

Hadoop概述

2019.05

相互学习，内部分享，请多多指教

目录

- Hadoop发展史
 - 萌芽期：google的三篇论文
 - 搜索引擎时代：hadoop的诞生
 - 数据仓库时代：hive的应用
 - 数据挖掘时代：spark等的发展
- Hadoop生态圈介绍
 - 狭义的Hadoop
 - 广义的Hadoop
- Hadoop组件讲解
 - Hdfs讲解与集群操作
 - Yarn讲解与集群操作
 - MapReduce讲解与集群操作
 - Hadoop经典案例：词频统计
- 思维拓展：Hadoop的现在和未来

Hadoop发展史

Hadoop发展史1：以前

- 以前：提升单机性能：IBM小型机、EMC企业级存储、Oracle企业级数据库

IBM

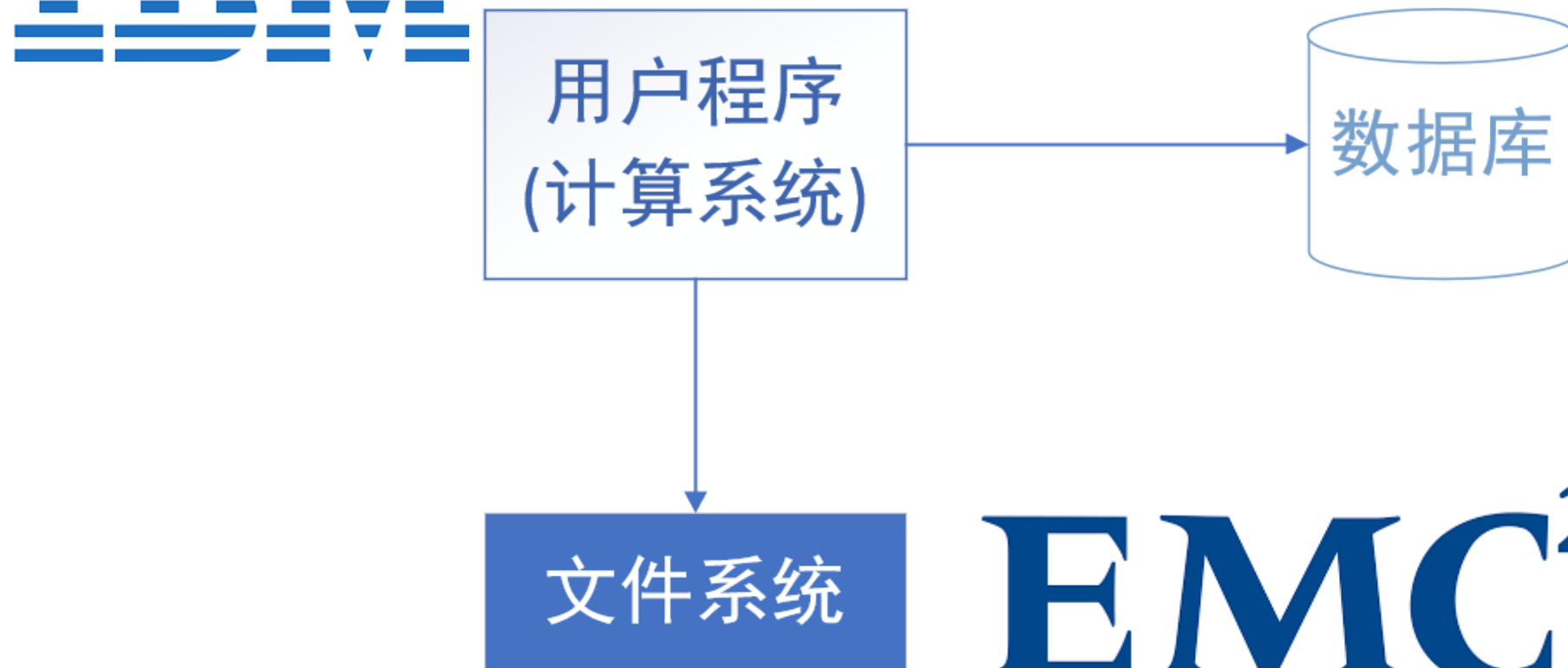
用户程序
(计算系统)

ORACLE®

数据库

文件系统

EMC²

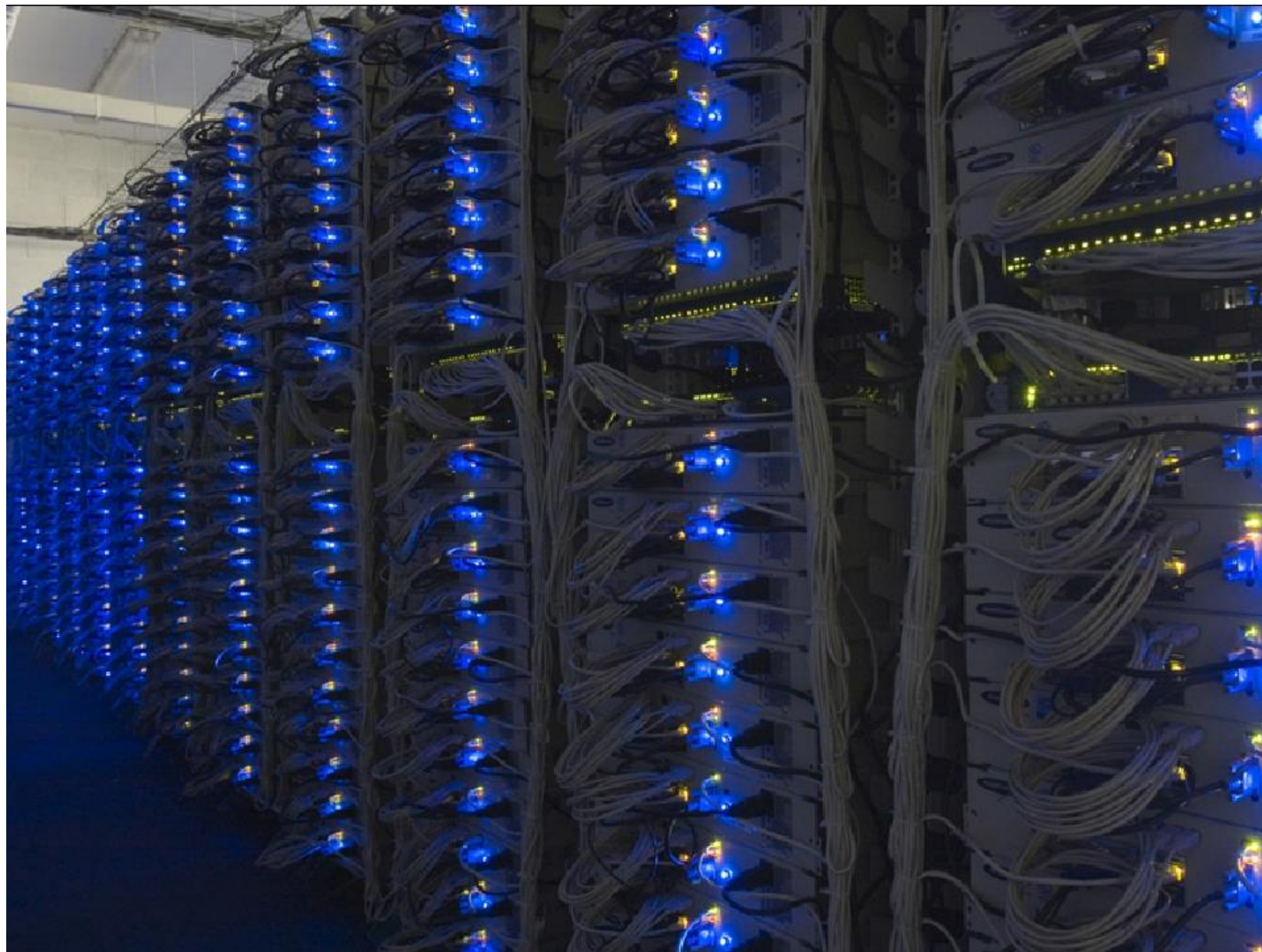


动机：谷歌的例子

- 100亿个网页
- 平均网页大小 = 20KB
- $100\text{亿} * 20\text{KB}$ = **200TB**
- 磁盘读取带宽 = 50MB/sec
- 读取数据所需时间 = 400万秒 = **46+ 天**
- 后续的数据处理与操作花费的时间可能会更多

数据之大，单机装不下——集群

- 2011年据统计，google约有100万台机器



集群计算需要面对的问题(1)

● 节点故障

虽然我们的笔记本电脑一年都很少故障，但

1000台服务器的集群 => 平均故障率 1 次/天

100万台服务器的集群 => 平均故障率 1000 次/天

如何保持数据的持续性，

即在某些节点故障的情形下不影响依旧能够使用数据

在运行时间较长的集群运算中，如何应对节点故障呢

集群计算需要面对的问题(2)

网络带宽瓶颈

网络带宽 = 1 Gbps

移动10TB 数据需要花费将近一天

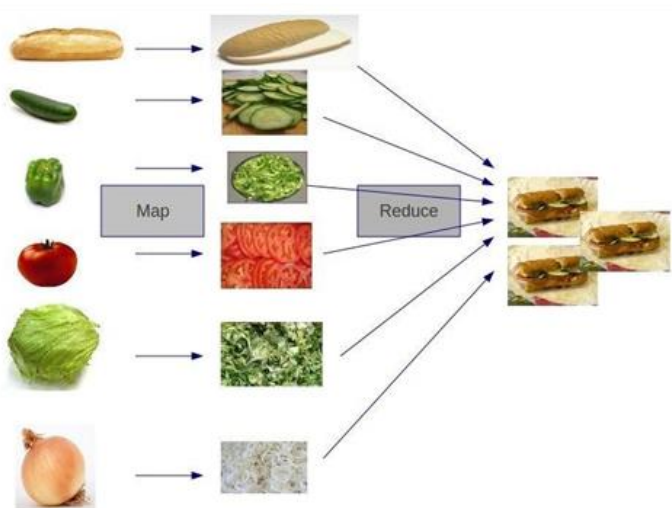
分布式编程非常复杂

需要一个简单的模型能够隐去所有的复杂性

Hadoop发展史2：2003年Google的三驾马车

Hadoop思想之源：Google的搜索引擎

- 大量的网页怎么存储（运用冗余防止数据丢失）
 - 分布式文件系统 GFS
- Page-Rank的计算问题（单台机器不够算）
 - 分布式计算框架 Map-Reduce



- 如何快速查到数据（响应时间仅为0.01秒，甚至更快）
 - NoSql数据库系统 Bigtable（论文发表于2006年）

文件系统

用户程序
(计算系统)

