(Big) Data Processing

Philip J. Cwynar MSIS, MBA

Chirayu Wongchokprasitti PhD

University of Pittsburgh
School of Information Sciences

Outline

- Data Warehousing
- Hadoop/MapReduce
- Pig
- Hive
- Spark

Introduction

Processing

"If you aren't taking advantage of big data, then you don't have big data, you have just a pile of data,"

-- Jay Parikh, VP of infrastructure at Facebook

Data Warehouses

A producer wants to know ...

Which are our lowest/highest margin customers ?

What is the most effective distribution channel?

Who are my customers and what products are they buying?

What product promo-tions have the biggest impact on revenue?

Which customers are most likely to go to the competition?

What impact will new products/services have on revenue and margins?

There is plenty of data, and yet …

"I can't find the data I need"
- data is scattered over the network
- many versions, subtle differences

"I can't get the data I need"
- need an expert to get the data

"I can't understand the data I found"
- available data poorly documented

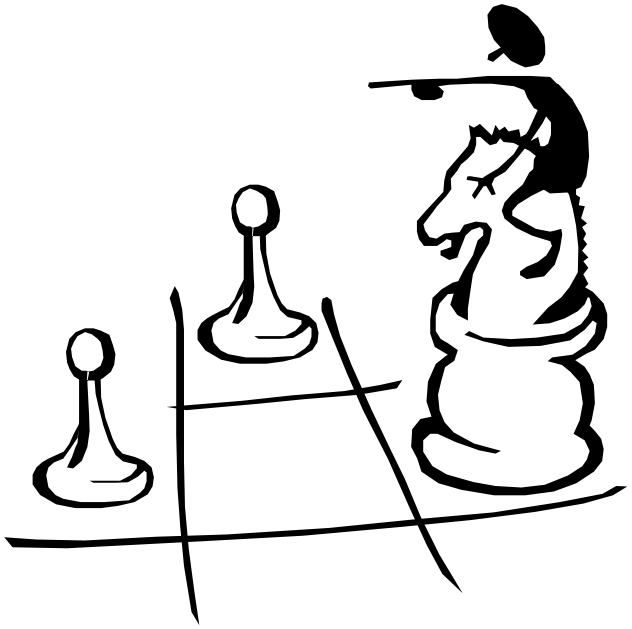
"I can't use the data I found"
- results are unexpected
- data needs to be transformed from one form to other

The need for business intelligence

- Maintain competitive edge
  - Market / customer knowledge
  - Fast, easy access to information
- Improve business efficiency
  - Reduce costs
  - Streamline processes

Data analysis and data warehousing

Data warehousing provides an enterprise with a memory



Data analysis provides the enterprise with intelligence

We want to know ...

- Given a database of 100,000 names, which persons are the least likely to default on their credit cards?
- Which types of transactions are likely to be fraudulent given the demographics and transactional history of a particular customer?
- If I raise the price of my product by $1, what is the effect on my ROI (Return on Investment)?
- If I offer only 2,500 airline miles as an incentive to purchase rather than 5,000, how many lost responses will result?
- If I emphasize ease-of-use of the product as opposed to its technical capabilities, what will be the net effect on my revenues?
- Which of my customers are likely to be the most loyal?

Definitions of a data warehouse

"A subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process"

[W.H. Inmon]

"A copy of transaction data, specifically structured for query and analysis"

[Ralph Kimball]

"A single, complete and consistent store of data obtained from a variety of different sources made available to end users in a what they can understand and use in a business context."

[Barry Devlin]

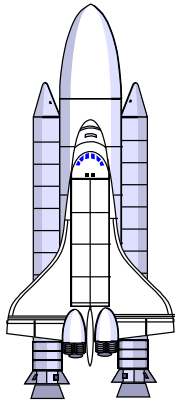
Data warehouse

- For organizational learning to take place, data from many sources must be gathered together and organized in a consistent and useful way – hence, “data warehousing”
- The data warehouse is a collection of data that is pulled together primarily from operational business systems and is structured and tuned for easy access and use by information consumers and analysts, especially for the purpose of decision making.
- The goal of data warehousing is to integrate enterprise wide corporate data into a single repository from which users can easily run queries.
- Data warehouse is an organization’s (enterprise’s) memory.
- ***DW provides a “Single version of the Truth”***

Explorers, farmers and tourists

Farmers: Harvest information from known access paths

Tourists: Browse information harvested by farmers



Explorers: Seek out the unknown and previously unsuspected rewards hiding in the detailed data

Data warehouse

Enterprise "Database"

Customers

Orders

Vendors

etc., …

etc., …

Transactions

Copied, organized summarized

Data Warehouse

Data Mining

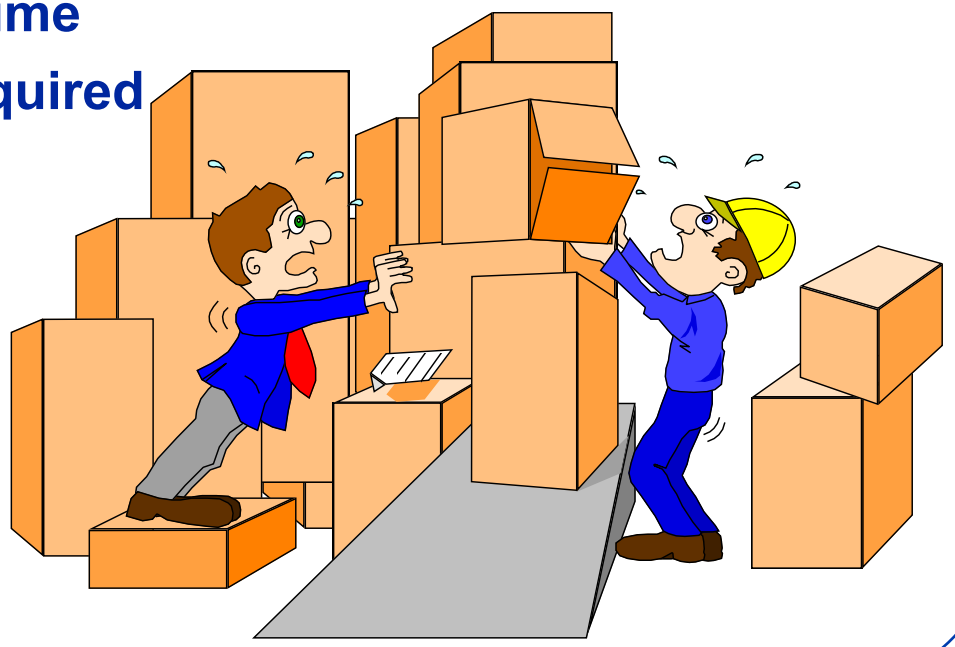
Data Miners:
- "Farmers" – they know
- "Explorers" – unpredictable

Data warehouse

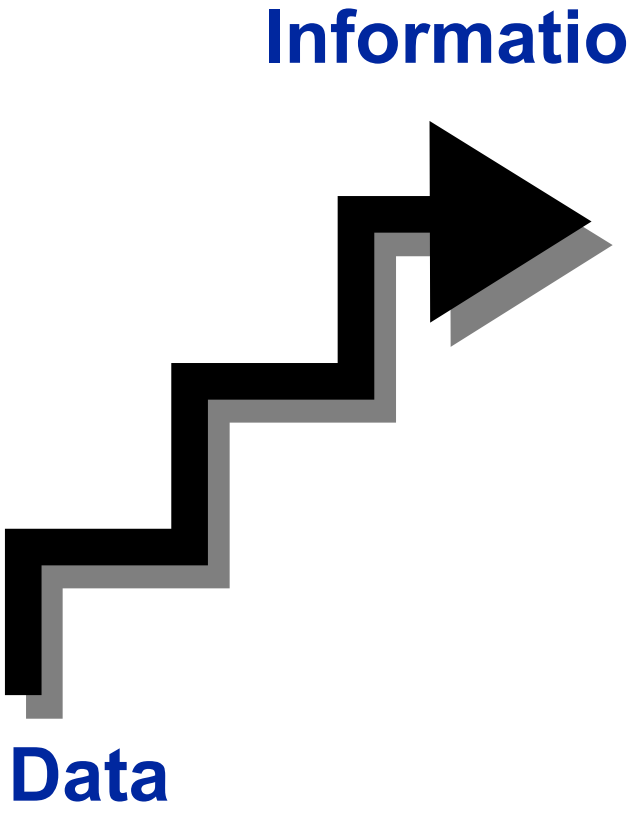
- A data warehouse is a copy of transaction data specifically structured for querying, analysis and reporting – hence, data mining.
- Note that the data warehouse contains a copy of the transactions which are not updated or changed later by the transaction system.
- Also note that this data is specially structured, and may have been transformed when it was copied into the data warehouse.

Expectations from a data warehouse

- Data should be integrated across the enterprise
- Summary data has a real value to the organization
- Historical data holds the key to understanding data over time
- What-if capabilities are required



What is data warehousing?



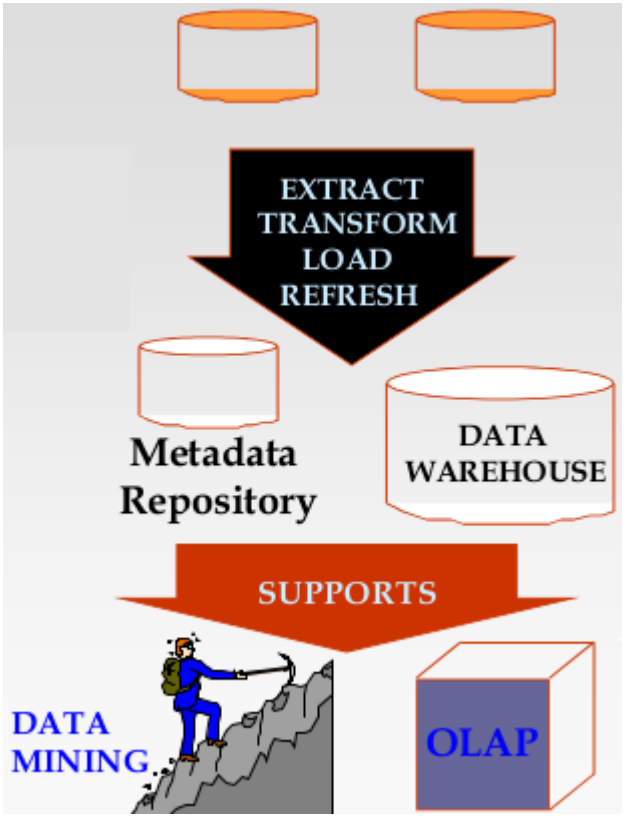
A process of transforming data into information and making it available to users in a timely enough manner to make a difference

[Forrester Research, April 1996]

Data Warehousing

- Intergraded data spanning over long time periods, often augmented with summary information
- Several gigabytes to terabytes common
- Interactive response times expected for complex queries; ad-hoc updates uncommon

external data sources



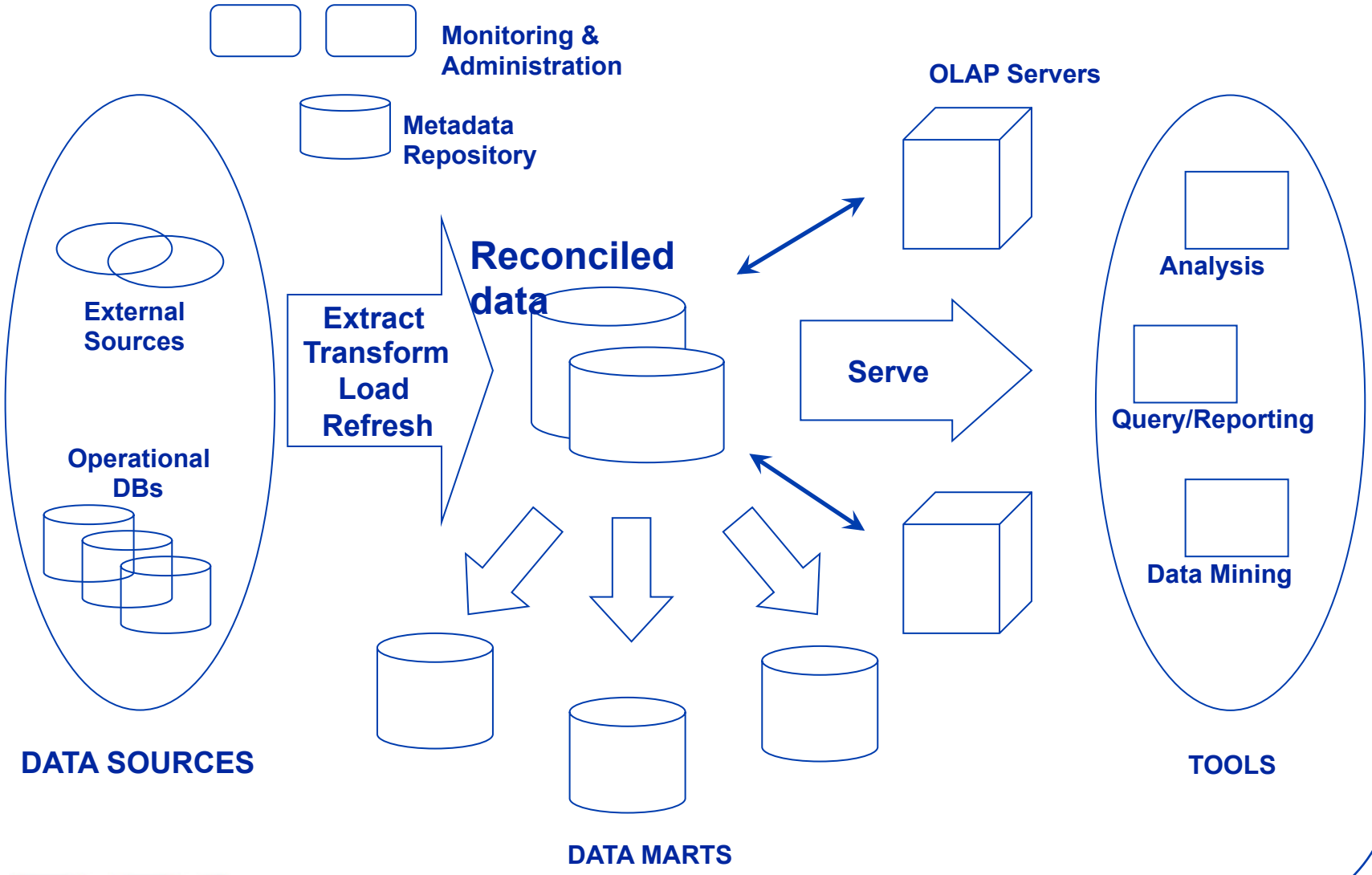
INFSCI 2711 slides of Prof. Zadorozhny class

