



大数据导论

Introduction to Big Data



第2章：大数据的分布式存储与处理

—— 以Hadoop为例

叶允明

计算机科学与技术学院

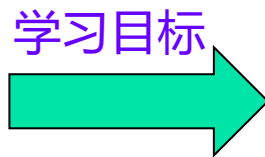
哈尔滨工业大学（深圳）

大数据分布式存储与处理部分的授课安排

- Hadoop入门知识

✓ 基础概念和思想

- 深入理解HDFS



✓ 系统架构设计能力

- Hadoop应用案例实践

✓ 应用开源工具的能力

主要参考资料

- 林子雨.《大数据技术原理与应用(第2版)》.人民邮电出版社, 2017.
- 第2、3、7、8章

(一) Hadoop入门

- 引例：Web搜索引擎
- Hadoop的基础架构
- Hadoop平台搭建示例

引例：Web搜索引擎

——理解大数据存储与处理面临的技术问题



大数据 hadoop



全部

图片

新闻

视频

地图

更多

工具

找到约 19,800,000 条结果 (用时 0.57 秒)

<https://zhuanlan.zhihu.com> > ... ▼

深入浅出大数据：到底什么是Hadoop？ - 知乎专栏

2019年1月15日 — 深入浅出**大数据**：到底什么是**Hadoop**？ 2年前·来自专栏鲜枣课堂·1998年9月4日，Google公司在美国硅谷成立。正如大家所知，它是一家做搜索引擎起家的 ...

[https://www.zhihu.com](https://www.zhihu.com/question) > question ▼

hadoop和大数据的关系？和spark的关系？ - 知乎

2015年11月25日 — Pig：是一个基于**Hadoop**的大规模**数据**分析工具，它提供的SQL-LIKE语言叫Pig Latin，该语言的编译器会把类SQL的**数据**分析请求转换为一系列经过优化处理的MapReduc...
34 个回答 · 最佳答案：1998年9月4日，Google公司在美国硅谷成立。正如大家所知，它是一家...

Hadoop到底是干什么用的？ 16 个回答 2019年7月9日

大数据方向除了**Hadoop**还有什么可学的？ 18 个回答 2015年12月10日

为什么很多公司的大数 ... 57 个回答 2015年9月22日

请问**大数据**中**Hadoop**的核心技术是什么？ 14 个回答 2019年10月30日

www.zhihu.com站内的其它相关信息

<https://www.huaweicloud.com/articles> ▼

大数据代表技术：Hadoop、Spark、Flink、Beam - 华为云

2021年2月5日 — **大数据**代表技术：**Hadoop**、Spark、Flink、Beam **Hadoop**：从2005年到2015年，说到**大数据**都是讲**hadoop**。**Hadoop**是一整套的技术框架，不是一个单一软件， ...

搜索引擎：网络文档集合的检索器



HTML源文件

```
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      
      <a class="navbar-brand" href="/"> Apache Hadoop</a>
    </div>

    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
```

.....



维基百科
自由的百科全书

首页
分类索引
特色内容
新闻动态
最近更改
随机条目
资助维基百科

帮助
帮助
维基社群
方针与指引

条目 讨论 大陆简体 汉 汉

维基台北写作聚于每月第二个

Apache Hadoop [编辑]

维基百科，自由的百科全书



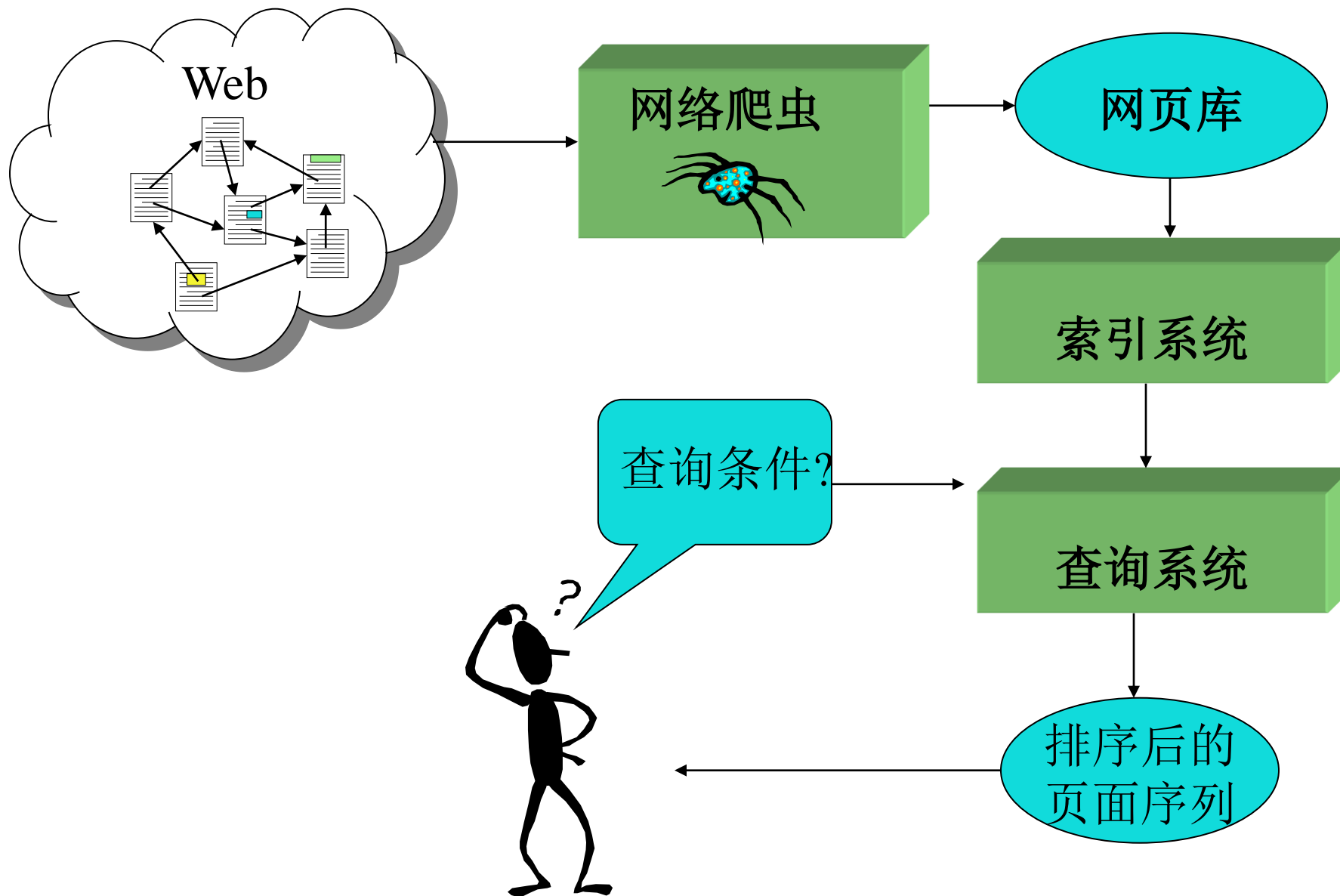
此条目的引用需要进行清理，
参考文献应符合正确的引用、脚注

Apache Hadoop是一款支持数据密集型分布式应用
Apache 2.0许可协议发布的**开源软件框架**。它支持在
大型集群上运行的应用程序。Hadoop是根据**谷歌公**
MapReduce和**Google文件系统的**论文自行实现而成
Hadoop模块都有一个基本假设，即硬件故障是常见
架自动处理。



```
<!DOCTYPE html>
<html class="client-nojs" lang="zh-Hans-CN" dir="ltr"
<head>
<meta charset="UTF-8"/>
<title>Apache Hadoop - 维基百科，自由的百科全书</title>
<script>document.documentElement.className="client-js";
"wgPageContentModel":"wikitext","wgRelevantPageName":
"user":"ready","user.options":"loading","ext.cite.styl
"ext.visualEditor.desktopArticleTarget.init","ext.vis
<script>(RLQ=window.RLQ||[]).push(function(){mw.load
});});</script>
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<script async="" src="/w/load.php?lang=zh-cn&amp;mod
<meta name="ResourceLoaderDynamicStyles" content=""/>
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<meta name="generator" content="MediaWiki 1.38.0-wmf.
```

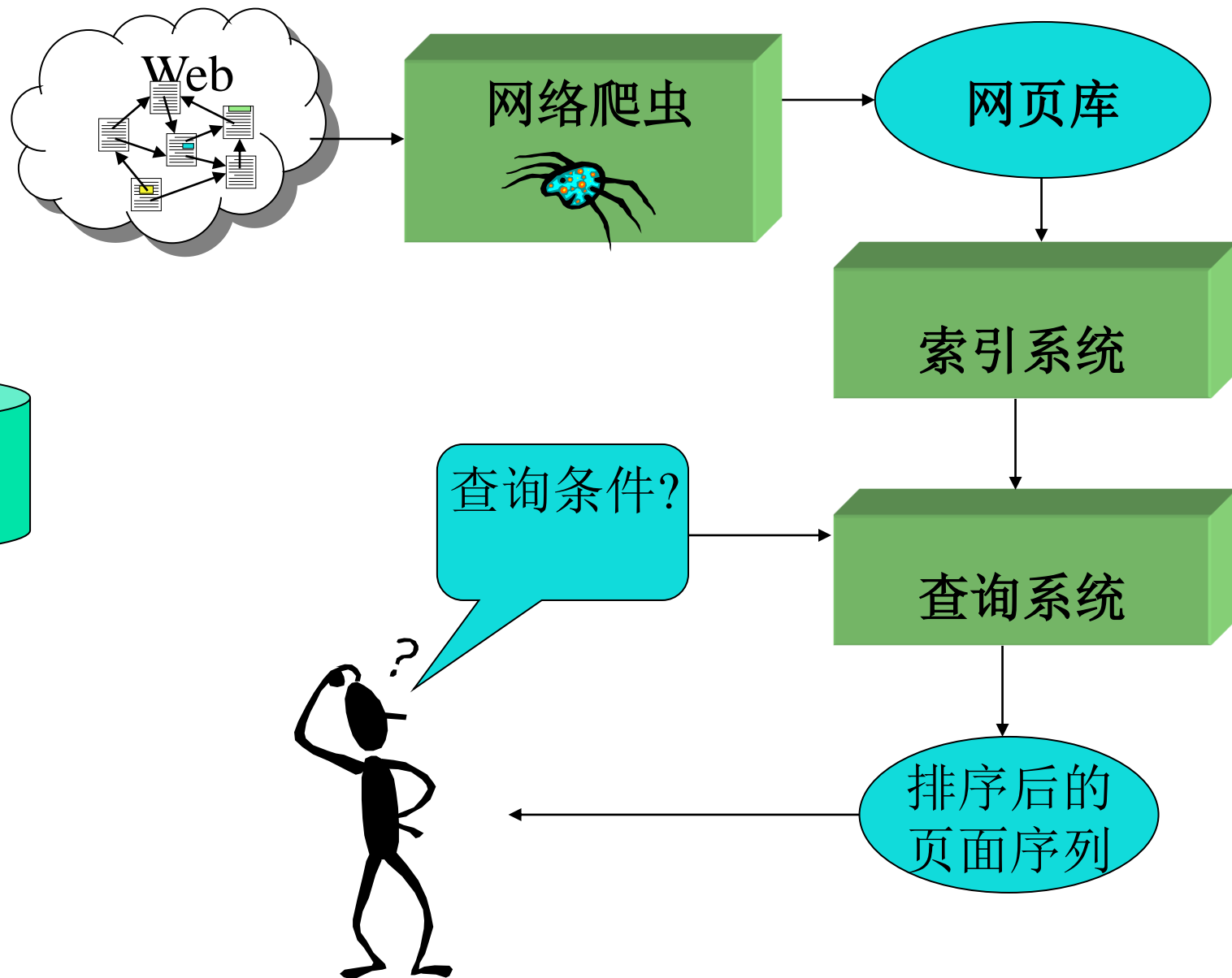
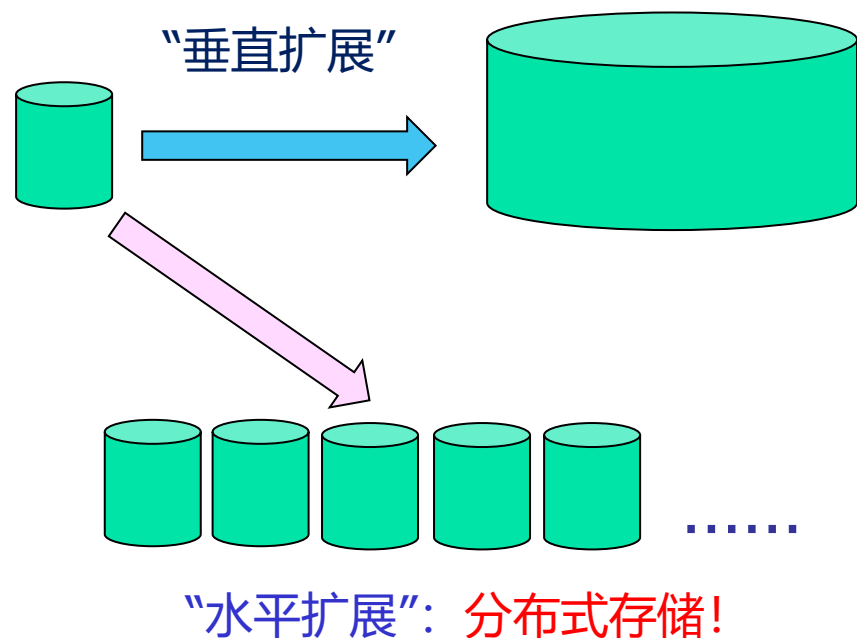
Web搜索引擎的基本原理



