ÖZYEĞİN UNIVERSITY

CCRG

Big Data Processing: Hadoop, NoSQL, CEP and Cloud

Büyük Veri İşleme: Hadoop Çerçevesi, NoSQL, Karmaşık Olay İşleme ve Bulut Bilişim

Asst. Prof. Dr. Ismail Ari

Büyük Veri ve Madenciliği, Şirince, 8-10 Ekim, 2015

Short Bio

İSMAİL ARI
Assistant Professor

Background

2004 Ph.D. Degree, Univ. Of California Santa Cruz
2000 M.S. Degree, Univ. Of Maryland B.C.
1998 B.S. Degree, Boğaziçi University

Experience

2013-2015 Teknopark Istanbul, GM
2009-2015 Özyegin University, Asst. Prof.
2011-2013 Co-Founder Havooz Ltd.
2004-2009 HP Labs, Palo Alto, Scientist
2001-2004 Storage Systems Research Center

Research Interests

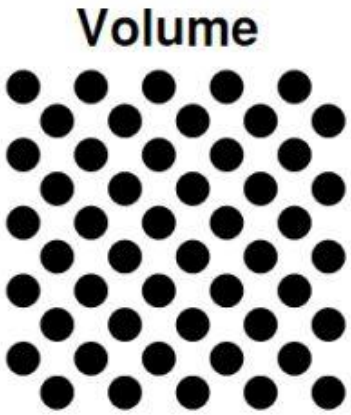
Big Data, Hadoop and Cloud Computing
Distributed, Networked Systems Performance
Databases and Data Stream Processing, CEP

Cloud Computing Research Group
http://cloud.ozyegin.edu.tr

E-mail: ismail.ari@ozyegin.edu.tr
http://faculty.ozyegin.edu.tr/ismailari

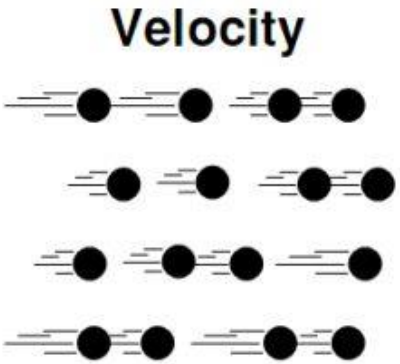
What is Big Data? *4V's*

We need new engines and methods to handle Big Data



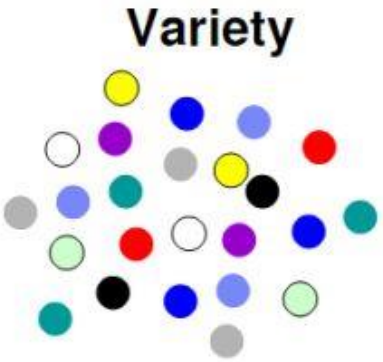
Data at Rest

Terabytes to Exabytes of existing data to process



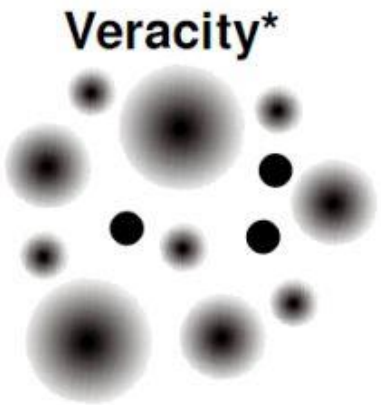
Data in Motion

Streaming data, Milliseconds to Seconds to respond



Data in Many Forms

Structured, Unstructured, Text, Multimedia

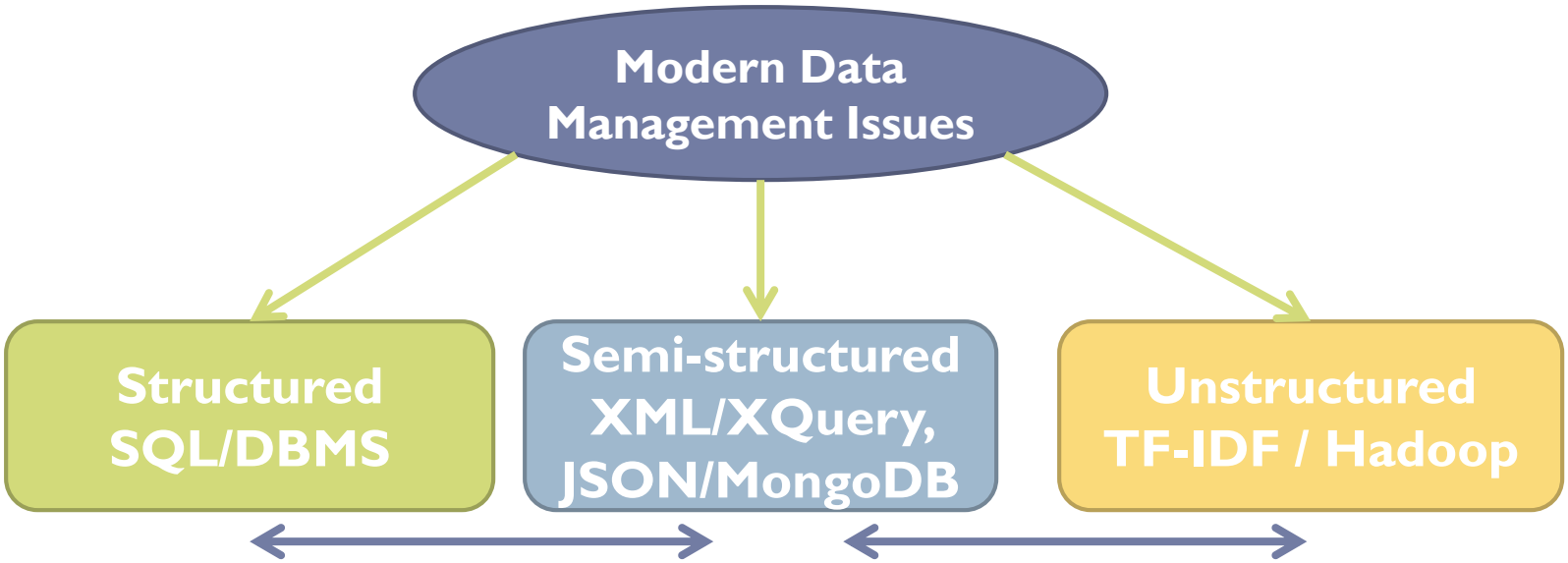


Data in Doubt

Uncertainty due to incomplete data, entry errors, processing, sensors, social media, latency of information, deception, modeling approximations, etc.

http://almaden.ibm.com/colloquium/resources/Why%20Big%20Data%20Krishna.PDF

What is Data? *Structured vs. Unstructured*



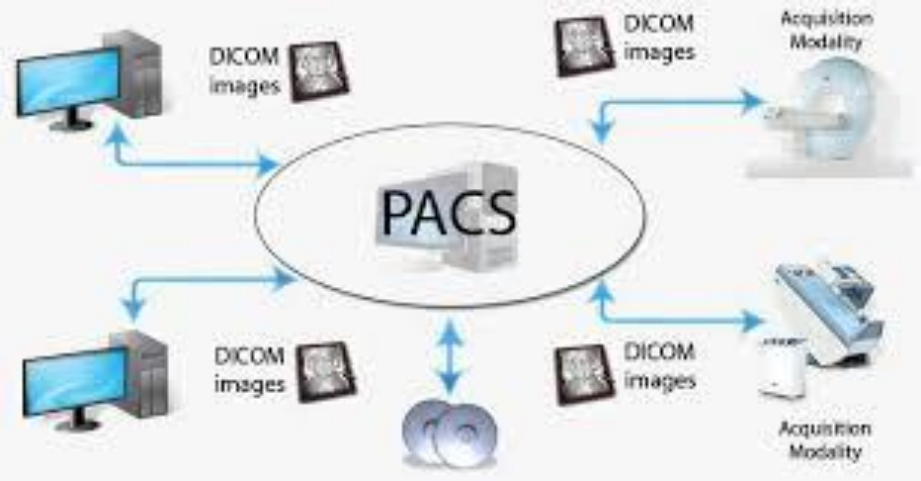
- New data formats
  - Javascript Object Notation (JSON)
  - Binary JSON (BSON)

Sector-Specific Data Formats

- Data serialization – deserialization
  - Serialize: Memory to storage or network; keep data and schema
  - Text, XML, JSON/BSON, Binary

OLE for Process Control

- OPC: Oil-gas, TÜPRAŞ
- DICOM-PACS: Healthcare, Hastaneler birliği
  - Digital Imaging and Comm. in Medicine, Picture Archiv. Comm. Sys.



Sector-Specific Data Formats

- Data serialization – deserialization
  - Serialize: Memory to storage or network; keep data and schema
  - Text, XML, JSON/BSON, Binary

- CDR (Call Detail Record): Mobile, Turkcell - Avea, BTK
  - <session, IMEI, src-dst #s, duration, vs.>
- ARINC 429 – Aviation, FAA, THY Teknik
  - Aeronaurical Radio INC. Avionic data transfer std.
- BASEL–XBRL, Borsa, Bankalar, MKK, BDDK
  - eXtensible Business Reporting Lang.
  - Financial Accounting Standard Board (FASB)

PMML ?

- ***Wikipedia*** defn:
- "The **Predictive Model Markup Language** (**PMML**) is an XML-based file format developed by the Data Mining Group to provide a way for applications to describe and exchange models produced by data mining and machine learning algorithms."
- DMG.org: Data Mining Group

How Big is BIG? ***Volume*** *issue ..*

- Big data as of 2015 ..
  - 3.2 billion internet users
  - 7 billion mobile phones
  - Ebay & Google: & Baidu 100 PB/day, (10s ExaB. stored)
  - NSA: 30 PB/day, Face: 700 TB/day and Twitter 100 TB/day
  - Scanning 1 PetaByte at 100 MB/sec = 116 days ~ 4 months

- We started touching BIG DATA in our daily lives
  - As customers we expect immediate answers to our questions and we do NOT tolerate (data) failures
  - Poor (zavallı) service providers

Istanbul Mass Transportation Scenario

- Mass transport vehicle tracking and accounting to assure <u>timely and reliable services</u>
  - 10,000 buses over 600+ routes
  - 10,000 bus stations (some being smart)
  - 3 million people/day to carry with a <u>single IstanbulCard</u> card over .. Metro lines, trains, light-rail, mini-buses, ferries, boats, etc.
  - 10+ different companies to accurately share the revenue

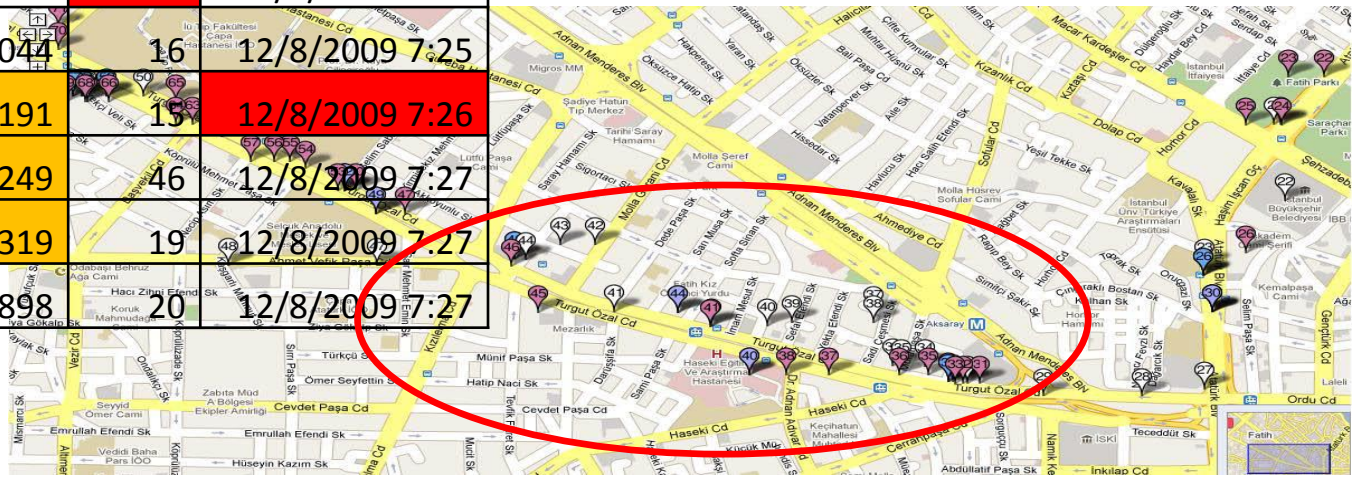
- ***Velocity*** will increase with taxi ridesharing, carpooling, minibus & waterjug (bidon) tracking apps, IoT

carticipate

ÖZYEĞİN UNIVERSITY

Veracity- Data Validation & Reconciliation

ID	LONGITUTE	LATITUDE	SPEED	DATE & TIME
00-123	28,863,169	4,105,348	42	12/8/2009 7:23
00-123	2,886,469	41,052,845	3	12/8/2009 7:23
00-123	28,866,064	41,052,856	26	12/8/2009 7:23
00-123	28,867,975	410,522	37	12/8/2009 7:24
00-123	2,886,879	4,105,189	1	12/8/2009 7:24
00-123	28,869,068	41,051,792	6	12/8/2009 7:24
00-123	28,869,884	41,051,376	16	12/8/2009 7:25
00-123	28,870,121	41,051,258	0	12/8/2009 7:25
00-123	2,887,055	41,051,044	16	12/8/2009 7:25
00-123	28,870,613	4,105,191	15	12/8/2009 7:26
00-123	28,868,597	4,105,249	46	12/8/2009 7:27
00-123	28,866,816	4,105,319	19	12/8/2009 7:27
00-123	288,657	41,053,898	20	12/8/2009 7:27



ÖZYEĞİN UNIVERSITY

Why are BIG data problems important?

- Easy problems have already been solved
  - OR they can be solved using a single computer.

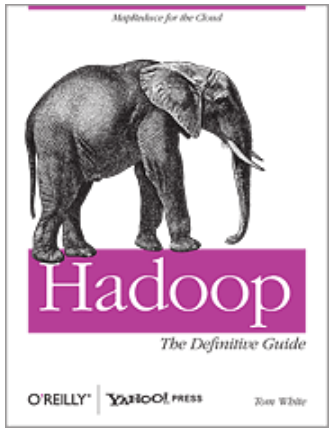
- BIG PROBLEMS remain (also where the $$$ is)
  - Most commercial, business, scientific problems
  - *a.k.a* **BIG DATA** problems.

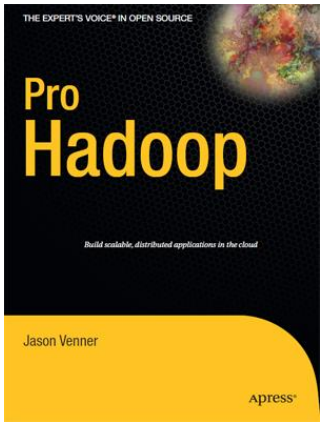
- BIG PROBLEMS can be feasibly solved using open-source distributed systems
  - However, you need to understand how DS work.

Need BIG Investments?

- Do we need to make BIG investments to solve BIG DATA problems? Not so much anymore!
  - Public Clouds allow everyone to access large-scale public computing resources and pay only as much as they use.

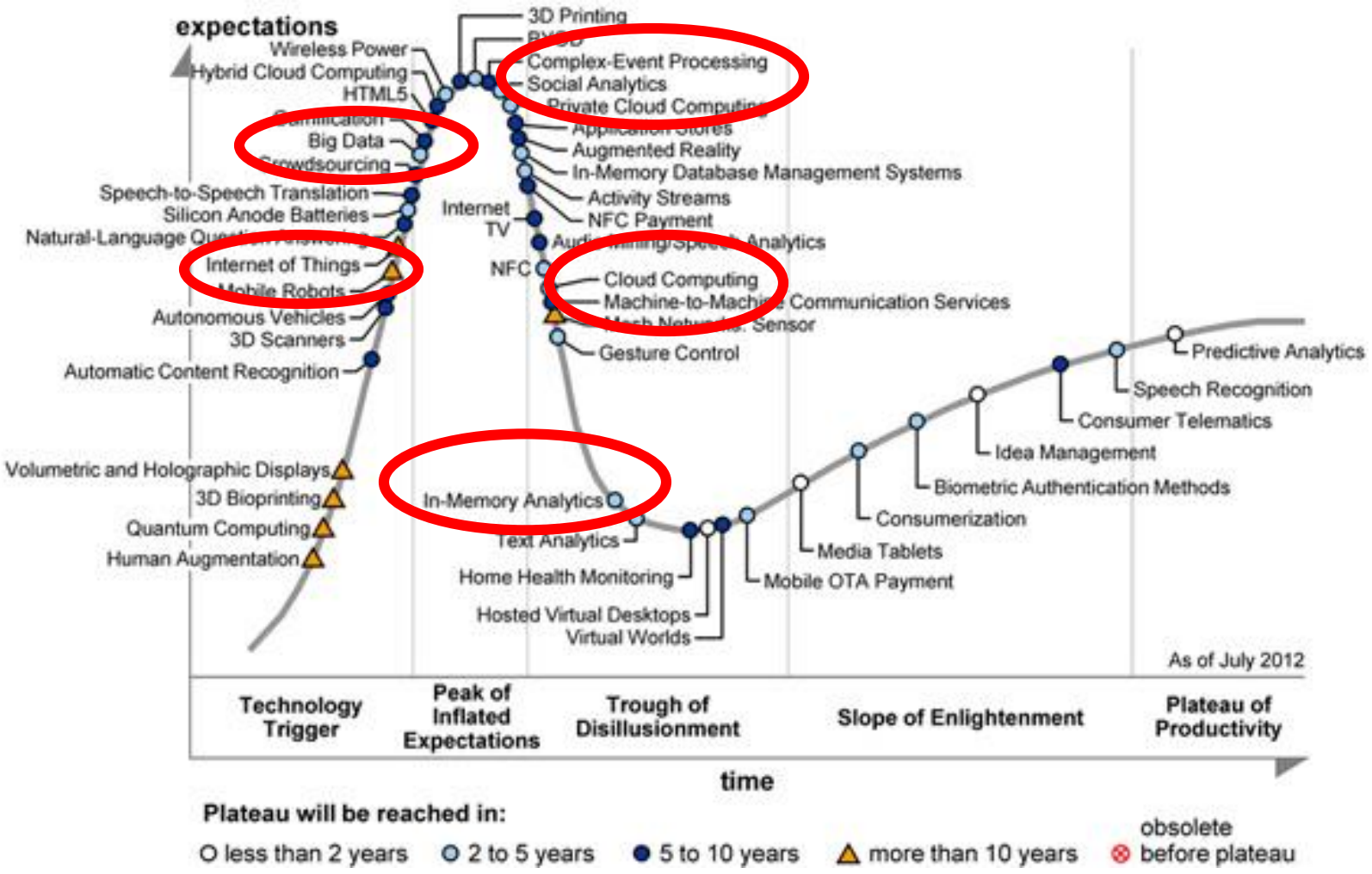
- Owning a private cloud is not expensive either.
  - You can build one using common-off-the-shelf components and open-source technology





Why learn a bit about Big Data & Cloud?

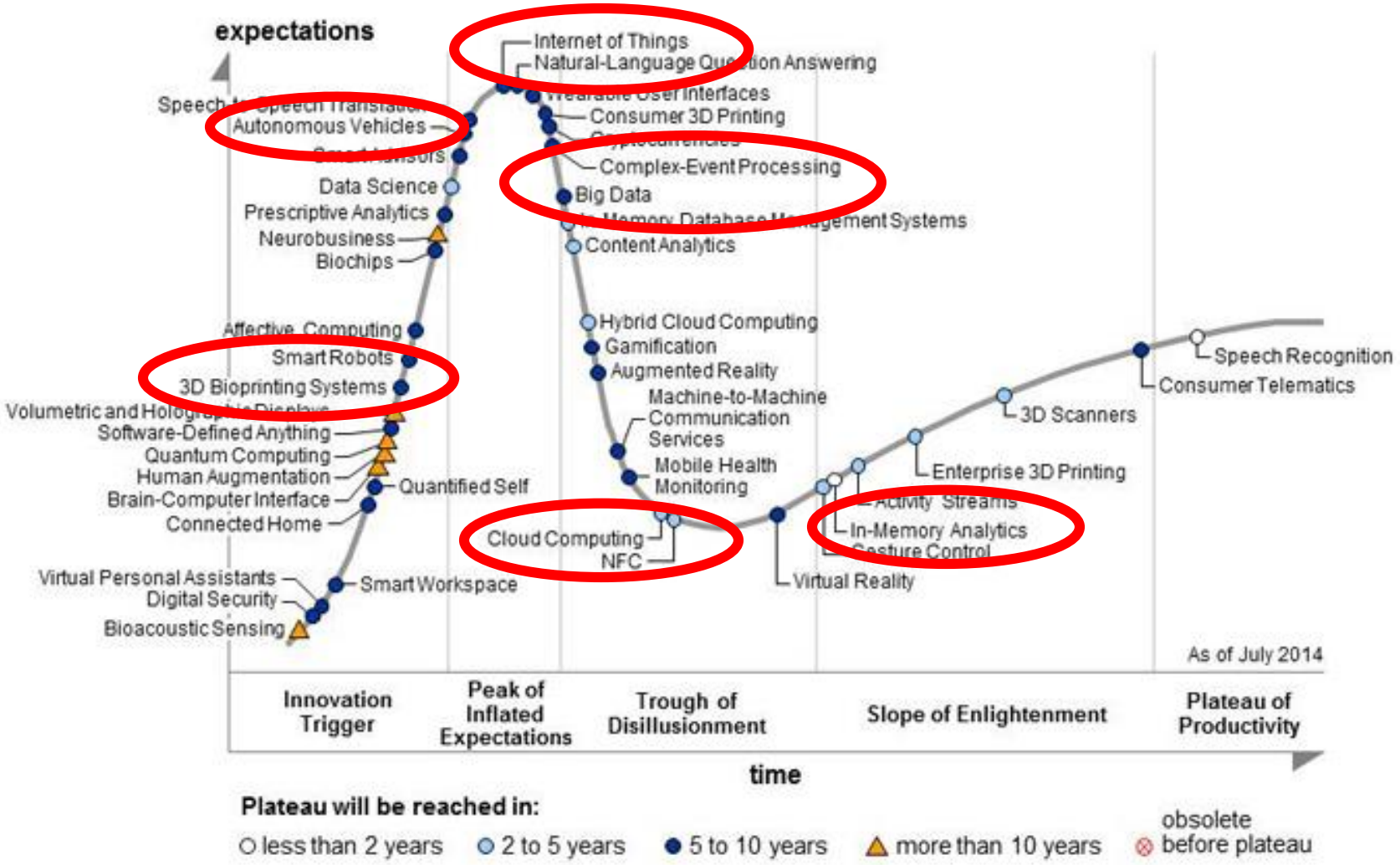
Gartner Hype Chart - 2012



ÖZYEĞİN UNIVERSITY

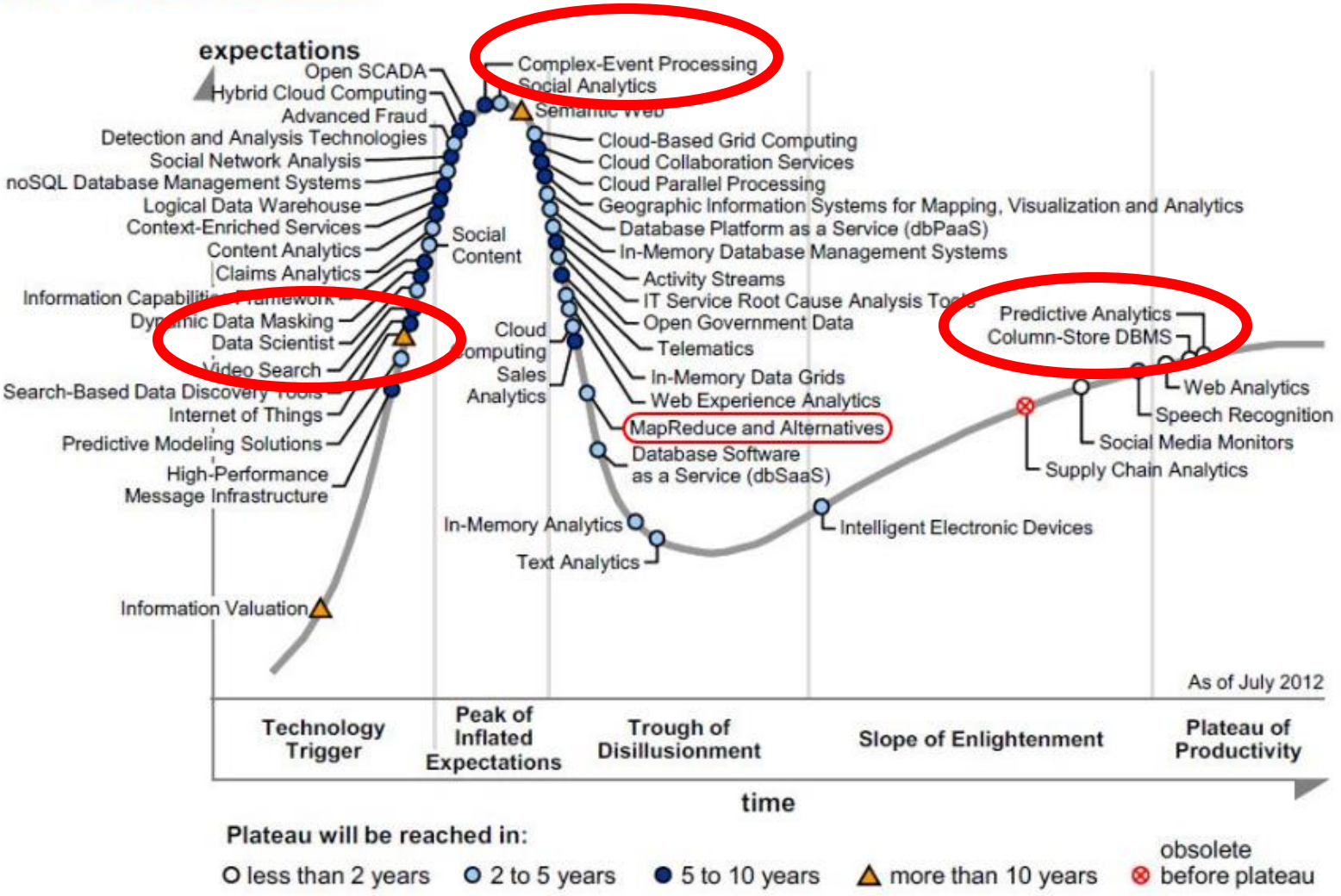
Why learn a bit about Big Data & Cloud?

