# SIGMOD 2012 Summary

Jeppe Rishede Thomsen

September 14, 2012

# Submissions

- Keynotes
  - Analytic Database Technologies for a New Kind of User - The Data Enthusiast, by Pat Hanrahan (Stanford)
  - Symbiosis in Scale Out Networking and Data Management, by Amin Vahdat (UCSD and Google)
- 16 Research sessions (+10 PODS sessions)
- 6 Industry sessions

# Awards

- SIGMOD Test-of-Time Awards
  - Executing SQL over Encrypted Data in the Database-Service-Provider Model
  - Visionary paper on "Database as a service" focusing on how to use cloud services while keeping some information hidden from the cloud service provider
- SIGMOD Best Paper Award
  - High-Performance Complex Event Processing over XML Streams
  - Introduced XSeq, an XPath extension orders of magnitude more efficient than existing XML engines.

# Topics

- Social Networks and Graph Databases
  - Partitioning, Clustering, subgraph isomorphism
- Temporal and Graph Databases
  - 2 on querying graphs, 1 on temporal alignment of queries
- Mobile Databases
  - 2 papers on Privacy, 1 on caching
- Distributed and Parallel Databases
- Social Media and Crowdsourcing
- Modern RDBMSs

# Distributed and Parallel Databases

- Calvin: Fast Distributed Transactions for Partitioned Database Systems
- Alexander Thomson, Thaddeus Diamond, Shu-Chun Weng, Kun Ren, Philip Shao, Daniel J. Abadi (Yale University)
- Advanced Partitioning Techniques for Massively Distributed Computation
- Jingren Zhou, Nicols Bruno, Wei Lin (Microsoft)
- SkewTune: Mitigating Skew in MapReduce Applications
- YongChul Kwon, Magdalena Balazinska, Bill Howe (University of Washington); Jerome Rolia (HP Labs)

# Calvin: Fast Distributed Transactions for Partitioned Database Systems

How do we provide fast transactions in a distributed database?

Issues:

- Several Popular distributed databases provide no transactional support (CouchDB, Cassandra, Amazon Dynamo)
- Some distributed databases limit transactions to subsets of data (Azure, Oracle NoSQL, Megastore)

Reasons:

- Reducing transactional support greatly simplifies implementation of a distributed database
- Ensuring ACID properties on queries over several partitions incur several network round trips
- For *embarrassingly partitionable* datasets it works very well

# Calvin: Fast Distributed Transactions for Partitioned Database Systems

For datasets with dependencies, users need to implement and ensure ACID properties in the application

- Slow development
- Complex code
- Poor performance

Calvin enables fast transactions over multiple partitions:

- Runs next to a non-transactional database system
- Precalculates a deterministic query plan before executing a query
- Enables near-liniar scalable shared nothing DB, providing full ACID transactions
- Node failures do not cause transactions to abort (deterministic query plan - either execute instructions later, or run on parallel replica)

# Advanced Partitioning Techniques for Massively Distributed Computation

How do we most efficiently partition or repartition data in large distributed systems?

- mapReduce scales well and can do concurrency too, but it forces developers to be aware of the mapReduce model
- Other systems (SCOPE, DryadLINQ, Tenzing, Hive) provide high level descriptive languages and offer a single machine programming abstraction.

- Introduce optimized partitioning techniques (Hash-, range-, index-based)
- Emphasis on finding good partition boundaries
- Identify data dependencies
- Considers physical data location

# SkewTune: Mitigating Skew in MapReduce Applications

How do we handle skew in MapReduce systems?

Existing solutions:

- User written skew resistant operators - extra burden on user, and only applies to certain operators
- Use very fine grained partitions - imposes a lot of overhead
- Get the complete output from an operator, sample it, then partition data before executing next operator - requires synchronization.

