MAD SKILLS: NEW ANALYSIS PRACTICES FOR BIG DATA

Jeffrey Cohen, Brian Dolan, Mark Dunlap, Joseph M. Hellerstein, Caleb Welton

November 10, 2015

presented by Ritwika Ghosh

mad (adj.): an adjective used to enhance a noun.

1. dude, you got skills.
2. dude, you got mad skills.

– UrbanDictionary.com

If you are looking for a career where your services will be in high demand, you should find something where you provide a scarce, complementary service to something that is getting ubiquitous and cheap.

So what's getting ubiquitous and cheap? Data.

And what is complementary to data? Analysis.

-Prof. Hal Varian, UC Berkeley, Chief Economist at Google

A bit of History

- Enterprise Data Warehouse(EDW) is queried by Business Intelligence(BI) software.
- A carefully constructed EDW was key.
- "Mission Critical, expensive resource, used for serving data intensive reports targeted at executive decision makers".

What has changed

- Super cheap storage.
- Massive-scale data sources in an enterprise has grown remarkably : everything is data
- Grassroots move to collect and leverage data in multiple organizational units : Rise of data driven culture espoused by Google, Wired etc.
- Sophisticated data analysis leads to cost savings and even direct revenue

MAD SKILLS

- New requirements : MAD Skills.
- M :

MAD SKILLS

- New requirements : MAD Skills.
- M : Magnetic (attract data and analysts)
- A :

MAD SKILLS

- New requirements : MAD Skills.
- M : Magnetic (attract data and analysts)
- A : Agile (rapid iteration)
- D :

MAD SKILLS

- New requirements : MAD Skills.
- M : Magnetic (attract data and analysts)
- A : Agile (rapid iteration)
- D : Deep (sophisticated analytics in Big Data)
- Analysts with MAD skills need to be complemented by MAD approaches to design and infrastructure.
-

This paper

- MAD analytics for Fox Interactive Media, using Greenplum .
- Data parallel statistical algorithms for modeling and comparing the densities of distribution.
- Critical database system features that enable agile design and flexible algorithm development.
- Challenging data warehousing orthodoxy :”Model Less, Iterate More”.

Fox Audience Network

- Serves ads across several Fox online publishers. (huge ad network).
- Greenplum Database system on 42 nodes:
  - 40 Sun X4500s for query processing,
  - 2 dual-core Opteron master nodes (one for failover).
- Big and Growing :
  - 200 TB of mirrored data. Fact table of 1.5T rows. (2009)
  - 5TB growth per day.
- Variety of data : Ad logs, CRM, User data.
- Diverse user set.
- Extensive use of R and Hadoop.

Fox Audience Network: Contd.

Diverse user base

Different needs, variety of reporting and statistical tools, command line access : Dynamic query ecosystem.

Dealing with ad-hoc questions

Question: : How many female WWF enthusiasts under the age of 30 visited the Toyota community over the last four days and saw a medium rectangle?

Problem : No set of pre-defined aggregates can possibly cover every question combining various variables.

Magnetic : Attracting users and Methods

Central Design Principle : Get data into the warehouse ASAP

- Analysts > DBAs : they like *all* data, they tolerate dirty data, they attract data, they produce data.
- Sandboxing allows analysts to feed datasets directly from main warehouse.
- Encourage novel data sources.
- Business > application.

Case Study: Audience Forecasting

3 million users login to IMDb.
2 million shared enough personal information to be able to attach 1 out of 2k attributes of behavior.
3 billion ads serving as tracking devices.

$$\text{Number of decisions} : 1.2 \times 10^{16}$$

Business cycle

Acquiring this data, strategically sub-sampling, determine scaling, change practices to suit : rinse and repeat.

