A tutorial on Big Data Management

Santosh Mohanty, Jayant Dani and Srikar Chilakamarri
Tata Consultancy Services

Date: 16\textsuperscript{th} December, 2011
Discussion Topics

- Introduction to Big Data
- Challenges with Big data
- Framework and Solution
- Hadoop Architecture
- Application Scenario
- Enterprise Data Management
“Big data are data sets that grow so large that they become awkward to work with using on-hand database management tools.”

-- wikipedia

“A data is a big data when the size of the data itself becomes part of the problem.”

-- o’reilly media, radar post
Introduction to Big Data

Big Data - Evolution

Continuous Growing Large Data Sets for Storage Processing Management together termed as Big Data
Big Data - Characteristics

1. Aggregation of Loosely structured data
2. Size of data – terabyte ➔ petabyte ➔ exabyte
3. Data is distributed
4. Often contains incomplete data
5. Flat schemas with few complex interrelationships
Exponential data growth (Yahoo, Google etc.)

RDBMS systems unable to handle this growth

Solution - MapReduce, Big-Table, GFS

BIG Data Management Problem – All industries started realizing by now

History of Big Data
Types of Data

Structured data
- It has a well defined data model.

Semi Structured data
- Portion of data has meta data associated to it.

Unstructured data
- Data does not have any pre defined data model.
Types of Enterprise Data Sources

- E-mail: 72%
- Word Documents: 46%
- Spreadsheets: 36%
- Customer Databases: 33%
- Presentations: 21%
- Online Portals/Corporate Sites: 20%
- Instant Messages: 13%
- Social Media News Feeds: 9%
- Other: 6%
- RSS Feeds: 6%
- Friend Contact Lists: 5%
Big Data Challenges

- **Decision Speed**
- **Concurrency & Throughput**
- **Processing Complexity**
- **Transactional Data Volume**

Challenges with Big Data
Big Data Challenges

Flexibility & Governance

Data Structure

Structured
Non-structured
Traditional DB Technology Evolution

- **Post-relationship or NoSQL model**
  - Breaks the rigidity of the entity-relationship

- **Entity-Relationship model**

- **Relational Model by Codd**

- **First Generation Databases**
  - Hierarchical model
  - Network model (Codasyl model)
### Big Data Methodology vs. MPP/SMP

<table>
<thead>
<tr>
<th>Symmetric multiprocessing (SMP) involves a multiprocessor computer hardware architecture where two or more identical processors are connected to a single shared main memory and are controlled by a single OS instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>In MPP all the processing elements are connected together to be one very large computer. This is in contrast to distributed computing where massive numbers of separate computers are used to solve a single problem.</td>
</tr>
<tr>
<td>Distributed computing ultimately use some combination of both SMP/MMP approaches.</td>
</tr>
</tbody>
</table>
Challenge at Facebook

Rate of data growth 15TB/day and growing.

- Desired Reports to be generated
  - # of men or women users for a particular period
  - # users who commented for a particular period

They used python scripts to generate them.

As the data grew, One day Data take more than day to process

Source: Apache Hadoop Goes Real time at Facebook by Dhruba Borthakur
Introduction to MapReduce Framework

MapReduce Framework consists of two parts.

**Map function:**
Operates on set of key, value pairs.
Map is applied in parallel on input data set.
This produces output keys and list of values for each key depending upon the functionality.
Mapper output are partitioned per reducer = Number Of reduce task for that job.

**Reduce function:**
Operates on set of key, value pairs.
Reduce is then applied in parallel to each group, again producing a collection of key, values.
Number of reducers can be set by the user.
MapReduce Framework

The overall MapReduce word count process

Input: Deer Bear River Car Car River Deer Car Bear

Splitting: Deer Bear River Car Car River Deer Car Bear

Mapping: Deer, 1 Bear, 1 River, 1 Car, 1 Car, 1 Deer, 1 Deer, 1

Shuffling: Bear, 1 Bear, 1 Car, 1 Car, 1 Car, 1 Deer, 1 Deer, 1

Reducing: Bear, 2 Car, 3 Deer, 2 River, 2

Final result: Bear, 2 Car, 3 Deer, 2 River, 2

Framework and Solution
# Introduction to Hadoop Core, HDFS

## Hadoop

<table>
<thead>
<tr>
<th><strong>HDFS</strong></th>
<th><strong>Hadoop MapReduce</strong></th>
<th><strong>Hadoop Common</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The Hadoop distributed File System (HDFS) is a distributed file system designed to run on commodity hardware.</td>
<td>Hadoop MapReduce is the implementation of MapReduce Framework.</td>
<td>Hadoop common is the Hadoop core that consists of common utilities and libraries.</td>
</tr>
</tbody>
</table>
Hadoop Architecture and Hstack