Gartner Hype Chart - 2015



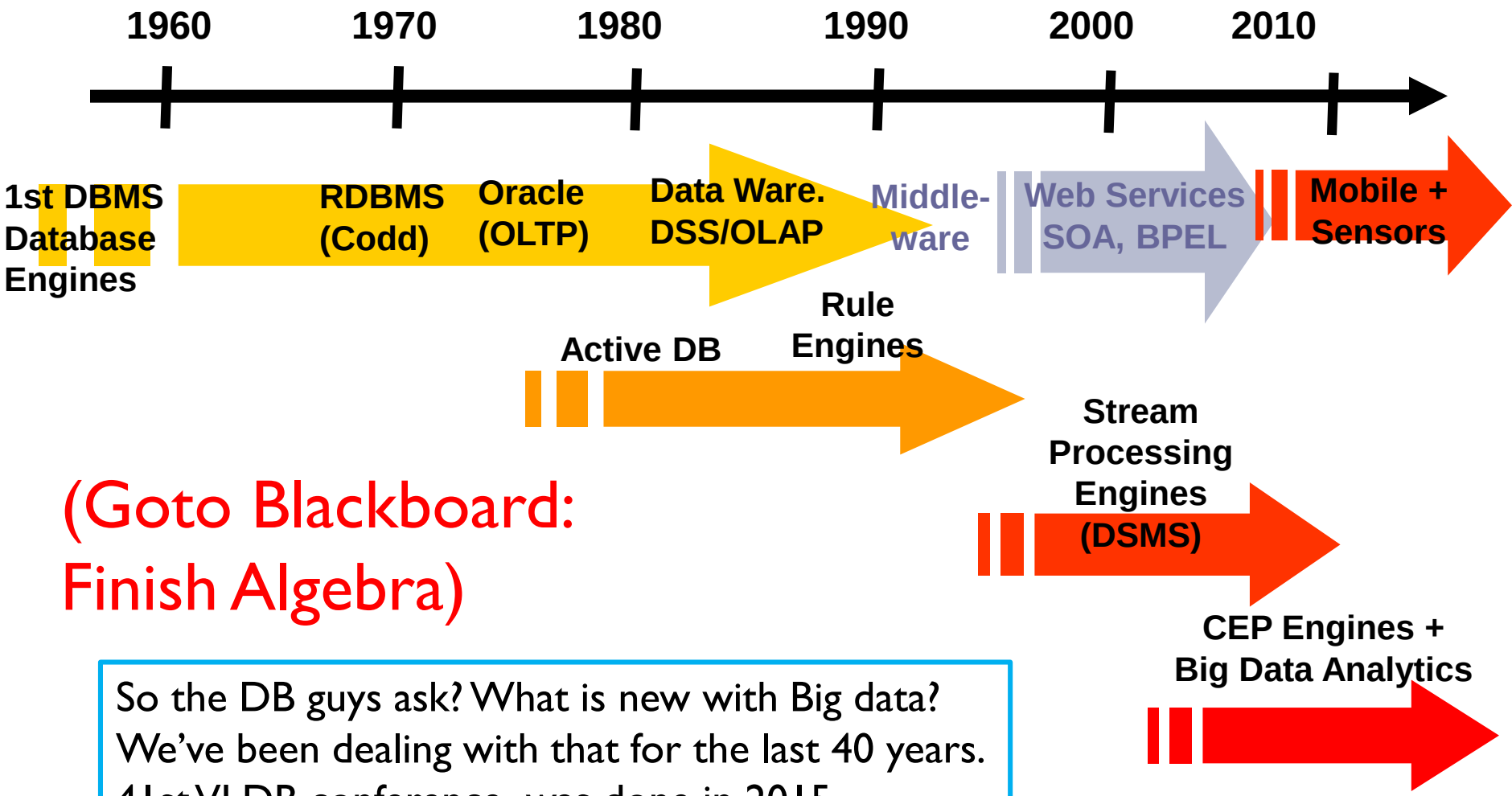
Gartner Hype Chart - Big Data 2012

Figure 1. Hype Cycle for Big Data, 2012



Source: Gartner (July 2012)

An un-official timeline for databases



(Goto Blackboard: Finish Algebra)

So the DB guys ask? What is new with Big data?
We've been dealing with that for the last 40 years.
41st VLDB conference was done in 2015.
36th SIGMOD, 31st ICDE.

ÖZYEĞİN UNIVERSITY

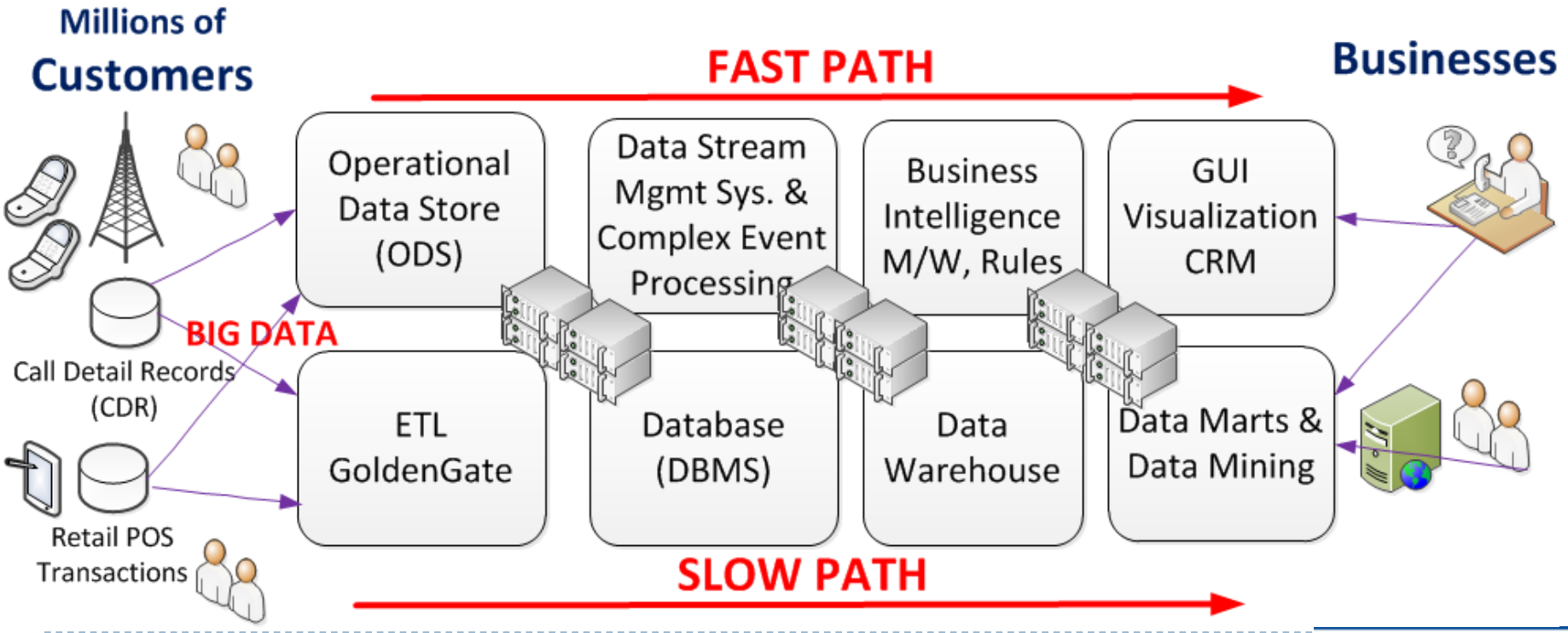
Evolutions of DBMS, DSMS, and EDW

OLTP →OLAP (*Cube, MDX*)→Data Mining (*DMX, ML tools*)

Last 40 years

Real-time EDW Architecture = EDW+CEP+Rules

Last 10 years



Cloud Computing Terminology

SaaS
Software as a Service (Portals)

PaaS
Platform as a Service (WS-*, Distrib.)

IaaS
Infra. as a Service (Virtual Machines)

Traditional

Software
(Email/Office, GIS, CRM)

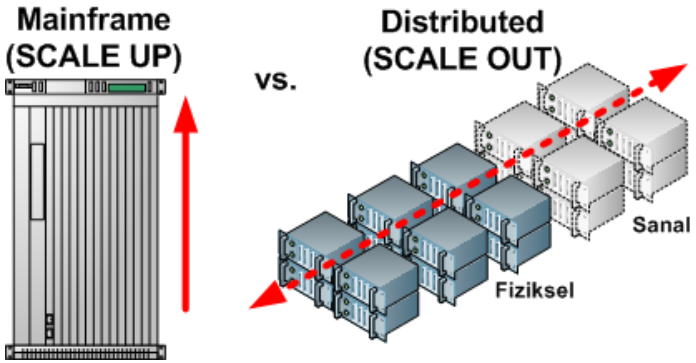
Platform
(Database, FS, Web Server)

Infrastructure
(Server, Storage, Network)

"as a Service"
= Over the Network (Private Network or Public Internet)

What is Hadoop?

- Trying to scale-up will always have limits
  - 1GB→1TB→100TB→1 PB→1 EB



- Hadoop is an open-source framework written in Java ..
  - Depends on scale-out (distribution, DFS+MPI ~ HDFS+MR)
  - It can be installed on cheaper hardware units
  - Reliability is built-in & is provided through replication
  - Same framework for structured, unstructured, semi-structured

Hadoop Users - Examples

http://wiki.apache.org/hadoop/PoweredBy

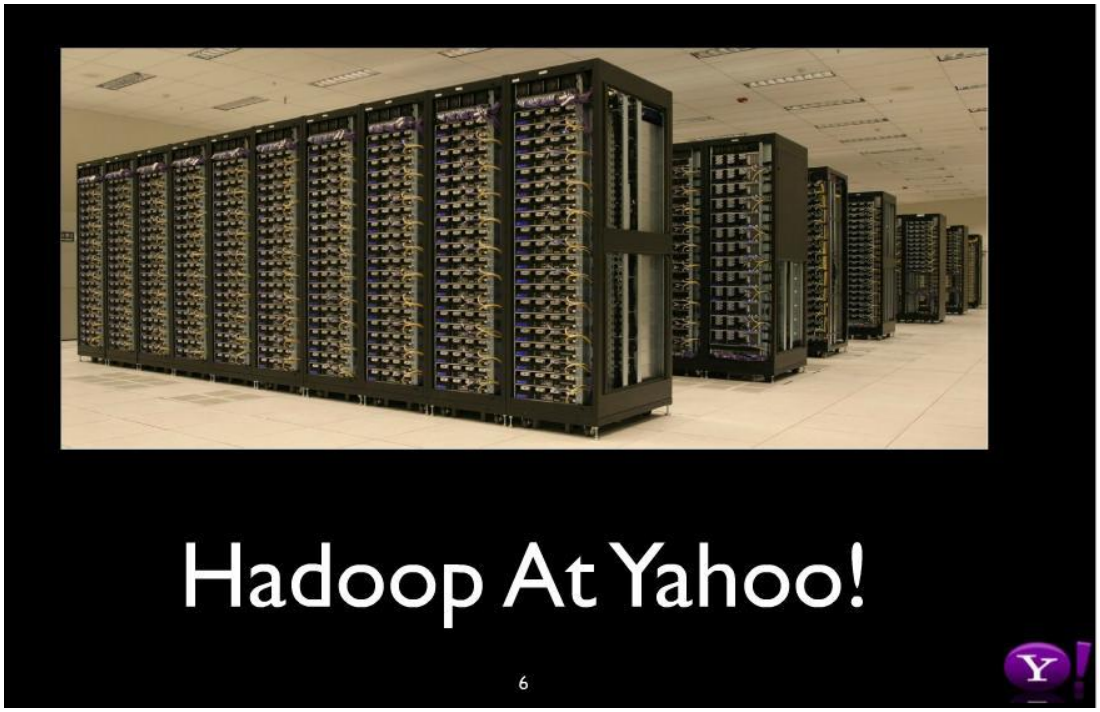
中国移动通信
CHINA MOBILE

- Telecomm - China Mobile
  - Hadoop cluster for data mining on Call Log Records
  - 5-8TB/day, with Hadoop 10 times faster, 1/5th the cost

- Other users and providers
  - IBM BigData, Watson AI created using Hadoop over 11M files
  - Yahoo, Facebook, LinkedIn, Twitter , SU,
  - Amazon, Rackspace
  - Turkcell, Avea and Telefonica Research
  - Ebay and Adobe
  - New York Times: 11 million images turned into 1.5TB PDF files in 1 day using a 100 instance Amazon EC2 cluster

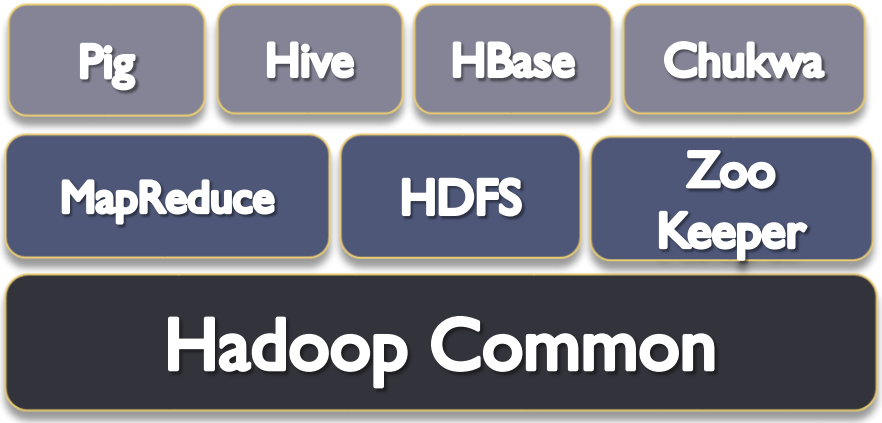
Yahoo Hadoop Cluster

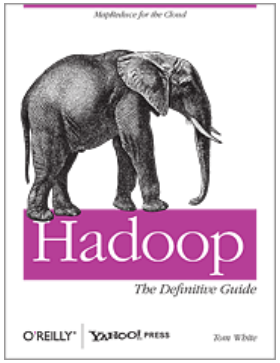
- System
  - 10+ Clusters
  - 25,000+ nodes
  - 3 Petabyte data
  - 700+ users
  - 10,000+ MR jobs/week
- Jobs
  - RSS feeds
  - Spams & Machine learning
  - Advertising optimization
  - Documents and Search index



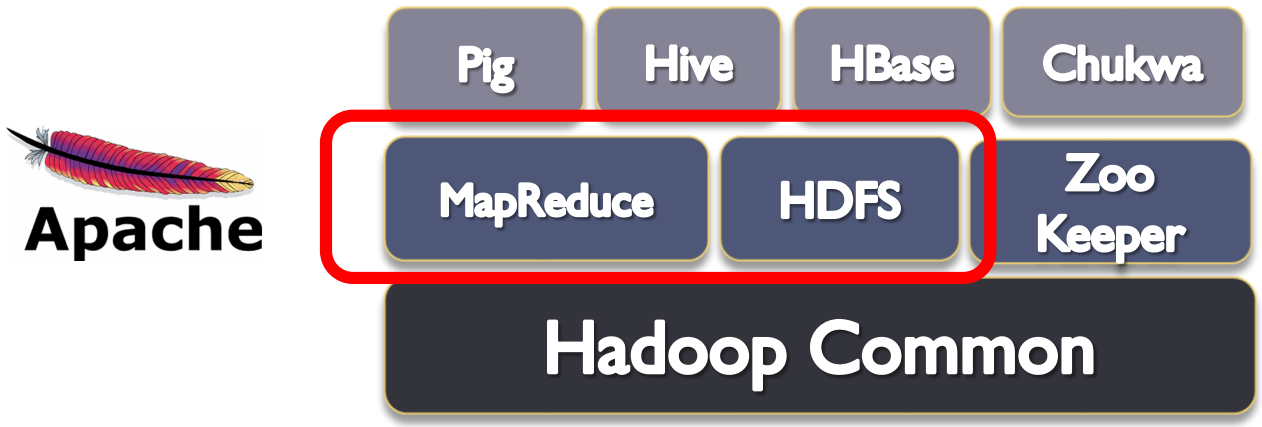
Apache Hadoop Project

- Large-scale, **data-intensive** distributed, reliable processing
  - TeraSort in 60 seconds
  - **WordCount**, Grep, Log Analysis
  - Web Graph Generation
  - Inverted Index Construction
  - Machine Learning, NaiveBayes, **K-means**, Expectation Maximization, ...





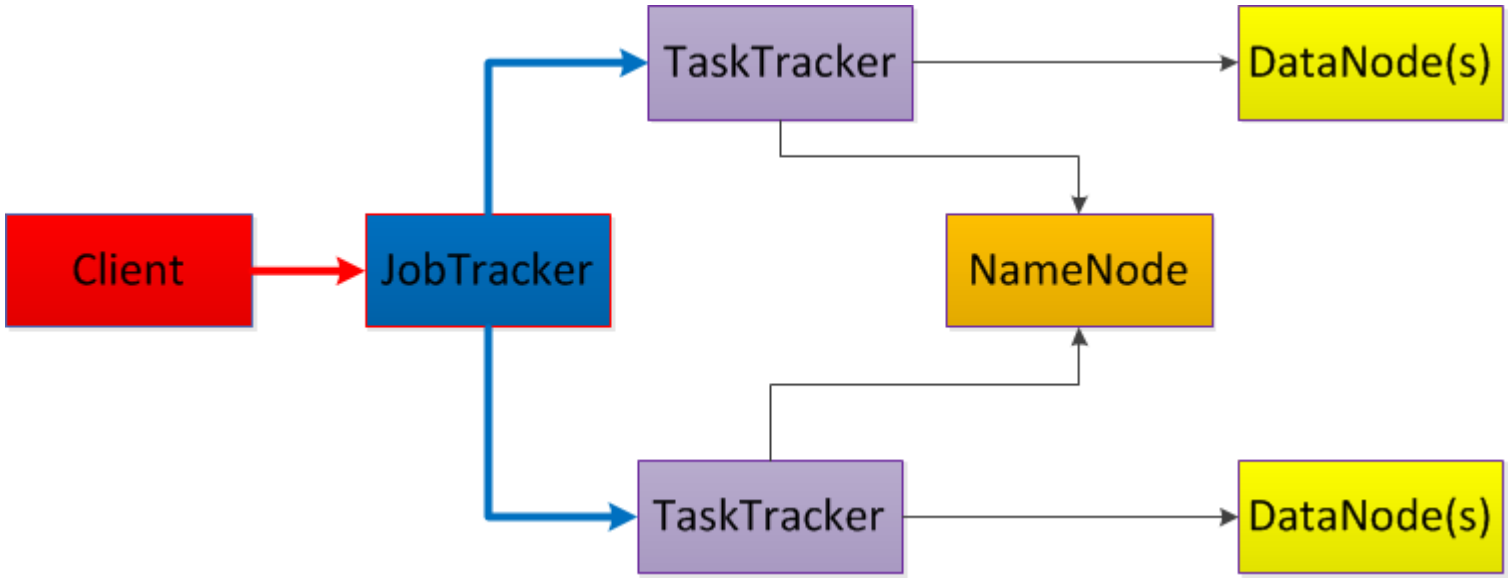
Hadoop Framework and Related Projects



- **Hadoop Common**: Libraries
- **MapReduce:** Distributed, data-intens. jobs
- **HDFS:** Distributed File System
- **HBase:** Storing and querying large-tables with thousands of columns
- **Hive:** Data warehouse with SQL-like query interface (HiveQL)
- **Pig:** Data stream processing language, parser, optimizer
- **ZooKeeper:** Coordination service
- **Chukwa:** Data logging (for auditing and testing)

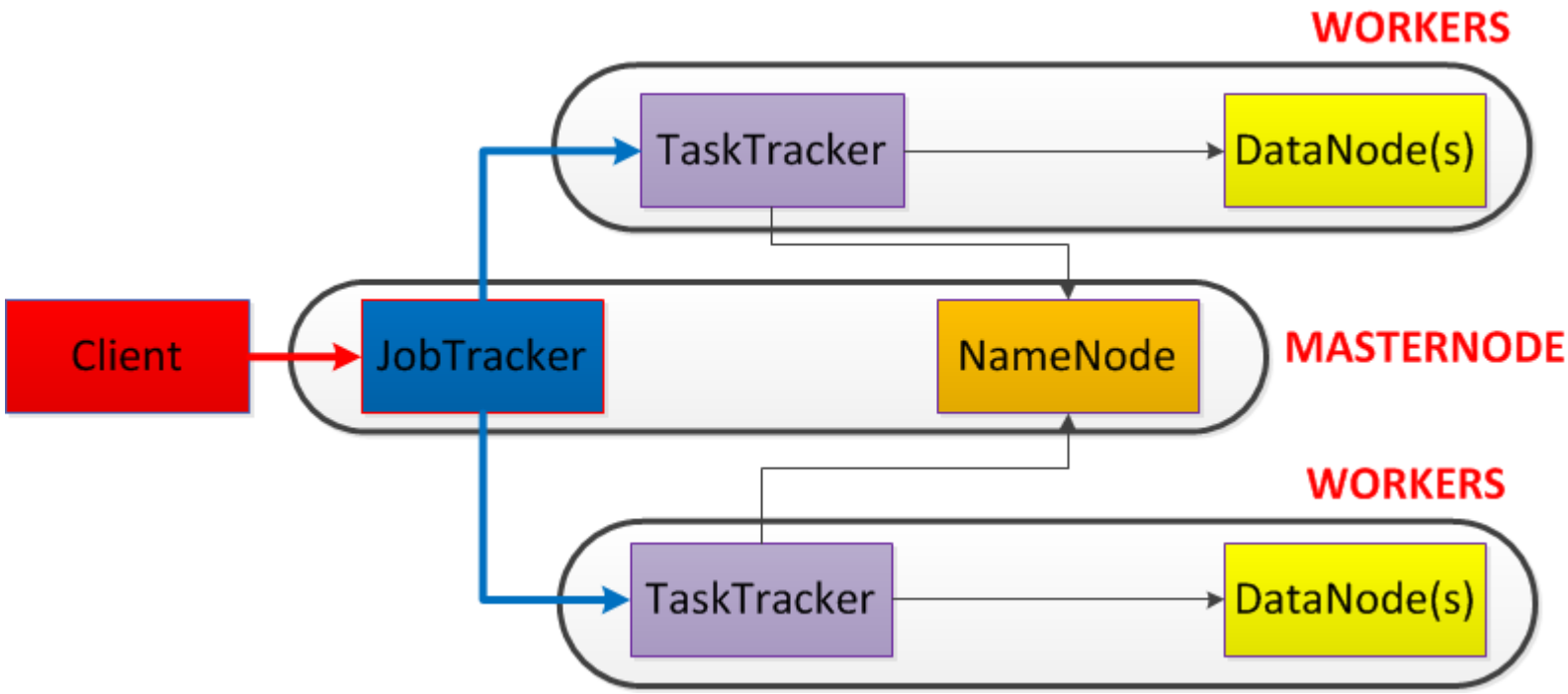
Hadoop Architecture

- Two main components
  - HDFS: NameNode, DataNode(s)
    - Others: Secondary NameNode, Checkpoint & Backup Nodes.
  - MapReduce: JobTracker, TaskTracker



