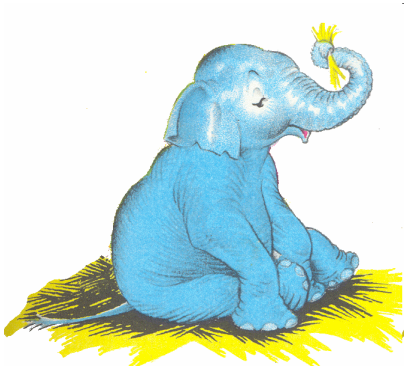
Big Data Fundamentals



Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

These slides and audio/video recordings of this class lecture are at:

http://www.cse.wustl.edu/~jain/cse570-13/

Overview

1. Why Big Data?
2. Terminology
3. Key Technologies: Google File System, MapReduce, Hadoop
4. Hadoop and other database tools
5. Types of Databases

Ref: J. Hurwitz, et al., "Big Data for Dummies," Wiley, 2013, ISBN:978-1-118-50422-2

Big Data

- Data is measured by 3V's:
- **Volume**: TB
- **Velocity**: TB/sec. Speed of creation or change
- **Variety**: Type (Text, audio, video, images, geospatial, ...)
- Increasing processing power, storage capacity, and networking have caused data to grow in all 3 dimensions.
- Volume, **Location**, Velocity, **Churn**, Variety, Veracity (accuracy, correctness, applicability)
- Examples: social network data, sensor networks, Internet Search, Genomics, astronomy, …

Why Big Data Now?

1. Low cost storage to store data that was discarded earlier
2. Powerful multi-core processors
3. Low latency possible by distributed computing: Compute clusters and grids connected via high-speed **networks**
4. Virtualization ⇒ Partition, Aggregate, isolate resources in any size and dynamically change it ⇒ Minimize latency for any scale
5. Affordable storage and computing with minimal man power via clouds
   ⇒ Possible because of advances in **Networking**

Why Big Data Now? (Cont)

6. Better understanding of task distribution (MapReduce), computing architecture (Hadoop),
7. Advanced analytical techniques (Machine learning)
8. Managed Big Data Platforms: Cloud service providers, such as Amazon Web Services provide Elastic MapReduce, Simple Storage Service (S3) and HBase – column oriented database. Google' BigQuery and Prediction API.
9. Open-source software: OpenStack, PostGresSQL
10. March 12, 2012: Obama announced $200M for Big Data research. Distributed via NSF, NIH, DOE, DoD, DARPA, and USGS (Geological Survey)

Big Data Applications

- Monitor premature infants to alert when interventions is needed
- Predict machine failures in manufacturing
- Prevent traffic jams, save fuel, reduce pollution

ACID Requirements

- **Atomicity**: All or nothing. If anything fails, entire transaction fails. Example, Payment and ticketing.
- **Consistency**: If there is error in input, the output will not be written to the database. Database goes from one valid state to another valid states. Valid=Does not violate any defined rules.
- **Isolation**: Multiple parallel transactions will not interfere with each other.
- **Durability**: After the output is written to the database, it stays there forever even after power loss, crashes, or errors.
- Relational databases provide ACID while non-relational databases aim for *BASE* (Basically Available, Soft, and Eventual Consistency)

Ref: http://en.wikipedia.org/wiki/ACID

Terminology

- **Structured Data**: Data that has a pre-set format, e.g., Address Books, product catalogs, banking transactions,
- **Unstructured Data**: Data that has no pre-set format. Movies, Audio, text files, web pages, computer programs, social media,
- **Semi-Structured Data**: Unstructured data that can be put into a structure by available format descriptions
- 80% of data is unstructured.
- Batch vs. Streaming Data
- **Real-Time Data**: Streaming data that needs to analyzed as it comes in. E.g., Intrusion detection. Aka "*Data in Motion*"
- **Data at Rest**: Non-real time. E.g., Sales analysis.
- **Metadata**: Definitions, mappings, scheme

Ref: Michael Minelli, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses," Wiley, 2013, ISBN:'111814760X

Relational Databases and SQL

- **Relational Database**: Stores data in tables. A "*Schema*" defines the tables, the fields in tables and relationships between the two. Data is stored one column/attribute

Order Table

Order Number	Customer ID	Product ID	Quantity	Unit Price

Customer Table

Customer ID	Customer Name	Customer Address	Gender	Income Range

- **SQL (Structured Query Language)**: Most commonly used language for creating, retrieving, updating, and deleting (**CRUD**) data in a relational database

