

第一周报告

学院数据分析项目部署,HADOOP安装与测试

By: 柳杰

April 17, 2017

1 本周所做任务列表

- 部署学院数据分析web项目
- 在单节点安装并测试Hadoop

2 部署学院数据分析项目

2.1 项目基本介绍

学院的数据分析项目是一个简单的javaweb应用程序,由springmvc,spring,mybatis,mysql组成基本项目架构. 由于azure平台上建立好的虚拟服务器是裸机环境,所以项目部署之前需要对服务器进行最基本的配置.

- 安装jdk
- 安装mysql
- 安装tomcat
- 上传并部署web应用

2.2 项目设计上的一些总结

2.2.1 关于数据库的设计简介

一开始我们考虑每个模块用一张大的关系表解决问题,后来经过测试确实单表虽然在基本sql操作上面简单,但是增加了数据维护的难度.更为致命的是当单表数据量增大的时候,CRUD操作性能将会急剧下降,所以我们使用分多表的方案.

2.2.2 sql语句优化

在sql语句方面我们进行了如下优化

- 通过开启慢查询日志来找出较慢的SQL
- 不做列运算: SELECT id WHERE age + 1 = 10, 任何对列的操作都将导致表扫描, 它包括数据库教程函数、计算表达式等等, 查询时要尽可能将操作移至等号右边
- 不用SELECT *
- OR改写成IN: OR的效率是n级别, IN的效率是log(n)级别, in的个数建议控制在200以内
- 尽量避免在WHERE子句中使用!=或<>操作符, 否则将引擎放弃使用索引而进行全表扫描
- 列表数据不要拿全表, 要使用LIMIT来分页, 每页数量也不要太大
- 使用同类型进行比较, 比如用'123'和'123'比, 123和123比

2.2.3 mybatis动态表名操作的配置

- 接口的动态参数注入:

```
public int insertDynamic(@Param("items")T entity , @Param("tableName")String tableName);
```
- 配置文件中用 \$tableName 动态的传入表名到sql语句中.