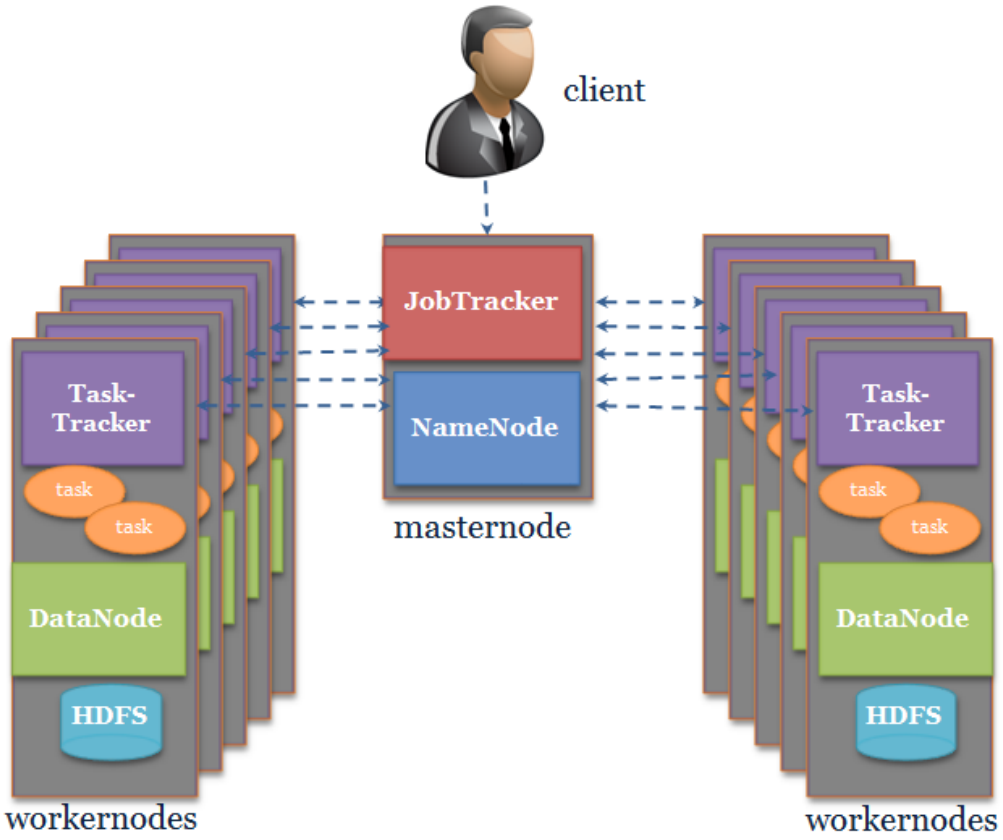
Hadoop Architecture

- From a physical machine perspective (client-server)



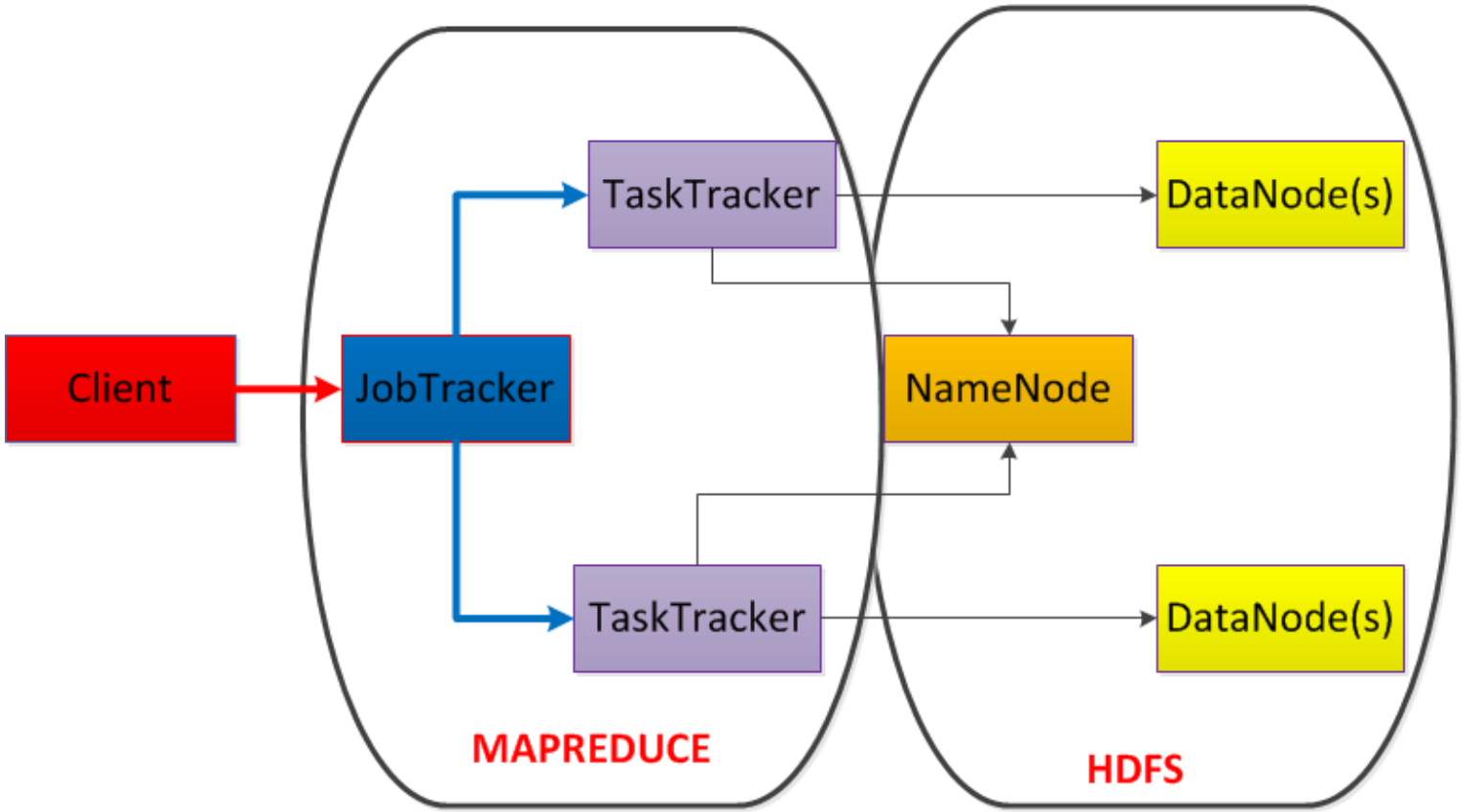
Hadoop Arhitecture

- From client-server perspective



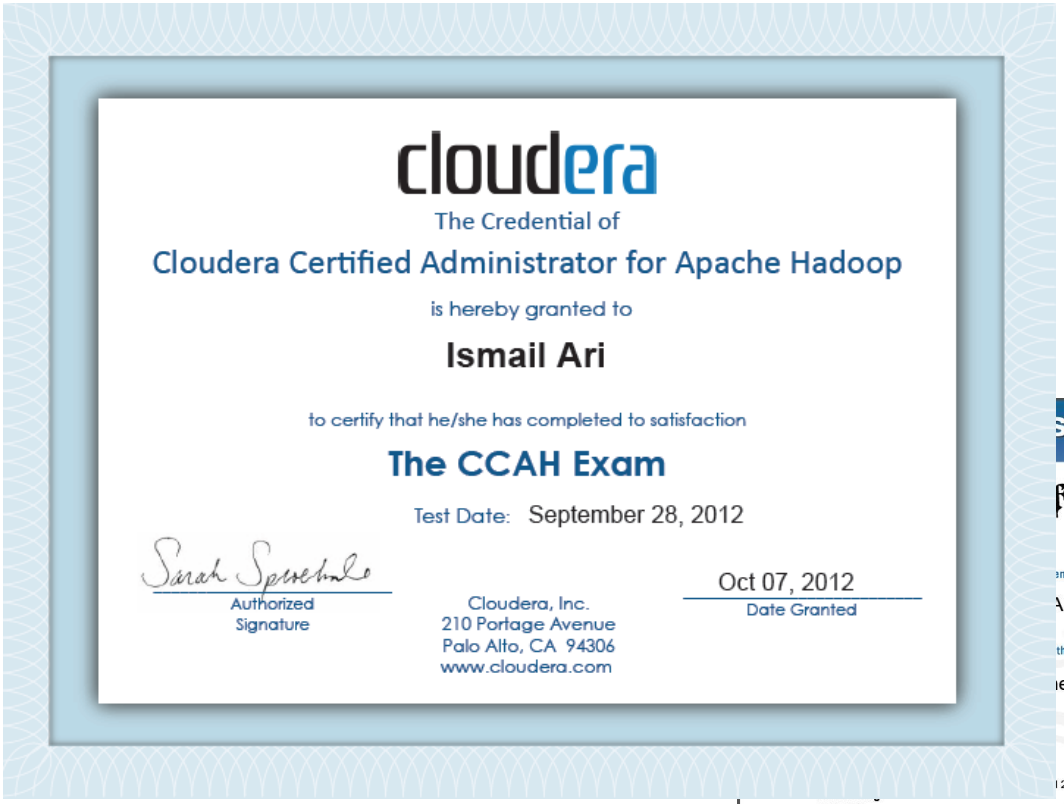
Hadoop Architecture

- A layered perspective or client-server view.



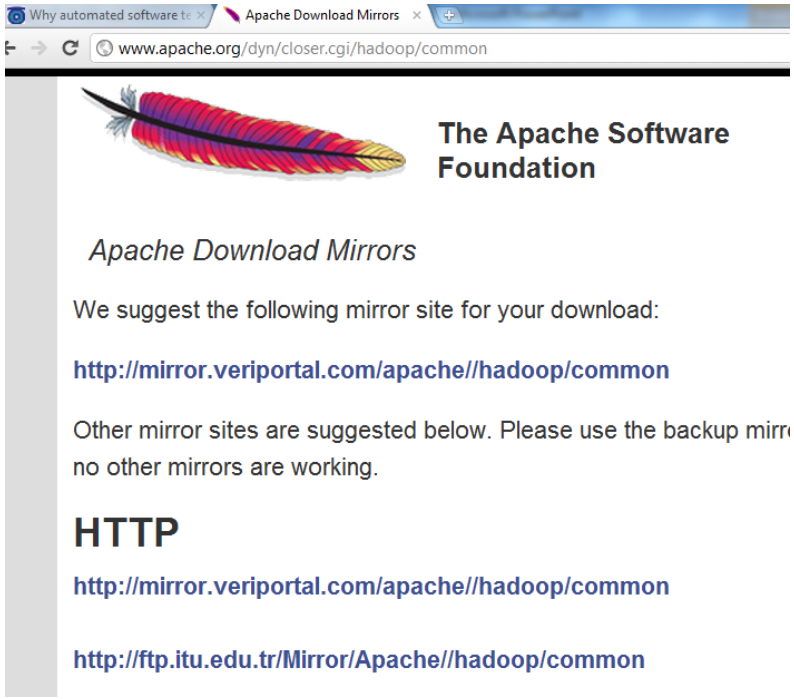
Hadoop Certification

- IBM's BigData University: http://bigdatauniversity.com
- Cloudera Certified Admin for Apache Hadoop (CCAH)



Where to download Hadoop?

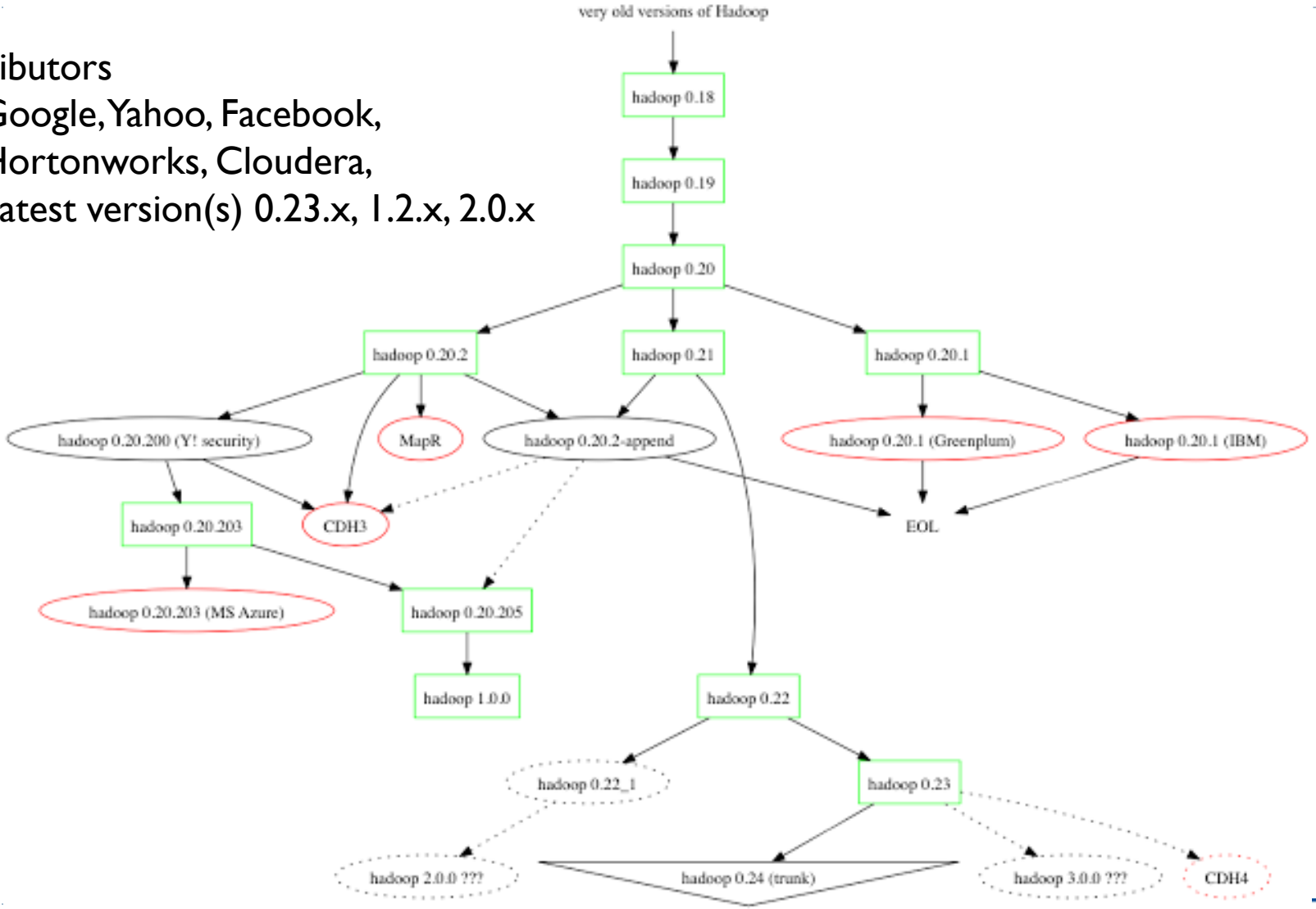
- Download Hadoop from Apache web site
  - www.apache.org/dyn/closer.cgi/hadoop/common
  - Releases: 0.23.x, 1.2.x, 2.0.x,



Hadoop Versions and Distributions

Contributors

Google, Yahoo, Facebook,
Hortonworks, Cloudera,
Latest version(s) 0.23.x, 1.2.x, 2.0.x



ÖZYEĞİN UNIVERSITY

How to install?

- Extract the hadoop archive (58MB)
  - Gunzip and Untar (tar –xvf)
- Configure "hadoop-env.sh" file
- Enter the master and slave machines' names to the "masters" and "slaves" files
- Configure "core-site.xml" and "hdfs-site.xml" files
- Format the NameNode
- Congratulations!
  - You have a single node Hadoop cluster

Running the Hadoop Cluster

- > start-all.sh
  - Starts both mapred and hdfs
- > stop-all.sh
  - Stops all the nodes and daemons

- Running modes:
  - LocalJobRunner (No HDFS , no services, just MR testing)
  - Psedo-distrib (same machine, multiple JVMs)
  - Fully-distributed

Hadoop – User and Admin Commands

- **HDFS user commands**

```
$ bin/hadoop fs -put food food
$ bin/hadoop fs -ls
$ bin/hadoop fs -cat food
$ bin/hadoop fs -rm food
$ bin/hadoop fs -mkdir juice
$ bin/hadoop fs –help (fs –du, etc.)
```

- **HDFS admin commands**

```
$ hadoop dfsadmin -report
$ hadoop dfsadmin -refreshNodes
$ hadoop dfsadmin -safemode leave
....
```

- **Java API for HDFS**

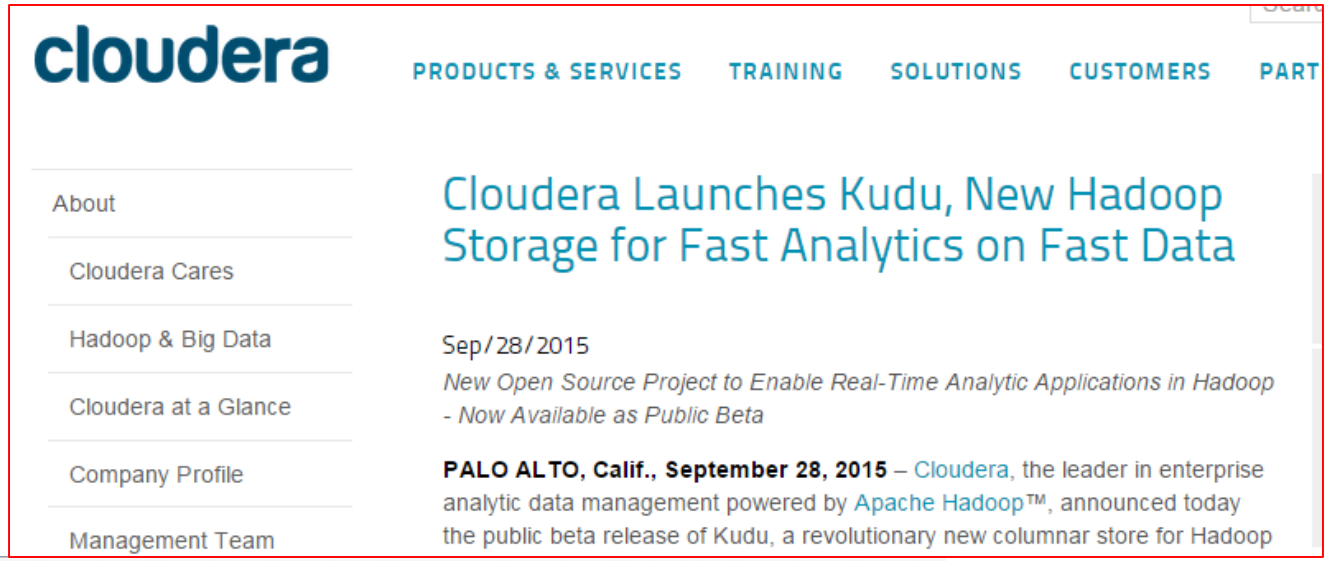
`org.apache.hadoop.fs.*`

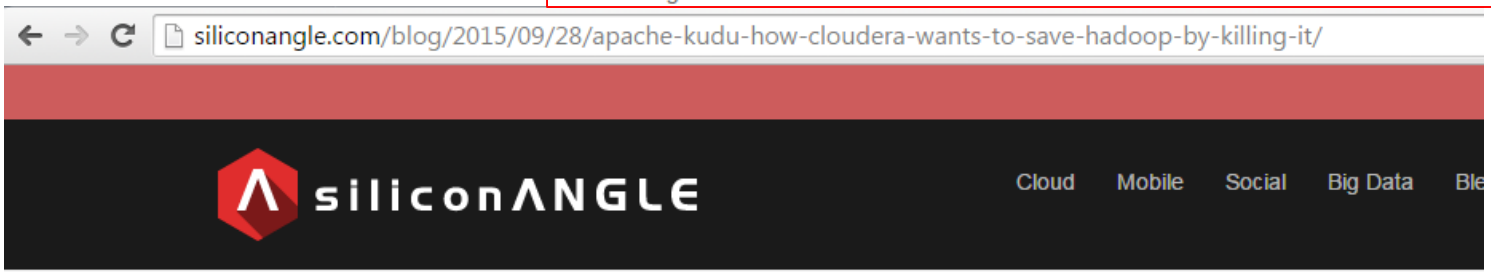
- **Web Interface (Demo)**

http://masternodeIP:50070
http://datanodeIP:50030

HADOOP IS NOT SO HOT ANYMORE

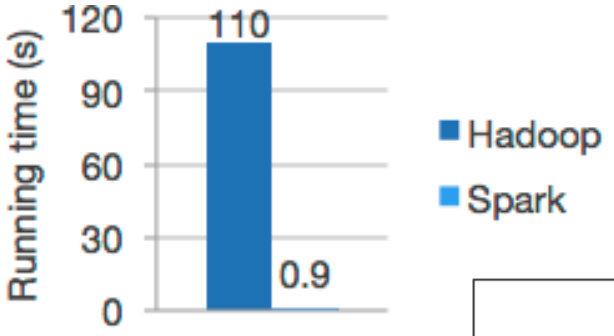
- What is ?





34
10

Apache Spark



← LOGIT performance of Hadoop vs. Spark

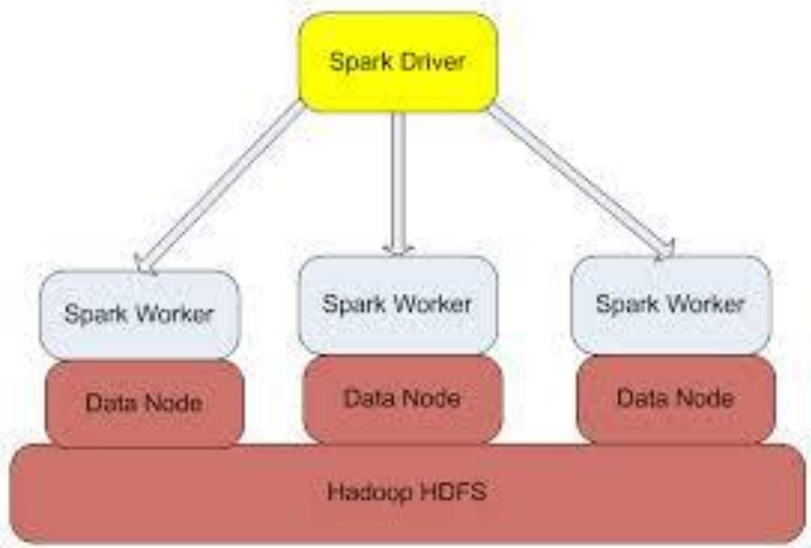
Recent Terasort benchmark: Spark has world record

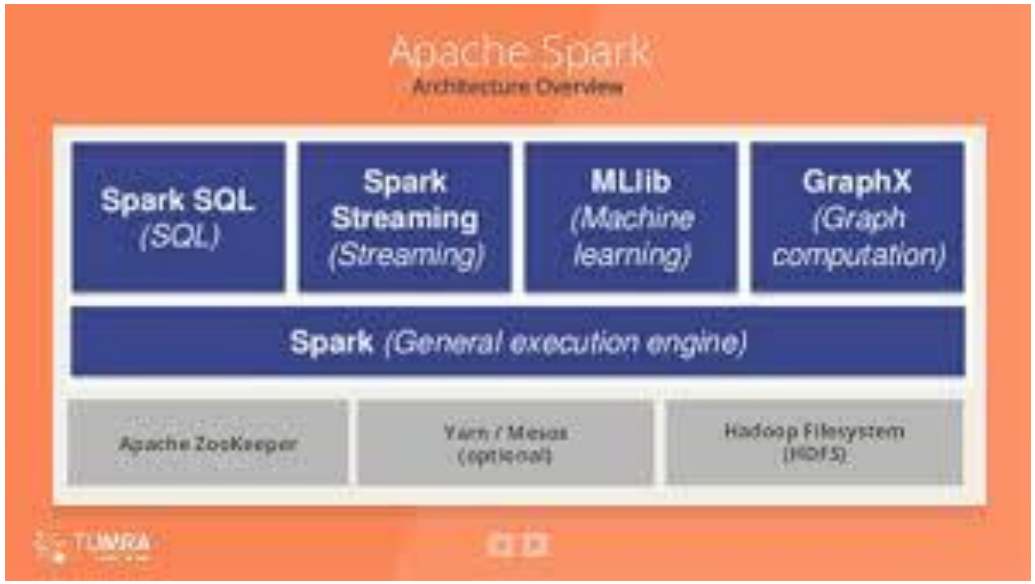
	Hadoop MR Record	Spark Record	Spark 1 PB
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s	570 GB/s
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network	virtualized (EC2) 10Gbps network
Sort rate	**1.42 TB/min**	**4.27 TB/min**	**4.27 TB/min**
Sort rate/node	**0.67 GB/min**	**20.7 GB/min**	**22.5 GB/min**