数据库

GFS冗余化数据存储结构

分布式文件存储系统

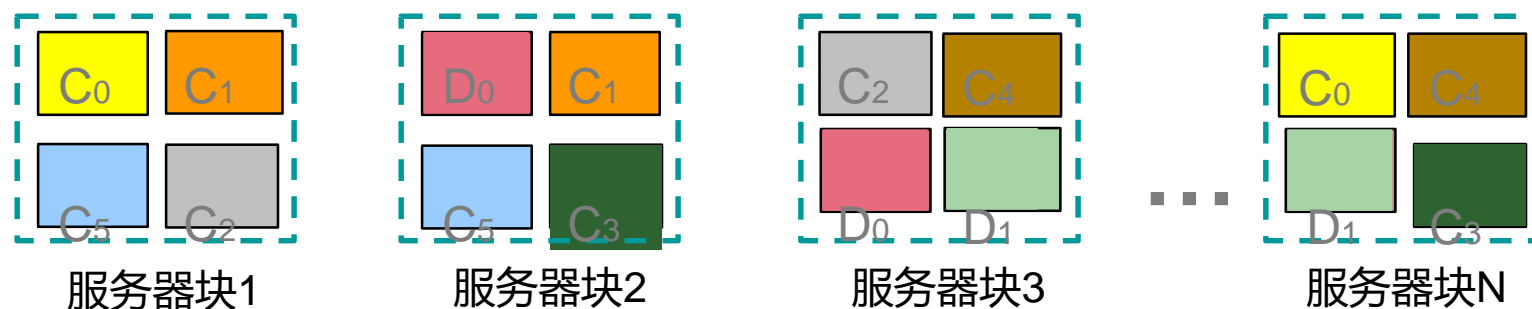
提供全局的文件命名空间，冗余度和可获取性
例如Google 的 GFS; Hadoop 的 HDFS

典型的应用场景与模式

超大级别的数据量(100GB到100TB级别)
数据很少就地整个被替换
最常见的操作为读取和追加数据

分布式文件系统

数据以“块状”形式在多台机器上存储
每个数据块都会重复地在多台机器上存储
保证数据的持续性和随时可取性



服务器块同时也用作计算服务器。

把运算挪向数据处！

Map-Reduce集群运算时问题的解决方案

在多节点上冗余地存储数据，以保证数据的持续性和一直可取性

将计算移向数据端，以最大程度减少数据移动

简单的程序模型隐藏所有的复杂度

大数据简史3：2006年Hadoop

- 2006年 Hadoop开源
- 2008年 成为Apache 顶级项目
 - GFS Hadoop HDFS
 - MapReduce Hadoop MapReduce
 - Bigtable Hbase



Hadoop之父：Doug Cutting

- 插播一句：开源的魅力
 - 优秀的软件：成就自己 Windows IBC Oracle EMC
 - 优秀的开源的软件：**成就世界**！Linux Java mysql Hadoop Spark...

大数据应用：搜索引擎时代

- 标志公司



- 解决的问题：搜索引擎需要大量的数据存储与计算

- 特点：

- 主要用于特定场景，开发难度高

- 代表应用

- GFS/HDFS
- MapReduce

大数据简史4：Hadoop的发展

- 2006 年 5 月，Yahoo! 建立了一个 300 个节点的 Hadoop 研究集群。
- 2007 年 4 月，研究集群增加到两个 1000 个节点的集群。
- 2007 年，百度开始使用 Hadoop 做离线处理。
- 2008 年，淘宝——云梯研究并使用Hadoop。
- **2008年 1月，Hadoop成为 Apache顶级项目。**
- 2008 年 2 月，Yahoo! 运行了世界上最大的 Hadoop 应用，1万个核。
- **2008年 8月，第一个 Hadoop商业化公司 Cloudera成立。**
- 2010 年 5 月，Avro、HBase 脱离 Hadoop 项目，成为 Apache 顶级项目。
- 2010 年 9 月，Hive、Pig脱离 Hadoop，成为 Apache 顶级项目。
- 2010年 -2011年，扩大的 Hadoop社区忙于建立大量的新组件（Crunch，Sqoop，Flume，Oozie等）来扩展 Hadoop的使用场景和可用性。
- 2011 年 1 月，ZooKeeper 脱离 Hadoop，成为 Apache 顶级项目。

大数据应用：数据仓库时代

- 标志事件：Hive、Hbase等的开源与应用

- 解决的问题：

- 用更低廉的人力（懂SQL）进行更多的分布式存储与计算来实现数据分析需求

- 特点：

- 数据多样化
- 用于大数据统计

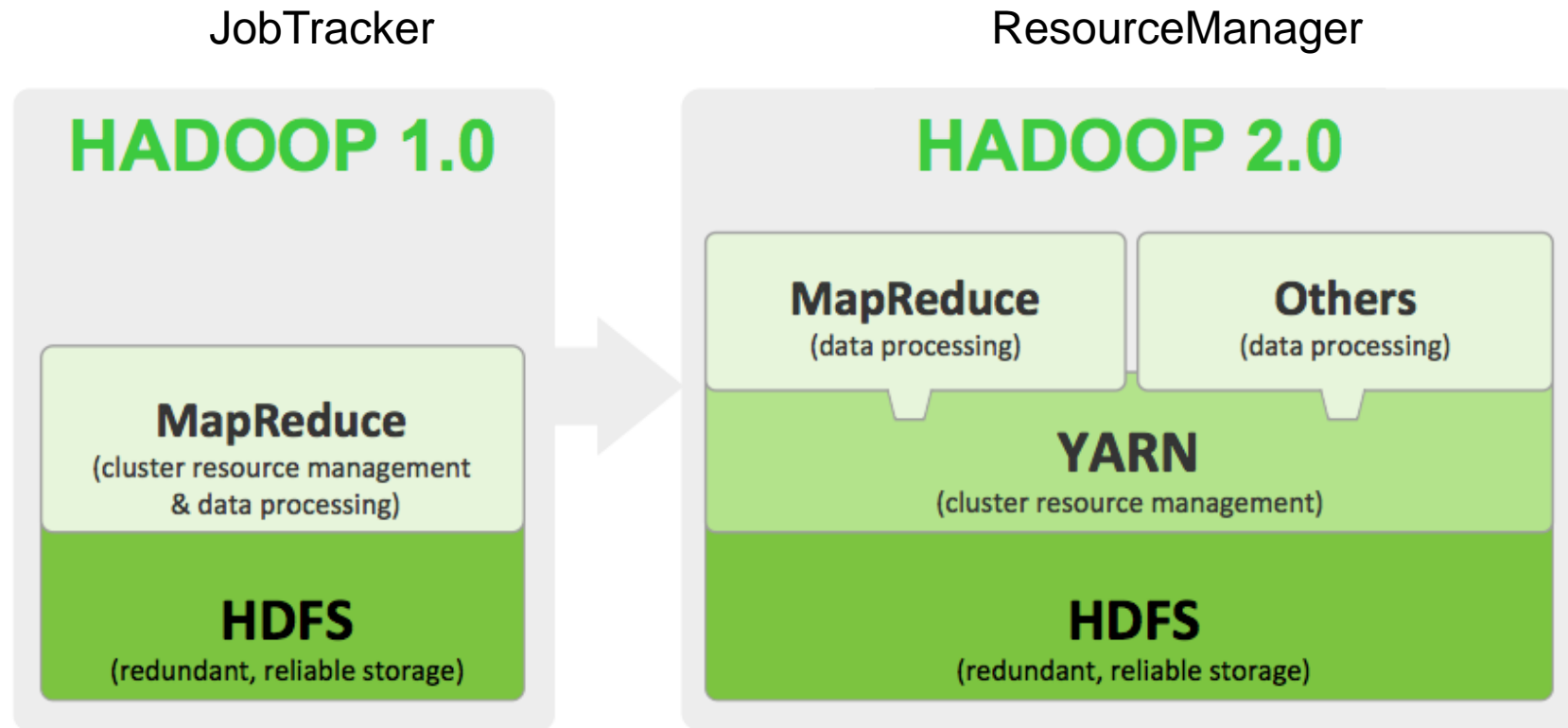
- 代表应用

- Hive、Hbase、Imapla等
- Hue以及公司自研的大数据操作管理软件



大数据简史5：2012年：从Yarn到百花齐放

- 2012 年 8 月，YARN 成为 Hadoop 子项目。



- 2014年 2月，Spark逐渐代替 MapReduce成为 Hadoop的缺省执行引擎，并成为 Apache基金会顶级项目。

大数据应用：数据挖掘时代

- 标志事件：Spark、TensorFlow等技术的发展

- FLAG、微软苹果、Netflix、BATJ...



NETFLIX

- 深层原因：

- 大数据技术的飞速发展，使得进行机器学习与深度学习算法在大数据量下成为可能，并产生价值

- 特点：

- 数据更加多样化，数据量级指数级增长
- 大数据分析预测，发现数据内在的联系

- 代表应用

- 推荐系统、用户画像
- AI技术、无人驾驶

Hadoop生态圈介绍

Hadoop生态圈介绍：Hadoop的组成

大数据计算 MapReduce	辅助工具 Common
资源管理与调度 Yarn	
大数据存储 HDFS	

Hadoop的三大发行版本

- **Apache**

- Apache版本最原始（最基础）的版本
- 技术最新

- **Cloudera**

- 最早的发行版，2008年成立，Doug Cutting加盟
- 兼容性、安全性、稳定性较高
- Cloudera在大型互联网企业中用的较多

- **Hortonworks**

- 后起之秀，2011年成立
- 离开源更加接近

广义的hadoop

分析报告工具

Zeppelin、Hue、Kibana、Kettle

安全
Kerberos

查询引擎与数据仓库

SparkSql、Hive、Impala、
Presto、Phoenix、ES、Kylin

图计算

Spark GraphX、Pregel

数据挖掘与机器学习

Spark ML、Mahout、
TensorFlow、Caffe

工作流

Azkaban、Oozie

批处理计算

MapReduce、**Spark**、Tez

流式计算

Storm、**Spark Streaming**、
Flink、**Structured Streaming**、
Beam、Kafka Streams

NoSQL 系统

Hbase、Cassandra、
MongoDB、Redis

集群管控

Ambari

资源调度

Yarn、Mesos、kubernets、**spark standalone**

数据序列化

Avro

大数据存储

HDFS、**Alluxio**、S3、Ceph...

