Tackling The Challenges of Big Data

Big Data Storage

Michael Stonebraker

Professor

Massachusetts Institute of Technology

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Introduction

Michael Stonebraker

Professor

Massachusetts Institute of Technology

History Lesson

- **1970's: relational model invented**
- **1984: DB2 released, RDBMS declared mainstream**
- **Circa 1990: RDBMS takes over**
  - "One-size fits all"
  - I'm the guy with the hammer; everything is a nail
- **2006: ICDE paper**
  - "One-size does not fit all"
  - Co-existence of several solutions
- **2013: One size fits none**

Traditional RDBMS Wisdom

- **Dynamic row-level locking**
- **Aries-style write-ahead log**
- **Replication (asynchronous or synchronous)**
  - Update the primary first
  - Then move the log to other sites
  - And roll forward at the secondary(s)

Traditional RDBMS Wisdom

- **Data is in disk block formatting (heavily encoded)**
- **With a main memory buffer pool of blocks**
- **Query plans**
  - Optimize CPU, I/O
  - Fundamental operation is read a row
- **Indexing via B-trees**
  - Clustered or unclustered

My Thesis

Current RDBMSs (the elephants)

- All date from the 1980's
- Are legacy systems
- Are currently not good at anything
- Suffer from "The Innovators Dilemma"
- Deserve to be sent to the home for tired software

Rest of this Module

- **I will explain why one size fits none**
- **Three main DBMS markets**
  - One-third data warehouses
  - One-third OLTP
  - One-third everything-else
- **Some conclusions at the end**

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Introduction

THANK YOU

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Data Warehouses

Michael Stonebraker

Professor

Massachusetts Institute of Technology

Data Warehouse Marketplace

- **Column stores are well along at replacing row stores**
- **Because they are a factor of 50 – 100 faster**

Why???

- **Most warehouses have a central fact table**
  - Who bought what item in what store at what time.
- **Surrounded by "dimension" tables**
  - Store, time, product, customer, …
- **So-called "star/snowflake schema"**
  - Check out anything written by Ralph Kimball for lots of detail

Why???

- **Typical warehouse query reads 4-5 attributes from a 100 column fact table**
  - Row store – reads all 100
  - Column store – reads just the ones you need
- **Compression is way easier and more productive in a column store**
  - Each block has only one kind of attribute

Why???

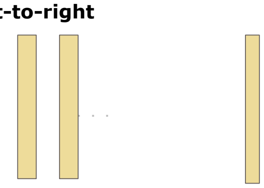
- **No big record headers in a column store**
  - They don't compress well
- **A column executor is wildly faster than a row executor**
  - Because of "vector processing"
  - See pioneering paper by Martin Kersten on this topic

The Participants

- **Native column store vendors**
  - HP/Vertica, SAP/Hana, Paraccel (Amazon), SAP/Sybase/IQ
- **Native row store vendors**
  - Microsoft, Oracle, DB2, Netezza
- **In transition**
  - Teradata, Asterdata, Greenplum

Three Slides on Vertica

Table is decomposed into a collection of materialized views, stored by column and sorted on all attributes left-to-right



Three Slides on Vertica

- A column is stored in 64K "chunklets". 1st attribute is stored uncompressed, remainder are compressed (delta compression, lempel-zipf, repeated values, huffman, …)
- Left-most column is usually delta encoded
- Chunklets are decompressed only when necessary
- Fundamental operation is "process a column"

Three Slides on Vertica

- To load fast, there is a main memory row-store in front of this column store.
  - Newly loaded tuples go there
  - In bulk, groups of rows are sorted, converted to column format and compressed
  - And written to new disk segments
  - Segment merge makes these segments bigger and bigger
  - Queries go to both places

Roughly Speaking

- This architecture also describes Paraccel and Hana
- It has nothing to do with the traditional RDBMS wisdom
- Over time the only successful warehouse products will be column stores
- The elephants have an "Innovator's Dilemma" problem

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Data Warehouses

THANK YOU

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

OLTP

Michael Stonebraker

Professor

Massachusetts Institute of Technology

OLTP Data Bases -- 3 Big Decisions

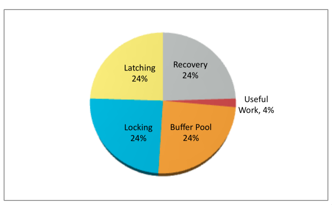
- **Main memory vs disk orientation**
- **Replication strategy**
- **Concurrency control strategy**

