facebook

Big Data Benchmark
The Cloudy Approach

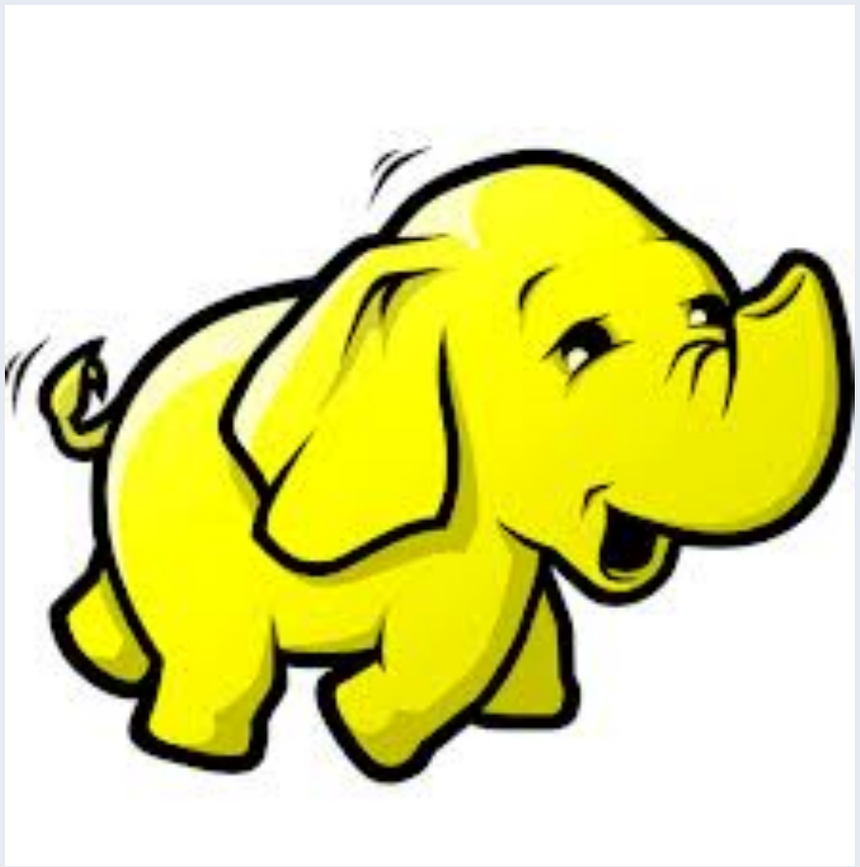
Dhruba Borthakur
Nov 7, 2012 at the **SDSC Industry Roundtable on Big Data and Big Value**

My Asks for a Cloudy Benchmark

1. BigData system is not a Traditional Database System
2. Metrics to compare BigData Systems
3. Measuring Value of a Big Data System

Use case for Hadoop@Facebook

- Analytics queries
  - 100PB + in a single cluster
  - 100+ million files, 50K Hive tables
  - 100K concurrent client
- Backups and Archival
  - Backups hundreds of MySQL databases into HDFS
- SLTP workload via Hbase
  - 6 billion messages/day



Analytics Data Growth(last 4 years)

	Facebook Users	Queries/Day	Scribe Data/ Day	Nodes in warehouse	Size (Total)
Growth	14X	60X	250X	260X	2500X

How to measure a BigData Service ?

What is BigData? Prospecting for Gold..

- "Finding Gold in the wild-west"
- A platform for huge data-experiments
- A majority of queries are searching for a single gold nugget
- Great advantage in keeping all data in one queryable system
- No structure to data, specify structure at query time

How to measure performance

- Traditional database systems:
  - Latency of queries
- Big Data systems:
  - How much data can we store and query? (the '**Big**' in BigData)
  - How much data can we query in parallel?
  - What is the value of this system?

