

1. 下面哪个程序负责 HDFS 数据存储。

a)NameNode b)Jobtracker c)Datanode d)secondaryNameNode e)tasktracker

答案 C datanode

2. HDFS 中的 block 默认保存几份？

a)3 份 b)2 份 c)1 份 d)不确定

答案 A 默认 3 份

3. 下列哪个程序通常与 NameNode 在一个节点启动？

a)SecondaryNameNode b)DataNode c)TaskTracker d)Jobtracker

答案 D

分析：

hadoop 的集群是基于 master/slave 模式，namenode 和 jobtracker 属于 master，datanode 和 tasktracker 属于 slave，master 只有一个，而 slave 有多个

SecondaryNameNode 内存需求和 NameNode 在一个数量级上，所以通常 secondary NameNode（运行在单独的物理机器上）和 NameNode 运行在不同的机器上。

JobTracker 和 TaskTracker

JobTracker 对应于 NameNode

TaskTracker 对应于 DataNode

DataNode 和 NameNode 是针对数据存放来而言的

JobTracker 和 TaskTracker 是对于 MapReduce 执行而言的

mapreduce 中几个主要概念，mapreduce 整体上可以分为这么几条执行线索：

jobclient，JobTracker 与 TaskTracker。

1、JobClient 会在用户端通过 JobClient 类将应用已经配置参数打包成 jar 文件存储到 hdfs，并把路径提交到 Jobtracker,然后由 JobTracker 创建每一个 Task（即 MapTask 和 ReduceTask）

并将它们分发到各个 TaskTracker 服务中去执行

2、JobTracker 是一个 master 服务，软件启动之后 JobTracker 接收 Job，负责调度 Job 的每一个子任务 task 运行于 TaskTracker 上，并监控它们，如果发现有失败的 task 就重新运行它。一般情况应该把 JobTracker 部署在单独的机器上。

3、TaskTracker 是运行在多个节点上的 slaver 服务。TaskTracker 主动与 JobTracker 通信，接收作业，并负责直接执行每一个任务。

TaskTracker 都需要运行在 HDFS 的 DataNode 上

4. Hadoop 作者

a)Martin Fowler b)Kent Beck c)Doug cutting

答案 C Doug cutting

5. HDFS 默认 Block Size

a)32MB b)64MB c)128MB

答案: B

6. 下列哪项通常是集群的最主要瓶颈

a)CPU b)网络 c)磁盘 IO d)内存

答案: C 磁盘

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？

1.cpu 处理能力强

2.内存够大

所以集群的瓶颈不可能是 a 和 d

3.如果是互联网有瓶颈，可以让集群搭建内网。每次写入数据都要通过网络（集群是内网），然后还要写入 3 份数据，所以 IO 就会打折扣。

同样可以参考下面帖子的实际案例

[集群瓶颈为什么磁盘 io](#)

(点此链接 360 可能会报警，可以完全信任)

7. 关于 SecondaryNameNode 哪项是正确的？

a)它是 NameNode 的热备 b)它对内存没有要求

c)它的目的是帮助 NameNode 合并编辑日志，减少 NameNode 启动时间

d)SecondaryNameNode 应与 NameNode 部署到一个节点

答案 C。

D 答案可以参考第三题

多选题：

8. 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

具体可查看

[什么是 Zookeeper, Zookeeper 的作用是什么，在 Hadoop 及 hbase 中具体作用是什么](#)

9. 配置机架感知的下面哪项正确

- a)如果一个机架出问题，不会影响数据读写
- b)写入数据的时候会写到不同机架的 **DataNode** 中
- c)MapReduce 会根据机架获取离自己比较近的网络数据

答案 ABC

具体可以参考

[hadoop 机架感知--加强集群稳固性，该如何配置 hadoop 机架感知](#)

10. Client 端上传文件的时候下列哪项正确

- a)数据经过 **NameNode** 传递给 **DataNode**
- b)Client 端将文件切分为 **Block**，依次上传
- c)Client 只上传数据到一台 **DataNode**，然后由 **NameNode** 负责 **Block** 复制工作