Reality Check on OLTP Data Bases

- **TP database size grows at the rate transactions increase**
- **1 Tbyte is a really big TP data base**
- **1 Tbyte of main memory buyable for around $30K (or less)**
  - (say) 64 Gbytes per server in 16 servers
- **If your data doesn't fit in main memory now, then wait a couple of years and it will…..**

Reality Check – Main Memory Performance

- **TPC-C CPU cycles**
- **On the Shore DBMS prototype**
- **"Elephants" should be similar**



To Go Fast

- **Must focus on overhead**
  - Better B-trees affects a small fraction of the path length
- **Must get rid of all four pie slices**
  - Anything less gives you a marginal win
  - Times10 as an example

Single Threading

- **Toast unless you do this**
  - Unless you get rid of queuing (somehow)
  - Or eliminate shared data structures (somehow)
- **H-Store (and VoltDB) statically divide shared memory among the cores**
  - Would be interesting to look at more flexible schemes

Main Memory

- **Again, you're toast unless you do this**
- **What happens if my data doesn't fit?**
  - See VLDB '14 paper by Debrabant et. al.

Concurrency Control

- **MVCC popular (NuoDB, Hekaton)**
- **Time stamp order popular (H-Store/VoltDB)**
- **Lightweight combinations of time stamp order and dynamic locking (Calvin, Dora)**
- **I don't know anybody who is doing normal dynamic locking**
  - It's too slow!!!!

What about Logging?

- **Command logging much faster than data logging**
  - See ICDE '14 paper by Malviya
- **HA is now a requirement**
  - Failover to a replica; rarely recover from a log

The Old Way vs The New Way

- **Main memory not disk**
- **Anti-caching not caching**
- **Command logging not data logging**
- **Failover not recovery from a log**
- **MVCC or timestamp order not dynamic locking**
- **Single threaded not multi-threaded**

New Way Systems

- **Hekaton (Microsoft)**
- **Hana (SAP)**
- **VoltDB, MemSQL, SQLFire, …**

Summary

- New is a factor of 100 or so faster than old
- If you don't care about performance, then stay with the elephants
- Otherwise, a changeover is in your future

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

OLTP

THANK YOU

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Everything Else

Michael Stonebraker

Professor

Massachusetts Institute of Technology

Everything Else

- NoSQL
- Array stores
- GraphDBMSs
- Hadoop

NoSQL – 75 or so Vendors

- **Give up SQL**
  - Completely misguided
  - SQL is compiled (at compile time) into the low level utterances of the NoSQL folks
  - Nobody codes in assembler any more!!!
  - Never bet against the compiler!!

NoSQL – 75 or so Vendors

- **Give up ACID**
  - If you are guaranteed that you won't need it (now or in the future) then you are ok
  - Otherwise, your hair will be on fire

NoSQL – 75 or so Vendors

- **Schema later**
  - Most support semi-structured data – adding a new "column" is trivial
  - Don't have to think about your data upfront
    - **Good or bad depending on your point of view**

NoSQL – Summary

- **Moving quickly toward SQL**
  - Cassandra and MongoDB are moving to (yup) SQL
- **Moving toward ACID**
  - Even Jeff Dean (Google) now admits ACID is a good idea
- **NoSQL**
  - Used to mean "No SQL"
  - Then meant "Not only SQL"
  - Moving toward "Not yet SQL" (i.e. convergence)

NoSQL – Summary

- **Systems are fine for "low end" applications**
  - E.g webby things
  - E.g. protection/ authentication data bases
  - Etc.

Array DBMSs and Complex Analytics

- Machine learning
- Data clustering
- Predictive models
- Recommendation engines
- Regressions
- Estimators

i.e. "Data Mining"

Complex Analytics

- By and large, they are defined on arrays
- As collections of linear algebra operations
- They are not in SQL!
- And often
  - Are defined on large amounts of data
  - And/or in high dimensions

Complex Analytics on Array Data – An Accessible Example

- Consider the closing price on all trading days for the last 20 years for two stocks A and B
- What is the covariance between the two time-series?