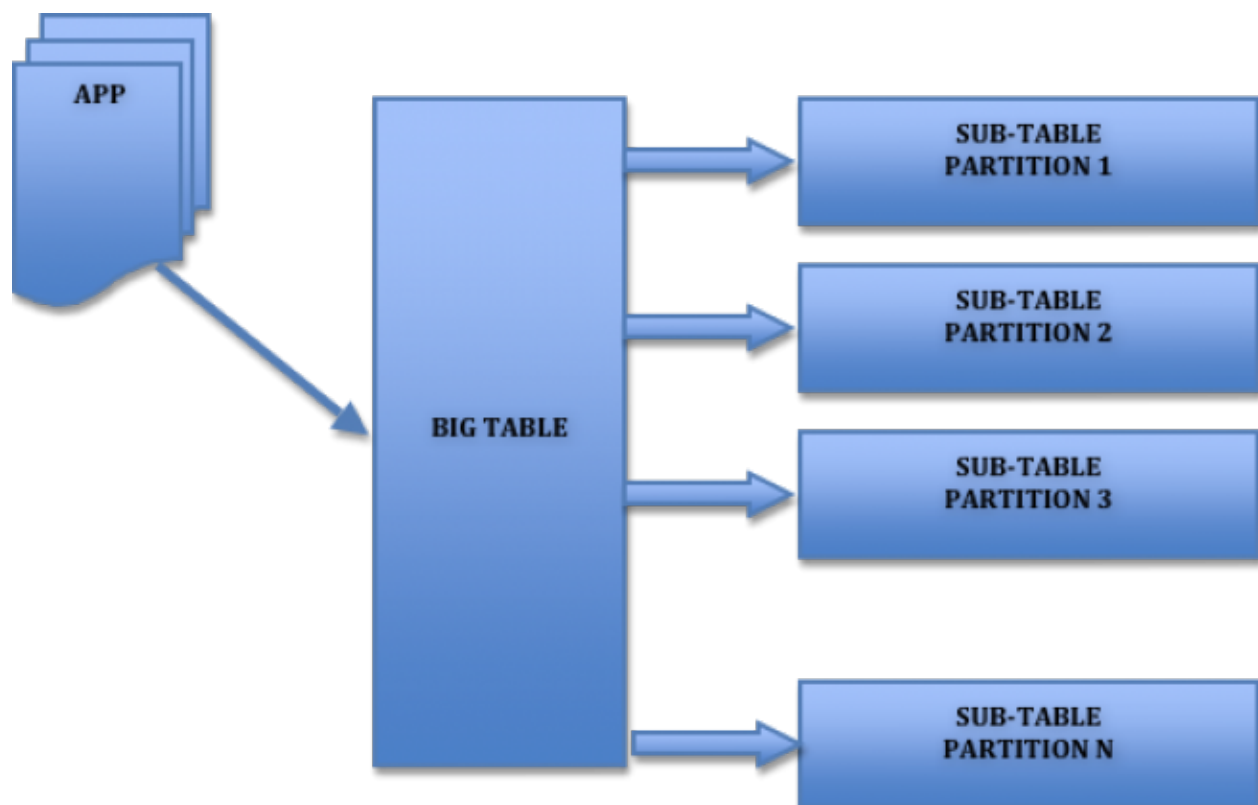


Figure 1: 大表拆分成小表

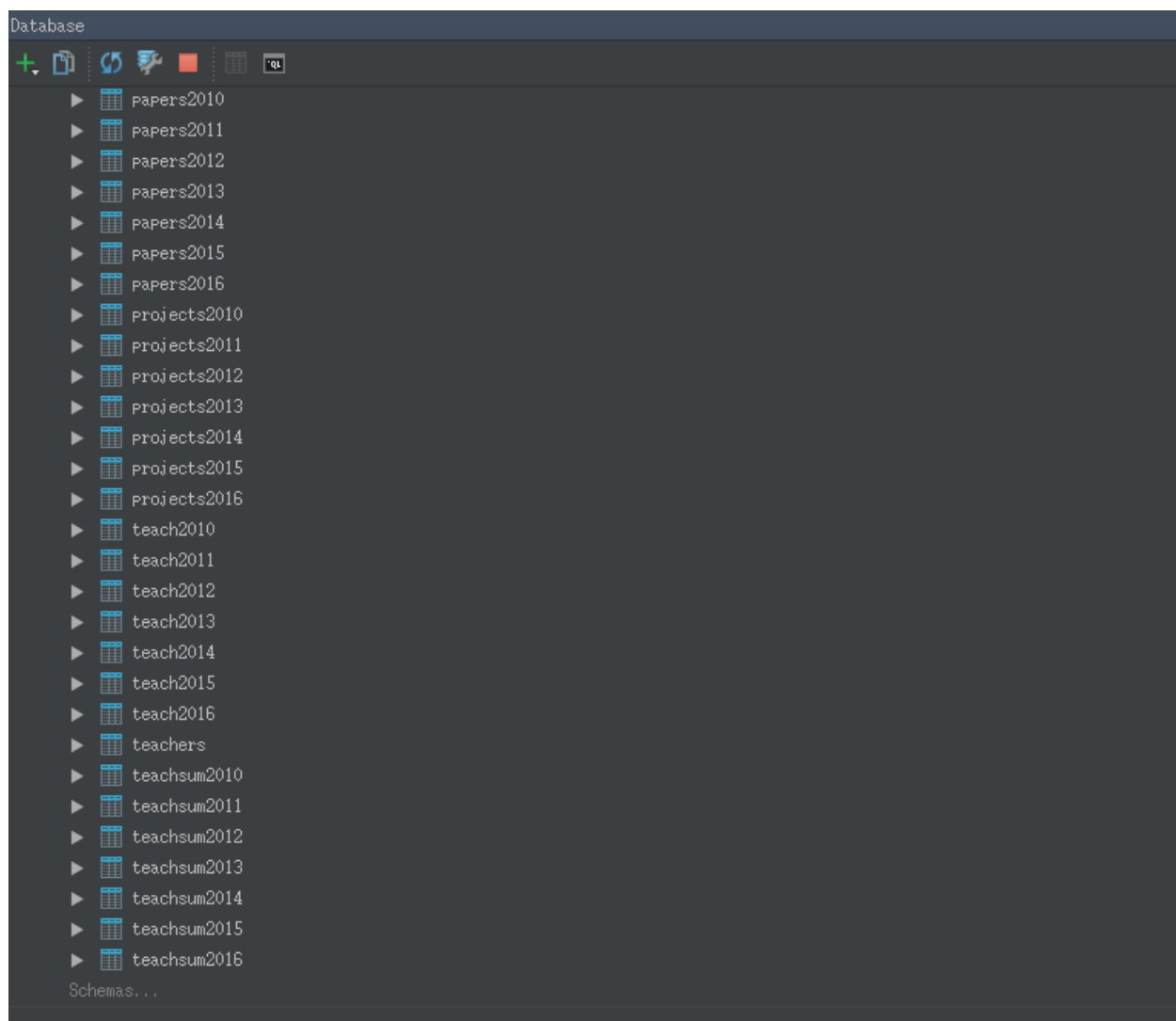


Figure 2: 数据库最终设计

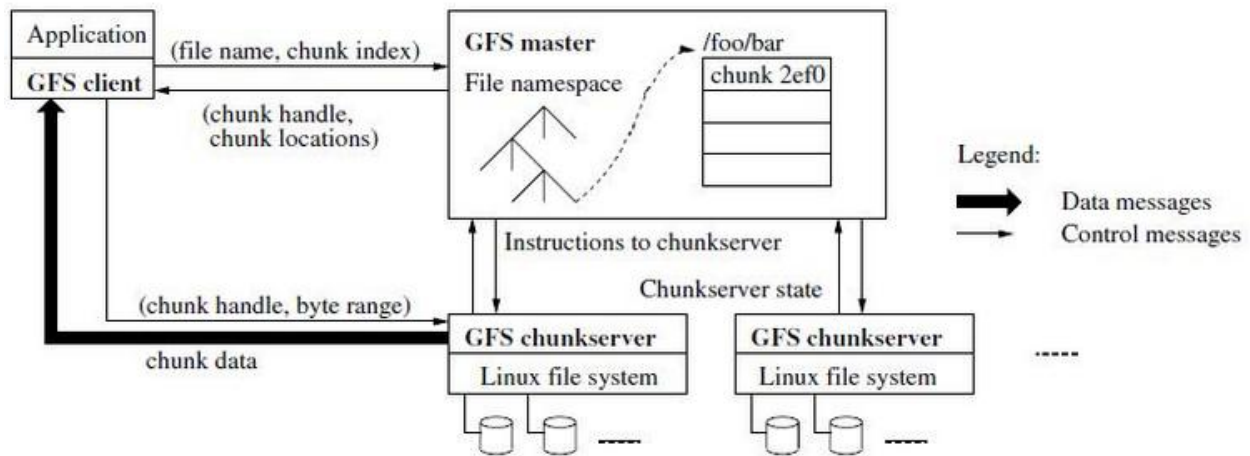


Figure 1: GFS Architecture

Figure 3: GFS架构

3 Hadoop的安装与测试

3.1 Hadoop之前的理论基础

3.1.1 GFS架构

一个GFS集群中包含一个单独的Master节点,多台Chunk服务器,并且被多个客户端访问,如下图所示。所有的这些机器通常都是普通的Linux机器,运行着用户级别(user-level)的服务进程。我们可以很容易的把Chunk服务器和客户端都放在同一台机器上,前提是机器资源允许,并且我们能够接受不可靠的应用程序代码带来的稳定性降低的风险。GFS存储的文件都被分割成固定大小的Chunk。在Chunk创建的时候,Master服务器会给每个Chunk分配一个不变的、全球唯一的64位的Chunk标识。Chunk服务器把Chunk以linux文件的形式保存在本地硬盘上,并且根据指定的Chunk标识和字节范围来读写块数据。出于可靠性的考虑,每个块都会复制到多个块服务器上。缺省情况下,我们使用3个存储复制节点,不过用户可以为不同的文件命名空间设定不同的复制级别。