答案 B

分析：

Client 向 **NameNode** 发起文件写入的请求。

NameNode 根据文件大小和文件块配置情况，返回给 **Client** 它所管理部分 **DataNode** 的信息。

Client 将文件划分为多个 **Block**，根据 **DataNode** 的地址信息，按顺序写入到每一个 **DataNode** 块中。

具体查看

[HDFS 体系结构简介及优缺点](#)

11. 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。

12. Cloudera 提供哪几种安装 CDH 的方法

- a)Cloudera manager b)Tarball c)Yum d)Rpm

答案：ABCD 具体可以参考

[Hadoop CDH 四种安装方式总结及实例指导](#)

判断题：

13. Ganglia 不仅可以进行监控，也可以进行告警。（正确）

分析：

此题的目的是考 Ganglia 的了解。严格意义上来讲是正确。

ganglia 作为一款最常用的 Linux 环境中的监控软件，它擅长的的是从节点中按照用户的需求以较低的代价采集数据。但是 ganglia 在预警以及发生事件后通知用户上并不擅长。最新的 ganglia 已经有了部分这方面的功能。但是更擅长做警告的还有 Nagios。Nagios，就是一款精于预警、通知的软件。通过将 Ganglia 和 Nagios 组合起来，把 Ganglia 采集的数据作为 Nagios 的数据源，然后利用 Nagios 来发送预警通知，可以完美的实现一整套监控管理的系统。

具体可以查看

[完美集群监控组合 ganglia 和 nagios](#)

14. Block Size 是不可以修改的。（错误）

它是可以被修改的

Hadoop 的基础配置文件是 `hadoop-default.xml`，默认建立一个 Job 的时候会建立 Job 的 Config，Config 首先读入 `hadoop-default.xml` 的配置，然后再读入 `hadoop-site.xml` 的配置（这个文件初始的时候配置为空），`hadoop-site.xml` 中主要配置需要覆盖的 `hadoop-default.xml` 的系统级配置。具体配置可以参考下

```
1. <property>
2.   <name>dfs.block.size</name>//block 的大小，单位字节，后面会提到用处，必须是 512 的倍数，因为采用 crc 作文件完整性校验，默认配置 512 是 checksum 的最小单元。
3.   <value>512000</value>
4.   <description>The default block size for new files.</description>
5. </property>
```

15. Nagios 不可以监控 Hadoop 集群，因为它不提供 Hadoop 支持。（错误）

分析：

Nagios 是集群监控工具，而且是云计算三大利器之一

16. 如果 NameNode 意外终止，SecondaryNameNode 会接替它使集群继续工作。（错误）

分析：

SecondaryNameNode 是帮助恢复，而不是替代，如何恢复，可以查看 [hadoop 根据 SecondaryNameNode 恢复 Namenode](#)

17. Cloudera CDH 是需要付费使用的。（错误）

分析:

第一套付费产品是 Cloudera Enterprise, Cloudera Enterprise 在美国加州举行的 Hadoop 大会 (Hadoop Summit) 上公开, 以若干私有管理、监控、运作工具加强 Hadoop 的功能。收费采取合约订购方式, 价格随用的 Hadoop 叢集大小变动。

18. Hadoop 是 Java 开发的, 所以 MapReduce 只支持 Java 语言编写。(错误)

分析:

rhadoop 是用 R 语言开发的, MapReduce 是一个框架, 可以理解是一种思想, 可以使用其他语言开发。

具体可以查看

[Hadoop 简介\(1\):什么是 Map/Reduce](#)

19. Hadoop 支持数据的随机读写。(错)

分析:

lucene 是支持随机读写的, 而 hdfs 只支持随机读。但是 HBase 可以来补救。

HBase 提供随机读写, 来解决 Hadoop 不能处理的问题。HBase 自底层设计开始即聚焦于各种可伸缩性问题: 表可以很“高”, 有数十亿个数据行; 也可以很“宽”, 有数百万个列; 水平分区并在上千个普通商用机节点上自动复制。表的模式是物理存储的直接反映, 使系统有可能提高高效的数据结构的序列化、存储和检索。

