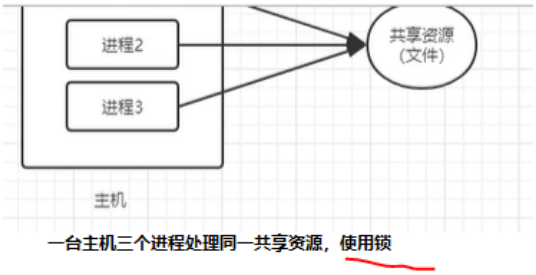
https://www.bilibili.com/video/BV1ek4y117Yq?p=146 (https://www.bilibili.com/video/BV1ek4y117Yq?p=146)

Google三篇论文 1、《分布式文件系统》 2、《分布式计算模型》 3、《BigTable》

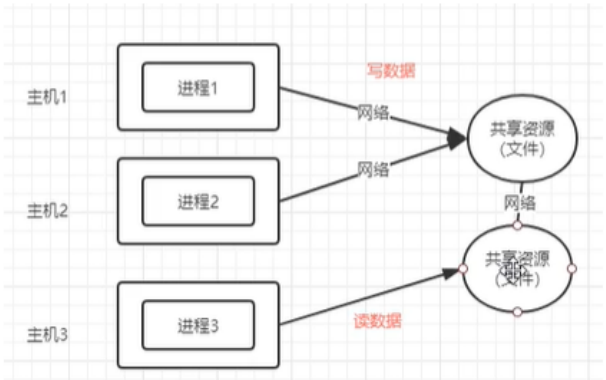
hadoop的三大核心：1.分布式存储 2.分布式计算 3.yarn-分布式资源管理调度模块