分布式协调服务

Zookeeper

数据搜集与迁移

Logstash、Flume、Sqoop

消息系统

Kafka、ActiveMQ、RabbitMQ

Hadoop组件讲解

设计Hadoop该注意哪些？

- 高可靠性：
 - 因为Hadoop假设计算元素和存储会出现故障，因为它维护多个工作数据副本，在出现故障时可以对失败的节点重新分布处理。
- 高扩展性：
 - 在集群间分配任务数据，可方便的扩展数以千计的节点。
- 高效性：
 - 在MapReduce的思想下，Hadoop是并行工作的，以加快任务处理速度。
- 高容错性：
 - 自动保存多份副本数据，并且能够自动将失败的任务重新分配。
- 兼容性
 - 不能只给自己玩，要给其它人开发的组件也能用

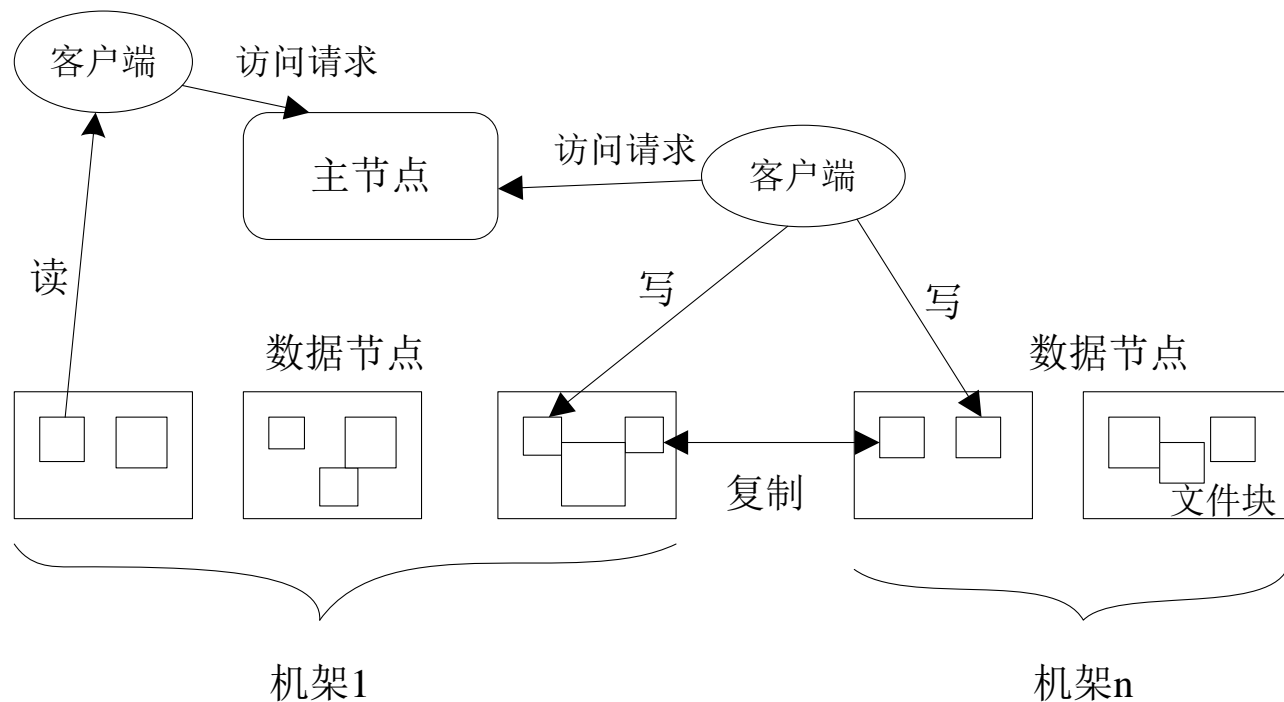
HDFS的设计特点

- HDFS (Hadoop Distributed File System , Hadoop分布式文件系统) , 它是一个高度容错性的系统, 适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问, 适合那些有着超大数据集的应用程序。
- HDFS的设计特点是：
 - **大数据文件**, 非常适合上T级别的大文件或者一堆大数据文件的存储, 如果文件只有几个G甚至更小就没啥意思了。
 - **文件分块存储**, HDFS会将一个完整的大文件平均分块存储到不同计算机上, 它的意义在于读取文件时可以同时从多个主机取不同区块的文件, 多主机读取比单主机读取效率要高得多。
 - **流式数据访问**, 一次写入多次读写, 这种模式跟传统文件不同, 它不支持动态改变文件内容, 而是要求让文件一次写入就不做变化, 要变化也只能在文件末添加内容。
 - **廉价硬件**, HDFS可以应用在普通PC机上, 这种机制能够让给一些公司用几十台廉价的计算机就可以撑起一个大数据集群。
 - **硬件故障**, **HDFS认为所有计算机都可能会出问题**, 为了防止某个主机失效读取不到该主机的块文件, 它将同一个文件块副本分配到其它某几个主机上, 如果其中一台主机失效, 可以迅速找另一块副本取文件。

HDFS的原理与架构

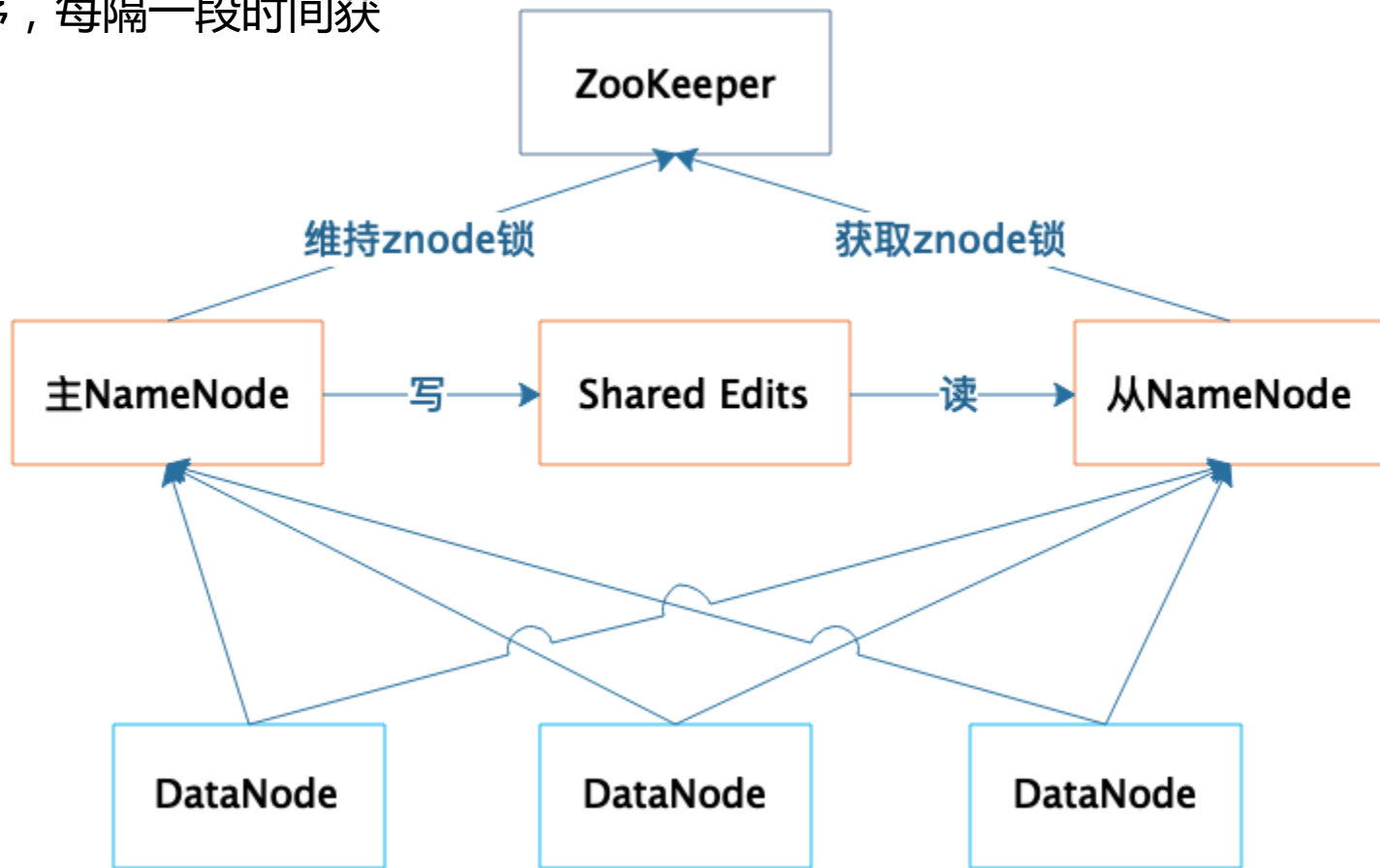
HDFS集群包括，**主节点(NameNode)** 和**数据节点(DataNode)** 以及**从节点(Secondary Namenode)**。

- **数据块(Block)**：大文件的存储会被分割为多个block进行存储。默认为128MB，每一个block会在多个datanode上存储多份副本，默认为3份。
- **主节点(NameNode):**
负责管理整个文件系统的元数据，以及每一个路径（文件）所对应的数据块信息。
- **数据节点(DataNode):**
负责管理用户的文件数据块，每一个数据块都可以在多个datanode上存储多个副本。



