法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象学院
  - 新浪微博：小象AI学院

大纲

- HDFS概述
- HDFS基本架构与原理
- YARN基本架构与原理
- 实战：Hadoop集群搭建

大纲

HDFS概述

HDFS概述

- HDFS源自于Goole的GFS论文
  - 发表于2003年10月
  - HDFS是GFS克隆版
- Hadoop Distributed File System
  - 易于扩展的分布式文件系统
  - 运行在大量普通廉价机器上，提供容错机制

HDFS优点

- 高容错性
  - 数据自动保存多个副本
  - 副本丢失后，自动恢复
- 适合大数据批处理
  - 移动计算不移动数据
  - 数据位置暴露给计算框架
  - GB、TB甚至PB级别数据
  - 百万规模以上的文件数量
  - 10K节点规模

HDFS优点

- 流式文件访问
  - 一次写入，多次读取
  - 保证数据一致性
- 构建成本低、安全可靠
  - 构建在廉价机器上
  - 通过多副本提高可靠性
  - 提供了容错和恢复机制

HDFS缺点

- 不适合低延迟数据访问
- 不适合大量小文件存储
  - 占用NameNode大量内存空间
  - 磁盘寻道时间超过读取时间
- 不适合并发写入
  - 一个文件只能有一个写入者
- 不提供文件随机修改
  - 只支持追加