20. NameNode 负责管理 metadata, client 端每次读写请求, 它都会从磁盘中读取或则会写入 metadata 信息并反馈 client 端。(个人认为正确, 欢迎提出其它意见) 分析:

1) 文件写入

Client 向 NameNode 发起文件写入的请求。

NameNode 根据文件大小和文件块配置情况, 返回给 Client 它所管理部分 DataNode 的信息。

Client 将文件划分为多个 Block, 根据 DataNode 的地址信息, 按顺序写入到每一个 DataNode 块中。

2) 文件读取

Client 向 NameNode 发起文件读取的请求。

NameNode 返回文件存储的 DataNode 的信息。

Client 读取文件信息。

具体查看

[hadoop 中 NameNode、DataNode 和 Client 三者之间协作关系](#)

21. NameNode 本地磁盘保存了 Block 的位置信息。(个人认为正确, 欢迎提出其它意见)

分析:

DataNode 是文件存储的基本单元，它将 **Block** 存储在本地文件系统中，保存了 **Block** 的 **Meta-data**，同时周期性地将所有存在的 **Block** 信息发送给 **NameNode**。

具体同样查看

[hadoop 中 NameNode、DataNode 和 Client 三者之间协作关系](#)

22. DataNode 通过长连接与 NameNode 保持通信。（错误）

首先明确一下概念：

（1）.长连接

Client 方与 **Server** 方先建立通讯连接，连接建立后不断开，然后再进行报文发送和接收。这种方式下由于通讯连接一直存在，此种方式常用于点对点通讯。

（2）.短连接

Client 方与 **Server** 每进行一次报文收发交易时才进行通讯连接，交易完毕后立即断开连接。此种方式常用于一点对多点通讯，比如多个 **Client** 连接一个 **Server**。

23. 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。（错误）

hadoop 只能阻止好人犯错，但是不能阻止坏人干坏事

具体可查看

[hadoop 安全性需不断加强](#)

24. Slave 节点要存储数据，所以它的磁盘越大越好。（错误）

分析:

一旦 **Slave** 节点宕机，数据恢复是一个难题

25. `hadoop dfsadmin -report` 命令用于检测 HDFS 损坏块。（错误）

分析:

`hadoop dfsadmin -report`

用这个命令可以快速定位出哪些节点 **down** 掉了，**HDFS** 的容量以及使用了多少，以及每个节点的硬盘使用情况。

当然 **NameNode** 有个 **http** 页面也可以查询，但是这个命令的输出更适合我们的脚本监控 **dfs** 的使用状况

```
1. Configured Capacity: 77209395855360 (70.22 TB)
2. Present Capacity: 76079914600683 (69.19 TB)
3. DFS Remaining: 60534707015680 (55.06 TB)
4. DFS Used: 15545207585003 (14.14 TB)
5. DFS Used%: 20.43%
6.
7. -----
8. Datanodes available: 107 (109 total, 2 dead)
9.
10. Name: 172.16.218.232:50010
11. Rack: /lg/dminterface0
12. Decommission Status : Normal
13. Configured Capacity: 1259272216576 (1.15 TB)
14. DFS Used: 185585852416 (172.84 GB)
15. Non DFS Used: 39060951040 (36.38 GB)
16. DFS Remaining: 1034625413120(963.57 GB)
17. DFS Used%: 14.74%
18. DFS Remaining%: 82.16%
19. Last contact: Wed Nov 18 10:19:44 CST 2009
20.
21. Name: 172.16.216.126:50010
22. Rack: /lg/dminterface2
23. Decommission Status : Normal
24. Configured Capacity: 661261402112 (615.85 GB)
25. DFS Used: 123147280384 (114.69 GB)
26. Non DFS Used: 8803852288 (8.2 GB)
27. DFS Remaining: 529310269440(492.96 GB)
28. DFS Used%: 18.62%
```

29. DFS Remaining%: 80.05%

30. Last contact: Wed Nov 18 10:19:46 CST 2009

26. Hadoop 默认调度器策略为 FIFO（正确）

具体参考

[Hadoop 集群三种作业调度算法介绍](#)