HDFS的HA

- **从节点(Secondary NameNode):**
用来监控HDFS状态的辅助后台程序，每隔一段时间获取HDFS元数据的快照。



Hadoop组件讲解——Map-Reduce1

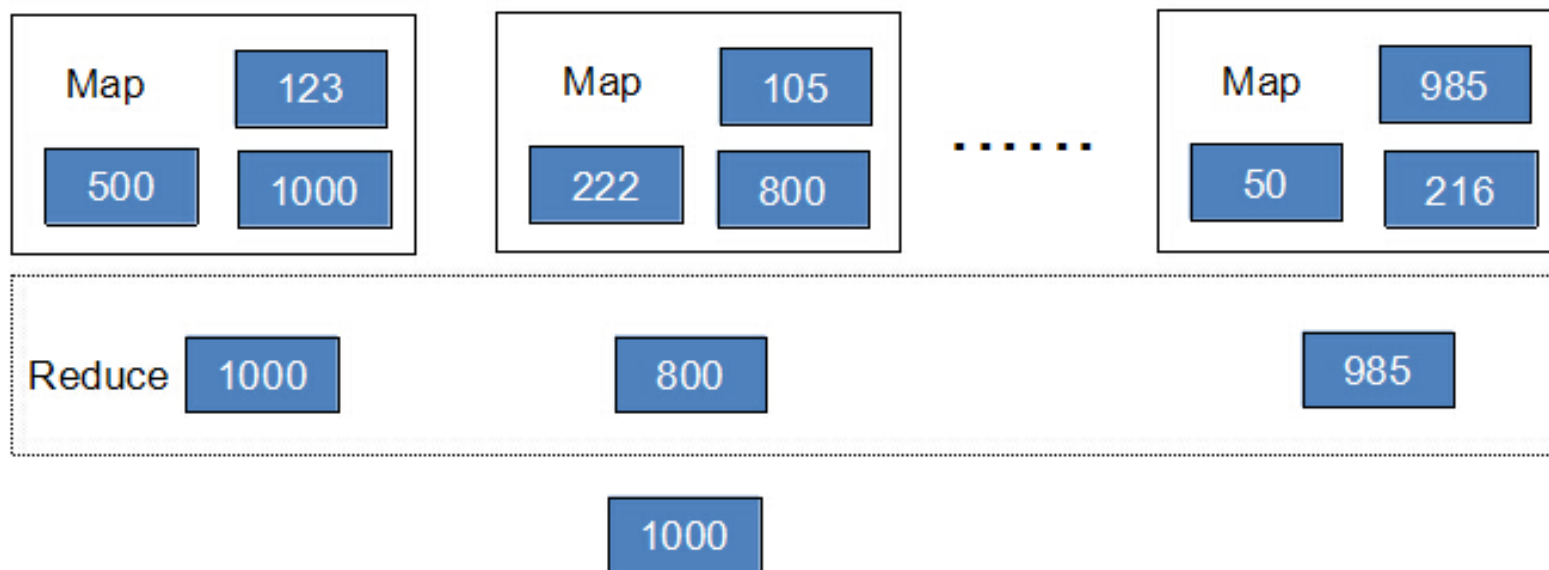
- MapReduce是一套从海量源数据提取分析最后返回结果集的编程模型，将文件分布式存储到硬盘是第一步，而从海量数据中提取分析我们需要的内容就是MapReduce做的事了。
- 例：一个银行有**上亿**储户，银行希望找到存储金额最高的金额是多少，按照传统的计算方式，我们会这样：

```
Long moneys[]...
Long max=0L;
for(int i=0;i<moneys.length;i++){
    if(moneys[i]>max){
        max=moneys[i];
    }
}
```

- 如果计算的数组长度少的话，这样实现是不会有问题的，还是面对海量数据的时候就会有问题。

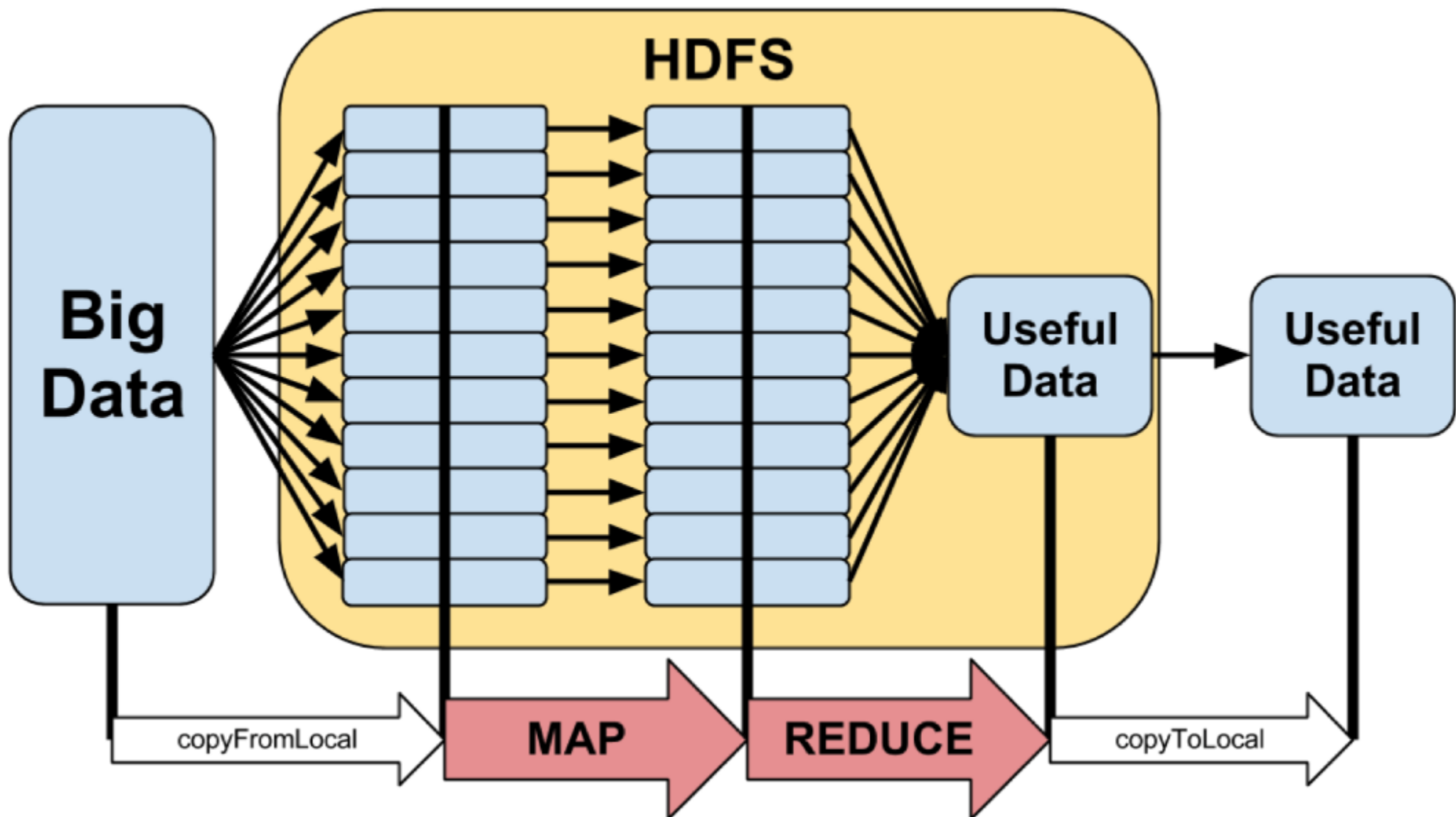
Hadoop组件讲解——Map-Reduce2

- MapReduce会这样做：首先数字是分布存储在不同块中的，以某几个块为一个Map，计算出Map中最大的值，然后将每个Map中的最大值做Reduce操作，Reduce再取最大值给用户。



- MapReduce的基本原理就是：**将大的数据分析分成小块逐个分析，最后再将提取出来的数据汇总分析，最终获得我们想要的内容。**当然怎么分块分析，怎么做Reduce操作非常复杂，Hadoop已经提供了数据分析的实现，我们只需要编写简单的需求命令即可达成我们想要的数据库。

Hadoop经典案例：词频统计



by @寒小阳(hanxiaoyang.ml@gmail.com)

总体流程

Map函数

MAP:
读取输入文本
产生一序列
键值对

The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now -- the robotics we're doing -- is what we're going to need

超大文本文档

(The, 1)
(crew, 1)
(of, 1)
(the, 1)
(space, 1)
(shuttle, 1)
(Endeavor, 1)
(recently, 1)
....

(key, value)

按照key排序:
将所有
有相同key的
键值对排在一起

(crew, 1)
(crew, 1)
(space, 1)
(the, 1)
(the, 1)
(the, 1)
(shuttle, 1)
(recently, 1)
...

(key, value)

Reduce函数

Reduce:
收集和统计
对应同一个key
的value并输出

(crew, 2)
(space, 1)
(the, 3)
(shuttle, 1)
(recently, 1)
...

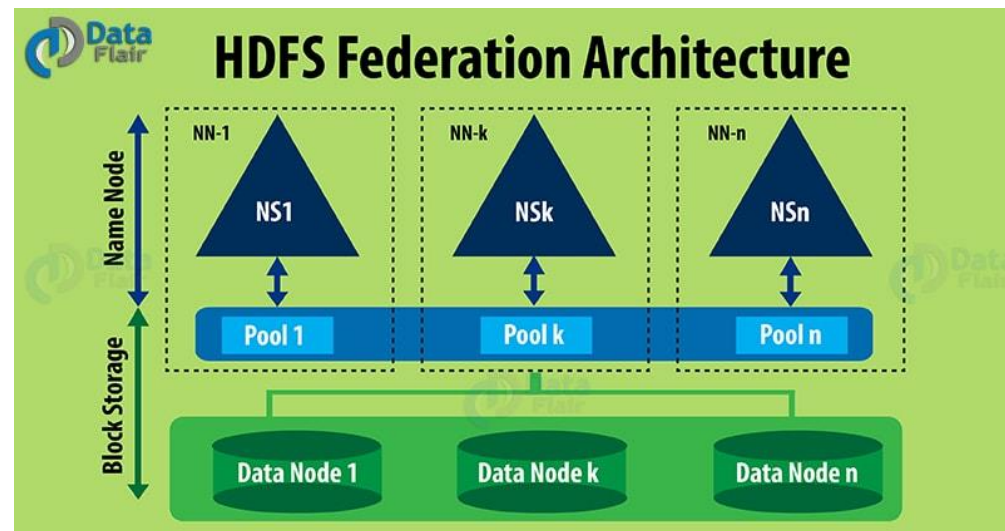
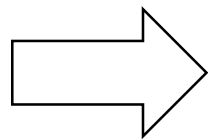
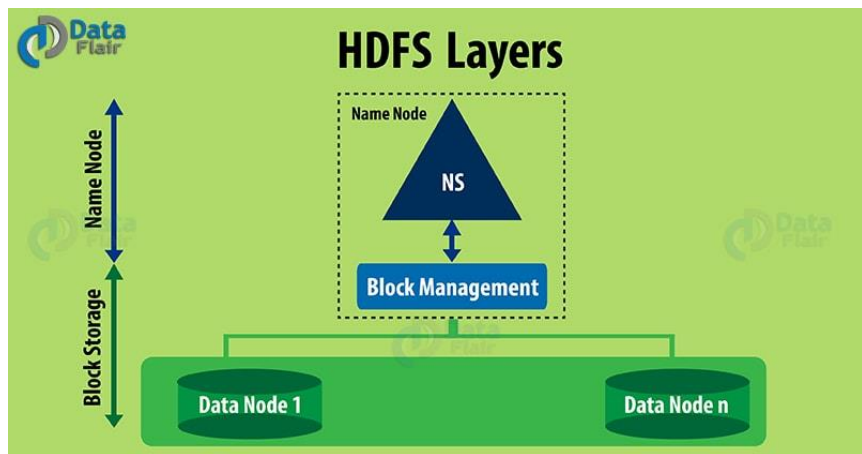
(key, value)

