

What Does 'Big Data' Mean and Who Will Win?

Michael Stonebraker

The Meaning of Big Data - 3 V's

- Big Volume
 - With simple (SQL) analytics
 - With complex (non-SQL) analytics
- Big Velocity
 - Drink from the fire hose
- Big Variety
 - Large number of diverse data sources to integrate

Big Volume - Little Analytics

- Well addressed by data warehouse crowd
- Who are pretty good at SQL analytics on
 - Hundreds of nodes
 - Petabytes of data

The Participants

- Row storage and row executor
 - Microsoft Madison, DB2, Netezza, Oracle(!)
- Column store grafted onto a row executor (wannabees)
 - Terradata/Asterdata, EMC/Greenplum
- Column store and column executor
 - HP/Vertica, Sybase/IQ, Paracel

Oracle Exadata is not:

a column store

a scalable shared-nothing architecture

Performance

- Row stores -- x1
- Column stores -- x50
- Wannabees -- x5 (???)

Big Data - Big Analytics

- Complex math operations (machine learning, clustering, trend detection,)
 - In your market, the world of the “quants”
 - Mostly specified as linear algebra on array data
- A dozen or so common ‘inner loops’
 - Matrix multiply
 - QR decomposition
 - SVD decomposition
 - Linear regression

Big Data - Big Analytics

An Example

- Consider closing price on all trading days for the last 5 years for two stocks A and B
- What is the covariance between the two time-series?

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

Now Make It Interesting ...

- Do this for all pairs of 4000 stocks
 - The data is the following 4000 x 1000 matrix

Stock	t_1	t_2	t_3	t_4	t_5	t_6	t_7	...	t_{1000}
S_1									
S_2									
...									
S_{4000}									

Hourly data? All securities?

Array Answer

- Ignoring the $(1/N)$ and subtracting off the means

$$\text{Stock} * \text{Stock}^T$$

- Now try it for companies headquartered in Charlotte!

Goal

- Good data management
- Integrated with complex analytics
 - Specified as arrays, not tables

Solution Options

- SAS et. al
 - Weak or non-existent data management
- SAS plus RDBMS
 - No integration
- RDBMS plus user-defined functions
 - Slowwww (X10 to X100)
- Array DBMS
 - Check out SciDB.org

Hadoop.....

- Simple analytics
 - X100 times a parallel DBMS
- Complex analytics (Mahout or roll-your-own)
 - X100 times Scalapack
- Parallel programming
 - Parallel grep (great)
 - Everything else (awful)
- Hadoop lacks
 - Stateful computations
 - Point-to-point communication

Big Velocity

- Trading volume on Wall Street going through the roof
- Breaking all their infrastructure
- And it will just get worse

Big Velocity

- Sensor tagging everything of value sends velocity through the roof
 - E.g. car insurance
- Smart phones as a mobile platform sends velocity through the roof
- State of multi-player internet games must be recorded - sends velocity through the roof

Two Different Solutions

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

P.S. I started StreamBase but I have no current relationship with the company

Two Different Solutions

- 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
 - Want to update a database at very high speed

My Suspicion

- You have 3-4 Big state - little pattern problems for every one Big pattern - little state problem

New OLTP

- You need to **ingest** a fire hose in real-time
- You need to perform high volume OLTP
- You often need **real-time** analytics



Solution Choices

- Old SQL
 - The elephants
 - Slowwww (X 50)
 - Non-starter
- No SQL
 - 75 or so vendors giving up both SQL and ACID
- New SQL
 - Retain SQL and ACID but go fast with a new architecture

No SQL

- Give up SQL
 - Interesting to note that Cassandra and Mongo are moving to (yup) SQL
- Give up ACID
 - If you need ACID, this is a decision to tear your hair out by doing it in user code
 - Can you guarantee you won't need ACID tomorrow?



VoltDB: an example of New SQL

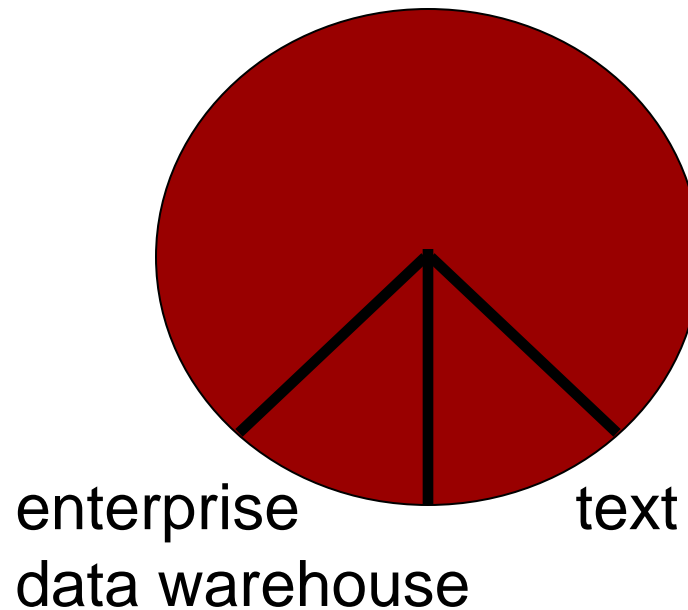
- A main memory SQL engine
- Open source
- Shared nothing, Linux, TCP/IP on jelly beans
- Light-weight transactions
 - Run-to-completion with no locking
- Single-threaded
 - Multi-core by splitting main memory
- About 100x RDBMS on TPC-C

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
 - Web pages
- And public data from the web?

The World of Data Integration

the rest of your data



Summary

- The rest of your data (public and private)
 - Is a treasure trove of incredibly valuable information
 - Largely untapped

Data Tamer

- Integrate the rest of your data
- Has to
 - Be scalable to 1000s of sites
 - Deal with incomplete, conflicting, and incorrect data
 - Be incremental
 - Task is never done

Data Tamer in a Nutshell

- Apply machine learning and statistics to perform automatic:
 - Discovery of structure
 - Entity resolution
 - Transformation
- With a human assist if necessary
 - WYSIWYG tool (Wrangler)

Data Tamer

- MIT research project
- Looking for more integration problems
 - Wanna partner?

Take away

- One size does not fit all
- Plan on (say) 6 DBMS architectures
 - Use the right tool for the job
- Elephants are not competitive
 - At anything
 - Have a bad ‘innovator’s dilemma’ problem