In [ ]:
```
1
```

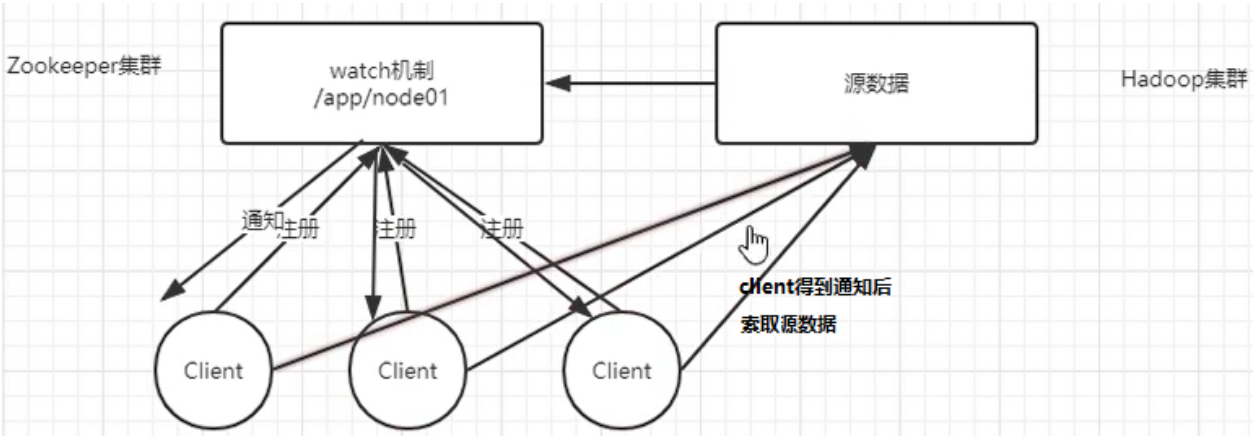


一台主机三个进程处理同一共享资源，使用锁

同一主机访问共享资源不须使用网络，不同主机则需要网络

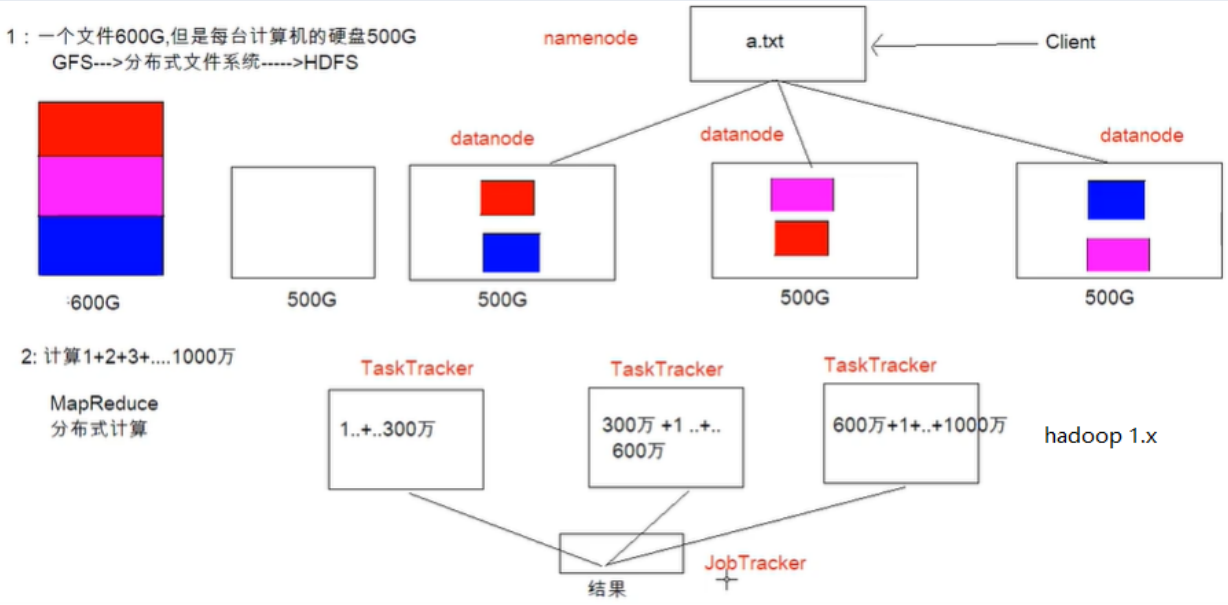


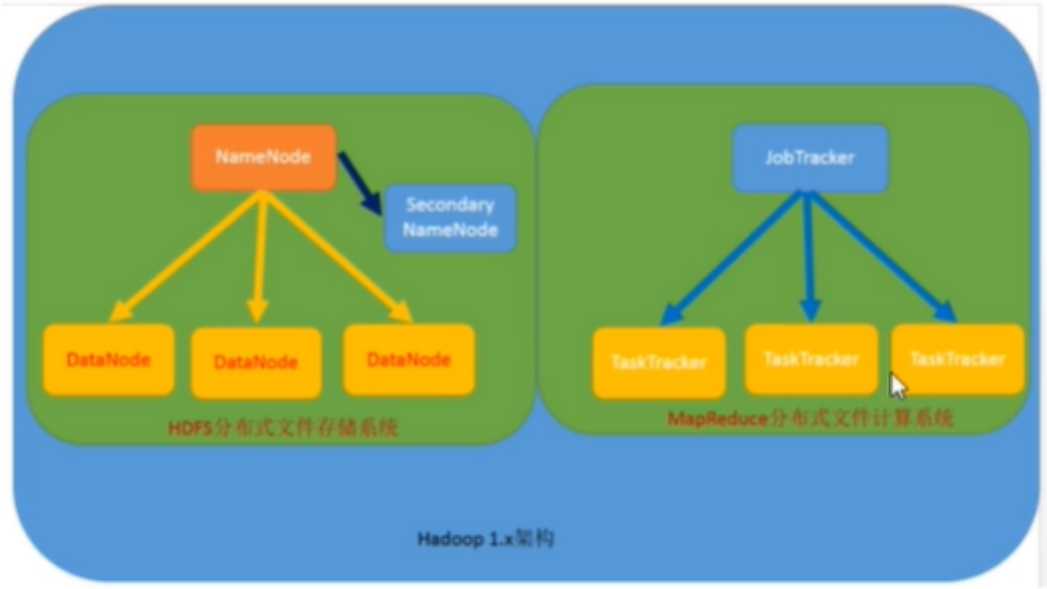
3台主机访问两个共享资源，且这两个共享资源必须要保持同步，使用zookeeper