Example: To find the gender of customers who bought XYZ:

Select CustomerID, State, Gender, ProductID from "Customer Table", "Order Table" where ProductID = XYZ

Ref: http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

Non-relational Databases

- **NoSQL**: Not Only SQL. Any database that uses non-SQL interfaces, e.g., Python, Ruby, C, etc. for retrieval. Typically store data in key-value pairs.
- Not limited to rows or columns. Data structure and query is specific to the data type
- High-performance in-memory databases
- RESTful (Representational State Transfer) web-like APIs
- Eventual consistency: BASE in place of ACID

NewSQL Databases

- Overcome scaling limits of MySQL
- Same scalable performance as NoSQL but using SQL
- Providing ACID
- Also called Scale-out SQL
- Generally use distributed processing.

Ref: http://en.wikipedia.org/wiki/NewSQL

Columnar Databases

ID	Name	Salary
101	Smith	10000
105	Jones	20000
106	Jones	15000

- In Relational databases, data in each row of the table is stored together: 001:101,Smith,10000; 002:105,Jones,20000; 003:106,John;15000
  - Easy to find all information about a person.
  - Difficult to answer queries about the aggregate:
    - How many people have salary between 12k-15k?
- In Columnar databases, data in each column is stored together. 101:001,105:002,106:003; Smith:001, Jones:002,003; 10000:001, 20000:002, 150000:003
  - Easy to get column statistics
  - Very easy to add columns
  - Good for data with high variety ⇒ simply add columns

Ref: http://en.wikipedia.org/wiki/Column-oriented_DBMS

Types of Databases

- **Relational Databases**: PostgreSQL, SQLite, MySQL
- **NewSQL Databases**: Scale-out using distributed processing

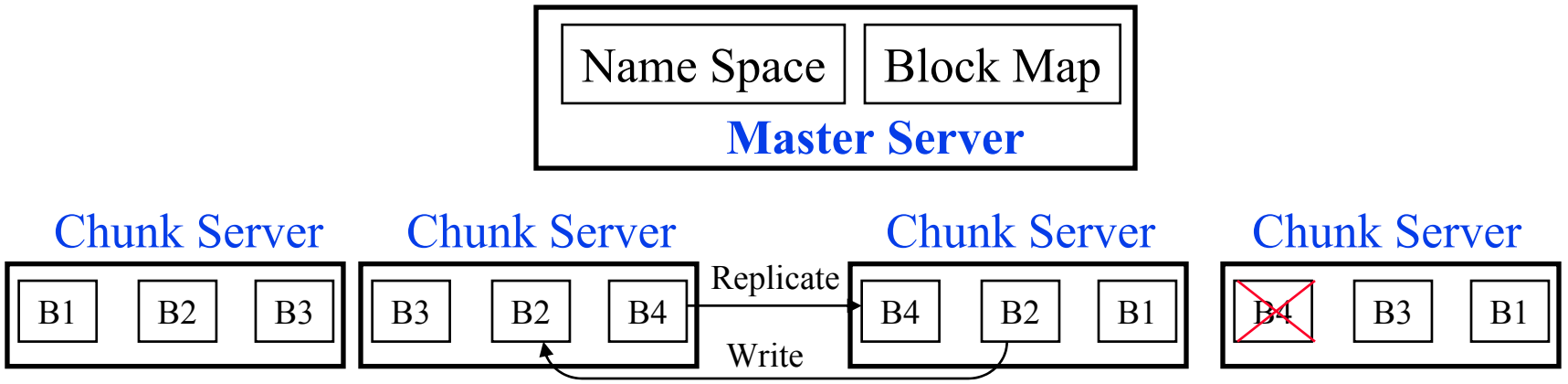
Non-relational Databases:

- **Key-Value Pair (KVP) Databases**: Data is stored as Key:Value, e.g., Riak Key-Value Database
- **Document Databases**: Store documents or web pages, e.g., MongoDB, CouchDB
- **Columnar Databases**: Store data in columns, e.g., HBase
- **Graph Databases**: Stores nodes and relationship, e.g., Neo4J
- **Spatial Databases**: For map and nevigational data, e.g., OpenGEO, PortGIS, ArcSDE
- **In-Memory Database (IMDB)**: All data in memory. For real time applications