序列化读取

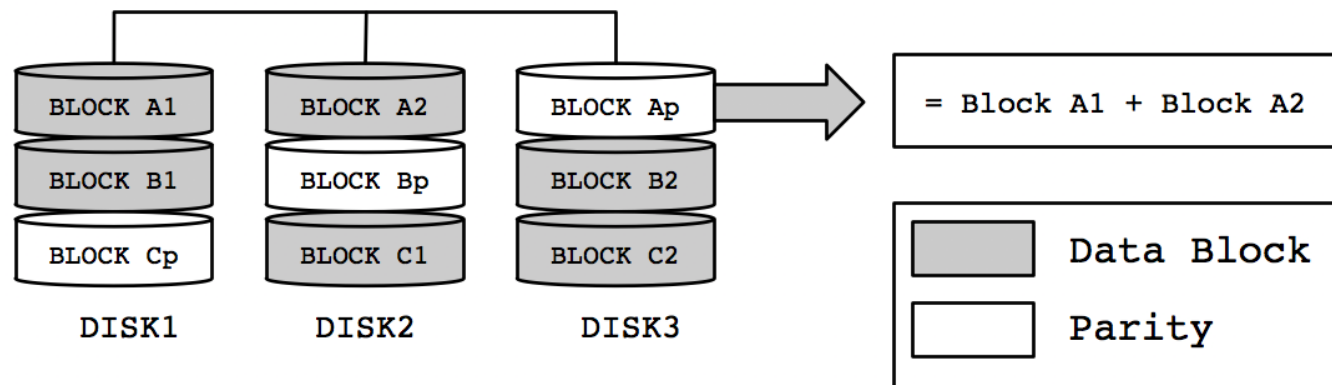
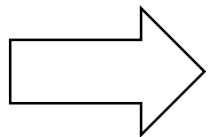
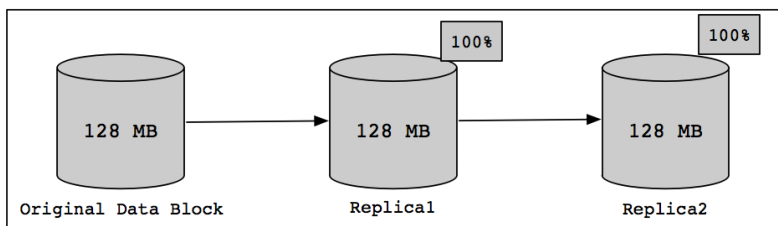
Hadoop的现在和未来

Hadoop的现在和未来——Federation与EC

● Federation

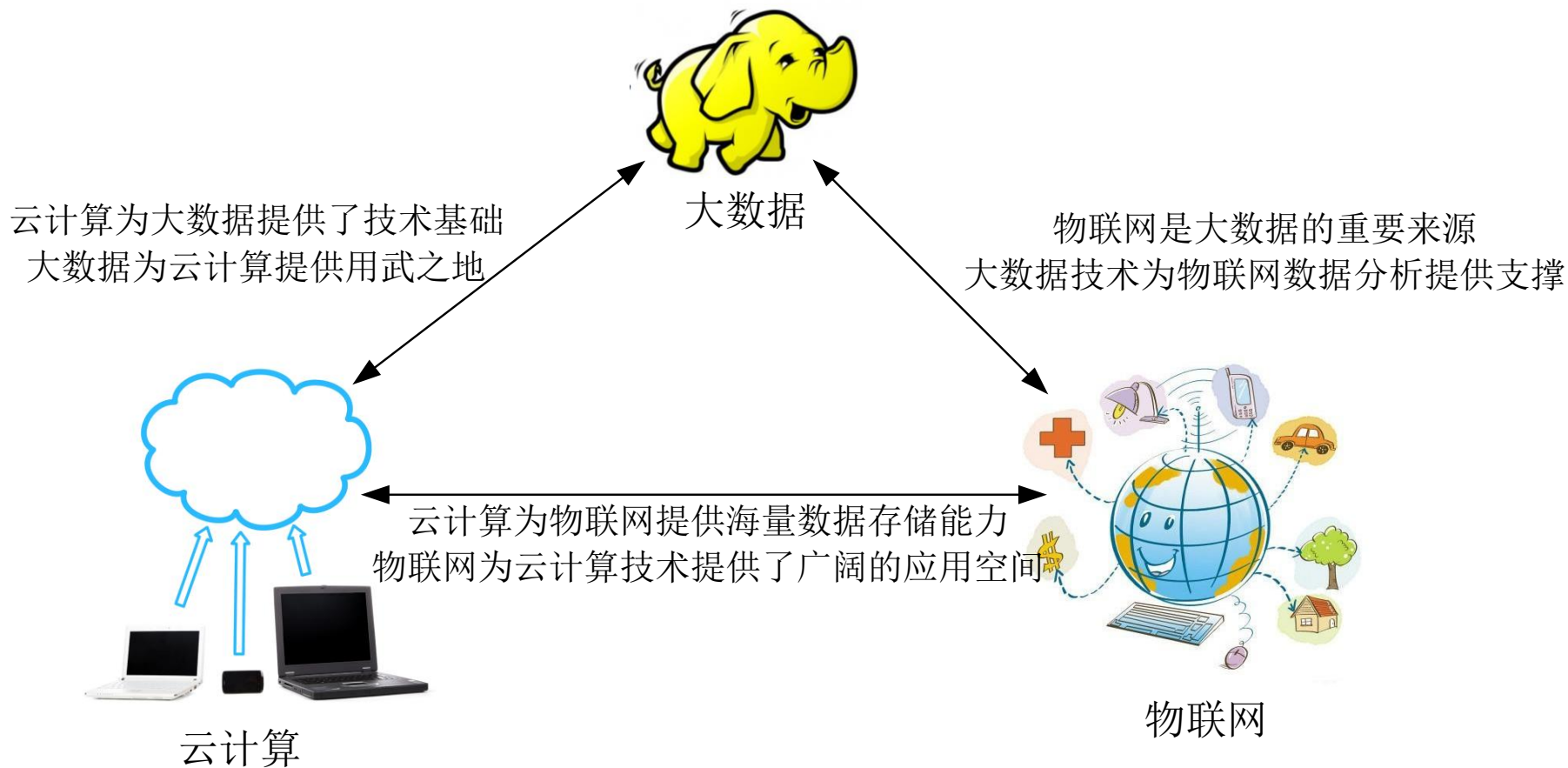


● Erasure Encoding(EC)



大数据、云计算、物联网

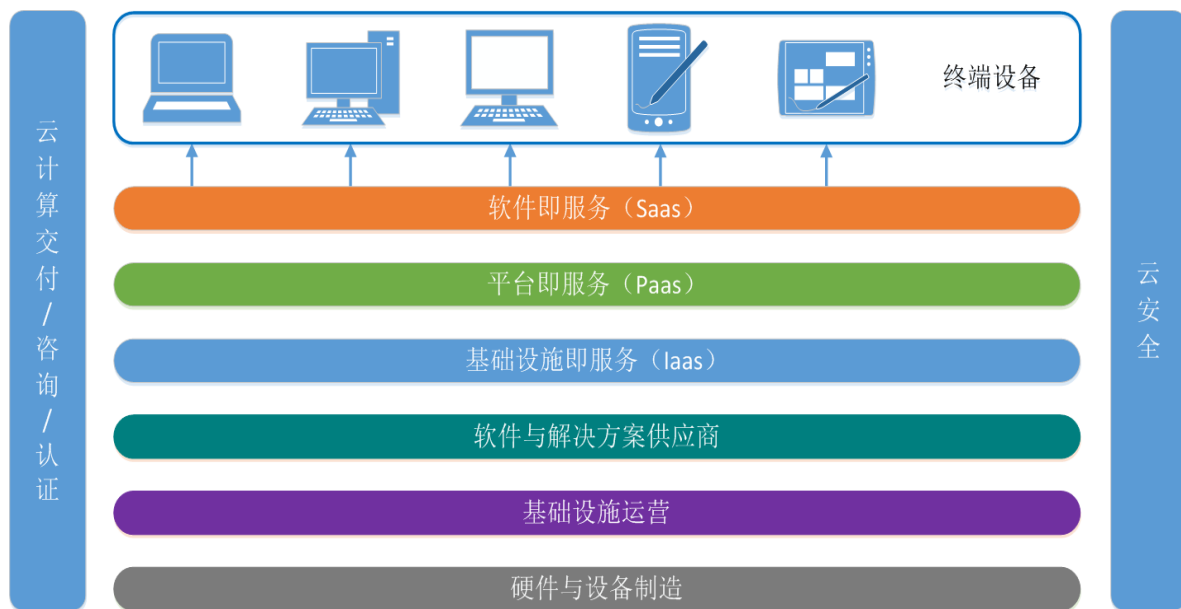
- 云计算、大数据和物联网代表了最新的技术发展趋势，三者既有区别又有联系



云计算

● 云计算概念

- 云计算实现了通过网络提供可伸缩的、廉价的分布式计算能力，用户只需要在具备网络接入条件的地方，就可以随时随地获得所需的各种IT资源



● 云计算关键技术

- 包括：虚拟化、分布式存储、分布式计算、多租户等

主流云服务商



物联网

- 物联网概念

- 物联网是物物相连的互联网，是互联网的延伸，它利用局部网络或互联网等通信技术把传感器、控制器、机器、人员和物等通过新的方式联在一起，形成人与物、物与物相联，实现信息化和远程管理控制

