One-Size-Fits-All: A DBMS Idea Whose Time has Come and Gone

Michael Stonebraker
December, 2008
One Size Fits All (OSFA)

It’s too hard for them to maintain multiple code bases for different specialized purposes

* engineering problem
* sales problem
* marketing problem
The OSFA Elephants

• Sell code lines that date from the 1970’s
  – Legacy code
  – Built for very different hardware configurations
  – And some cannot adapt to grids….

• That was designed for business data processing (OLTP)
  – Only market back then
  – Now warehouses, science, real time, embedded, ..
• Store fields in one record contiguously on disk
• Use B-tree indexing
• Use small (e.g. 4K) disk blocks
• Align fields on byte or word boundaries
• Conventional (row-oriented) query optimizer and executor
Terminology -- “Row Store”

E.g. DB2, Oracle, Sybase, SQLServer, Greenplum, Netezza, DatAllegro, Datupia, …
At This Point, RDBMS is “long in the tooth”

• There are at least 6 (non trivial) markets where a row store can be clobbered by a specialized architecture

  – Warehouses (Vertica, SybaseIQ, KX, …)
  – OLTP (H-Store)
  – RDF (Vertica et. al.)
  – Text (Google, Yahoo, …)
  – Scientific data (MatLab, ASAP prototype)
  – Streaming data (StreamBase Coral8, …)
Definition of “Clobbered”

• A factor of 50 in performance
Current DBMSs

• 30 years of “grow only” bloatware
• That is not good at anything
• And that deserves to be sent to the “home for tired software”
The DBMS Landscape – Performance Needs

OLTP

Data Warehouse

Other apps

high

low

high

high

OLTP
One Size Does Not Fit All -- Pictorially

ASAP, etc

Open source

Vertica/C-Store

H-Store

Elephants get only “the crevices”
Stonebraker’s Prediction

- The DBMS market will move over the next decade or so from OSFA
  - To specialized (market-specific) architectures
  - And open source systems
- Presumably to the detriment of the elephants
A Couple of Slides of Color on Some of the Markets

Data warehouses
OLTP
Scientific and intelligence data
C-Store prototype (2004-5)
Commercialized by Vertica Systems (2005)
Data Warehouses – Column Stores are the Answer

Column Store:

IBM  60.25  10,000  1/15/2006
MSFT 60.53  12,500  1/15/2006

Used in: Sybase IQ, Vertica

Row Store:

IBM  60.25  10,000  1/15/2006
MSFT 60.53  12,500  1/15/2006

Used in: Oracle, SQL Server, DB2, Netezza,…
• Read only what you need
  • “Fat” fact tables are typical
  • Analytics read only a few columns
• Better compression
• Execute on compressed data
• Materialized views help row stores and column stores about equally
Example of “Clobber”

- Vertica on an 2 processor system costing ~$2K
- Netezza on a 112 processor system costing ~$1M
- Customer load time benchmark
  - Vertica 2.8 times faster – per processor/disk
- Customer query benchmark
  - Vertica 34X on 1/56th the hardware (factor of 1904)
Other Examples

- C-store paper (VLDB ’05)
- Vertica has run about 50 benchmarks
  - Against all comers
  - Yet to win by less than a factor of 20 against a row store
  - About an order of magnitude better than other column stores
  - Only thing that comes close is KX
Things to Demand From ANY BI DBMS

- Scalable
  - Runs on a grid, with partitioning
- Replication for HA/DR
- “no knobs” operation (more than index selection)
  - Cannot hire enough DBAs
- On-line update – in parallel with query
- Ability to run multiple analyses on compatible data
  - Time travel
- On-the-fly reprovisioning
OLTP – The Big Picture

• Where the time goes (TPC-C) (Sigmod ’07)
  – 25% -- the buffer pool
  – 25% -- locking
  – 25% -- latching
  – 25% -- recovery
  – 2% -- useful work

• Have to focus on overhead, not on algorithms or data structures
Introducing VoltDB

• Based on H-Store collaboration between: MIT, Brown, Yale & Vertica Systems

• An innovative database management system purpose-built for:
  – Performance on OLTP Workloads
  – Scalability
  – High availability
  – Low cost of entry
  – Low cost of administration
VoltDB Assumptions

• Main memory operation
  • 1 TB is a VERY big OLTP data base
  • No disk stalls
• No user stalls (disallowed in all apps)
• Run transactions to completion
  • Single threaded
  • Eliminate “latch crabbing”
  • And locking
VoltDB Assumptions

- Built-in high availability and disaster recovery
  - Failover to a replica
  - No redo log
VoltDB Assumptions – Most Transactions are single-sited

• Simple transactions are naturally single-sited:
  – Place my order
  – Read my reservation
  – Update my user information
• Other transactions can be made single sited though design
  – Replicate read-mostly data to all grid cells
  – Break transactions into separate read & write transactions
  – We know other tricks as well
OLTP Performance

• Elephant
  • 850 TPS (1/2 the land speed record per processor)

• H-Store
  • 70,416 TPS (41X the land speed record per processor)

• VoltDB
  • ~10,000 TPS
VoltDB Summary

- No buffer pool overhead
  - There isn’t one
- No crash recovery overhead
  - Done by failover
  - (optional) Asynchronous data transmission to reporting system
  - (optional) Asynchronous local data archive
- No latching or locking overhead
  - Transactions are run to completion – single threaded
Scientific Data – Array Storage

- Factor of 100 penalty to simulate arrays on top of tables
Why SciDB?

• Net result
  • Mentality of “roll your own from the ground up” for every new science project
  • Realization by the science community that this is long-term suicide
• Community wants to get behind something better
  • Great commonality of needs among domains
Our Partnership

• Science and high-end commercial folks
  • Who will put up some resources
  • And review design

• DBMS brain trust
  • Who will design the system, oversee its construction, and perform needed research

• Non-profit company
  • Which will manage the open source project
  • And support the resulting system
  • May need long term funding help
The SciDB Data Model

• Tables?
  • Makes a few of you happy
  • Used by Sloan Sky Survey
• But
  • PanStarrs (Alex Szalay) wants arrays and scalability
The SciDB Data Model

• Arrays?
  • Superset of tables (tables with a primary key are a 1-D array)
  • Makes HEP, remote sensing, astronomy, oceanography folks happy

• But
  • Not biology and chemistry (who wants networks and sequences)
Other Features
Which Science Guys Want
(These could be in RDBMS, but Aren’t)

• Uncertainty
  • Data has error bars
  • Which must be carried along in the computation
    (interval arithmetic)
  • Will look at more sophisticated error models later
Other Features

• Provenance (lineage)
  • What calibration generated the data
  • What was the “cooking” algorithm
  • In general – repeatability of data derivation

• Supported by a command log
  • with query facilities (interesting research problem)
  • And redo
Other Features

- Named versions
- No overwrite
  - Keep all the data
Time Line

• Q4/08
  • start company, begin research activities
• Late 2009
  • Demoware available
• Late 2010
  • V1 ships
SciDB Has a Good Chance at Success

- Community realizes shared infrastructure is good
- “Lighthouse” customers
- Strong team
- Computation goes inside the DBMS
  - Easier to share
  - And reuse
Summary

- Vertica
- VoltDb
- SciDB
  - Special purpose
  - fast