Warehousing issues

- **Semantic Integration**: When getting data from multiple sources, must eliminate mismatches, e.g., different units (temperature, weight, currency).
- **Heterogeneous Sources**: Must access data from a variety of source formats and repositories
- **Load, Refresh, Purge**: Must load data, periodically refresh it, and purge too-old data
- **Metadata Management**: Must keep track of sources, loading time, and other information for all data in the warehouse

Data Warehouse Architecture

Data warehouse architecture



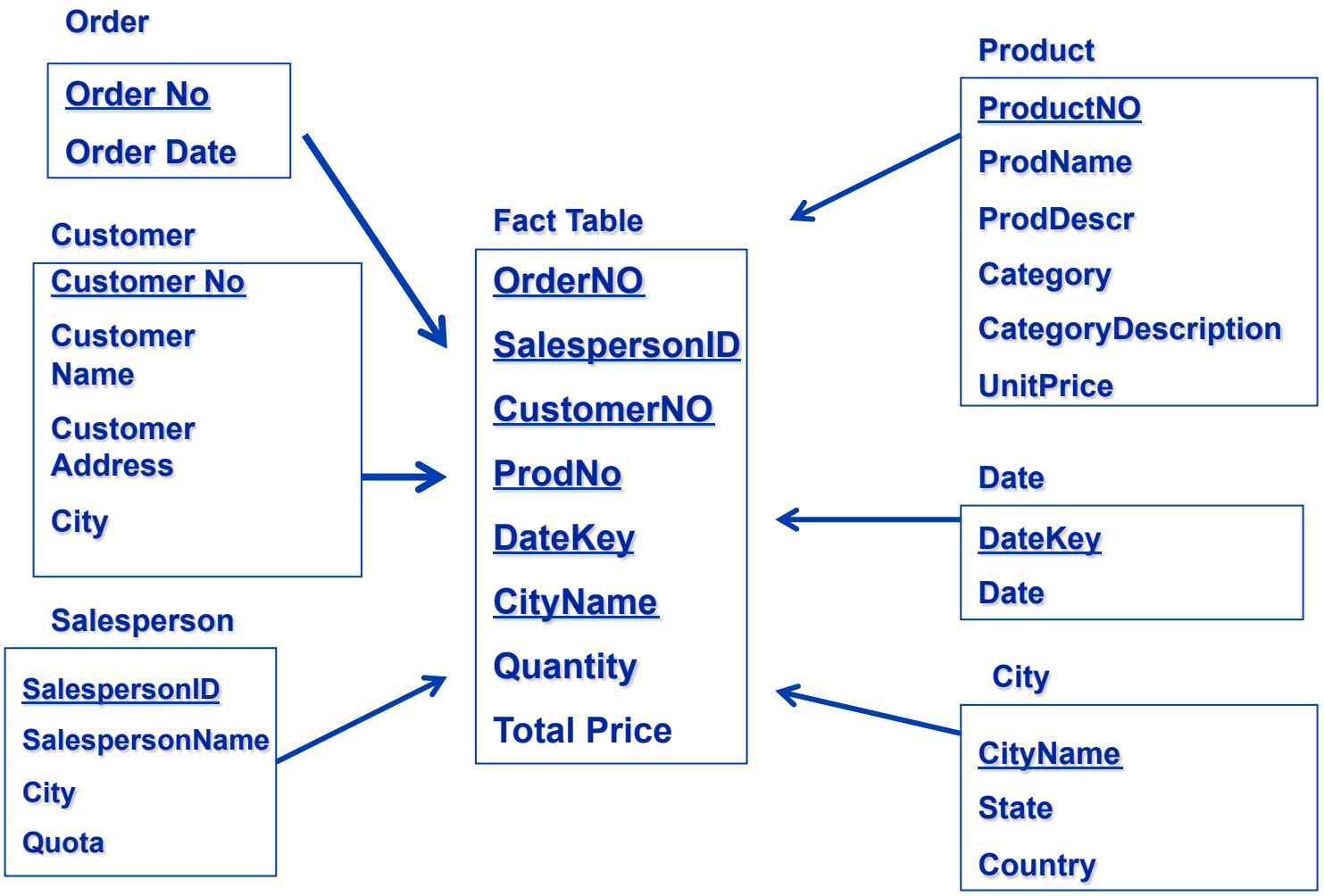
Data warehouse architecture

- Data warehouse server
  - almost always a relational DBMS, rarely flat files
- OLAP servers
  - to support and operate on multi-dimensional data structures
- Clients
  - Query and reporting tools
  - Analysis tools
  - Data mining tools

Data warehouse schema

- “Star” schema
- “Fact constellation” schema
- “Snowflake” schema

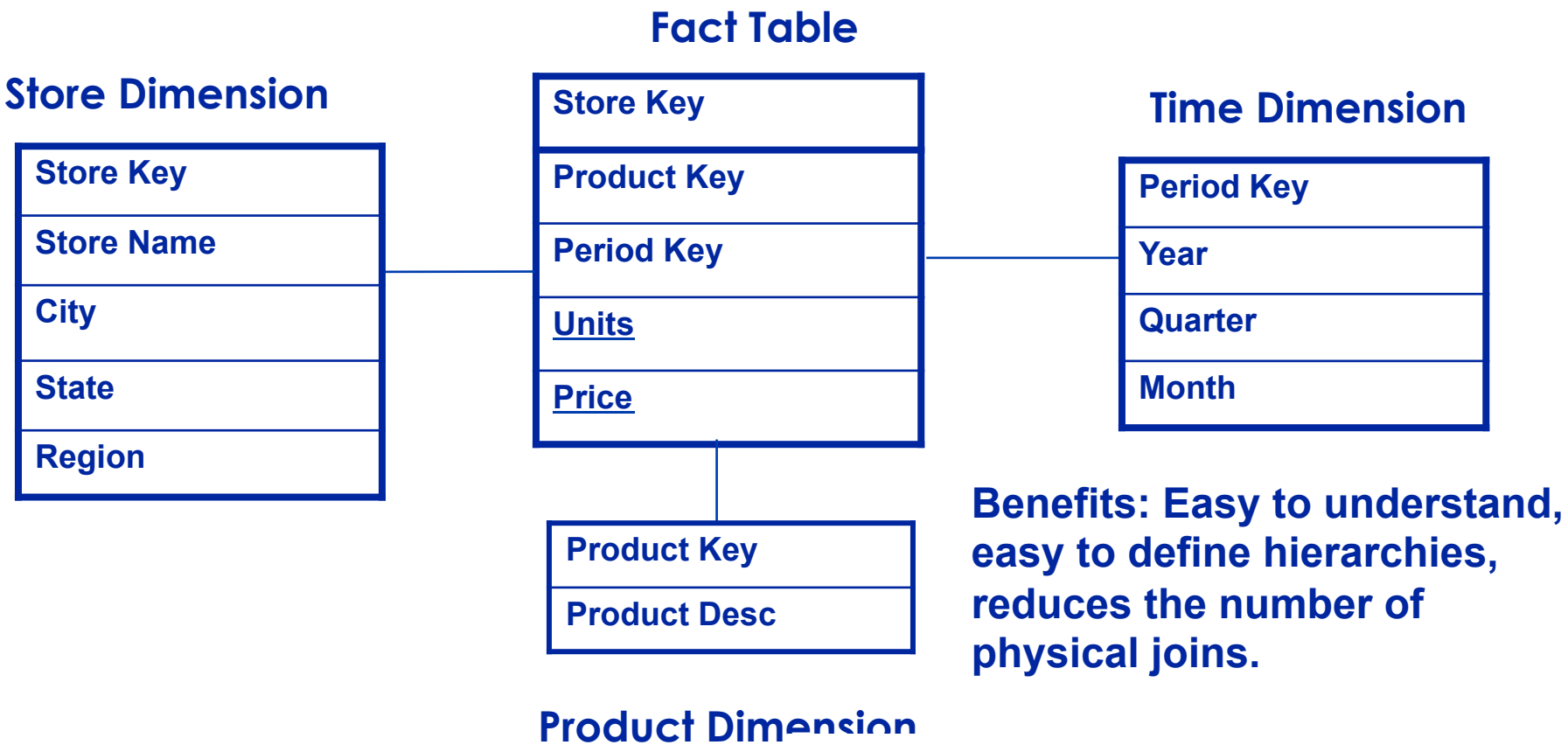
Example of a "star" schema



"Star" schema

- A single, typically large, fact table and one table for each dimension
- Every fact points to one tuple in each of the dimensions and has additional attributes
- Does not capture hierarchies directly
- Generated keys are used for performance and maintenance reasons

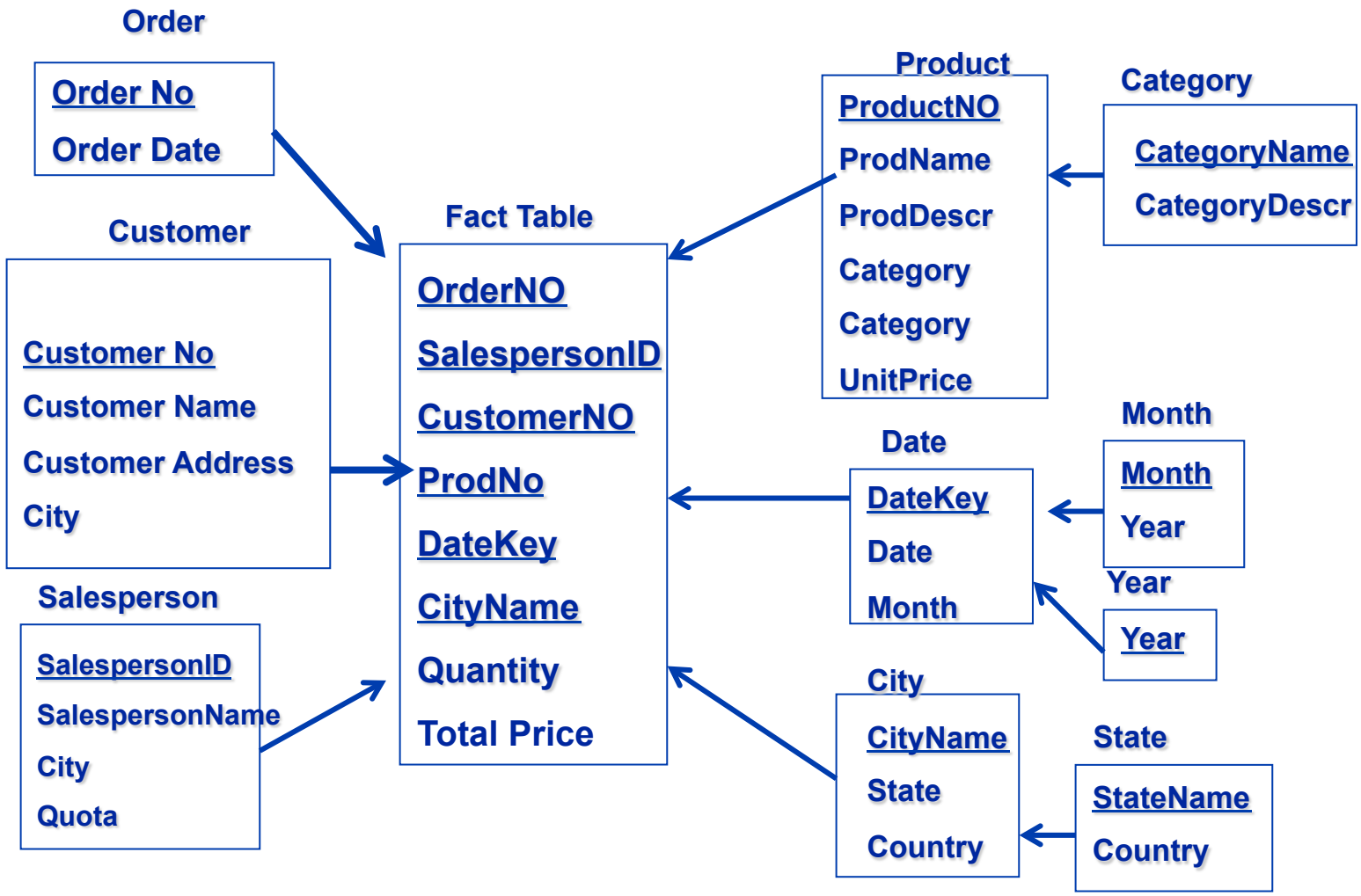
"Star" schema (cont.)



Benefits: Easy to understand, easy to define hierarchies, reduces the number of physical joins.

Fact table is large, updates are frequent; dimension tables are small, updates are rare

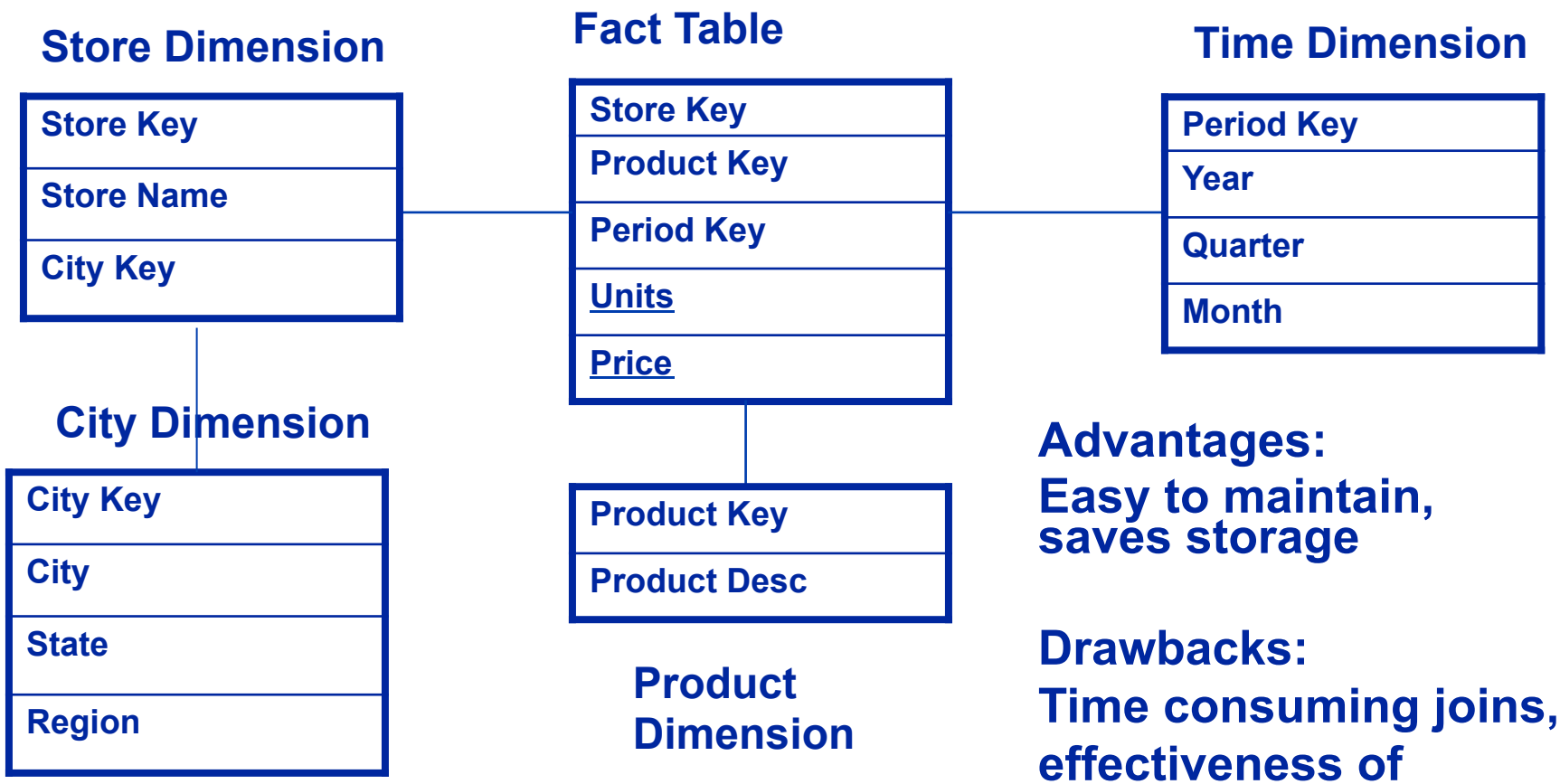
Example of a "snowflake" schema



"Snow flake" schema

- A single, large, and central fact table and one or more tables for each dimension.
- Dimension tables are normalized, i.e., split dimension table data into additional tables
- Represent dimensional hierarchy directly by normalizing the dimension tables

"Snow flake" schema (cont.)