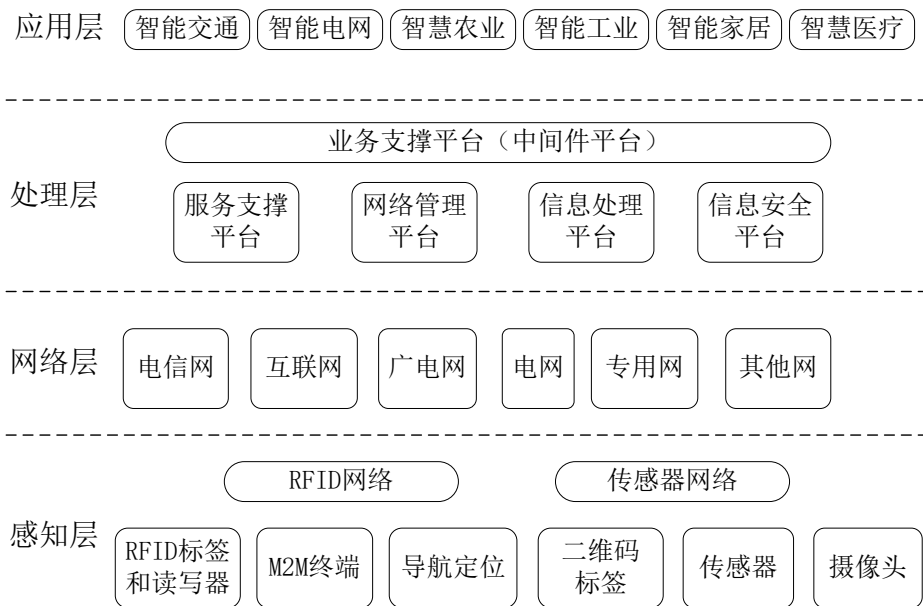
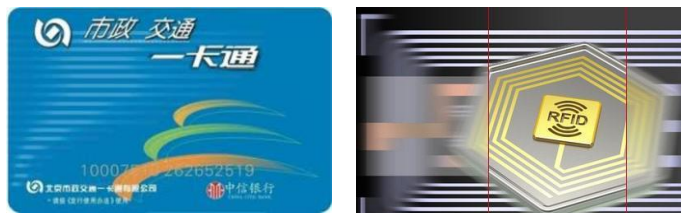
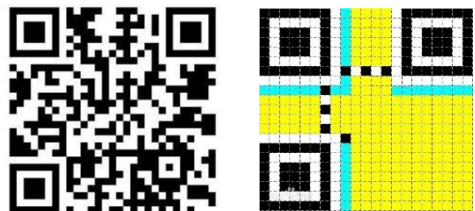


图1-9 物联网体系架构

- 物联网关键技术

- 物联网中的关键技术包括识别和感知技术（二维码、RFID、传感器等）、网络与通信技术、数据挖掘与融合技术等



(a)温湿度传感器



(b)压力传感器




(c)烟雾传感器

Hadoop的现在和未来：真*数据迁移

	强烈建议使用		建议使用		不建议使用
	10Mbps	100Mbps	1Gbps	10Gbps	
10GB	3 小时	17 分钟	2 分钟	10 秒	
100GB	30 小时	3 小时	17 分钟	2 分钟	
1TB	12.5 天	30 小时	3 小时	17 分钟	
10TB	125 天	12.5 天	30 小时	3 小时	
100TB	3.5 年	125 天	12.5 天	30 小时	
1PB	35 年	3.5 年	125 天	12.5 天	


AWS Snowball – Petabyte Scale Data Transport

Ruggedized case
"8.5G Impact"




© 2016 SoftNAS, Inc.

50 TB
10GE network



E-ink shipping label

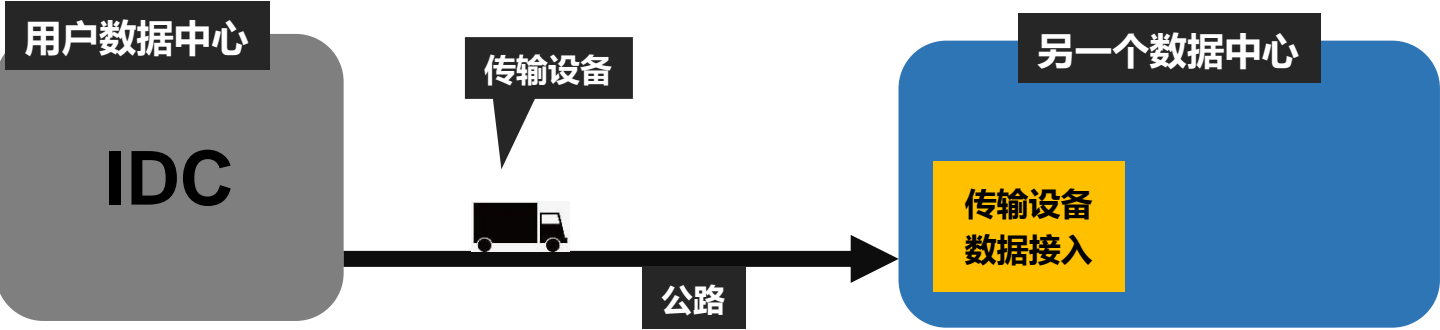
Rain & dust resistant



Tamper-resistant case & electronics

All data encrypted end-to-end

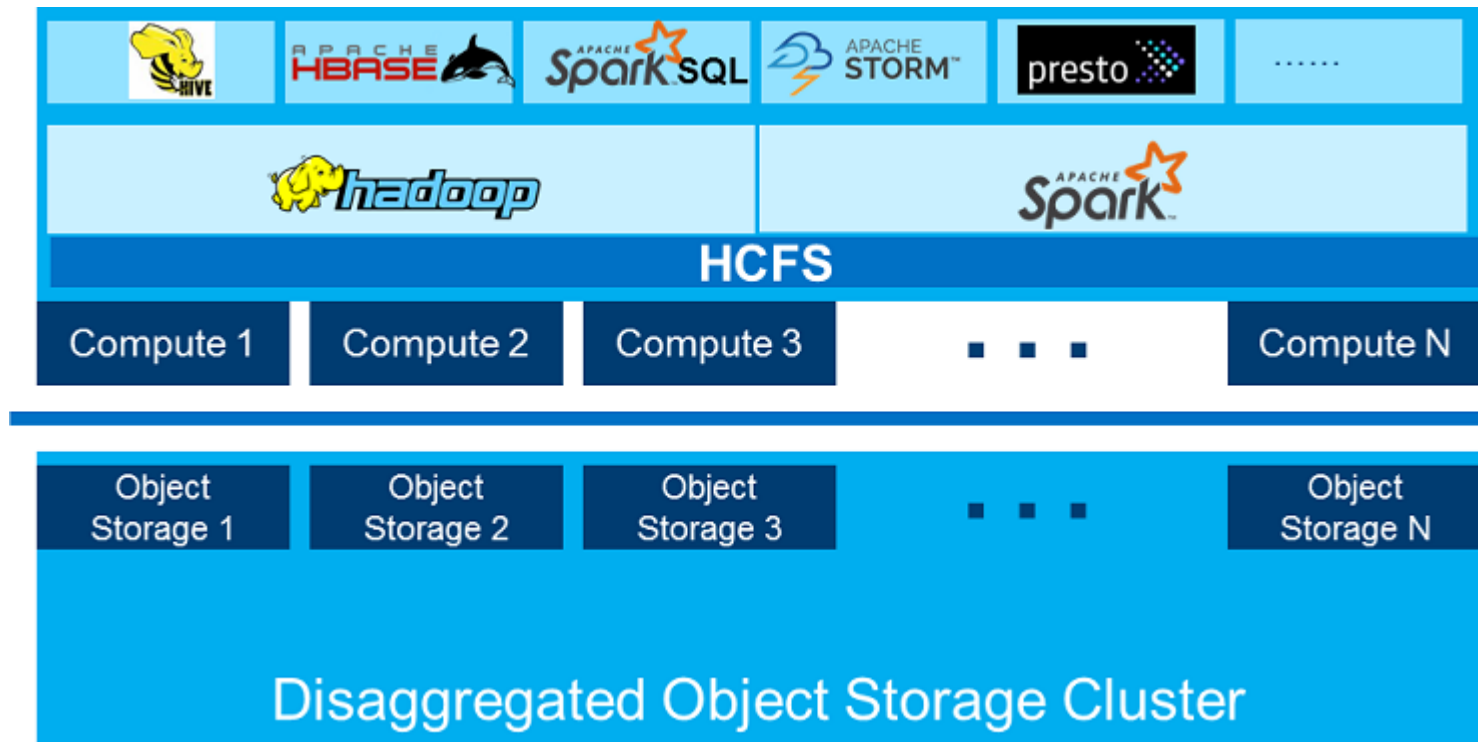
SoftNAS



Hadoop的现在和未来：存储与计算分离

- 数据本地化(Data Locality)的概念已不适用
 - 数据平衡操作成本高昂；
 - 网络性能逐年提高已经不成为性能瓶颈；
 - 存储/计算Co-location架构也无法保证高数据本地化率(30% in Facebook)；
- 存储需求与计算需求增长不对称，大数据集群扩容后存在计算资源使用率低
 - 传统Hadoop架构中存储资源与计算资源是绑定在一起的，因此当组织需要更多的存储资源，他们必须购买可能不需要的计算资源。长期下去，这种购买模式会导致越来越多的计算资源闲置，对IT预算造成浪费。
- 趋势：存储与计算解耦、中心化共享存储式架构
 - 简化管理、降低成本、改善使用率；
 - 共享式存储方便数据分析类产品协同
 - 改善数据保护及安全