Cloud Databases: Any data that is run in a cloud using IAAS, VM Image, DAAS

Google File System

- Commodity computers serve as "Chunk Servers" and store multiple copies of data blocks
- A master server keeps a map of all chunks of files and location of those chunks.
- All writes are propagated by the writing chunk server to other chunk servers that have copies.
- Master server controls all read-write accesses



Ref: S. Ghemawat, et al., "The Google File System", OSP 2003, http://research.google.com/archive/gfs.html

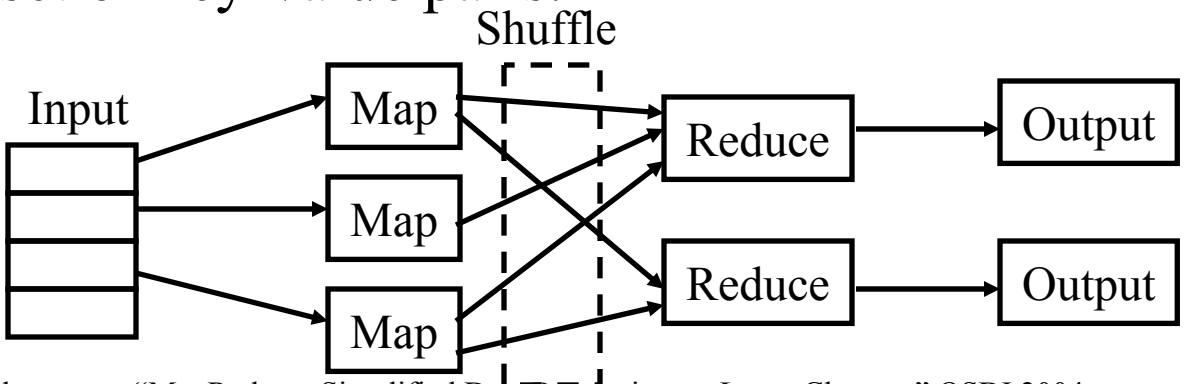
BigTable

- Distributed storage system built on Google File System
- Data stored in rows and columns
- Optimized for sparse, persistent, multidimensional sorted map.
- Uses commodity servers
- Not distributed outside of Google but accessible via Google App Engine

Ref: F. Chang, et al., "Bigtable: A Distributed Storage System for Structured Data," 2006, http://research.google.com/archive/bigtable.html

MapReduce

- Software framework to process massive amounts of unstructured data in parallel
- **Goals**:
  - Distributed: over a large number of inexpensive processors
  - Scalable: expand or contract as needed
  - Fault tolerant: Continue in spite of some failures
- **Map**: Takes a set of data and converts it into another set of key-value pairs..
- **Reduce**: Takes the output from Map as input and outputs a smaller set of key-value pairs.



Ref: J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," OSDI 2004, http://research.google.com/archive/mapreduce-osdi04.pdf

MapReduce Example

- 100 files with daily temperature in two cities.
  Each file has 10,000 entries.
- For example, one file may have (Toronto 20), (New York 30), ..
- Our goal is to compute the maximum temperature in the two cities.
- Assign the task to 100 Map processors each works on one file. Each processor outputs a list of key-value pairs, e.g., (Toronto 30), New York (65), …
- Now we have 100 lists each with two elements. We give this list to two reducers – one for Toronto and another for New York.
- The reducer produce the final answer: (Toronto 55), (New York 65)

Ref: IBM. "What is MapReduce?," http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/

MapReduce Optimization

- **Scheduling**:
  - Task is broken into pieces that can be computed in parallel
  - Map tasks are scheduled before the reduce tasks.
  - If there are more map tasks than processors, map tasks continue until all of them are complete.
  - A new strategy is used to assign Reduce jobs so that it can be done in parallel
  - The results are combined.
- **Synchronization**: The map jobs should be comparables so that they finish together. Similarly reduce jobs should be comparable.
- **Code/Data Collocation**: The data for map jobs should be at the processors that are going to map.
- **Fault/Error Handling**: If a processor fails, its task needs to be assigned to another processor.

