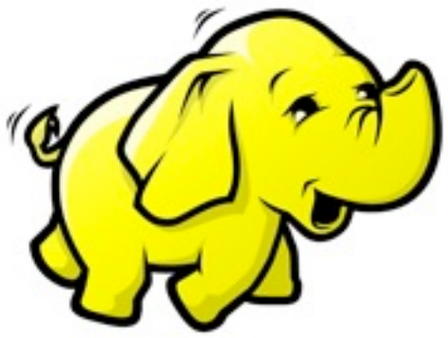


A brief of Hadoop

马士华
@mashihua

Hadoop是什么

- 一个开源的Apache Top Project
- 实现Google的GFS文件系统和MapReduce模型
- Hadoop子项目中的Google系统架构中核心要素实现

★ 实现了Google的BigTable模型（HBase和Hypertable）

★ 实现Google的Sawzall (Pig)

★ 实现Google的Chubby（ZooKeeper）

★ FaceBook实现了一个数据仓库（Hive）

★ Machine learning library（Mahout）

History

- Nutch代码的一部分（2004-2006）
- Y!聘请了Doug Cutting组建14人的团队（2006）
- 成为Apache TLP Lucene的子项目（2006-2008）
- 成为Apache TLP项目（2008）

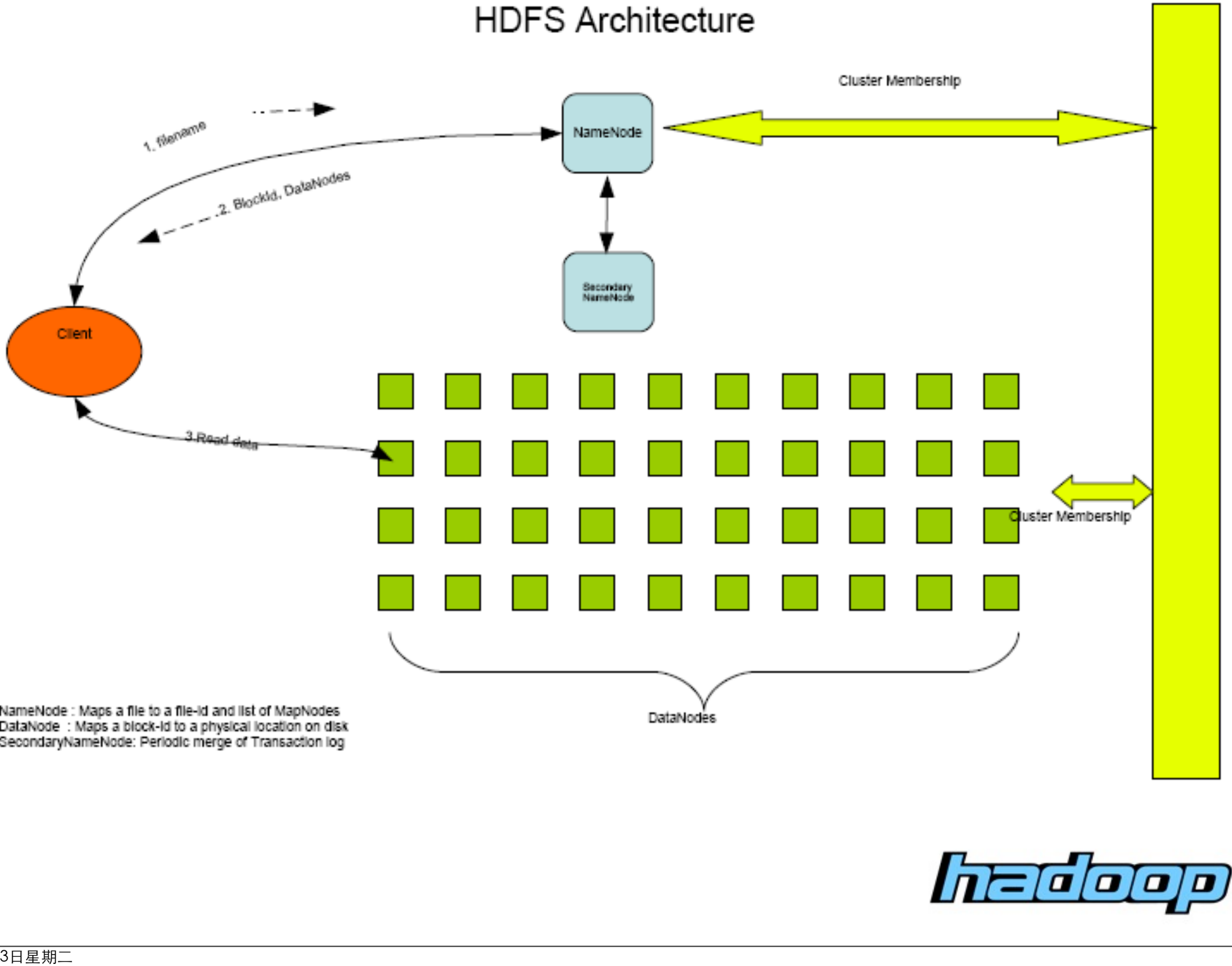
The Problem

- 数据日益变得庞大
- 多久能够遍历1T的数据?
- 多久能够更新1T的数据?
- 需要一个infrastructure来存储数据和处理数据

