MapReduce & Hadoop

技术、原理及应用

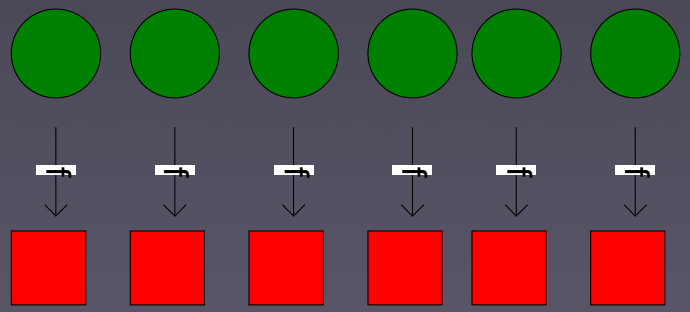
Jeremy Chow(coderplay@gmail.com)

问题描述

- 大规模数据集的挑战
- 摩尔定律已经失效
- 当前的并行技术
- 数据分布
- 集群系统的伸缩性
- 其它

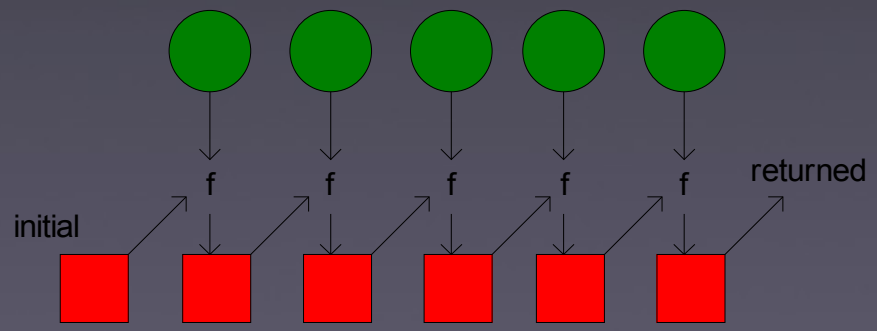
Map

对列表中的每个元素应用 f 函数，并按顺序返回结果列表



Fold

遍历列表，列表每个元素与一累加器应用 f 函数． 函数 f 返回下一个累加器的值，此值和列表的下一元素组合起来应用于 f.



MapReduce

- 从数据源获取的数据（文件，数据表等）形成 <key, value> 对输入到 map() 函数中
- map() 输出中间结果
- map 过程结束后，所有相同 key 值的中间 values 组合成一个列表
- reduce() 根据 <key,values> 产生结果集
- 在实际中，经常是单个 key 对应单个 value.

WordCount 实例

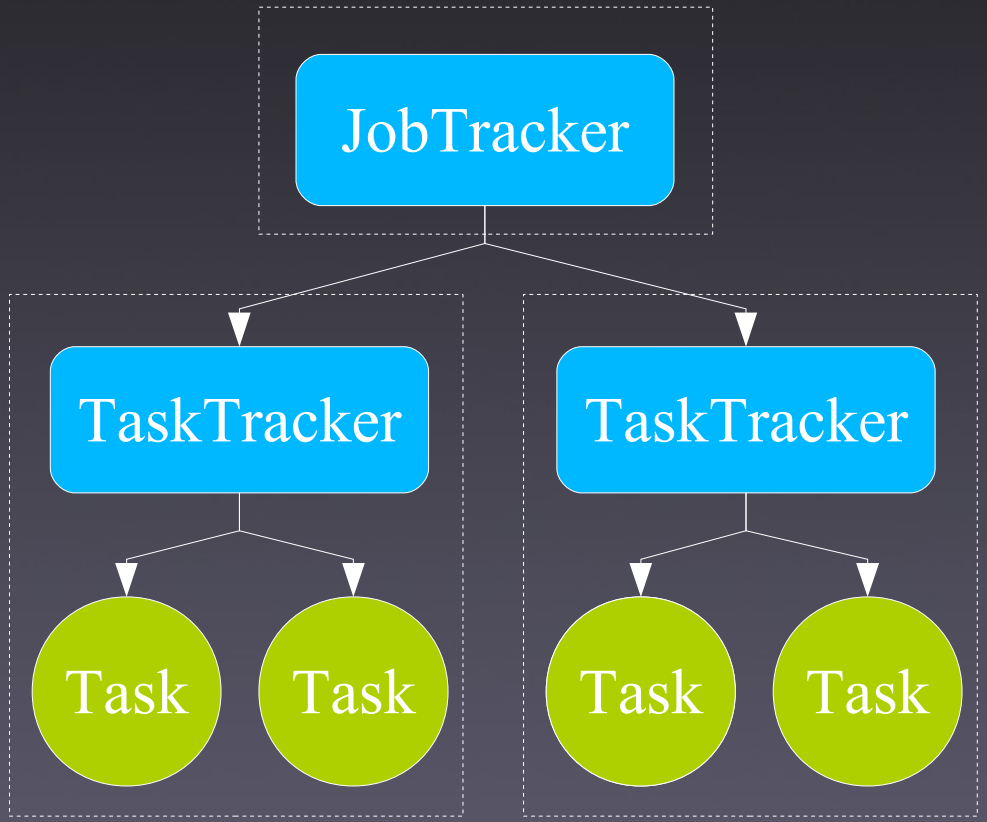
```
map(String input_key, String input_value):
  // input_key: 文档名
  // input_value: 文档内容
  for each word w in input_value:
    EmitIntermediate(w, "1");


reduce(String output_key, Iterator intermediate_values):
  // output_key: 一单词
  // output_values: 计数列表
  int result = 0;
  for each v in intermediate_values:
    result += ParseInt(v);
 Emit(AsString(result));
```