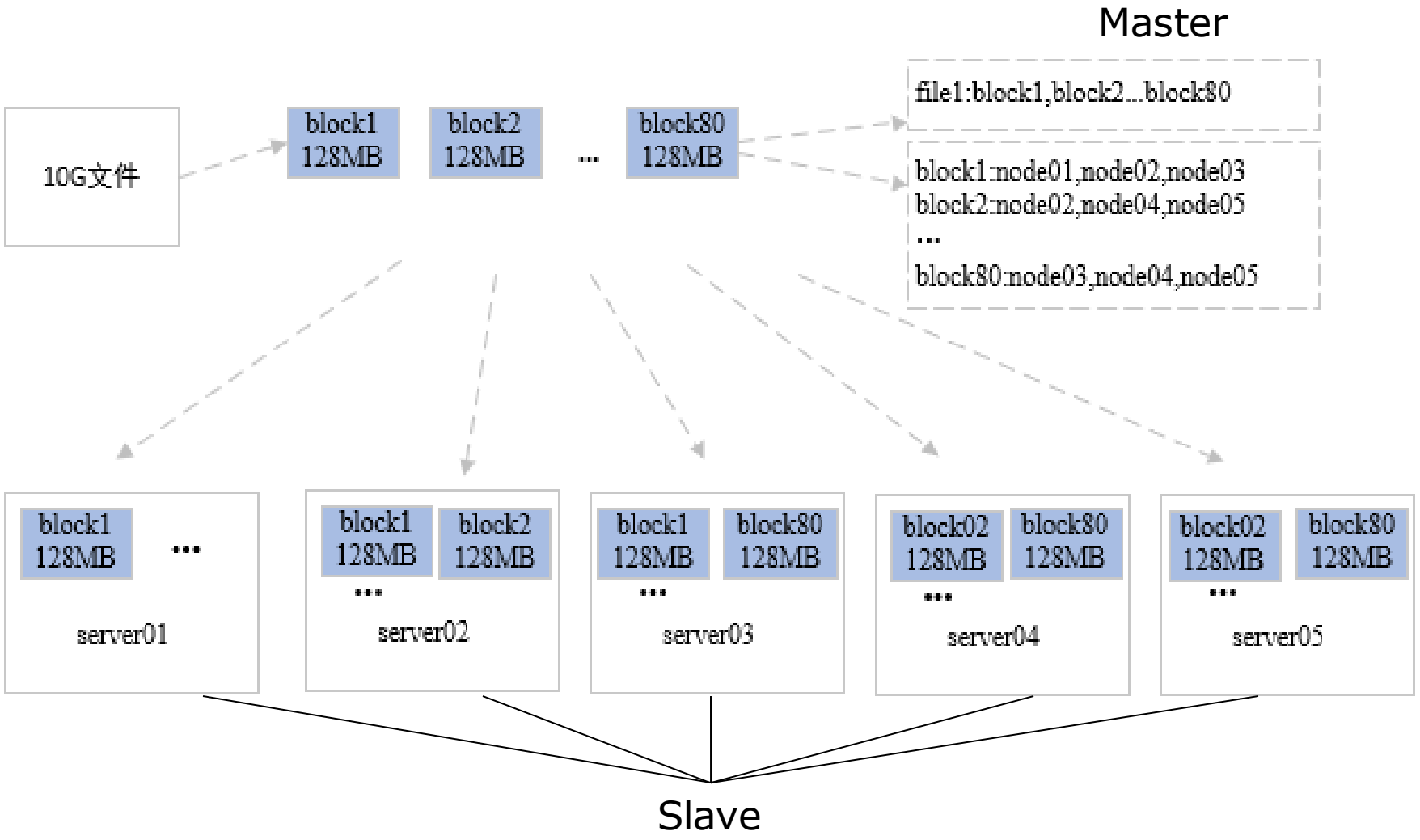
大纲

HDFS基本架构与原理

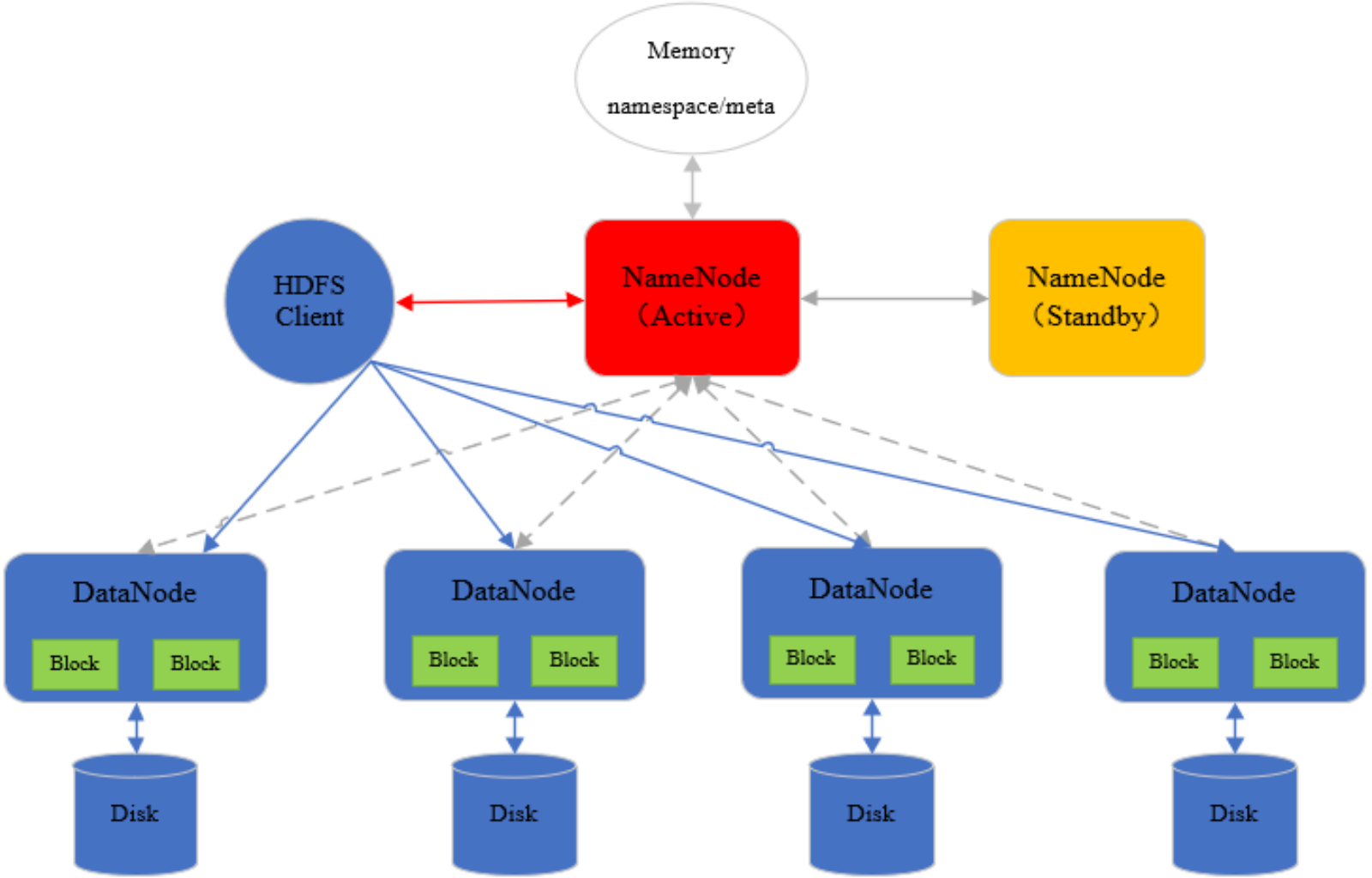
HDFS设计需求

- 超大文件
- 流式数据访问
- 低成本
- 数据一致性
- 高吞吐率
- 易扩展
- 高容错

HDFS设计实现


Master
10G文件
block1
128MB
block2
128MB
...
block80
128MB
file1:block1,block2...block80
block1:node01,node02,node03
block2:node02,node04,node05
...
block80:node03,node04,node05
