Reminiscences on Influential Papers

This issue's contributors are two women who have been extremely influential mentors in my life. I feel privileged that I had the chance to work with them during my time at IBM Almaden. So I asked what influenced them. Enjoy reading!

While I will keep inviting members of the data management community, and neighboring communities, to contribute to this column, I also welcome unsolicited contributions. Please contact me if you are interested.

Pınar Tözün, *editor*
IT University of Copenhagen, Denmark
pito@itu.dk

Fatma Özcan
Systems Research@Google, CA, USA
fatma.ozc@gmail.com

Hamid Pirahesh, Joseph M. Hellerstein, and Waqar Hasan.

Extensible/Rule Based Query Rewrite Optimization in Starburst.

In Proceedings of ACM SIGMOD Conference, pages 39-48, 1992.

This paper introduces QGM (Query Graph Model) and the rewrite engine that optimizes SQL queries using query transformations. QGM is a unique graph representation that describes the data flow and dependencies in a query, without dictating an execution order, hence capturing the semantics of an SQL query. This is critical in normalizing and optimizing queries as SQL allows expressing the same semantic query using many different constructs. Many database systems represent queries as a tree of algebraic operators, which makes it harder to normalize the query into a canonical representation.

This paper also introduces a rewrite engine, which organizes query transformations into rule classes by understanding the intrinsic interactions between them. This organization allows the rules to be applied in certain orders and controlled manner, and guarantees convergence to the normalized representation.

This paper had a tremendous impact on my career, as it provided me with a deep understanding of SQL query semantics and optimization. Although this work was done for Db2, it provided an abstraction that made it easier to understand the other SQL processors that I worked with later on. It conceptualized query semantics and optimization in a very elegant way, which was also recognized by the SIGMOD Test of Time Award in 2002.

Yuanyuan Tian
Microsoft Gray Systems Lab
yuanyuantian@microsoft.com

Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski.

Pregel: A System for Large-scale Graph Processing.

In Proceedings of ACM SIGMOD Conference, pages 135–146, 2010.

In summer 2008, I obtained my PhD degree from University of Michigan, with a thesis titled "Querying Graph Databases". In the fall of the same year, I joined the well-known database research group at IBM Almaden Research Center in the height of big data era. Immediately, I embraced myself with MapReduce and Hadoop, studying join algorithms on MapReduce, data placement policies on HDFS, federation between Hadoop and enterprise data warehouses, etc. But I never forgot my first love in research – graphs. Researchers and industry practitioners applied MapReduce/Hadoop to solve various big data problems, including relational work-

loads, machine learning, and of course graph analytics. A large body of work, including some of my own, developed specific MapReduce algorithms or MapReduce-based frameworks for graph analytics. However, the restricted Map and Reduce APIs, along with repeated reads and writes from HDFS for iterative algorithms (e.g. PageRank) limits the performance for the above approaches.

In 2010, the seminal Pregel paper was published in SIGMOD. This marks the start of specialized big graph processing engines. Pregel adopts the Bulk Synchronous Parallel (BSP) model to support iterative graph algorithms. It also employs a "think like a vertex" programming model, where computation is expressed at the level of a single vertex and communication is done through message passing. Compared to MapReduce, this programming model is more intuitive for expressing graph algorithms. In addition, during execution graph data is kept in distributed memory across iterations, leading to huge performance improvement.

Since its introduction, Pregel has sparked a large number of research works on extending its basic framework in different aspects. Some of my own research on graphs was also heavily impacted and inspired by Pregel. In one work, we extended Pregel's vertex-centric framework with a subgraph-centric model, where computation can be expressed at a subgraph level and value propagation within each subgraph could bypass network communication. In another work, we adapted the basic Pregel framework to apply point-in-time analysis on dynamic interaction graphs, in which new interactions (edges) are continually added over time. By August 2022, the Pregel paper has 4700+ citations on Google Scholar! It has become a must-to-read paper for anyone entering the field of graph analytics. No wonder that it received the SIGMOD Test of Time Award in 2020. Moreover, it also impacted the graph industry in fundamental ways. Apache Giraph was created as the open source counterpart of Pregel, and today many graph databases/systems, such as GraphX / GraphFrame and JanusGraph, adopt the Pregel-like APIs to support graph algorithms.