In [ ]:
```
1
```

In [ ]:
```
1
```





文件系统核心模块：

NameNode：集群当中的主节点，管理元数据(文件的大小，文件的位置，文件的权限)，主要用于管理集群当中的各种数据

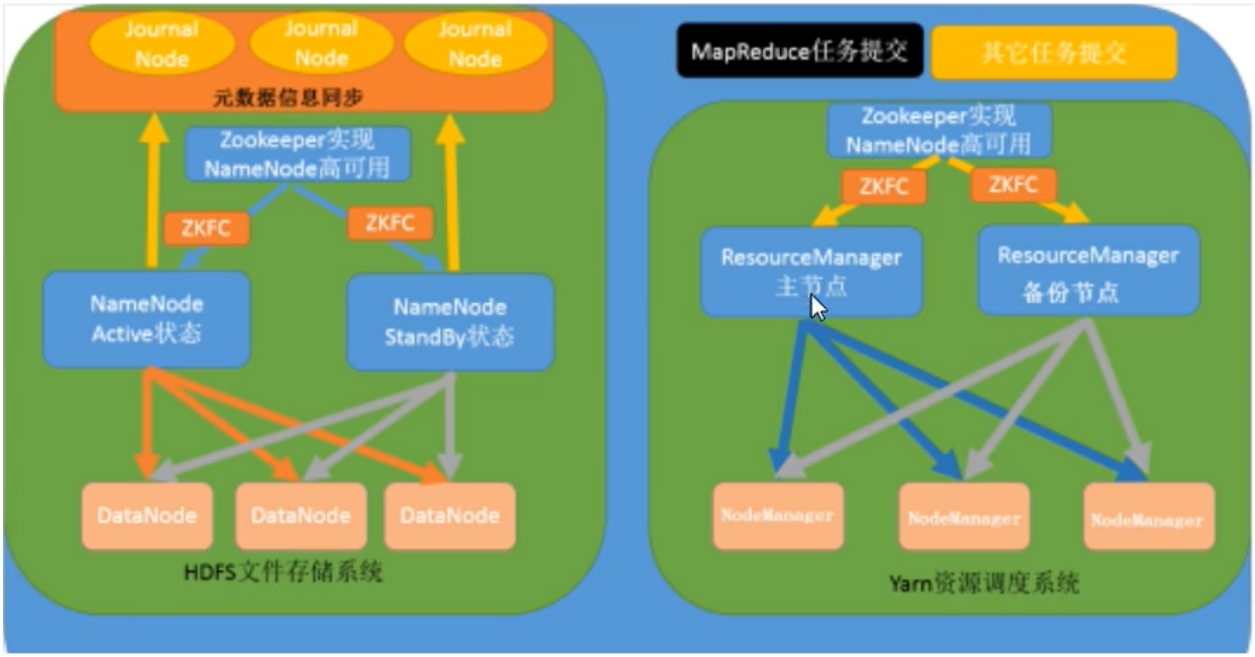
secondaryNameNode：主要能用于hadoop当中元数据信息的辅助管理

