# Hadoop

## 简答题

### （★）通常情况下集群模式的瓶颈在哪?

首先，瓶颈一般是指在整体中的关键限制因素，磁盘IO是指数据往磁盘读写，现在的科技速度最快的属[固态硬盘](https://www.baidu.com/s?wd=%E5%9B%BA%E6%80%81%E7%A1%AC%E7%9B%98&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)了，读的速度很大有1G/秒左右，但是写入速度最快几百兆/秒，集群中数据在cpu和内存之间速度快的可以忽略，处理速度也可以忽略，相对这些速度，磁盘读写就显得慢了，旁贷一下现在好一点的数据库oracle存储数据都是写日志先暂存然后等机器空闲再写入到磁盘，这些都是为了提高效率，不然执行一条操作等半天

### （★★）hadoop单机模式、伪分布模式、集群模式的区别是什么？

1、单机模式（独立模式）（Local或Standalone  Mode）

　　-默认情况下，Hadoop即处于该模式，用于开发和调式。

　　-不对配置文件进行修改。

　　-使用本地文件系统，而不是分布式文件系统。

　　-Hadoop不会启动NameNode、DataNode、JobTracker、TaskTracker等守护进程，Map()和Reduce()任务作为同一个进程的不同部分来执行的。

　　-用于对MapReduce程序的逻辑进行调试，确保程序的正确。

2.2、伪分布式模式（Pseudo-Distrubuted Mode）

　　-Hadoop的守护进程运行在本机机器，模拟一个小规模的集群

　　-在一台主机模拟多主机。

　　-Hadoop启动NameNode、DataNode、JobTracker、TaskTracker这些守护进程都在同一台机器上运行，是相互独立的Java进程。

　　-在这种模式下，Hadoop使用的是分布式文件系统，各个作业也是由JobTraker服务，来管理的独立进程。在单机模式之上增加了代码调试功能，允许检查内存使用情况，HDFS输入输出，

　　以及其他的守护进程交互。类似于完全分布式模式，因此，这种模式常用来开发测试Hadoop程序的执行是否正确。

　　-修改3个配置文件：core-site.xml（Hadoop集群的特性，作用于全部进程及客户端）、hdfs-site.xml（配置HDFS集群的工作属性）、mapred-site.xml（配置MapReduce集群的属性）

　　-格式化文件系统

3、全分布式集群模式（Full-Distributed Mode）

　　-Hadoop的守护进程运行在一个集群上

　　-Hadoop的守护进程运行在由多台主机搭建的集群上，是真正的生产环境。

　　-在所有的主机上安装JDK和Hadoop，组成相互连通的网络。

　　-在主机间设置SSH免密码登录，把各从节点生成的公钥添加到主节点的信任列表。

-修改3个配置文件：core-site.xml、hdfs-site.xml、mapred-site.xml，指定NameNode和JobTraker的位置和端口，设置文件的副本等参数

　　-格式化文件系统

### （★★）hadoop体系内常见的进程有哪些？

NameNode：管理集群，并记录DataNode文件信息。

Zkfc：ZKFC的HealthMonitor主要是监控NameNode是否正常，如果不正常进行备用机切换

SecondaryNameNode：主要做合并元数据。也可以做冷备，对一定范围内数据做快照性备份。

DataNode：存储数据。

ResourceManager：负责集群中所有资源的统一管理和分配，它接收来自各个节点（NodeManager）的资源汇报信息，并把这些信息按照一定的策略分配给各个应用程序（实际上是ApplicationManager）。

NodeManager：是YARN中每个节点上的代理，它管理Hadoop集群中单个计算节点，包括与ResourceManger保持通信，监督Container的生命周期管理，监控每个Container的资源使用（内存、CPU等）情况，追踪节点健康状况，管理日志和不同应用程序用到的附属服务。