HDFS特性

- 数据的错误检测和快速、自动的恢复
- 流式数据访问，提高数据访问的高吞吐量
- 大规模数据集，适合G-T级的大文件
- 一次写入多次读取的文件访问模型
- Unix下的平台的可移植性
- 流式的数据复制

HDFS Architecture



NameNode : Maps a file to a file-Id and list of MapNodes
DataNode : Maps a block-Id to a physical location on disk
SecondaryNameNode: Periodic merge of Transaction log

HDFS结构

- NameNode负责集群中元数据信息的保存, DataNode节点的有效性,集群配置信息，Block的复制。

  ✴ NameNode元数据信息包括：文件名，副本数，BlockID，对应的DataNode节点，文件属性，ACL等。

- SecondName负责周期性的帮助NameNode检查和合并元数据FsImage。

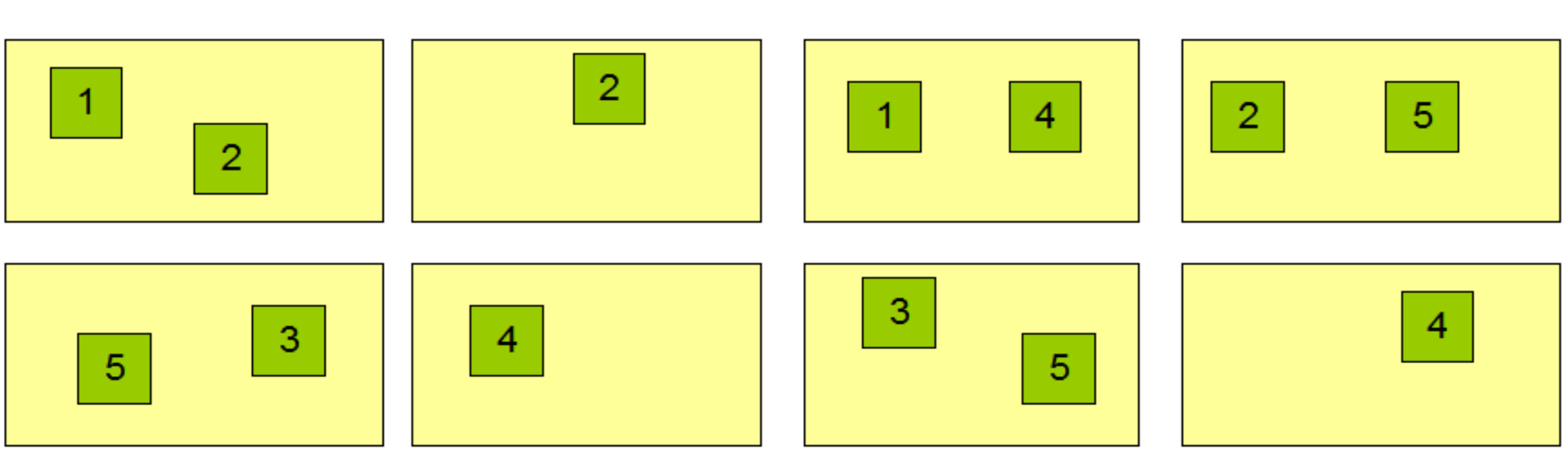
- DataNode存贮实际的Block，周期性的向NameNode发送块状态报告 。响应客户端的数据请求。