3x Performance with 10x Less Machines

Spark Architecture

- spark.apache.org

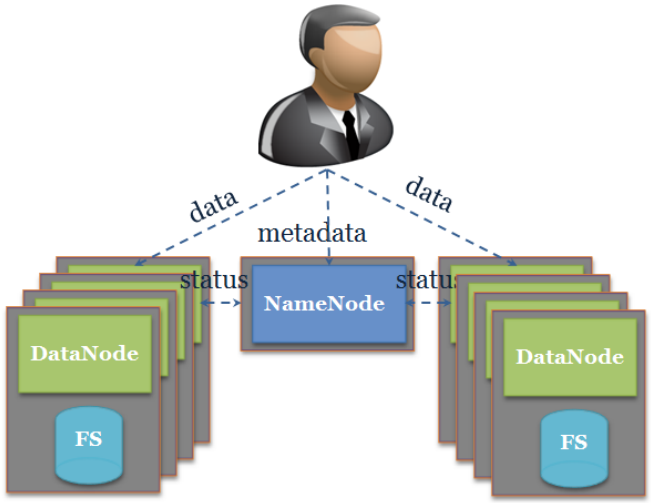




ÖZYEĞİN UNIVERSITY

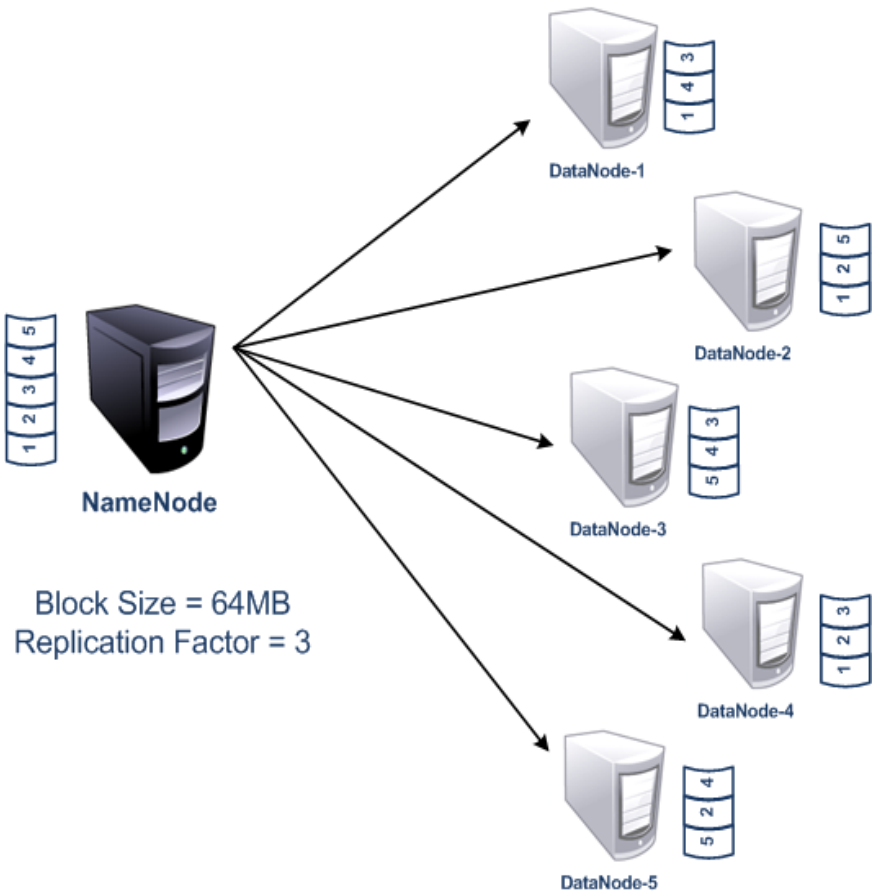
Back to HDFS – NameNode, DataNode

- **NameNode:**
  - Filesystem naming operations: Open, close, rename file & dir.
  - Tracks block locations on DataNodes



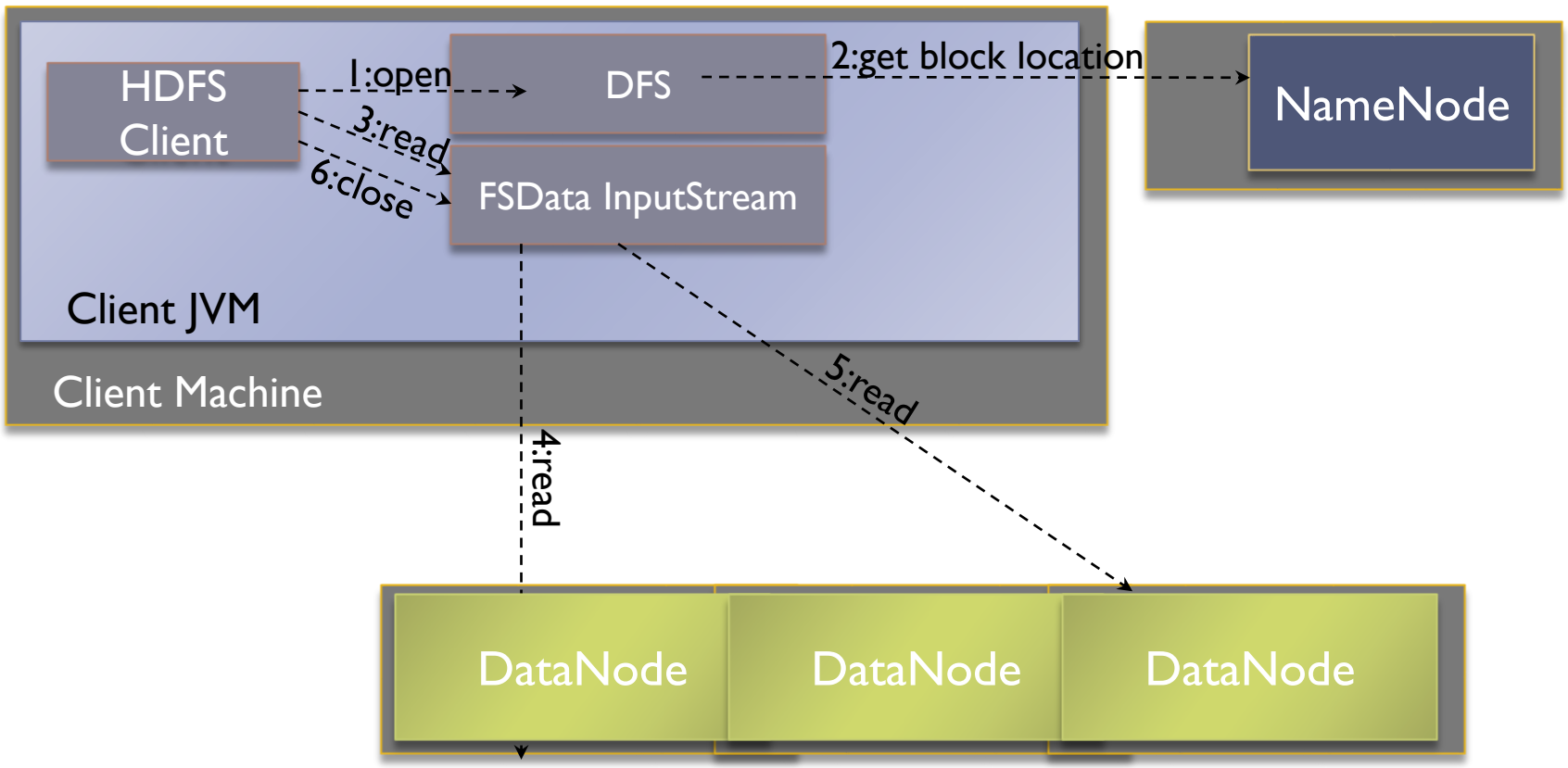
- **DataNodes:**
  - Hold data blocks
  - Clients send data blocks directly to DataNodes
  - Clients read data blocks directly from DataNodes as well

Replication and Locality in Hadoop



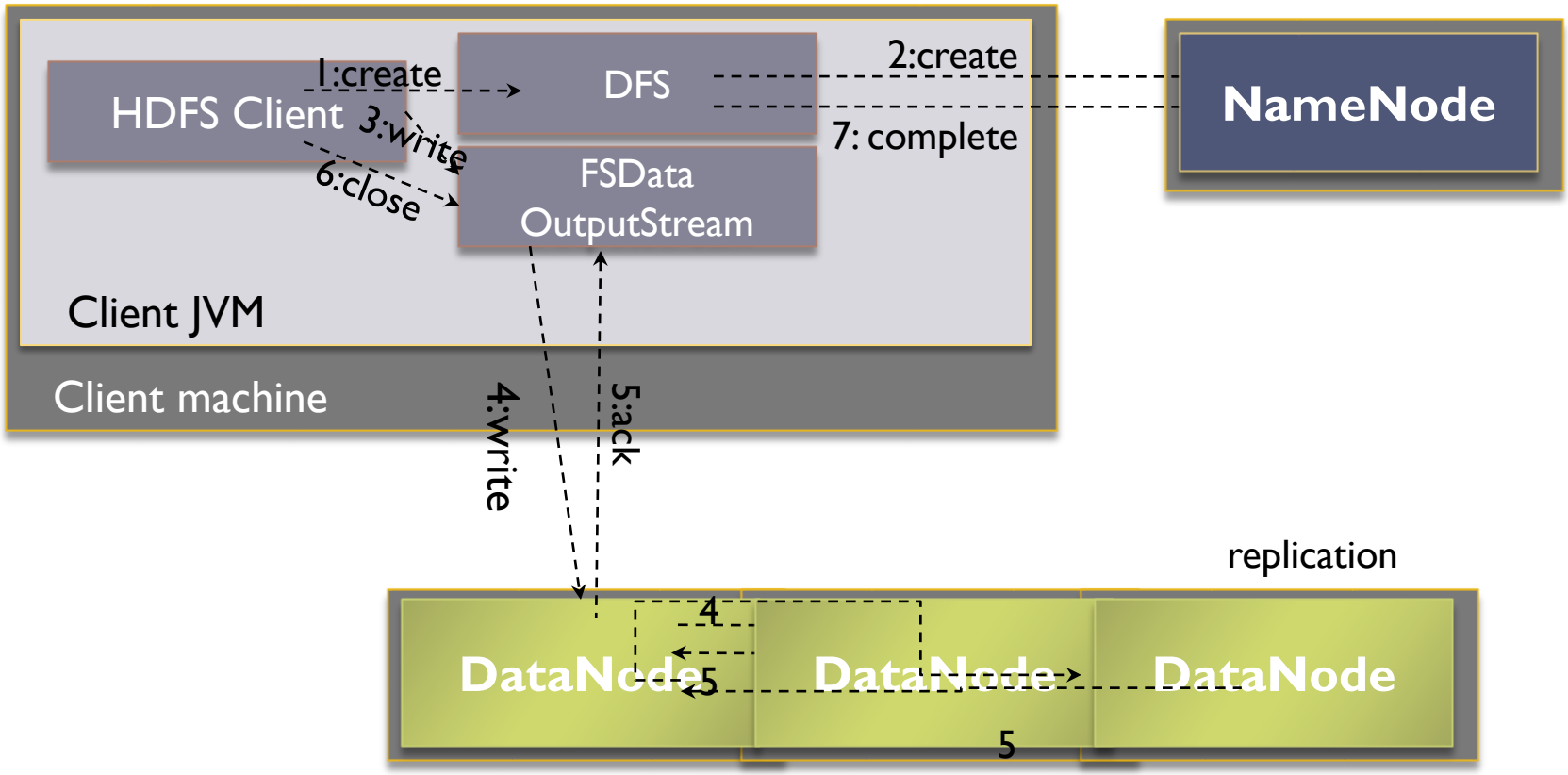
- Replication
  - 64MB Block striping
  - 3 replicas on datanodes
- Rack-aware placement
  - A Hadoop CLUSTER can consist of one or more RACKS
  - Rack: 30-40 NODES (computers) connected to the same network switch and are physically close together
  - Hadoop has network topology awareness

HDFS Read


HDFS Client
1:open
DFS
2:get block location
NameNode
3:read
6:close
FSData InputStream
Client JVM
Client Machine
5:read
4:read
DataNode
DataNode
DataNode

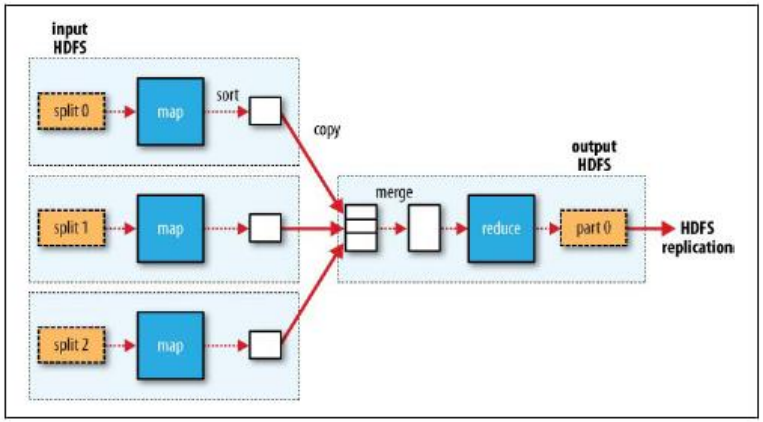
HDFS Write

For each data block



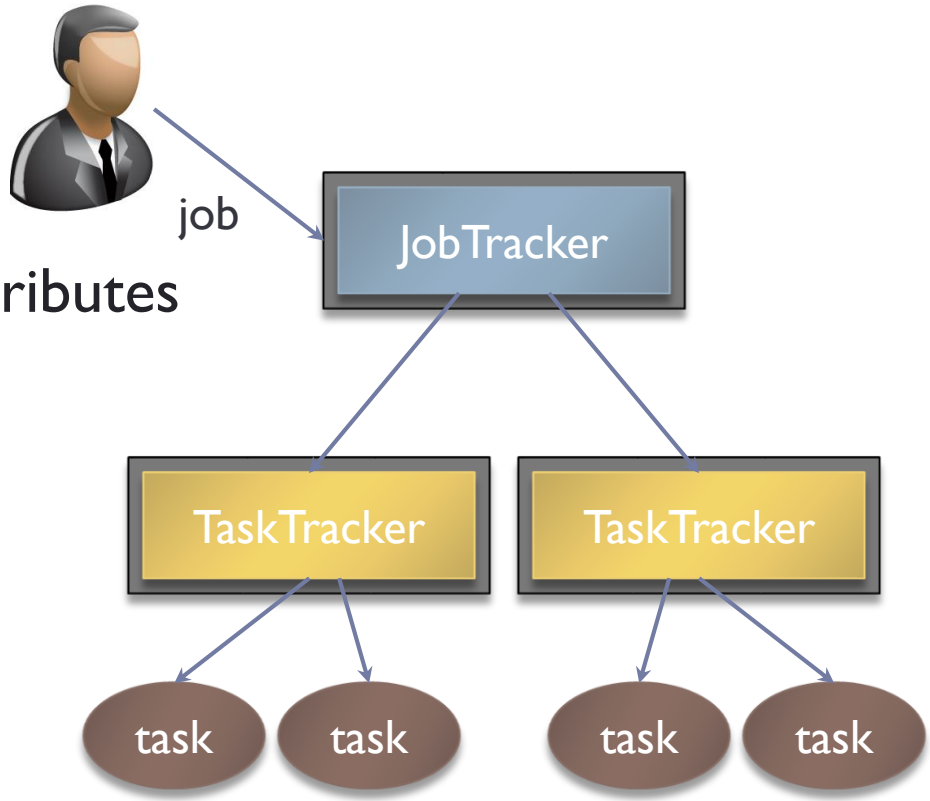
Hadoop – MapReduce

- Automated job distribution, task tracking and fault-tolerance framework for parallel data processing
- Provides programming API
  - (JAVA, perl, Phyton, php,C,C++)
  - `org.apache.hadoop.mapred.* OLD`
  - `org.apache.hadoop.mapreduce.* NEW`
- Has data processing pipelines
- Web-based Status tracking
  http://datanode:50030

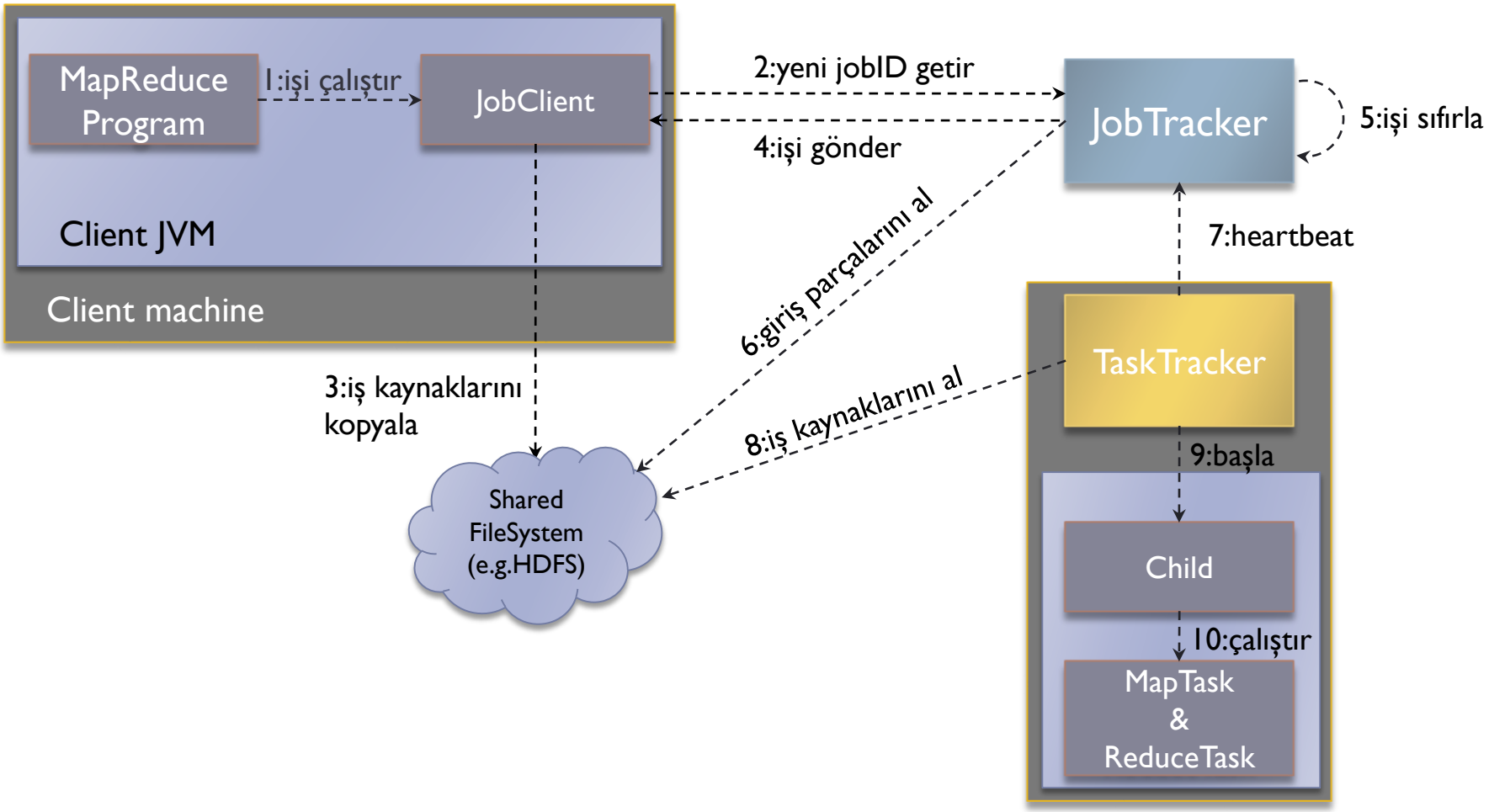


MapReduce Topology

- **JobTracker,**
  - Starts MR jobs
  - Selects TaskTrackers and distributes loads based on data locality
- If a **TaskTracker** fails, its job is remapped to another node
- Jobs will stop if JobTracker fails



MapReduce Operation



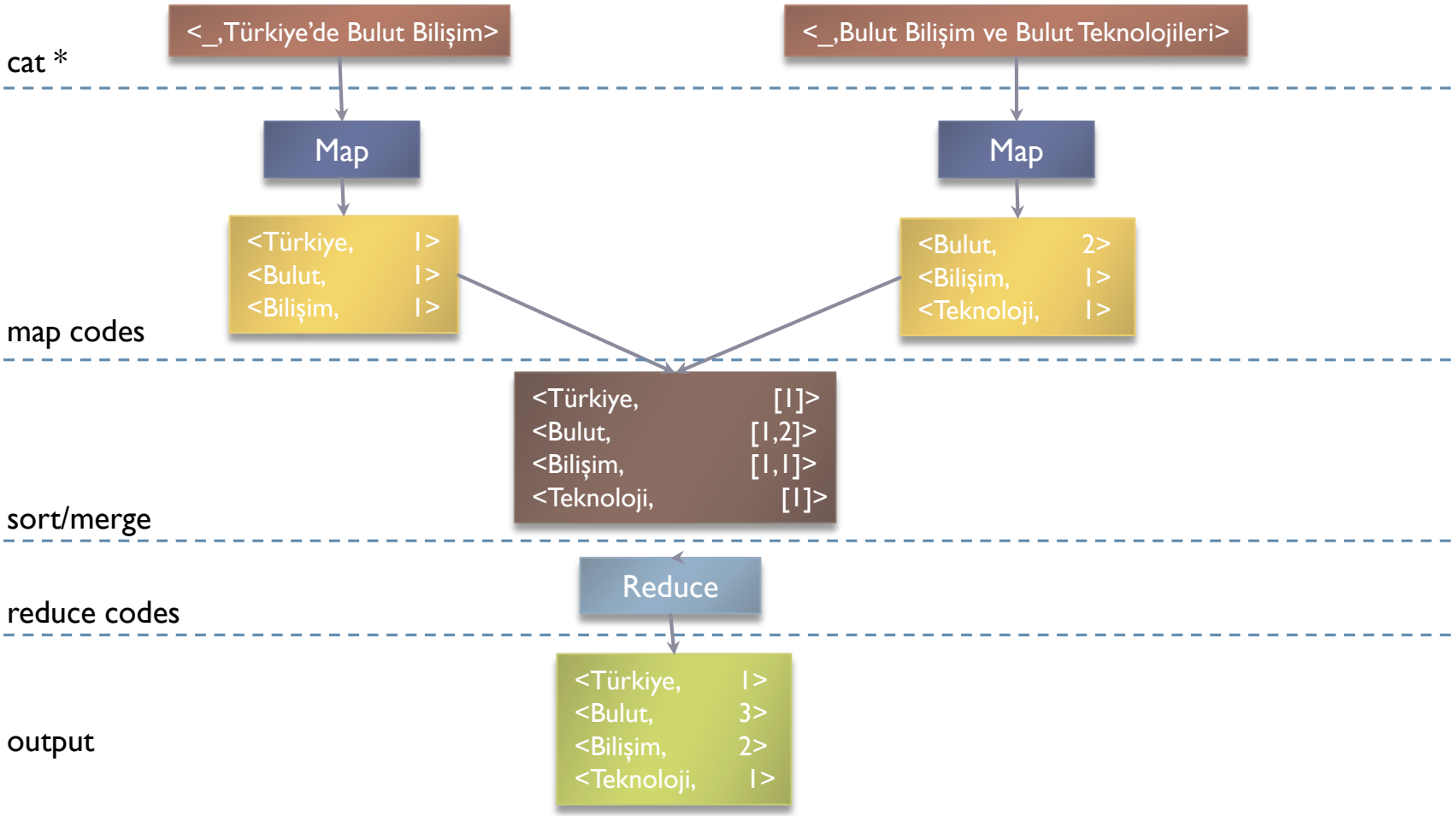
MapReduce Operation – Logical Example

File 1:

"Türkiye'de Bulut Bilişim"

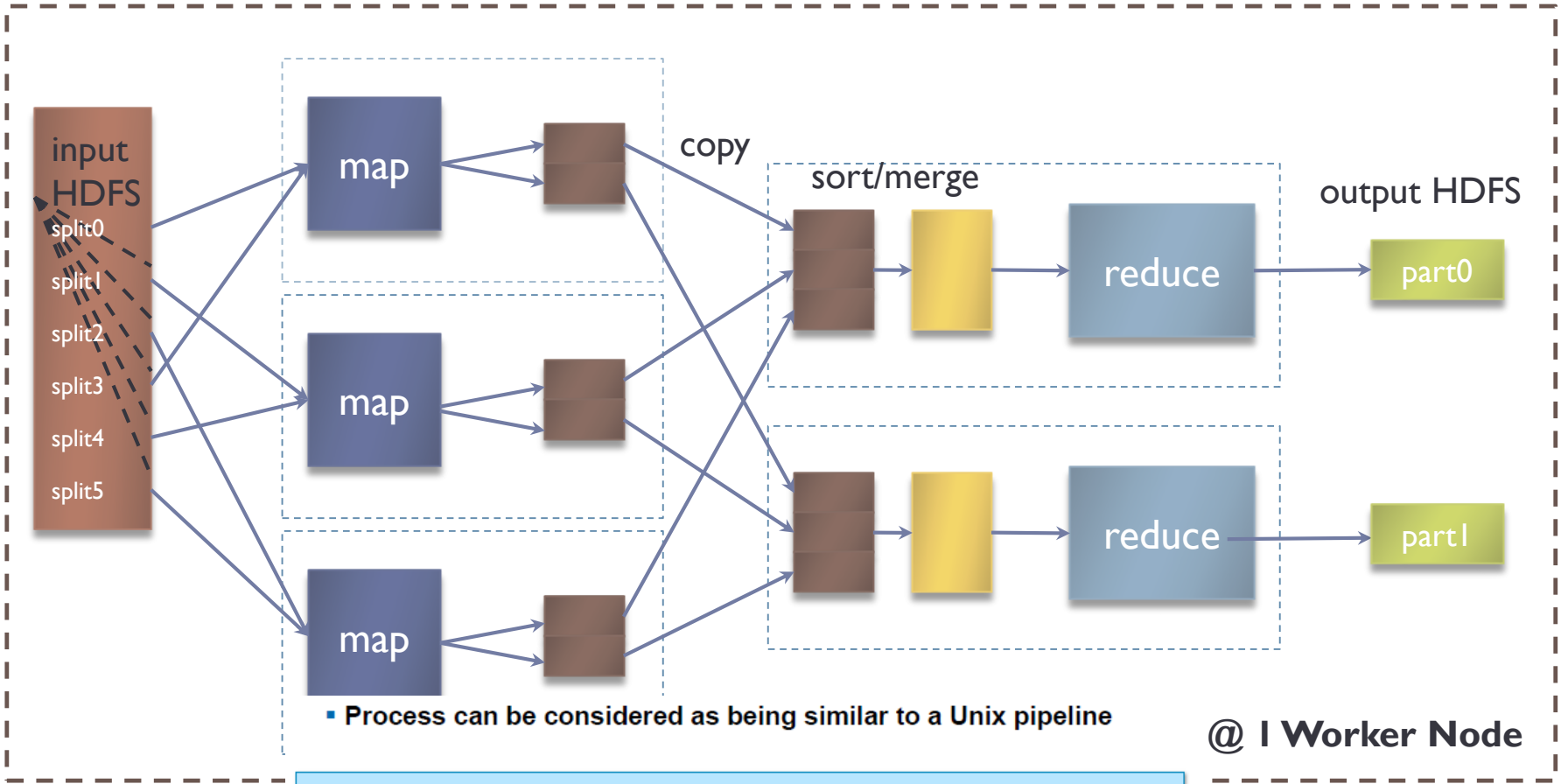
File 2:

