Partition Engine* partitions the input query Q to achieve the query's service level objective (SLO); the *Monitor Engine* is executed within each VM allocated to process Q and aims at guaranteeing that each VM meets the expected SLO; the *Capacity Planner* provides a number of VMs initially to process Q within the agreed SLO, minimizing the computational cost and penalty; and the *Orchestration Engine* is responsible for the communication between the modules.

The paper entitled *HEDC: A Histogram Estimator for Data in the Cloud* by Yingjie Shi et al. introduces an approach for histogram estimate for data in the cloud. With increasing popularity of cloud based data management, improving the performance of queries in the cloud is an urgent issue to solve. Summary of data distribution and statistical information has been commonly used in traditional database to support query optimization and histograms are of particular interest. Since it could be much expensive to construct the exact histogram on massive data, building the approximate histogram is a more feasible solution. They propose a histogram estimator called *HEDC*. The workflow of HEDC is built on an extended MapReduce framework, and takes a novel block based sampling mechanism to leverage the sampling efficiency and estimate accuracy.

3.2 Energy Efficient Data Centers

In the paper entitled *Cloud Computing for Environment-Friendly Data Centers*, Michael Pawlish et al. consider the carbon footprint and utilization rates in data centers. Previous literature shows that low utilization rates in data centers are due to the forecasting of demand to meet spikes in data center use. This management policy has led to many servers running idle the majority of the time which is a waste of resources. The authors argue that a majority of the data centers should be downsized through decommissioning of phantom servers, virtualization, and shifting spikes in demand to a cloud provider. They adopt data mining techniques of decision trees and case-based reasoning to conduct analysis for decision support in cloud computing at data centers.

3.3 Computing Models

Zhuhua Cai et al. propose a system called *GraphInc* in the paper *Facilitating Real Time Graph Mining*. While incremental processing is critical for real-time graph mining, designing incremental graph algorithms is challenging. GraphInc is built on top of the Pregel model and provides efficient incremental processing of large graphs. Users are allowed to write programs as if on

batch workloads and the algorithms are automatically converted to an incremental one by memorizing and reusing subcomputations. Programmers thus can develop graph analytics in the Pregel model without worrying about the continuous nature of the data. GraphInc integrates new data in real-time in a transparent manner, by automatically identifying opportunities for incremental processing.

3.4 Privacy and Security

Privacy and security are critical issues to solve for processing sensitive data in the cloud. These are covered from two different aspects in following two papers respectively: privacy preserving query processing in the cloud, and efficient encryption based approach to achieve data security and query security for data processing in the cloud.

Xu Han et al focus on the privacy protection problem in the cloud in their paper entitled *Differentially Private Top-k Query over MapReduce*. They propose an efficient privacy protection algorithm called *DiffMR*, which aims to process top-k query as well as satisfying differential privacy. They adopt an exponential mechanism to select top-k records from big data sets based on specified score function to avoid the privacy leak. In order to get more accurate results, they reduce the reject rate and perform exponential selection multiple times during the MapReduce progress. Laplace noise will be added at last and then post-processing will be performed to improve the quantity of results.

The paper *A Security Aware Stream Data Processing Scheme on the Cloud and its Efficient Execution Methods* by Katsuhiro Tomiyama et al. evaluates queries over encrypted data streams in the cloud. A public cloud may be managed by a third party and outside the firewall of the organization, which brings the problem of data security and query security. The authors propose a scheme based on CryptDB to evaluate queries over encrypted data streams. They describe performance issues incurred by the proposed scheme, and present an approach to reduce the encryption cost and amounts of data to be transmitted. In addition, they propose an approach to reduce memory usage by analyzing a plan tree in a stream processing engine (SPE). The experiments demonstrate that their approaches can improve the memory utilization.

4. CONCLUSIONS

CloudDB has been held successfully four times associated with CIKM since 2009. During the four years, cloud computing has undergone significant development and attracted major interest from both industry and academia. CloudDB workshop series aims to address the challenges of large scale database services based on the cloud computing infrastructure in a timely fashion, with increasing number of participants. Topics of CloudDB 2012 covered query optimization, data security and privacy, big data analytics, and large scale social applications in the cloud. The participants agreed that many open challenges remain open, such as cloud data security and privacy, and the efficiency of big data management in the cloud.

5. ACKNOWLEDGMENTS

The workshop was partially supported by the Research Fund of Renmin University of China (No. 11XNL010) and by grant R01LM009239 from the National Library of Medicine. We would like to thank the program committee members, keynote speakers, authors and attendees, for making CloudDB 2012 a very successful workshop. We also express our great appreciation for the support from Renmin University of China and Emory University.

6. REFERENCES

[1] X. Chen, G. Lebanon, H. Wang, and M. J. Zaki, editors. *CIKM '12: Proceedings of the 21st ACM international conference on Information and knowledge management*, New York, NY, USA, 2012. ACM. 605120.

[2] X. Meng, Y. Chen, J. Lu, and J. Xu. Report on the second international workshop on cloud data management (clouddb 2010). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1969–1970, New York, NY, USA, 2010. ACM.

[3] X. Meng, Z. Ding, and H. Hu. Report on the third international workshop on cloud datamanagement (clouddb 2011). In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 2637–2638, New York, NY, USA, 2011. ACM.

[4] X. Meng, J. Lu, J. Qiu, Y. Chen, and H. Wang. Report on the first international workshop on cloud data management (clouddb 2009). *SIGMOD Rec.*, 39(1):58–60, Sept. 2010.

[5] X. Meng, A. Silberstein, and F. Wang. Clouddb 2012: fourth international workshop on cloud data management. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2754–2755, New York, NY, USA, 2012. ACM.

CALL FOR PARTICIPATION

ACM SIGMOD/PODS 2013

New York City, New York, USA

Millennium Broadway Hotel, Times Square

June 23-28, 2013

http://www.sigmod.org/2013/

PLATINIUM SPONSORS

GOLD SPONSORS

SILVER SPONSORS

PLATINIUM PUBLISHER

SILVER PUBLISHER

The annual ACM SIGMOD/PODS conference is a leading international forum for database researchers, practitioners, developers, and users to explore cutting-edge ideas and results, and to exchange techniques, tools, and experiences. We are delighted to invite you to attend ACM SIGMOD/PODS, to be held in New York, New York, from June 23 to June 28, 2013.

The SIGMOD/PODS conference will feature a broad variety of technical and industrial events that will be of interest to people in academia, research labs, and in industry. The program includes 3 keynote speakers, one panel, 8 tutorials, 11 workshops, over 90 research presentations and 15 industrial experience presentations.

New York City is the most populous city in the United States and exerts a significant impact upon global commerce, finance, media, art, fashion, research, technology, education, and entertainment. The conference will be held at the Millennium Broadway Hotel in Times Square. This Art Deco hotel is located in New York City's renowned Theater District. It is minutes from a host of activities and promises an exciting venue for this year's conference.

Registration is now open: www.regonline.com/2013sigmodpods
The deadline for early registration is May 25, 2013.

SIGMOD General Chairs:
Kenneth A. Ross (Columbia University, USA)
Divesh Srivastava (AT&T Labs-Research, USA)
SIGMOD Program Chair:
Dimitris Papadias (HKUST, Hong Kong)
PODS General Chair:
Richard Hull (IBM T.J. Watson Research, USA)
PODS Program Chair:
Wenfei Fan (University of Edinburgh, UK)