

分布式云平台

讲师：肖斌



- Hadoop2.x
 - Hadoop 2.x

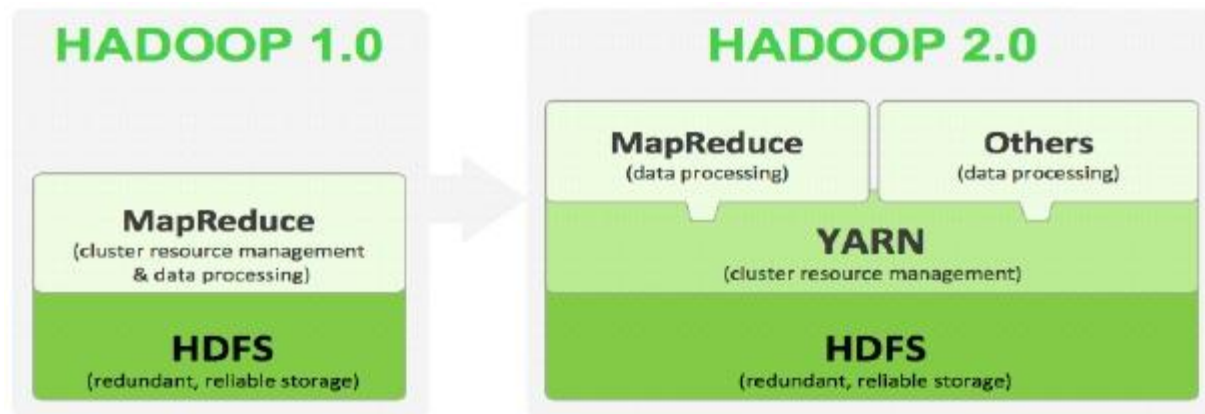


- **Hadoop 2.0产生背景**

- Hadoop 1.0中HDFS和MapReduce在高可用、扩展性等方面存在问题
- HDFS存在的问题
 - NameNode单点故障，难以应用于在线场景
 - NameNode压力过大，且内存受限，影响系统扩展性
- MapReduce存在的问题
 - JobTracker单点故障
 - JobTracker访问压力大，影响系统扩展性
 - 难以支持除MapReduce之外的计算框架，比如Spark、Storm等



- Hadoop 1.x与Hadoop 2.x



- Hadoop 2.x由HDFS、MapReduce和YARN三个分支构成；
 - HDFS：NN Federation、HA；
 - MapReduce：运行在YARN上的MR；
 - YARN：资源管理系统