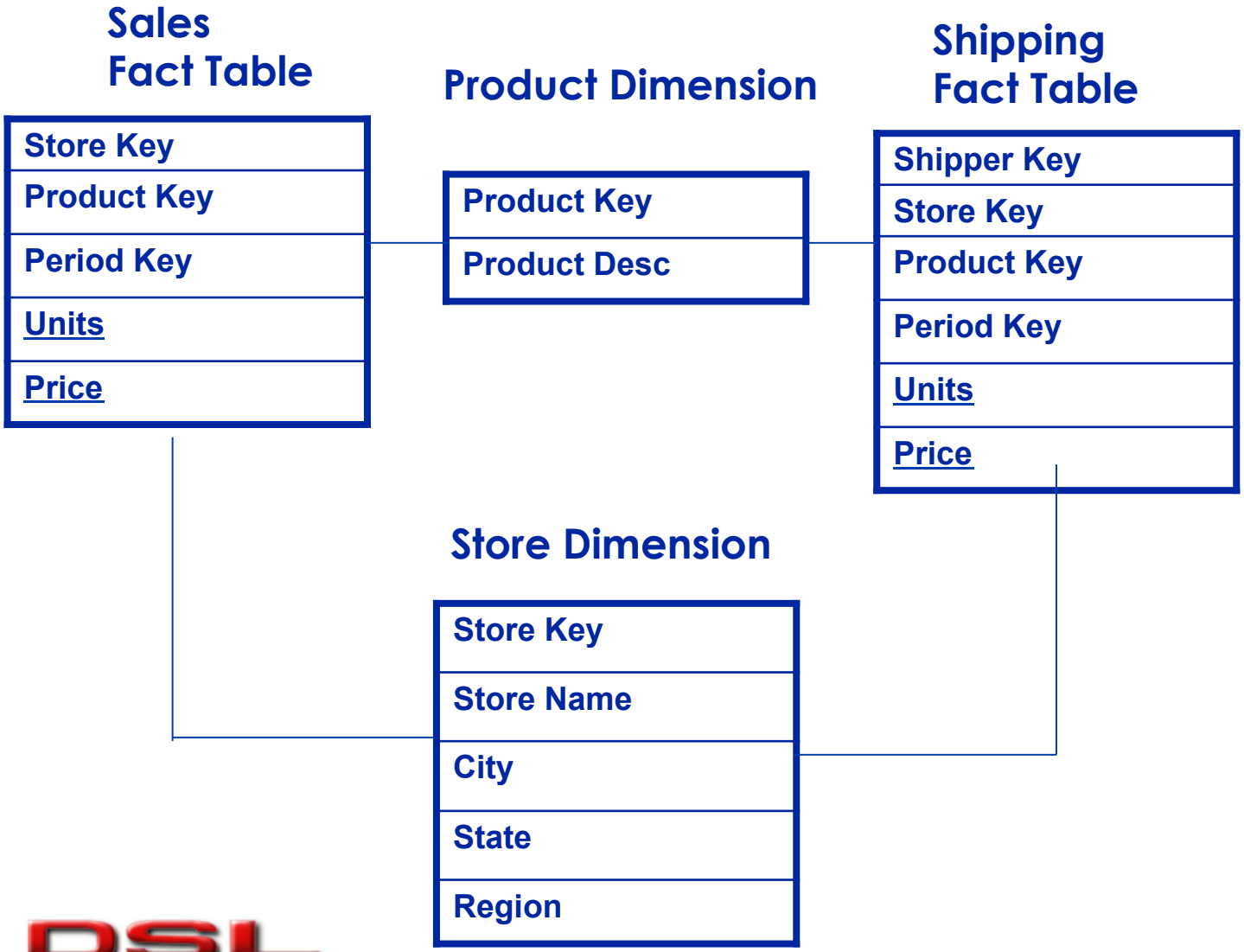
Advantages:
Easy to maintain, saves storage

Drawbacks:
Time consuming joins, effectiveness of browsing suffers, report generation may be slow

"Fact constellation" schema

- Multiple fact tables share dimension tables.
- This schema is viewed as collection of stars hence called "galaxy schema" or "fact constellation."
- Sophisticated application may require such schema.
- Example: Projected expense and the actual expense may share dimensional tables.

"Fact constellation" schema (cont.)



Virtual warehouse

Created by providing a database view on the operational databases.

- Materialize some summary views for efficient query processing
- Easier to build ☺
- May put too much load on the operational DB servers ☹

OLAP

OLAP and Data Warehousing

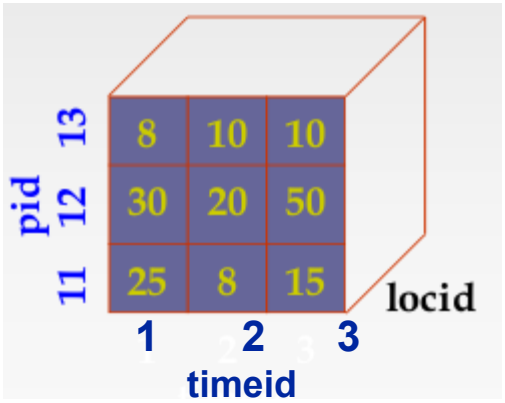
- Data Warehousing: Consolidate data from many sources into one large repository
  - ETL: Extract, Transform (semantic integration), and Load
- OLAP: Online Analytic Processing
  - Complex SQL queries and views
  - Queries based on spreadsheet-style and "multidimensional" view of data
  - Interactive and "online" queries

OLAP: Multidimensional data model

Collection of numeric measures which depend on a set of dimensions

e.g., measure Sales, dimensions Product (pid), Location (locid), and Time (timeid).

Slice locid = 1 is shown:



pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

Sales

OLAP queries

- Influenced by both SQL and spreadsheets
- A common operation is to aggregate (roll-up) a measure over one or more dimensions
  - Find total sales
  - Find total sales for each city or for each state
  - Find top five products ranked by total sales

Operations in multidimensional data model

- **Aggregation** (*roll-up*)
  - dimension reduction: e.g., total sales by city
  - summarization over aggregate hierarchy: e.g., total sales by city and year -> total sales by region and by year
- **Selection** (*slice*) defines a sub-cube
  - e.g., sales where city = Palo Alto and date = 1/15/96
- **Navigation** to detailed data (*drill-down*)
  - e.g., (sales - expense) by city, top 3% of cities by average income
- **Visualization** operations (e.g., *pivot*)

OLAP Queries

- **Drill-down**: The inverse of roll-up
  - e.g., given total sales by state, can drill-down to get total sales by city
  - e.g., can also drill-down on different dimension to get total sales by product for each state
- **Pivoting**: Aggregation on selected dimensions
  - e.g., pivoting on Location and Time

  yields the following cross-tabulation:

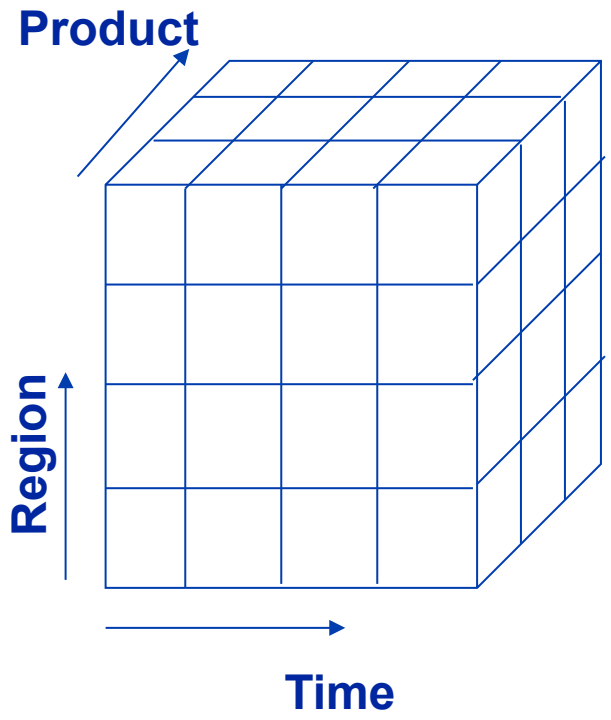
	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	339

OLAP Cube

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

OLAP operations

"drill up" (also "roll up")



Category (e.g., electrical appliance)

Subcategory (e.g., kitchen)

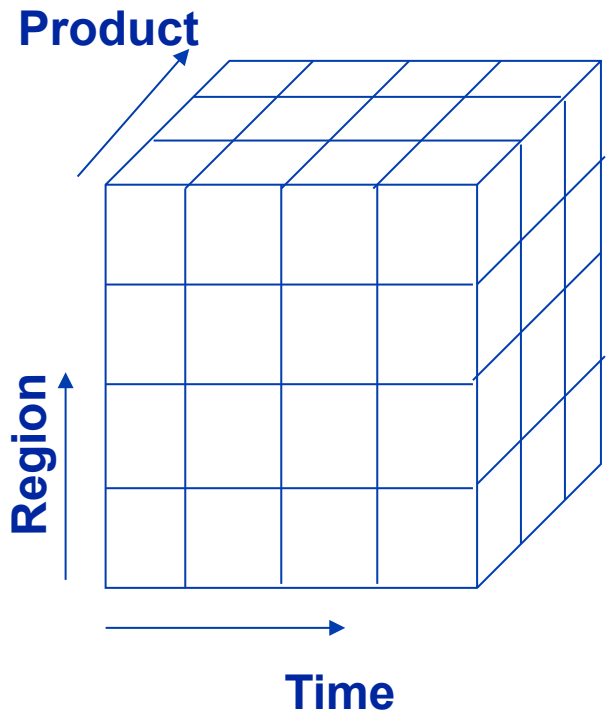
Product (e.g., toaster)

Aggregating at different levels of a dimension hierarchy

- e.g., given total sales per product, we can drill up to get sales per category
- e.g., given total sales by city, we can drill up to get sales by state

OLAP operations

"drill down" (also "roll down")



Category (e.g., electrical appliances)

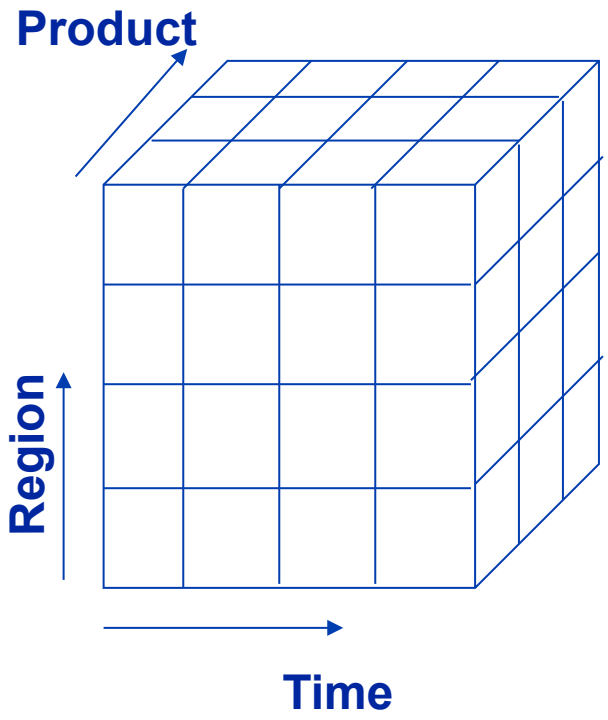
↳ Subcategory (e.g., kitchen)

↳ Product (e.g., toaster)

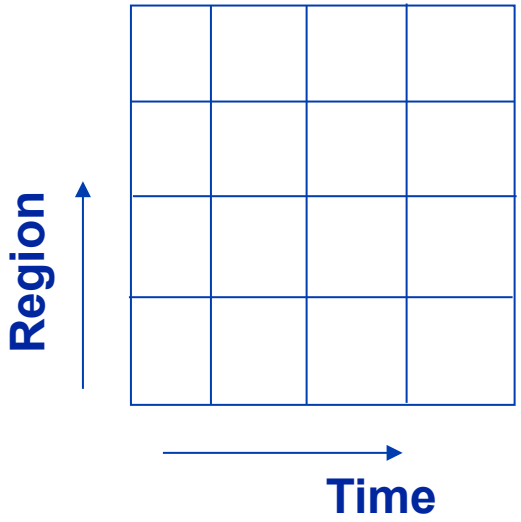
- The inverse of roll-up
  - e.g., given total sales by category, can drill-down to get total sales by product
  - e.g., can also drill-down on different dimension to get total sales by product for each state

OLAP operations

"slice and dice"