Hadoop

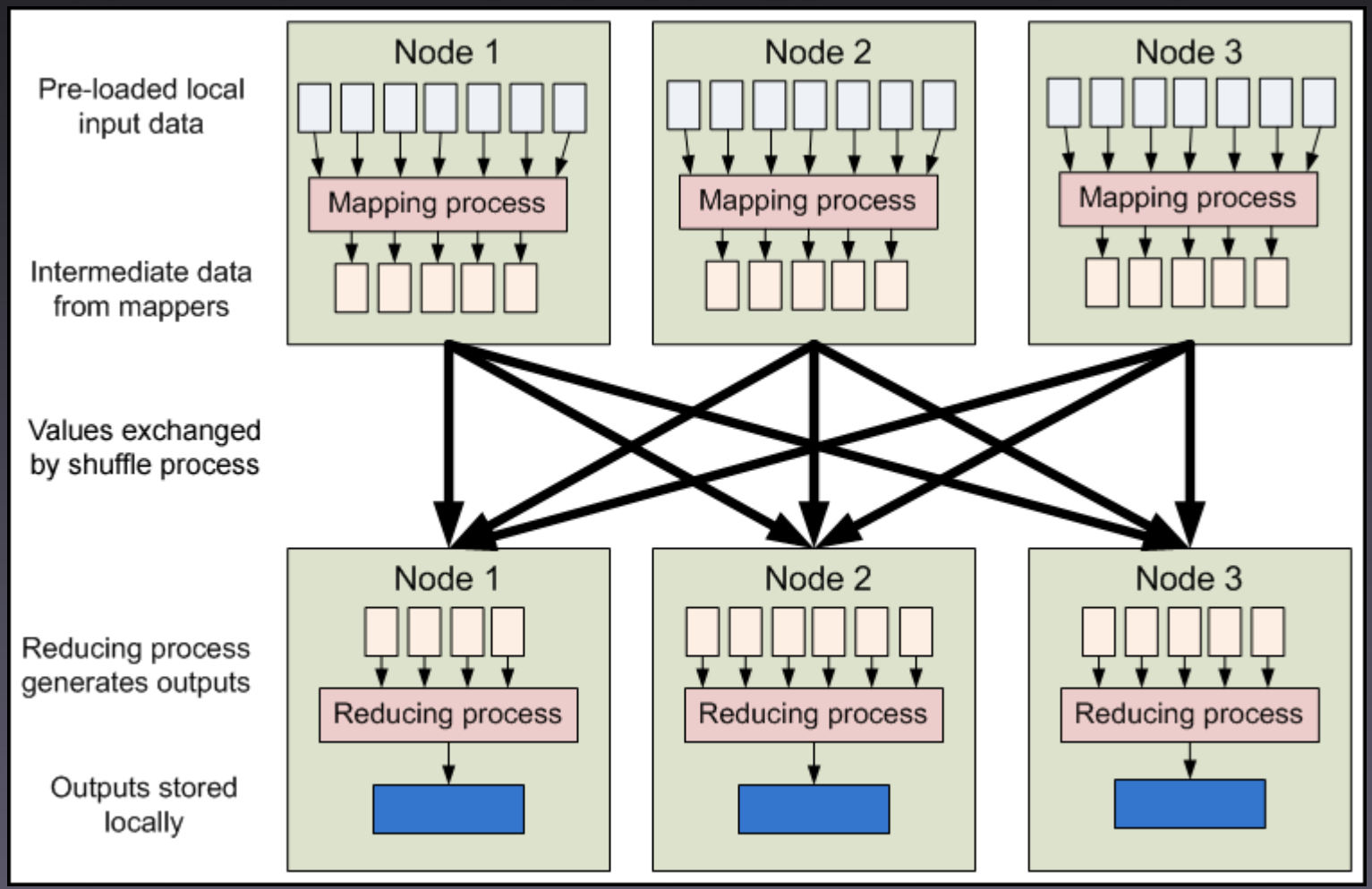
- 可伸缩：1000 节点以上
- 经济： 使用普通机器
- 可靠
- 高效
- 开源