"Bulut Bilişim ve Bulut Teknolojileri"

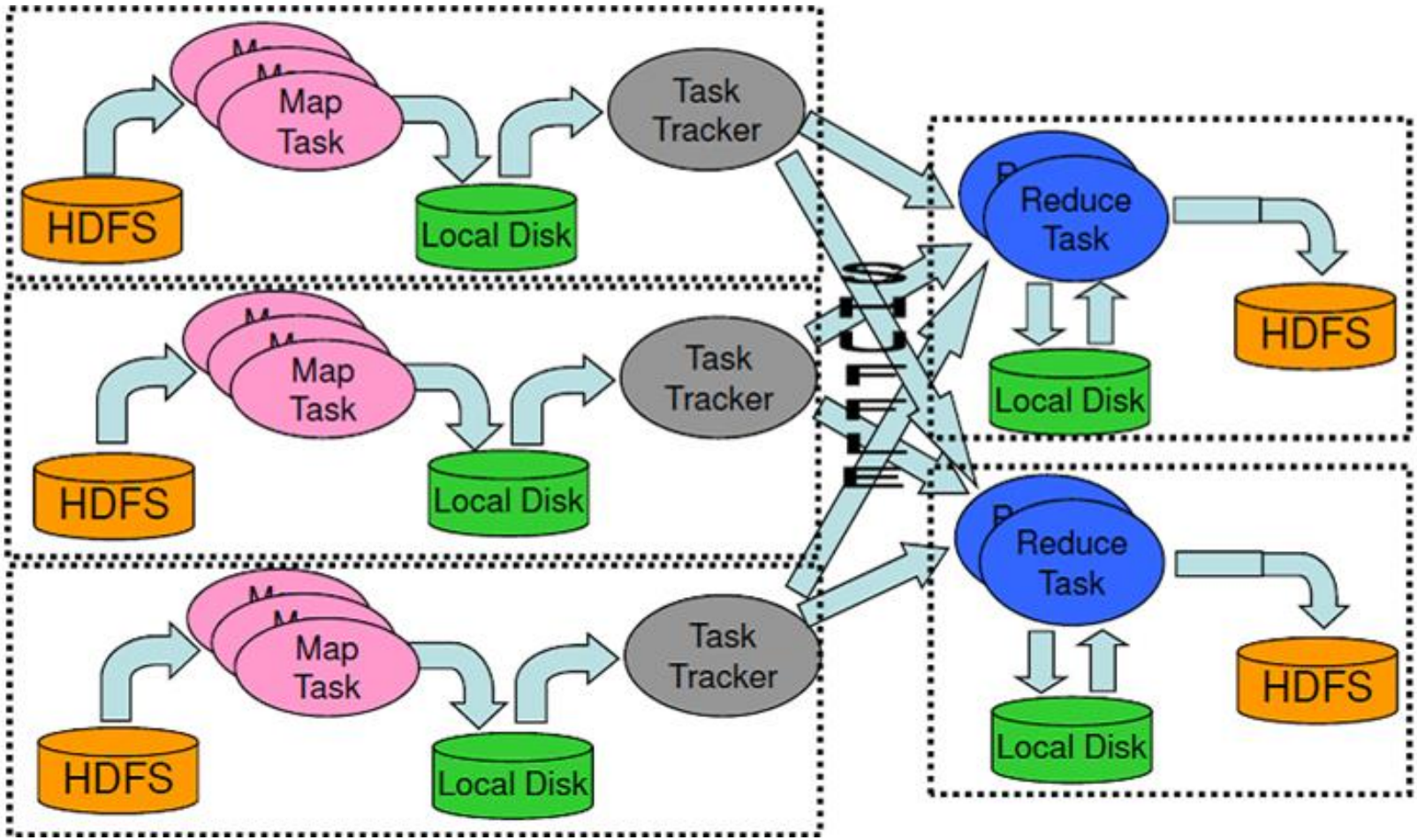


ÖZYEĞİN UNIVERSITY

MapReduce Operation – Physical



HDFS-MapReduce Operation Together



From Tuning Hadoop for Performance by Srigurunath Chakravarthi

Hadoop - MapReduce

- MapReduce program execution
  - **> hadoop fs -put /test/input/testData/ /input**
  - **> hadoop jar hd-example.jar wordcount /input output**
  - **> hadoop fs –cat output/*00**

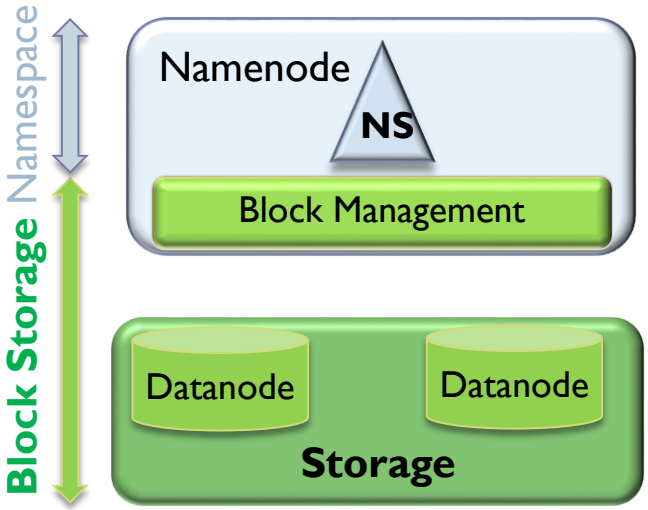
Hadoop 0.23 (dotNext) Highlights

- Overview and Introduction
  - HDFS Federation
  - MapReduce v2 (YARN – Yet Another Resource Negotiator)
  - http://grid.chu.edu.tw/cloudcom2012/slides

- Primary focus is scalability
  - HDFS Federation – larger namespace & scalability
  - YARN aka MRv2 – Job run reliability
    - Agility & Evolution

From: CloudCom 2012 Invited Talk, Viraj Batt of Yahoo

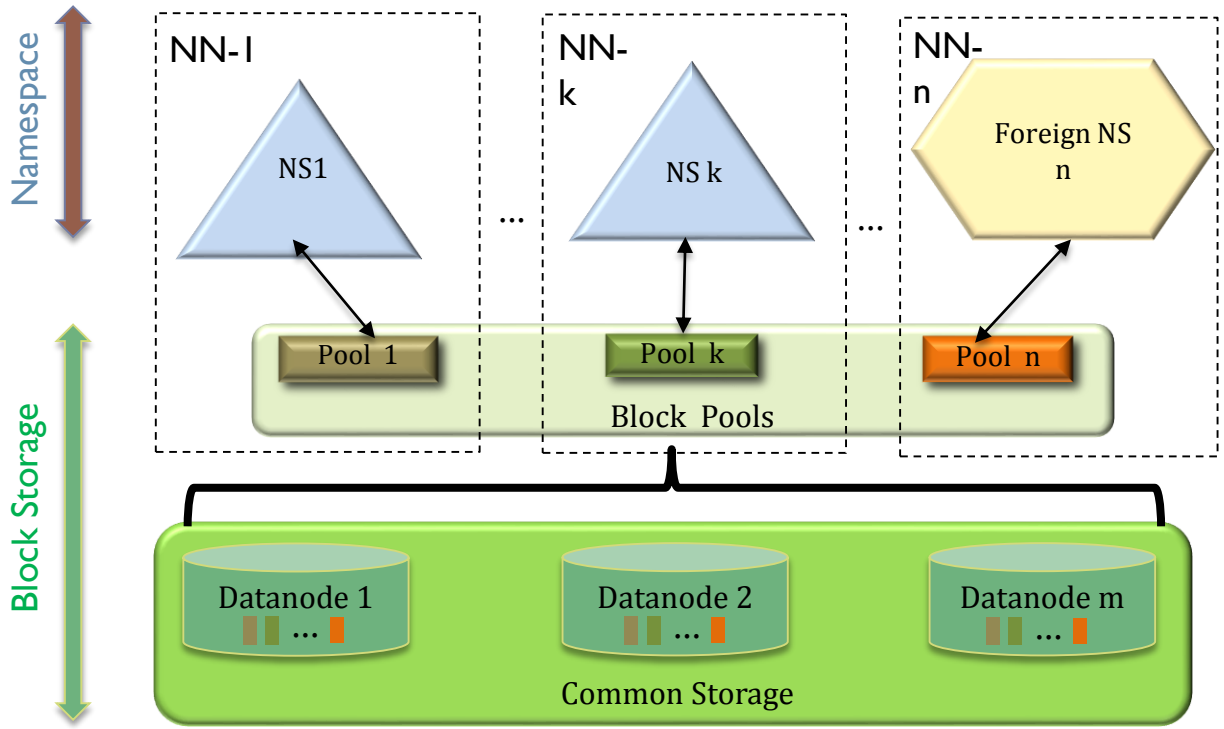
Non Federated HDFS Architecture



Implemented as

- Single ***Namespace Volume***
  - Namespace Volume = Namespace + Blocks
- Single namenode with a namespace
  - Entire namespace is in memory
  - Provides Block Management
- Datanodes store block replicas
  - Block files stored on local file system

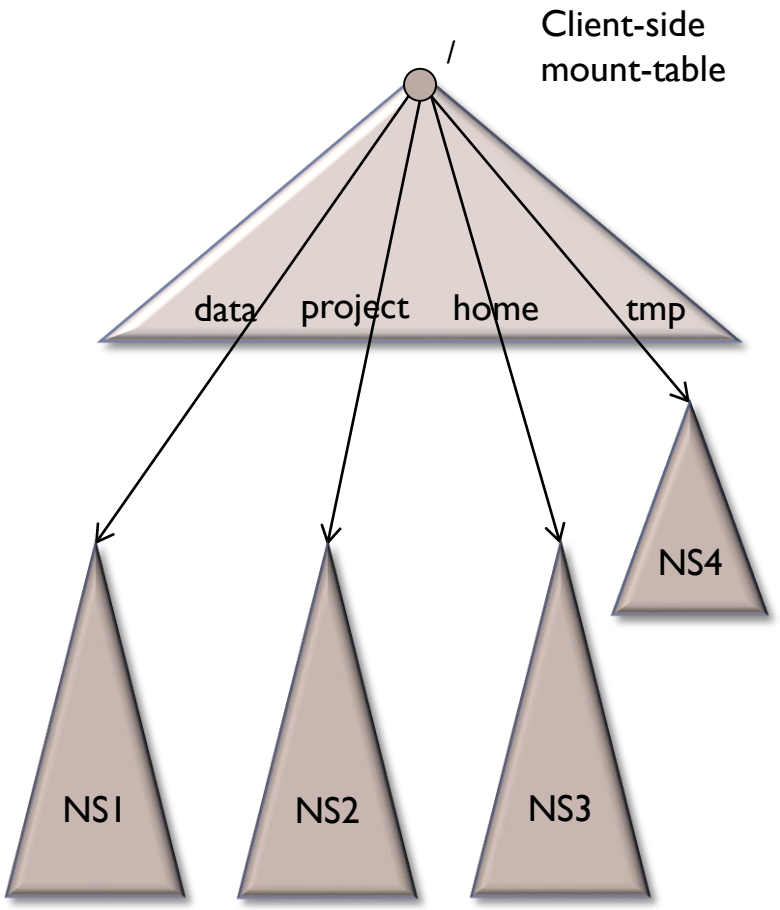
HDFS Federation



- It is an administrative/operational feature for better managing resources
- Multiple **independent** Namenodes and Namespace Volumes in a cluster
  - Namespace Volume = Namespace + Block Pool
  - Set of blocks for a Namespace Volume is called a ***Block Pool***
  - DNs store blocks for all the Namespace Volumes – no partitioning

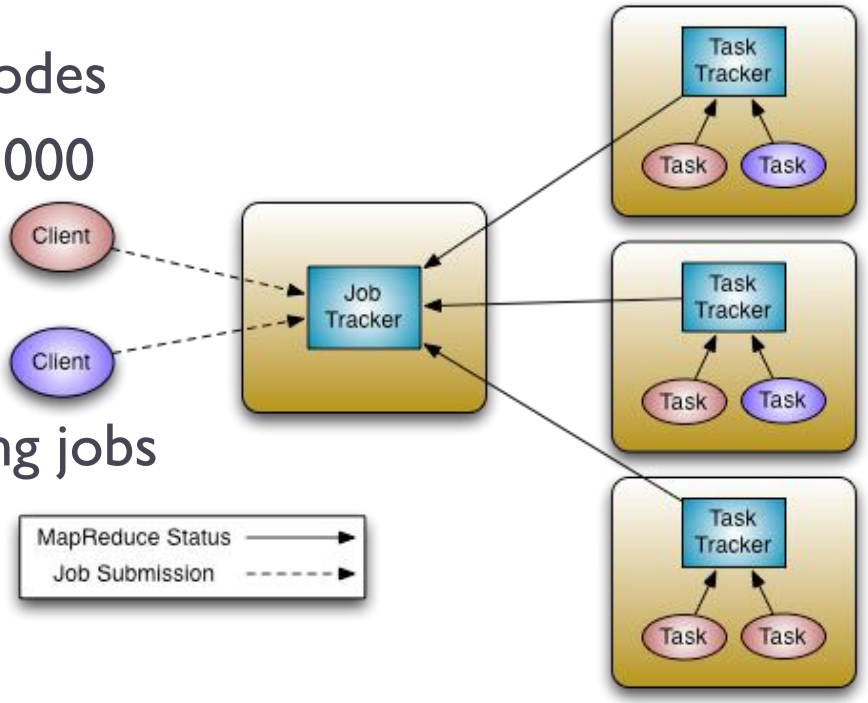
Managing Namespaces

- Federation has multiple namespaces – Don't we need a single global namespace?
  - Key is to share the data and the names used to access the data
- A global namespace is one way
  - Client-side mount table is another way to share.



Current Limitations of the Job Tracker

- Scalability
  - Maximum Cluster size – 4,000 nodes
  - Maximum concurrent tasks – 40,000
- Single point of failure
  - Failure kills all queued and running jobs
  - Jobs need to be re-submitted by users



- Restart is very tricky due to complex state
- Hard partition of resources into map and reduce slots

Current Limitations of the Job Tracker

- Lacks support for alternate paradigms
  - Iterative applications implemented using MR are 10x slower.
  - Example: K-Means, PageRank

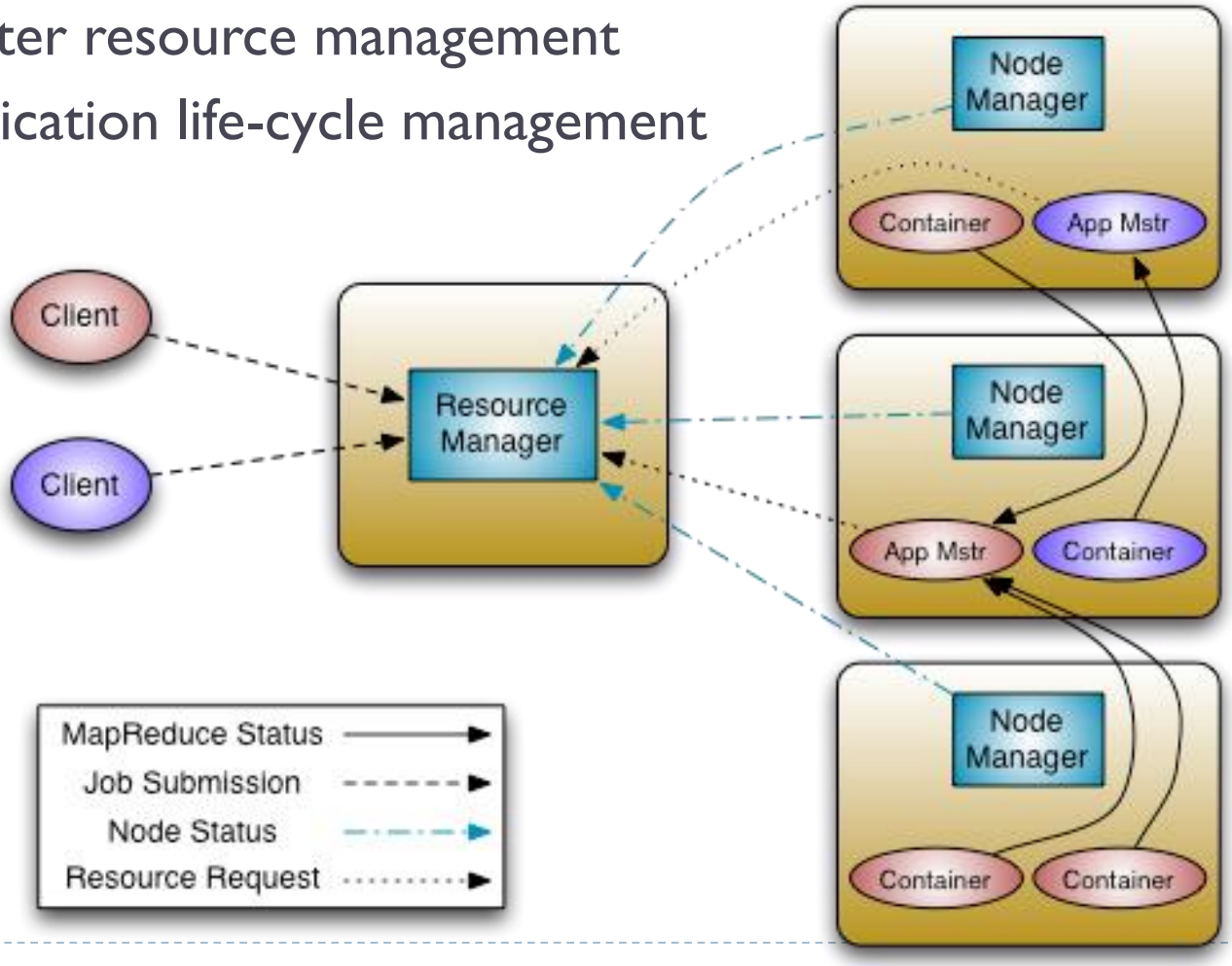
- Lack of wire-compatible protocols
  - Client and cluster must be of same version
  - Applications and workflows cannot migrate to different clusters

Design Theme for YARN

- Scalability - Clusters of 6,000-10,000 machines
  - Each machine with 16 cores, 48G/96G RAM, 24TB/36TB disks
  - 100,000 concurrent tasks
  - 10,000 concurrent jobs
- Agility & Evolution – Ability for customers to control upgrades to the grid software stack.
- Reliability and Availability
- Wire Compatibility

Design Methodology

- Split up the two major functions of JobTracker
  - Cluster resource management
  - Application life-cycle management



Architecture of YARN

- Resource Manager
  - Global resource scheduler
  - Hierarchical queues
- Node Manager
  - Per-machine agent
  - Manages the life-cycle of container
  - Container resource monitoring
- Application Master
  - Per-application
  - Manages application scheduling and task execution

Improvements on current Job Tracker

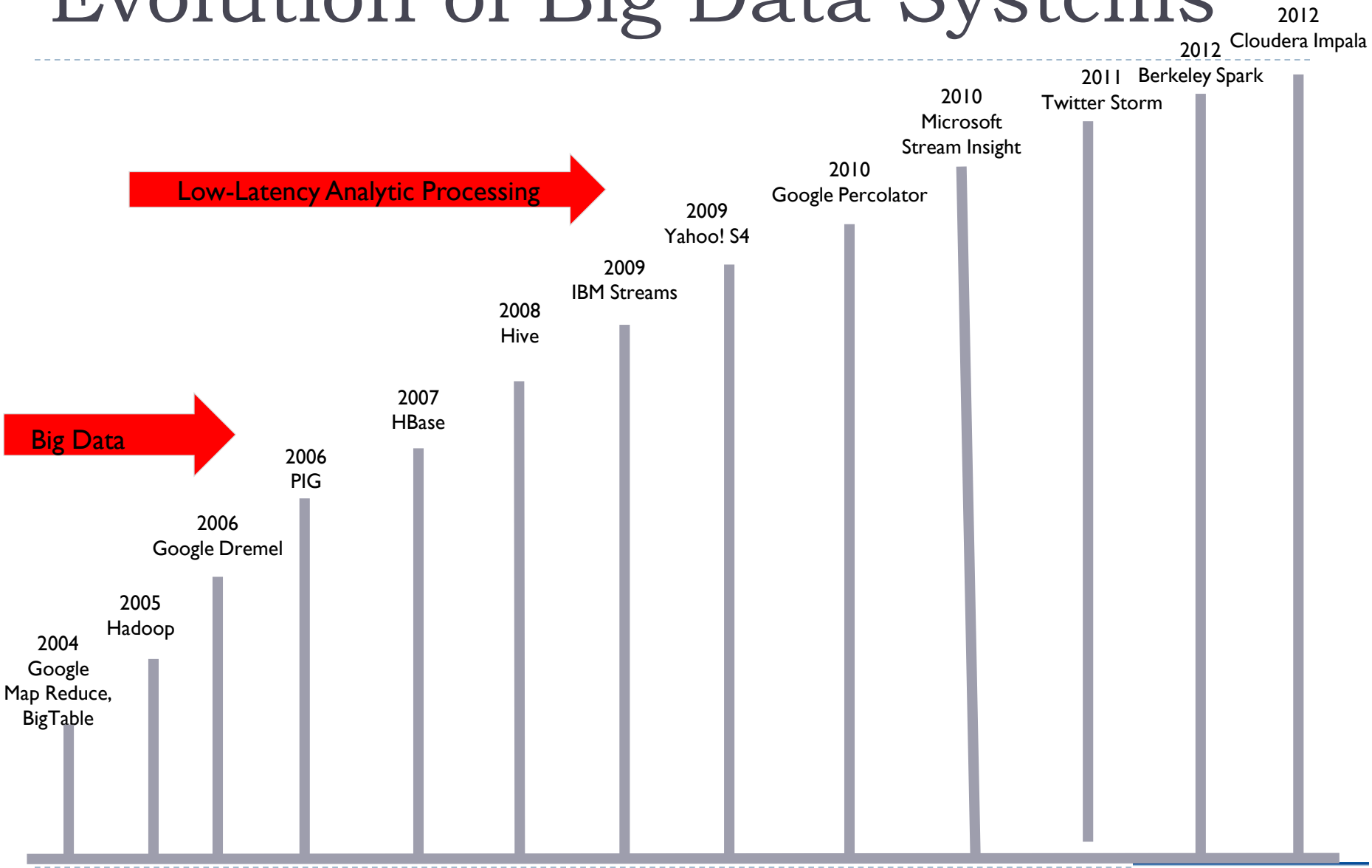
- Innovation and Agility
  - Multiple versions of MapReduce can run in the same cluster
    - Faster deployment cycles for improvements
    - Customers upgrade MapReduce versions on **their** schedule
  - **Users can run customized versions of MapReduce**
    - HOP (Hadoop Online Prototype): MapReduce that allows data to be *pipelined* between tasks and between jobs
- Performance Improvements
  - Small Job Optimizations

Improvements on current Job Tracker

- Support for programming paradigms other than MapRed.
  - MPI (OpenMPI)
  - Master-Worker
  - Machine Learning (Spark)
  - Iterative processing (Yahoo S4 and Twitter Storm)
  - Graph Processing (Giraph)
  - Run all on the same Hadoop cluster

- (Goto: EUROPAR 2013 PRESENTATION)