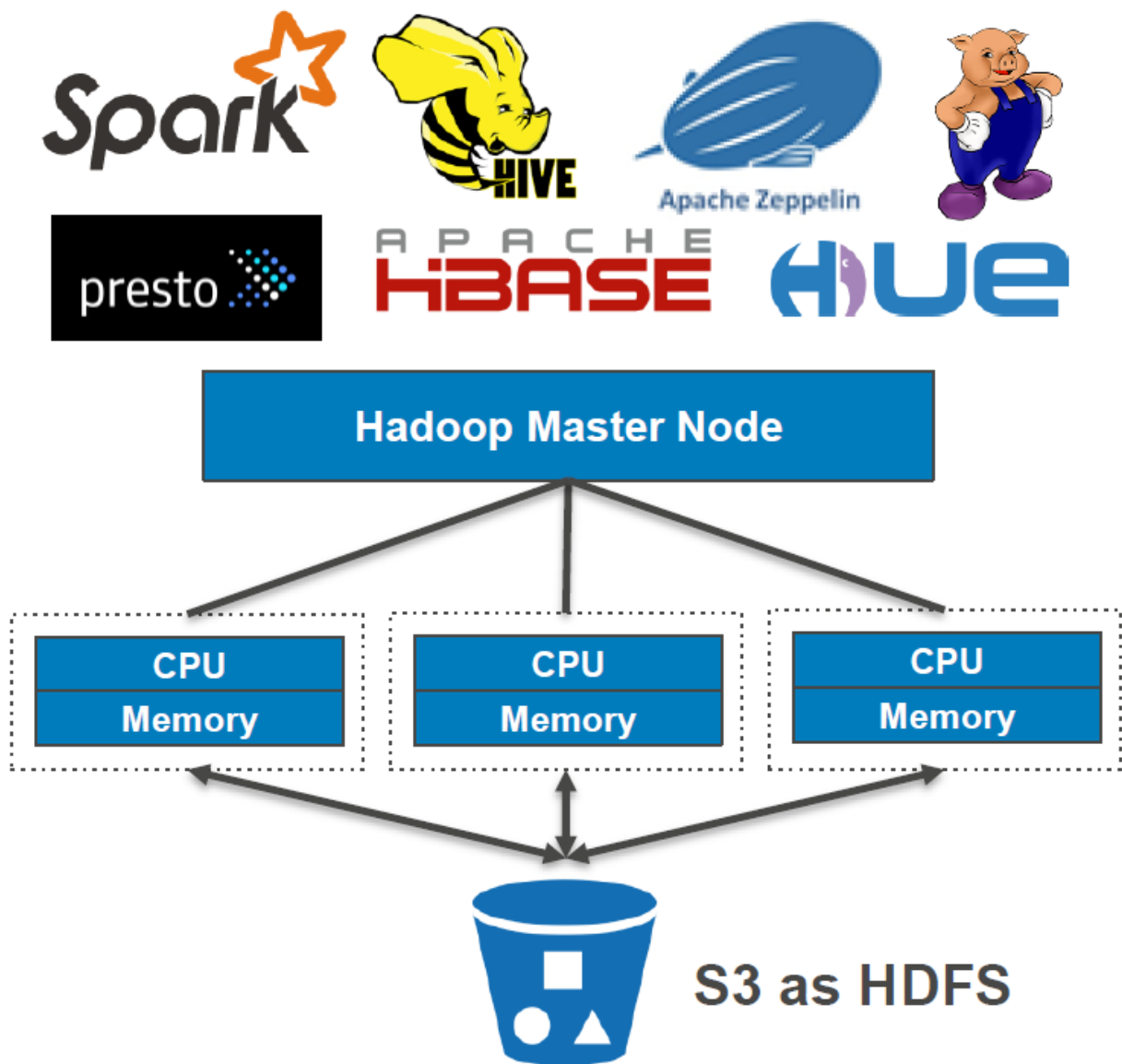
计算与存储解耦 - BDaaS参考架构



备注：Hadoop Compatible File System (HCFS)

参考：Unlock Big Data Analytics Efficiency with Compute and Storage Disaggregation on Intel® Platforms

Hadoop的现在和未来：EMR

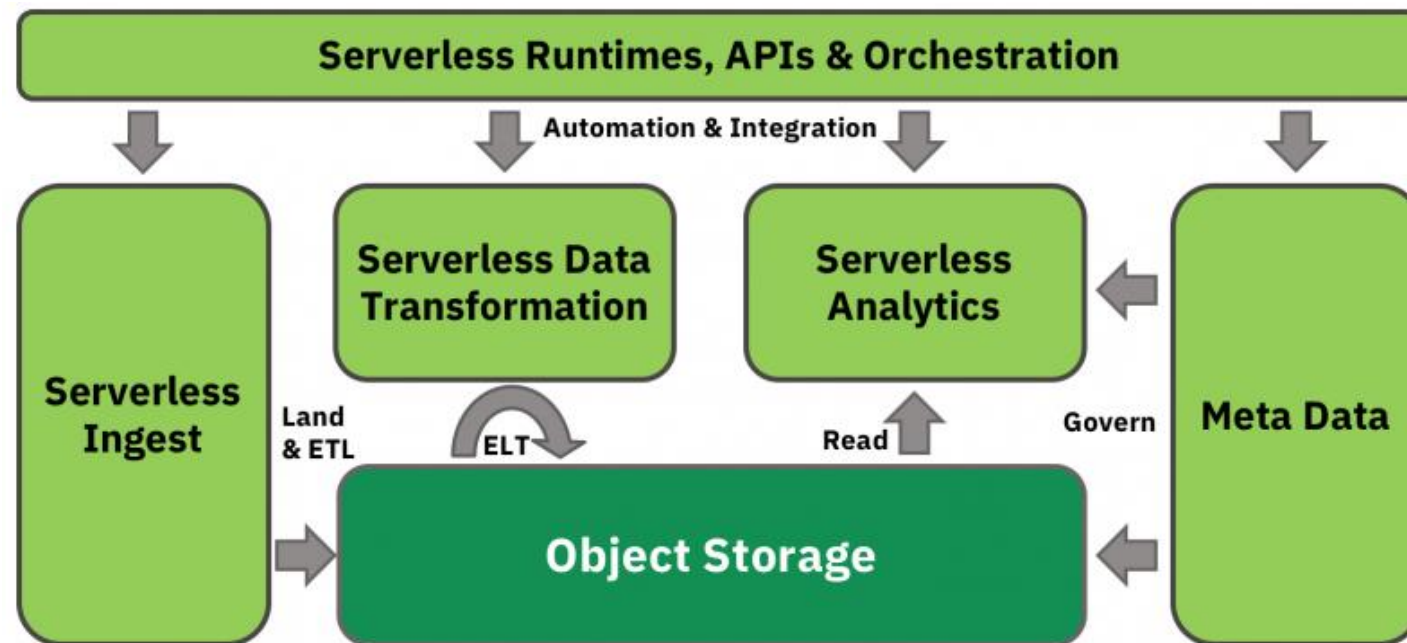


- 存储与计算分离
- CPU与内存资源独立可弹性伸缩
- 数据存储的对象存储上，不需要存储三倍的数据
- 可以根据特定的需求定制特殊的作业优化
 - Spark：内存密集型
 - Hive：CPU密集型
 - Hbase：IO密集型

Hadoop的现在和未来：DBaaS与Serverless

- 趋势: BDaaS向着Serverless方向演进，进一步降低大数据分析门槛及使用成本
- 无需集群部署及运维管理成本；
- 按需付费，无闲置成本；
- 按使用弹性扩容；
- 高可用、容错