DataNode：集群当中的从节点，主要用于存储集群当中的各种数据

数据计算核心模块：

JobTracker：接收用户的计算请求任务，并分配任务给从节点

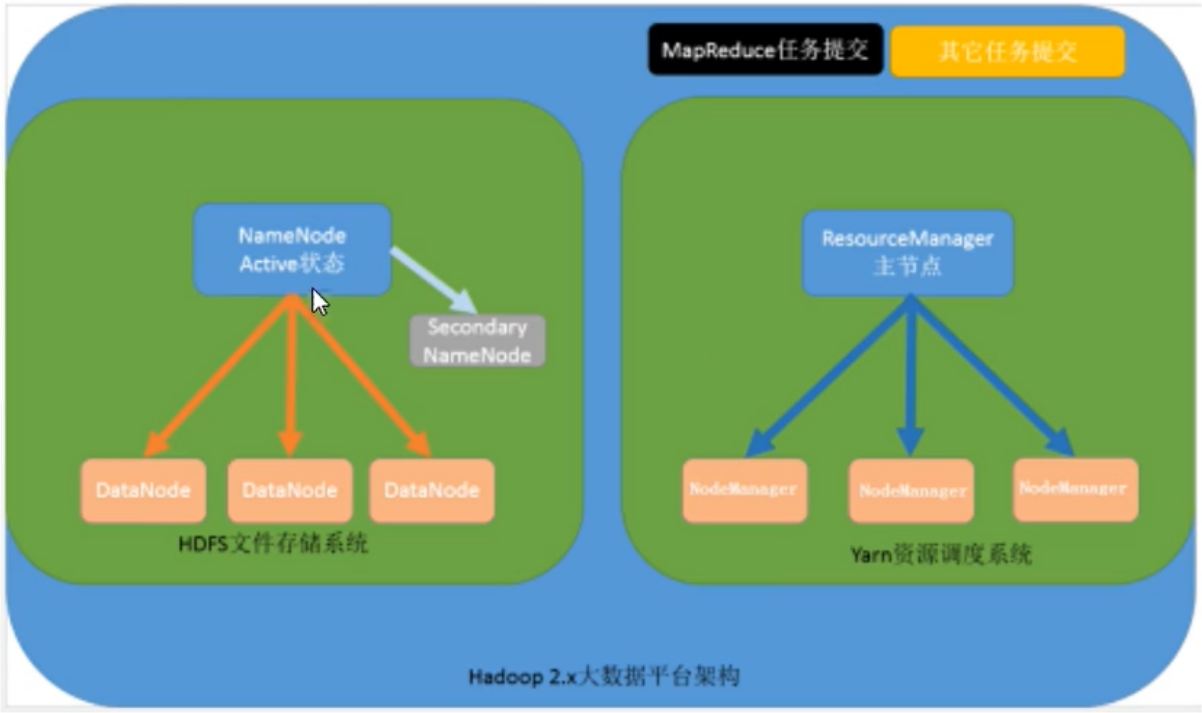
TaskTracker：负责执行主节点JobTracker分配的任务



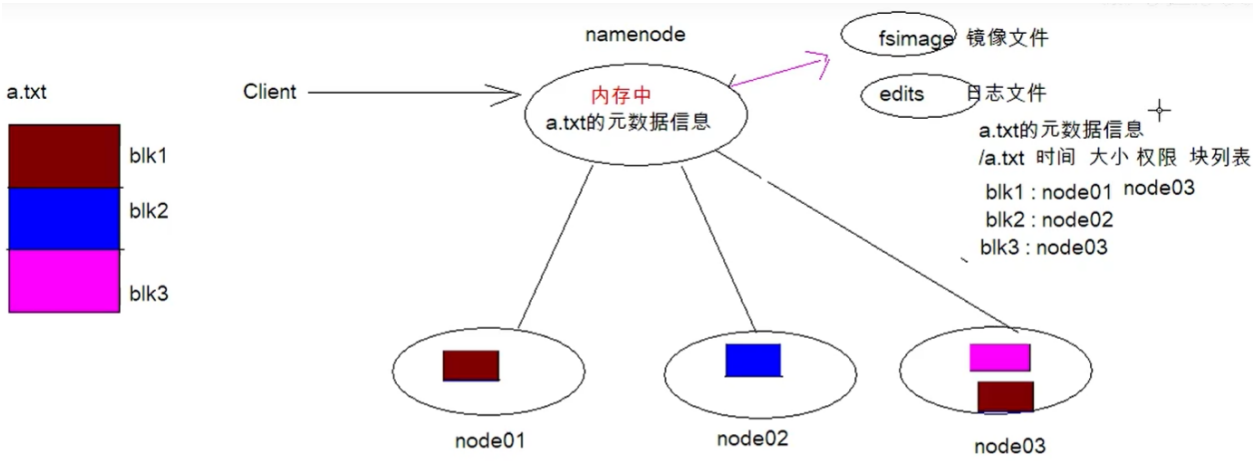
hadoop 2.x 生产中使用

学习使用下：

NameNode与ResourceManager单节点架构模型

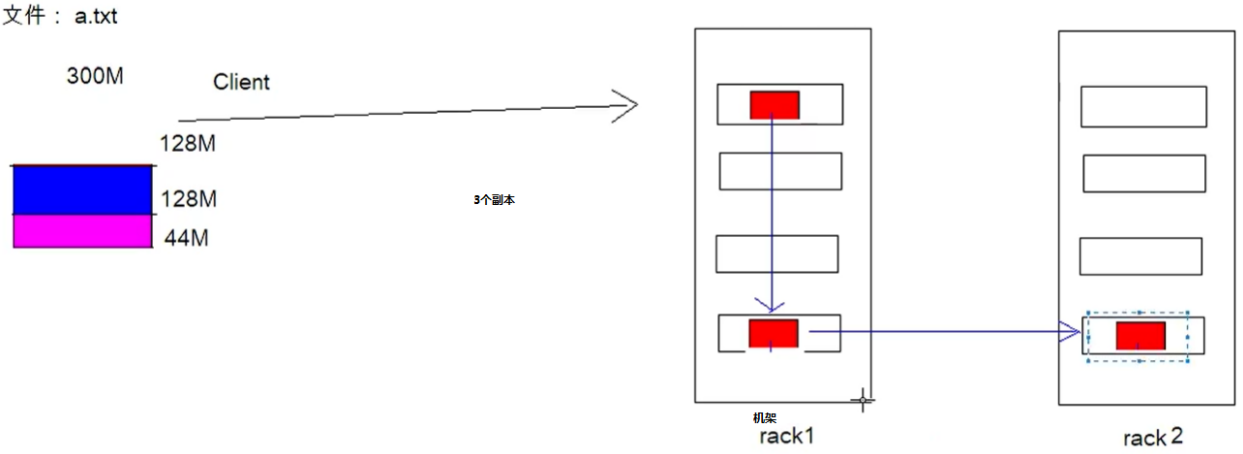


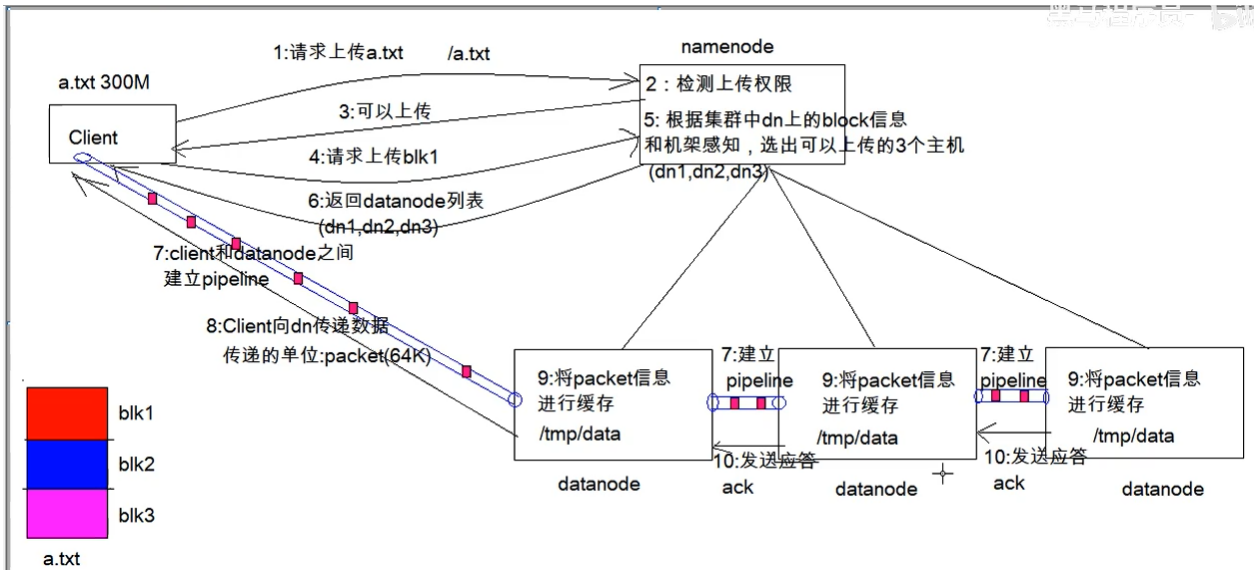
In [ ]:
```
1
```

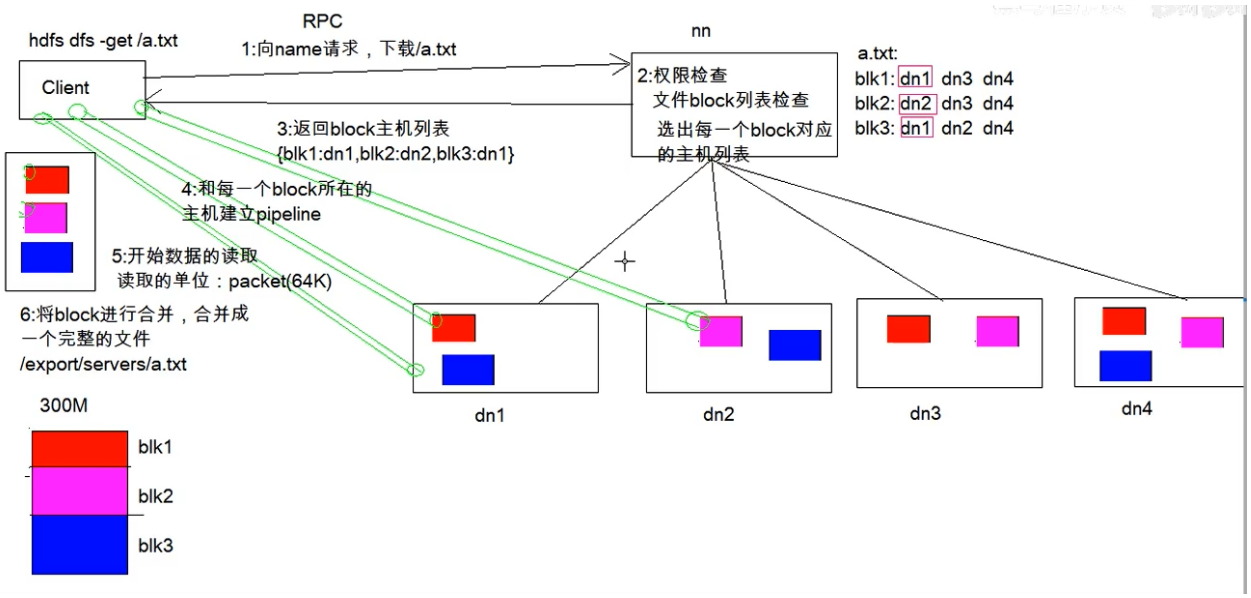

namenode
fsimage 镜像文件
edits 日志文件
a.txt
Client
内存中
a.txt的元数据信息
a.txt的元数据信息