Hadoop 机制

- JobConf
- JobClient
- JobTracker
- TaskTracker
- Task



Hadoop 处理流程

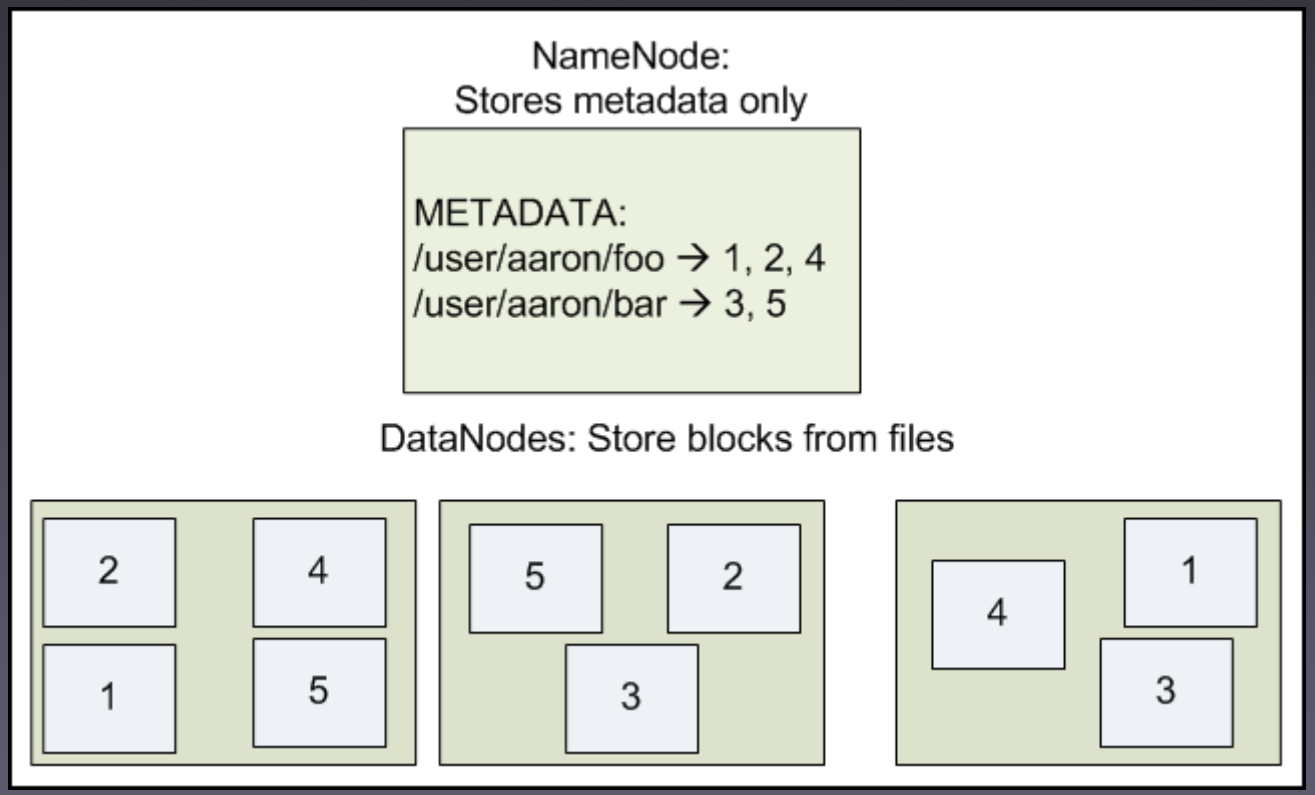

Pre-loaded local input data
Intermediate data from mappers
Values exchanged by shuffle process
Reducing process generates outputs
Outputs stored locally
Node 1
Mapping process
Node 2
Mapping process
Node 3
Mapping process
Node 1
Reducing process
Node 2
Reducing process
Node 3
Reducing process

Hadoop 的一些接口

- Mapper , Reducer, Combiner
- 自定义数据类型
- 自定义输入格式 InputFormat
- 自定义输出格式 OutputFormat
- 数据分区 Patitioner
- 辅助数据的分布
- 任务配置 JobConf

HDFS 分布式文件系统

- 名称节点
- 数据节点



案例及相关项目

- Nutch 网页抓取索引及搜索
- Hbase 分布式数据库
- Hive 数据仓库
- Mathout
- Zoopkeeper
- Pig
- Hama
- 云计算

案例： Redpoll

- 什么是 Redpoll
- 什么人会对 Redpoll 感兴趣
- 聚类
- 分类
- 文本数据挖掘
- PageRank, LSI, etc

谢 谢！