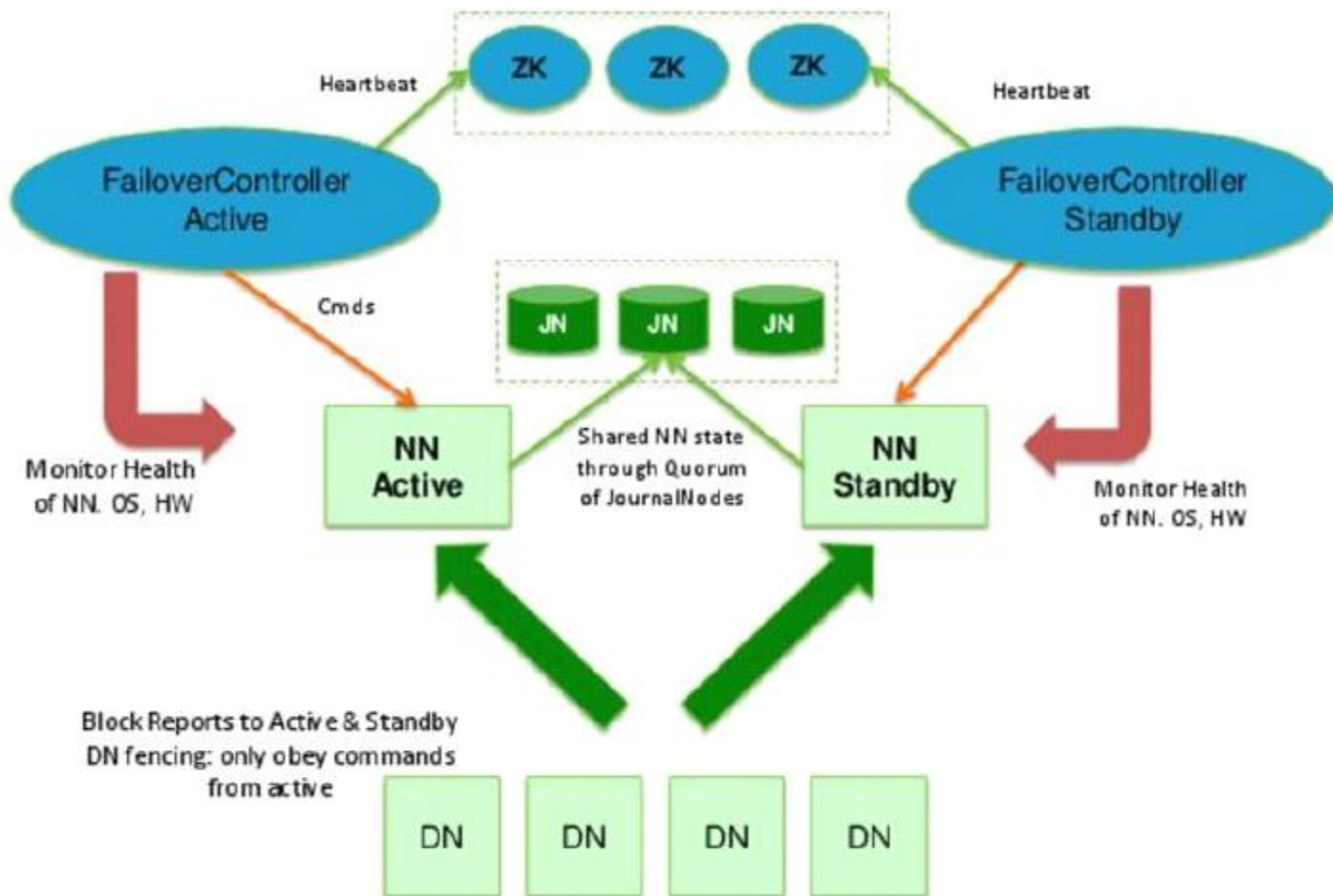


- **HDFS 2. x**

- 解决HDFS 1.0中单点故障和内存受限问题。
- 解决单点故障
 - HDFS HA：通过主备NameNode解决
 - 如果主NameNode发生故障，则切换到备NameNode上
- 解决内存受限问题
 - HDFS Federation(联邦)
 - 水平扩展，支持多个NameNode；
 - 每个NameNode分管一部分目录；
 - 所有NameNode共享所有DataNode存储资
- 2.x仅是架构上发生了变化，使用方式不变
- 对HDFS使用者透明
- HDFS 1.x中的命令和API仍可以使用



- HDFS 2.x HA



HDFS 2.0 HA

- 主备NameNode
- 解决单点故障
 - 主NameNode对外提供服务，备NameNode同步主NameNode元数据，以待切换
 - 所有DataNode同时向两个NameNode汇报数据块信息
- 两种切换选择
 - 手动切换：通过命令实现主备之间的切换，可以用HDFS升级等场合
 - 自动切换：基于Zookeeper实现
- 基于Zookeeper自动切换方案
 - Zookeeper Failover Controller：监控NameNode健康状态，
 - 并向Zookeeper注册NameNode
 - NameNode挂掉后，ZKFC为NameNode竞争锁，获得ZKFC 锁
 - 的NameNode变为active