Measure Cost of Storage

- **Distributed Network Encoding of data**
  - Encoding is better than replication
  - Use algorithms that minimize network transfer for data repair
- **Tradeoff cpu for storage & network**
  - Remember lineage of data, e.g. record query that created it
  - If data is not accessed for sometime, delete it
  - If a query occurs, recompute the data using query lineage

Measure Network Encoding

Start the same: triplicate every data block

Background encoding

- Combine third replica of blocks from a single file to create parity block
- Remove third replica
- Reed Solomon encoding for much older files

A file with three blocks A, B and C (XOR Encoding)

http://hadoopblog.blogspot.com/2009/08/hdfs-and-erasure-codes-hdfs-raid.html

Measuring Data Discovery: Crowd Sourcing

- There are 50K tables in a single warehouse
- Users are Data Adminstrators themselves
- Questions about a table are directed to users of that table
- Automatic query lineage tools

Measuring Testabilty

- **Traditional systems**
  - Recreate load using tests
  - Validate results
- **Big Data Systems**
  - Cannot replicate production load on test environment
  - Deploy new service on a small percentage of service
    - Monitor metrics
    - Rolling upgrades
  - Gradually deploy to larger section of service

Fault Tolerance and Elasticity

- Commodity machines
- Faults are the norm
- Anomalous behavior rather than complete failures
  - 10% of machines are always 50% slower than the others

Measuring Fault Tolerance and Elasticity

- **Fault tolerance is a must**
  - Continuously kill machines during benchmarking
  - Slow down 10% of machine during benchmark
- **Elasticity is necessary**
  - Add/remove new machines during benchmarking

Measuring Value of the System

- cost /GB decreasing with time
  - So users can store more data
  - But users need a metric to determine whether this cost is worth it
- What is the **VALUE** of this system?
  - A metric that aids the user (and not the service provider)

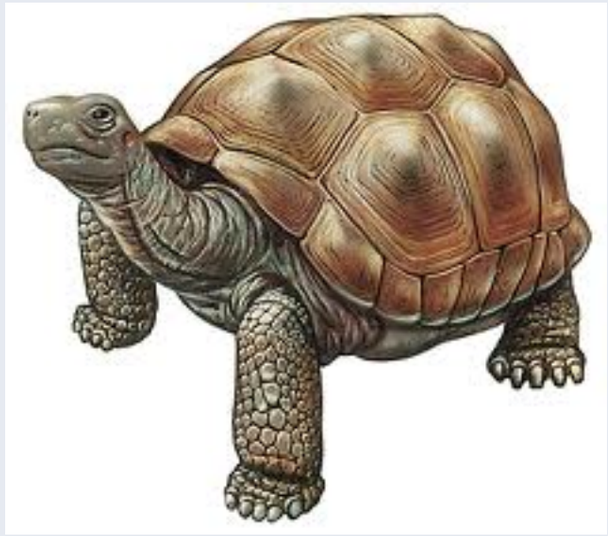
Value per Byte (VB) for the System

- A new metric named VB
  - Ability to compare differences in value ove time
  - If VB increases with time, then user is satisfied

- As soon as you touch a byte, its VB is MAX (say 100)
- System VB = weighted sum of VB of each byte in the system

VB – Even a turtle ages with time

- The VB decreases with time
  - A more recent access has more value than an older access
  - Different ageing models (linear, exponential)



Why use Hive instead of a Parallel DBMS?

- Stonebraker/DeWitt from the DBMS community:
  - Quote “major step backwards”
  - Published benchmark results which show that Hive is not as performant as a traditional DBMS
  - http://database.cs.brown.edu/projects/mapreduce-vs-dbms/
    - Hive query is 50 times slower than DBMS query
    - Conclusion: Facebook’s 4000 node cluster (100PB) can be replaced by a 20 node DBMS cluster
- What is wrong with the above conclusion?

Hive/Hadoop instead of Parallel DBMS?

- Dr Stonebraker's proposal would put 5 PB per node on DBMS
  - What will be the io throughput of that system? **Abysmal**
  - How many concurrent queries can it support? **Certainly not 100K concurrent clients**
  - He is using a wrong metric to make a conclusion
- Hive/Hadoop is very very slow
  - Hive/Hadoop needs to be fixed to reduce query latency
  - But an existing DBMS cannot replace Hive/Hadoop

Questions?
dhruba@gmail.com

http://hadoopblog.blogspot.com/