block1
128MB
...
server01
block1
128MB
block2
128MB
...
server02
block1
128MB
block80
128MB
...
server03
block02
128MB
block80
128MB
...
server04
block02
128MB
block80
128MB
...
server05
Slave

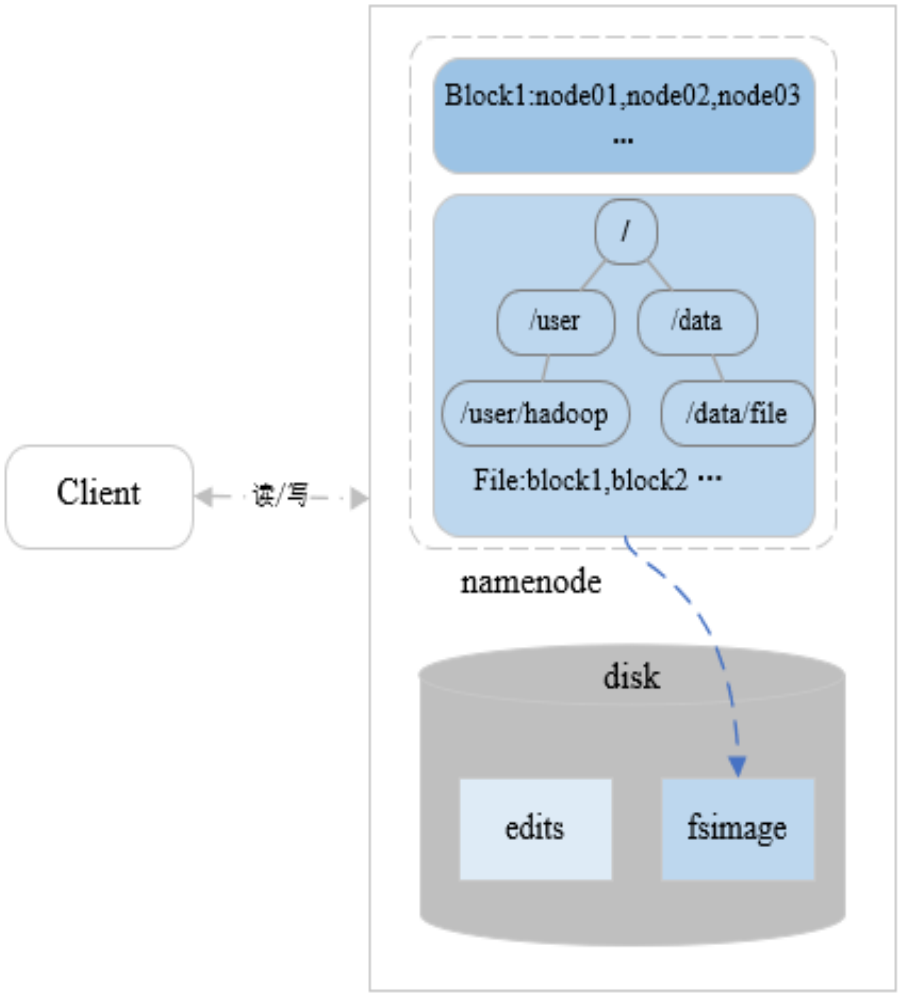
HDFS架构图


Memory
namespace/meta
HDFS
Client
NameNode
（Active）
NameNode
（Standby）
DataNode
Block
Block
Disk
DataNode
Block
Block
Disk
DataNode
Block
Block
Disk
DataNode
Block
Block
Disk

HDFS核心概念

Active NameNode

- 主Master（只有一个）
- 管理HDFS文件系统的命名空间
- 维护文件元数据信息
- 管理副本策略（默认3个副本）
- 处理客户端读写请求



HDFS核心概念

Standby NameNode

- Active NameNode的热备节点
- 周期性同步edits编辑日志，定期合并fsimage与edits到本地磁盘
- Active NameNode故障快速切换为新的Active