### （★★）Hadoop的集群搭建步骤

a) 创建hadoop账号

b) 更改ip

c) 安装Java 更改/etc/profile 配置环境变量

d) 修改host文件域名

e) 安装ssh 配置无密码登录

f) 解压hadoop

g) 配置hadoop  conf下面的配置文件

h) Hadoop namenode -format  格式化

i) Start 启动

### （★★★）hadoop生态体系的构成以及每个组件的作用

重点组件：

HDFS：分布式文件系统

MAPREDUCE：分布式运算程序开发框架

HIVE：基于大数据技术（文件系统+运算框架）的SQL数据仓库工具

HBASE：基于HADOOP的分布式海量数据库

ZOOKEEPER：分布式协调服务基础组件

Mahout：基于mapreduce/spark/flink等分布式运算框架的机器学习算法库

Oozie：工作流调度框架azkaban

Sqoop：数据导入导出工具

Flume：日志数据采集框架

### （未做）CDH版本和对应Hadoop版本号

<https://www.cloudera.com/documentation/enterprise/release-notes/topics/cdh_vd_cdh_package_tarball_516.html?tdsourcetag=s_pcqq_aiomsg>

### （★★）谈谈 hadoop1 和 hadoop2 的区别

Hadoop 2.x架构有一个额外的新组件：YARN（Yet Another Resource Negotiator）

### （★★）文件大小默认为 64M，改为 128M 有啥影响？。

文件block块需要根据我们的实际生产中的需求来更改。如果block定义的太小，大的文件都会被切分成太多的小文件，减慢用户上传效率；如果block定义的太大，那么大量的小文件可能都会存到一个block块中，增加namenode的管理内存压力。

### （★★）NameNode 与 SecondaryNameNode 的区别与联系？

namenode主要负责元数据的管理和与client的通讯。secondarynamenode主要负责对namenode的edits日志文件和fsimage文件进行合并。

### （★★★）hadoop运行原理

包括HDFS和Mapreduce两部分。

1）HDFS自动保存多个副本，移动计算。缺点是小文件存取占用namenode内存，写入只支持追加，不能随机修改。

它存储的逻辑空间称为block，文件的权限类似linux。整体架构分三种节点，NN,SNN,DN

NN 负责读写操作保存metadata(Ownership Permission blockinfo)

SNN 负责辅助NN合并fsimage和edits，减少nn启动时间

DN 负责存数据，每个数据（文件）分割成若干block，每个block默认3个副本。启动后像NN发送心跳保持联系

NN保存的metadata在hdfs启动后加载到计算机内存，除block位置信息的metadata保存在OS文件系统中的fsimage文件中，对metadata的操作日志保存在OS文件系统中的edits文件中。block位置信息是hdfs启动后由DN上报NN再加载到内存的。

HDFS的安全模式：直到NN完全加载完metadata之前的这段时间。期间不能写入文件，DN检查各个block完整性，并修复。

2）MapReduce

离线计算框架，过程分为split map shuffle reduce四个过程

架构节点有：Jobtracker TaskTracker

Split将文件分割，传输到mapper，mapper接收KV形式的数据，经过处理，再传到shuffle过程。

Shuffle先进行HashPartition或者自定义的partition，会有数据倾斜和reduce的负载均衡问题；再进行排序，默认按字典排序；为减少mapper输出数据，再根据key进行合并，相同key的数据value会被合并；最后分组形成（key,value{}）形式的数据，输出到下一阶段

Reduce输入的数据就变成了，key+迭代器形式的数据，再进行处理

### （★★★）Hadoop的核心配置文件

core-default.xml

hdfs-default.xml

mapred-default.xml

### （★）首先集群的目的是为了节省成本，用廉价的 pc 机，取代小型机及大型机。小型机和大型机有什么特点？

小型机与PC服务器的区别：在英文里这两位都叫server（服务器），小型机是国内的习惯称呼。pc服务器则主要指基于intel处理器的x86架构，是一个通用开放的系统。

而不同品牌的小型机架构大不相同，使用risc、mips处理器，像美国sun、日本fujitsu等公司的小型机是基于sparc处理器架构，而美国hp公司的则是基于pa－risc架构，compaq公司是alpha架构，ibm和sgi等的也都各不相同；i／o总线也不相同，fujitsu是pci，sun是sbus，等等，这就意味着各公司小型机机器上的插卡，如网卡、显示卡、scsi卡等可能也是专用的；

操作系统一般是基于unix的，像sun、fujitsu是用sun solaris，hp是用hp－unix，ibm是aix，等等，所以小型机是封闭专用的计算机系统。使用小型机的用户一般是看中unix操作系统的安全性、可靠性和专用服务器的高速运算能力，虽然小型机的价格是pc服务器的好几倍。