Evolution of Big Data Systems



From: CloudCom 2012 Invited Talk, Viraj Batt of Yahoo

Private Cloud & Hadoop Market

- Private Cloud Market ~55 billion (100bil x 55%)
- Hadoop + NoSQL ~10 billion by 2014

http://www.marketresearchmedia.com/?p=568
http://slashdot.org/topic/bi/hadoop-market-will-hit-2-2-billion-by-2018-report/
http://www.forbes.com/sites/joemckendrick/2012/02/13/cloud-computing-market-hot-but-how-hot-e

- Hadoop Distros and Support Groups
  - Apache Hadoop
  - **Hortonworks**, Yahoo, Microsoft
  - **Cloudera** CDH, IBM, Oracle
  - **MapR** MR7 (C, VFS), Google, Amazon
  - Twitter, Facebook

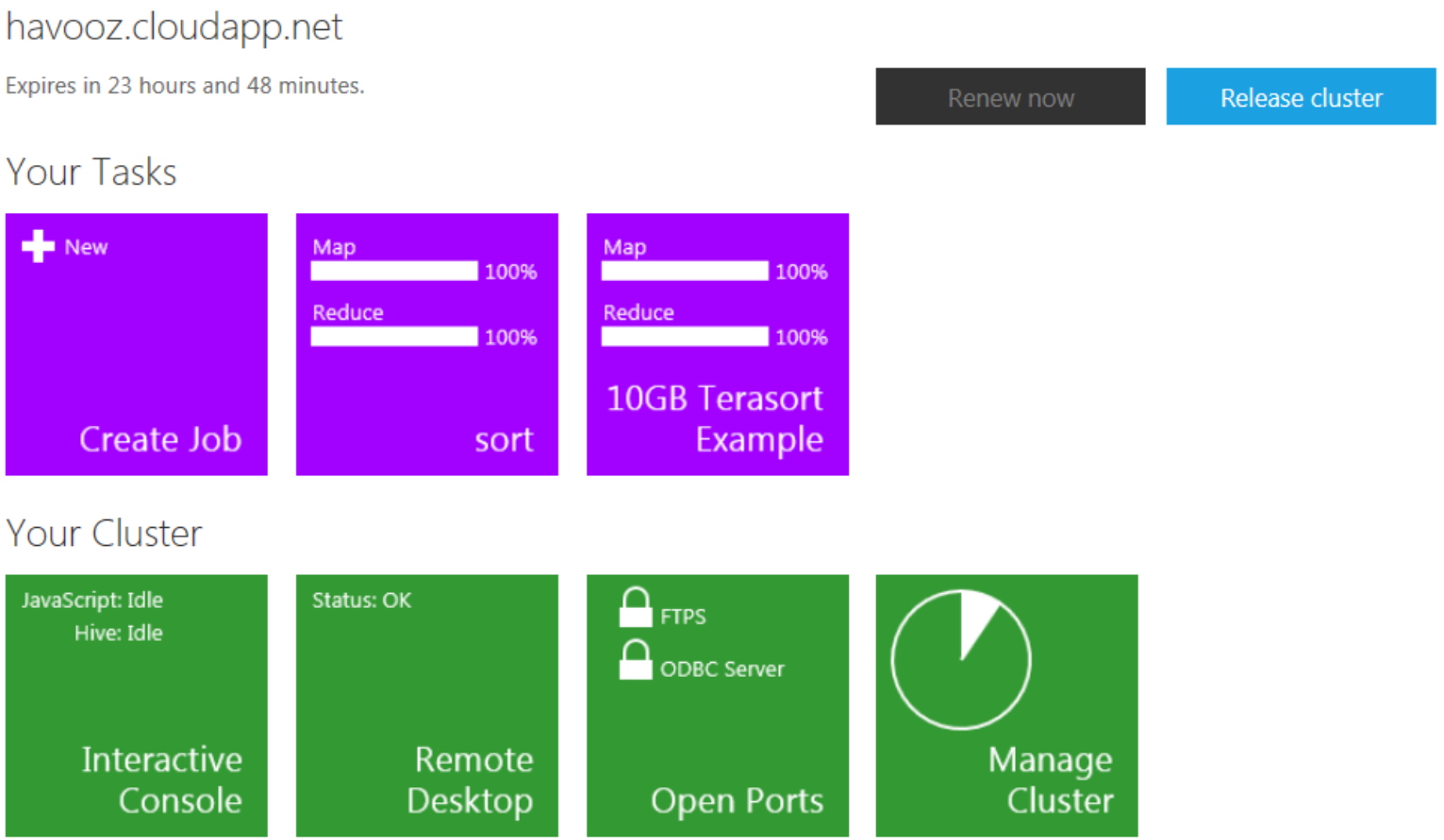
Cloud and Big Data Companies

Hadoop on Public Cloud

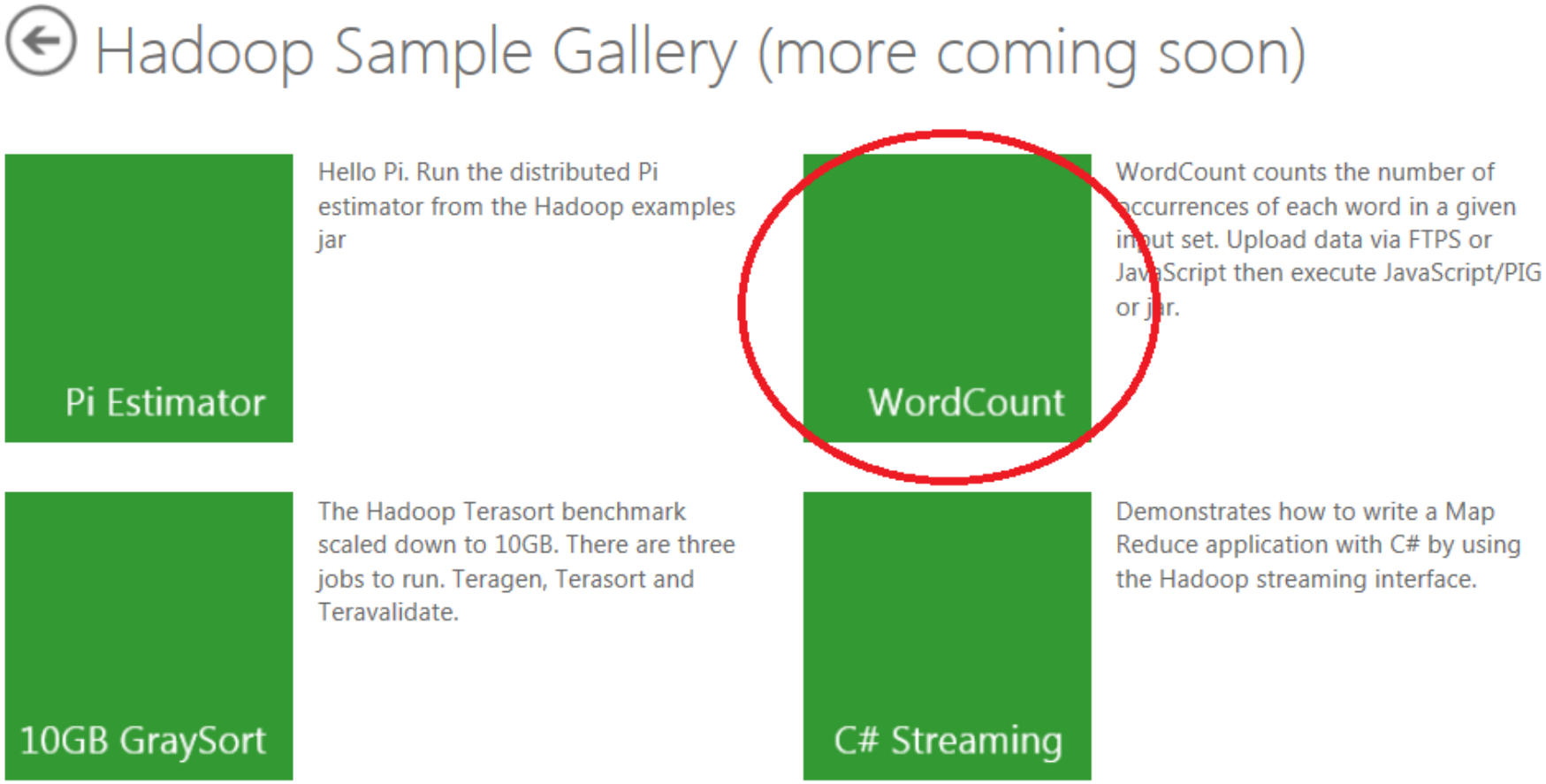
- Amazon AWS: Amazon Elastic MapReduce, Dynamo
  - You probably know Amazon EC2, S3, EBS
- Microsoft Azure – HadoopOnAzure, HD Insight
- EMC/Greenplum: Pivotal HD
- IBM Enterprise Cloud
  - IBM – Compro: cloudpro.com.tr

Hadoop on Azure

- http://hadooponazure.com

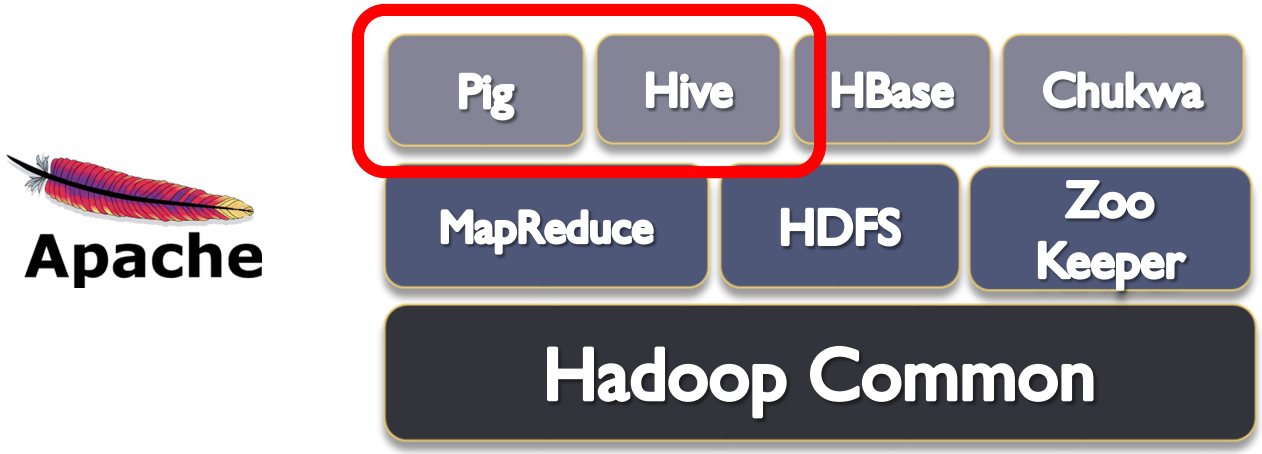


Hadoop on Azure Samples



(Goto: BIG DATA STARTUPS PRESENTATION)

Hadoop Framework and Related Projects



- **Hadoop Common**: Libraries
- **MapReduce:** Distributed, data-intens. jobs
- **HDFS:** Distributed File System
- **HBase:** Storing and querying large-tables with thousands of columns
- **Hive:** Data warehouse with SQL-like query interface (HiveQL)
- **Pig:** Data stream processing language, parser, optimizer
- **ZooKeeper:** Coordination service
- **Chukwa:** Data logging (for auditing and testing)

Hive, Pig, Jaql - Common Properties

- Pig, Hive, and Jaql have much in common.
  - They all translate high-level languages into MapReduce jobs
  - These languages let you write code much smaller than Java.
  - Interoperability can work both ways
  - All these technologies run on top of Hadoop so, they have the same limitations with respect to random reads and writes and low-latency queries as Hadoop does

Hive - HiveQL

- Hive provides data warehousing tools to ETL data, and query this data stored in Hadoop files.
  - Turns Hadoop into a data warehouse with a dialect of SQL for querying.

- HiveQL is a declarative language developed at FB.
  - Unlike PigLatin, you do not specify the data flow, but instead describe the result you want and Hive figures out how to build a data flow to achieve it.
  - Also unlike Pig, a schema is required
  - HiveQL is a relationally complete language but not a Turing complete language. It can be extended through UDFs to be Turing complete.

Hive

- Hive is a data warehouse developed on top of Hadoop.
  - Hive Query Lang. (HQL) is a declerative language like SQL
  - Thrift Server: Port that waits for queries
  - Parser: HiveQL, Ysmart Parser and Query plan

```
SELECT sum(l_extendedprice) / 7.0 AS avg_yearly
FROM (SELECT l_partkey, 0.2*avg(l_quantity) AS t1
        FROM lineitem
        GROUP BY l_partkey) AS inner,
        (SELECT l_partkey, l_quantity, l_extendedprice
        FROM lineitem, part
        WHERE p_partkey = l_partkey) AS outer
WHERE outer.l_partkey = inner.l_partkey;
AND outer.l_quantity < inner.t1;
```

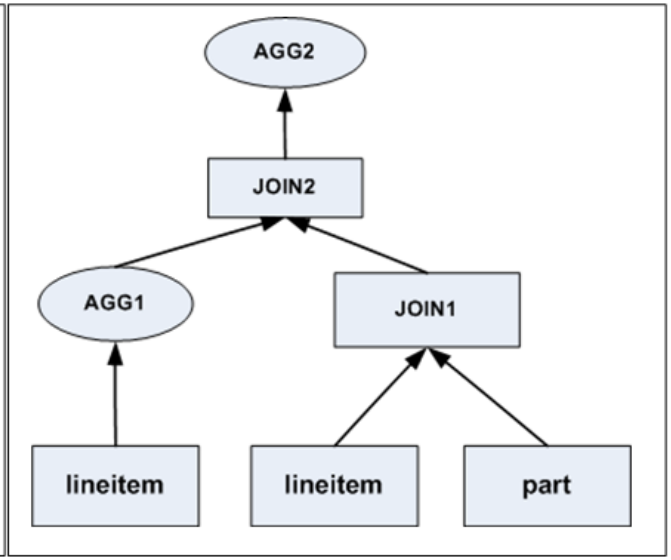


Figure1: HiveQL ile yazılmış TPC-H Q17 no'lu analitik benchmark sorgusu ve sağda derlenmiş sorgu planı.

HiveQL

- Automatically translated to a MapReduce job

```
Job1: generate both inner and outer,
      and then join them
Map:
 lineitem -> (k: l_partkey,
              v:(l_quantity,l_extendedprice))
 part -> (k:p_partkey,v:null)
Reduce:
 get inner: aggregate l_quantity for each (l_partkey)
 get outer: join with (l_partkey=p_partkey)
 join inner and outer
```

Pig Latin

- Pig is a high level (data flow) language for expressing data analysis.
  - You program by connecting things together.
  - Complex data structures and levels of nesting are supported
  - It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks.
  - Pig does not require that the data have a schema,
    - So it is well suited to processing unstructured data.
    - Pig can use a schema if you supply one.
  - PigLatin is relationally complete
    - Which means it is at least as powerful as relational algebra.
  - PigLatin is not Turing complete (*),.
    - (*) Turing completeness requires looping constructs, an infinite memory model, and conditional constructs.

Jaql

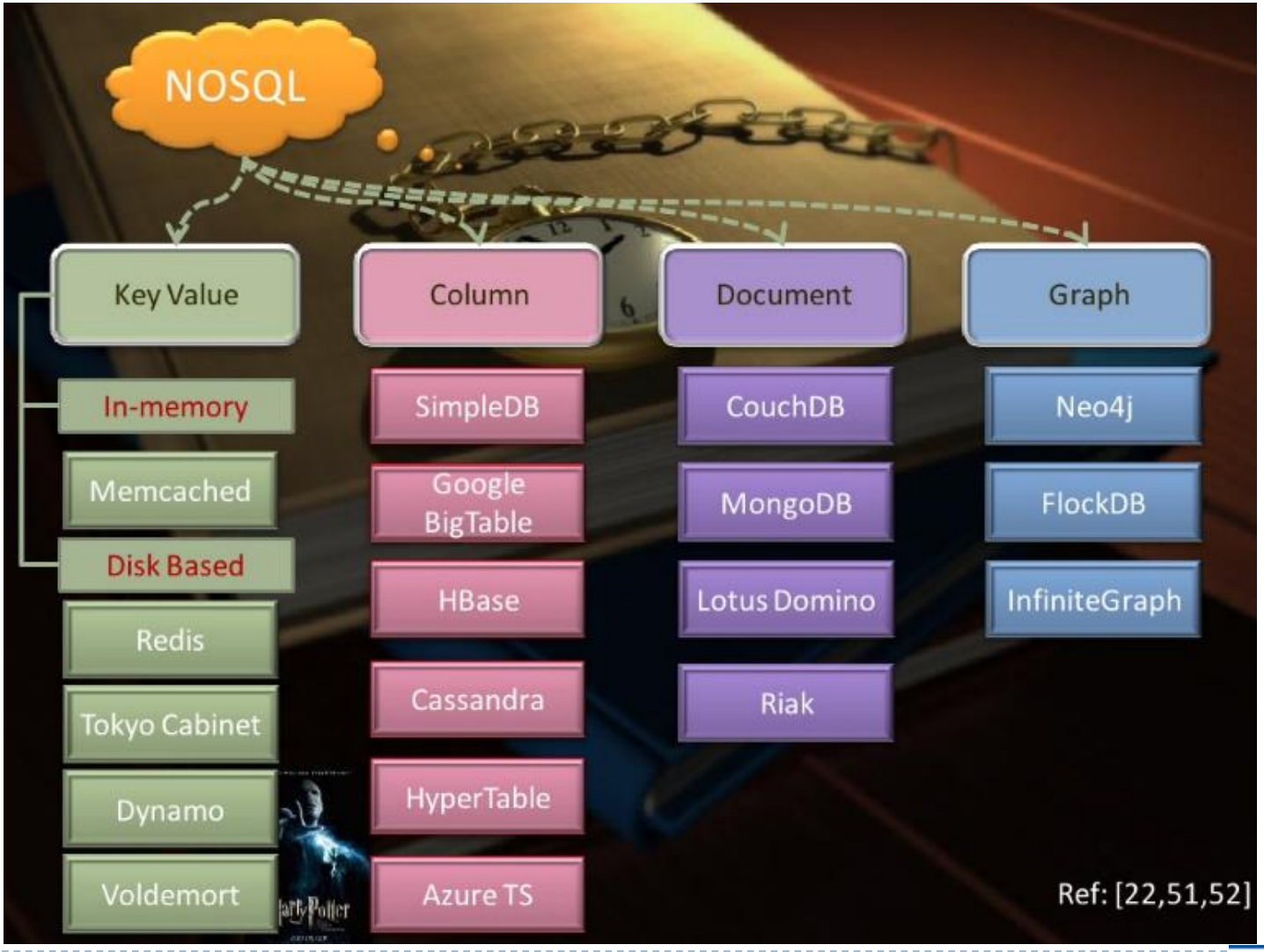
- Developed by IBM

- Jaql, or jackal, is a (data flow) query language
  - Native format JSON (JavaScript Object Notation)
  - Schemas are optional
  - Turing complete on its own without the need for extension through UDFs.

NoSQL

- 1970-2000 RDBMS (Oracle, DB2, MySQL, SQL Server)
- 2000-2005 Web Scale: Google, FB, Twitter (Amazon Dynamo, IBM) → Lustre,
- 2005-2010 Cassandra, Hbase, Mongo, CoubDB, riak, redis, (hazelcast)
- 2010 + Adaption of Cloud (DBaaS, NoSQL as a Service)

NoSQL (Not-Only SQL) Categories



NoSQL Varieties

- Key-value: Riak, Redis, Memcached, Dynamo, Voldemort, Hamster,
- Document-oriented: Mongodb, couchdb, orientdb, ravendb
- Columnar DB: Cassandra, Hbase, Amazon Simpledb
- Graph DB: Neo4j, Twitter flockdb, Orientdb, Infinitegraph

- Common properties:
  - open-source and commercial,
  - flexible schema (people like JSON, HTTP/REST API)
  - use case specific performance,
  - horizontal-scale wth global key – sharding,

http://bigdatauniversity.com → NoSQL and DBaaS Course

NoSQL Deep Dive- Key-Val. and Doc. Stores

- Key-value:
  - Shard well, high performance CRUD, value is a blob.
  - Good for: Session store and lookup, shopping carts,
  - Not good for: Interconnected data and queries, query based on value (and not key)

- Document (JSON, XML, BSON)
  - Value is visible and examinable,
  - Good: Event logging, blogs, metadata for web/mobile apps
  - Not good: if data is relational and concistency is required

http://bigdatauniversity.com → NoSQL and DBaaS Course

NoSQL Deep Dive – Columnar and Graph DB

- Columnar DB: Big table clones
  - Focus on storing and querying columns of data,
  - Good for: Sparse Rows with different columns
  - Not good: TPC-C and ACID transactions.

- Graph DB: Data consists of nodes and relationships
  - Good: if data resembles graph, social networks, routing, spatial, recommendation engines
  - Not good: May not scale well horizontally if frequent updates of all subnodes are required

http://bigdatauniversity.com → NoSQL and DBaaS Course

DB Choice – RDBMS or NoSQL, Pyhsical or Cloud

- Queries: TPC-C (lookup) or TPC-H (aggregation) style?
- Scalability needs: How big will your data get?
- Need consistency?
- Schema flexibility needed?
- Your skill-set, existing technology (legacy support,integration)
- Availability, replication, fault-tolerance requirements and ease of doing these.
- Do you have to Host/Manage all yourself? No →Cloud.
- Where is you data created? Locally-globally? Global→Cloud

IBM Cloudant

- NoSQL Document DB in Cloud
  - JSON objects - HTTP API
  - Supports Web and Mobile apps easily and scales.
  - Online –offline (synch later) clients and apps supported,
- MapReduce, Full text search, Geospatial supports are embedded.
  - Primary (B-tree), Secondary (map-reduce), Lucene FTI, R-tree,
- Easy-to-use dashboard
- Cloudant Managed or Local service models are available.
- Always online 24 x 7. Disaster recovery with Replicas.

NoSQL-DB Terminology Comparion

	DBMS	Mongo	Cassandra	Hbase
DB:	DB	DB	Keyspace	
Table:	Table	Collection	Column Family	Table
Row:	Row	Document	Row	Row
Store:	Btree	Btree	HexByteArray	Multi-map
Upsert:	No	Yes	Yes	
Distrbution :	?	DBs	All	

Considering HBase as a kind of large relational database is clearly a misleading option. HBase is essentially a key-value store.

http://internetmemory.org/fr/index.php/Synapse/understanding_the_hbase_data_model

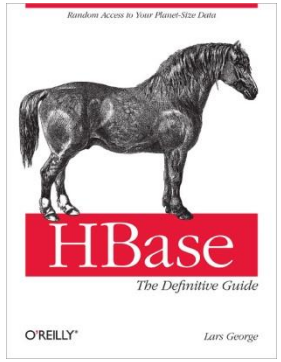
ÖZYEĞİN UNIVERSITY

Apache HBASE

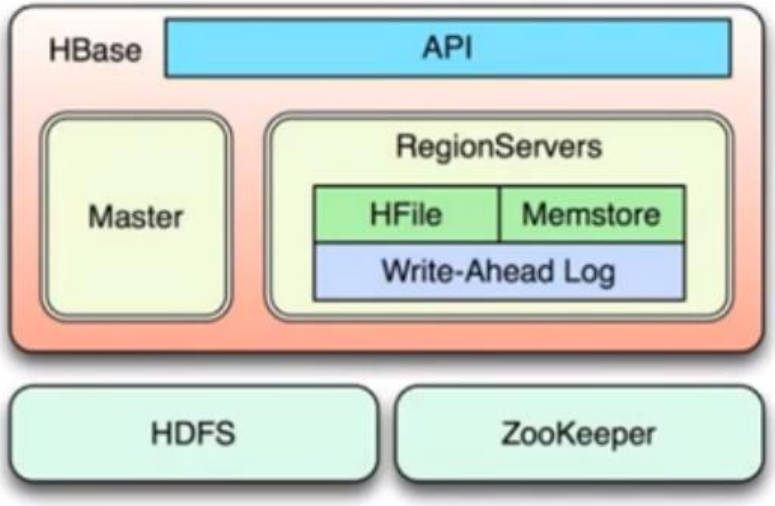
- HBase is an open-source, distributed, versioned, column-oriented store modeled after Google's Bigtable: A Distributed Storage System for Structured Data
- HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.
- Use Apache HBase software when you need random, real-time read/write access to your Big Data.
- This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware.