HDFS核心概念

- NameNode元数据文件
  - edits：编辑日志，客户端对目录和文件的写操作操首先被记到edits日志中，如：创建文件、删除文件等
  - fsimage：文件系统元数据检查点镜像文件，保存了文件系统中所有的目录和文件信息，如：一个目录下有哪些子目录、子文件，文件名，文件副本数，文件由哪些块组成等
- NameNode内存中保存一份最新的镜像信息
  - 镜像内容=fsimage + edits
- NameNode定期将内存中的新增的edits与fsimage合并保存到磁盘

HDFS核心概念

DataNode

- Slave工作节点，可以启动多个
- 存储数据块和数据校验和
- 执行客户端的读写请求操作
- 通过心跳机制定期向NameNode汇报运行状态和所有块列表信息
- 在集群启动时DataNode向NameNode提供存储的Block块列表信息

HDFS核心概念

Block数据块

- 文件写入到HDFS会被切分成若干个Block块
- 数据块大小固定，默认大小128MB，可自定义修改
- HDFS最小存储单元
- 若一个块的大小小于设置的数据块大小，则不会占用整个块的空间
- 默认情况下每个Block有三个副本

HDFS核心概念

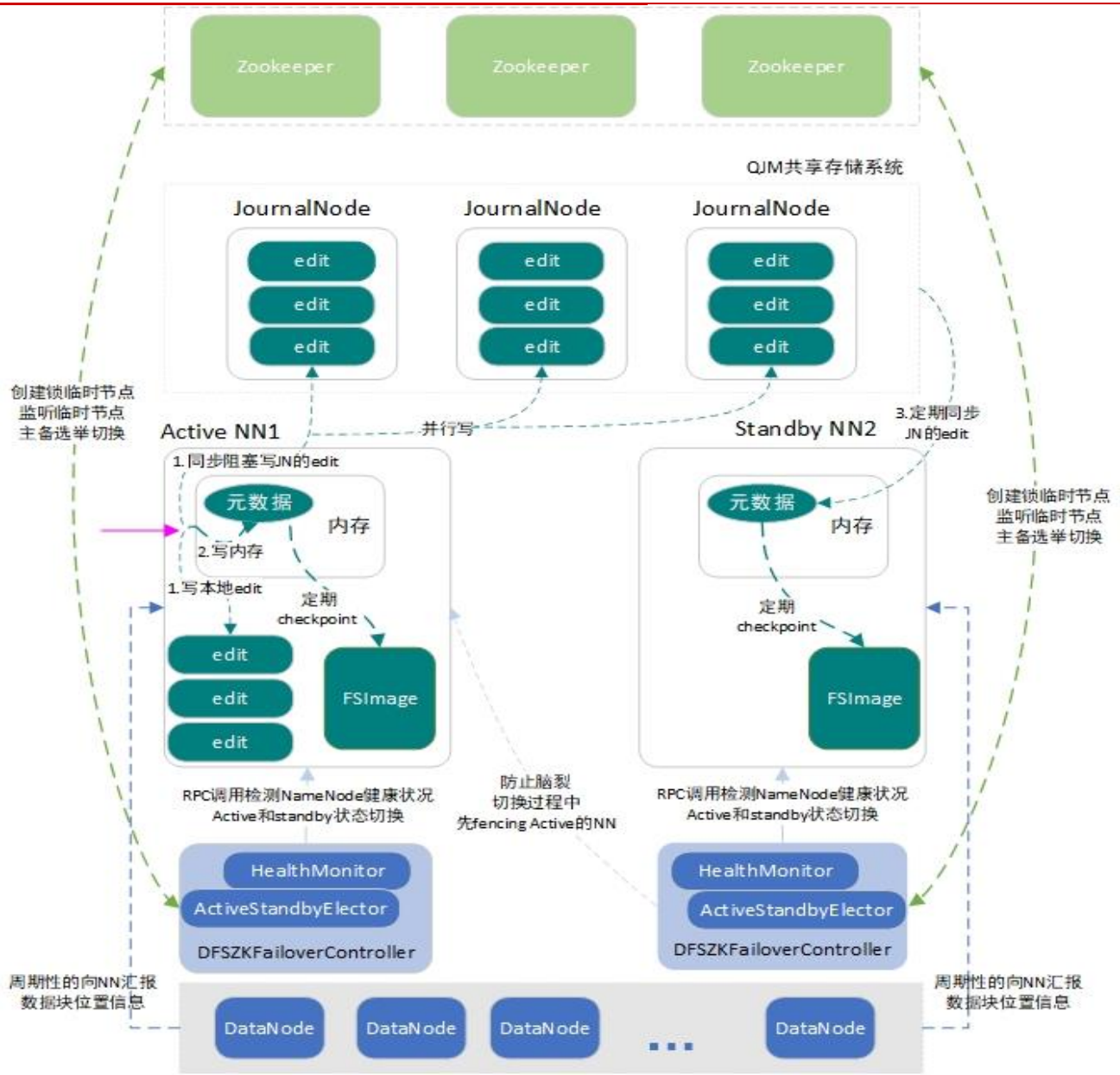
Client

- 文件切分
- 与NameNode交互获取文件元数据信息
- 与DataNode交互，读取或写入数据
- 管理HDFS

HDFS为什么不适合存储小文件

- 元数据信息存储在NameNode内存中，内存大小有限
- NameNode存储Block数目有限
  - 一个block元信息消耗大约150byte内存
  - 存储1亿个block，大约需要20GB内存
  - 如果一个文件大小为10K，则1亿个文件大小仅有1TB，却消耗NameNode 20GB内存
- 存取大量小文件消耗大量的磁盘寻道时间

HDFS高可用原理


Zookeeper
Zookeeper
Zookeeper
QJM共享存储系统
JournalNode
JournalNode
JournalNode
edit
edit
edit
edit
edit
edit
edit
edit
edit
创建锁临时节点
监听临时节点
主备选举切换
Active NN1
并行写
Standby NN2
3.定期同步
JN的edit
1.同步阻塞写JN的edit
元数据
内存
2.写内存
1.写本地edit
定期
checkpoint
edit
edit
edit
FSImage
元数据
内存
定期
checkpoint
FSImage
创建锁临时节点
监听临时节点
主备选举切换
RPC调用检测NameNode健康状况
Active和standby状态切换
防止脑裂
切换过程中
先fencing Active的NN
RPC调用检测NameNode健康状况
Active和standby状态切换
HealthMonitor
ActiveStandbyElector
DFSZKFailoverController
HealthMonitor
ActiveStandbyElector
DFSZKFailoverController
周期性的向NN汇报
数据块位置信息
周期性的向NN汇报
数据块位置信息
DataNode
DataNode
DataNode
...
DataNode

大纲

YARN基本架构与原理

YARN产生背景

- 运维成本
  - 如果采用“一个框架一个集群”的模式，则可能需要多个管理员管理这些集群，进而增加运维成本，而共享模式通常需要少数管理员即可完成多个框架的统一管理。
- 数据共享
  - 随着数据量的暴增，跨集群间的数据移动不仅需花费更长的时间，且硬件成本也会大大增加，而共享集群模式可让多种框架共享数据和硬件资源，将大大减少数据移动带来的成本。