2、相对而言，pc服务器具有较高的性能价格比，通用性是其最大的一个优点，会使用pc机就可以很容易地使用pc服务器，它的硬件结构与pc机差不多，用户心里有底；它的操作系统一般是windows nt/2000，又对用惯windows 95／98的用户来说容易掌握，而且应用软件也更丰富。另外，价格便宜也是pc服务器的优势，价位处于pc机与昂贵的小型机之间，具有最佳的性能价格比。

大型机的特色：

大型机（Mainframe）的功能、价格以及性能上都在小型机和微型计算机之上，是一种用于大规模计算的计算机系统，由于早期产品体积庞大因而被称为大型机。

1964年IBM公司花费50亿美金开发出了IBM SYSTEM/360大型机，开始了大型机的历史。大型机通常采用集中式体系架构，这种架构的优势之一是其出色的I/O处理能力，因而最适合处理大规模事务数据。大型机不单纯用来进行传统的海量数据处理和财务事务处理。在一些场合，它可作为企业的中心架构，用来提高安全性，可用性和可管理性。大型机也可用来安装多个操作系统，可以运行所有的主流的软件包。

大型机通常用于政府、银行、交通、保险公司和大型制造企业。特点是处理数据能力强大、稳定性和安全性又非常高。system/360开始的一系列的IBM计算机是大型机的典型代表，而IBM也是全世界最大的大型机制造商。目前生产大型机的企业主要有两家：IBM和UNISYS。IBM大型机是其z系列服务器。顺便说下，Sun和HP不生产大型机，但它们生产性能和用途类似于IBM大型机的高端Unix服务器。

### （★）写出下面的命令

杀死一个job

Hadoop job –list 拿到jobid，Hadoop job –kill jobid

删除HDFS上的/tmp/aaa目录

Hadoop fs –rm –r /tmp/aaa

加入一个新的存储节点、删除一个计算节点（需要刷新集群状态命令）

加新节点：Hadoop-daemon.sh start datanode Hadoop-daemon.sh start nodemanager

均衡数据节点上的数据：./bin/start-balancer.sh

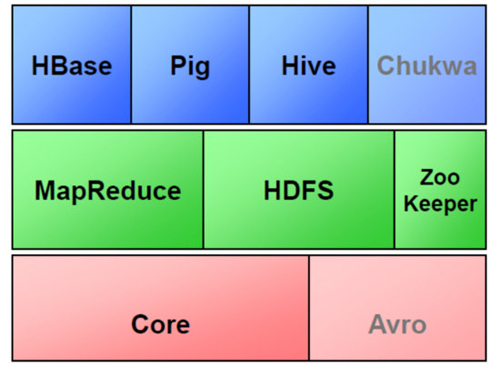
删除节点：Hadoop mradmin –refreshnodes Hadoop dfsadmin -refreshnodes