- Architecture
- Principle behind the Architecture
- Namenode
- Datanode
- Job tracker
- Secondary Namenode
- MapReduce
- How MapReduce works in Hadoop
- Sample problem on MapReduce
- HDFS + MapReduce - sample program
- Other Hstack products
HDFS Architecture

Secondary Name Node
(copy of FSIMAGE)

Name Node
(Meta Data for Datanodes)

Block operations

Rack 1

DataNode

DataNode

DataNode

Replication

Rack 2

DataNode

DataNode

Block operations

Hadoop Architecture
Hadoop MapReduce Architecture

- Secondary NameNode
- NameNode
- JobTracker

Rack 1:
- Task Tracker
- DataNode
- Task Tracker
- DataNode

Rack 2:
- Task Tracker
- DataNode
- Task Tracker
- DataNode
Designing Principles of Hadoop

Created by Yahoo (Doug Cutting)

- Process internet scale data
- Save costs by distributed workload on # inexpensive computers

Features

- Fault tolerant
- Scalable
- Reduce communications
- Distribute data
- Processing at where the data is lies
- Make parallelism by multi-instance of OS
- Relatively inexpensive HW ($2k – 4K)
- Reliability <- replication

Bring processing to Data!

- Hadoop = HDFS + MapReduce infrastructure
Components of HDFS

- Secondary NameNode
- DataNode
- NameNode
- Jobtracker
- Tasktracker
### NameNode

| **Meta data in Memory** | • The entire metadata is in main memory  
<table>
<thead>
<tr>
<th></th>
<th>• No demand paging of meta-data</th>
</tr>
</thead>
</table>
| **Types of Metadata**  | • List of files  
|                        | • List of Blocks for each file  
|                        | • List of DataNodes for each block  
|                        | • File attributes, e.g., creation time, replication factor |
| **A Transaction Log**  | • Records file creations, file deletions, etc |
DataNode

**Block Server**
- Stores data in the local file system (e.g. ext3)
- Stores meta-data of a block (e.g. CRC)
- Serves data and meta-data to Clients

**Block Report**
- Periodically sends a report of all existing blocks to the NameNode

**Facilitates Pipelining of Data**
- Forwards data to other specified DataNodes

Hadoop Architecture
Jobtracker and Tasktracker

**Tasktracker**
- Keep a track of the map/reduce task given to that particular node.
- A heartbeat is sent to jobtracker to check its status.

**Jobtracker**
- Scheduler for MapReduce job
- Jobtracker pushes the job to tasktracker
- Keeps the job as close to the data as possible.
Secondary Namenode

- Memory requirements
- Separate Node
- Backup For NameNode
- Restore from failure

Hadoop Architecture
MapReduce in Hadoop

Jobtracker recognizes nearest datanode with data in the network

If heartbeat fails, then task fails

HeartBeat used by jobtracker to keep track of tasktracker

Jobtracker consults namenode for tasktracker and datanode for assigning mapper and reducer

Hadoop Architecture

Client submission
Jobtracker submission
Consultation
Recognition
Heart Beat
Failure
Rescheduling
Other HStacks Product

- **Pig**
  - High level language for writing MapReduce programs
  - Generally used in ETL processes

- **Hive**
  - A data warehouse facility on top of HDFS
  - Has SQL-like syntax

- **HBase**
  - Clone of BigTable
  - Column oriented NoSQL database

Hadoop Architecture
Big Data – Do’s and Don’ts

Do’s

• To process large amounts of semi-structured data like analyzing log files
• When your processing can easily be made parallel like a sorting of an entire countries census data
• Running batch jobs is acceptable. For example website crawling by search engines
• When you have access to lots of cheap hardware

Don’ts

• To process GBs or few TBs of data
• Processing data that can fit into memory and processed without too much of a trouble.
• Thinking of replacing existing BI solution with Big Data Solution
• Implementing Hadoop as the only technology for Big Data Solution.
Sample Problem on MapReduce