/a.txt 时间 大小 权限 块列表
blk1 : node01 node03
blk2 : node02
blk3 : node03
blk1
blk2
blk3
node01
node02
node03

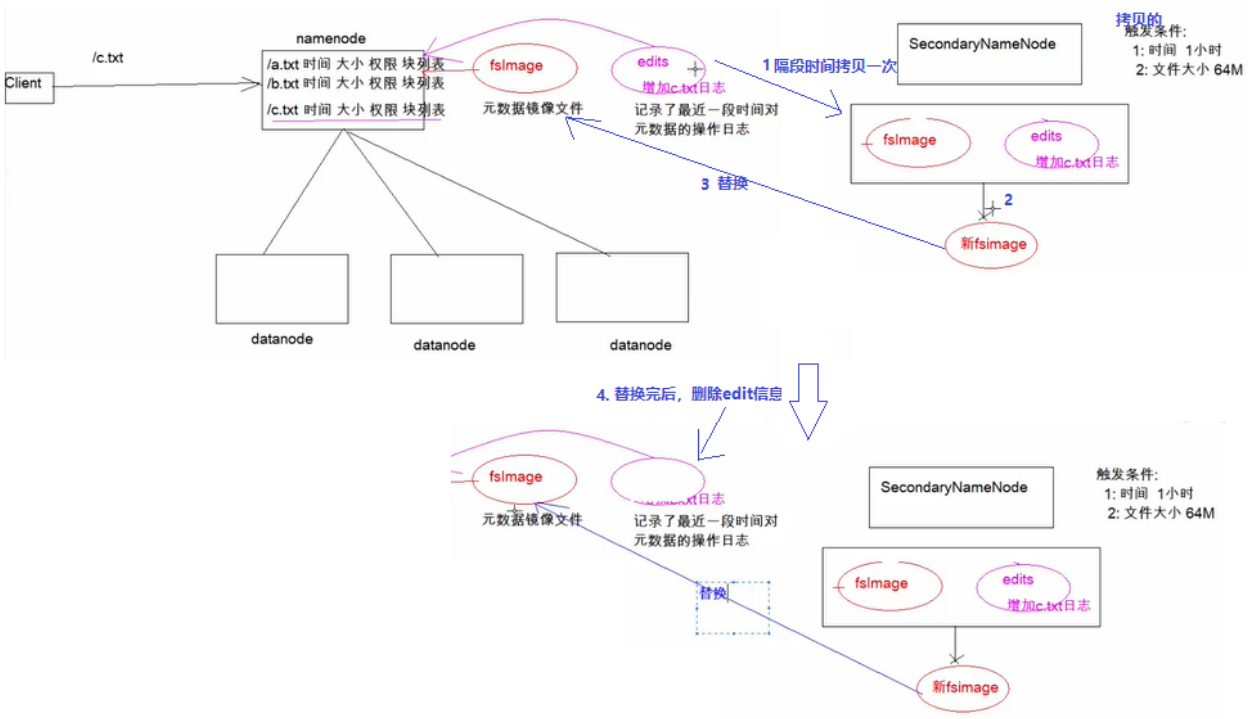
机架

文件： a.txt


300M
Client
128M
128M
44M
3个副本
机架
rack1
rack 2


1:请求上传a.txt /a.txt
namenode
a.txt 300M
2：检测上传权限
Client
3:可以上传
5: 根据集群中dn上的block信息
和机架感知，选出可以上传的3个主机
(dn1,dn2,dn3)
4:请求上传blk1
6:返回datanode列表
(dn1,dn2,dn3)
7:client和datanode之间
建立pipeline
8:Client向dn传递数据
传递的单位:packet(64K)
9:将packet信息
进行缓存
/tmp/data
7:建立
pipeline
9:将packet信息
进行缓存
/tmp/data
7:建立
pipeline
9:将packet信息
进行缓存
/tmp/data
10:发送应答
ack
10:发送应答
ack
datanode
datanode
datanode
blk1
blk2
blk3
a.txt


RPC
hdfs dfs -get /a.txt
1:向name请求，下载/a.txt
Client
nn
2:权限检查
文件block列表检查
选出每一个block对应
的主机列表
a.txt:
blk1: dn1 dn3 dn4
blk2: dn2 dn3 dn4
blk3: dn1 dn2 dn4
3:返回block主机列表
{blk1:dn1,blk2:dn2,blk3:dn1}
4:和每一个block所在的
主机建立pipeline
5:开始数据的读取
读取的单位：packet(64K)
6:将block进行合并，合并成
一个完整的文件
/export/servers/a.txt
300M
blk1
blk2
blk3
dn1
dn2
dn3
dn4