Web搜索引擎的大数据存储与处理问题

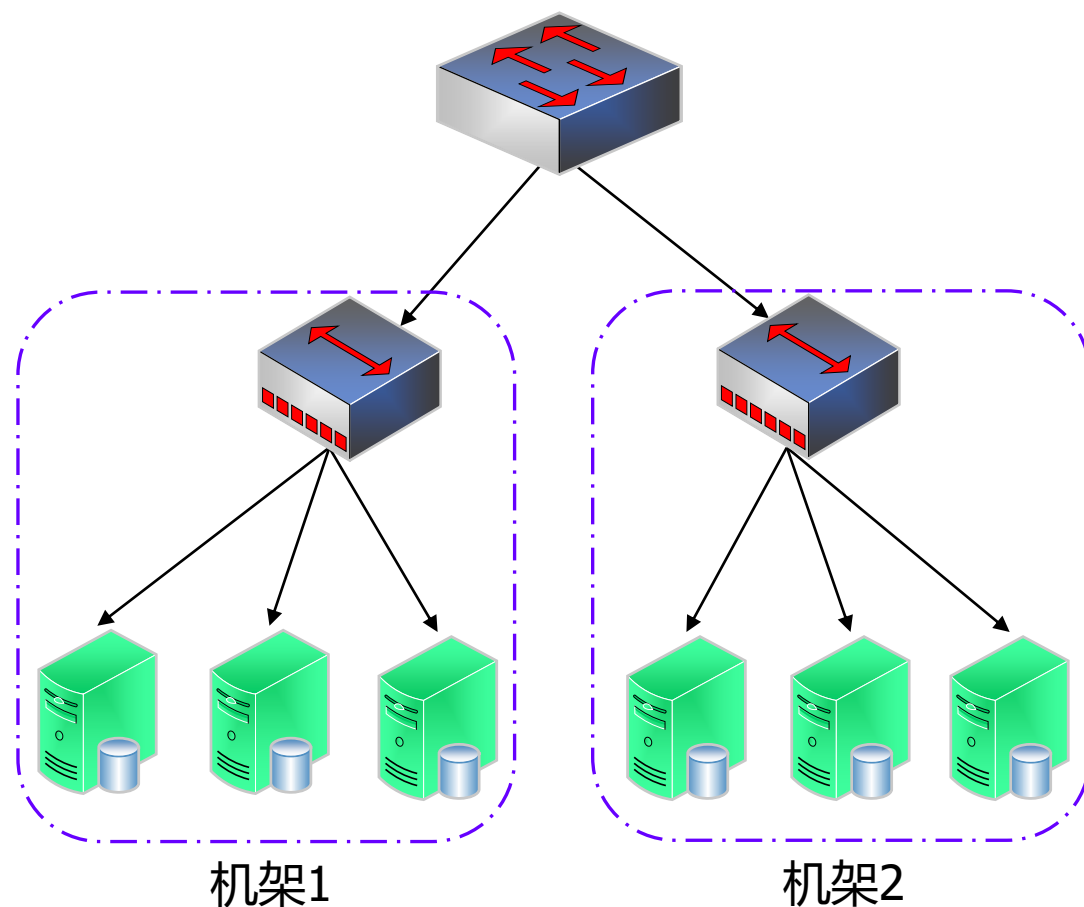
- 1万亿个网页

- 问题1：怎么快速存储？



大数据分布式存储的主要技术问题分析

- 数据怎么分布？
- 存取性能如何？
 - 包括并发读、写
- 硬件故障怎么办？
 - 磁盘故障、其它硬件和网络故障
- 系统的访问接口是否简单易用？
- 硬件性能需求及成本如何？

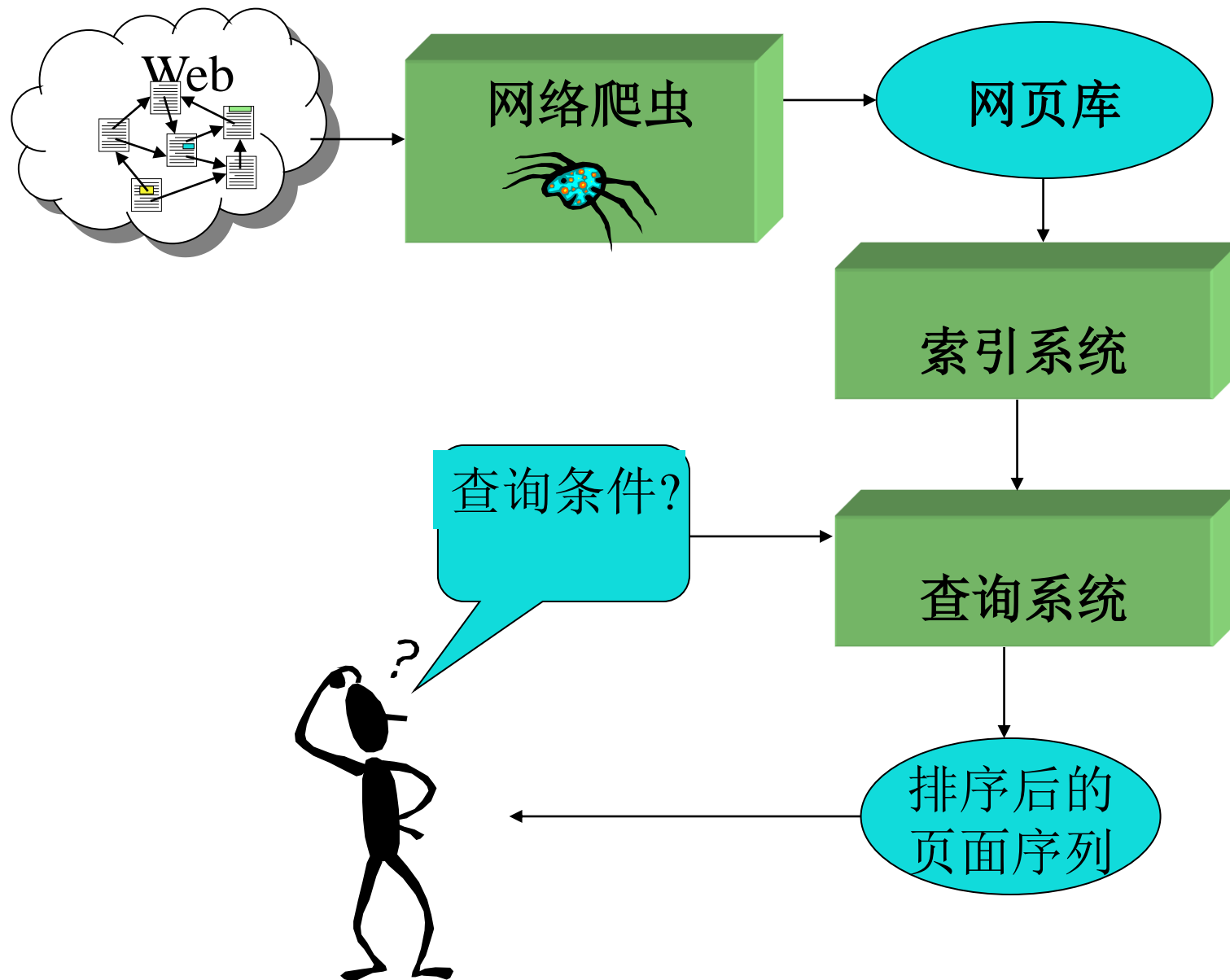


Web搜索引擎的大数据挑战

- 1万亿个网页

- 问题1：怎么快速存储？

- 问题2：怎么快速检索？



问题2：怎么快速检索？

```
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed"
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      
      <a class="navbar-brand" href="/"> Apache Hadoop</a>
    </div>

    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
```

网页1的源代码文档

```
<!DOCTYPE html>
<html class="client-nojs" lang="zh-Hans-CN" dir="ltr"
<head>
<meta charset="UTF-8"/>
<title>Apache Hadoop - 维基百科，自由的百科全书</title>
<script>document.documentElement.className="client-js";
"wgPageContentModel":"wikitext","wgRelevantPageName":
"user":"ready","user.options":"loading","ext.cite.sty
"ext.visualEditor.desktopArticleTarget.init","ext.vis
<script>(RLQ=window.RLQ||[]).push(function(){mw.load
});});</script>
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<script async="" src="/w/load.php?lang=zh-cn&mod
<meta name="ResourceLoaderDynamicStyles" content=""/>
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<link rel="stylesheet" href="/w/load.php?lang=zh-cn&
<meta name="generator" content="MediaWiki 1.38.0-wmf.
```

网页2的源代码文档

快速检索模型：基于倒排索引（Inverted Index）

文档1:

Hadoop is open-source.

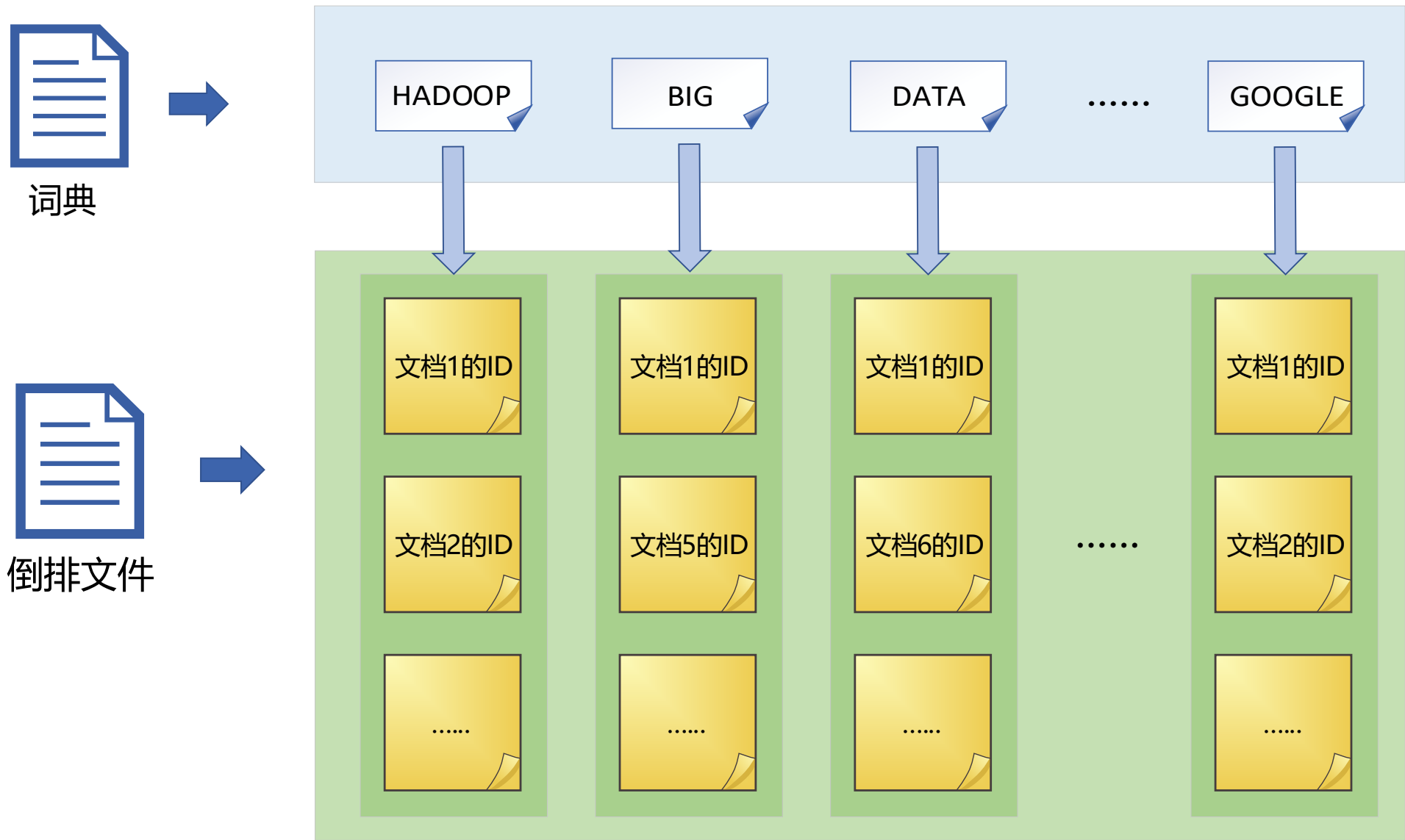
文档2:

**HDFS is a distributed
file system.**

文档3:

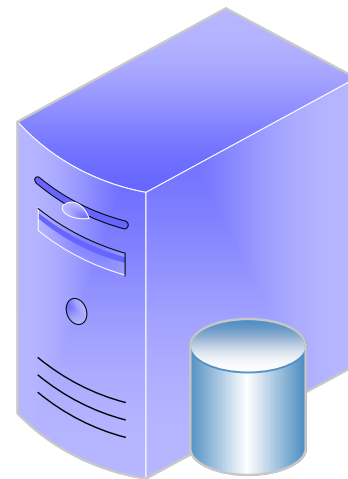
**A Hadoop system can
deliver high availability.**

倒排索引 (Inverted Index)

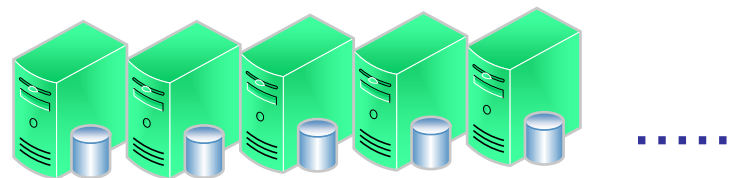


怎么快速构建倒排索引?