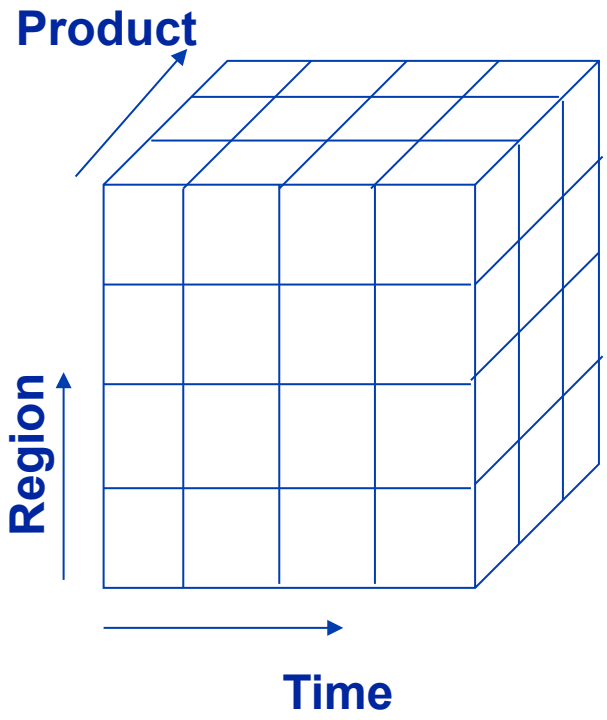
Product=Toaster

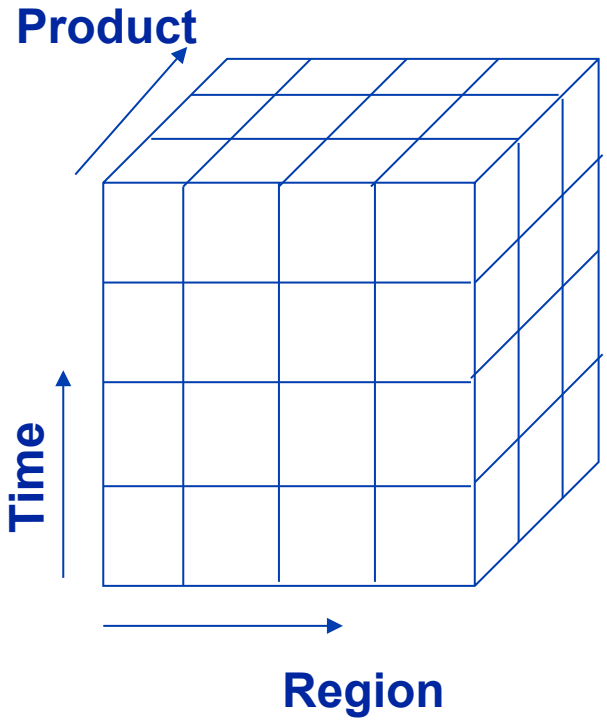


- a.k.a. "selection"
- defines a sub-cube
  - e.g., sales of toasters
  - e.g., sales where region = PA and date = 1/15/96

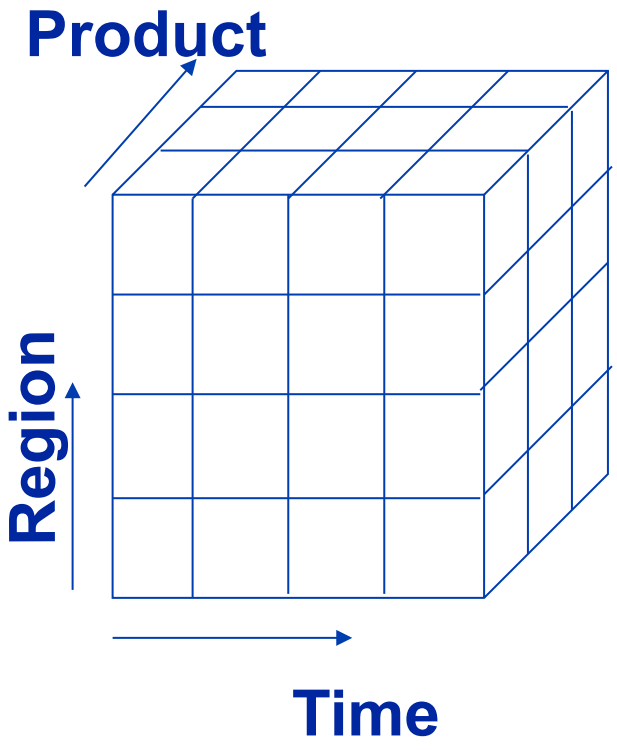
OLAP operations

“pivot” (rotation), a visual operation

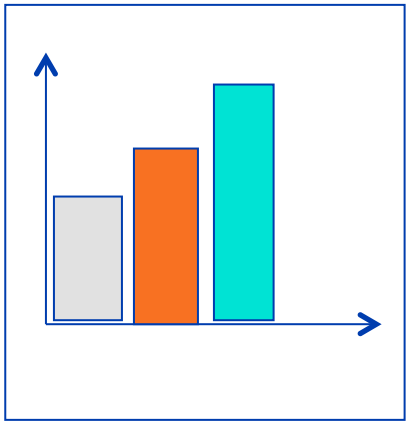




Presentation



Reporting Tool



Report

Hadoop and MapReduce

What's in the name?

"The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid's term."

-- Hadoop project's creator, Doug Cutting

Tom White, Hadoop: The Definitive Guide, 3rd Edition, 2012

Motivation

- 1990:
  - One drive – 1,370 MB with transfer speed of 4.4 MB/s
  - Read full drive in around 5 minutes.
- Today:
  - One drive – 1Tb with transfer speed of 100 MB/s (access speed has not kept up with disk capacity)
  - Read full drive in 2.5 hours (writing is even slower!)
- What if
  - We had 100 drives, each holding 1/100 of the data
  - We could read the data in less than 2 minutes

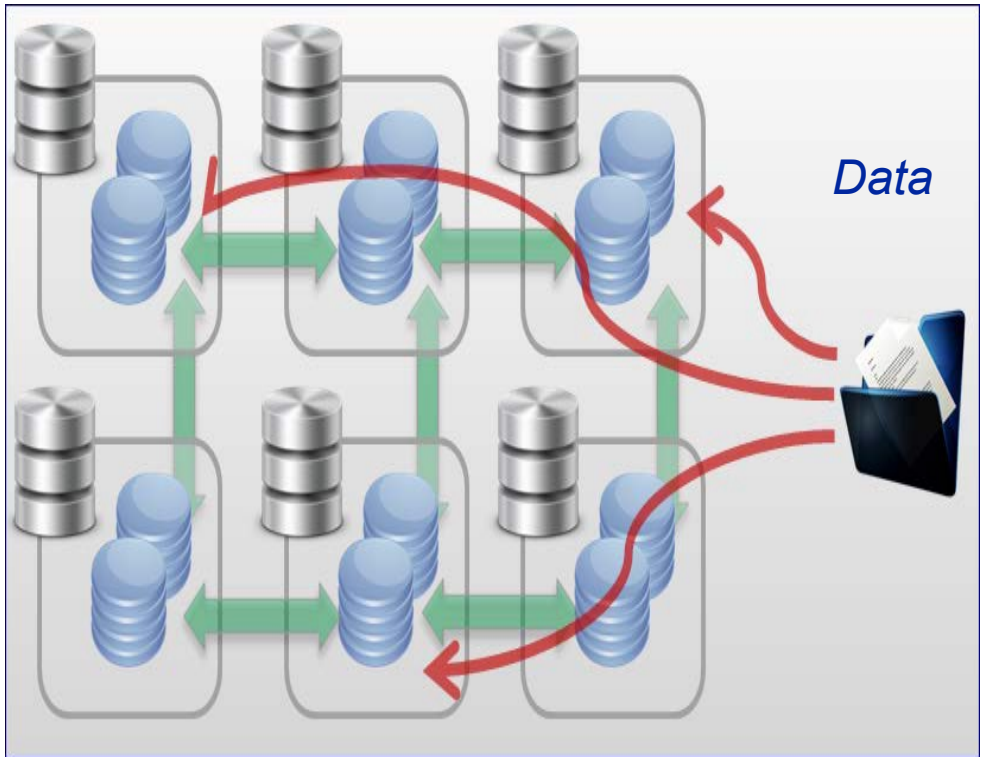
Tom White, Hadoop: The Definitive Guide, 3rd Edition, 2012

What is Hadoop?

Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware

- **Scalable**
- **Reliable**
- **Cluster of inexpensive commodity hardware**

Moving Computation to Data



Computation

Hadoop

The Hadoop project includes:

- **Hadoop Common**: The common utilities that support the other Hadoop modules (A set of components and interfaces for distributed file systems and general I/O, e.g., serialization, Java RPC, persistent data structures).
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data located on large clusters of commodity machines.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets. A distributed data processing model and execution environment that runs on large clusters of commodity machines.

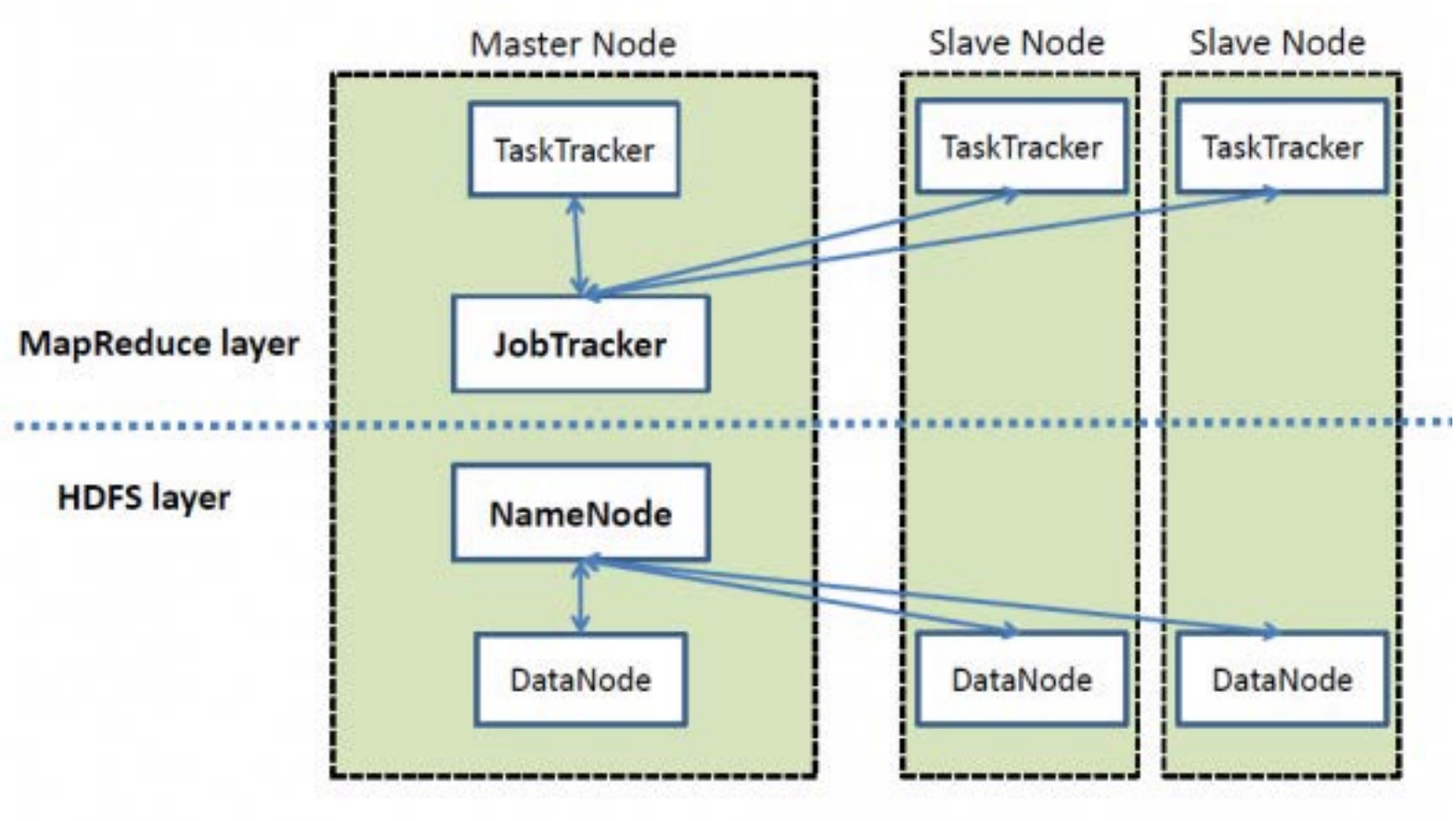
http://hadoop.apache.org/

Hadoop and MapReduce

- Automatic parallel execution, fault tolerance, load balancing
- Run-timer takes care of failing nodes, data partitioning, result merging
- Runs on huge cluster of commodity machines
- Primitive operations: split the data, process them separately, combine the result

More than ten thousand distinct programs have been implemented using MapReduce at Google

Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified data processing on large clusters. *CACM* 51(1), January 2008


Master Node
Slave Node
Slave Node
TaskTracker
TaskTracker
TaskTracker
JobTracker
MapReduce layer
HDFS layer
NameNode
DataNode
DataNode
DataNode