### （★）如何确定Hadoop集群的健康状态

通过页面监控,脚本监控。

### （★★）Hadoop 生态圈中各种框架的运用场景？

Hadoop生态圈



①HBase

Google Bigtable的开源实现

列式数据库

可集群化

可以使用shell、web、api等多种方式访问

适合高读写(insert)的场景

HQL查询语言

NoSQL的典型代表产品

②Hive

数据仓库工具。可以把Hadoop下的原始结构化数据变成Hive中的表

支持一种与SQL几乎完全相同的语言HiveQL。除了不支持更新、索引和事务，几乎SQL的其它特征都能支持

可以看成是从SQL到Map-Reduce的映射器

提供shell、JDBC/ODBC、Thrift、Web等接口

**③Zookeeper**

Google Chubby的开源实现

用于协调分布式系统上的各种服务。例如确认消息是否准确到达，防止单点失效，处理负载均衡等

应用场景：Hbase，实现Namenode自动切换

工作原理：领导者，跟随者以及选举过程

**④Sqoop**

用于在Hadoop和关系型数据库之间交换数据

通过JDBC接口连入关系型数据库

**⑤Chukwa**

架构在Hadoop之上的数据采集与分析框架

主要进行日志采集和分析

通过安装在收集节点的“代理”采集最原始的日志数据

代理将数据发给收集器

收集器定时将数据写入Hadoop集群

指定定时启动的Map-Reduce作业队数据进行加工处理和分析

**⑥Pig**

Hadoop客户端

使用类似于SQL的面向数据流的语言Pig Latin

Pig Latin可以完成排序，过滤，求和，聚组，关联等操作，可以支持自定义函数

Pig自动把Pig Latin映射为Map-Reduce作业上传到集群运行，减少用户编写Java程序的苦恼

**⑦Avro**

数据序列化工具，由Hadoop的创始人Doug Cutting主持开发

用于支持大批量数据交换的应用。支持二进制序列化方式，可以便捷，快速地处理大量数据

动态语言友好，Avro提供的机制使动态语言可以方便地处理 Avro数据。

**⑧Cassandra**

NoSQL，分布式的Key-Value型数据库，由Facebook贡献

与Hbase类似，也是借鉴Google Bigtable的思想体系

只有顺序写，没有随机写的设计，满足高负荷情形的性能需求

### （★★）解释“hadoop”和“hadoop 生态系统”两个概念。

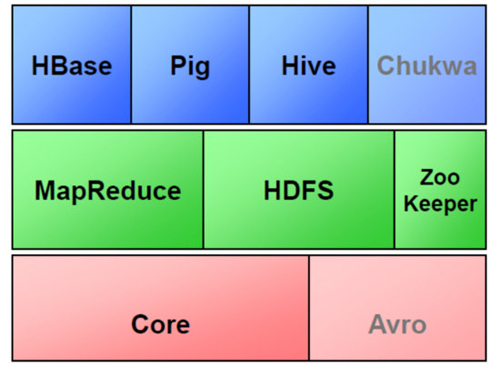
Hadoop 是什么

(1)Hadoop是一个开源的框架，可编写和运行分布式应用处理大规模数据，是专为离线和大规模数据分析而设计的，并不适合那种对几个记录随机读写的在线事务处理模式。Hadoop=HDFS（文件系统，数据存储技术相关）+ Mapreduce（数据处理），Hadoop的数据来源可以是任何形式，在处理半结构化和非结构化数据上与关系型数据库相比有更好的性能，具有更灵活的处理能力，不管任何数据形式最终会转化为key/value，key/value是基本数据单元。用函数式变成Mapreduce代替SQL，SQL是查询语句，而Mapreduce则是使用脚本和代码，而对于适用于关系型数据库，习惯SQL的Hadoop有开源工具hive代替。

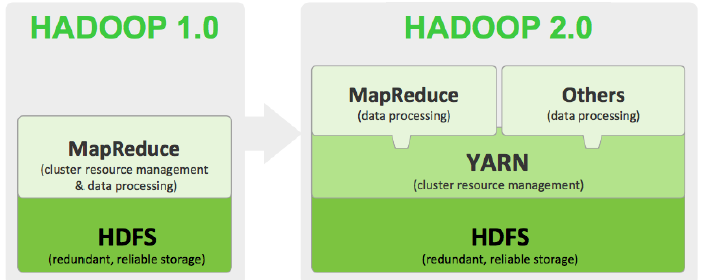
(2)Hadoop就是一个分布式计算的解决方案.

Hadoop生态圈就是围绕hadoop框架的一系列工具集成

如下图



### （★★★）说明 Hadoop 2.0 的基本构成。



Hadoop1.x构成： HDFS、MapReduce(资源管理和任务调度)；运行时环境为JobTracker和TaskTracker；

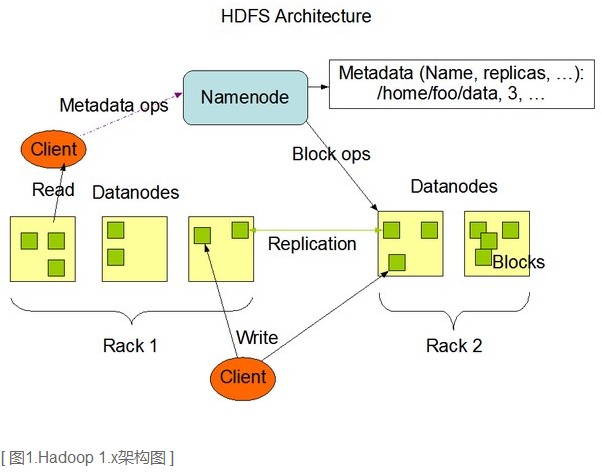
Hadoop2.0构成：HDFS、MapReduce/其他计算框架、YARN； 运行时环境为YARN

