4 Data Processing

1. **Data Warehousing**
2. Hadoop/MapReduce
3. Pig
4. Hive

1. Data warehouses

demands -- find, get the data we need, understand and use the data we found

problems – data in many versions, subtle differences;

need expert to get data;

available data poorly documented;

unexpected results, needs to be transformed.

The need for business intelligence (competitive, access, efficiency, cost, streamlin process)

Data analysis \_provides enterprise with \_intelligence

Data warehousing\_...\_memory

Definitions of Data warehouse:

Subject-oriented, integrated, time-variant, non-volatile;

Copy of transaction data, specifically structured for query and analysis;

Single, complete and consistent store of data.

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.

Note that:

contains a copy of the transactions which are not updated or changed later by the transaction system.

specially structured, and may have been transformed when it was copied into the data warehouse.

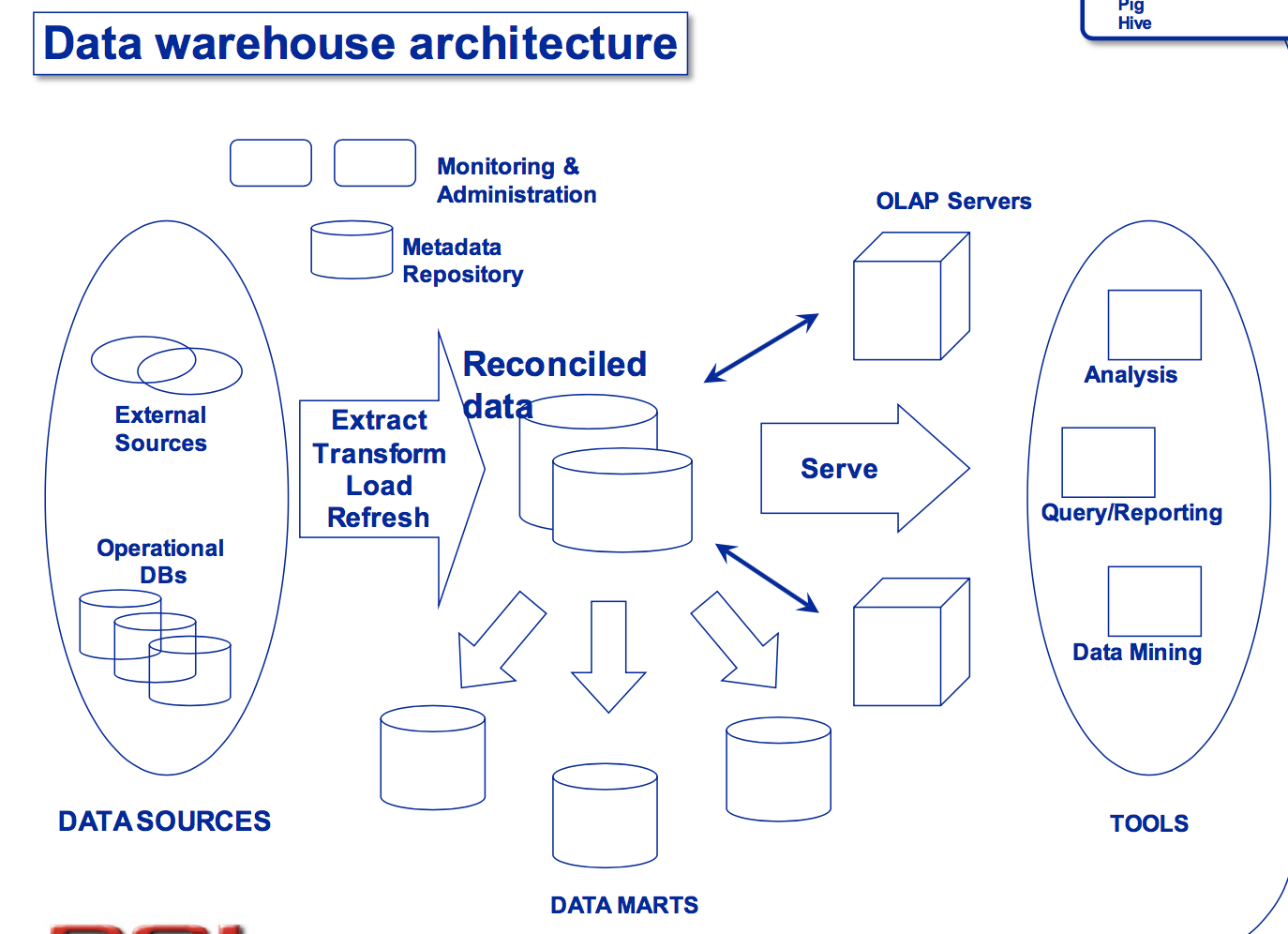
**Expectations:**

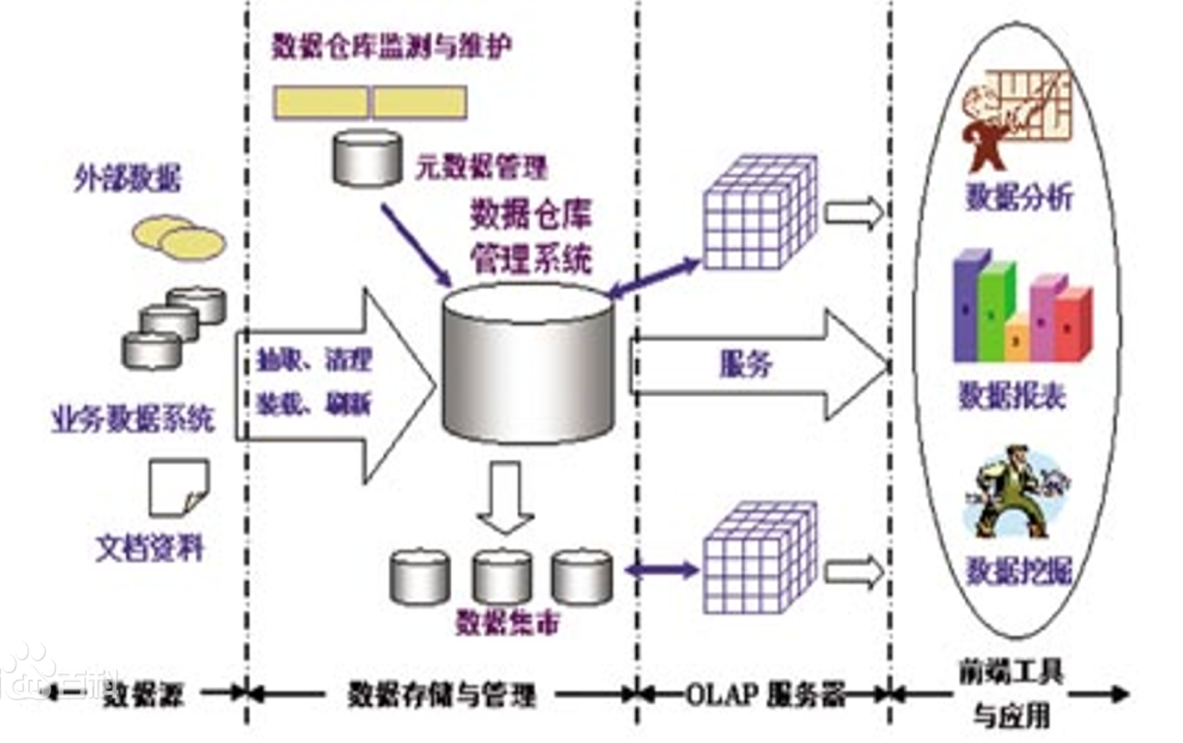
* Data should be integrated across the enterprise
* Summary data has a real value to the organization
* Historical data holds the key to understanding data over time
* What-if capabilities are required

**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 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

**OLAP:** Online Analytic Processing

**OLAP: Multidimensional data model**

**Collection of numeric measures which depend on a set of dimensions**

**Operations in multidimensional data model：**

* **Aggregation**
* **Selection**
* **Navigation**
* **Visualization**

**OLAP queries**

* **Drill-down**
* **Pivoting**

**Hadoop and MapReduce**

**What**

Hadoop是项目的总称。主要是由HDFS和MapReduce组成。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算.