  * (1/N) * sum (Ai - mean(A)) * (Bi - mean (B))

Now Make It Interesting …

- **Do this for all pairs of 15000 stocks**
  - The data is the following 15000 x 4000 matrix

Stock	t_1	t_2	t_3	t_4	t_5	t_6	t_7	….	t_{4000}
S_1									
S_2									
…									
S_{15000}									

Array Answer

- **Ignoring the (1/N) and subtracting off the means ….**

Stock * StockT

Tackling the Challenges of Big Data © 2014 Massachusetts Institute of Technology

System Requirements

- **Complex analytics**
  - Covariance is just the start
  - Defined on arrays!!
- **Data management**
  - Leave out outliers
  - Just on securities with a market cap over $10B
- **Scalability to many cores, many nodes and out-of-memory data**

Tackling the Challenges of Big Data © 2014 Massachusetts Institute of Technology

Array DBMSs -- e.g. SciDB

- **Array SQL**
  - For joins filters,…
- **Built in functions**
  - For SVD, Co-variance, eigenvalues,…
- **User-defined extensions**
  - If you don't see what you need

Array DBMSs -- Summary

- **Will get tractions**
  - When the world moves to complex analytics
- **Don't look at all like the traditional wisdom**

Graph DBMSs

- **Focus on things like Facebook/twitter graphs**
- **OLTP focus (Neo4J)**
- **Analytics focus (shortest path, minimum cut set, …)**
- **Can you beat**
  - RDBMS simulations
  - Array simulations
- **Jury is still out**

Tackling The Challenges of Big Data
Big Data Storage
Modern Databases
Everything Else

THANK YOU

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data
Big Data Storage
Modern Databases
Hadoop

Michael Stonebraker
Professor
Massachusetts Institute of Technology

What is Hadoop?

- **Open source version of Google's Map-Reduce**
- **Two operations**
  - Map (basically filter, transform)
  - Reduce (basically rollup)
- **Very good for "embarrassingly parallel" operations**
  - E.g. document search

The Hadoop Stack

- **Hive (or Pig) at the top**
  - Think SQL
- **Hadoop (Map-Reduce) in the middle**
- **HDFS (a file system) at the bottom**
- **Runs across any number of nodes**
  - Scalable!

Possible Uses for Hadoop Stack

- **Embarrassingly parallel computations**
- **SQL aggregates (e.g. warehouse-style queries)**
  - Factor of 100 worse than a warehouse DBMS
- **Complex analytics**
  - Factor of 100 worse than an array DBMS
- **Scientific codes (e.g. computational fluid dynamics)**
  - Factor of 100 worse than MPI-based systems

Hadoop Usage at Facebook

- **95+% Hive**
  - For which Hadoop layer is a disaster

What is Happening Now?

- **Cloudera, Hortonworks and Facebook are ALL doing the same thing**
  - Defining and building an execution engine that processes Hive without using Hadoop layer
- **Effectively moving to compete in the warehouse market**
  - All warehouse vendors have Hive interfaces

Most Likely Future

- **There is a small market for embarrassingly parallel Hadoop framework**
- **There is a much bigger market for a Hive-SQL framework**
  - Execution engines will look like data warehouse products
- **HDFS may or may not survive**
  - It is also horribly inefficient

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Hadoop

THANK YOU

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Summary

Michael Stonebraker

Professor

Massachusetts Institute of Technology

Thoughts While Shaving

- **Warehouses will be a column store market**
  - If you are not running one now, you will have to switch
  - Ask your vendor what his column store plans are
- **OLTP will be a main memory market**
  - If you are not running one now, you will have to switch
  - Ask your vendor what his main memory plans are

Thoughts While Shaving

- **Array DBMSs and Graph DBMS may get traction**
  - You should (at the very least) understand what they are good for
- **NoSQL**
  - Is popular for low-end applications
  - Especially document management, web stuff and places where you want schema-later
  - ACID-less

Thoughts While Shaving

- **The Hadoop stack will morph into something completely different**
  - Hold onto your seat belt!!
  - At the very least -- see if you are contemplating embarrassingly parallel applications – if not, you are in deep doo-doo
- **Current elephant products will only survive long-term in low performance applications**

The Curse -- May You Live in Interesting Times

- **Lots of new DBMS ideas and products!!!**
- **BI folks will keep more and more stuff**
  - Warehouses will get bigger and bigger
- **Sea change from simple analytics to complex analytics expected**
- **The "internet of things" is a force to be dealt with**
  - i.e. everything on the planet of material significance will be sensor-tagged -- generating yet more data deluge

The Curse -- May You Live in Interesting Times

Hire a really really good chief data officer to help you sort out the future

Tackling The Challenges of Big Data

Big Data Storage

Modern Databases

Summary

THANK YOU

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data

Big Data Storage

THANK YOU

PROFESSIONAL EDUCATION

CSAIL