“垂直扩展”

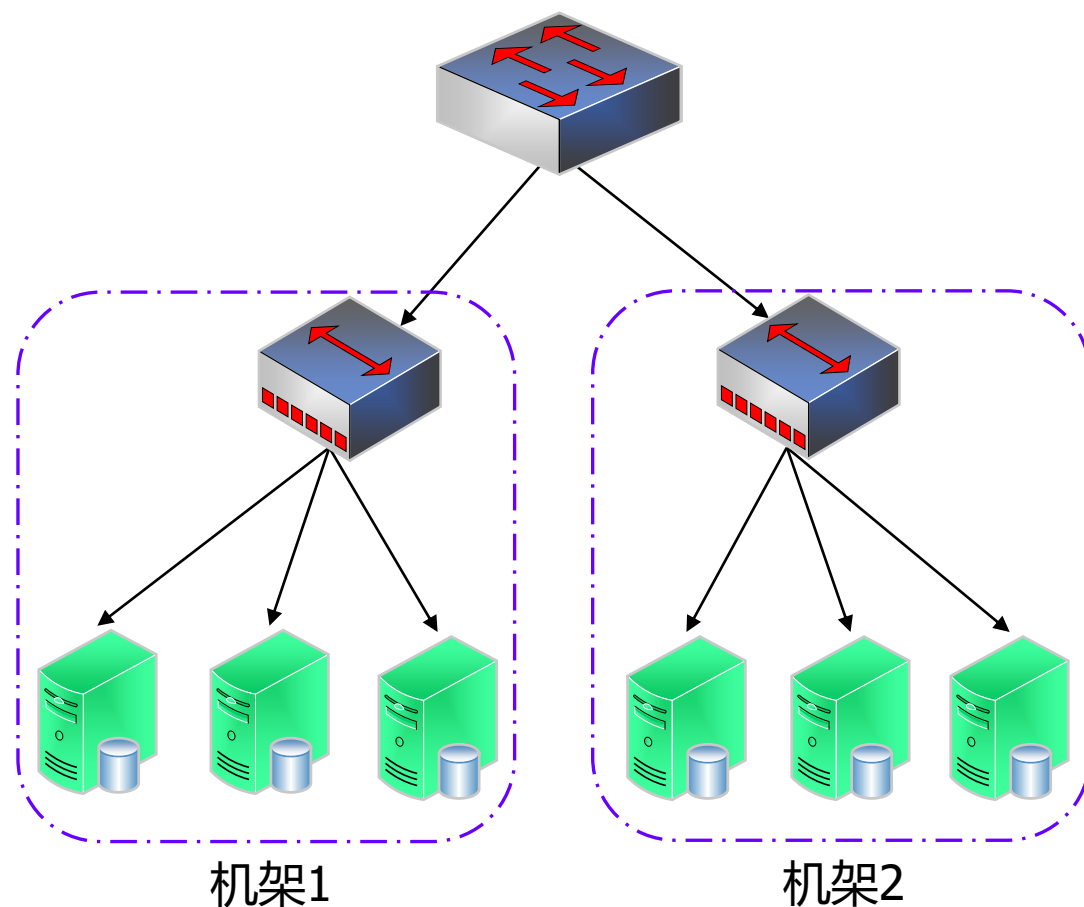


“水平扩展”：分布式处理！



大数据分布式处理的主要技术问题分析

- 计算任务（负载）如何分配？
- 分布式通信带来的额外开销问题？
 - 数据分发、处理结果收集
 - 大数据分布处理可带来的数据传输问题
- 硬件故障怎么办？
 - 节点故障、网络故障
- 系统的访问接口是否简单易用？
- 硬件性能需求及成本如何？



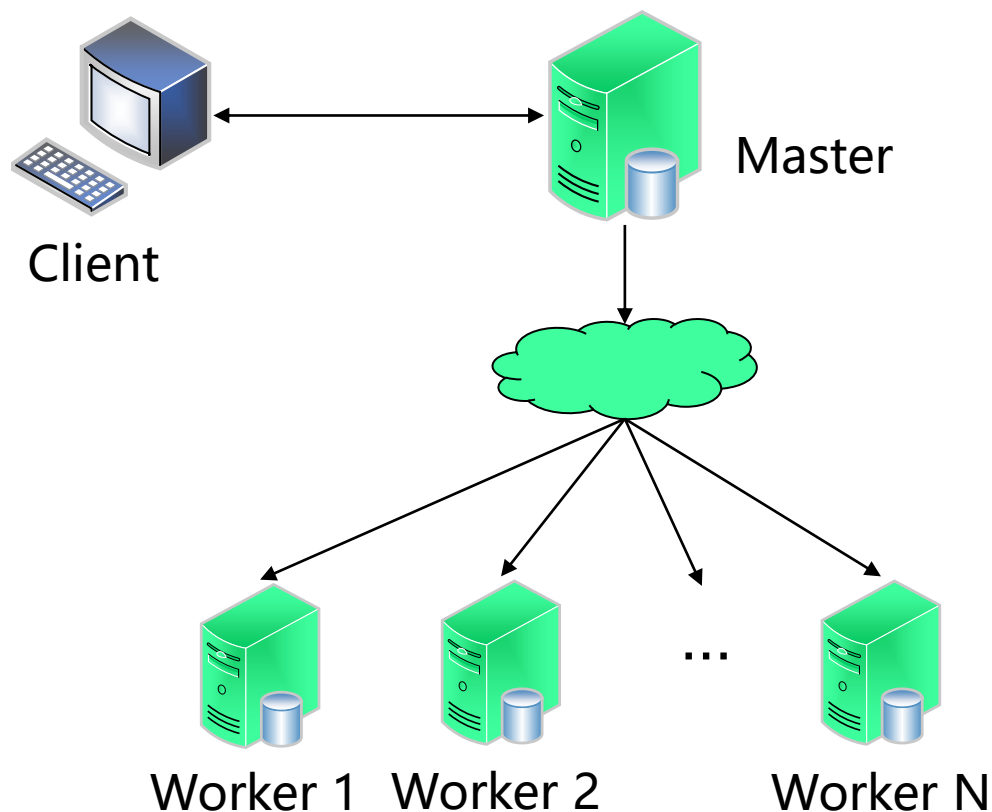
Hadoop基础架构概览

Hadoop的诞生

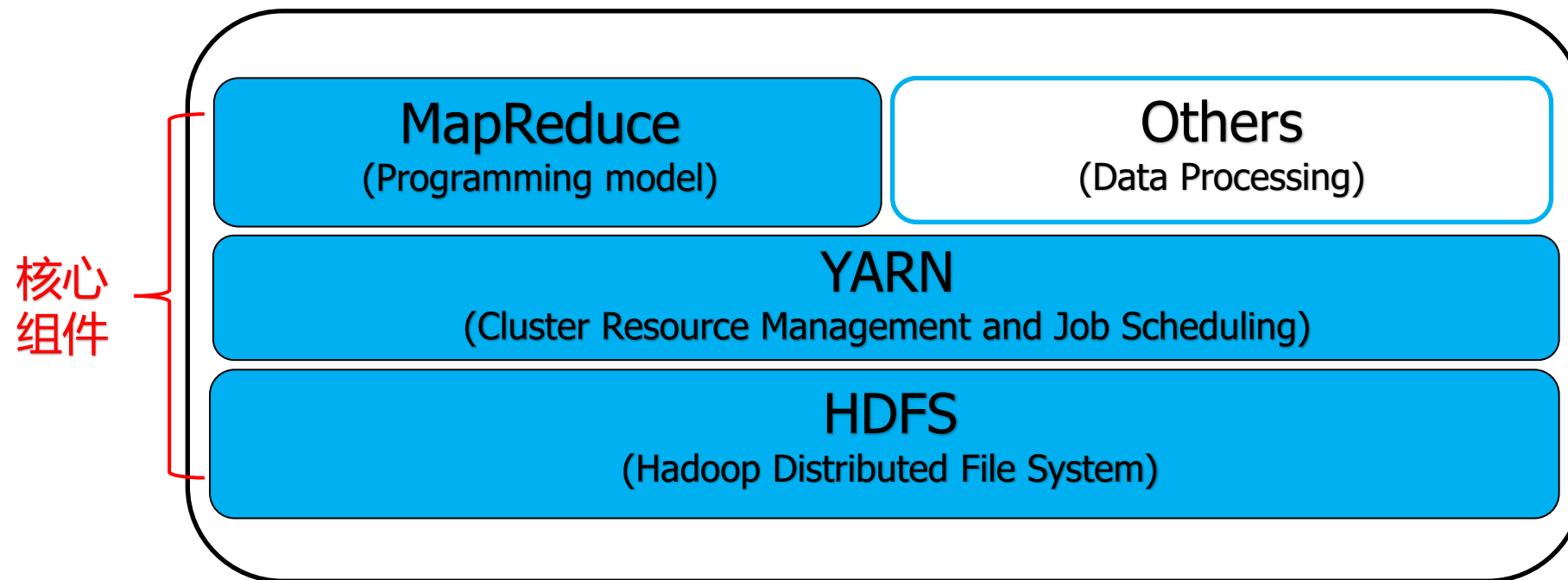
- 产生背景：为解决互联网时代的大数据存储与计算架构问题
 - 硬件故障问题、存储成本问题、快速计算问题.....
- Hadoop：实现高效数据存储、处理的一种分布式框架
 - 谷歌的GFS和MapReduce的开源实现版本
- Hadoop可以解决PB级别的数据存储与计算问题
 - $1\text{PB}=2^{10}\text{TB}=2^{20}\text{GB!}$
- Hadoop基于Java语言开发：具有很好的跨平台性
 - Hadoop 上的应用程序也可用其它语言编写，如C/C++

Hadoop分布式框架的基本思想

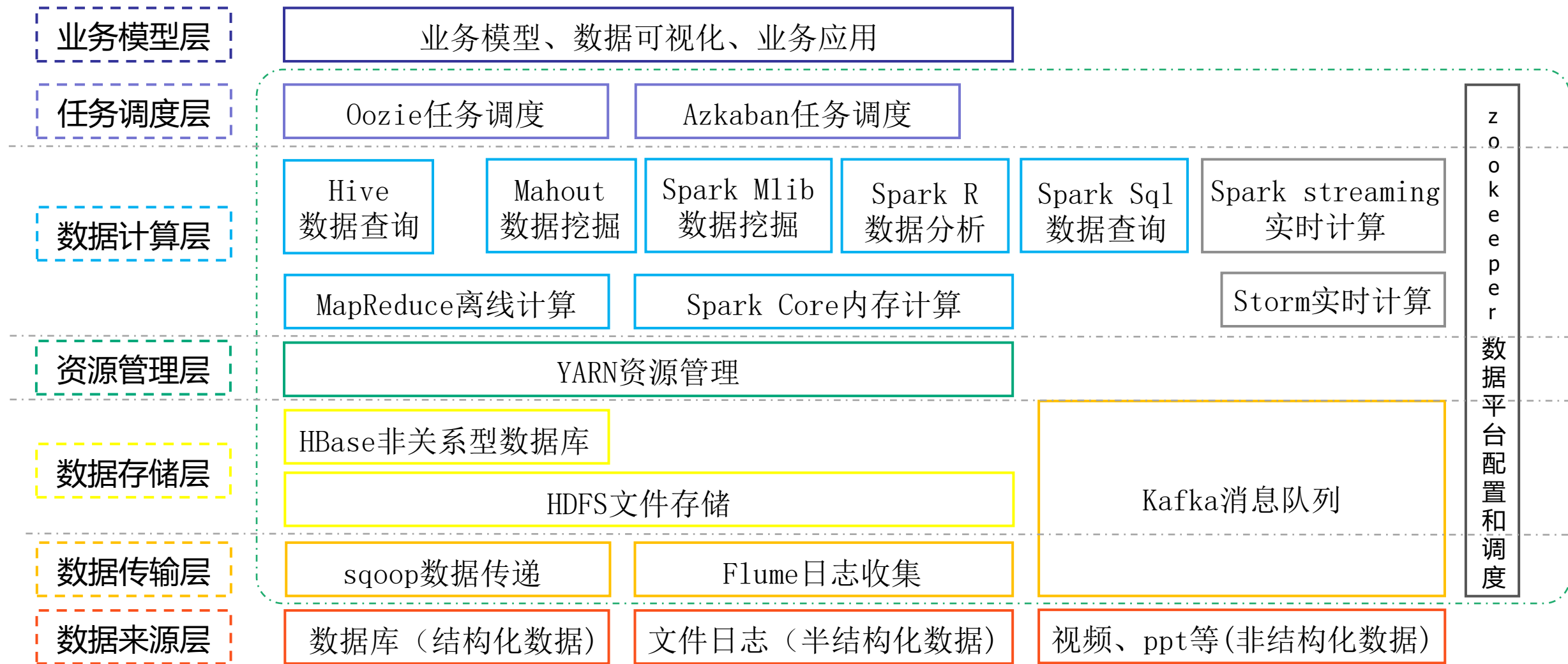
- Master-slave架构
- 分布式存储：HDFS
- 分布式计算：Mapreduce
- 存储与处理的一体化！