/c.txt
Client
namenode
/a.txt 时间 大小 权限 块列表
/b.txt 时间 大小 权限 块列表
/c.txt 时间 大小 权限 块列表
fsImage
元数据镜像文件
edits
增加c.txt日志
记录了最近一段时间对
元数据的操作日志
1 隔段时间拷贝一次
SecondaryNameNode
拷贝的
触发条件:
1: 时间 1小时
2: 文件大小 64M
fsImage
edits
增加c.txt日志
2
新fsimage
3 替换
datanode
datanode
datanode
4. 替换完后，删除edit信息
fsImage
元数据镜像文件
记录了最近一段时间对
元数据的操作日志
SecondaryNameNode
触发条件:
1: 时间 1小时
2: 文件大小 64M
fsImage
edits
增加c.txt日志
替换
新fsimage

In [ ]:
```
1
```

In [ ]:
```
1
```

yarn核心出发点是为了分离资源管理与作业监控，实现分离的做法是拥有一个全局的资源管理（ResourceManager，RM），以及每个应用程序对应一个的应用管理器（ApplicationMaster，AM）

总结一句话就是说：yarn主要就是为了调度资源，管理任务等

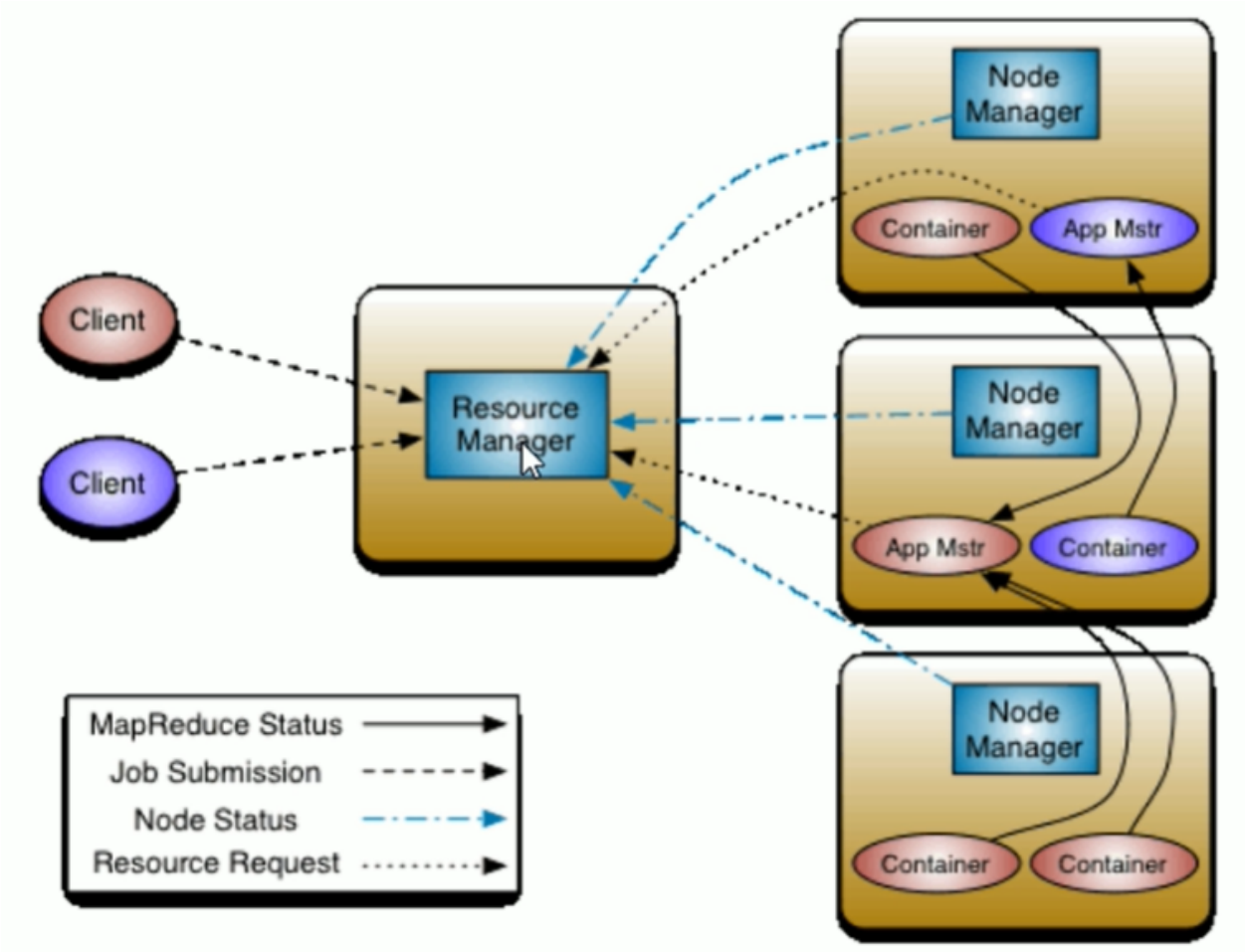
其调度分为两个层级来说：

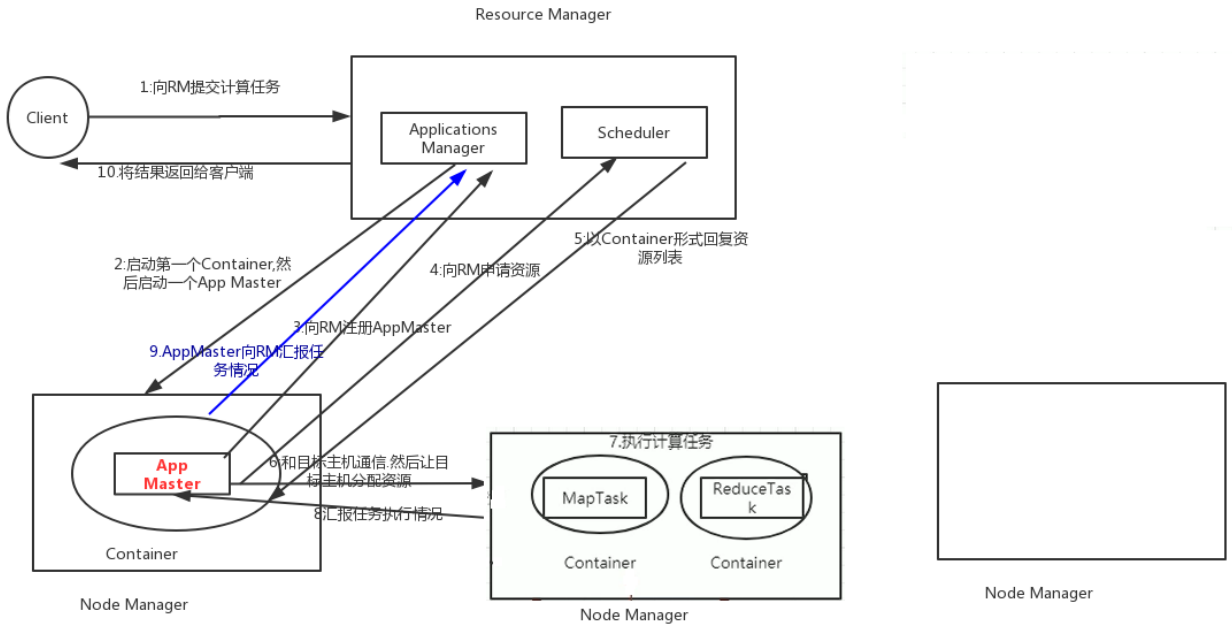
- 一级调度管理：

  计算资源管理(CPU,内存，网络IO，磁盘)

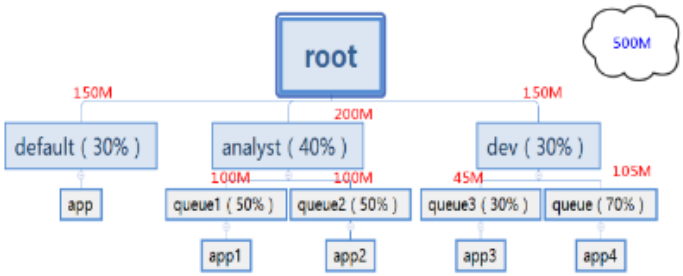
- 二级调度管理：

  任务内部的计算模型管理 (AppMaster的任务精细化管理)