　　1、HDFS：HA、NameNode Federation

　　2、MapReduce/其他计算框架：运行在YARN之上的MapReduce通常称之为MapReduce2.0（MRv2）

　　3、YARN：资源管理系统（Yet Another Resource Negotiator），在其之上可以运行各种计算框架，如：MapReduce、Storm、Spark等；

### （★）请说说 hadoop1 的 HA 如何实现？



为保障集群的HA，设计的是一种基于NAS的共享存储机制，即主备NameNode间通过NAS进行元数据的同步。该方案有什么缺点呢，主要有以下几点：

定制化硬件设备：必须是支持NAS的设备才能满足需求

复杂化部署过程：在部署好NameNode后，还必须额外配置NFS挂载、定制隔离脚本，部署易出错

简陋化NFS客户端：Bug多，部署配置易出错，导致HA不可用

Ps：具体和hadoop2的对比见博客：https://blog.csdn.net/u010738184/article/details/78270416

### （★★）Hadoop 体系结构（HDFS 与 MapReduce 的体系结构）、 Hadoop 相比传统数据存储方式（比如 mysql）的优势？

Hadoop是什么 开源的 分布式存储 + 分布式计算平台

搭建大型数据仓库，PB级数据的存储、处理、分析、统计等业务

优势

1、高扩展

2、低成本

3、成熟的生态圈

见：https://blog.csdn.net/chengqiuming/article/details/78602185

### （★★★）说说你们做的 hadoop 项目流程？

1.使用flume把日志数据导入到hdfs中

2.对数据进行清洗、清洗后的数据易于我们使用

3.明细日志使用hbase存储,能够利用ip、时间查询

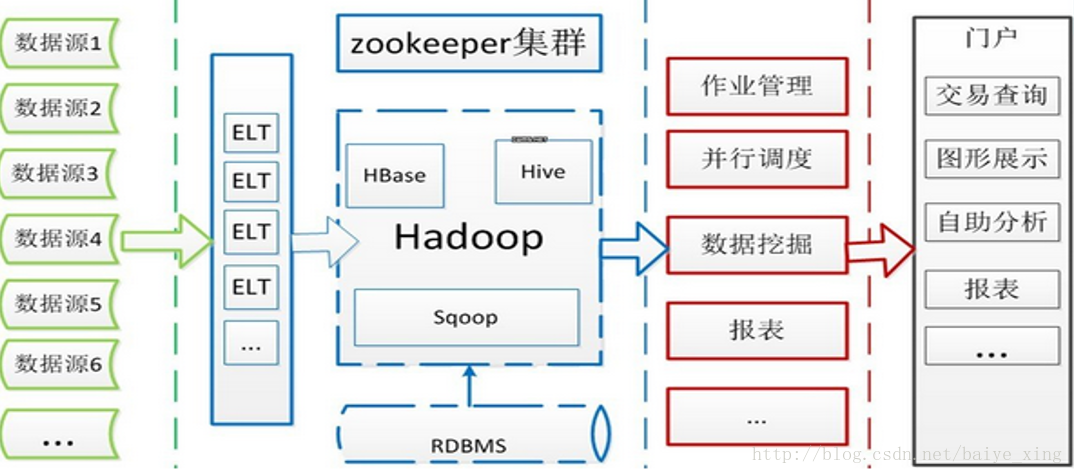
4.使用hive进行数据的多维分析

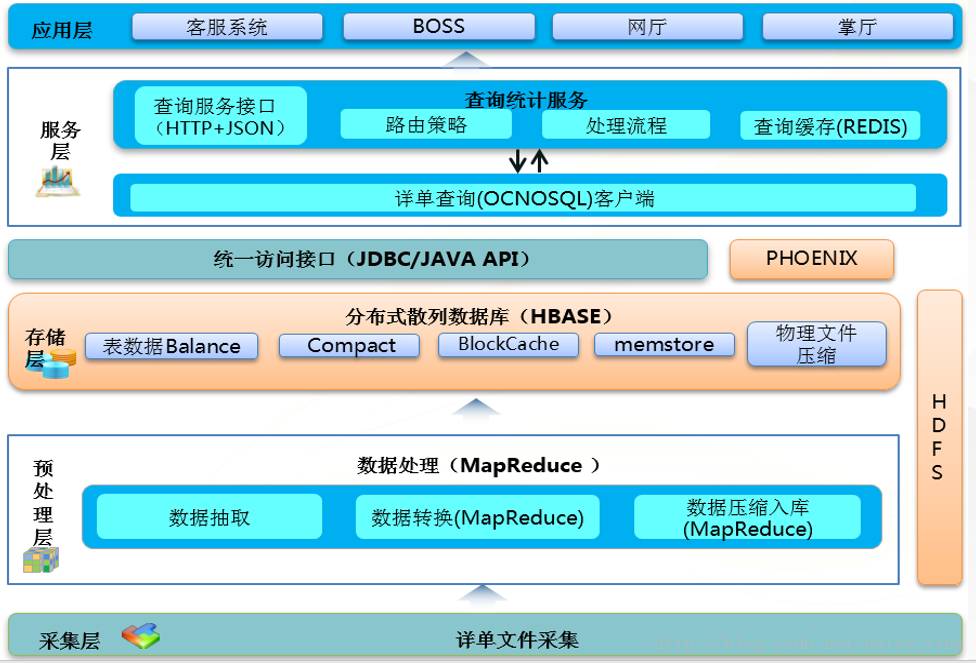
5.把hive分析结果使用sqoop导出到mysql中

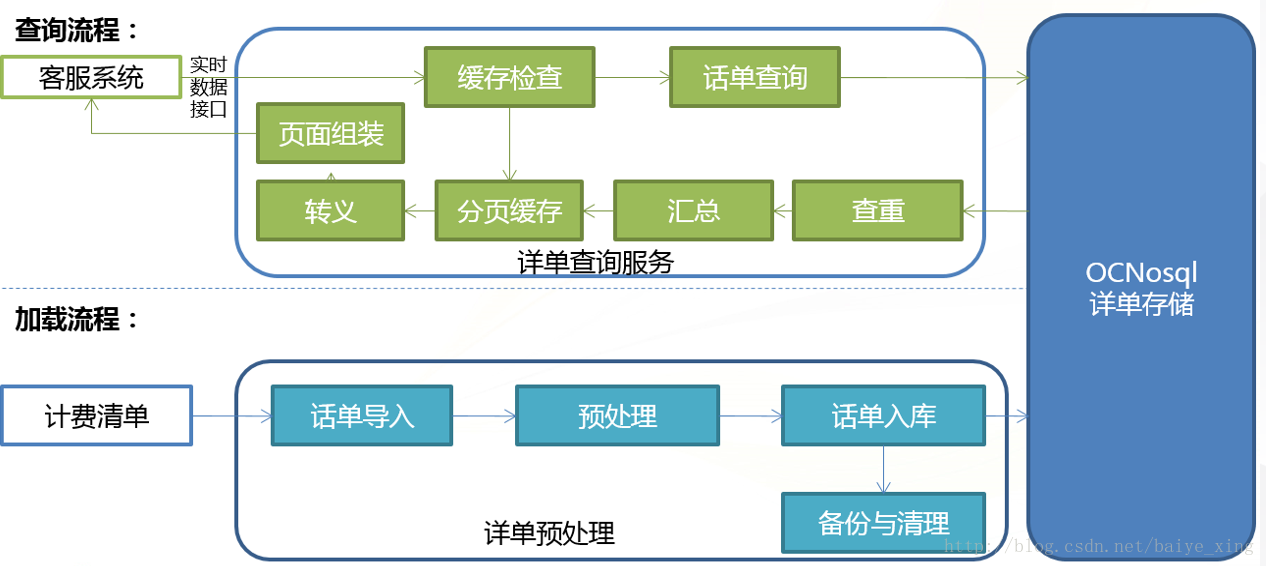
6.提供视图工具供用户使用

### （★★★）你们公司的服务器架构是怎么样的（分别说下 web 跟 hadoop）？

Hadoop Ha 高可用







详见：https://blog.csdn.net/baiye\_xing/article/details/76228522

### （★）在 hadoop 组中你主要负责那部分？

搭建集群