**Sentimate Gaging:** Counts of occurrences of sentiment in a given input set.

The code is divided into two parts – Mapper and Reducer

1. Initialize a variable sentencevalue=1
2. Tokenize using tab
3. set first value as srnum
4. set second value as comment_String
5. For each comment_String check whether it contains a word from the two pre-defined dictionaries (simple text files) of **positive words and negative words**
6. If the comment_String contains a positive word multiply the sentencevalue with 1 and if the comment_String contains a negative word multiply with -1
7. Check the variable sentencevalue
   - if sentencevalue>0
     - set key="Positive"
     - set value=concatenation of the srnum and the comment_String
   - else
     - set key="Negative"
     - set value=concatenation of the srnum and the comment_String

 Mapper

Reducer

1. The reducer method will get as input the key and a collection of values with similar key (We have only 2 key values here Positive and Negative)
2. It iterate through the values and concatenate the value part into a string say str
3. Lastly it set the context writer as key and value as str
Sample Results – Derived from Sample Problem

### Feature and Sentiment analysis

**Positive**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sentiment</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance</td>
<td>Positive</td>
<td>477</td>
</tr>
<tr>
<td>Battery</td>
<td>Negative</td>
<td>74,458</td>
</tr>
<tr>
<td>Battery</td>
<td>Positive</td>
<td>74,747</td>
</tr>
<tr>
<td>Camera</td>
<td>Positive</td>
<td>74,454</td>
</tr>
<tr>
<td>General</td>
<td>Negative</td>
<td>148,959</td>
</tr>
<tr>
<td>General</td>
<td>Positive</td>
<td>208,268</td>
</tr>
<tr>
<td>Memory</td>
<td>Negative</td>
<td>74,425</td>
</tr>
<tr>
<td>MusicPlayer</td>
<td>Negative</td>
<td>146,143</td>
</tr>
<tr>
<td>Radio</td>
<td>Positive</td>
<td>74,622</td>
</tr>
</tbody>
</table>

**Negative**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sentiment</th>
<th>Count</th>
</tr>
</thead>
</table>

### Application Scenario
Hadoop in Social Collaboration

Web servers → Logs streamed into Data collectors like scribe

Data collector → Collected data is sent to real time Hadoop cluster

Real time Hadoop cluster

Hadoop Hive warehouse

MySQL Database

User data sent to Hadoop Hive warehouse

Collected logs sent to Hadoop Hive warehouse

Reporting based on hive queries

Source: http://hadoopblog.blogspot.com/2009/06/hdfs-scribe-integration.htm
Hadoop in Market Research (Retail)

Application Scenario

*hs – Hadoop streaming
Hadoop as ETL (in Financial Service)

Hadoop Ecosystem

- HBase
  - Operational Data Store
  - Aged Data
- Hive
  - Data Warehouse
- Hadoop
  - HDFS
  - HBase BulkLoader
  - Java API

Application Scenario

MQ Queue

Intra-day Feeds

Hadoop Adapter

Flat Files

End of Day Feeds

Hadoop API

BI/Analytics

HQL/JDBC

JDBC/Sqoop

Map/Reduce

Data Mart

Operational Data Store

Aged Data

Operational Data Store

Aged Data

Operational Data Store

Aged Data

Operational Data Store

Aged Data

Operational Data Store

Aged Data
Enterprise Data Management (EDM)

DAMA International
Data management is the **development, execution and supervision** of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets.

Wikipedia
Enterprise Data Management or EDM is:

**A concept** – referring to the ability of an organization to precisely define, easily integrate and effectively retrieve data for both internal applications and external communication.

**A business objective** – focused on the *creation* of accurate, consistent and transparent data content. EDM emphasizes data precision, granularity and meaning and is concerned with how the content is *integrated* into business applications as well as how it is *passed along* from one business process to another.

**We define**
Enterprise Data management **is a process** to effectively manage lifecycle of the enterprise data. It includes architectures, methods, policies and services to acquire, validate, store, protect and enhance value of the enterprise data.
EDM – Process Framework

Iterative process

- Changing businesses
- Changing technology and supported data formats
- Increasing dependency to make fact based strategies

What does the process suggest?

- A well defined step by step approach
- An iterative process to refine or redefine strategies
Data and its Processes

- Data Architecture
- Data Quality
- Master data
- Data Lifecycle management
- Data Security
- Data Governance
- Metadata Management

- Store in a safe place
- Identify most valuable
- Install security
- Nurture it to grow
- Understand Value
- Plot location info
- Establish rules and policies for governance
More subjects would augment this paradigm... Data management is a vast, complex and deep routed subject
Have seen all the reports... But, is my data healthy?

I don’t know IT, I need to know how good is the underlying data for my process?

What is the best way to optimize my data processes?

All of my technology vendors can help. Only a few products are effective.