Deep : learning from data

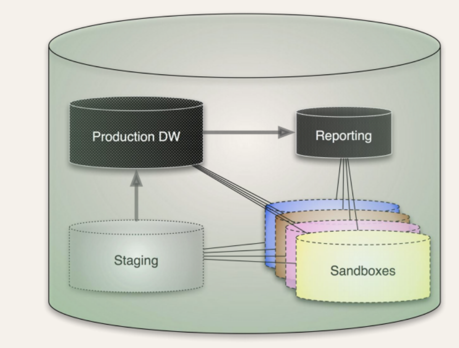
- Infinite cycles of drill down and roll up : No single number is the answer.
- Anomaly detection, longitudinal variance, distribution functions.
- Statistical modeling : curves and models, as opposed to points !

"If you don't reveal some insights soon, I'm going to be forced to slice, dice, and drill!"

MAD Modeling

Intelligently staging cleaning and integration of data

- Staging schema : raw fact tables/ logs
- Production Data Warehouse schema : aggregates for reporting tools and casual users.



Data Parallel statistics

- A hierarchy of mathematical concepts in SQL (MapReduce as well).
- Abstraction levels : Scalar $\rightarrow$ Vector $\rightarrow$ Function $\rightarrow$ Functional.
- Encapsulated as stored procedures and UDFs.
- Need to be able to use statistical vocabulary.

Vectors and Matrices

Let A and B be two matrices of identical dimensions. Matrix Addition:

```
SELECT A.row_number, A.vector + B.vector
  FROM A, B
WHERE A.row_number = B.row_number;
```

Multiplication of matrix and a vector *Av*:

```
SELECT 1, array_accum(row_number,vector*v) FROM A;
```

Matrix transpost of an $m \times n$:

```
SELECT S.col_number,
       array_accum(A.row_number, A.vector[S.col_number])
  FROM A, generate_series(1,3) AS S(col_number)
Group by S.col_number;
```

Matrix Multiplication

```
SELECT A.row_number, B.column_number,
       SUM(A.value * B.value)
FROM A, B
WHERE A.column_number = B.row_number
GROUP BY A.row_number, B.column_number
```

Example: tf-idf and Cosine similarity

Document similarity : Fraud detection

- Create triples of ($document, term, count$).
- Create marginals along *document* and *term* using group by queries.
- Expand each triple with a tf-idf score.
- Obtain cosine similarity of two document vectors x, y : $\theta = \frac{x.y}{||x||^2||y||^2}$

Let A have one row per document vector.

```
SELECT a1.row_id AS document_i, a2.row_id AS document_j,
 (a1.row_v * a2.row_v) /
((a1.row_v * a1.row_v) * (a2.row_v * a2.row_v)) AS theta
FROM a AS a1, a AS a2
WHERE a1.row_id > a2.row_id
```

Squares

Large dense matrices: distance matrix D, covariance matrices.

- OLS : modeling seasonal trends.
- Statistical estimate of β^* best satisfying $Y = X\beta$.
- $X = n \times k, Y = \{o_1, \ldots, o_n\}, \beta^* = (X'X)^{-1}X'y$.
- coefficient of determination:

$$SSR = b'\beta - \frac{1}{n}(\sum y_i)^2$$

$$TSS = (\sum y_i)^2 - \frac{1}{n}(\sum y_i)^2$$

$$R^2 = \frac{SSR}{TSS}$$

Routine to compute OLS

```
CREATE VIEW ols AS
    SELECT pseudo_inverse(A) * b as beta_star,
    (transpose(b) * (pseudo_inverse(A) * b)
      - sum_y2/count) -- SSR
      / (sum_yy - sumy2/n) -- TSS
          as r_squared
FROM (
  SELECT sum(transpose(d.vector) * d.vector) as A,
         sum(d.vector * y) as b,
         sum(y)^2 as sum_y2,
        sum(y^2) as sum_yy,
        count(*) as n
  FROM design d
) ols_aggs;
```

MAD DBMS

- Magnetic : painless and efficient data insertion.
- Agile : physical storage evolution easy and efficient.
- Deep : powerful flexible programming environment.

Conclusions

- Database is not proprietary hardware : parallel computation engine.
- Storage is not expensive, math is not hard.
- SQL is flexible and highly extensible.

Issues with Paper

- How are queries parallelized? If we write in R, its not automatic.
- MapReduce here vs Hadoop?
- Ad for Greenplum :)