Master节点管理所有的文件系统元数据。这些元数据包括名字空间、访问控制信息、文件和Chunk的映射信息、以及当前Chunk的位置信息。Master节点还管理着系统范围内的活动,比如,Chunk租用管理、孤儿Chunk的回收、以及Chunk在Chunk服务器之间的迁移。Master节点使用心跳信息周期地和每个Chunk服务器通讯,发送指令到各个Chunk服务器并接收Chunk服务器的状态信息。

GFS客户端代码以库的形式被链接到客户程序里。客户端代码实现了GFS文件系统的API接口函数、应用程序与Master节点和Chunk服务器通讯、以及对数据进行读写操作。客户端和Master节点的通信只获取元数据,所有的数据操作都是由客户端直接和Chunk服务器进行交互的。我们不提供POSIX标准的API的功能,因此,GFS API调用不需要深入到Linux vnode级别。

无论是客户端还是Chunk服务器都不需要缓存文件数据。客户端缓存数据几乎没有什么用处,因为大部分程序要么以流的方式读取一个巨大文件,要么工作集太大根本无法被缓存。无需考虑缓存相关的问题也简化了客户端和整个系统的设计和实现。(不过,客户端会缓存元数据。)Chunk服务器不需要缓存文件数据的原因是,Chunk以本地文件的方式保存,Linux操作系统的文件系统缓存会把经常访问的数据缓存在内存中。

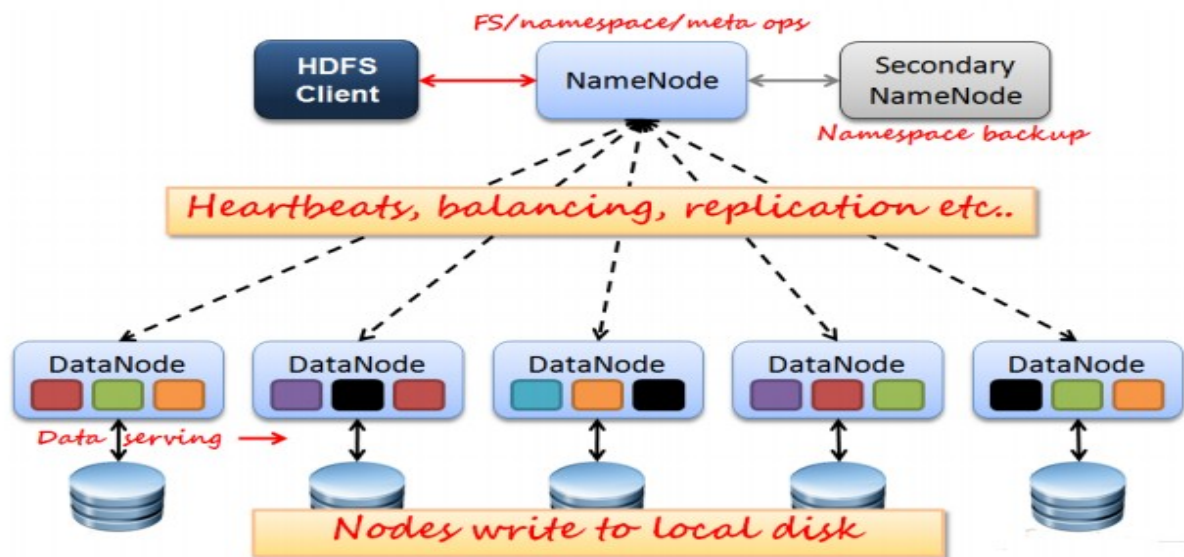


Figure 4: hdfs架构

3.1.2 HDFS基本概念

- **NameNode**: 是Master节点，是大领导。管理数据块映射；处理客户端的读写请求；配置副本策略；管理HDFS的名称空间；
- **SecondaryNameNode**: 是一个小弟，分担大哥namenode的工作量；是NameNode的冷备份；合并fsimage和fsedits然后再发给namenode。
- **DataNode**: Slave节点，奴隶，干活的。负责存储client发来的数据块block；执行数据块的读写操作。
- **热备份**: b是a的热备份，如果a坏掉。那么b马上运行代替a的工作。
- **冷备份**: b是a的冷备份，如果a坏掉。那么b不能马上代替a工作。但是b上存储a的一些信息，减少a坏掉之后的损失。
- **fsimage**: 元数据镜像文件（文件系统的目录树。）
- **edits**: 元数据的操作日志（针对文件系统做的修改操作记录）
- **namenode**内存中存储的是=fsimage+edits。
- **SecondaryNameNode**负责定时默认1小时，从namenode上，获取fsimage和edits来进行合并，然后再发送给namenode。减少namenode的工作量。