Hadoop系统的核心组件

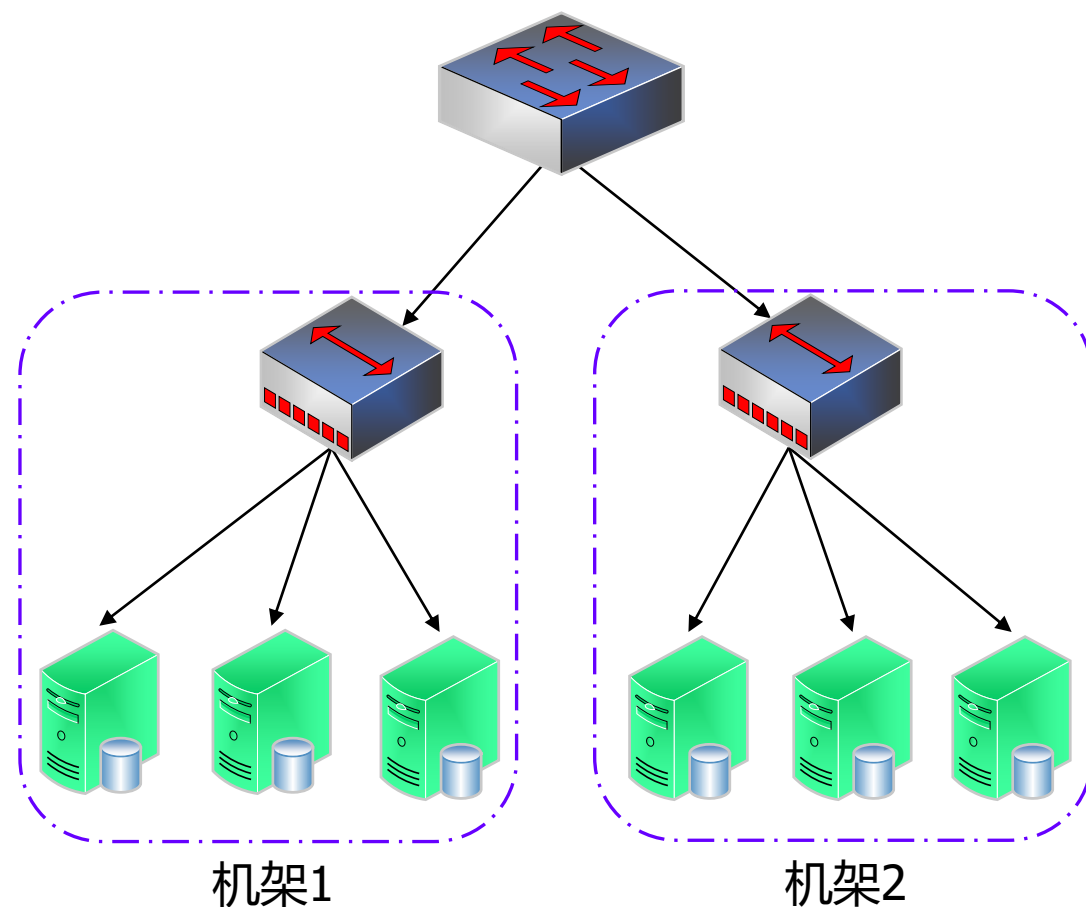


Hadoop生态



回顾：大数据分布式存储的主要技术问题分析

- 数据怎么分布？
- 存取性能如何？
 - 包括并发读、写
- 硬件故障怎么办？
 - 磁盘故障、其它硬件和网络故障
- 系统的访问接口是否简单易用？
- 硬件性能需求及成本如何？



HDFS简介

- “高容错、低成本的分布式大磁盘”，设计需求：

- 简单的文件访问模型：类似于linux的文件系统！

- PB级数据的可靠存储

- 流数据读写：高吞吐率

- 支持上万台服务器集群

- 对硬件设备性能要求低

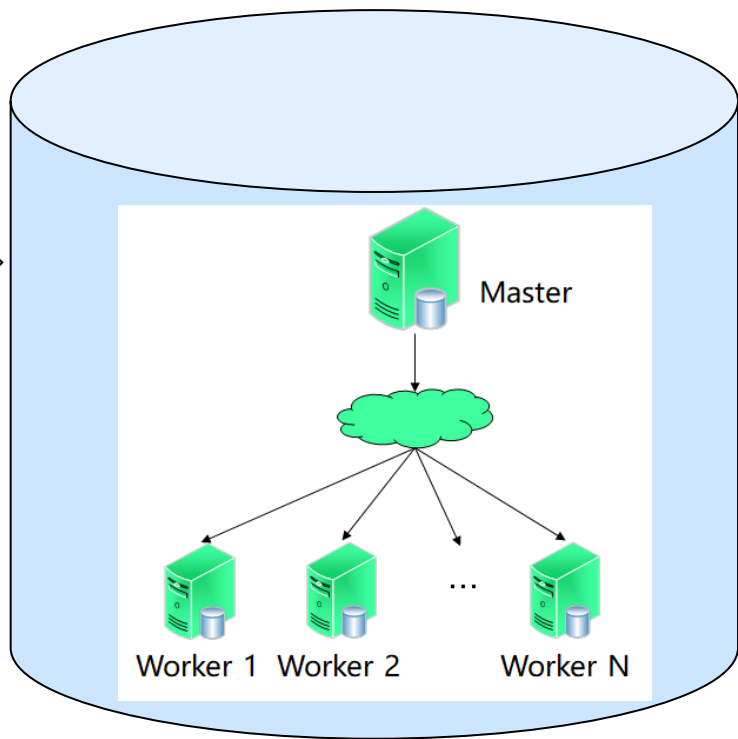
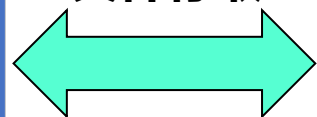
- 集群规模具有可扩展性

- 兼容性好，支持跨平台



存储系统
Client

文件存取



HDFS分布式存储系统

HDFS的文件系统

```
(base) ices@ices-master:~$ hdfs dfs -ls /exp2/douban
Found 3 items
-rw-r--r--   3 ices supergroup 518555983 2021-10-17 16:20 /exp2/douban/comment_split.txt
-rw-r--r--   3 ices supergroup 471869510 2021-10-12 12:02 /exp2/douban/comments.txt
-rw-r--r--   3 ices supergroup    6783 2021-10-11 20:59 /exp2/douban/movie_comment.json
```

➤ 文件URL定位: `hdfs://10.28.36.101:8020/exp2/douban/commens.txt`

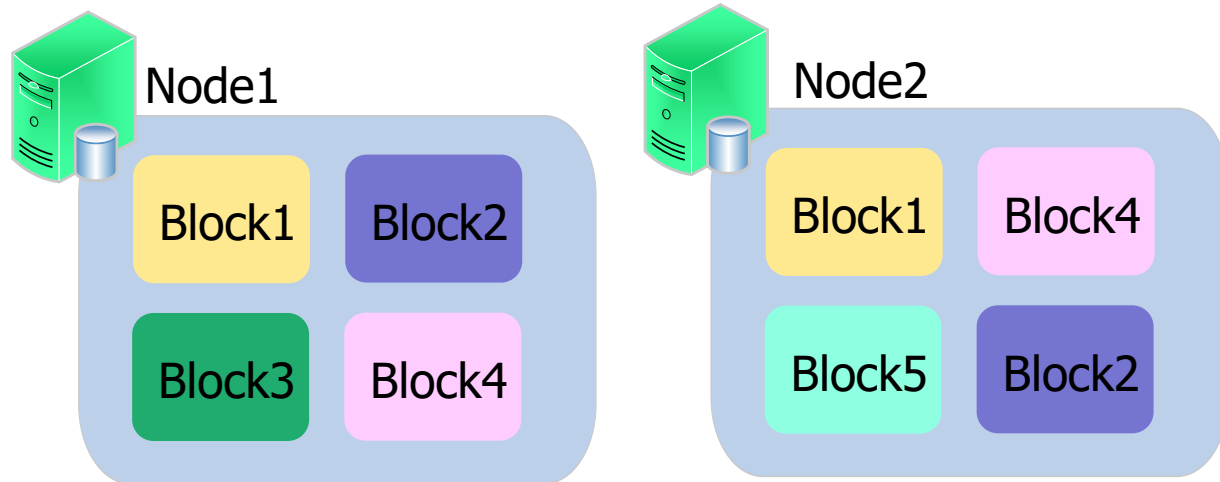
➤ 简单的文件访问模型: 类似于linux的文件系统! ✓

HDFS文件的“分块”存储思想

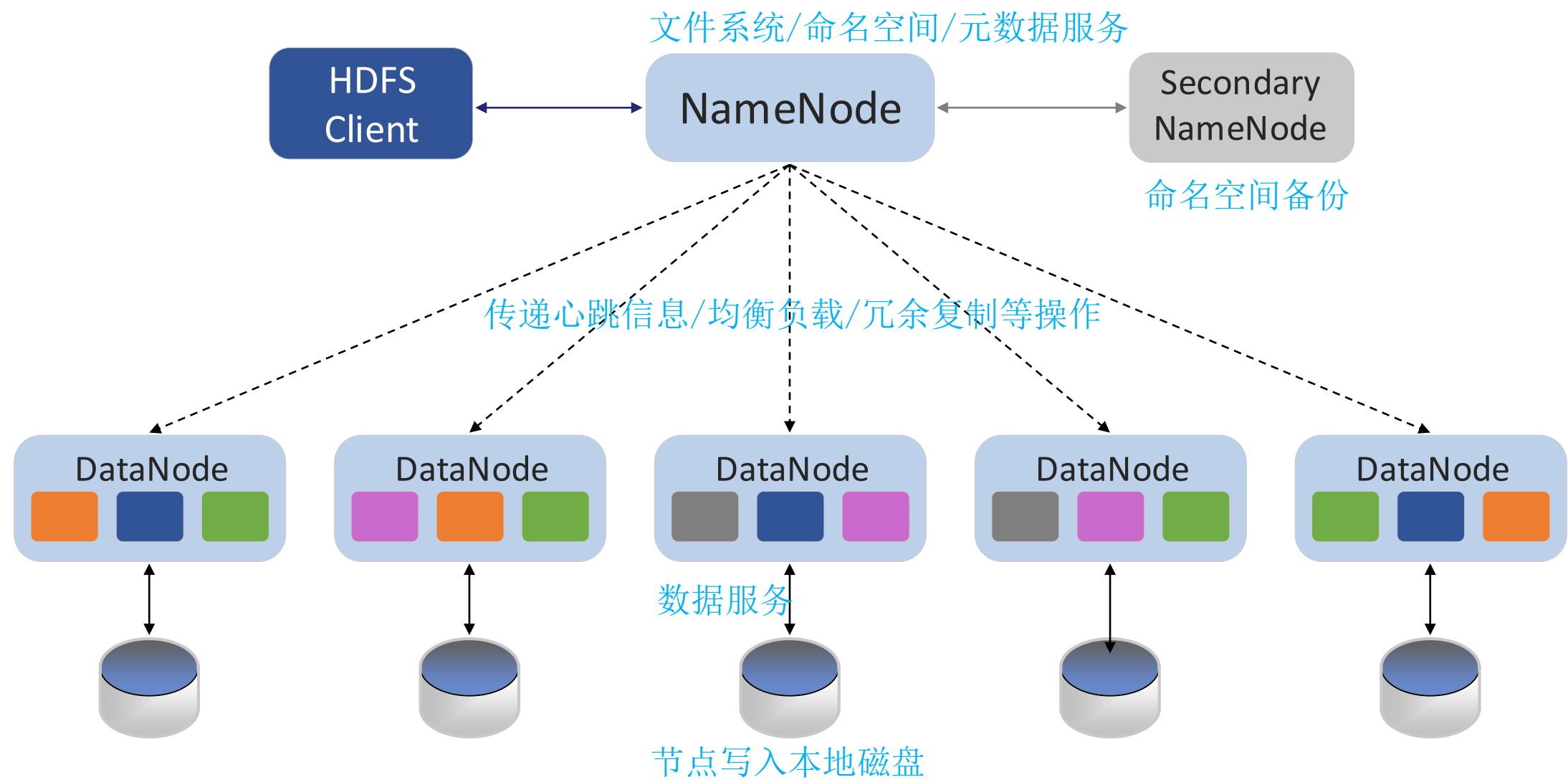
- 默认最基本的存储单位是数据块（如128MB），一个大规模文件被切分成不同的块，每个块尽可能地存储于不同的数据节点中
 - 块的大小远远大于普通文件系统，可以最小化寻址开销
 - 支持大规模文件存储、简化系统设计、适合数据备份