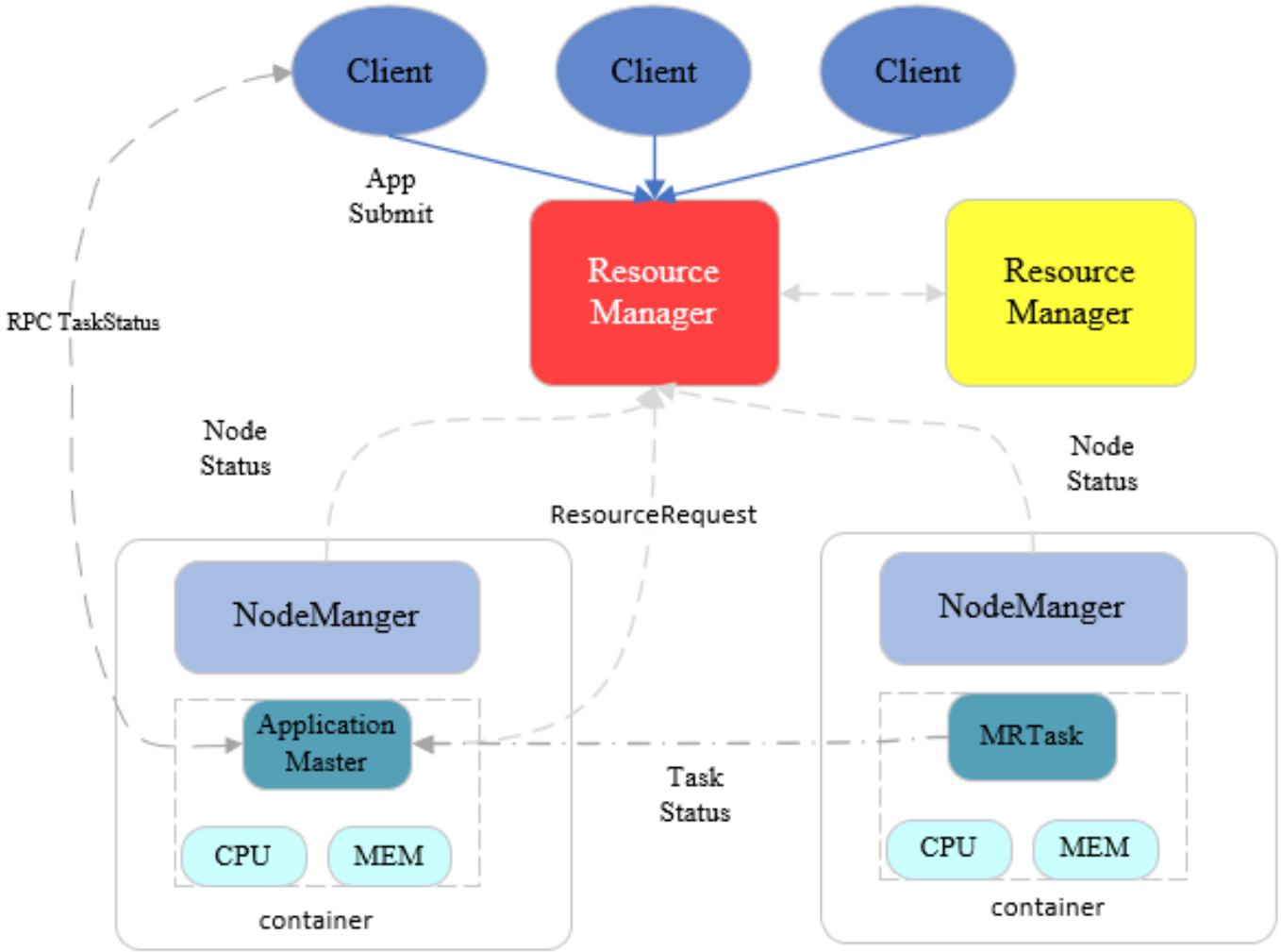
YARN产生背景

- 计算资源共享
  - 离线批处理任务主要集中在凌晨执行，资源使用有明显的波峰波谷
  - 交互式数据分析，资源使用主要集中在工作日的上班时段
  - 实时流计算、图计算、数据模型训练等各种类型的计算任务需要不同的CPU、内存等计算资源

YARN简介

- Yet Another Resource Negotiator的简称
- 从Hadoop2.0版本开始引入YARN
- YARN主要功能
  - 集群资源管理系统
  - 负责集群的统一管理和调度
  - 与客户端交互，处理客户端请求

YARN基本架构


Client
Client
Client
App
Submit
Resource
Manager
Resource
Manager
RPC TaskStatus
Node
Status
ResourceRequest
Node
Status
NodeManger
NodeManger
Application
Master
MRTask
Task
Status
CPU
MEM
CPU
MEM
container
container

YARN核心组件

ResourceManager

- 整个集群只有一个Master
- 详细功能
  - 处理客户端请求
  - 启动/监控ApplicationMaster
  - 监控NodeManager
  - 资源分配和调度

YARN核心组件

NodeManager

- 每个节点只有一个，集群中会有多个，一般与DataNode一一对应，在相同的机器上部署。
- 详细功能
  - 单个节点上的资源监控和管理
  - 定时向ResourceManager汇报本机的资源使用情况
  - 处理来自ResourceManager的请求，为作业的执行分配Container
  - 处理来自ApplicationMaster的请求，启动和停止Container

YARN核心组件

ApplicationMaster

- 每个应用程序只有一个，负责应用程序的管理，资源申请和任务调度。
- 详细功能
  - 与ResourceManager协商为应用程序申请资源
  - 与NodeManager通信启动/停止任务
  - 监控任务运行状态和失败处理

YARN核心组件

Container

- 任务运行环境的抽象，只有在分配任务的时候才会抽象出一个Container。
- 详细功能
  - 描述一系列信息
  - 任务运行资源（节点、内存、CPU）
  - 任务启动命令
  - 任务运行环境