Hbase Architecture

- **Hbase** uses **HDFS** as its storage layer
  - HDFS gives reliability (replication) and failover
  - Standalone and Distributed (Pseudo & Fully)
- **Master** manages the cluster
  - **RegionServers** manage data
- **Zookeeper**
  - Coordinates the cluster
- **HDFS API** – Native Java



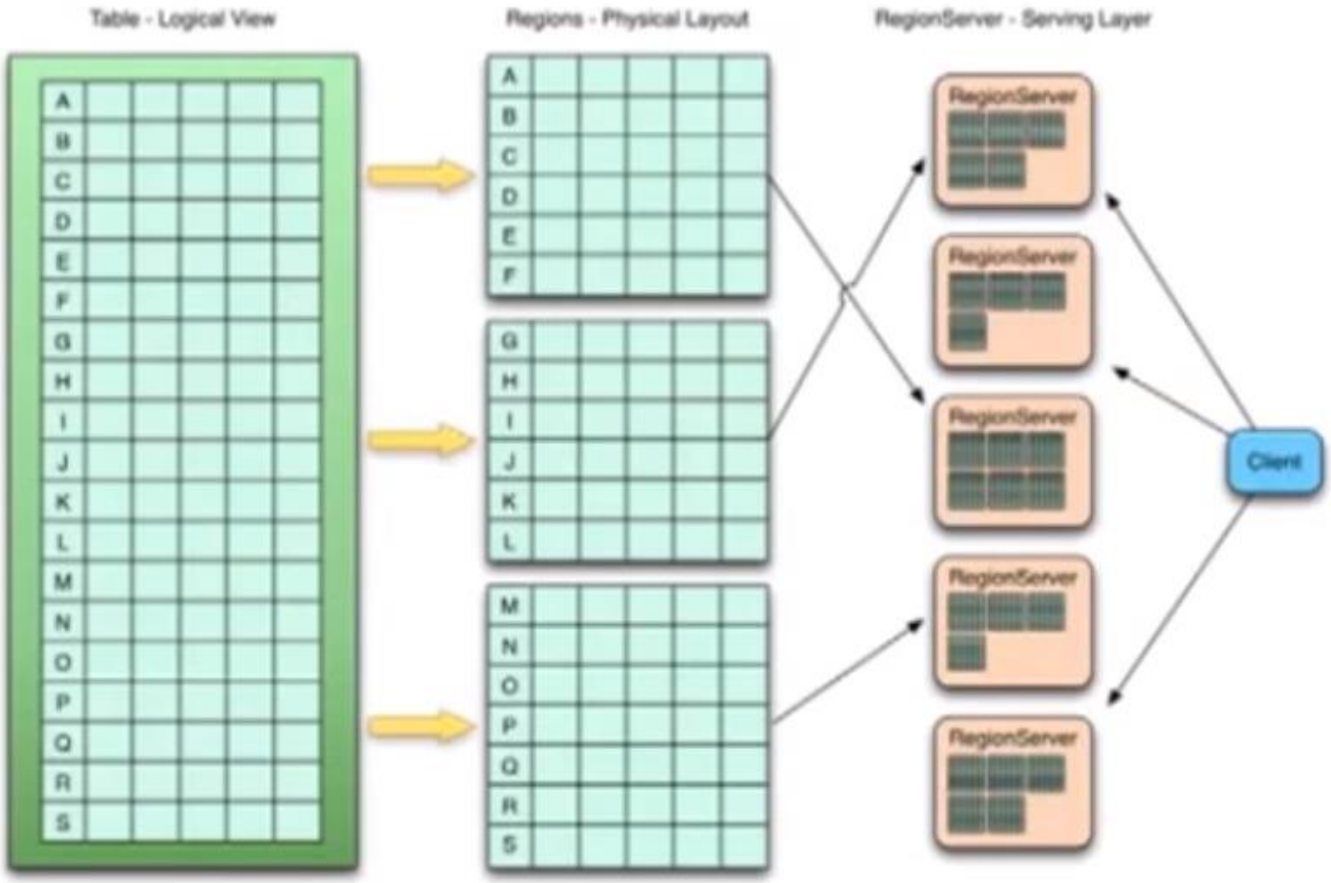
HBase Architecture



ÖZYEĞİN UNIVERSITY

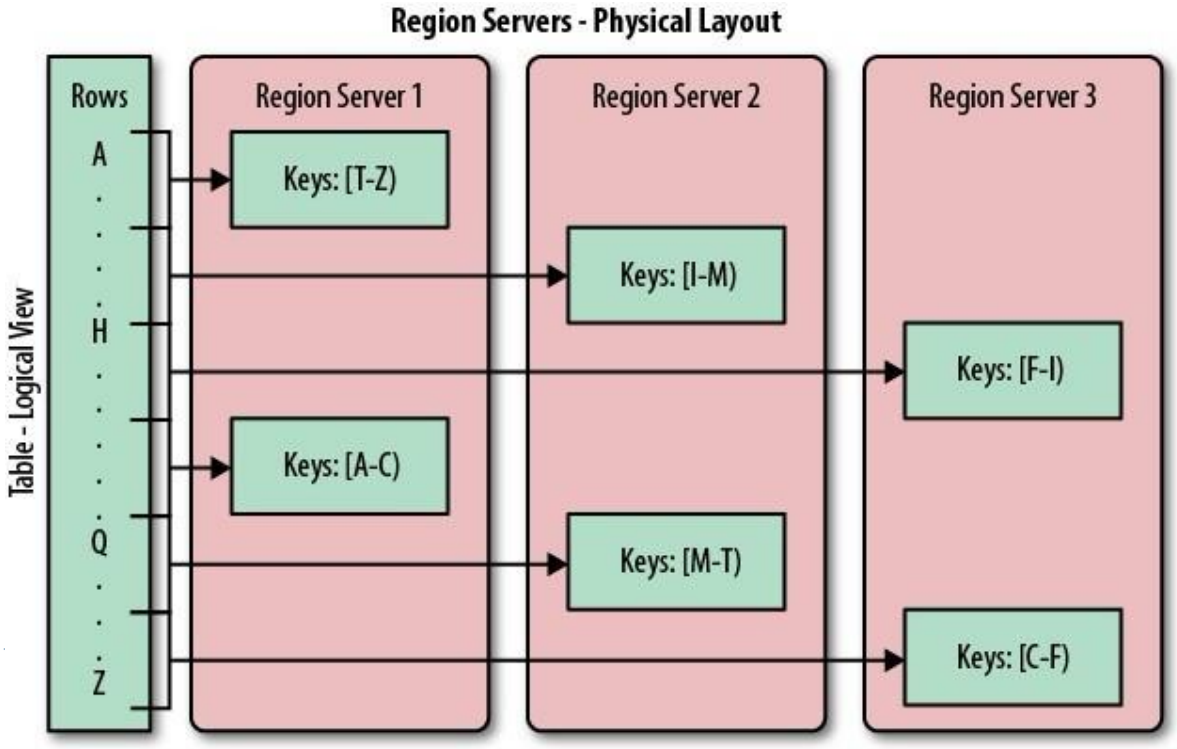
Hbase – Tables and Auto Sharding

- **(Logical) Tables in Hbase – Rows Sorted by keys**
  - **Regions**: Physical, sorted, contiguous splits: A-F, G-L, M-S
  - **RegionsServers** serve **Regions**



Hbase Regions – Physical Layout

- Sorted Rows (keys) are randomly distributed over RegionServers via **AutoSharding or manually.**
- Data is also divided into **column families** and stored as multiple **HFiles** (either compressed or uncompressed)
- **Bloom Filters** are stored at the end of each Hfile



ÖZYEĞİN UNIVERSITY

Hbase Installation and Setup

- Download and unzip package
  - tar xvzf hbase-0.94.1.tar.gz
- Start service:
  - bin/start-hbase.sh (start as ROOT).
  - http://localhost:60010
- Go to Hbase shell: bin/hbase shell
  - help, list
  - create 'users', 'info' (create table 'users' -a container- with single column family 'info')
  - describe 'users'
  - drop 'users'

Put – Get from Command Line

- Put a row:
  - put 'users', 'u1' (prim key), 'info:name', 'Rogers'
- Get a row:
  - get 'users', 'u1'
- Put (add) more:
  - put 'users', 'u1', 'info:email', 'mail@rogers.com'
- Scan table:
  - scan 'users' // (select * from users)

ÖZYEĞİN UNIVERSITY

Apache Cassandra

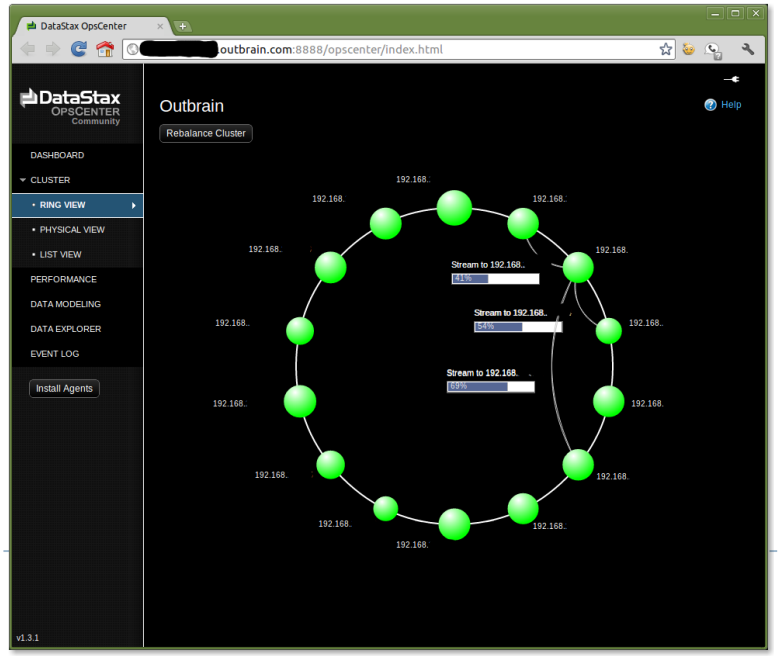
- Cassandra Java ile yazılmış bir NoSQL veritabanıdır.
- Cassandra, HBase ile beraber Facebook içinde kullanılmaya devam edilmektedir.
- One of the largest Cassandra clusters:
  - 100 TB with 150 Nodes
- Cassandra şemaları (schema) zorunlu tutmaz.
- Cassandra verileri HexByte array olarak saklar.
  - Bir sütun için sorgu yapmak istiyorsak o sütunun ikincil keyleri olmak zorundadır.
  - Cassandra çalışma zamanında verilerin bir kısmını **memtable** içinde saklar,
  - Kapatıldığında ya da heap alanına sığmayacak veri boyutlarına ulaşıldığında **sstable** içinde saklar.

Apache Cassandra

- Linear scalability and proven fault-tolerance on commodity hardware.
- Best-in-class replication support across multiple datacenters.
- Cassandra is in use at Netflix, Twitter, Urban Airship, Constant Contact, Reddit, Cisco, OpenX, Digg, and more companies that have large, active data sets.
- Cassandra provides full Hadoop integration, including with Pig and Hive.

Cassandra Cluster Setup and Configuration

- Download Cassandra from Apache Cassandra site
  - http://cassandra.apache.org/download/
  - Unzip and Browse directories: **/bin, /conf, /lib**
  - Configure settings in **/conf/cassandra.yaml** file.
  - First node, initial token = 0, Seeds (new cluster node IP & Ports?)
  - Seed servers can be SPoF



ÖZYEĞİN UNIVERSITY

Cassandra Services and Data Loading

- **bin/cassandra:** Cassandra executable server
- **bin/nodetool:** For cluster and node settings
  - Check status, Update tokens, Add/remove nodes
- **bin/cassandra-cli :** Cassandra command line interfacce,
  - Fast schema creation, etc.
- **bin/sstableloader:**
  - Mevcut sstable'ları cassandra içine aktarmak için kullanılır.
  - **sstableloader** program to transfer between DB-Cassandra

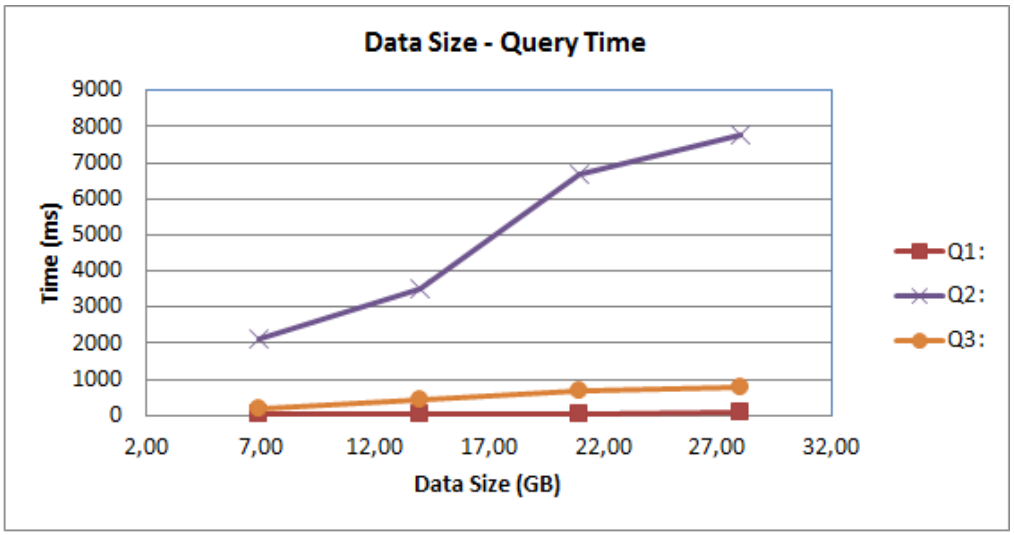
- Data Loading using Client tools
  - hector, astyanax, jdbc, etc.

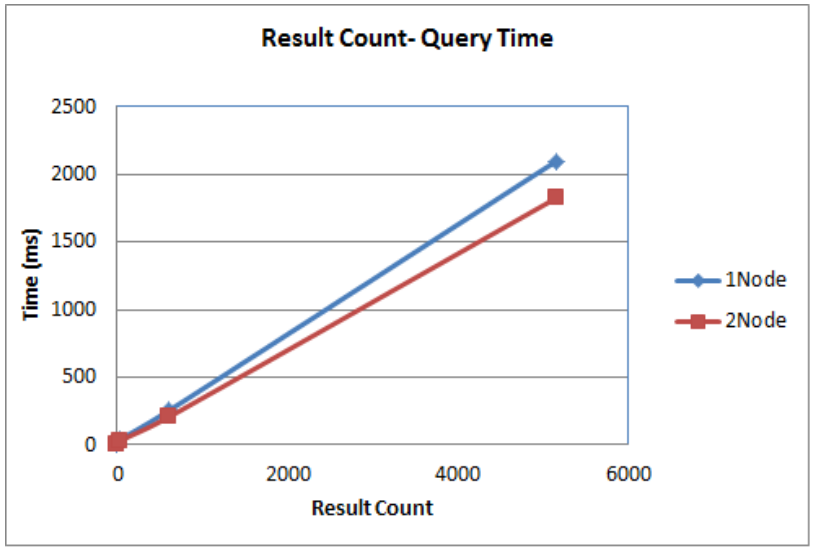
ÖZYEĞİN UNIVERSITY

Cassandra – Parametes Affecting Performance

- Insert Time, Select Time (Q1, Q2, Q3) – CR/UD
  - Dataset size (WAP: 7GB, 14GB, 21GB, 28GB)
  - Number of nodes (1,2,4,8)
  - Number of threads (1,2,4,8,16,32)
  - Number of columns stored (1→32, cardinality)
  - Number of columns stored & indexed (1→32 cols)
  - Number of results output by query (selectivity)

Cassandra Query Results – With Avea WAP

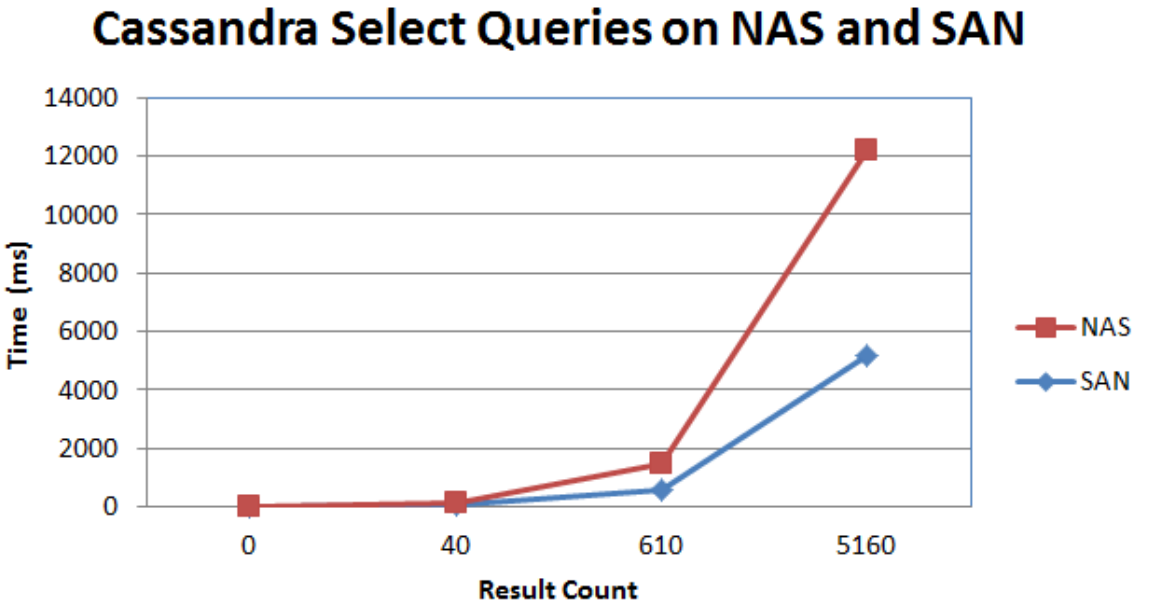




	3.5GB (run1)	3.5GB (run2)	7.09GB (run1)	7.09GB (run2)	10.68GB (run1)	10.68GB (run2)	14.25GB (run1)	14.25GB (run2)
Q1: select E3,E5 from wap where E3='905077122135'	109	32	78	39	636	54	231	69
Q2: select E3,E5 from wap where E3='905306971704'	3133	2102	3534	3484	42882	6666	11814	7768
Q3: select E3,E5 from wap where E3='905377366707'	291	207	464	421	5306	660	1312	792

Cassandra - NAS vs. SAN

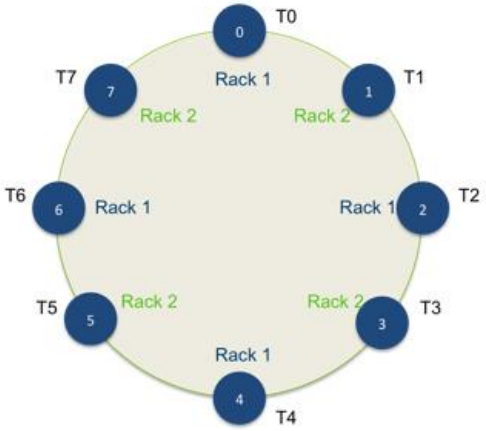
- Cost comparison
  - IBM SAN - $5000/TB
  - WD 2TB NAS - $100/TB



ÖZYEĞİN UNIVERSITY

Cassandra – Adding a new Node

- Why? For data and load distribution & balancing
- **cassandra.yaml:**
  - Do the <IP , port > settings
  - Change **initial_token** value to non-zero (0)
- Tokens: 0 – 2^127 (keyspace?)
  - Calculating initial_token for N nodes,
  - *Node i* gets tokens: i x (2^127/N) → (i+1) x (2^127/N)
  - e.g. For 2 nodes:
    - Node0 initial_token: 0
    - Node1 initial_token: 85070591730234615865843651857942052864



ÖZYEĞİN UNIVERSITY

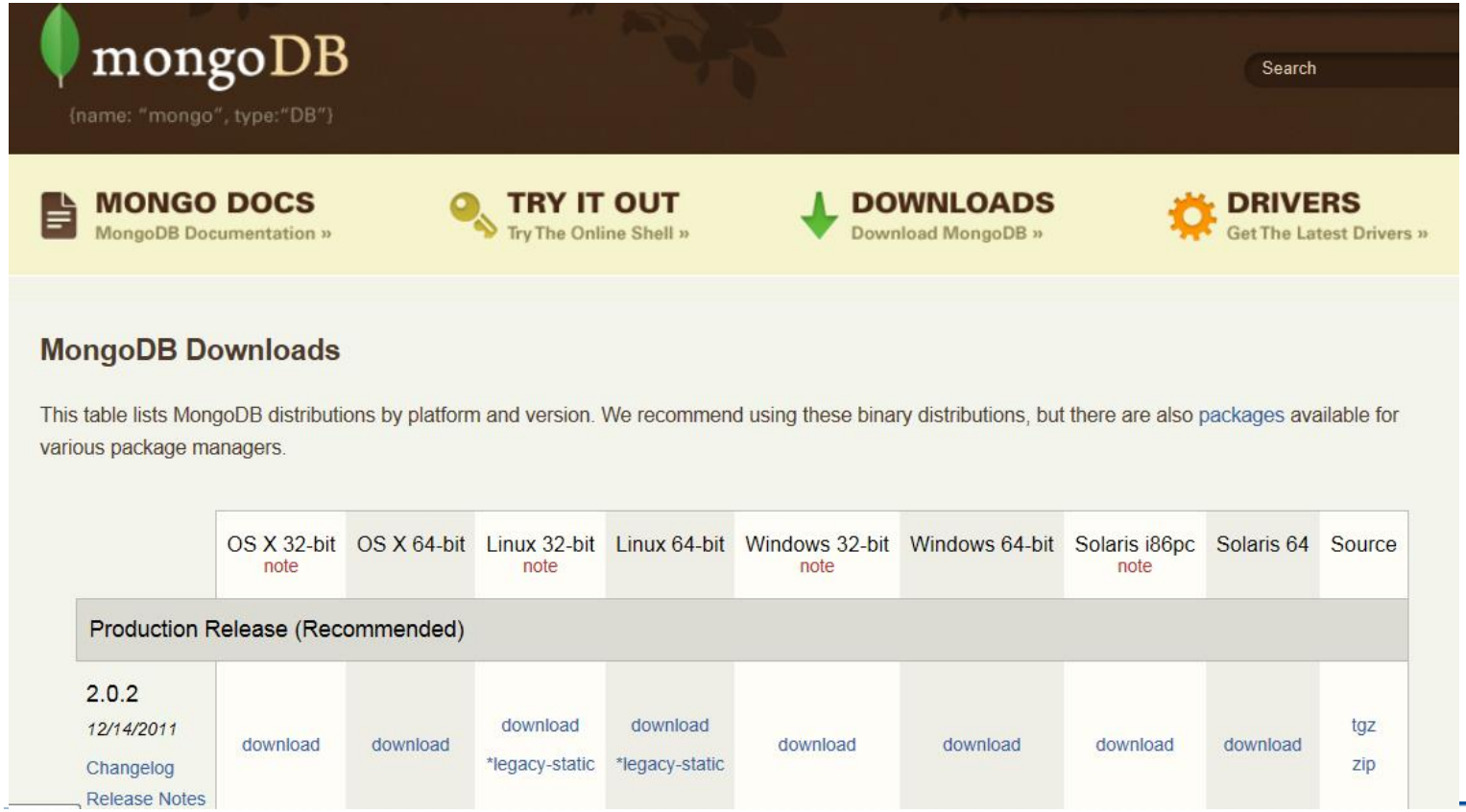
MongoDB Tutorial

- MongoDB is a scalable, high-performance, open source NoSQL database written in C++.
- MongoDB: Hu***mongo***us: Large
  - Document-oriented storage
- JSON: JavaScript Object Notation
- What does document-oriented mean?
  - JSON objects; Not relational; Not OODB.
  - Rows: JSON documents
  - Tables: Collections
  - Join → Embedding and Linking

http://www.mongodb.org/

MongoDB

- **Download & Unzip:** http://www.mongodb.org/downloads
- Available for 4 different OS & 8 architectures.



