Scalable Complex Analytics and DBMSs

Michael Stonebraker

DBg Database Group
MIT Computer Science and Artificial Intelligence Lab

Simple Analytics

- SQL operations
  - count, sum, max, min, avg
  - Optional group_by
- Defined on tables
- User interface is Business Intelligence Tools
  - Cognos, Business Objects, …
- Appropriate for traditional business applications

Simple Analytics

- Well served by the data warehouse crowd
- Who are good at this stuff
  - even on petabytes

Complex Analytics

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

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?

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

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 ….

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

Use Case Requirements

- Complex analytics
  - Covariance is just the start
  - Defined on arrays
  - Graphs are just sparse 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

Data Scientist Job Description

- Ignore the 80 - 90% of the time spent cleaning and assembling the data
  - Separate talk on data curation

- Until (tired) {

  Data management operation(s);

  Complex analytics operations(s);

  }

Solution Options for Data Management

- Hard Code
  - Separate stack from the bare metal up for each project (LHS is 40M lines of code)
  - No uniform treatment of meta data (often encoded in the file name)
  - Can't share data easily
  - Depends on the "cheap PostDoc" model

Solution Options for Data Management

- Use a DBMS
  - Get sharing, indexing, protection, queries, crash recovery, ….
- Please, please, please use a DBMS
  - If you get nothing else from this talk, please take note of this!
  - Take a page from the business data processing playbook!
- Yabut – I can code a faster solution
  - But you are dooming your successor to maintaining it!
  - And requirements change!!!!

DBMS Options

- Traditional row store (Postgres, MySQL, Oracle, Big Table, ...)
  - Stores the data on disk row-by-row
  - Not competitive on data intensive queries
  - For a collection of very good technical reasons

DBMS Options

- Column store (Vertica, Red Shift, DB2-Blu, Impala, …)
  - Store the data column-by-column
  - Easier to compress; much faster executor; often read less than all columns
  - Generally 50 X row stores on this kind of stuff
- In the data warehouse market
  - This technology is in the process of completely taking over

DBMS Options

- Array store (SciDB, Rasdaman, HDF-5)
  - Data model is an array, not a table
  - Query language is typically array-SQL
  - Store the data in multi-dimensional tiles (chunks)
- Advantages
  - Same conceptual model as linear algebra
  - No table to array conversion required (which is very slow)
  - Dimensions are not stored (space advantage)
  - Multi-dimensional queries are very very fast, since the storage structure is “chunked”

Array Query Language (AQL)

```
SELECT Geo-Mean ( T.B )
FROM Test_Array T
WHERE
     T.I BETWEEN :C1 AND :C2
AND T.J BETWEEN :C3 AND :C4
AND T.A = 10
GROUP BY T.I;
```

User-defined aggregate on an attribute B in array T

Subsample

Filter

Group-by

DBMS Options

- Map-Reduce (open source version is Hadoop)
  - Good for embarassingly parallel problems only
  - Which this stuff is not!!!
  - Abandoned by Google in 2011 (or so)
  - Cloudera has a DBMS (Impala) – NOT built on Map-Reduce
- This interface is essentially dead

Two Things to Keep in Mind (1) (Data Base 101)

- Always send the query to the data (Kbytes)
  - Minimizes data comm
- Do not bring the data to the query (Tbytes)!
  - Forward pointer to HPC

Two Things to Keep in Mind (2)

- On matrix multiply, there are five orders of magnitude difference between Python and Intel-optimized C++
- Example
  - One order of magnitude between LaPack/BLAS/MKL and "smart Russians in C++"
  - Java is another order of magnitude down (Spark, Mahout, …)
- Very difficult to compete with optimized packages and Intel engineers!!!

Analytics Options

- Code in SQL
  - Matrix multiply is a 3-way self join
  - If the data is sparse enough, this may be ok
  - On dense data this will be a disaster (SQL and Python are likely to have similar performance)
- Madlib is a package that did this
  - And was quickly recoded in C++
- Bill Howe will probably have a different opinion
  - I suspect

Analytics Options (Loose Integration)

- Code in a stat package (R, SAS, SPSS, Mahout, …)
  - Copy the world from the DBMS to the package (slow)
  - Learn 2 interfaces
  - You're in the plumbing business!
  - Parallel packages are just coming into existence
  - Most stat packages are main-memory only
- I don't like this option at all!
  - Long term slog through the swamp

Analytics Options (Tight Integration)

- Run stat code as a user-defined function
  - Inside the DBMS
  - Called through extensions to SQL

Example Query

```
SELECT A.i * B.j
FROM A, B
WHERE
        A.k > 100 and
        B.m < 200
```

Analytics Options (Tight Integration)

- Learn one interface
- No “copy the world” problem
- Run stat code as a user-defined function
  - Inside the DBMS
  - Automatic parallelism (at least in SciDB)

(Some of the) Detailed Options

- Loose coupling
  - {R, SAS, SPSS} + your favorite DBMS
- Tight coupling
  - SciDB + Scalapack
  - SciDB + R
  - Vertica + R

A Note on Hadoop/HDFS

- Impala is not coded on top of HDFS
  - Drills through to underlying Linux files
  - Looks exactly like a parallel column store (e.g. Vertica, Redshift, …)
- "Hadoop market" and "data warehouse market" are converging
- Current marketing slogo is "data lakes"
  - Creates a data swamp by ignoring data curation issues
  - Or a junk drawer

A Note on Spark

- 70+% of Spark access is SparkSQL
- However, Spark has
  - No persistence
  - No meta data
  - No main memory sharing
  - Java (slow)
- I expect all of this to get fixed over time
  - And Spark will follow the trajectory of Hadoop to become a data warehouse market
- Remainder is Scala (slow)
  - Remains to be seen how Spark will play in the general distributed computing space….

Issues in Using ScalaPack in SciDB

- Block cyclic organization
  - which DBMS does not support
- MKL
  - Which DBMSs won't use for crash recovery issues
- Tile organization
  - Scalapack is dense-only
  - SciDB is a single format for dense and sparse

The Future

- Co-design of analytics and DBMS storage organization
  - To get rid of these issues
  - Intel-supported project at MIT and UTenn

An Exercise at NERSC

- General NERSC architecture is
  - A compute server
  - A storage server
  - A compute-side file cache; scheduled in advance

Issues

- DBMS wants to be “always on” service
  - Incompatible with scheduling the file cache
- Send the data to the query not the other way around
  - Every time somebody wants data access, need to move the world

At NERSC

- SciDB runs
  - Managing many, many Tbytes of data
  - On dedicated nodes
- Could not get Vertica to run at all
  - Painful aspects of batch job focus (scheduling the file cache; open file limit)

Summary

- Stand on the shoulders of those who went before you, not on their feet
  - Please don't write a complete stack for each new project
- Want to tightly couple DBMSs and linear algebra
  - Or you get 2 interfaces
  - And copy the world

Summary

- Array DBMSs are likely to be attractive
  - Check out SciDB.org
- Hadoop and Spark will probably morph into something that looks like a DBMS
  - Turkey performance in the meantime
- HPC needs to become interactive
  - Or DBMSs probably won't run there