Limitations on skew handling:

- Handles skew from uneven distribution of input data
- Handles skew from uneven processing time of input
- Does NOT handle uneven processing power of nodes

# SkewTune: Mitigating Skew in MapReduce Applications

SkewTune:

- Replaces existing mapReduce implementation
- Optimizes existing mapReduce programs without rewrite
- Existing mapReduce programs still work
- Compatible with existing pipelining optimizations (no synchronization required.)
- Does *late skew detection*

# Social Media and Crowdsourcing

- The Value of Social Media Data in Enterprise Applications
- Shivakumar Vaithyanathan, IBM Almaden Research Center
- Anatomy of a Gift Recommendation Engine Powered by Social Media
- Yannis Pavlidis, Madhusudan Mathihalli, Indrani Chakravarty, Arvind Batra, Ron Benson, Ravi Raj, Robert Yau, Mike McKiernan, Venky Harinarayan, Anand Rajaraman (@WalmartLabs)
- Designing a Scalable Crowdsourcing Platform
- Chris Van Pelt, Alex Sorokin (CrowdFlower)

# The Value of Social Media Data in Enterprise Applications

Problem: How can data from social media be used to create *social entities*?

- IBM datamines facebook and other social networks to create *social entities* (companies, people, products)
- Have gathered enough data to often be able to distinguish two entities of the same name if mentioned in some context.

**kind of unnerving!** But still only for research.

# Anatomy of a Gift Recommendation Engine Powered by Social Media

Wallmarts gift recommendation engine: ShopyCat.

- It is a facebook application
- Lets you browse products based on what your friend might likes
- Mines your interests, as well as letting you manually specify some.

# Anatomy of a Gift Recommendation Engine Powered by Social Media

How to use Facebook to recommend *good* gifts for users? i.e.

- When is it a good time to give a friend a gift? (e.g. birthday)
- How to use friends interests to suggest specific or categories of gifts?
- Which types of products should be available to the gift recommendation engine to cover gift categories? (from other sites than wallmart)
- What is a good gift in each gift category (is an item *giftable*?

- Several existing gift recommendation engines (Gifty, Etsy, etc.) Use semi static information (besides using likes, birthdays, life events)
- ShopyCat: Only gift recommendation engine which also uses users activity

# Designing a Scalable Crowdsourcing Platform

How can we use people to solve problems that are hard for computers?

- CrowdFlower: a crowdsourcing platform for (many) people to solve small parts tasks
- Focus on 3 metrics: Quality, Cost, and Speed. Can at most do 2 at once.
- Defines "CrowdFlower Markup Language" for task submitters to define tasks.
- Different from competition (i.e. Mechanical Turk) in that it takes care of crowd quality for the task submitter.
- Can use workforce from other similar services. (many very specialized such crowdsourcing services already exist)

# Modern RDBMSs

- Query Optimization in Microsoft SQL Server PDW
- Srinath Shankar, Rimma Nehme, Josep Aguilar-Saborit, Andrew Chung, Mostafa Elhemali, Alan Halverson, Eric Robinson, Mahadevan Sankara Subramanian, David DeWitt, Csar Galindo-Legaria (Microsoft)
- F1-The Fault-Tolerant Distributed RDBMS Supporting Google's Ad Business
- Jeff Shute, Mircea Oancea, Stephan Ellner, Ben Handy, Eric Rollins, Bart Samwel, Radek Vingralek, Chad Whipkey, Xin Chen, Beat Jegerlehner, Kyle Littlefield, Phoenix Tong (Google)
- Oracle In-Database Hadoop: When MapReduce Meets RDBMS
- Xueyuan Su, Yale University; Garret Swart, Oracle

# Oracle In-Database Hadoop: When MapReduce Meets RDBMS

**Solution:** Implement direct support for Hadoop programs directly in Oracle DB

- Source compatibility with Hadoop. users are able to run native Hadoop applications
- Access to Oracle RDBMS resident data
- Minimal dependency on the Apache Hadoop infras- tructure. Oracle In-Database Hadoop framework is not built on top of actual Hadoop clusters.
- Greater efficiency in execution due to data pipelining, as Oracle knows more.
- Seamless integration of MapReduce functionality with Oracle SQL.

# Old Cup

Contains 30 cl

# SIGMOD Mug!


Contains 82 cl

# Conclusion

30 cl vs. 82 cl = 275% more coffee

**Conclusion**

SIGMOD made me 275% more efficient!! ;)

# End of Presentation

Thank You For Listening