Big Data, Disruption and the 800 Pound Gorilla in the Corner

Michael Stonebraker

DBg Database Group
MIT Computer Science and Artificial Intelligence Lab

The Meaning of Big Data - 3 V's

- Big Volume
  - Business intelligence – simple (SQL) analytics
  - Data Science -- complex (non-SQL) analytics
- Big Velocity
  - Drink from a fire hose
- Big Variety
  - Large number of diverse data sources to integrate

Big Volume - Little Analytics

- Well addressed by the data warehouse crowd
  - Multi-node column stores with sophisticated compression
- Who are pretty good at SQL analytics on
  - Hundreds of nodes
  - Petabytes of data

But All Column Stores are not Created Equal…

- Performance among the products differs by a LOT
- Maturity among the products differs by a LOT
- Oracle is not multi-node and not a column store
- Some products are native column stores; some are converted row stores
- Some products have a serious marketing problem

Possible Storm Clouds

- NVRAM
- Networking no longer the “high pole in the tent”
- All the money is at the high end
  - Vertica is free for 3 nodes; 1 Tbyte

- Modest disruption, at best….
  - Warehouses are getting bigger faster than resources are getting cheaper

The Big Disruption

- Solving yesterday's problem!!!!
  - Data science will replace business intelligence
  - As soon as we can train enough data scientists!
  - And they will not be re-treaded BI folks
- After all, would you rather have a predictive model or a big table of numbers?

Data Science Template

Until (tired) {

Data management;

Complex analytics (regression, clustering, bayesian analysis, …);

}

Data management is SQL,
complex analytics is (mostly) array-based!

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?

$$(1/N) * \text{sum}\ (A_i - \text{mean}(A)) * (B_i - \text{mean}\ (B))$$

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 ….

$$Stock * Stock^{T}$$

How to Support Data Science (1st option)

- Code in Map-Reduce (Hadoop) for HDFS (file system) data
  - Drink the Google Koolaid

Map-Reduce

- 2008: The best thing since sliced bread
  - According to Google
- 2011: Quietly abandoned by Google
  - On the application for which it was purpose-built
  - In favor of BigTable
  - Other stuff uses Dremmel, Big Query, F1,…
- 2015: Google officially abandons Map-Reduce

Map-Reduce

- 2013: It becomes clear that Map-Reduce is primarily a SQL (Hive) market
  - 95+% of Facebook access is Hive
- 2013: Cloudera redefines Hadoop to be a three-level stack
  - SQL, Map-Reduce, HDFS
- 2014: Impala released; not based on Map-Reduce
  - In effect, down to a 2-level stack (SQL, HDFS)
  - Mike Olson privately admits there is little call for Map-Reduce
- 2014: But Impala is not even based on HDFS
  - A slow, location-transparent file system gives DBMSs severe indigestion
  - In effect, down to a one-level stack (SQL)

The Future of Hadoop

- The data warehouse market and Hadoop market are merging
  - May the best parallel SQL column stores win!

- HDFS is being marketed to support “data lakes”
  - Hard to imagine big bucks for a file system
  - Perfectly reasonable as an Extract-Transform and Load platform (stay tuned)
  - And a “junk drawer” for files (stay tuned)

How to Support Data Science (2nd option -- 2015)

- For analytics, Map-Reduce is not flexible enough
- And HDFS is too slow
- Move to a main-memory parallel execution environment
  - Spark – the new best thing since sliced bread
  - IBM (and others) are drinking the new koolaid

Spark

- No persistence -- which must be supplied by a companion storage system
- No sharing (no concept of a shared buffer pool)
- 70% of Spark is SparkSQL (according to Matei)
  - Which has no indexes
- Moves the data (Tbytes) to the query (Kbytes)
  - Which gives DBMS folks a serious case of heartburn

- What is the future of Spark? (stay tuned)

How to Support Data Science (3rd option)

- Move the query to the data!!!!!
  - Your favorite relational DBMS for persistence, sharing and SQL
- But tighter coupling to analytics
  - through user-defined functions (UDFs)
  - Written in Spark or R or C++ …
- UDF support will have to improve (a lot!)
  - To support parallelism, recovery, …
- But…..
  - Format conversion (table to array) is a killer
  - On all but the largest problems, it will be the high pole in the tent

How to Support Data Science (4th option)

- Use an array DBMS
- With the same in-database analytics

- No table-to-array conversion
- Does not move the data to the query
- Likely to be the most efficient long term solution

- Check out SciDB; check out SciDB-R

The Future of Complex Analytics, Spark, R, and ….