27. 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。（错误）

分析：

首先明白什么是 RAID，可以参考百科[磁盘阵列](#)。

这句话错误的地方在于太绝对，具体情况具体分析。题目不是重点，知识才是最重要的。

因为 hadoop 本身就具有冗余能力，所以如果不是很严格不需要都配备 RAID。具体参考第二题。

28. 因为 HDFS 有多个副本，所以 NameNode 是不存在单点问题的。（错误）

分析：

NameNode 存在单点问题。了解详细信息，可以参考

[Hadoop 中 Namenode 单点故障的解决方案及详细介绍 AvatarNode](#)

29. 每个 map 槽就是一个线程。（错误）

分析：首先我们知道什么是 map 槽, map 槽->map slot

map slot

只是一个逻辑值

(

org.apache.hadoop.mapred.TaskTracker.TaskLauncher.numFreeSlots)，而不是对应着一个线程或者进程

具体见：

[hadoop 中槽-slot 是线程还是进程讨论](#)

30. Mapreduce 的 input split 就是一个 block。（错误）

[InputFormat 的数据划分、Split 调度、数据读取三个问题的浅析](#)

31. NameNode 的 Web UI 端口是 50030，它通过 jetty 启动的 Web 服务。（错误）

分析：

根据下面，很显然 JOBTRACKER 的 Web UI 端口是 50030

默认端口	设置位置
9000	namenode
8020	namenode
8021	JT RPC
50030	mapred.job.tracker.http.address JobTracker administrative web GUI
50070	dfs.http.address NameNode administrative web GUI
50010	dfs.datanode.address DataNode control port
50020	dfs.datanode.ipc.address DataNode IPC port, used for block transfer
50060	mapred.task.tracker.http.address Per TaskTracker web interface
50075	dfs.datanode.http.address Per DataNode web interface
50090	dfs.secondary.http.address Per secondary NameNode web interface

设置位置	描述信息
namenode	交互端口
namenode	RPC交互端口
JT RPC	交互端口
mapred.job.tracker.http.address JobTracker administrative web GUI	JOBTRACKER的HTTP服务器和端口
dfs.http.address NameNode administrative web GUI	NAMENODE的HTTP服务器和端口
dfs.datanode.address DataNode control port	DATANODE控制端口，主要用于DATANODE初始化时向NAMENODE提出注册和应答请求
dfs.datanode.ipc.address DataNode IPC port, used for block transfer	DATANODE的RPC服务器地址和端口
mapred.task.tracker.http.address Per TaskTracker web interface	TASKTRACKER的HTTP服务器和端口
dfs.datanode.http.address Per DataNode web interface	DATANODE的HTTP服务器和端口
dfs.secondary.http.address Per secondary NameNode web interface	辅助DATANODE的HTTP服务器和端口

32. 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 200 GB。（错误）

hadoop 为各个守护进程（namenode,secondarynamenode,jobtracker,datanode,tasktracker）统一分配的内存存在 hadoop-env.sh 中设置，参数为 HADOOP_HEAPSIZE，默认为 1000M。具体参考[hadoop 集群内存设置](#)

33. DataNode 首次加入 cluster 的时候，如果 log 中报告不兼容文件版本，那需要 NameNode 执行“Hadoop namenode -format”操作格式化磁盘。（错误）

分析：

首先明白介绍，什么 ClusterID

ClusterID

添加了一个新的标识符 ClusterID 用于标识集群中所有的节点。当格式化一个 Namenode，需要提供这个标识符或者自动生成。这个 ID 可以被用来格式化加入集群的其他 Namenode。

详细内容可参考

[hadoop 集群添加 namenode 的步骤及常识](#)

以上答案通过多个资料验证，对于资料不充分的内容，都标有“个人观点”，给出本测试题抱着谨慎的态度，希望大家多批评指正。

出自 about 云

本文链接：

<http://www.aboutyun.com/thread-6787-1-1.html>

**BIG
DATAS**

大数据中国社区网邀请您加入

<http://www.bigdatas.cn>

社区QQ群： 300041458

招募版主、合作伙伴