NameNode and DataNode

- Single NameNode
  - manages the file system namespace
  - regulates access to files by clients
  - opening, closing, and renaming files and directories
  - determines the mapping of blocks to DataNodes
- Many DataNodes
  - serving read and write requests from the file system's clients
  - block creation, deletion, and replication upon instruction from the NameNode

http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

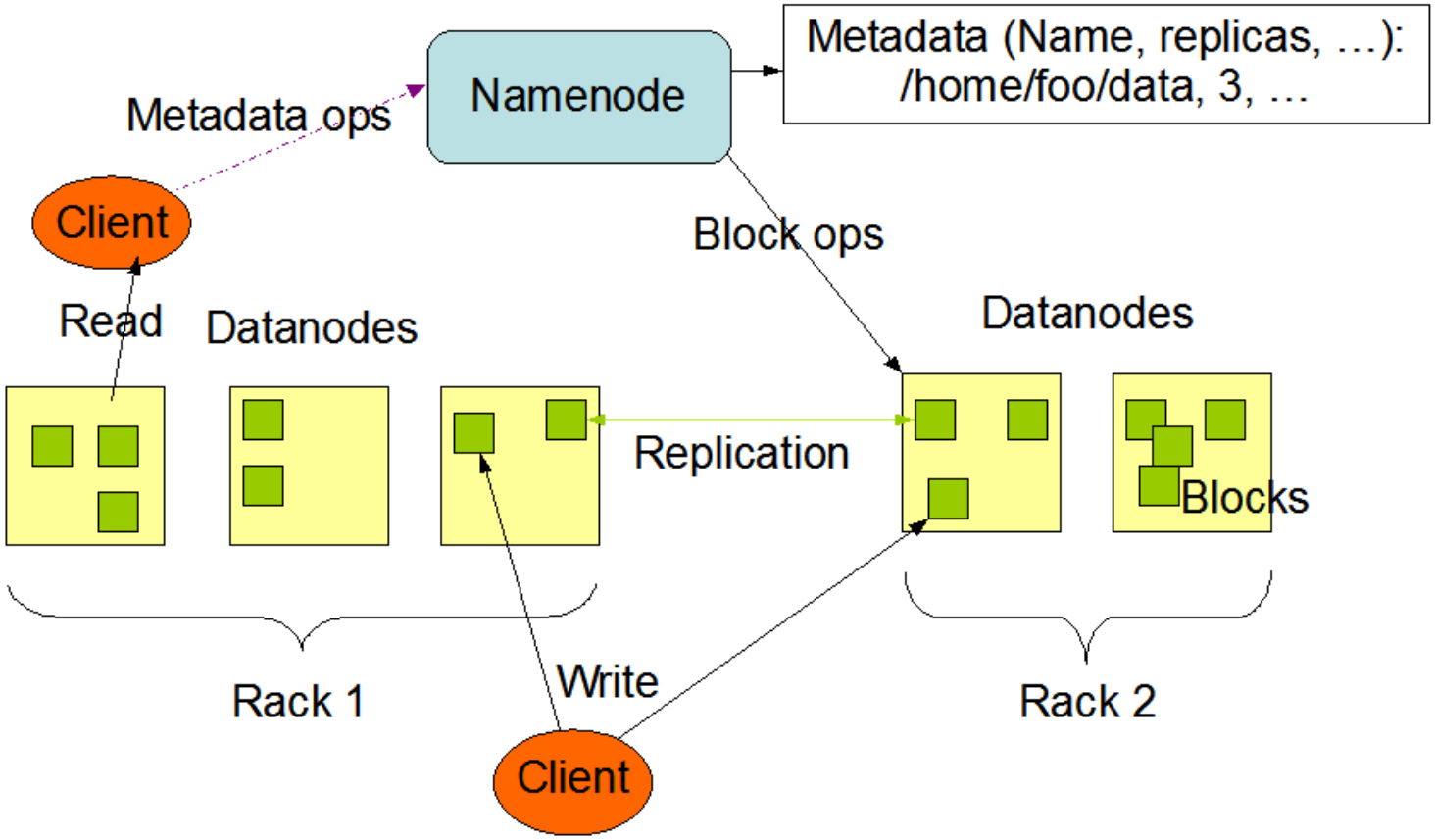
Commands

- hadoop fs -<command> <arguments>
- help,
- cat, chmod, cp, get
- ls, mkdir, mv, put
- rm, rmr
- copyFromLocal, copyToLocal

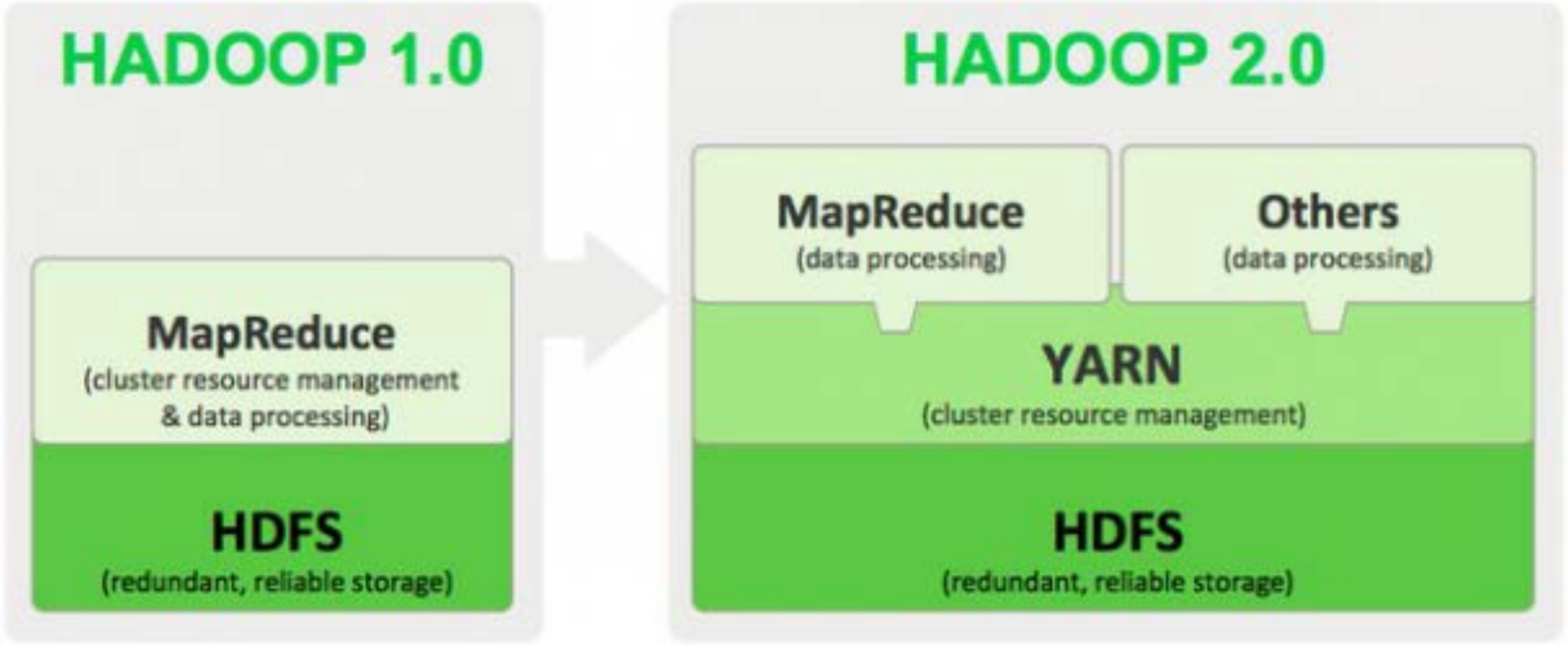
http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

Architecture

HDFS Architecture

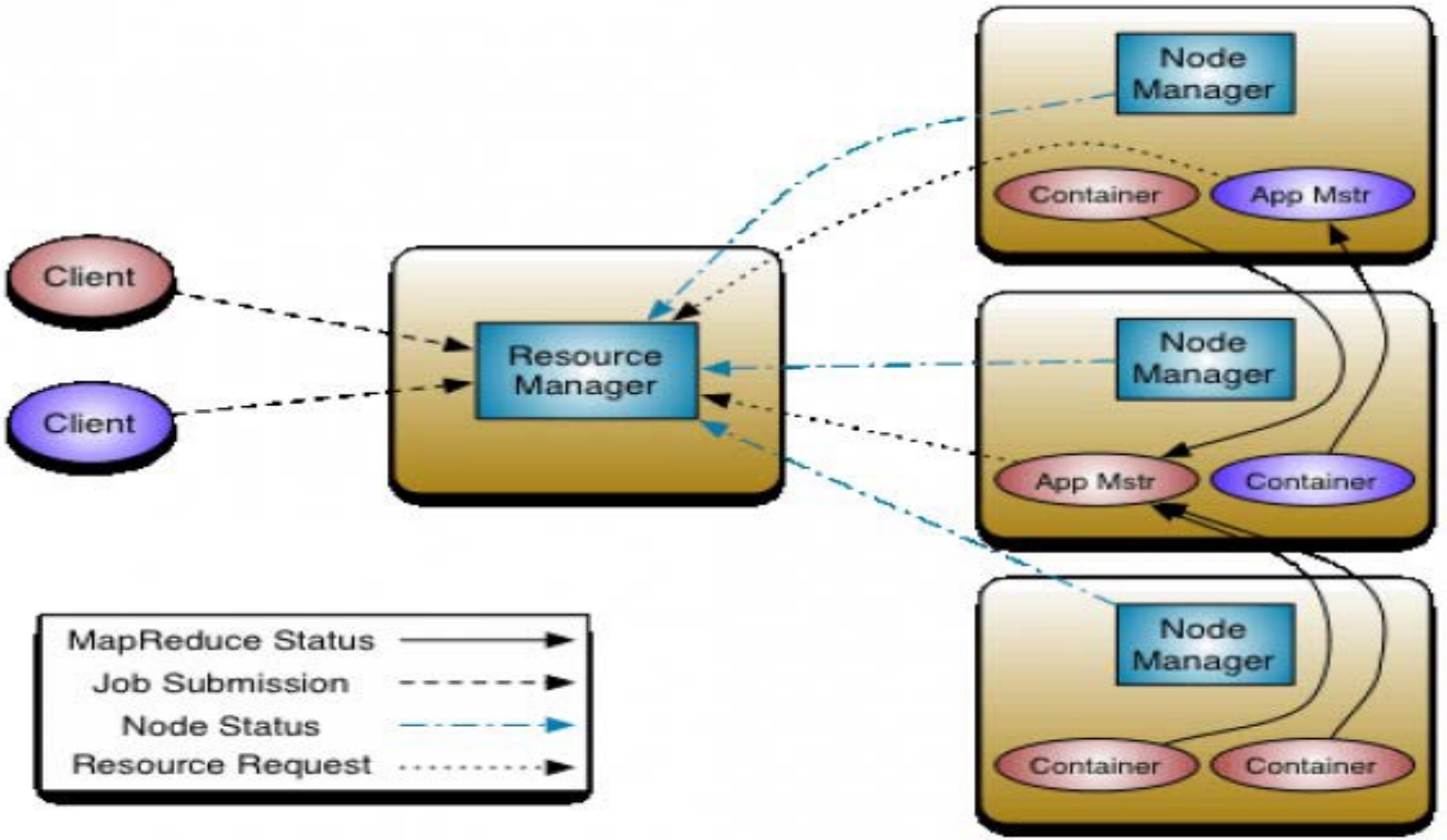


http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html



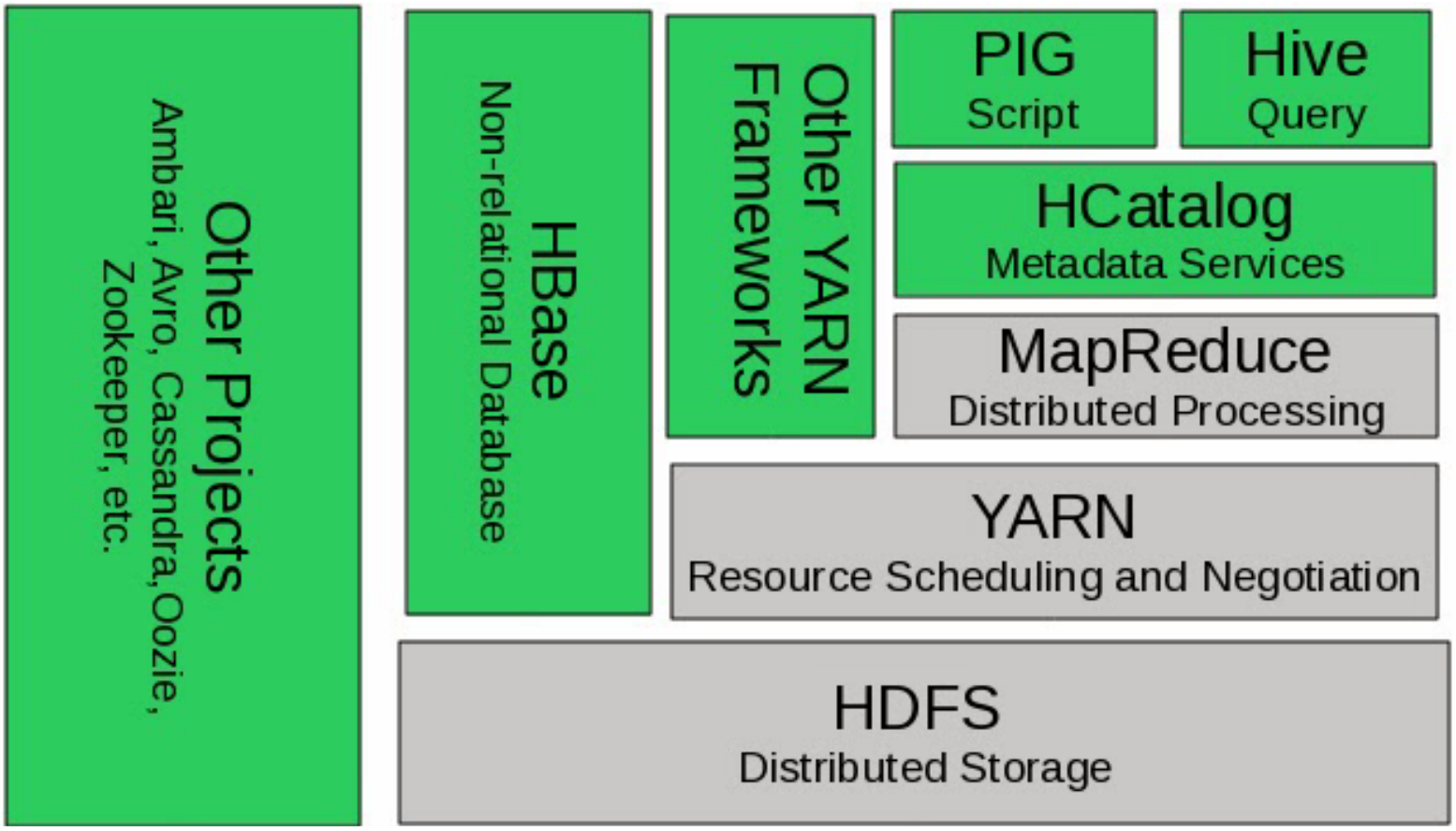
YARN enhances the power of a Hadoop compute cluster