YARN容错

YARN容错性

- ResourceManager
  - 基于Zookeeper实现高可用
- NodeManager
  - NodeManger故障将导致运行在该节点的任务失败，任务失败后，ResourceManager将失败任务通知对应的ApplicationMaster
  - ApplicationMaster决定如何处理失败的任务
- ApplicationMaster
  - ApplicationMaster失败后，由ResourceManager负责重启
  - RMAppMaster会保存已经运行完成的Task，重启后无需重新运行

大纲

实战：Hadoop集群搭建

Hadoop集群搭建

环境准备

- 安装JDK1.8
- 创建hadoop用户和组
- 配置所有机器的/etc/hosts文件，将所有机器的IP与主机名映射添加到该文件中，保证每台机器的/etc/hosts文件内容相同！
- 安装启动Zookeeper

Hadoop集群搭建

集群规划

主机名	IP	安装软件	运行的进程
node01	192.168.183.100	JDK、Hadoop Zookeeper	NameNode(Active) DFSZKFailoverController(zkfc) ResourceManager(Standby) QuorumPeerMain(zookeeper)
node02	192.168.183.101	JDK、Hadoop Zookeeper	NameNode(Standby) DFSZKFailoverController(zkfc) ResourceManager(Active) QuorumPeerMain(zookeeper) Jobhistory
node03	192.168.183.102	JDK、Hadoop Zookeeper	DataNode NodeManager JournalNode QuorumPeerMain(zookeeper)

Hadoop集群搭建

集群规划

主机名	IP	安装软件	运行的进程
node04	192.168.183.103	JDK、Hadoop	DataNode NodeManager JournalNode
node05	192.168.183.104	JDK、Hadoop	DataNode NodeManager JournalNode