NameNode元数据

Block Replication

Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes



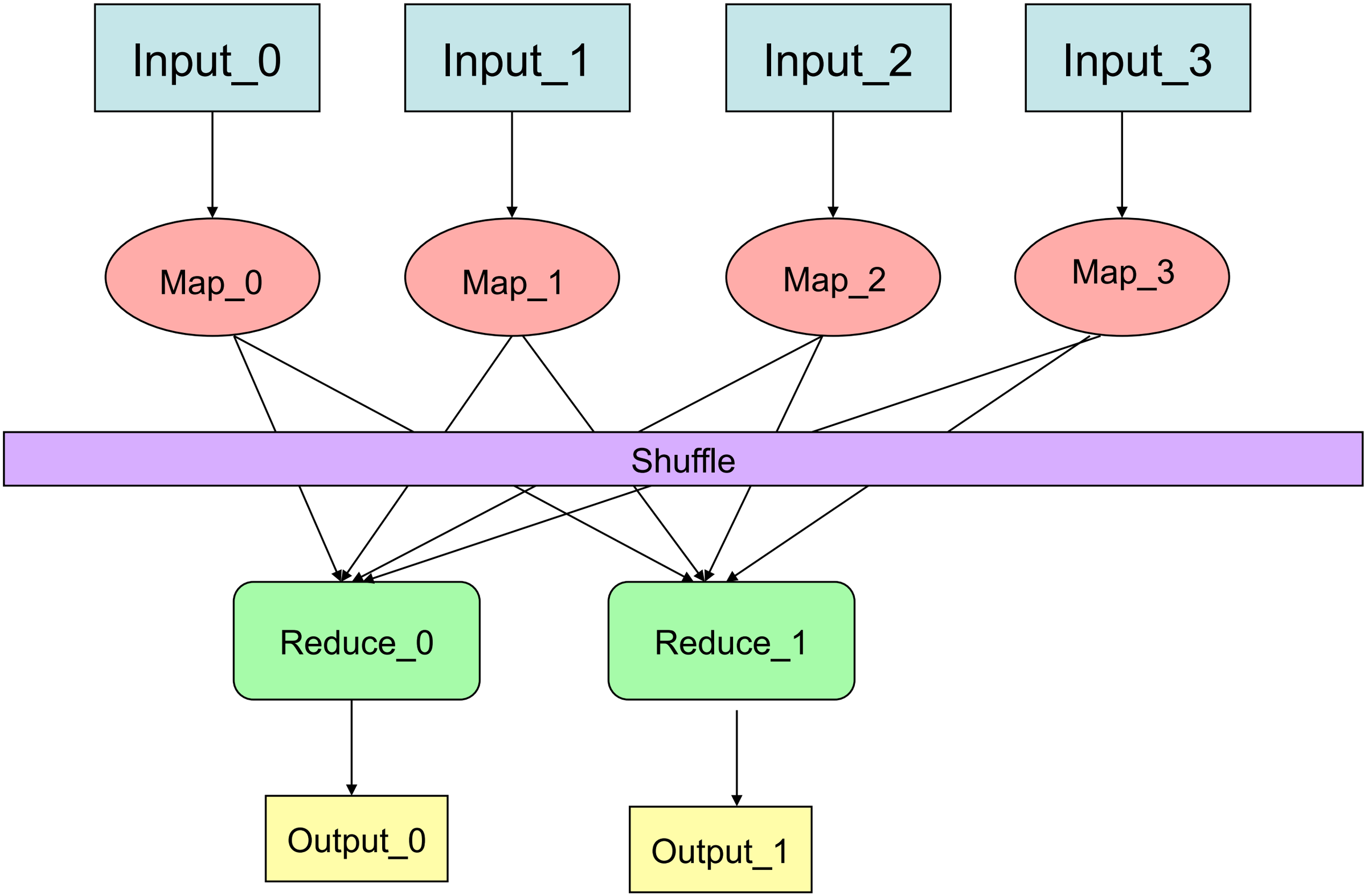
hadoop

HDFS数据

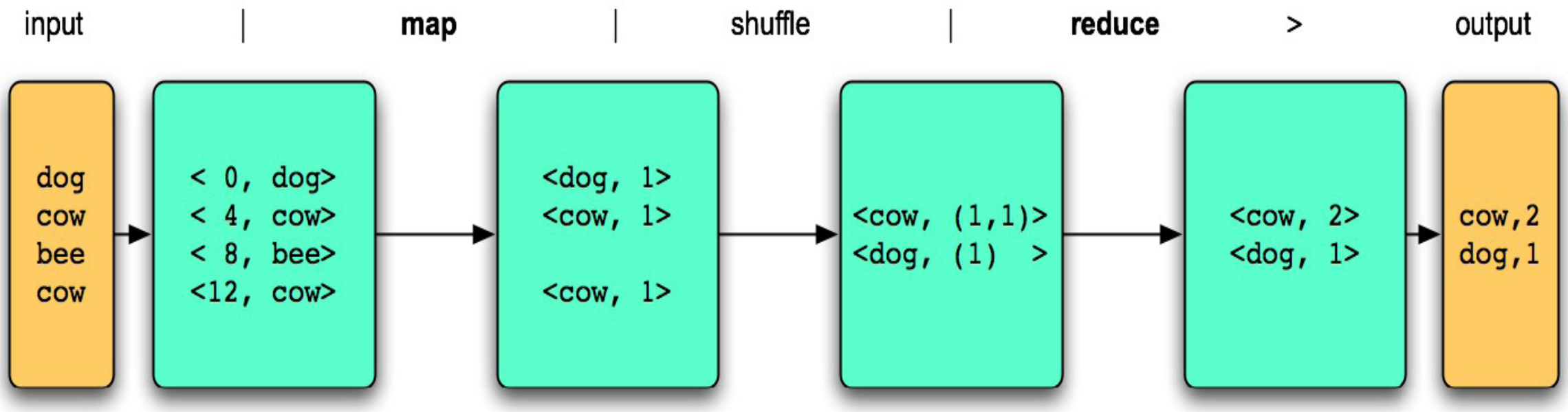
- 数据正确性
  - ✤ 使用CRC32验证。
- 文件创建:
  - ✤ 客户端每512byte计算一次CRC32
  - ✤ DataNode存储这些checksum信息
- 文件访问:
  - ✤ 客户端从NameNode获取元数据信息
  - ✤ 从DataNode中得到checksum和数据进行验证
  - ✤ 如果验证出错，获取另外的复制块

MapReduce

- Unix 命令:
  - cat input | grep | sort | uniq -c > output
- MapReduce模型:
  - Input | Map | Shuffle & Sort | Reduce | Output


Input_0
Input_1
Input_2
Input_3
Map_0
Map_1
Map_2
Map_3
Shuffle
Reduce_0
Reduce_1
Output_0
Output_1

MapReduce




```
cat *  |  grep -Eh '(cow|dog)' |  sort  |  uniq -c  >  output
```

MapReduce Code

```
public void map(LongWritable key, Text val,
    OutputCollector<Text, IntWritable> output,Reporter reporter) throws
    IOException {

    if (pattern.matcher(val.toString()).matches()) {
          output.collect(val, new IntWritable(1));
          }
}

public void reduce(Text key, Iterator<IntWritable> vals,
    OutputCollector<Text, IntWritable> output, Reporter reporter) throws
    IOException {

    int sum = 0;
    while (vals.hasNext()) {
          sum += vals.next().get();
    }
    output.collect(key, new IntWritable(sum));
}
```

MapReduce特性

- 面向成批处理，非Online访问
- Ad hoc queries
- 框架处理分布式
- High-Ordered Function
- 简单模型：Key-Value

MapReduce

- 单节点Job Tracker
  - ✦接受Job提交
  - ✦分派Job的Map和Reduce的Job
  - ✦打包Job的Task到Task Tracker
  - ✦重新调度出错的Task
- 多个Task Tracker
  - ✦启动子VM运行task
  - ✦Job Tracker报告Task的运行情况

Hadoop现实

- Powered By Hadoop有80家公司
- Yahoo! Search Webmap—2008.2

★页面之间的链接数超过1000亿

★Webmap输出的压缩数据超过300TB

★有单一的MapReduce任务同时在1万多个CPU的核（core）上运行

★用于生产集群硬盘空间占用超过5PB

Thanks