- Hold onto your seat belt
  - 1st step; DBMSs as a persistence layer under Spark
  - 2nd step; ????
- "The wild west"
- Disruption == opportunity
- What will the Spark market look like in 2 years????
  - My guess: substantially different than today

Big Velocity

- Big pattern - little state (electronic trading)
  - Find me a ‘strawberry’ followed within 100 msec by a ‘banana’
- Complex event processing (CEP) (Storm, Kafka, StreamBase …) is focused on this problem
  - Patterns in a firehose

Big Velocity – 2nd Approach

- Big state - little pattern
  - For every security, assemble my real-time global position
  - And alert me if my exposure is greater than X
- Looks like high performance OLTP
  - NewSQL engines (VoltDB, NuoDB, MemSQL …) address this market

In My Opinion….

- Everybody wants HA (replicas, failover, failback)
- Many people have complex pipelines (of several steps)
- People with high-value messages often want “exactly once” semantics over the whole pipeline
- Transactions with transactional replication do exactly this

- My prediction: OLTP will prevail in the “important message” market!

Possible Storm Clouds

- RDMA – new concurrency control mechanisms
- Transactional wide-area replicas enabled by high speed networking (e.g. Spanner)
  - But you have to control the end-to-end network
  - To get latency down

- Modest disruption, at best

Big Variety

- Typical enterprise has 5000 operational systems
  - Only a few get into the data warehouse
  - What about the rest?
- And what about all the rest of your data?
  - Spreadsheets
  - Access data bases
- And public data from the web?

Traditional Solution -- ETL

- Construct a global schema
- For each local data source, have programmer
  - Understand the source
  - Map it to the global schema
  - Write a script to transform the data
  - Figure out how to clean it
  - Figure out how to “dedup” it
- Works for 25 data sources. What about the rest?

Who has More Data Sources?

- Large manufacturing enterprise
  - Has 325 procurement systems
  - Estimates they would save $100M/year by “most favored nation status”
- Large drug company
  - Has 10,000 bench scientists
  - Wants to integrate their “electronic lab notebooks”
- Large auto company
  - Wants to integrate customer databases In Europe
  - In 40 languages

Why So Many Data Stores?

- Enterprises are divided into business units, which are typically independent
  - For business agility reasons
  - With independent data stores
- One large money center bank had hundreds
  - The last time I looked

And there is NO Global Data Model

- Enterprises have tried to construct such models in the past…..
  - Multi-year project
  - Out-of-date on day 1 of the project, let alone on the proposed completion date
- Standards are difficult
  - Remember how difficult it is to stamp out multiple DBMSs in an enterprise
  - Let alone Macs…

Why Integrate Silos?

- Cross selling
- Combining procurement orders
  - To get better pricing
- Social networking
  - People working on the same thing
- Rollups/better information
  - How many employees do we have?
- Etc….

Data Curation/Integration

- Ingest
- Transform (euros to dollars)
- Clean (-99 often means null)
- Schema map (your salary is my wages)
- Entity consolidation (Mike Stonebraker and Michael Stonebraker are the same entity)

Why is Data Integration Hard?

- Bought $100K of widgets from IBM, Inc.
- Bought 800K Euros of m-widgets from IBM, SA
- Bought -9999 of *wids* from 500 Madison Ave., NY, NY 10022

- Insufficient/incomplete meta-data: May not know that 800K is in Euros
- Missing data: -9999 is a code for “I don’t know”
- Dirty data: *wids* means what?

Why is Data Integration Hard?

- Bought $100K of widgets from IBM, Inc.
- Bought 800K Euros of m-widgets from IBM, SA
- Bought -9999 of *wids* from 500 Madison Ave., NY, NY 10022

- Disparate fields: Have to translate currencies to a common form
- Entity resolution: Is IBM, SA the same as IBM, Inc.?
- Entity resolution: Are m-widgets the same as widgets?

Data Integration (Curation) AT SCALE is a VERY Big Deal

- Biggest problem facing many enterprises
- 800 pound gorilla in the corner!

A Bunch of Startups With New Ideas

- Tamr
- Trifacta
- Paxata
- Alteryx
- Cambridge Semantics
- Clear Story
- …

To Achieve Scalability….

- Must pick the low-hanging fruit automatically
  - —Machine learning
  - —Statistics
- Rarely an upfront global schema
  - —Must build it “bottom up”
- Must involve human (non-programmer) experts to help with the cleaning

Tamr is an example of this approach

Data Lakes

- Solve only the ingest problem
- Which is at most 5% of the problem
  - Leaving the remaining 95% unsolved
- Generates a data swamp not a data lake
  - Enterprise junk drawer

Take away

- Look for disruption points
  - Opportunity!
- Look for pain
  - The 800 pound gorilla