Hadoop是一种分布式数据和计算的框架。它很擅长存储大量的半结构化的数据集。数据可以随机存放，所以一个磁盘的失败并不会带来数据丢失。Hadoop也非常擅长分布式计算——快速地跨多台机器处理大型数据集合。

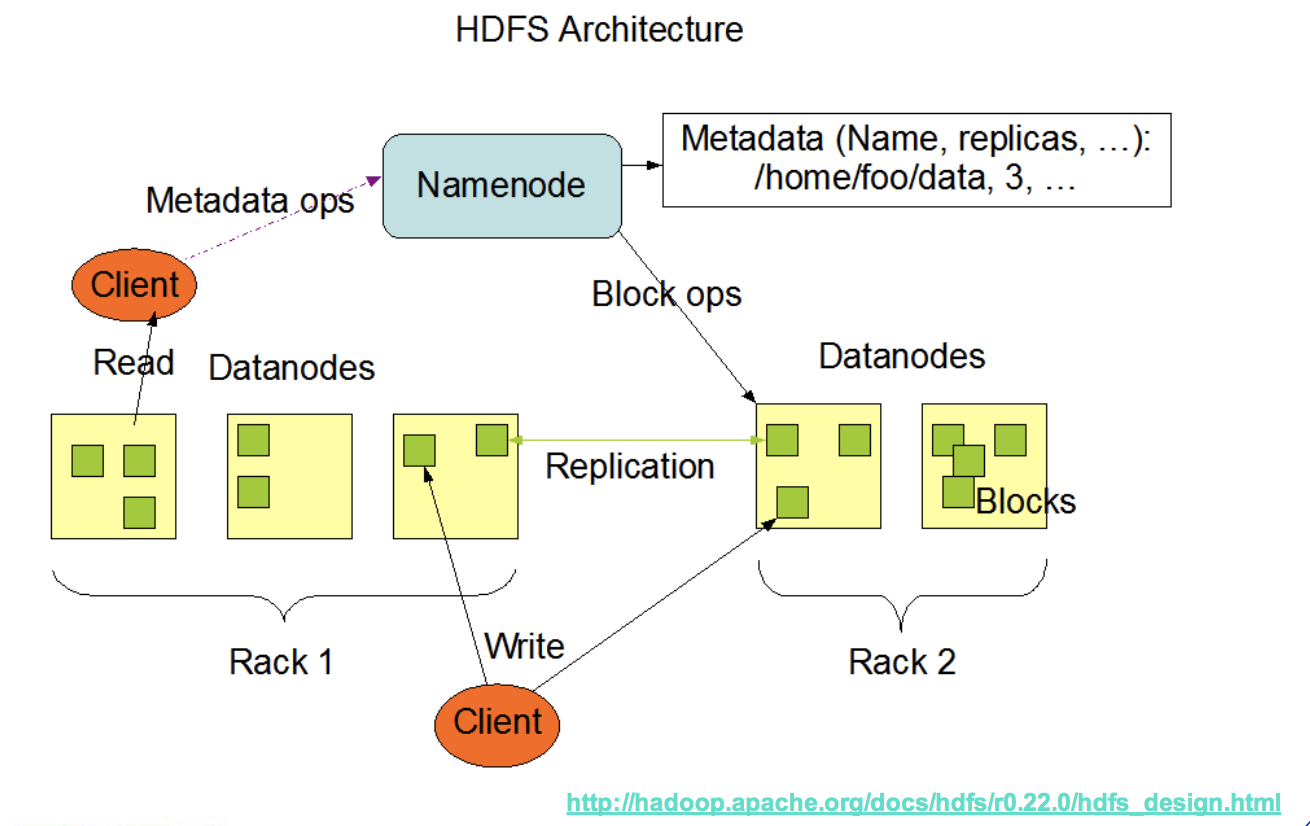
MapReduce是处理大量半结构化数据集合的编程模型。编程模型是一种处理并结构化特定问题的方式。例如，在一个关系数据库中，使用一种集合语言执行查询，如SQL。告诉语言想要的结果，并将它提交给系统来计算出如何产生计算。还可以用更传统的语言(C++，Java)，一步步地来解决问题。这是两种不同的编程模型，MapReduce就是另外一种。

**Subprojects overview**

**Assumptions and goals:**

* **Hardware failure**
* **Streaming data access**
* **Large data sets**
* **Simple coherency model for files**
* **“Moving computation is cheaper than moving data”**
* **Portability across heterogeneous hardware and software platforms**