ÖZYEĞİN UNIVERSITY

Mongo Installation and Start

- MondoDB consists of only 10 executable files!
- **Create a data directory** for MongoDB:
  - C:\mymongodb
- **Run mongod.exe** and point to the data directory:
  - \bin\mongod --dbpath "c:\mymongodb "

```
C:\Users\ismaila\Desktop\HAVOOZ\MONGODB\mongodb-win32-x86_64-2.0.2\bin>mongod --dbpath "c:\mymongodb"
Sun Feb 26 18:39:30 [initandlisten] MongoDB starting : pid=4576 port=27017 dbpath=c:\mymongodb 64-bit host=OZUNA
Sun Feb 26 18:39:30 [initandlisten] db version v2.0.2, pdfile version 4.5
Sun Feb 26 18:39:30 [initandlisten] git version: 514b122d308928517f5841888ceaa4246a7f18e3
Sun Feb 26 18:39:30 [initandlisten] build info: windows (6, 1, 7601, 2, 'Service Pack 1') BOOST_LIB_VERSION=1_42
Sun Feb 26 18:39:30 [initandlisten] options: { dbpath: "c:\mymongodb" }
Sun Feb 26 18:39:30 [initandlisten] journal dir=c:/mymongodb/journal
Sun Feb 26 18:39:30 [initandlisten] recover : no journal files present, no recovery needed
Sun Feb 26 18:39:30 [initandlisten] waiting for connections on port 27017
Sun Feb 26 18:39:30 [websvr] admin web console waiting for connections on port 28017
```

  - Run the client: \bin\mongo

```
C:\Users\ismaila\Desktop\HAVOOZ\MONGODB\mongodb-win32-x86_64-2.0.2\bin>ls
bsondump.exe      mongod.exe        mongoexport.exe   mongoimport.exe   mongos.exe        mongotop.exe
mongo.exe         mongodump.exe     mongofiles.exe    mongorestore.exe  mongostat.exe

C:\Users\ismaila\Desktop\HAVOOZ\MONGODB\mongodb-win32-x86_64-2.0.2\bin>mongo
MongoDB shell version: 2.0.2
connecting to: test
>
```

MongoDB Interactive Shell

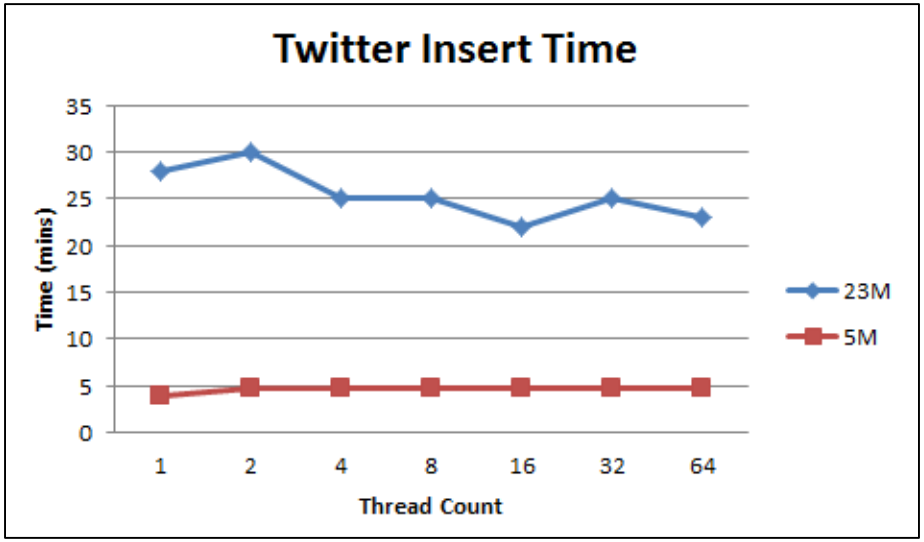
- mongo uses Javascript API to interact with the DB
- Variables: e.g. **'db'** current database connection
- db.**foo**.save({ name : "sara"}); // Implicitly creates collection **foo**
- **show collections → foo**

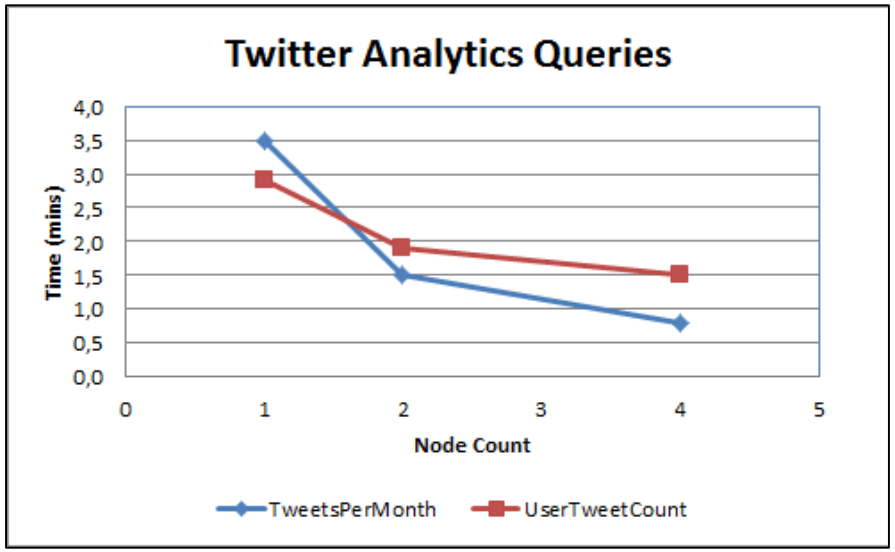
```
C:\Users\ismaila\Desktop\HAVOOZ\MONGODB\mongodb-win32-x86_64-2.0.2\bin>mongo
MongoDB shell version: 2.0.2
connecting to: test
> help
        db.help()                    help on db methods
        db.mycoll.help()             help on collection methods
        rs.help()                    help on replica set methods
        help admin                   administrative help
        help connect                 connecting to a db help
        help keys                    key shortcuts
        help misc                    misc things to know
        help mr                      mapreduce

        show dbs                     show database names
        show collections             show collections in current database
        show users                   show users in current database
        show profile                 show most recent system.profile entries with time
        show logs                    show the accessible logger names
        show log [name]              prints out the last segment of log in memory, 'glo
        use <db_name>                set current database
        db.foo.find()                list objects in collection foo
        db.foo.find( { a : 1 } )     list objects in foo where a == 1
        it                           result of the last line evaluated; use to further
        DBQuery.shellBatchSize = x   set default number of items to display on shell
        exit                         quit the mongo shell
>
```

MongoDB Results – Turkish Twitter Data

- Twitter Data: 5→23 million rows, 5.4GB data
  - Multi-threaded Insert Time
  - Analytics queries (*e.g. Tweets/Month*) written with MapReduce

```
String map = "function(){emit(this.createDate.getMonth(), 1);};";
String reduce = "function(key,values){var result = 0;for(v in values){result += values[v];}return result;};";
```

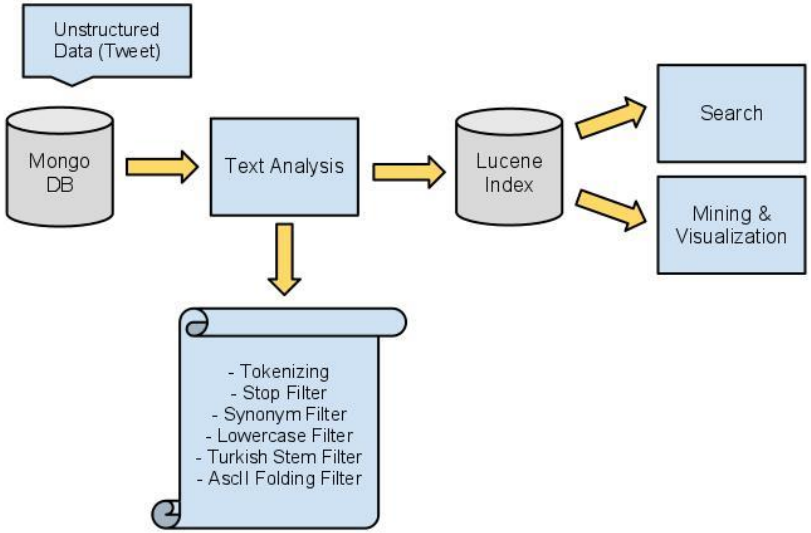


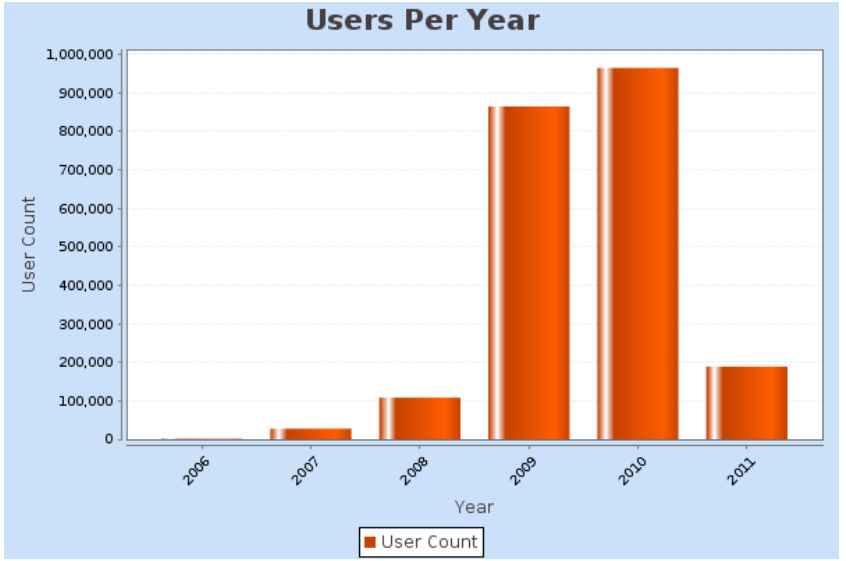


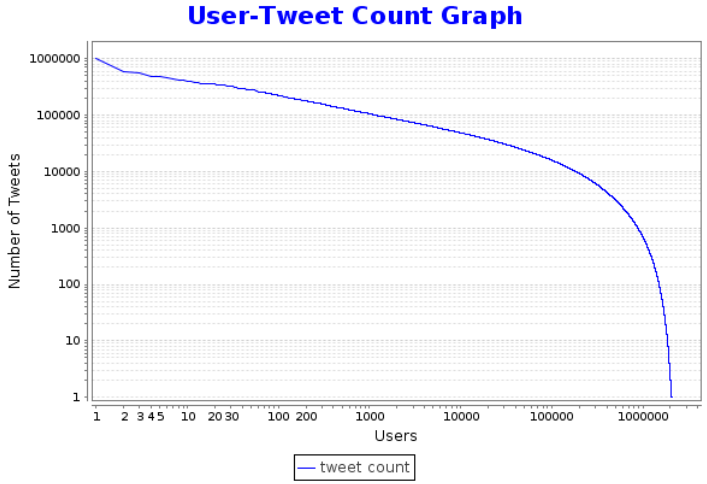
ÖZYEĞİN UNIVERSITY

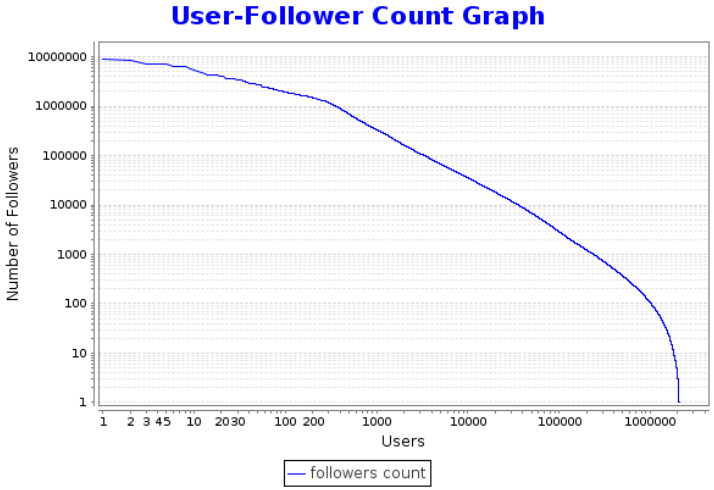
MondoDB- Twitter Statistics Obtained

- Architecture and Results









ÖZYEĞİN UNIVERSITY

MongoDB

- Twitter Data Model

```
> db.twitterStatus.findOne()
{
"_id" : ObjectId("4ec8f0b744ae25ad9c3e4851"),
"statusId" : NumberLong("25907631489"),
"userId" : NumberLong(168434390),
"statusText" : "bir güne on ders saati sıdırmak zorunda bırakanlar utansın..",
"createDate" : ISODate("2010-09-28T21:00:00Z"),
"source" : "<a href=\"http://mobile.twitter.com\" rel=\"nofollow\">mobile web</a>",
"retweetCount" : NumberLong(0),
"inReplyToStatusId" : NumberLong(-1),
"inReplyToUserId" : NumberLong(-1),
"isFavorited" : false,
"isRetweeted" : false,
"isTruncated" : false,
"latitude" : 0,
"longitude" : 0
}
```


ÖZYEĞİN UNIVERSITY

MongoDB Data Update

- Example
  - person = db.people.findOne({ name : "sara" });
  - person.city = "New York";
  - db.people.save(person);

- Java Integration
  - Create a connection to mongod server and send JSON query directly.

ÖZYEĞİN UNIVERSITY

SQL to MongoDB

CREATE TABLE USERS (a Number, b Number)

db.createCollection("users")

INSERT INTO USERS VALUES(3,5)

db.users.insert({a:3,b:5})

SELECT * FROM users

db.users.find()

SQL to MongoDB

SELECT a,b FROM users WHERE age=33

db.users.find({age:33}, {a:1,b:1})

SELECT * FROM users WHERE age=33 ORDER BY name

db.users.find({age:33}).sort({name:1})

CREATE INDEX myindexname ON users(name)

db.users.ensureIndex({name:1})

SQL to MongoDB

UPDATE users SET ordernum=1 WHERE age=45

db.users.update({age:45}, {$set:{ordernum:1}}, false, true)

About: .update(exp, exp, upsert, multi)

update() only modifies the first matched object. If you want to modify all matched objects, you need to use the multi flag.

Upsert means "update if present; insert (a single document) if missing

SQL to MongoDB

Aggregation

SELECT COUNT(*) FROM users

```
db.users.aggregate([
 { $group: {_id:null, count:{$sum: 1}} }
])
```

SELECT SUM(price) FROM orders

```
db.users.aggregate([
 { $group: {_id:null,
            total:{$sum:"$price"} } }
])
```

Other Related Apache Projects

- Apache Mahout
  - Scalable machine learning library, has clustering, CF, etc.
  - Works with Apache Hadoop
- Apache Nutch is an OS web-search software project.
  - Stemming from Apache Lucene, it now builds on Apache Solr
  - Adding Crawler, Linker, Parser support by Apache Tika
- Apache Lucene
  - Lucene is a full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

Other Related Apache Projects

- Apache OpenNLP (Türkçe için muadili: Zemberek)
  - Supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution.
- Apache Solr
  - OS enterprise search server based on the Lucene, with XML/HTTP and JSON, Python, etc. APIs, hit highlighting, faceted search, caching, replication, and a web administration interface.
- Apache ORO
  - A set of text-processing Java classes that provide regular expressions, AWK-like regular expressions, glob expressions,

Apache Avro and Thrift

- Apache Avro
  - It is a data serialization system.

- Apache Thrift: http://thrift.apache.org/
  - Apache Thrift allows you to define data types and service interfaces in a simple definition file.
  - Taking that file as input, the compiler generates code to be used to easily build RPC clients and servers that communicate seamlessly across programming languages.
  - Instead of writing a load of boilerplate code to serialize and transport your objects and invoke remote methods, you can get right down to business.

Apache ZooKeeper

- Apache ZooKeeper is an effort to develop and maintain an open-source server which enables highly reliable distributed coordination.
  - It runs the expensive Paxos (distributed consensus) protocol

Apache Whirr

- A set of libraries for deploying & running cloud services
  - Deploying & runing Hadoop clusters
  - *To manage the clusters use:C hef, Puppet, Ganglia-Hyperic, etc.*

- Whirr provides:
  - 1. A cloud-neutral way to run services.
  - 2. A common service API. The details of provisioning are particular to the service.
  - 3. Smart defaults for services. You can get a properly configured system running quickly, while still being able to override settings as needed.

ÖZYEĞİN UNIVERSITY

Data Collection, Integration, Logging

- Distributed data collection and logging projects
  - http://incubator.apache.org/projects/

- Flume, Sqoop,
  - **Flume:** Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. http://flume.apache.org/

Sqoop

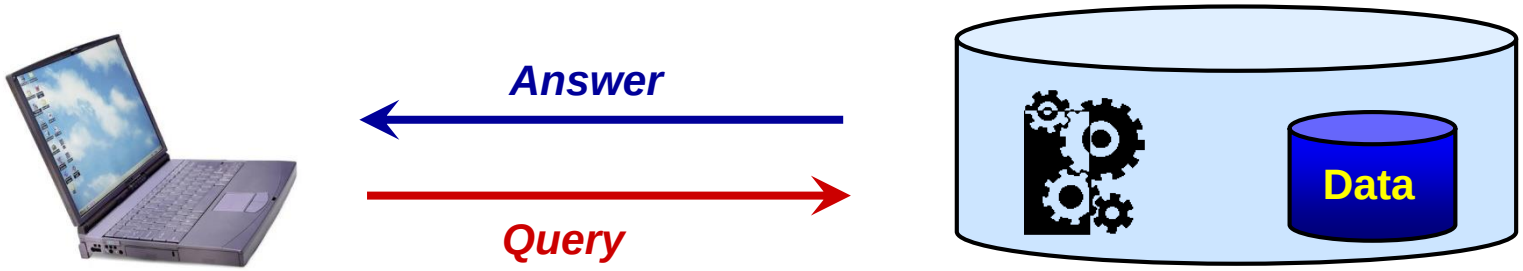
Data Collection, Integration, Logging

- **Chukwa**
  - Hadoop subproject devoted to large-scale log collection.
  - http://wiki.apache.org/hadoop/Chukwa

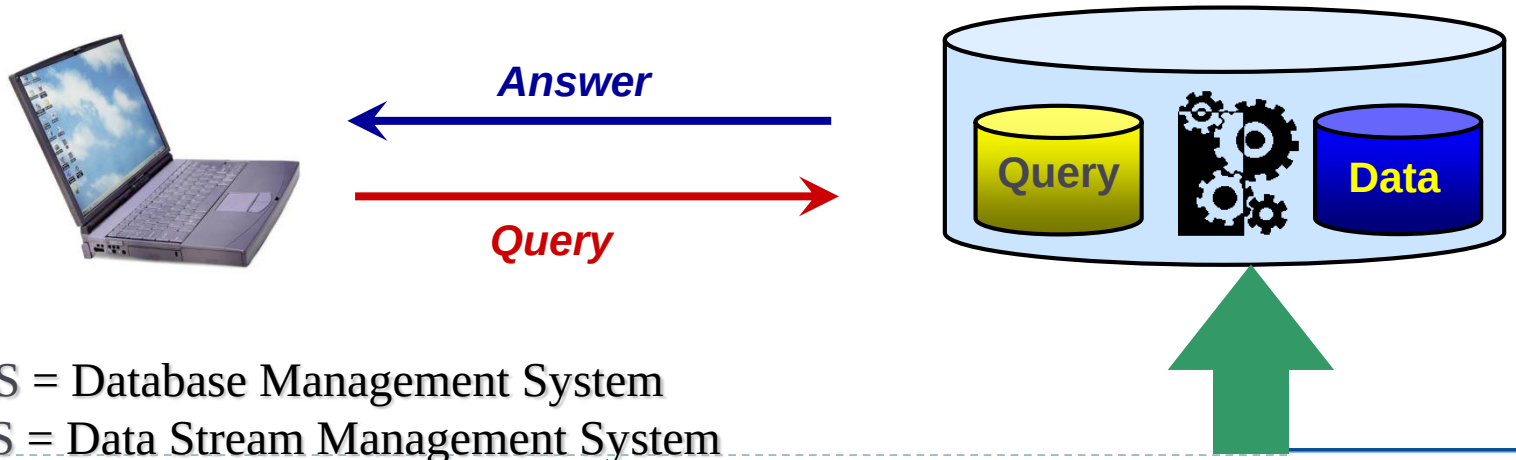
- **Fluentd:** http://fluentd.org/
  - "Fluentd" is a OSS lightweight and flexible log collector. Fluentd receives logs as JSON streams, buffers them, and sends them to other systems like S3, MongoDB, Hadoop, or other Fluentds.
- **Scribe**
  - Server for aggregating log data streamed in real-time from a large number of servers. Scribe was developed at FB using Apache Thrift and released in 2008.

Traditional vs. Continuous Query Processing

- Traditional Queries: Persistent relations, *One-time queries, Predictable*



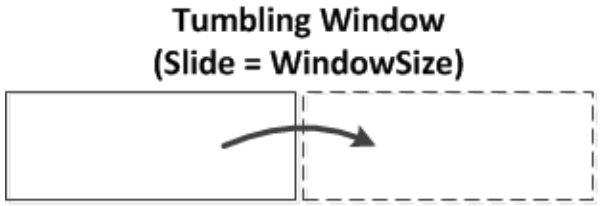
- Continuous Queries: Transient streams, Continuous queries, Unpredictable

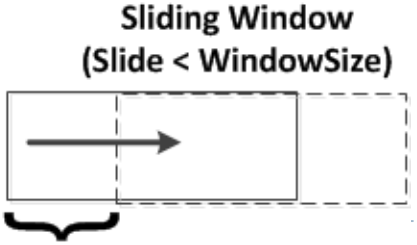


DBMS = Database Management System
DSMS = Data Stream Management System

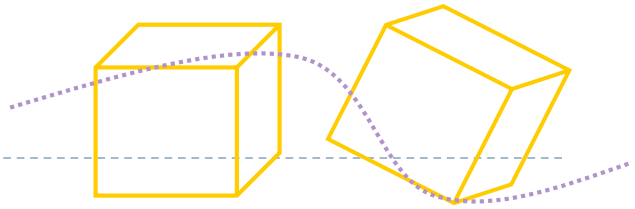
Complex Event Processing (CEP) Introduction

- An *event* is an atomic signal, message or tuple
- Complex Event Processing
  - Correlates at least two *events of different types*
  - Low-level events → Composite events →Complex events
- CEP is expected to generate *higher-level semantics*
  - Point to the root cause (or intent) of multiple events
  - The output event is expected to be more actionable
- CEP has a role of *summarization*
  - Using filtering, aggregation, etc. reducing high data volumes





Stream Warehousing



- Pattern analysis over multi-dimensional stream data
  - **GOTO: E-Cube Research**
- Applications
  - Emergency-evacuation planning, disease control, *etc.*
- Existing systems
  - Traditional OLAP works over persistent data (not stream)
  - Concept hierarchies + CEP pattern detection
- Our approach
  - MD analysis combining concept and pattern hierarchies

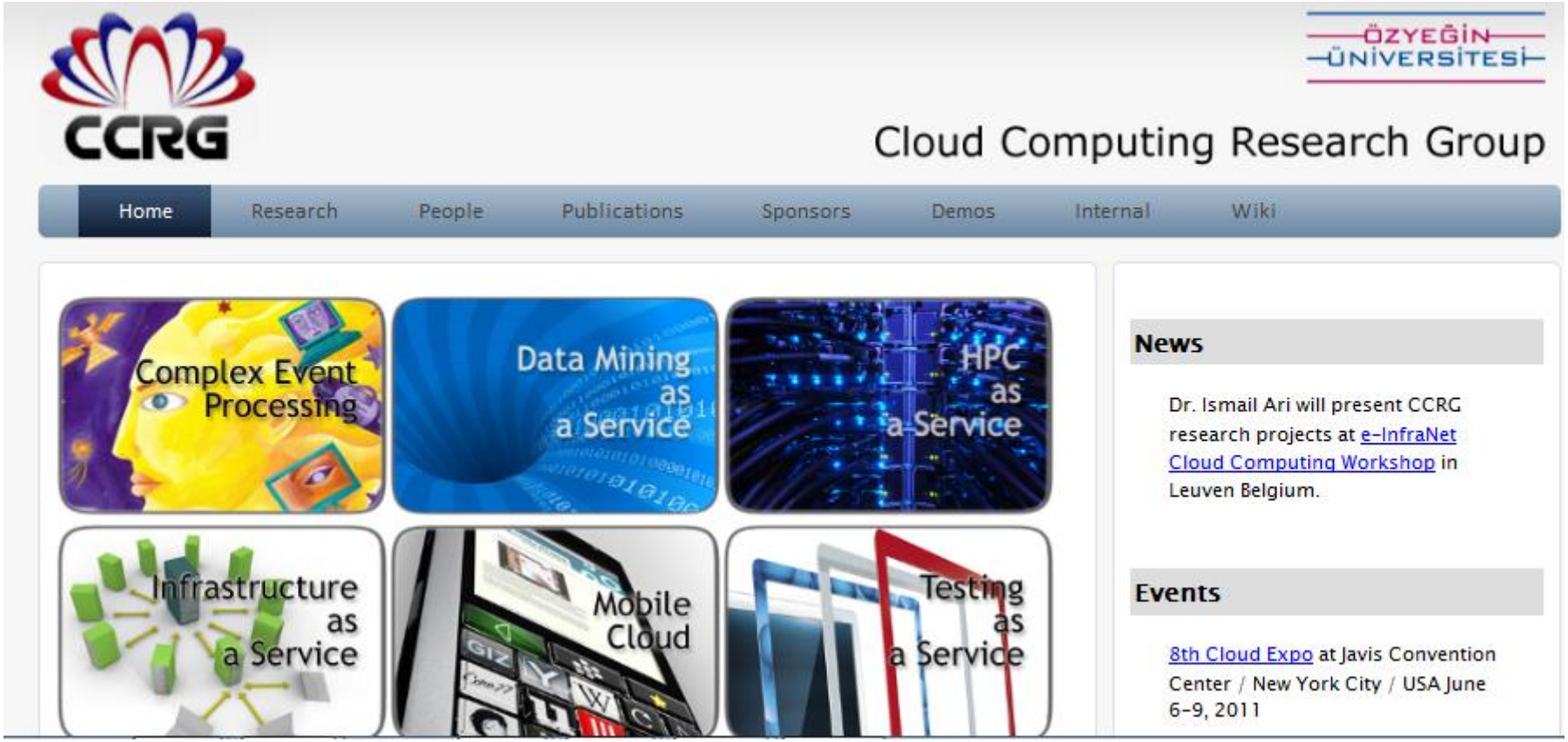
Stream Mining

- Continuous training of data mining models
  - Imagine moving clusters, branching decision trees, linear and logistics regressions changing slope
- Challenges
  - Limited memory, one-pass algorithms, processing time
- Our approach

```
QUERY 1:
SELECT Apriori(parameters, table.feature1, table.feature2)
FROM event.win:length(5) AS table
```

GOTO: Online ARM over Fast Data

Cloud Computing Research Group



http://cloud.ozyegin.edu.tr

ÖZYEĞİN UNIVERSITY