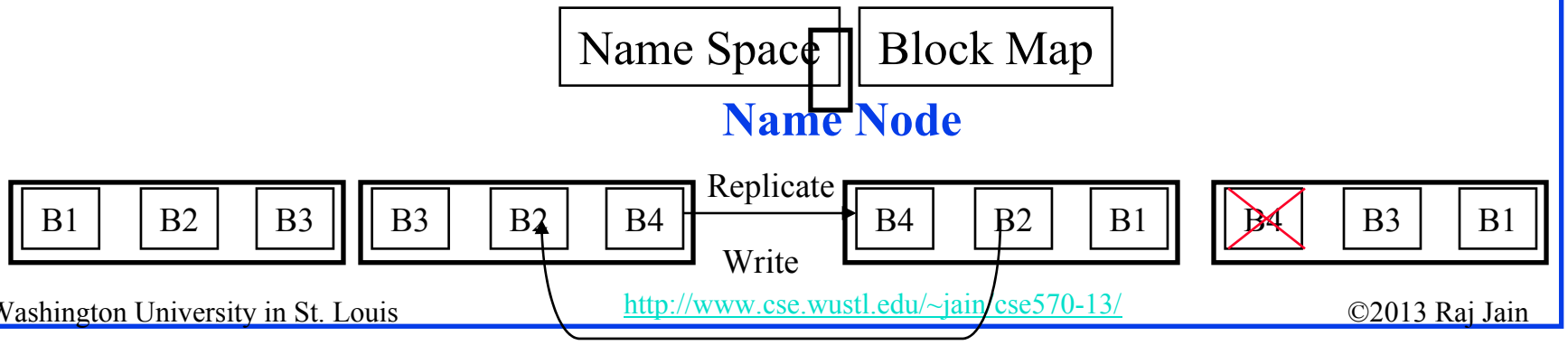
Story of Hadoop

- Doug Cutting at Yahoo and Mike Caferella were working on creating a project called "Nutch" for large web index.
- They saw Google papers on MapReduce and Google File System and used it
- Hadoop was the name of a yellow plus elephant toy that Doug's son had.
- In 2008 Amr left Yahoo to found Cloudera.
  In 2009 Doug joined Cloudera.

Ref: Michael Minelli, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses," Wiley, 2013, ISBN:'111814760X

Hadoop

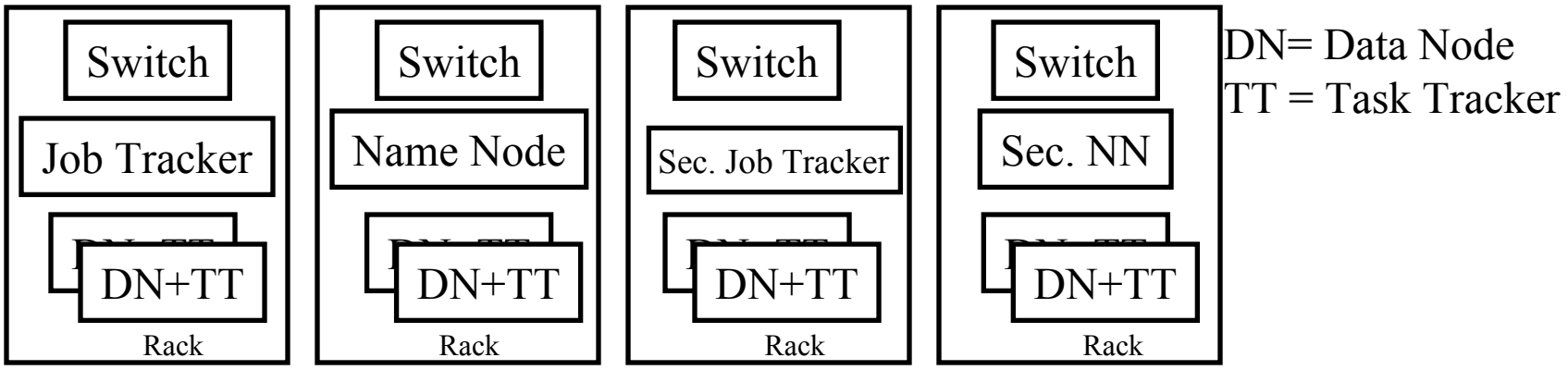
- An open source implementation of MapReduce framework
- Three components:
  - Hadoop Common Package (files needed to start Hadoop)
  - Hadoop Distributed File System: HDFS
  - MapReduce Engine
- HDFS requires data to be broken into blocks. Each block is stored on 2 or more data nodes on different racks.
- **Name node**: Manages the file system name space
  ⇒ keeps track of where each block is.