### （★★★）什么是 jobtracker？jobtracker 有哪些特别的函数

提示：jobtracker 负责接收用户提交的作业，负责启动、跟踪任务执行。是一个 rpc

服务端jobtracker 有哪些特别的函数：JobSubmitProtocol

### （★★）什么是 tasktracker？

提示：mr 的客户端，接收 jobtracker 的发出的指令，用来执行任务的

### （★）hadoop 推测执行时如何实现的？

研究和分析Hadoop推测执行算法在异构环境下性能较差的问题,在深入研究源码的基础上提出改进算法。该算法根据系统负载情况自动调节后备任务的执行,实现系统负载均衡。采用Zaharia提出的历史平均剩余完成时间来估计剩余时间,并使用剩余时间值大于20%的方法来判断掉队者,进而得到更精确的掉队者队列。该算法在一定程度上提高了异构环境中推测执行的性能。

### （★★）Linux 中使用命令行，如和查看 hadoop 集群中所有运行的任务？或是 kill 掉任务？

提示：hadoop job -list

### （★）什么 hadoop streaming？

Hadoop Streaming是Hadoop提供的一个编程工具，它允许用户使用任何可执行文件或者脚本文件作为Mapper和Reducer，例如：

采用shell脚本语言中的一些命令作为mapper和reducer（cat作为mapper，wc作为reducer）

$HADOOP\_HOME/bin/hadoop jar $HADOOP\_HOME/contrib/streaming/hadoop-\*-streaming.jar \

-input myInputDirs \

-output myOutputDir \

-mapper cat \

-reducer wc

详见：http://dongxicheng.org/mapreduce/hadoop-streaming-programming/

### （★★）有关项目细节，产生的日志量多大，集群规模，处理数据要用多久

http://www.cnblogs.com/ggjucheng/archive/2013/01/03/2843243.html

### （★★）大数据建模和清洗

### （★★）HadoopHA 集群中，各个服务的启动和关闭的顺序

启动流程：先启动zookeeper，再启动hdfs，然后启动yarn最后单独启动resourcemanager

关闭流程：先关闭hdfs，再关闭yarn，然后单独关闭resourcemanager，最后关闭zookeeper

### （★★）请写出以下执行命令

1）杀死一个job

2）删除hdfs 上的/tmp/aaa 目录、