3.1.3 MapReduce 作业的生命周期

- **步骤1 作业提交与初始化**. 用户提交作业后, 首先由JobClient实例将作业相关信息, 比方将程序jar包、作业配置文件、分片元信息文件等上传到分布式文件系统（一般为HDFS）上, 当中, 分片元信息文件记录了每一个输入分片的逻辑位置信息. 然后JobClient通过RPC通知JobTracker. JobTracker 收到新作业提交请求后, 由作业调度模块对作业进行初始化: 为作业创建一个JobInProgress对象以跟踪作业执行状况, 而JobInProgress 则会为每一个Task 创建一个TaskInProgress 对象以跟踪每一个任务的执行状态, TaskInProgress 可能须要管理多个“Task 执行尝试”（称为“TaskAttempt”）。

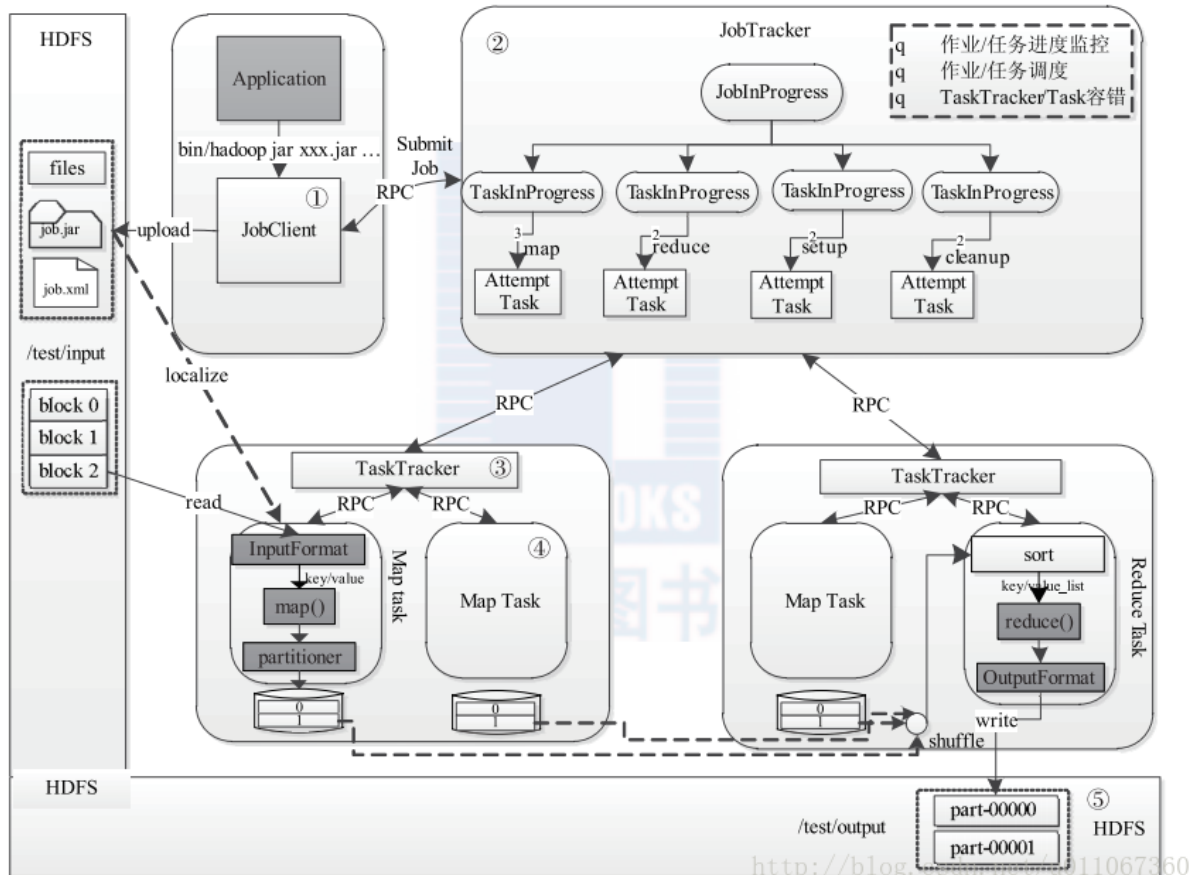


Figure 5: MapReduce作业生命周期

- 步骤2 任务调度与监控。前面提到，任务调度和监控的功能均由JobTracker 完毕。TaskTracker 周期性地通过Heart beat 向JobTracker 汇报本节点的资源使用情况，一旦出现 空暇资源，JobTracker 会依照一定的策略选择一个合适的任务使用该空暇资源，这由任务调度器完毕。任务调度器是一个可插拔的独立模块，且为双层架构，即首先选择作业，然后从该作业中选择任务，当中，选择任务时须要重点考虑数据本地性。此外，JobTracker 跟踪作业的整个执行过程，并为作业的成功执行提供全方位的保障。首先，当TaskTracker 或者Task 失败时，转移计算任务；其次，当某个Task 执行进度远落后于同一作业的其它Task 时，为之启动一个同样Task，并选取计算快的Task 结果作为终于结果。
- 步骤3 任务执行环境准备。执行环境准备包含JVM 启动和资源隔离，均由TaskTracker 实现。TaskTracker 为每一个Task 启动一个独立的JVM 以避免不同Task 在执行过程中相互影响；同一时候，TaskTracker 使用了操作系统进程实现资源隔离以防止Task 滥用资源。
- 步骤4 任务执行。TaskTracker 为Task 准备好执行环境后，便会启动Task。在执行过程中，每一个Task 的最新进度首先由Task 通过RPC 汇报给TaskTracker，再由TaskTracker汇报给JobTracker。
- 步骤5 作业完毕。待全部Task 运行完毕后，整个作业运行成功。

3.2 hadoop单节的安装与配置

3.2.1 获取软件安装包

- 用wget命令获取hadoop安装包并解压
- 配置hadoop和java基本应用环境(etc/profile)

```
1 export HADOOP_HOME=/home/liujie/software/hadoop-1.2.1
3 export JAVA_HOME=/usr/jdk1.8.0_121
5 export JRE_HOME=$JAVA_HOME/jre
7 export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
9 export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$HADOOP_HOME/bin:$PATH
```

- 配置core-site.xml文件

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7
8   <property>
9
10     <name>hadoop.tmp.dir</name>
11     <value>/home/liujie/hadoop</value>
12
13   </property>
14
15   <property>
16