I need to make an update to my reference data, where all do I apply the changes?

I need to introduce a new application, what is its impact?

I have loads of data, how do I know which one to retire and which is most critical?

I have some files from external agencies, is it relevant to my organization?

I need someone to answer these... NOW!!
What Enterprise should do...

Define and Detail Enterprise Data Management Strategy

- Solutions for every aspect of data problem
- Many similar products offered by vendors to address the problem
- Well defined processes to measure the problem intensity

Always asks these questions while reviewing strategy and progress

- Are the measures easily understood by all users in enterprise?
- Can the tools be customized “on the fly” to cater to specific requirements?
- Has the perpetual scope of data management defined today?
- Does any tool give me a complete picture of enterprise data?
EDM Landscape

Enterprise Data manager

Semantic interpreter

A platform based on “Loosely couple, tightly integrated” principle

Key result areas

✓ Uniform metrics at all levels of abstraction
✓ One single interface at enterprise level
✓ Highly componentized services
✓ Adaptable to any technology/tools
✓ Diagnostic capability
EDM Platform – Logical View

Enterprise Portal

Enterprise Services

EDM Tools

Governance data

Security data

Data Models

ILM processes

EDM Services

Repository

Repository

Repository

Repository

Repository

Data stores

Adapters

Enterprise Metadata

EDM Processes

Enterprise Portal

Enterprise Services

EDM Platform – Logical View
EDM Platform Features

• “One answer to a question”. - Metrics consolidation (DQI, DEI, DVI, DUI)
• Have detailed information on request - Drill down to the lowest level of detail
• Clearly communicate the status – Dashboards refreshed periodically
• Enable seamless communication – Centralized Metadata driven architecture
• Transparency to enterprise data – One single integrated metadata view
• Resource management – Highly componentized EDM services
• Rational judgment of proposals- Use of features (Impact analysis, IQ, DRI)
• Ease of configuring new jobs- Configurable service components
• Complementary services – Data processing and storage capability
## Challenges

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Platform components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have seen all the reports.. But, is my <strong>data healthy</strong>?</td>
<td>Data Effectiveness Index</td>
</tr>
<tr>
<td>I don’t know IT, I need to know <strong>how good is the underlying data</strong> for my process?</td>
<td>Process level – Data quality index</td>
</tr>
<tr>
<td>What is the best way to <strong>optimize my data processes</strong>?</td>
<td>Resource utilization- Technical metadata</td>
</tr>
<tr>
<td>All of my technology vendors can help. Only a <strong>few products</strong> are effective</td>
<td>Tools and components as services</td>
</tr>
<tr>
<td>I need to make an update to my reference data, where all do I apply the changes?</td>
<td>Centralized metadata</td>
</tr>
<tr>
<td>I need to introduce a new application, what is its <strong>impact</strong>?</td>
<td>Impact analysis- Enterprise architecture</td>
</tr>
<tr>
<td>I have loads of data, how do I know which one to retire and which is most <strong>critical</strong>?</td>
<td>Data Value index and data usage index</td>
</tr>
<tr>
<td>I have some files from external agencies, is it <strong>relevant</strong> to my organization?</td>
<td>Intelligence quotient and Data relevance index</td>
</tr>
</tbody>
</table>
EDM in a Big Data Environment

**Impact**
- Increasing data volume, Decreasing data quality
- Modeling and governance are challenged
- Reduced importance of technology, Speed gains forefront

**Next GEN EDM**
- **Metadata driven architectures** gain importance
- “Perfection hinders progress”- EDM metrics shall **not look for accuracy**
- Governance policies and rules shall become **more complex**.
- Organizations **cannot afford mixed bag of tools**, uniformity shall be desired
- EDM platform gains importance to **complement and leverage** existing tools
EDM Platform – Measures

Platform Performance

- Service effectiveness score: $f(\text{functionality, performance, integration, interoperability, security, collaboration})$
- User defined = AVERAGE (AVERAGE (SCORE (Functional)), AVERAGE (SCORE (Non-Functional)))
- System defined = (System parameters, SLAs, Tolerance, Recorded value)
Revisiting Discussion Points...

- Introduction to Big Data
- Challenges with Big data
- Framework and Solution
- Hadoop Architecture
- Application Scenario
- Enterprise Data Management
A tutorial on
BIG Data Management

Thank You for Listening

Santosh Mohanty  Jayant Dani  Srikar Chilakamarri

Tata Consultancy Services

ENGINEERING SUSTAINABLE SOLUTIONS
30 December 2011

Big Data Management