3）加入一个新的存储节点和删除有一个计算节点需要刷新集群状态命令

答案：

1）hadoop job list 查看job-id， 然后 hadoop job -kill job-id

2）Hadoop fs -rmr /tmp/aaa

3) 添加节点

方式1：静态添加datanode，停止namenode方式

1.停止namenode

2.修改slaves文件，并更新到各个节点

3.启动namenode

4.执行hadoop balance命令。

ps:（此项为balance集群使用，如果只是添加节点，则此步骤不需要）

方式2：动态添加datanode，不停namenode方式

1.修改slaves文件，添加需要增加的节点host或者ip，并将其更新到各个节点

2.在datanode中启动执行启动datanode命令。命令：sh hadoop-daemon.sh start datanode

3.可以通过web界面查看节点添加情况。或使用命令：sh hadoop dfsadmin -report

4.执行hadoop balance命令。（此项为balance集群使用，如果只是添加节点，则此步骤不需要）

针对第4点，start-balancer.sh可以执行-threshold参数。

-threshold参数是指定平衡的阈值。

-threshold的默认是10，即每个datanode节点的实际hdfs存储使用量/集群hdfs存储量

举例：

datanode hdfs使用量1.2G；

集群总hdfs存储量10T即10G；

则t值为1.2/10 = 0.12；

当执行balance的-t参数小于0.12时，集群进行balance；

命令为：start-balancer.sh -threshold 0.1

注：

1. balance命令可以在namenode或者datanode上启动；

可以随时停止balance命令。

balance的默认带宽是1M/s。

2. slave文件是用于重启时使用。集群的start和stop需要读取slave文件。

启用datanode时只要在hdfs-site中配置了namenode位置，就可以将信息push给namenode。

查看namenode的http管理界面，可查看节点添加情况

### （★★）描述一下 hadoop 中，有哪些地方使用了缓存机制，作用分别是什么

环形缓冲区 ：

存储OutputCollector类收集Mapper类的输出结果

当环形缓冲区的数据存储量（100M）达到缓冲区大小的80%或遇到EOF标识符时，HashPartitioner开始对缓冲区数据进行分区。分区结束后，根据分区进行数据排序(key.compartTo函数)。