Hadoop (Cont)

- **Data node**: Constantly ask the job tracker if there is something for them to do

  $\Rightarrow$ Used to track which data nodes are up or down
- **Job tracker**: Assigns the map job to task tracker nodes that have the data or are close to the data (same rack)
- **Task Tracker**: Keep the work as close to the data as possible.



Hadoop (Cont)

- Data nodes get the data if necessary, do the map function, and write the results to disks.
- Job tracker than assigns the reduce jobs to data nodes that have the map output or close to it.
- All data has a check attached to it to verify its integrity.

Apache Hadoop Tools

- **Apache Hadoop**: Open source Hadoop framework in Java. Consists of Hadoop Common Package (filesystem and OS abstractions), a MapReduce engine (MapReduce or YARN), and Hadoop Distributed File System (HDFS)
- **Apache Mahout**: Machine learning algorithms for collaborative filtering, clustering, and classification using Hadoop
- **Apache Hive**: Data warehouse infrastructure for Hadoop. Provides data summarization, query, and analysis using a SQL-like language called HiveQL.
  Stores data in an embedded Apache Derby database.
- **Apache Pig**: Platform for creating MapReduce programs using a high-level "Pig Latin" language. Makes MapReduce programming similar to SQL. Can be extended by user defined functions written in Java, Python, etc.

Ref: http://hadoop.apache.org/, http://mahout.apache.org/, http://hive.apache.org/, http://pig.apache.org/

Apache Hadoop Tools (Cont)


- **Apache Avro**: Data serialization system.
  Avro IDL is the interface description language syntax for Avro.
- **Apache HBase**: Non-relational DBMS part of the Hadoop project. Designed for large quantities of sparse data (like BigTable). Provides a Java API for map reduce jobs to access the data. Used by Facebook.
- **Apache ZooKeeper**: Distributed configuration service, synchronization service, and naming registry for large distributed systems like Hadoop.
- **Apache Cassandra**: Distributed database management system. Highly scalable.

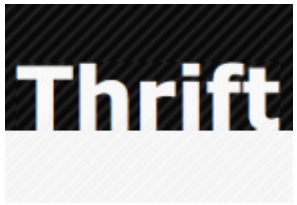
Ref: http://avro.apache.org/, http://cassandra.apache.org/, http://hbase.apache.org/ , http://zookeeper.apache.org/

Apache Hadoop Tools (Cont)

- **Apache Ambari**: A web-based tool for provision, managing and monitoring Apache Hadoop cluster
- **Apache Chukwa**: A data collection system for managing large distributed systems
- **Apache Sqoop**: Tool for transferring bulk data between structured databases and Hadoop
- **Apache Oozie**: A workflow scheduler system to manage Apache Hadoop jobs

Ref: http://incubator.apache.org/chukwa/, http://oozie.apache.org/, https://sqoop.apache.org/, http://incubator.apache.org/ambari/

Apache Other Big Data Tools



- **Apache Accumulo**: Sorted distributed key/value store based on Google's BigTable design. 3rd Most popular NOSQL wide-column system. Provides cell-level security. Users can see only authorized keys and values. Originally funded by DoD.
- **Apache Thrift**: IDL to create services using many languages including C#, C++, Java, Perl, Python, Ruby, etc.
- **Apache Beehive**: Java application framework to allow development of Java based applications.
- **Apache Derby**: A RDBMS that can be embedded in Java programs. Needs only 2.6MB disk space. Supports JDBC (Java Database Connectivity) and SQL.

Ref: http://en.wikipedia.org/wiki/Apache_Accumulo, http://en.wikipedia.org/wiki/Apache_Thrift, http://en.wikipedia.org/wiki/Apache_Beehive, http://en.wikipedia.org/wiki/Apache_derby,

Other Big Data Tools

- **Cascading**: Open Source software abstraction layer for Hadoop.
  Allows developers to create a .jar file that describes their data sources, analysis, and results without knowing MapReduce. Hadoop .jar file contains Cascading .jar files.
- **Storm**: Open source event processor and distributed computation framework alternative to MapReduce. Allows batch distributed processing of streaming data using a sequence of transformations.
- **Elastic MapReduce (EMR)**: Automated provisioning of the Hadoop cluster, running, and terminating. Aka Hive.
- **HyperTable**: Hadoop compatible database system.