17     <name>dfs.name.dir</name>
18     <value>/hadoop/name</value>
19
20   </property>
21
22   <property>
23
24     <name>fs.default.name</name>
25     <value>hdfs://liujie-Lenovo-Y430P:9000</value>
26
27   </property>
28
29 </configuration>
```

- 配置hadoop-env.sh

```
1 # Set Hadoop-specific environment variables here.
2
3 # The only required environment variable is JAVA_HOME. All others are
4 # optional. When running a distributed configuration it is best to
5 # set JAVA_HOME in this file, so that it is correctly defined on
6 # remote nodes.
7
8 # The java implementation to use. Required.
9 export JAVA_HOME=/usr/jdk1.8.0_121
10
11 # Extra Java CLASSPATH elements. Optional.
12 # export HADOOP_CLASSPATH=
13
14 # The maximum amount of heap to use, in MB. Default is 1000.
15 # export HADOOP_HEAPSIZE=2000
16
17 # Extra Java runtime options. Empty by default.
18 # export HADOOP_OPTS=-server
19
20 # Command specific options appended to HADOOP_OPTS when specified
21 export HADOOP_NAMENODE_OPTS="-Dcom.sun.management.jmxremote_${HADOOP_NAMENODE_OPTS}"
22 export HADOOP_SECONDARYNAMENODE_OPTS="-Dcom.sun.management.jmxremote_${HADOOP_SECONDARYNAMENODE_OPTS}"
23 export HADOOP_DATANODE_OPTS="-Dcom.sun.management.jmxremote_${HADOOP_DATANODE_OPTS}"
24 export HADOOP_BALANCER_OPTS="-Dcom.sun.management.jmxremote_${HADOOP_BALANCER_OPTS}"
25 export HADOOP_JOBTRACKER_OPTS="-Dcom.sun.management.jmxremote_${HADOOP_JOBTRACKER_OPTS}"
26 # export HADOOP_TASKTRACKER_OPTS=
27 # The following applies to multiple commands (fs, dfs, fsck, distcp etc)
28 # export HADOOP_CLIENT_OPTS
29
30 # Extra ssh options. Empty by default.
31 # export HADOOP_SSH_OPTS="-o ConnectTimeout=1 -o SendEnv=HADOOP_CONF_DIR"
```