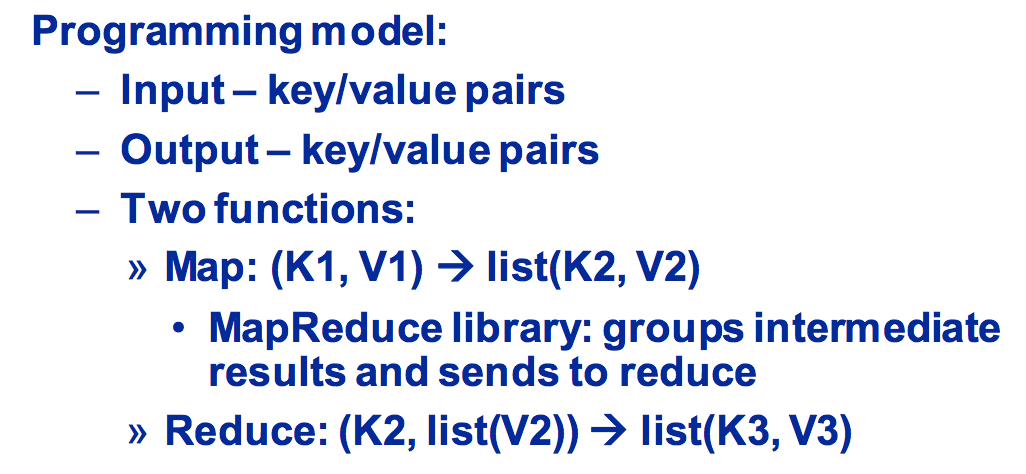
HDFS Architecture



内部机制是将一个文件分割成一个或多个块，这些块被存储在一组数据节点中。名字节点用来操作文件命名空间的文件或目录操作，如打开，关闭，重命名等等。它同时确定块与数据节点的映射。数据节点负责来自文件系统客户的读写请求。数据节点同时还要执行块的创建，删除，和来自名字节点的块复制指令。

MapReduce

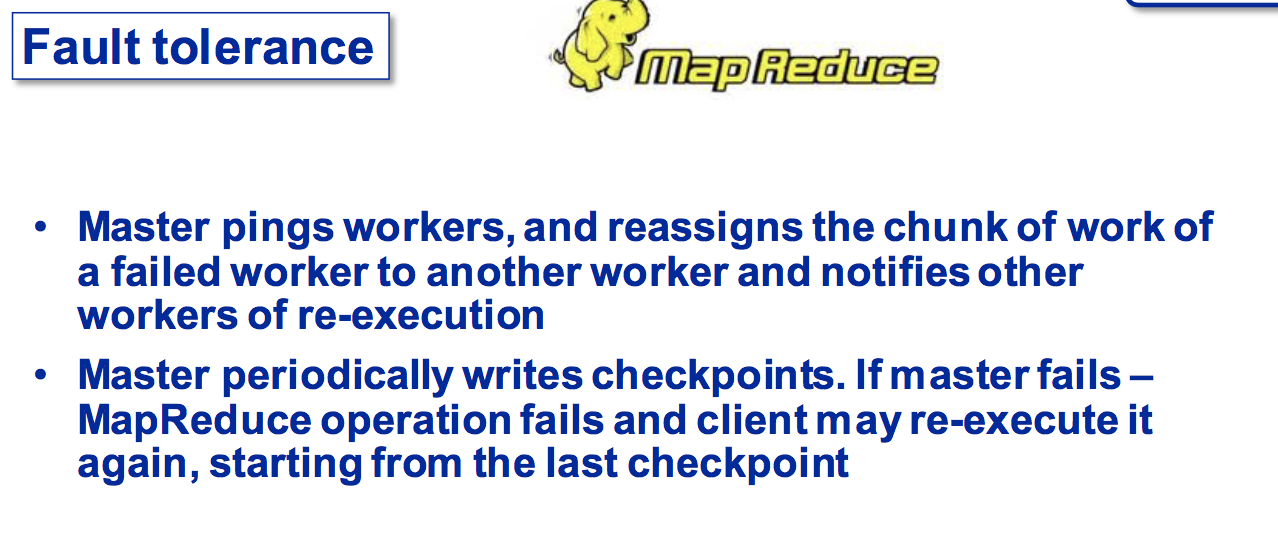
* Automatic parallel execution, fault tolerance, load balancing
* Run-timer takes care of failing nodes, data partitioning, result merging
* Runs on huge cluster of commodity machines
* Primitive operations: split the data, process them separately, combine the result



Example

MapReduce Data flow

Fault tolerance:



Pig

• Created at Yahoo!

• Higher abstraction layer

• Richer data structures

• Set of data transformations

• Extensible

• Designed for batch processing

• 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.

Example（**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)

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

example