当分区排序操作结束时，将数据溢写到磁盘中(分区且有序)。

Hadoop的分布式缓存：

在执行MapReduce时，可能Mapper之间需要共享一些信息，如果信息量不大，可以将其从HDFS加载到内存中，这就是Hadoop分布式缓存机制。

使用方法详见：https://www.cnblogs.com/itboys/p/6129986.html

### （★★★）搭建 hadoop 集群，master/slave 都运行那些服务

Master运行：NameNode - HDFS的守护程序

记录文件是如何分割成数据块及这些数据块被存储到哪些节点；

对内存和I/O进行集中管理；

是个单点，发生故障将使集群崩溃；

协调客户端对文件的访问；

管理文件系统的命名空间，记录命名空间内的改动或空间本身属性的改动，记录每个文件数据块在各个Datanode上的位置和副本信息；

Namenode 使用事务日志记录HDFS元数据的变化。使用映像文件存储文件系统的命名空间，包括文件映射，文件属性等。

Secondary Namenode

监控HDFS状态的辅助后台程序；

每个集群一般都有一个；

与Namenode通信，定期保存HDFS元数据快照，快照可以降低Namenode宕机的影响；

协助Namenode合并事务日志文件；

当Namenode故障时可作为备用Namenode使用

Zkfc

在正常的情况下，ZKFC的HealthMonitor主要是监控NameNode主机上的磁盘还是否可用（空间），NameNode负责维护集群上的元数据信息，当磁盘不可用的时候，NN就该进行切换了

Slave运行：DataNode

每台slave服务器都运行一个；

负责把HDFS数据块读写到本地文件系统；

负责所在物理节点 的存储管理；

一次写入，多次读取（不修改）

文件由数据块组成，典型的块大小是64MB，可配置；

数据块尽量散布在各节点

NodeManager

NM是ResourceManager在slave机器上的代理，负责容器管理，并监控它们的资源使用情况，以及向ResourceManager/Scheduler提供资源使用报告

### （★★）配置机架感知的下面哪项正确

如果一个机架出问题，不会影响数据读写

写入数据的时候会写到不同机架的 DataNode 中

MapReduce 会根据机架获取离自己比较近的网络数据

## 选择题

### （★）下列哪项可以作为集群的管理？

A、Puppet B、Pdsh C、Cloudera Manager D、Zookeeper

答案 ABD

### （★★★）下列哪个是 Hadoop 运行的模式

A、单机版 B、伪分布式 C、分布式

答案 ABC 单机版,伪分布式只是学习用的。

### （★★★）Cloudera 提供哪几种安装 CDH 的方法

a)Cloudera manager

b)Tar ball

c)Yum

d)Rpm

## 判断题

### （★）Ganglia 不仅可以进行监控，也可以进行告警。（y ）

### （★）Nagios 不可以监控 Hadoop 集群，因为它不提供 Hadoop 支持。（ ）

### （★）Hadoop 是 Java 开发的，所以 MapReduce 只支持 Java 语言编写。（x ）

### （未做）Hadoop 支持数据的随机写。（ ）

### （★）Slave 节点要存储数据，所以它的磁盘越大越好。（ x）

### （★★）Hadoop 默认调度器策略为 FIFO，并支持多个 Pool 提交 Job。（ x）

### （★）集群内每个节点都应该配 RAID，这样避免单磁盘损坏，影响整个节点运行。（ x）

### （★★）Hadoop 环境变量中的 HADOOP\_HEAPSIZE 用于设置所有 Hadoop 守护线程的内存。它默认是 200MB。（ x）

### （★★）Hadoop1.0 和 2.0 都具备完善的 HDFS HA 策略。（x）

### （★）PIG 是脚本语言，它与 mapreduce 无关。（x）

### （未做）hadoop 框架中文件拆分时如何被触发的？（block 是怎么触发）

### （★）Cloudera CDH 是需要付费使用的。（y ）

### （★）Hadoop 自身具有严格的权限管理和安全措施保障集群正常运行。（x ）