Ref: http://en.wikipedia.org/wiki/Cascading, http://en.wikipedia.org/wiki/Hypertable, http://en.wikipedia.org/wiki/Storm_%28event_processor%29

Other Big Data Tools (Cont)

- **Filesysem in User-space (FUSE)**: Users can create their own virtual file systems. Available for Linux, Android, OSX, etc.
- **Cloudera Impala**: Open source SQL query execution on HDFS and Apache HBase data
- **MapR Hadoop**: Enhanced versions of Apache Hadoop supported by MapR. Google, EMC, Amazon use MapR Hadoop.
- **Big SQL**: SQL interface to Hadoop (IBM)
- **Hadapt**: Analysis of massive data sets using SQL with Apache Hadoop.

Ref: http://en.wikipedia.org/wiki/Filesystem_in_Userspace, http://en.wikipedia.org/wiki/Big_SQL, http://en.wikipedia.org/wiki/Cloudera_Impala , http://en.wikipedia.org/wiki/MapR, http://en.wikipedia.org/wiki/Hadapt

Analytics

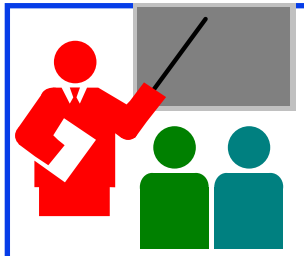
Analytics: Guide decision making by discovering patterns in data using statistics, programming, and operations research.

- **SQL Analytics**: Count, Mean, OLAP
- **Descriptive Analytics**: Analyzing historical data to explain past success or failures.
- **Predictive Analytics**: Forecasting using historical data.
- **Prescriptive Analytics**: Suggest decision options. Continually update these options with new data.
- **Data Mining**: Discovering patterns, trends, and relationships using Association rules, Clustering, Feature extraction
- **Simulation**: Discrete Event Simulation, Monte Carlo, Agent-based
- **Optimization**: Linear, non-Linear
- **Machine Learning**: An algorithm technique for learning from empirical data and then using those lessons to predict future outcomes of new data

Ref: Michael Minelli, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses," Wiley, 2013, ISBN:111814760X

Analytics (Cont)

- **Web Analytics**: Analytics of Web Accesses and Web users.
- **Learning Analytics**: Analytics of learners (students)
- **Data Science**: Field of data analytics



Summary

1. Big data has become possible due to low cost storage, high performance servers, high-speed networking, new analytics
2. Google File System, BigTable Database, and MapReduce framework sparked the development of Apache Hadoop.
3. Key components of Hadoop systems are HDFS, Avro data serialization system, MapReduce or YARN computation engine, Pig Latin high level programming language, Hive data warehouse, HBase database, and ZooKeeper for reliable distributed coordination.
4. Discovering patterns in data and using them is called Analytics. It can be descriptive, predictive, or prescriptive
5. Types of Databases: Relational, SQL, NoSQL, NewSQL, Key-Value Pair (KVP), Document, Columnar, Graph, and Spatial

Reading List

- J. Hurwitz, et al., "Big Data for Dummies," Wiley, 2013, ISBN:978-1-118-50422-2 (Safari Book)
- A. Shieh, "Sharing the Data Center Network," NSDI 2011, http://www.usenix.org/event/nsdi11/tech/full_papers/Shieh.pdf
- IBM. "What is MapReduce?," http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/
- Michael Minelli, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses," Wiley, 2013, ISBN:111814760X

Wikipedia Links

- http://en.wikipedia.org/wiki/Database
- http://en.wikipedia.org/wiki/Big_data
- http://en.wikipedia.org/wiki/Apache_Hadoop
- http://en.wikipedia.org/wiki/ACID
- http://en.wikipedia.org/wiki/Analytics
- http://en.wikipedia.org/wiki/Prescriptive_analytics
- http://en.wikipedia.org/wiki/Predictive_analytics
- http://en.wikipedia.org/wiki/Prescriptive_Analytics
- http://en.wikipedia.org/wiki/Apache_HBase
- http://en.wikipedia.org/wiki/Apache_Hive
- http://en.wikipedia.org/wiki/Apache_Mahout
- http://en.wikipedia.org/wiki/Apache_Pig
- http://en.wikipedia.org/wiki/Apache_ZooKeeper
- http://en.wikipedia.org/wiki/Apache_Accumulo

Wikipedia Links (Cont)