Serverless - BDaaS参考架构



参考：Data management and analytics using serverless form factors

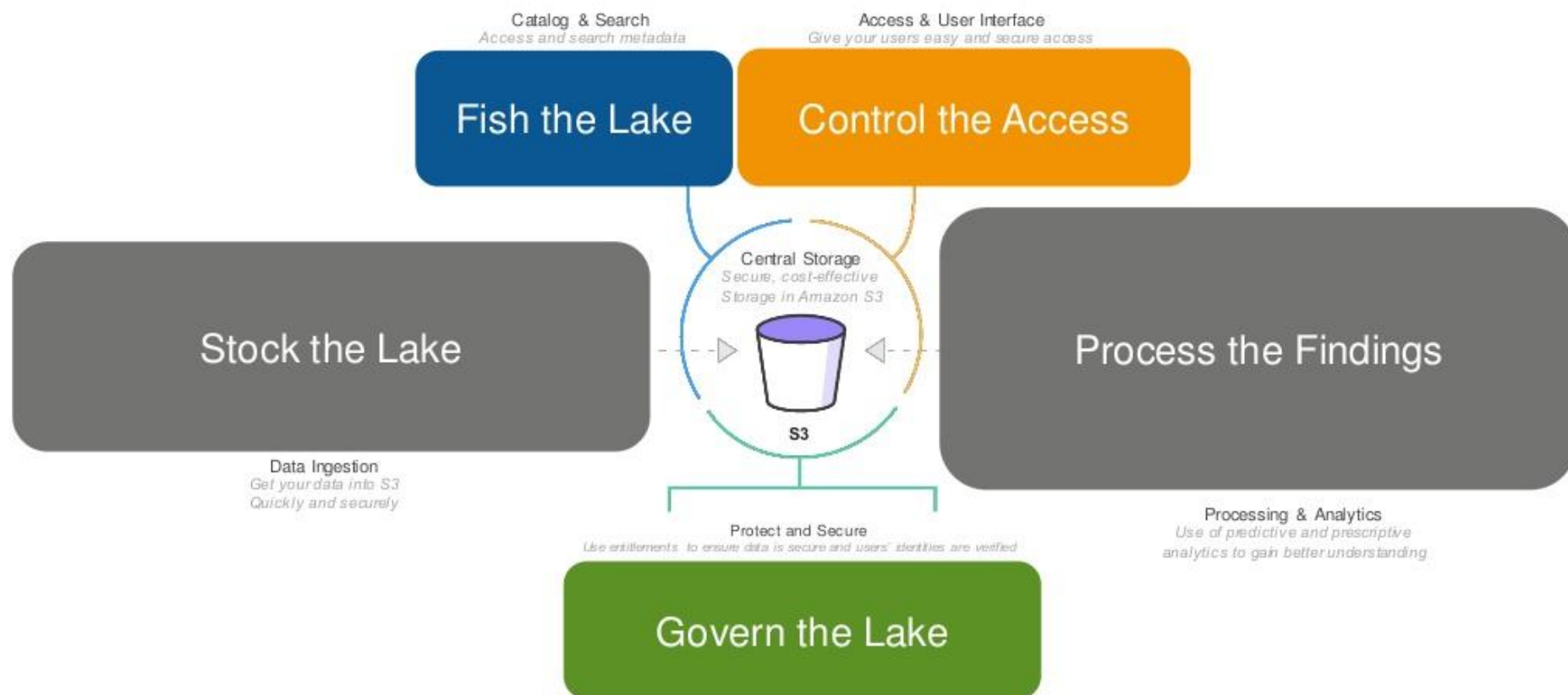
Hadoop的现在和未来：数据湖

数据湖是一种新兴起的架构方法，用于在集中式存储库中存储和分析海量异构数据。用来解决传统数据架构中存在数据孤岛、数据存储成本等问题。

AWS围绕着以**S3为中心的数据湖存储**构建了数据汇聚、数据分析、元数据管理、数据湖治理、安全与访问控制等一系列服务。

基于数据湖架构的新服务也在不断演进。

Data Lake reference architecture

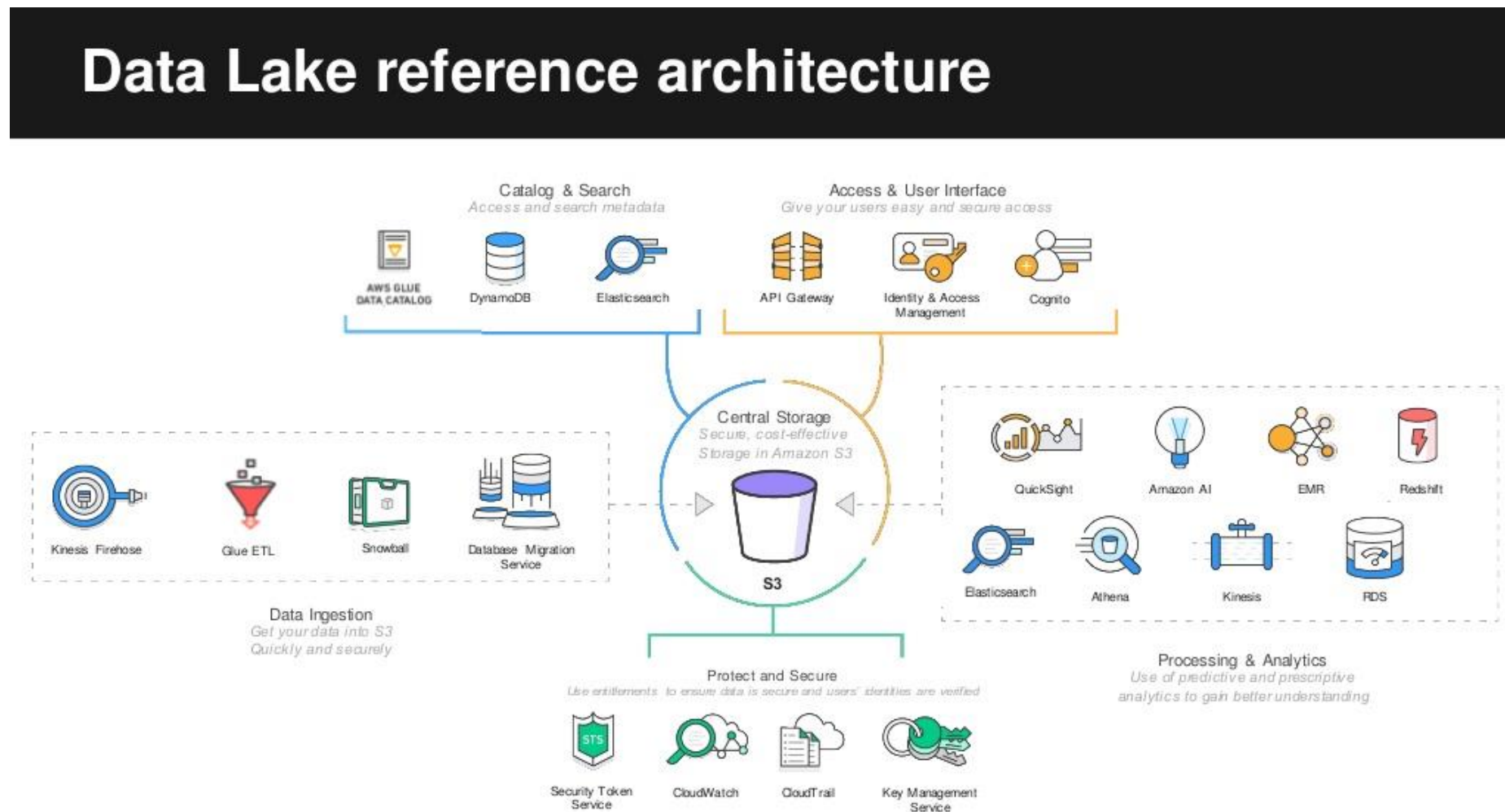


Hadoop的现在和未来：数据湖

数据湖是一种新兴起的架构方法，用于在集中式存储库中存储和分析海量异构数据。用来解决传统数据架构中存在数据孤岛、数据存储成本等问题。

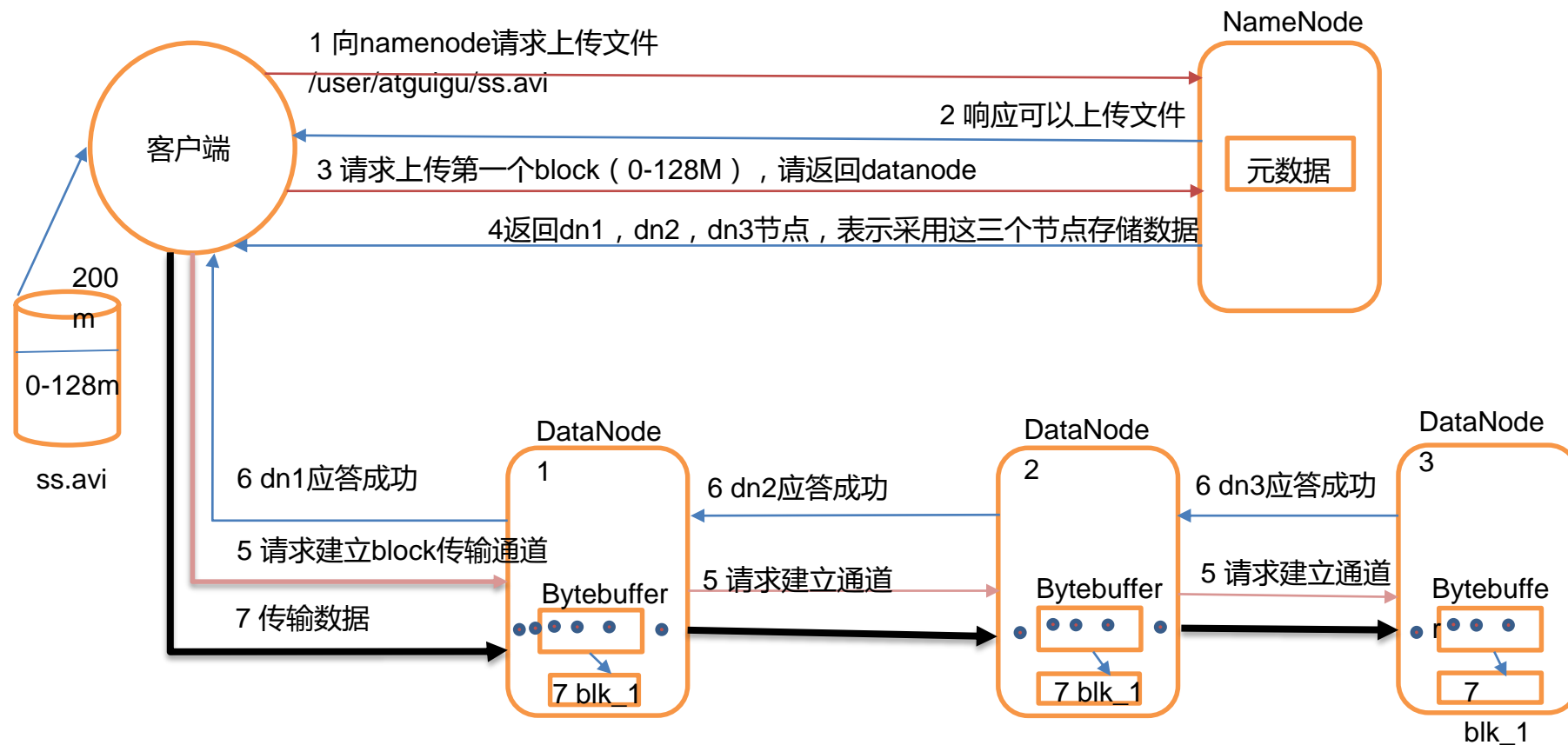
AWS围绕着以**S3为中心的数据湖存储**构建了数据汇聚、数据分析、元数据管理、数据湖治理、安全与访问控制等一系列服务。

基于数据湖架构的新服务也在不断演进。

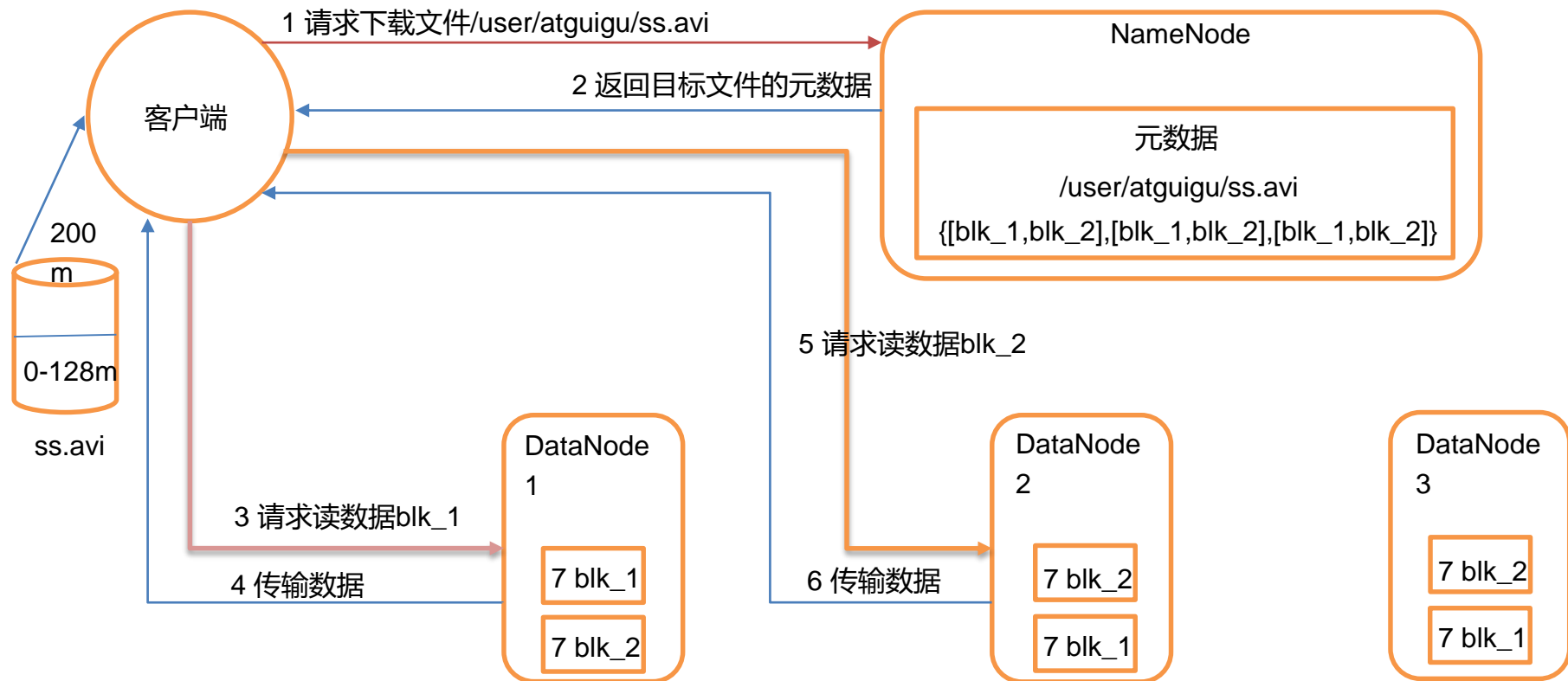


敬请指正！

HDFS写数据流程



HDFS读数据流程



Yarn的原理与架构简介

ResourceManager

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

NodeManager

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

ApplicationMaster

- 为应用程序申请资源，并分配给内部任务
- 任务调度、监控与容错