➤ PB级数据的可靠存储 ✓

➤ 流数据读写：高吞吐率 ✓

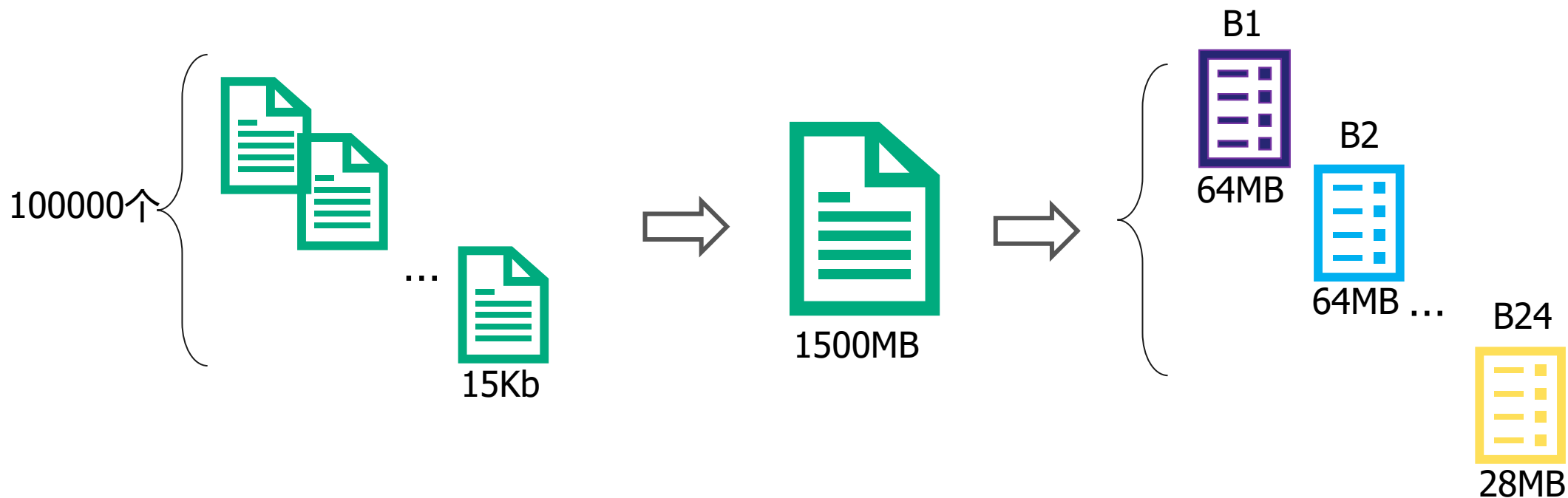


HDFS的基本架构

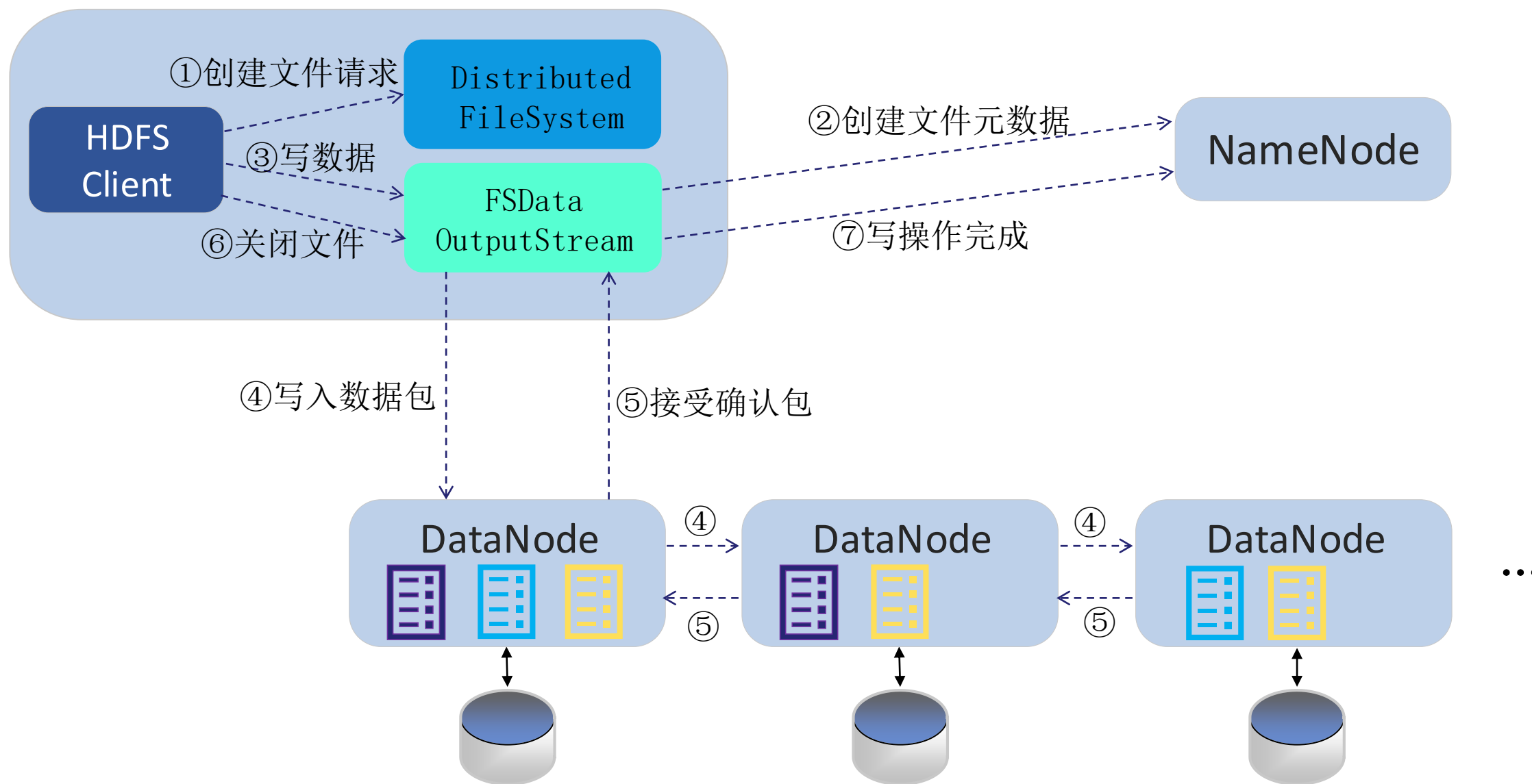


网页存储应用示例（问题）

- 用户需求：将10万个网页存储到HDFS系统中：

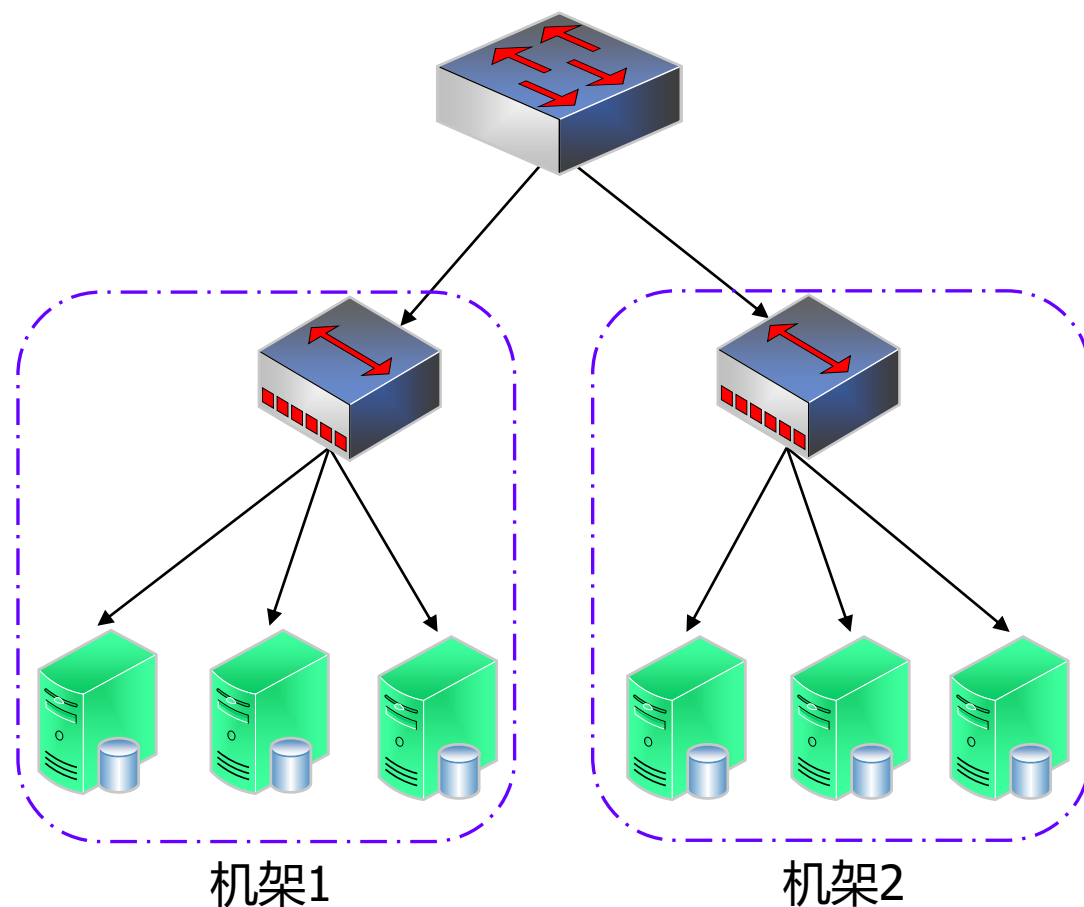


网页存储应用示例（流程）



回顾：大数据分布式处理的主要技术问题分析

- 计算任务（负载）如何分配？
- 分布式通信带来的额外开销问题？
 - 数据分发、处理结果收集
 - 大数据分布处理可带来的数据传输问题
- 硬件故障怎么办？
 - 节点故障、网络故障
- 系统的访问接口是否简单易用？
- 硬件性能需求及成本如何？



MapReduce简介

- MapReduce是一个统一的分布式并行计算软件框架，可以实现：
 - 计算任务的划分和调度
 - 数据的分布传输
 - 计算及处理结果的收集
 - 处理系统节点出错检测和失效恢复
 - 系统管理、负载平衡、计算性能优化
 -
 - 提供简单、易用的编程接口