注意：在集群搭建演示的过程中会启动两个namenode和两个resourcemanager，演示完成关闭node01的RS和node02的NN，节省资源

Hadoop集群搭建

集群搭建

注意：没有特别指出用root用户操作，一律使用hadoop用户操作！

1. 使用hadoop用户解压并安装到apps路径下

   - 使用hadoop用户进入到在/home/hadoop/apps目录下

   cd /home/hadoop/apps

   注意：如果没有/home/hadoop/apps路径，自行在/home/hadoop路径下创建apps文件夹：mkdir /home/Hadoop/apps

   - 使用rz将本机的hadoop安装包上传到/home/hadoop/apps目录下
   - 解压安装文件

   tar -zxvf hadoop-2.7.4.tar.gz

Hadoop集群搭建

- 使用root用户创建软链接

ln -s /home/hadoop/apps/hadoop-2.7.4 /usr/local/hadoop

- 使用root用户修改软链接属主

chown -R hadoop:hadoop /usr/local/hadoop

Hadoop集群搭建

注意：Hadoop配置文件路径是/usr/local/hadoop/etc/hadoop

- 使用root用户将hadoop相关内容添加到环境变量中

vim /etc/profile

添加内容如下：

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_HOME=$HADOOP_HOME
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin
```

- 使用root用户重新编译环境变量使配置生效

source /etc/profile

Hadoop集群搭建

2. 配置HDFS

2.1使用hadoop用户进入到Hadoop配置文件路径

cd /usr/local/hadoop/etc/hadoop

2.2修改hadoo-env.sh

修改JDK路径export JAVA_HOME=/usr/local/jdk

2.3 配置core-site.xml

2.4 配置hdfs-site.xml

注意：Hadoop配置文件路径是/usr/local/hadoop/etc/hadoop, 由于pdf文件无法粘贴复制内容所以core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml这四个配置文件会在qq群文件中提供

Hadoop集群搭建

3. 配置YARN

3.1 修改yarn-site.xml

3.2 修改mapred-site.xml

3.3 在/usr/local/hadoop路径下创建hdpdata文件夹

cd /usr/local/hadoop

mkdir hdpdata

注意：Hadoop配置文件路径是/usr/local/hadoop/etc/hadoop, 由于pdf文件无法粘贴复制内容所以core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml这四个配置文件会在qq群文件中提供

Hadoop集群搭建

4. 修改slaves文件，设置datanode和nodemanager启动节点主机名称

- 在slaves文件中添加节点的主机名称，内容如下黑体字，红体字不要加入到slaves文件中。

node03

node04

node05

注意：node03，node04，node05是我的虚拟机主机名称，在这三台机器上启动datanode和nodemanager，同学根据自己集群主机名称情况自行修改。

Hadoop集群搭建

5. 配置hadoop用户免密码登陆

配置node01到node01、node02、node03、node04、node05的免密码登陆

- 在node01上生产一对钥匙

ssh-keygen -t rsa

- 将公钥拷贝到其他节点，包括自己本机

ssh-copy-id -i node01

ssh-copy-id -i node02

ssh-copy-id -i node03

ssh-copy-id -i node04

ssh-copy-id -i node05

Hadoop集群搭建

配置node02到node01、node02、node03、node04、node05的免密码登陆

- 在node02上生产一对钥匙

ssh-keygen -t rsa

- 将公钥拷贝到其他节点，包括自己本机

ssh-copy-id -i node01

ssh-copy-id -i node02

ssh-copy-id -i node03

ssh-copy-id -i node04

ssh-copy-id -i node05

注意：两个namenode之间要配置ssh免密码登陆

Hadoop集群搭建

6. 将配置好的hadoop拷贝到其他节点

scp -r hadoop-2.7.4 hadoop@node02:/home/hadoop/apps

scp -r hadoop-2.7.4 hadoop@node03:/home/hadoop/apps

scp -r hadoop-2.7.4 hadoop@node04:/home/hadoop/apps

scp -r hadoop-2.7.4 hadoop@node05:/home/hadoop/apps

在每个节点**分别**执行如下四步操作

第一步：使用root用户创建软链接

ln -s /home/hadoop/apps/hadoop-2.7.4 /usr/local/hadoop

第二步：使用root用户修改软链接属主

chown -R hadoop:hadoop /usr/local/hadoop

Hadoop集群搭建

第三步：使用root用户添加环境变量

vim /etc/profile

添加内容：

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

export YARN_HOME=$HADOOP_HOME

export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop

export PATH=$PATH:${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin

第四步：使用root用户重新编译环境变量使配置生效

source /etc/profile

Hadoop集群搭建

集群启动步骤

（注意使用hadoop用户启动，严格按照顺序启动）

1. 启动journalnode（分别在node03、node04、node05上执行启动）

   /usr/local/hadoop/sbin/hadoop-daemon.sh start journalnode

   运行jps命令检验，node03、node04、node05上多了JournalNode进程

2. 格式化HDFS

   - 在node01上执行命令:

   hdfs namenode -format

   - 格式化成功之后会在core-site.xml中的hadoop.tmp.dir指定的路径下生成dfs文件夹，将该文件夹拷贝到node02的相同路径下

   scp -r hdpdata hadoop@node02:/usr/local/hadoop

Hadoop集群搭建

3. 在node01上执行格式化ZKFC操作

hdfs zkfc -formatZK

- 执行成功，日志输出如下信息

INFO ha.ActiveStandbyElector: Successfully created /hadoop-ha/ns in ZK

4. 在node01上启动HDFS

sbin/start-dfs.sh

5. 在node02上启动YARN

sbin/start-yarn.sh

在node01单独启动一个ResourceManger作为备份节点

sbin/yarn-daemon.sh start resourcemanager

Hadoop集群搭建

6. 在node02上启动JobHistoryServer

sbin/mr-jobhistory-daemon.sh start historyserver

启动完成node02会增加一个JobHistoryServer进程

7. hadoop安装启动完成

- HDFS HTTP访问地址

NameNode (active)：http://192.168.183.100:50070

NameNode (standby)：http://192.168.183.101:50070

- ResourceManager HTTP访问地址

ResourceManager：http://192.168.183.101:8088

- 历史日志HTTP访问地址

JobHistoryServer：http://192.168.183.101:19888

Hadoop集群搭建

集群验证

1. 验证HDFS 是否正常工作及HA高可用

- 首先向hdfs上传一个文件

```
hadoop fs -put /usr/local/hadoop/README.txt /
```

- 在active节点手动关闭active的namenode

```
sbin/hadoop-daemon.sh stop namenode
```

- 通过HTTP 50070端口查看standby namenode的状态是否转换为active

手动启动上一步关闭的namenode

```
sbin/hadoop-daemon.sh start namenode
```

Hadoop集群搭建

2.验证YARN是否正常工作及ResourceManager HA高可用

- 运行测试hadoop提供的demo中的WordCount程序：

  hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /wordcount/input /wordcount/output

- 验证ResourceManager HA

  手动关闭node02的ResourceManager

  sbin/yarn-daemon.sh stop resourcemanager

  通过HTTP 8088端口访问node01的ResourceManager查看状态

  手动启动node02 的ResourceManager

  sbin/yarn-daemon.sh start resourcemanager

疑问

- 小象问答官网
  - http://wenda.chinahadoop.cn

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象学院
- 新浪微博：小象AI学院