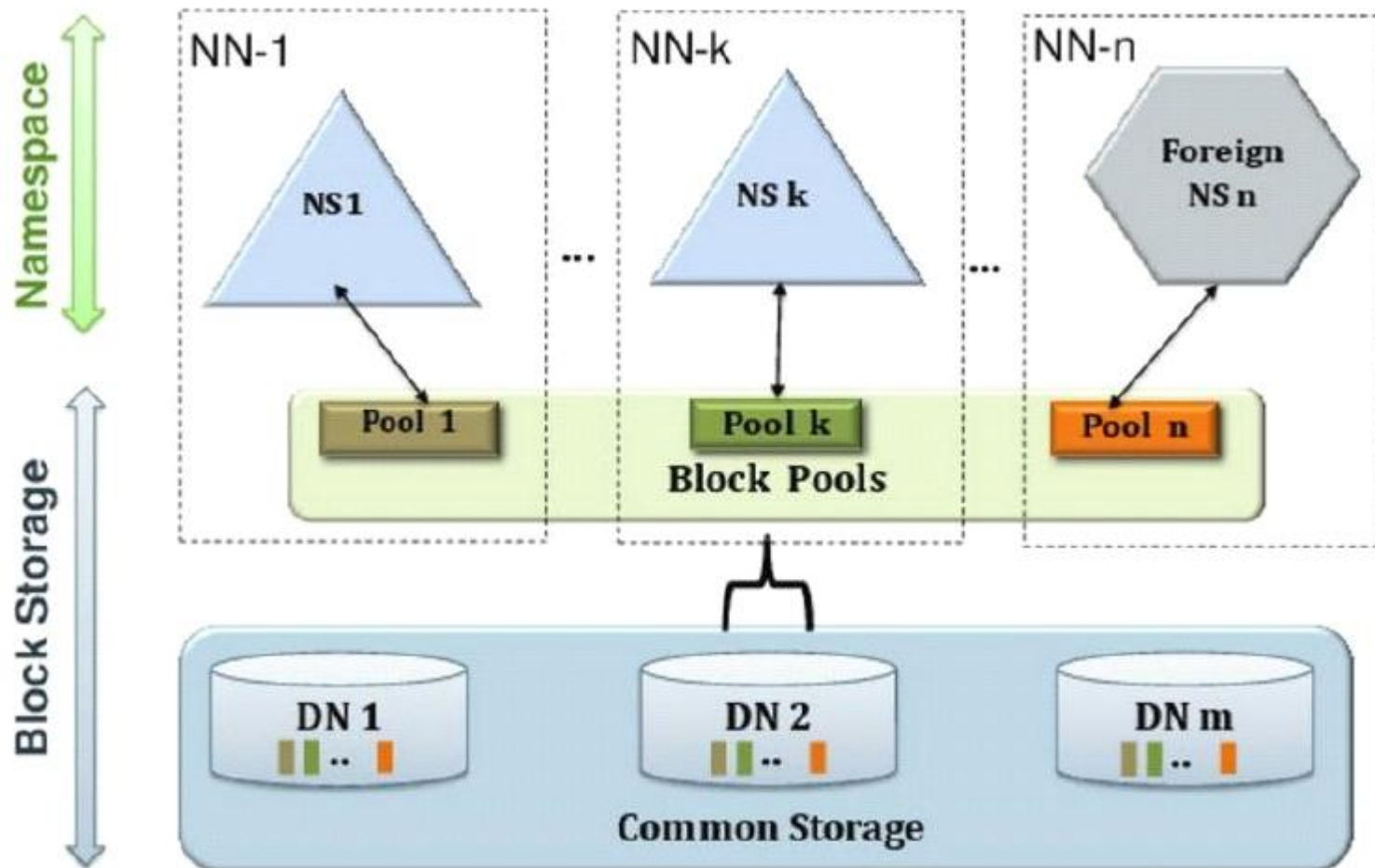


- **HDFS 2.x Federation**

- 通过多个namenode/namespace把元数据的存储和管理分散到多个节点中，使到namenode/namespace可以通过增加机器来进行水平扩展。
- 能把单个namenode的负载分散到多个节点中，在HDFS数据规模较大的时候不会也降低HDFS的性能。可以通过多个namespace来隔离不同类型的应用，把不同类型应用的HDFS元数据的存储和管理分派到不同的namenode中。



- HDFS 2.0 Federation



- YARN

- YARN : Yet Another Resource Negotiator ;
- Hadoop 2.0新引入的资源管理系统，直接从MRv1演化而来的；
 - 核心思想：将MRv1中JobTracker的资源管理和任务调度两个功能分开，分别由ResourceManager和ApplicationMaster进程实现
 - ResourceManager：负责整个集群的资源管理和调度
 - ApplicationMaster：负责应用程序相关的事务，比如任务调度、任务监控和容错等
- YARN的引入，使得多个计算框架可运行在一个集群中
 - 每个应用程序对应一个ApplicationMaster
 - 目前多个计算框架可以运行在YARN上，比如MapReduce、Spark、Storm等



- **MapReduce On YARN**

- MapReduce On YARN : MRv2
- 将MapReduce作业直接运行在YARN上，而不是由JobTracker和TaskTracker构建的MRv1系统中
- 基本功能模块
 - YARN：负责资源管理和调度
 - MRAppMaster：负责任务切分、任务调度、任务监控和容错等
 - MapTask/ReduceTask：任务驱动引擎，与MRv1一致
- 每个MapRduce作业对应一个MRAppMaster
 - MRAppMaster任务调度
 - YARN将资源分配给MRAppMaster
 - MRAppMaster进一步将资源分配给内部的任务
- MRAppMaster容错
 - 失败后，由YARN重新启动
 - 任务失败后，MRAppMaster重新申请资源



- YARN

- YARN : Yet Another Resource Negotiator ;
- Hadoop 2.0新引入的资源管理系统，直接从MRv1演化而来的；
 - 核心思想：将MRv1中JobTracker的资源管理和任务调度两个功能分开，分别由ResourceManager和ApplicationMaster进程实现
 - ResourceManager：负责整个集群的资源管理和调度
 - ApplicationMaster：负责应用程序相关的事务，比如任务调度、任务监控和容错等
- YARN的引入，使得多个计算框架可运行在一个集群中
 - 每个应用程序对应一个ApplicationMaster
 - 目前多个计算框架可以运行在YARN上，比如MapReduce、Spark、Storm等