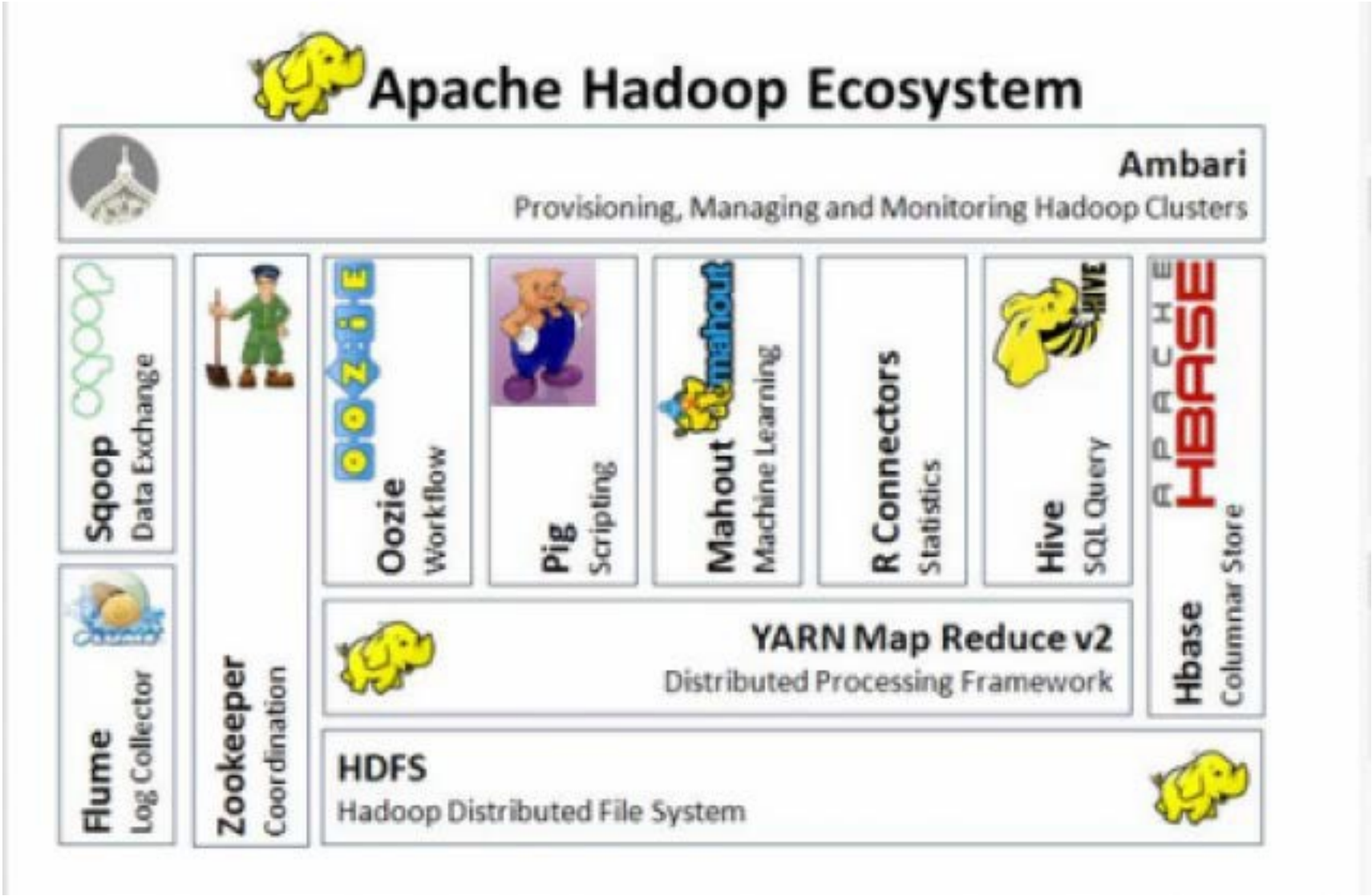
Apache Hadoop NextGen MapReduce (YARN)



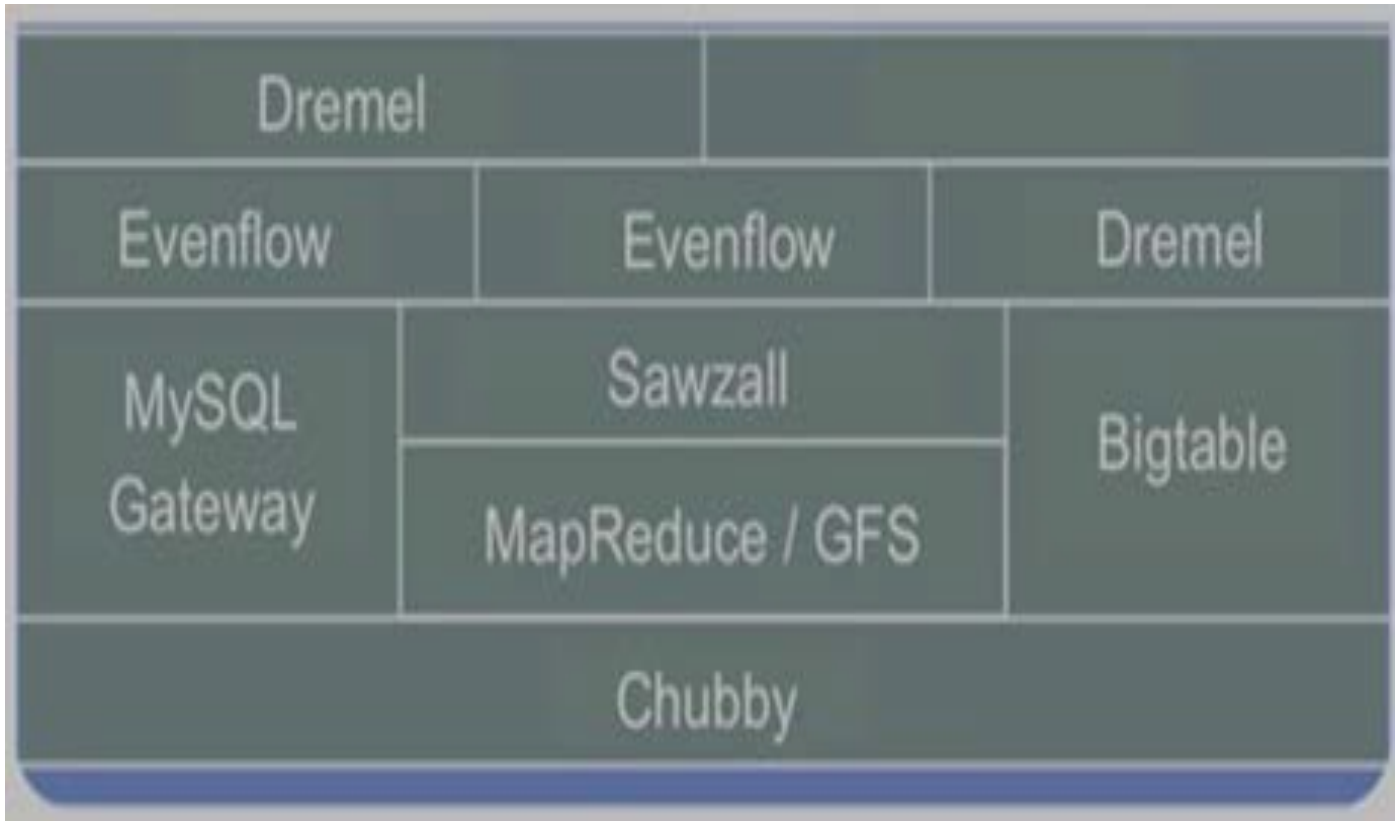
Improved cluster utilization

Supports Other Workloads

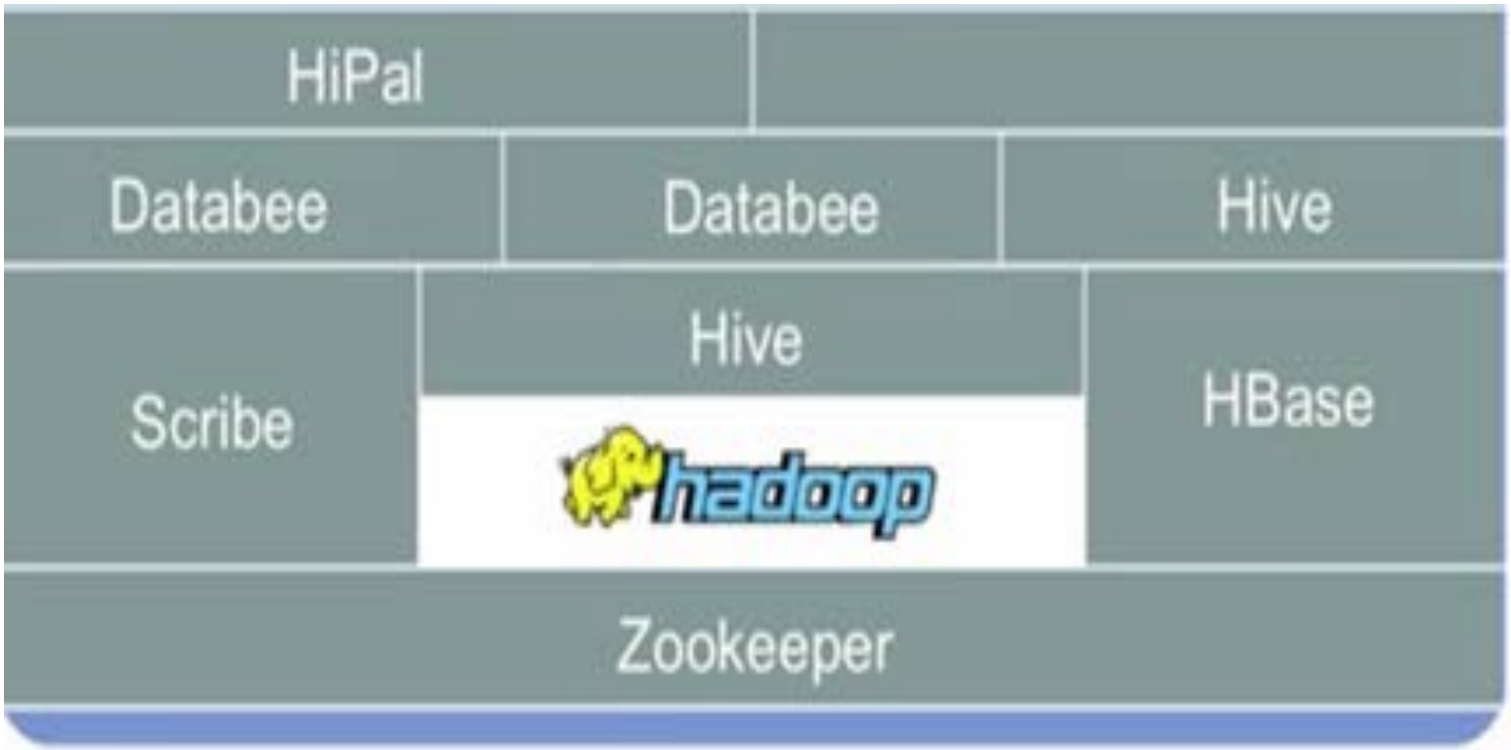

hadoop
Other Projects
Ambari, Avro, Cassandra, Oozie, Zookeeper, etc.
HBase
Non-relational Database
Other YARN Frameworks
PIG
Script
Hive
Query
HCatalog
Metadata Services
MapReduce
Distributed Processing
YARN
Resource Scheduling and Negotiation
HDFS
Distributed Storage


Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

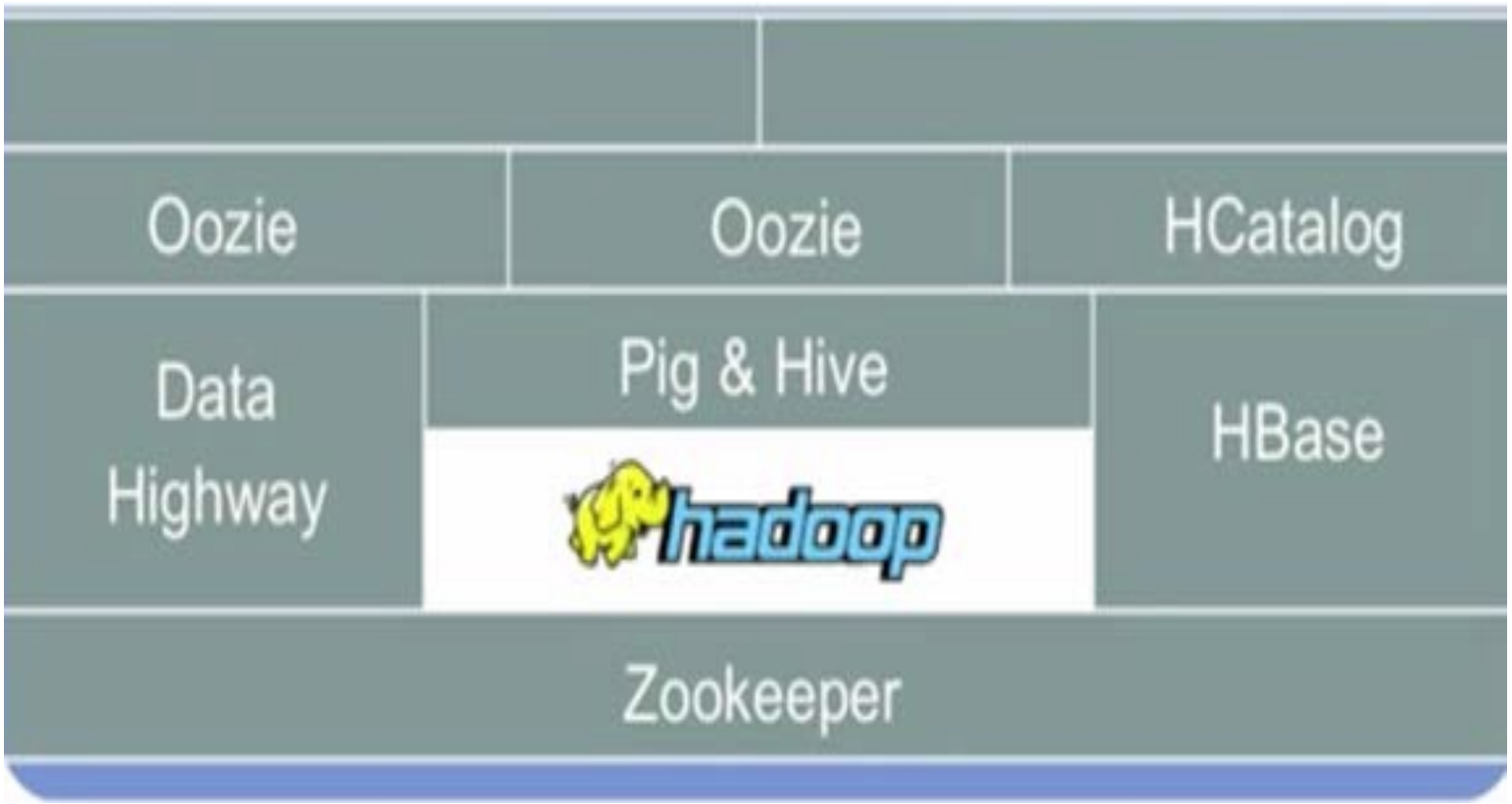
Original Google Stack



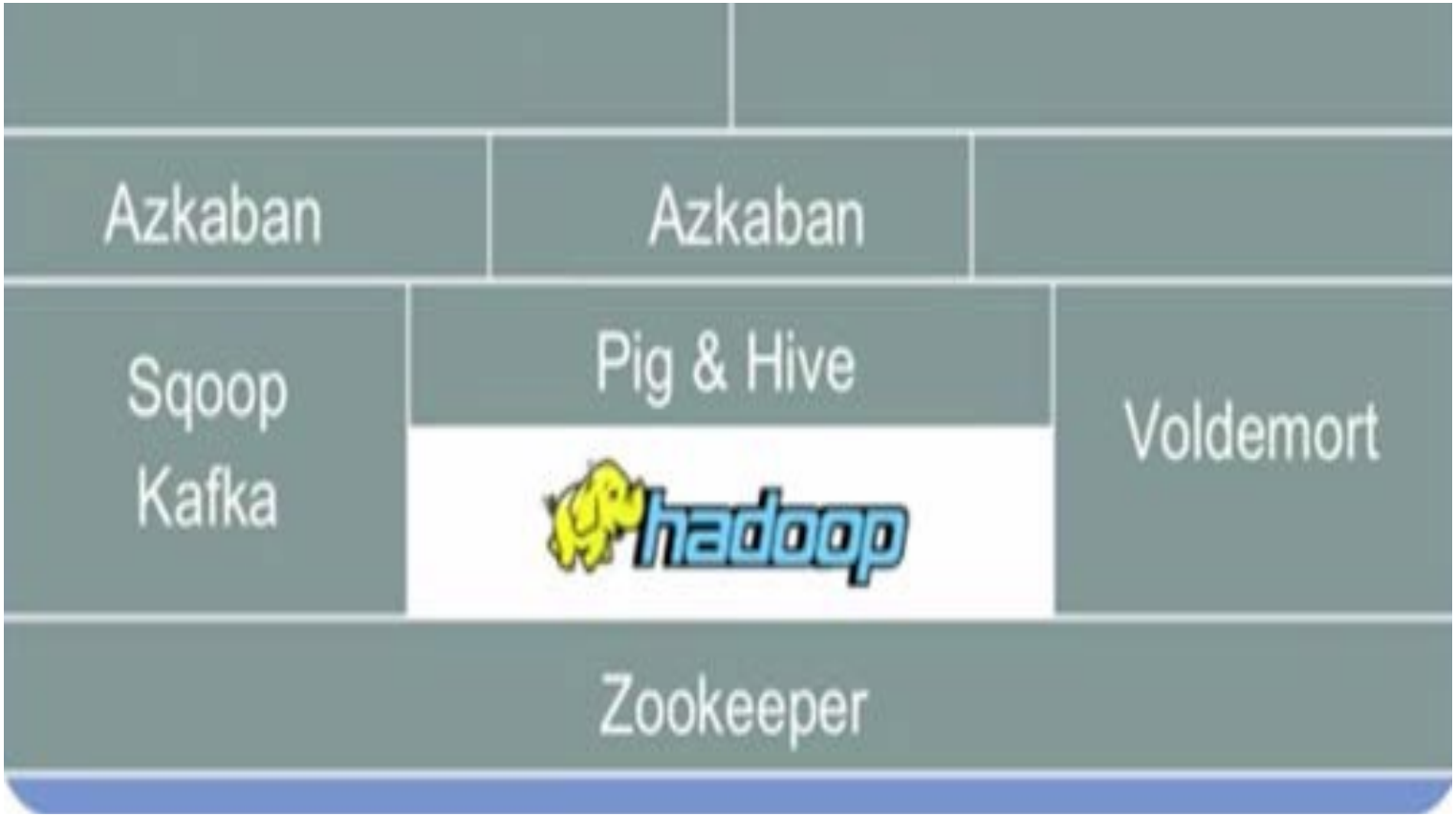
Facebook's Version of the Stack



Yahoo's Version of the Stack



LinkedIn's Version of the Stack

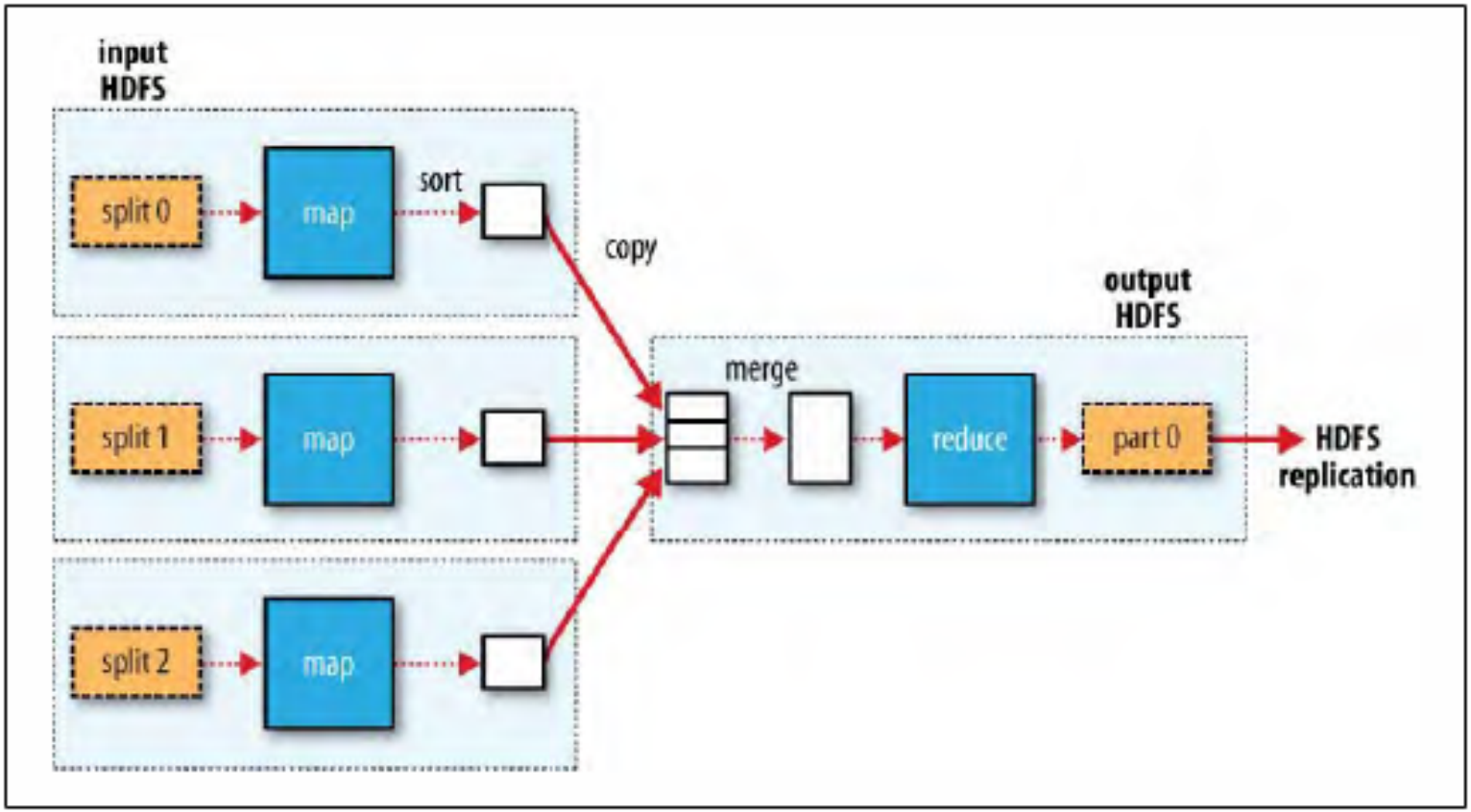


Hadoop and MapReduce

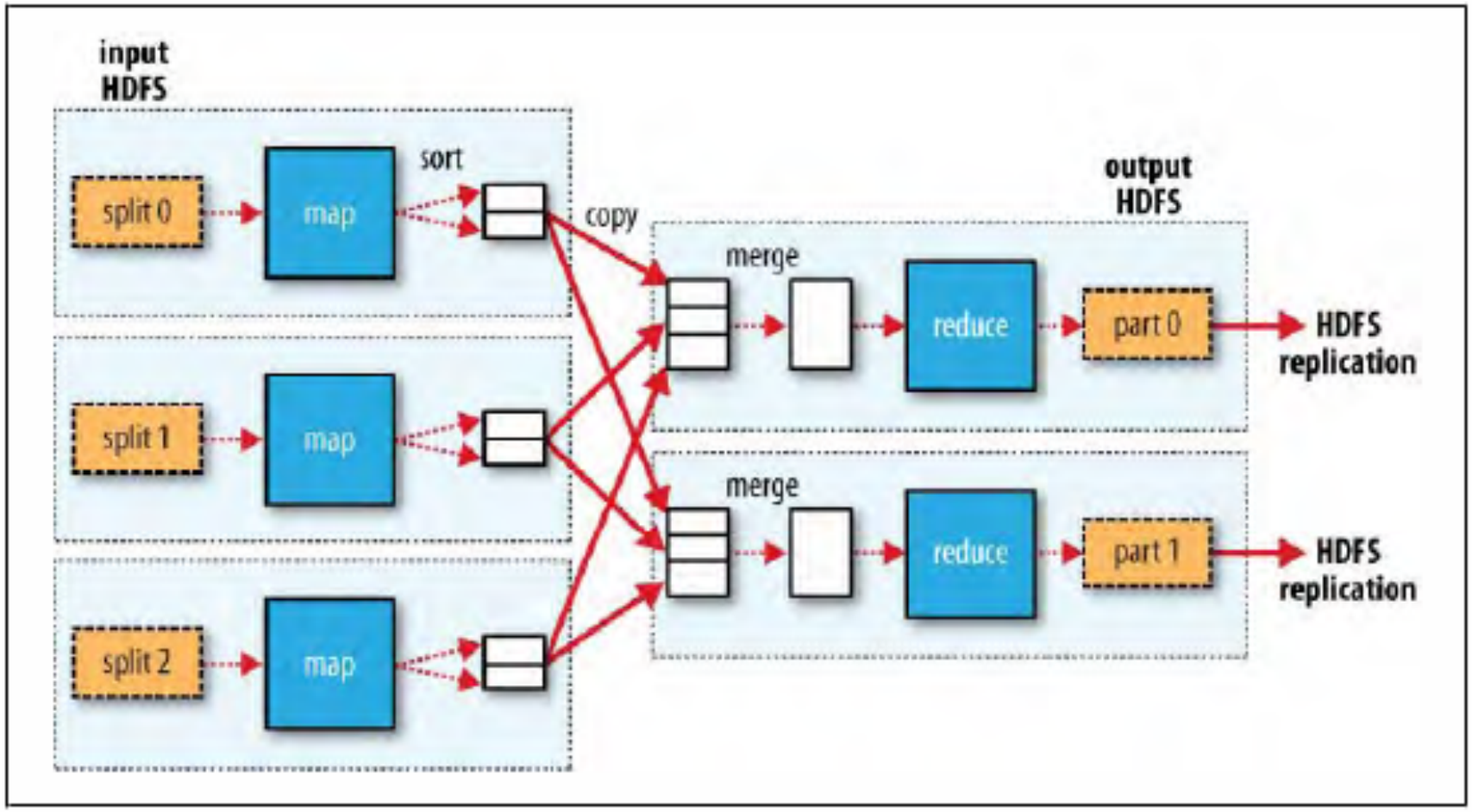
Programming model:

- Input – key/value pairs
- Output – key/value pairs
- Two functions:
  - » Map: (K1, V1) → list(K2, V2)
    - MapReduce library: groups intermediate results and sends to reduce
  - » Reduce: (K2, list(V2)) → list(K3, V3)

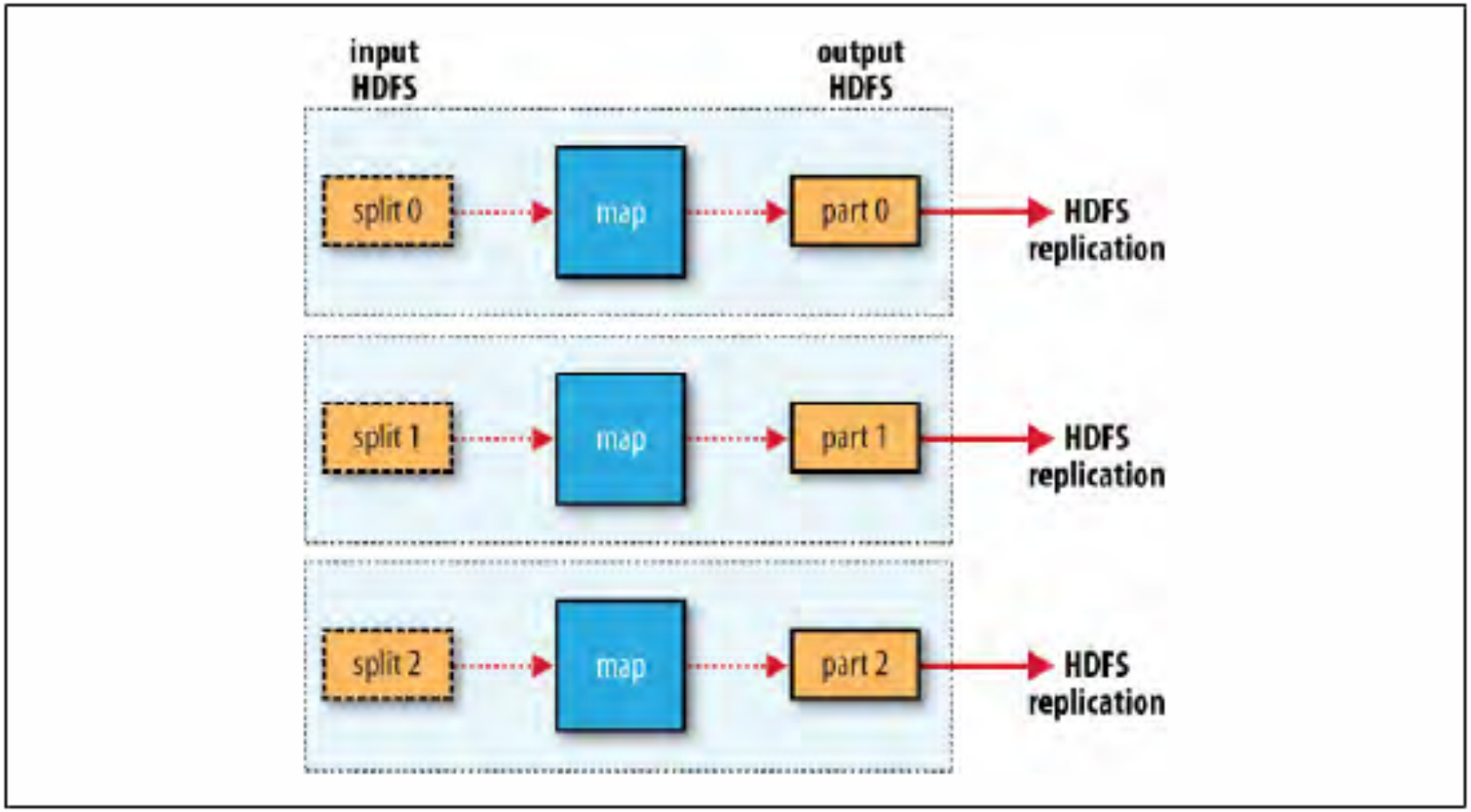
MapReduce data flow (single reduce task)