MapReduce的基本思想

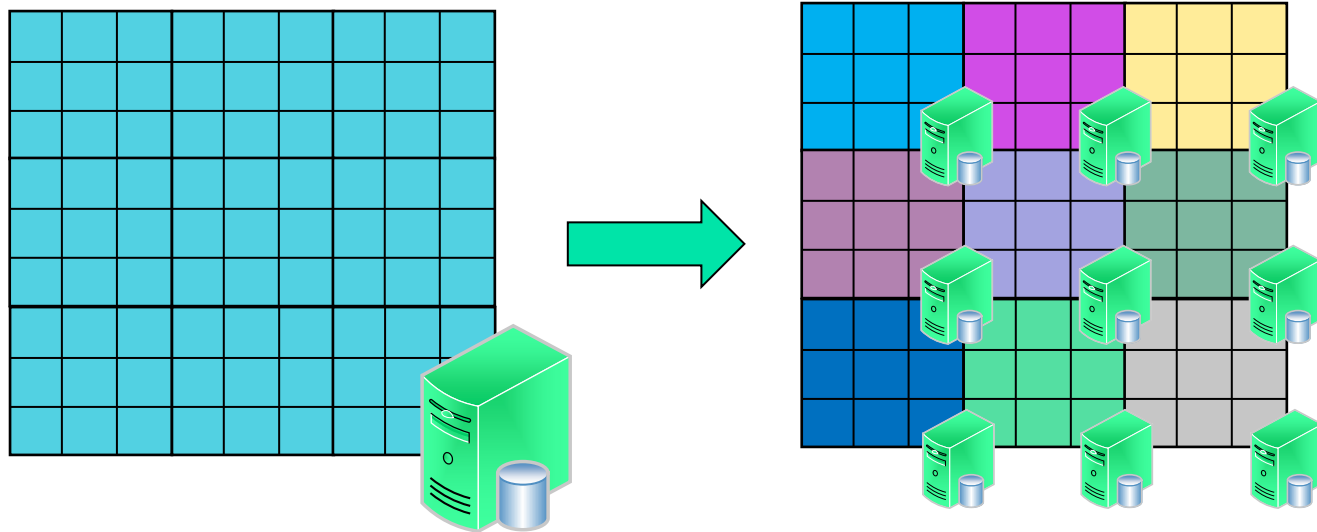
- 采用分而治之的思想实现大规模数据的并行运算

- Map函数：

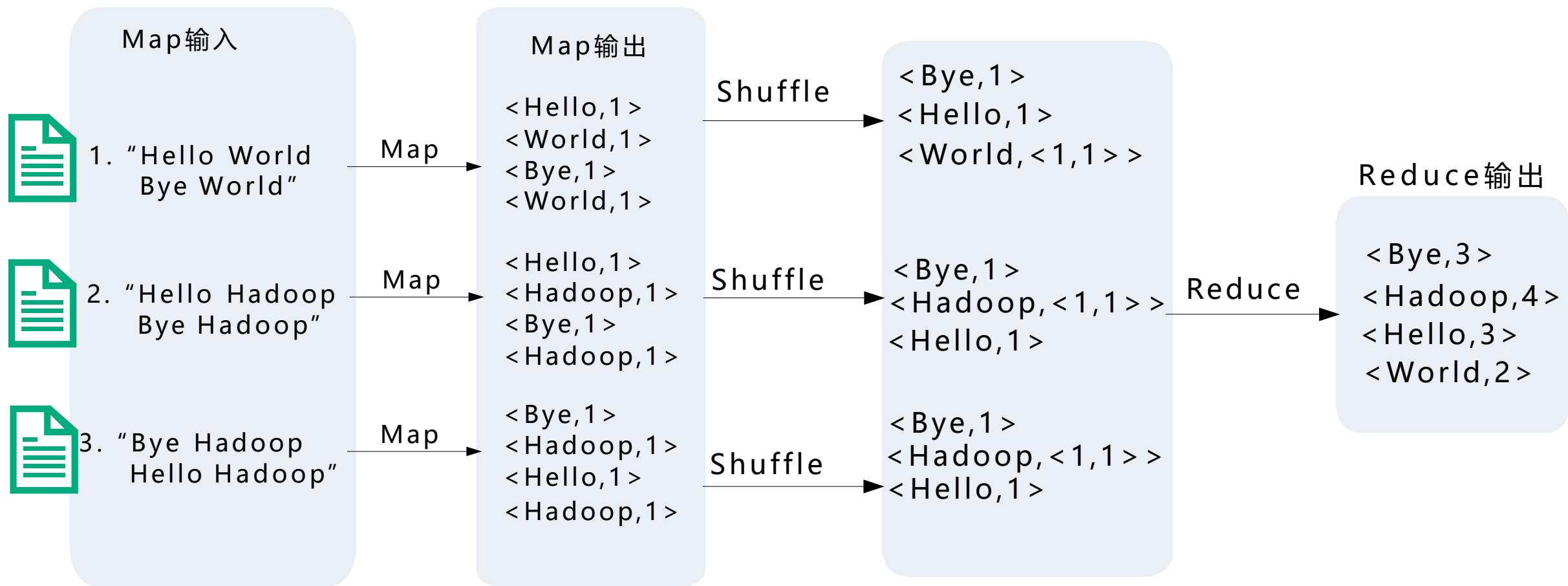
- 大量数据记录进行重复、简单处理
- 只需要局部信息，获得中间结果

- Reduce函数：

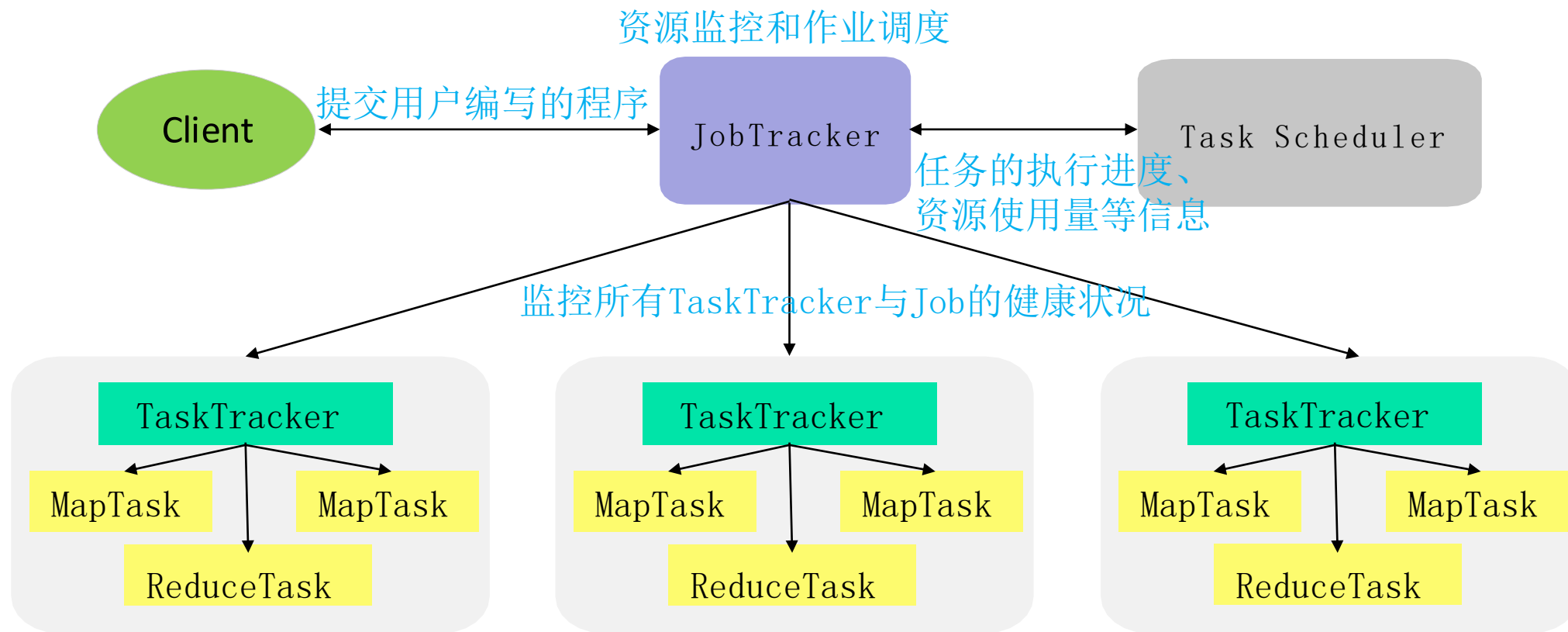
- 整理（全局的）中间结果
- 产生最终结果输出



一个Map-Reduce任务分解示例



MapReduce的基本框架

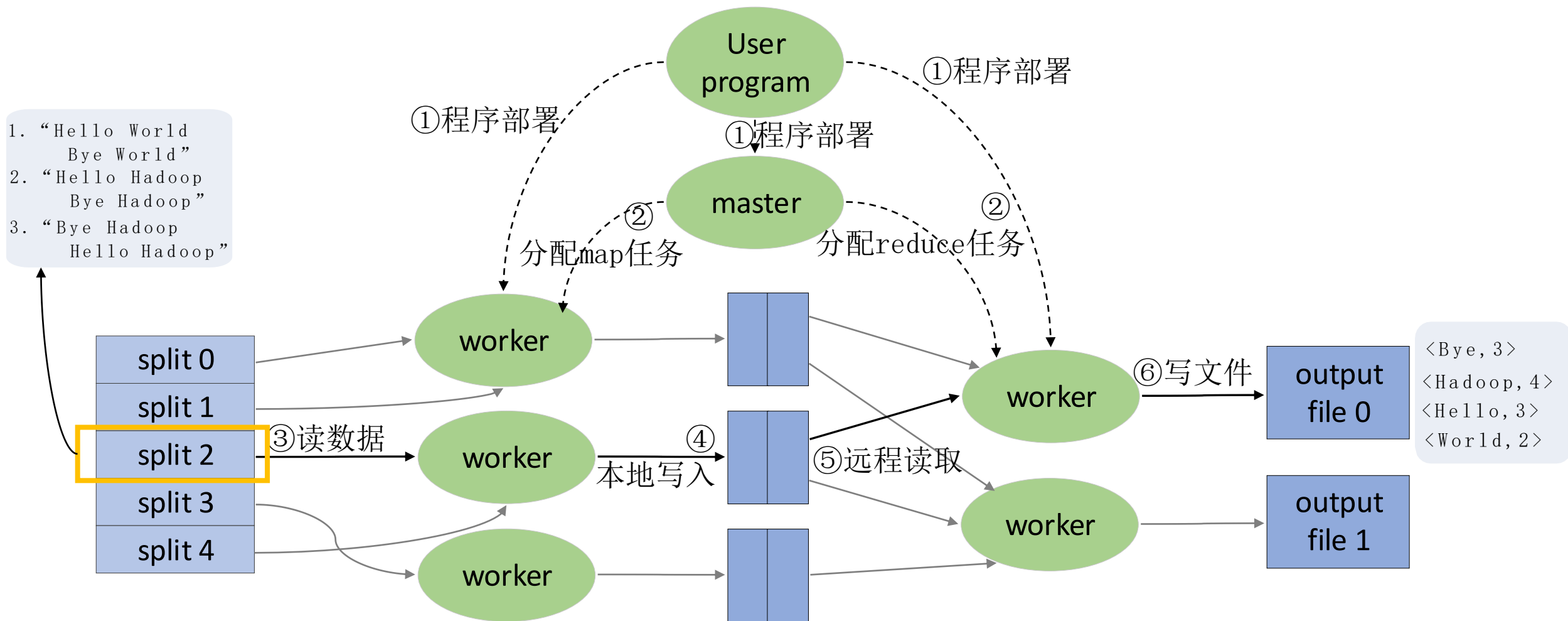


TaskTracker使用“slot”等量划分本节点上的资源量

slot 分为Map slot 和Reduce slot 两种，分别供MapTask 和Reduce Task 使用

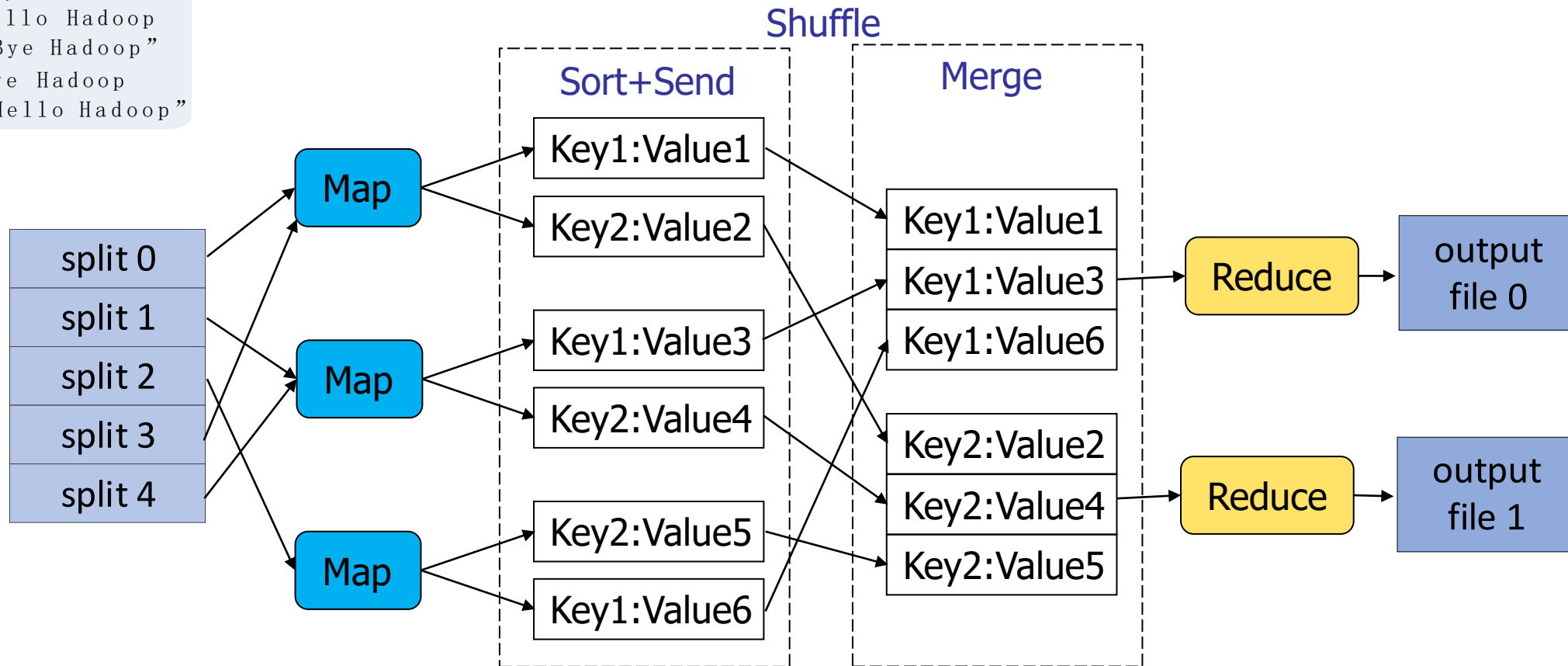
MapReduce的工作流程

“移动代码，而不是移动数据！”



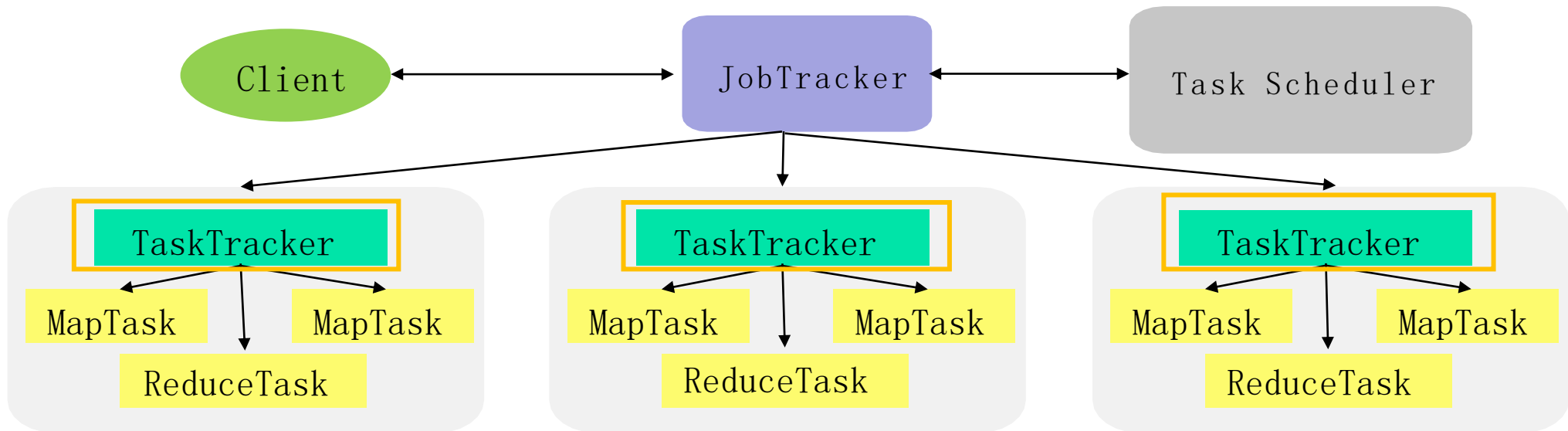
MapReduce中的 “局部计算” 和 “全局计算”

1. “Hello World
Bye World”
2. “Hello Hadoop
Bye Hadoop”
3. “Bye Hadoop
Hello Hadoop”