队列调度器



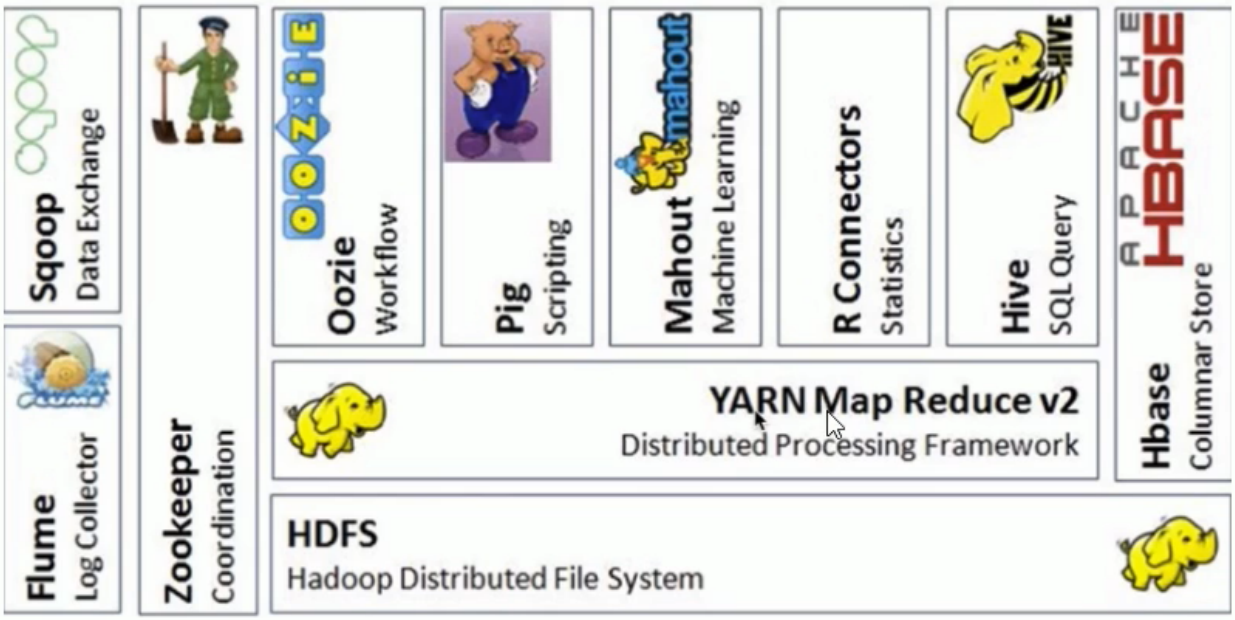
容量调度器

apache版本默认使用

In [ ]:
```
1
```

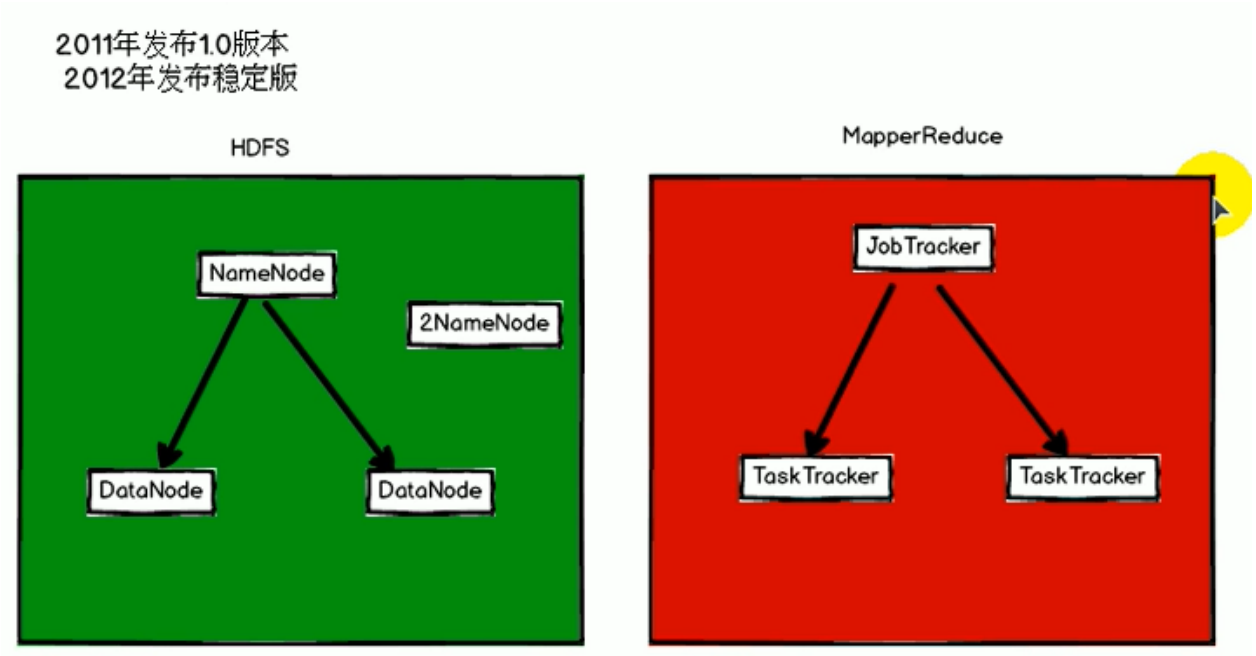
hadoop生态系统

- 广义的Hadoop：指的是Hadoop生态系统，Hadoop生态系统是一个很庞大的概念，hadoop是其中最重要最基础的一个部分，生态系统中每一子系统只解决某一个特定的问题域（甚至可能更窄），不搞统一型的全能系统，而是小而精的多个小系统；



In [ ]:
```
1
```

https://www.bilibili.com/video/BV174411X7Pk?from=search&seid=16140933637673040363 (https://www.bilibili.com/video/BV174411X7Pk?from=search&seid=16140933637673040363)



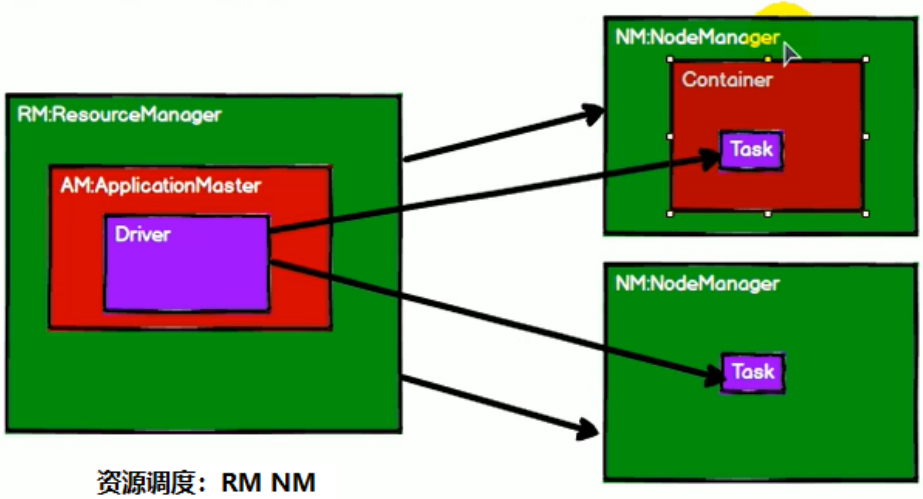
hadoop 1.0版本

MR的缺点：

mr基于数据集的计算，所以面向数据

1. 基本运算规则从存储介质中获取（采集）数据，然后进行计算，最后将结果存储到介质中，所以主要应用于一次性计算，不适合于数据挖掘和机器学习这样的迭代计算和图形挖掘计算。
2. MR基于文件存储介质的操作，所以性能非常的慢
3. MR和hadoop紧密耦合在一起，无法动态替换

2013年发布2.X版本 (Yarn)



container类似虚拟机，使得task和nm彼此独立，但是又有关系：内存分配

同上图

资源调度：RM NM

计算：Driver Task

AM Container起到 使计算和资源调度解耦合的作用

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```