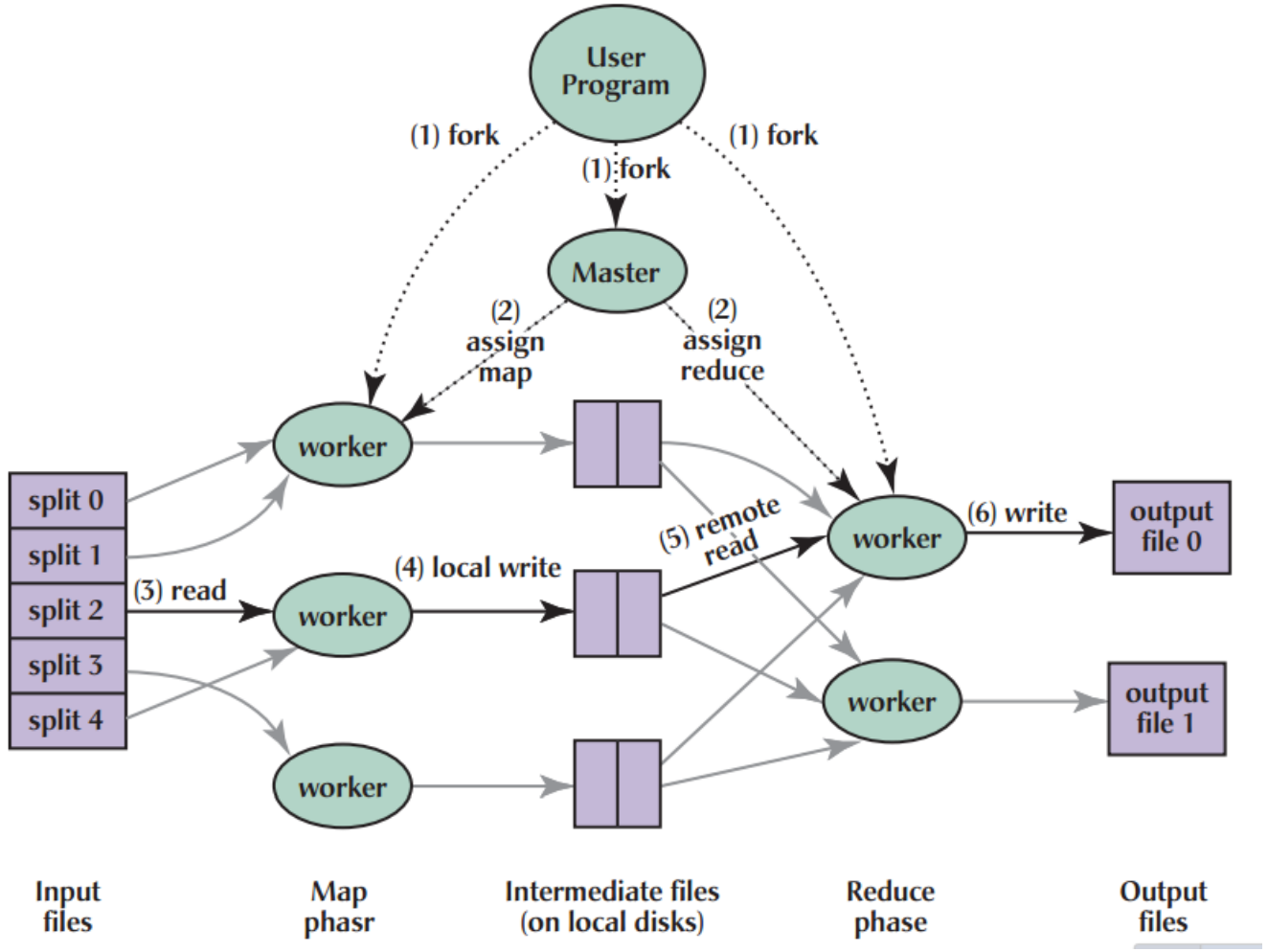
MapReduce data flow (multiple reduce tasks)



MapReduce data flow (no reduce tasks)



Hadoop and MapReduce



Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified data processing on large clusters. *CACM* 51(1), January 2008

Fault tolerance

- Master pings workers, and reassigns the chunk of work of a failed worker to another worker and notifies other workers of re-execution
- Master periodically writes checkpoints. If master fails – MapReduce operation fails and client may re-execute it again, starting from the last checkpoint

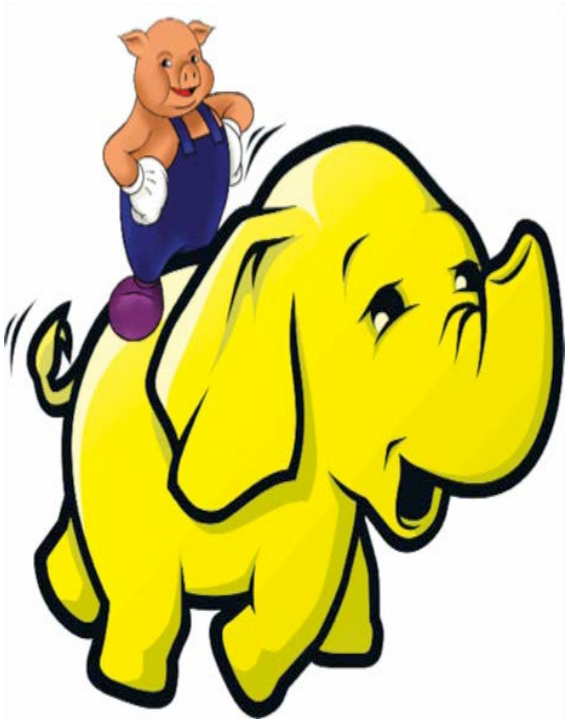
Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified data processing on large clusters. *CACM* 51(1), January 2008

Pig

PIG

High level programming on top of Hadoop MapReduce

- **The language: Pig Latin**
- **Data analysis problems as data flows**
- **Originally developed at Yahoo 2006**



Pig

http://en.wikipedia.org/wiki/Pig_Latin

- Pig is made up of two pieces:
  - The language used to express data flows, called Pig Latin.
  - The execution environment to run Pig Latin programs. There are currently two environments: local execution in a single JVM and distributed execution on a Hadoop cluster.
- A Pig Latin program is made up of a series of operations, or transformations, that are applied to the input data to produce output.

Tom White, Hadoop: The Definitive Guide, 3rd Edition, 2012

Running pig programs

Three ways, work in local and MapReduce mode

- Script
  - » Pig can run a script file that contains Pig commands
- Grunt
  - » Grunt is an interactive shell for running Pig commands
- Embedded
  - » You can run Pig programs from Java using the PigServer class

Tom White, Hadoop: The Definitive Guide, 3rd Edition, 2012

Pig: Example

- Calculate maximum recorded temperature by year:

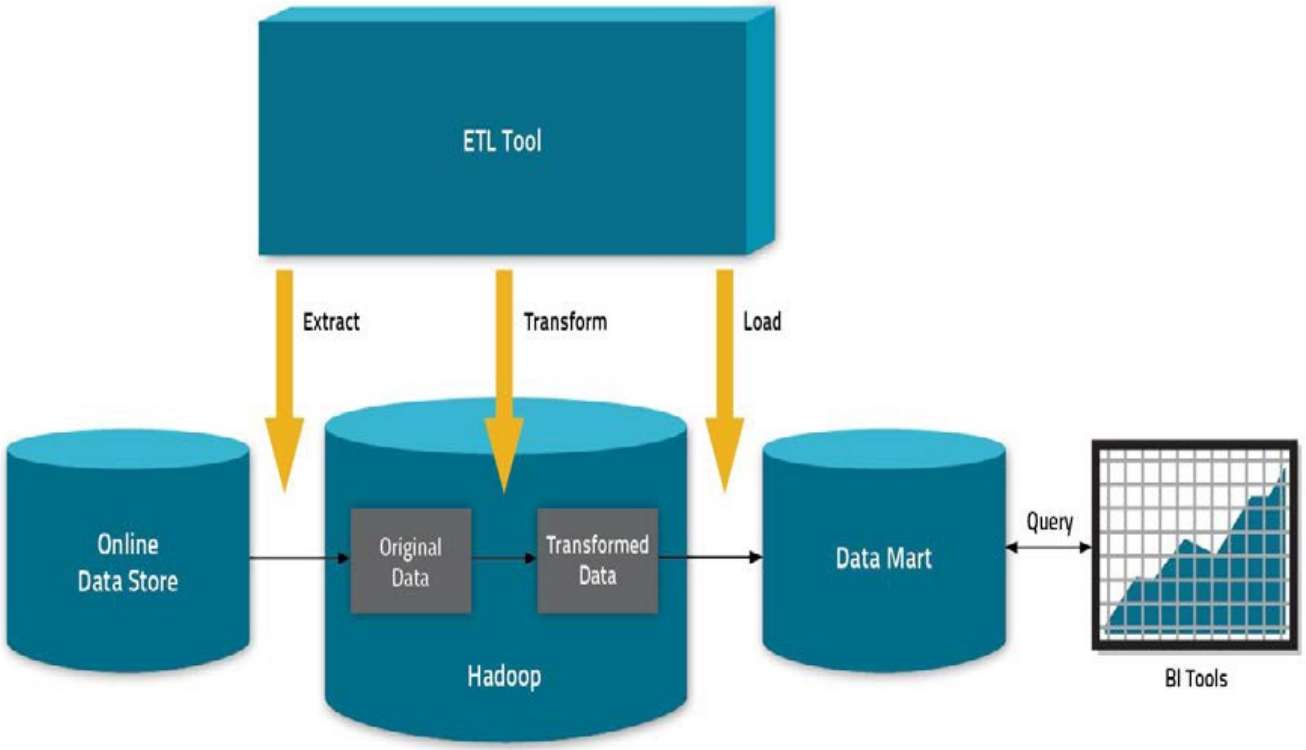
```
records = LOAD 'input/ncdc/micro-tab/sample.txt'
AS (year:chararray, temperature:int);
filtered_records = FILTER records BY temperature != 9999;
grouped_records = GROUP filtered_records BY year;
max_temp = FOREACH grouped_records GENERATE group,
MAX(filtered_records.temperature);
DUMP max_temp;
```

- Possible result:

(1949, 111)
(1950, 22)

Tom White, Hadoop: The Definitive Guide, 3rd Edition, 2012

Pig for ETL


ETL Tool
Extract
Transform
Load
Online
Data Store
Original
Data
Transformed
Data
Hadoop
Data Mart
Query
BI Tools

Hive

Hive

- Created at Facebook
- Data Warehouse on the top of Hadoop
  - Map-Reduce for execution
  - HDFS for storage
- HiveQL -SQL like query language
  - Heavily influenced by MySQL
- Storage: flat files (no indexes)

http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

Apache Hive

- Data warehouse software facilitates querying and managing large datasets residing in distributed storage
- SQL-like language!
- Facilitates querying and managing large datasets in HDFS
- Mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL

Query language

- DDL
  - {create/alter/drop} {table/view/partition}
  - create table as select
- DML
  - Insert overwrite
- QL
  - Sub-queries in from clause
  - Equi-joins (including Outer joins)
  - Multi-table Insert
  - Sampling
  - Lateral Views
- Interfaces
  - JDBC/ODBC/Thrift

QL


```
SELECT [ALL | DISTINCT]
select_expr, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[SORT BY col_list] ]
[LIMIT number]
```

Example

Calculate maximum recorded temperature by year:

- CREATE TABLE records (year STRING, temperature INT)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '\t';
- LOAD DATA LOCAL INPATH 'input/ncdc/micro-tab/sample.txt'
  OVERWRITE INTO TABLE records;
- hive> SELECT year, MAX(temperature)
  > FROM records
  > WHERE temperature != 9999
  > GROUP BY year;

1949 111

1950 22

http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

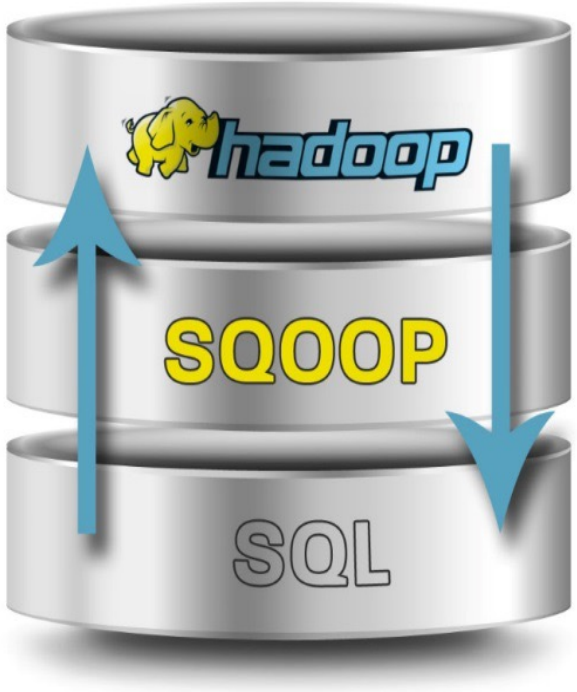
Example

- The SQL query on the previous slide is nothing special: just a SELECT statement with a GROUP BY clause for grouping rows into years, which uses the MAX() aggregate function to find the maximum temperature for each year group.
- The remarkable thing is that Hive transforms this query into a MapReduce job, which it executes on our behalf, then prints the results to the console.
- There are some nuances such as the SQL constructs that Hive supports and the format of the data that we can query—and we shall explore some of these in this chapter—but it is the ability to execute SQL queries against raw data that gives Hive its power.

http://hadoop.apache.org/docs/hdfs/r0.22.0/hdfs_design.html

Apache Sqoop

Tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases



Apache Spark™ is a fast and general engine for large-scale data processing

Speed

- Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Ease of Use

- Write applications quickly in Java, Scala, Python, R.

Generality

- Combine SQL, streaming, and complex analytics.

Runs Everywhere

- Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3.

Suggested readings (processing)

http://www.dwinfocenter.org/

http://www.vogella.com/articles/ApacheHadoop/article.html

http://hadoop.apache.org/docs/r0.20.2/mapred_tutorial.html

Pig:

http://pig.apache.org/docs/r0.9.1/start.html

Hive:

http://hive.apache.org/

Hadoop lectures (Tom White):

http://www.youtube.com/watch?v=Aq0x2z69syM

http://www.youtube.com/watch?v=2SpTvWiXBcA

LinkedIn lecture Jakob Homan

http://www.youtube.com/watch?v=SS27F-hYWfU