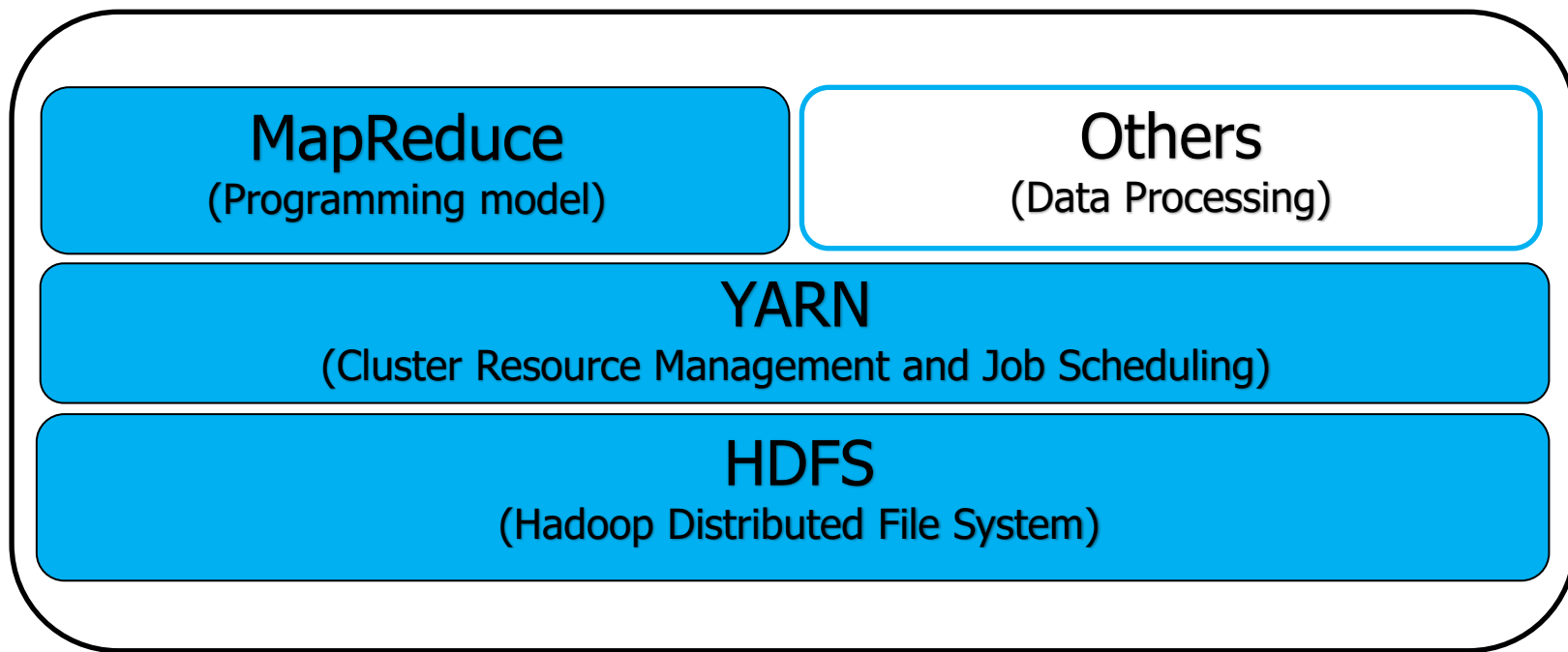
MapReduce 1.0计算框架的缺点

- 既是一个计算框架也是一个资源管理调度框架，存在一些缺陷：
 - 存在单点故障
 - JobTracker “大包大揽” 导致任务过重（任务多时内存开销大，一般上限4000节点）
 - 容易出现内存溢出（分配资源只考虑MapReduce任务数，不考虑CPU、内存）
 - 资源划分不合理（强制划分Map slot和Reduce slot）



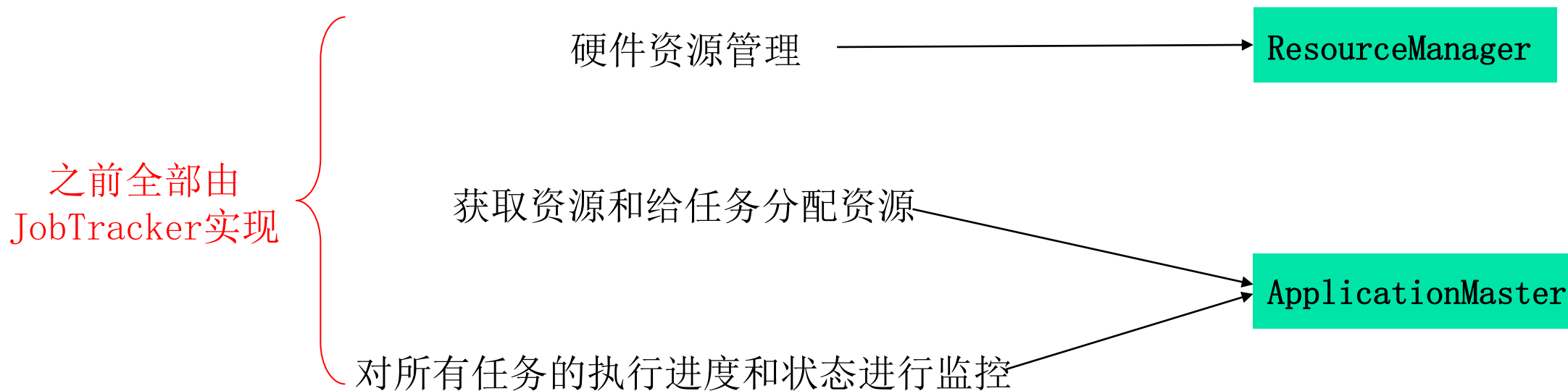
YARN的产生

- Hadoop2.0以后， MapReduce1.0中的资源管理调度功能被单独分离出来形成了YARN，它是一个纯粹的资源管理调度框架
- MapReduce2.0成为了运行在YARN之上的一个纯粹的计算框架



YARN的总体设计思路

- 将原JobTracker三大功能分层解耦：



YARN的基本框架

- ResourceManager

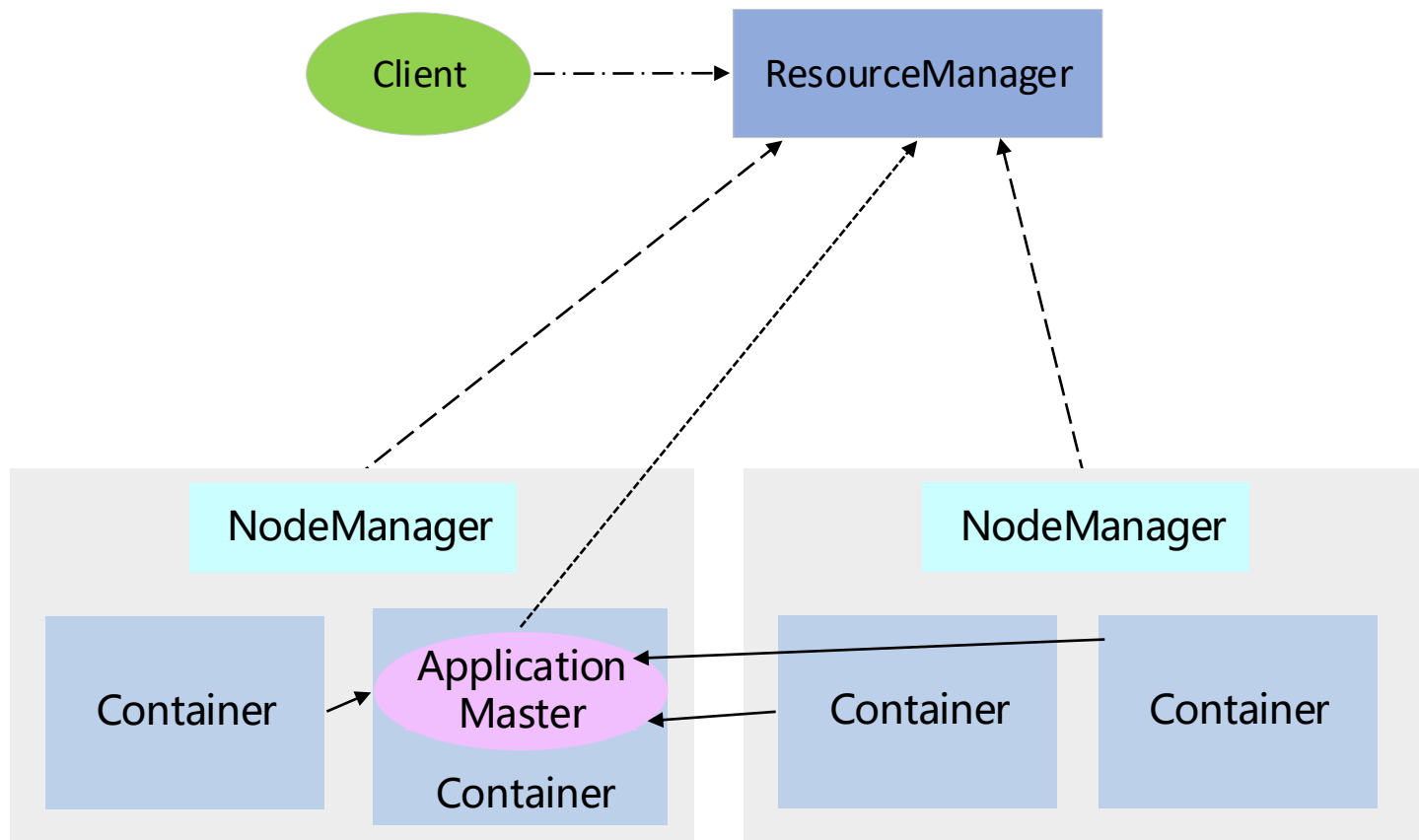
- 处理客户端请求
- 启动/监控ApplicationMaster
- 监控NodeManager
- 资源分配与调度

- ApplicationMaster

- 为应用程序申请资源，分配给内部任务
- 任务调度、监控与容错

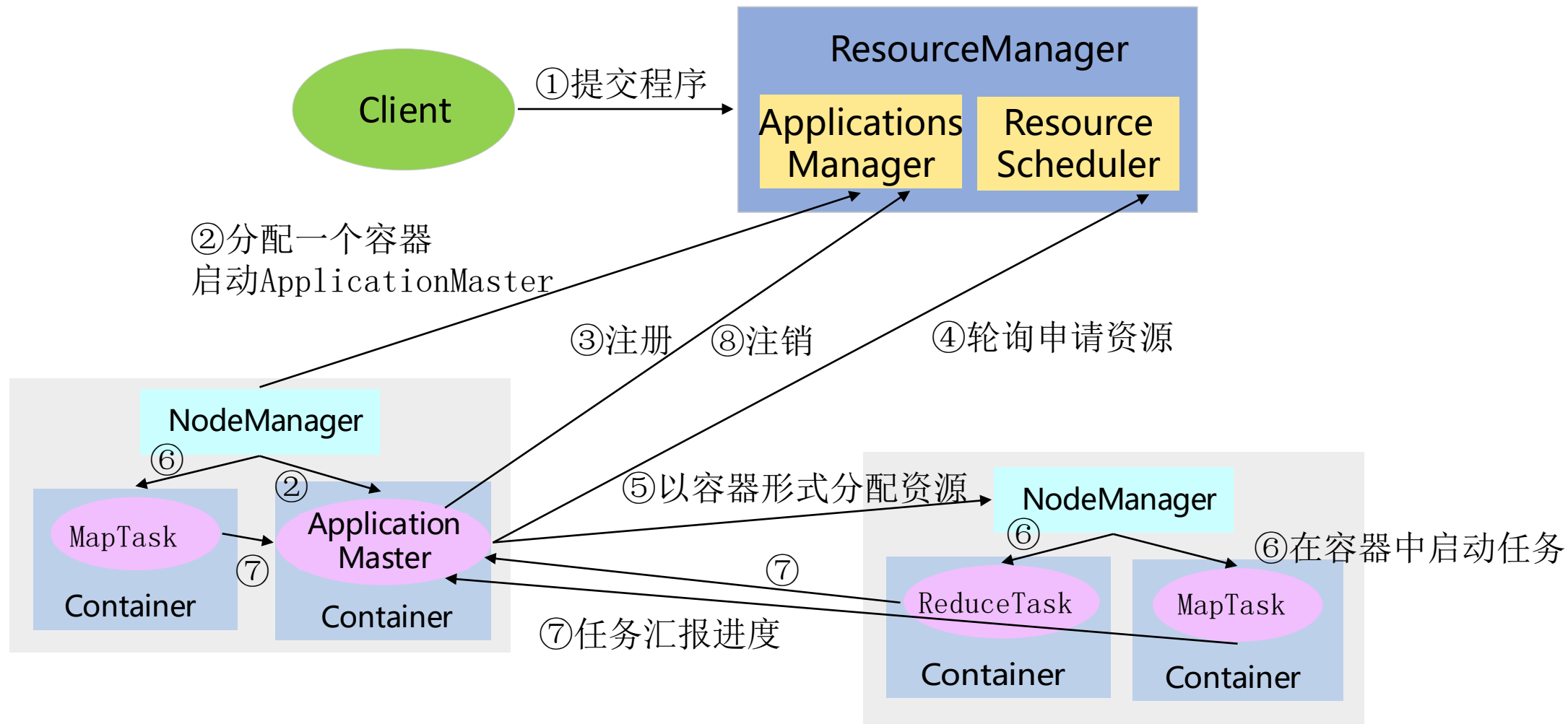
- NodeManager

- 单个节点上的资源管理
- 处理来自ResourceManager的命令
- 处理来自ApplicationMaster的命令



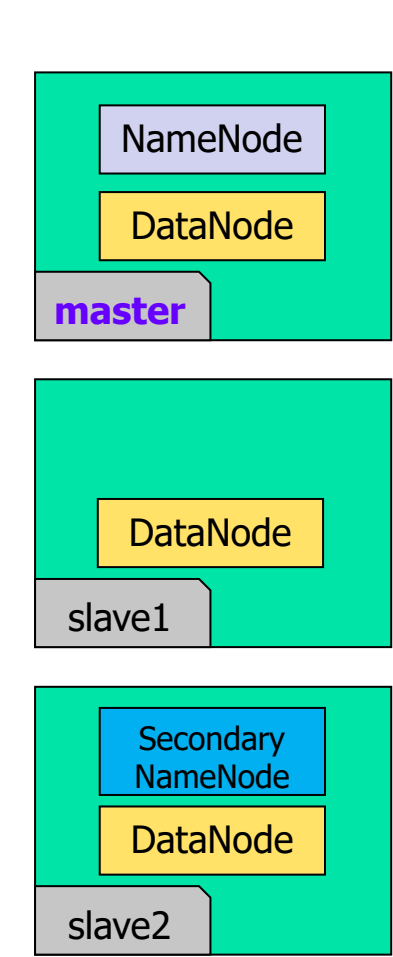
程序提交 - · - · →
节点状态 - - - →
资源请求 →
任务状态 ———→