- http://en.wikipedia.org/wiki/Apache_Avro
- http://en.wikipedia.org/wiki/Apache_Beehive
- http://en.wikipedia.org/wiki/Apache_Cassandra
- http://en.wikipedia.org/wiki/Relational_database
- http://en.wikipedia.org/wiki/Relational_database_management_system
- http://en.wikipedia.org/wiki/Column-oriented_DBMS
- http://en.wikipedia.org/wiki/Spatial_database
- http://en.wikipedia.org/wiki/SQL
- http://en.wikipedia.org/wiki/NoSQL
- http://en.wikipedia.org/wiki/NewSQL
- http://en.wikipedia.org/wiki/MySQL
- http://en.wikipedia.org/wiki/Create,_read,_update_and_delete
- http://en.wikipedia.org/wiki/Unstructured_data
- http://en.wikipedia.org/wiki/Semi-structured_data

Wikipedia Links (Cont)

- http://en.wikipedia.org/wiki/Apache_derby
- http://en.wikipedia.org/wiki/Apache_Thrift
- http://en.wikipedia.org/wiki/Big_SQL
- http://en.wikipedia.org/wiki/Cascading
- http://en.wikipedia.org/wiki/Cloudera_Impala
- http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems
- http://en.wikipedia.org/wiki/Filesystem_in_Userspace
- http://en.wikipedia.org/wiki/Hadapt
- http://en.wikipedia.org/wiki/Hypertable
- http://en.wikipedia.org/wiki/MapR
- http://en.wikipedia.org/wiki/Storm_%28event_processor%29

References

- J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," OSDI 2004, http://research.google.com/archive/mapreduce-osdi04.pdf
- S. Ghemawat, et al., "The Google File System", OSP 2003, http://research.google.com/archive/gfs.html
- F. Chang, et al., "Bigtable: A Distributed Storage System for Structured Data," 2006, http://research.google.com/archive/bigtable.html
- http://avro.apache.org/
- http://cassandra.apache.org/
- http://hadoop.apache.org/
- http://hbase.apache.org/
- http://hive.apache.org/
- http://incubator.apache.org/ambari/
- http://incubator.apache.org/chukwa/
- http://mahout.apache.org/
- http://oozie.apache.org/
- http://pig.apache.org/
- http://zookeeper.apache.org/
- https://sqoop.apache.org/

Acronyms

- ACID — Atomicity, Consistency, Isolation, Durability
- API — Application Programming Interface
- ArcSDE — Arc Spatial Database Engine
- BASE — Basically Available, Soft, and Eventual Consistency
- CRUD — Create, Retrieve, Update, and Delete
- DAAS — Database as a Service
- DARPA — Defense Advanced Research Project Agancy
- DBMS — Database Management System
- DN — Data Node
- DoD — Department of Defense
- EMC — Name of a company
- FUSE — Filesysem in User-space
- HDFS — Hadoop Distributed File System
- IAAS — Infrastructure as a Service
- IBM — International Business Machines
- ID — Identifier

Acronyms (Cont)

- IDL Interface Description Language
- IMDB In-Memory Database
- JDBC Java Database Connectivity
- KVP Key-Value Pair
- NewSQL New SQL
- NoSQL Not Only SQL
- OLAP Online Analytical Processing
- OpenGEO
- OSDI Operating Systems Design and Implementation
- OSX Apple Mac Operating System version 10
- PortGIS Port Geographical Information System
- PostGresSQL PostGress SQL
- RDBMS Relational Database Management System
- REST Representation State Transfer
- SQL Structured Query Language
- TB Terabyte

Acronyms (Cont)

- TT Task Tracker
- US United States
- USGS United States Geological Survey
- VM Virtual Machine
- YARN Yet Another Resource Negotiator

Quiz 10A

- The 3V's that define Big Data are _____, _____, and _____
- ACID stands for _____, _____, _____, and _____
- BASE stands for _____ _____, _____, and _____ Consistency.
- _____ data is the data that has pre-set format.
- Data in _____ is the data that is streaming.

Your Name: _____

Solution to Quiz 10A

- The 3V's that define Big Data are *Volume*, *velocity*, and *variety*
- ACID stands for *Atomicity*, *Consistency*, *Isolation*, and *Durability*
- BASE stands for *Basically* *Available*, *Soft*, and *Eventual* Consistency.
- *Structured* data is the data that has pre-set format.
- Data in *Motion* is the data that is streaming.