YARN的工作流程

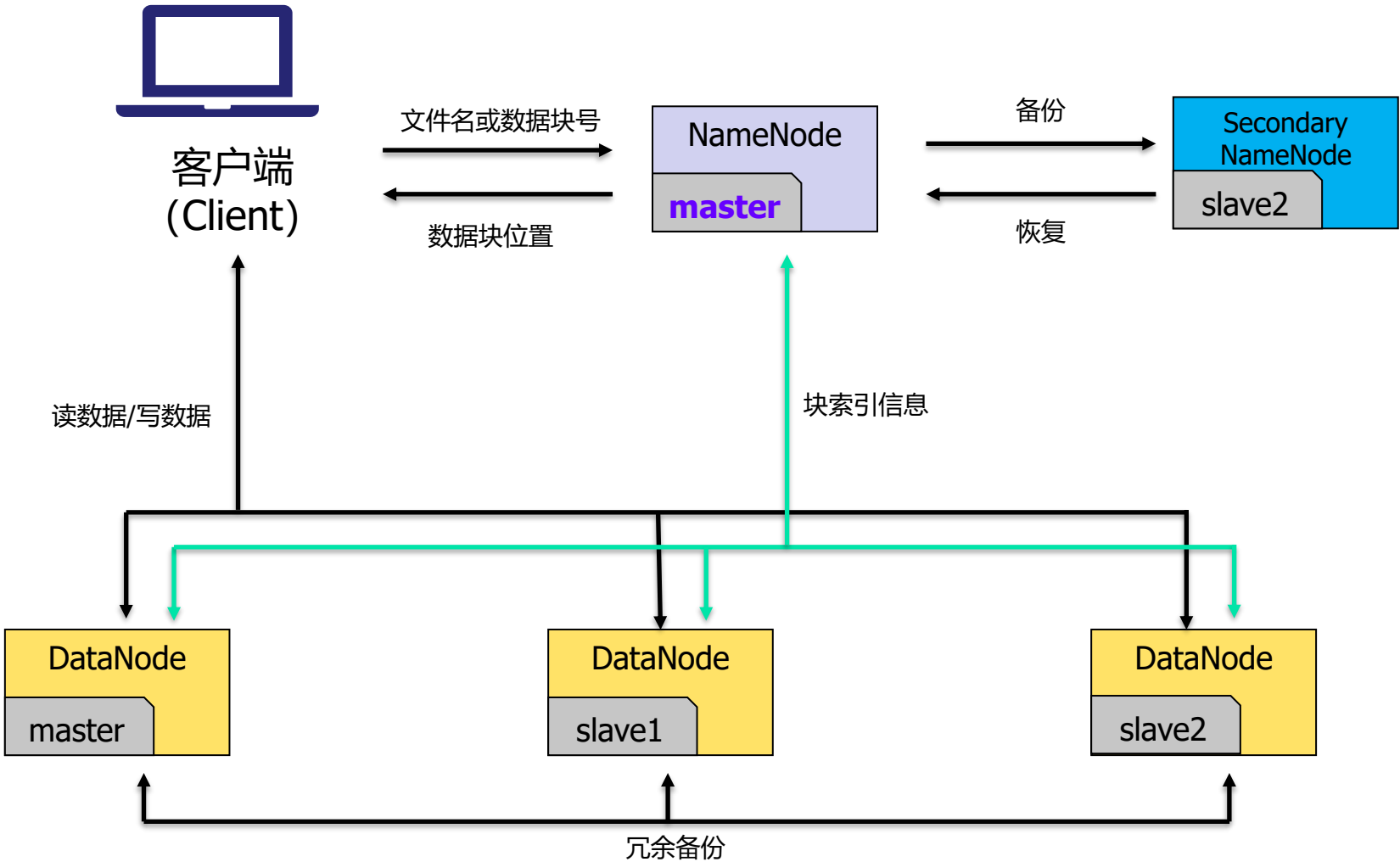


Hadoop平台搭建示例

HDFS环境示例



物理节点规划






集群架构

Browse Directory

/




Go!



Browse Directory

/exp2/douban

Go!



Show 25 entries

<input type="checkbox"/>	Permission	Owner
<input type="checkbox"/>	-rw-r--r--	ices
<input type="checkbox"/>	-rw-r--r--	ices
<input type="checkbox"/>	-rw-r--r--	ices




Showing 1 to 3 of 3 entries

Hadoop, 2021.

Showing 1 to 10 of 10 entries

Hadoop, 2021.

Search:

Name	
comment_split.txt	
comments.txt	
movie_comment.json	

Previous

1

Next

Previous

1

Next

File information - comment_split.txt

Download

Head the file (first 32K)

Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741856

Block Pool ID: BP-424571681-10.249.182.54-1633752692273

Generation Stamp: 1032

Size: 134217728

Availability:

- ices-master
- ices-slave1
- ices-slave2

Close

HDFS命令行操作 (1)

查看所有命令

```
$ hadoop fs
```

```
[-appendToFile <localsrc> ... <dst>]  
[-cat [-ignoreCrc] <src> ...]  
[-chgrp [-R] GROUP PATH...]  
[-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]  
[-chown [-R] [OWNER][:[GROUP]] PATH...]  
[-copyFromLocal [-f] [-p] <localsrc> ... <dst>]  
[-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  
[-count [-q] <path> ...]  
[-cp [-f] [-p] <src> ... <dst>]  
[-df [-h] [<path> ...]]  
[-du [-s] [-h] <path> ...]  
[-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  
[-getmerge [-nl] <src> <localdst>]  
[-help [cmd ...]]  
[-ls [-d] [-h] [-R] [<path> ...]]  
[-mkdir [-p] <path> ...]  
[-moveFromLocal <localsrc> ... <dst>]
```

HDFS命令行操作 (2)

创建目录 /bigdata

```
$ hadoop fs -mkdir /bigdata
```

移动根目录文件 test.txt 到 /bigdata

```
$ hadoop fs -mv /test.txt /bigdata
```

显示 /bigdata 目录信息

```
$ hadoop fs -ls /bigdata
```

```
(base) ices@ices-master:~$ hadoop fs -mkdir /bigdata
(base) ices@ices-master:~$ hadoop fs -mv /test.txt /bigdata
(base) ices@ices-master:~$ hadoop fs -ls /bigdata
Found 1 items
-rw-r--r--    3 ices supergroup          8 2021-11-02 15:42 /bigdata/test.txt
```

HDFS存取管理的Java API示例

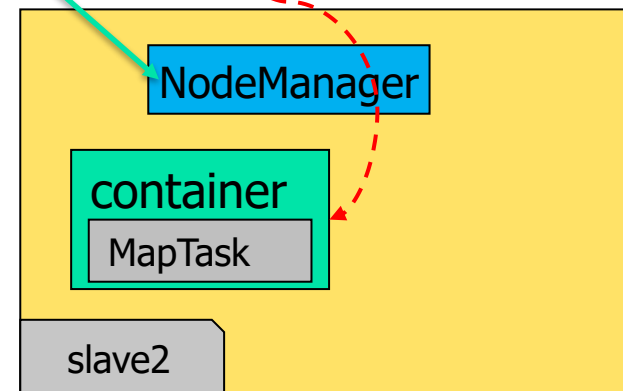
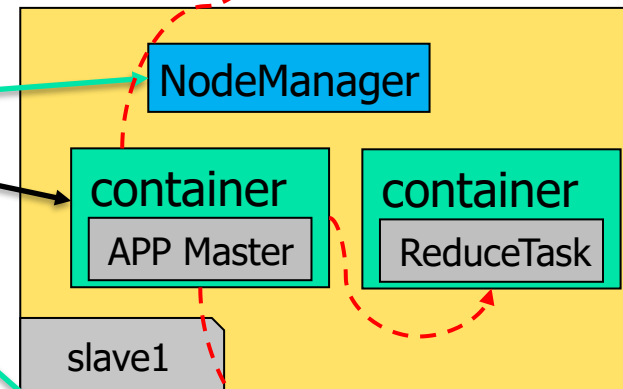
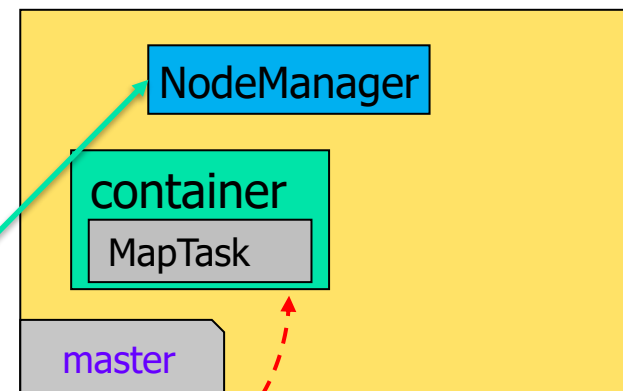
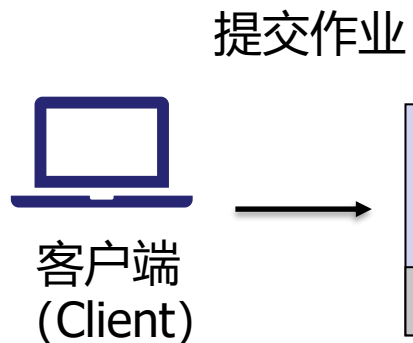
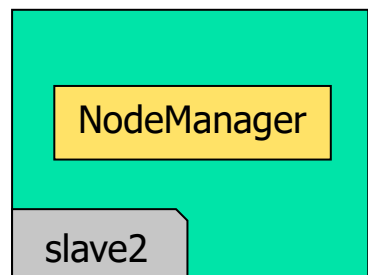
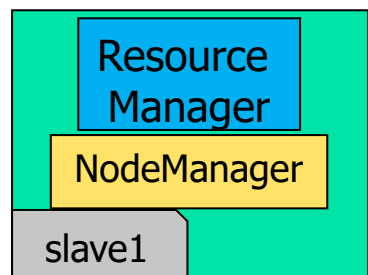
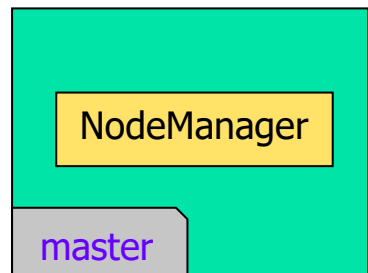
```
public class HdfsClient {
    @Test
    public void testMkdirs() throws IOException, URISyntaxException, InterruptedException {
        // 1 获取文件系统
        Configuration configuration = new Configuration();
        FileSystem fs = FileSystem.get(new URI("hdfs://ices-master:8020"),
                                       configuration, "ices");

        // 2 创建目录
        fs.mkdirs(new Path("/hitsz/bigdata/"));

        // 3 列出目录下文件夹和文件名称
        FileStatus[] statuses = fs.listStatus(new Path("/hitsz"));
        for (FileStatus file : statuses){
            String isDir = file.isDirectory() ? "Folder " : "File";
            String path = file.getPath().getName();
            System.out.println(isDir+ "\t" +path);
        }

        fs.close();
    }
}
```

MapReduce on Yarn 环境示例



MapReduce作业运行示例

启动MapReduce作业:

Hadoop jar XXX.jar(jar包) XXX(类名) /input /output

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hadoop jar join.jar ReduceJoin /input /ou
tput
20/05/13 20:23:01 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
20/05/13 20:23:09 INFO Configuration.deprecation: session.id is deprecated. Inst
ead, use dfs.metrics.session-id
20/05/13 20:23:09 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName
=JobTracker, sessionId=
20/05/13 20:23:09 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
20/05/13 20:23:16 INFO input.FileInputFormat: Total input paths to process : 2
20/05/13 20:23:16 INFO mapreduce.JobSubmitter: number of splits:2
20/05/13 20:23:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_lo
cal281746740_0001
20/05/13 20:23:17 INFO mapreduce.Job: The url to track the job: http://localhost
:8080/
20/05/13 20:23:17 INFO mapreduce.Job: Running job: job_local281746740_0001
20/05/13 20:23:17 INFO mapred.LocalJobRunner: OutputCommitter set in config null
20/05/13 20:23:17 INFO output.FileOutputCommitter: File Output Committer Algorit
hm version is 1
20/05/13 20:23:17 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hado
op.mapreduce.lib.output.FileOutputCommitter
20/05/13 20:23:17 INFO mapred.LocalJobRunner: Waiting for map tasks
20/05/13 20:23:17 INFO mapred.LocalJobRunner: Starting task: attempt_local281746
740_0001_m_000000_0
20/05/13 20:23:17 INFO output.FileOutputCommitter: File Output Committer Algorit
hm version is 1
```

https://blog.csdn.net/qq_43374605

实践任务：一个简化的分布式大数据存储与处理示例

- 从点评网站上下载1万个网页并保存到Hadoop分布式文件存储系统中
- 解析各个网页，用Hadoop分布式处理框架统计出top 20的关键词

致谢

- 一小部分图表、文字参考了教材、互联网上的开放资料等，本文件仅供公益性的学习参考，在此表示感谢！如有版权要求请联系：
yym@hit.edu.cn，谢谢！