```

33 # Where log files are stored. $HADOOP_HOME/logs by default.
34 # export HADOOP_LOG_DIR=${HADOOP_HOME}/logs
35
36 # File naming remote slave hosts. $HADOOP_HOME/conf/slaves by default.
37 # export HADOOP_SLAVES=${HADOOP_HOME}/conf/slaves
38
39 # host:path where hadoop code should be rsync'd from. Unset by default.
40 # export HADOOP_MASTER=master:/home/$USER/src/hadoop
41
42 # Seconds to sleep between slave commands. Unset by default. This
43 # can be useful in large clusters, where, e.g., slave rsyncs can
44 # otherwise arrive faster than the master can service them.
45 # export HADOOP_SLAVE_SLEEP=0.1
46
47 # The directory where pid files are stored. /tmp by default.
48 # NOTE: this should be set to a directory that can only be written to by
49 # the users that are going to run the hadoop daemons. Otherwise there is
50 # the potential for a symlink attack.
51 # export HADOOP_PID_DIR=/var/hadoop/pids
52
53 # A string representing this instance of hadoop. $USER by default.
54 # export HADOOP_IDENT_STRING=$USER
55
56 # The scheduling priority for daemon processes. See 'man nice'.
57 # export HADOOP_NICENESS=10

```

● 配置hdfs-site.xml

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7
8   <property>
9
10     <name>dfs.data.dir</name>
11     <value>/hadoop/data</value>
12
13   </property>
14
15 </configuration>

```

● 配置mapred-site.xml

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7
8   <property>
9
10     <name>mapred.job.tracker</name>
11     <value>liujie-Lenovo-Y430P:9001</value>
12
13   </property>
14
15 </configuration>

```

3.2.2 安装过程中遇到的问题和解决的办法

问题描述 启动hadoop报错 Permission denied, please try again

解决办法

- 安装open-ssh服务
- 使用ssh-keygen命令生成公钥,可以让hadoop免去密码访问一些服务
- 去掉AuthorizedKeysFile前面的# 号注释
- 修改 etc - ssh - sshd _ config里面的PermitRootLogin without-password 为 PermitRootLogin yes

```
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$ ./start-all.sh
Warning: HADOOP_HOME is deprecated
64
starting namenode, logging to /home/liujie/software/hadoop-1.2.1/libexec/../logs/hadoop-liujie-namenode-liujie-Lenovo-Y430P.out
liujie@localhost:~$ password:
localhost: starting datanode, logging to /home/liujie/software/hadoop-1.2.1/libexec/../logs/hadoop-liujie-datanode-liujie-Lenovo-Y430P.out
liujie@localhost:~$ password:
localhost: starting secondarynamenode, logging to /home/liujie/software/hadoop-1.2.1/libexec/../logs/hadoop-liujie-secondarynamenode-liujie-Lenovo-Y430P.out
starting jobtracker, logging to /home/liujie/software/hadoop-1.2.1/libexec/../logs/hadoop-liujie-jobtracker-liujie-Lenovo-Y430P.out
liujie@localhost:~$ password:
localhost: starting tasktracker, logging to /home/liujie/software/hadoop-1.2.1/libexec/../logs/hadoop-liujie-tasktracker-liujie-Lenovo-Y430P.out
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$ jps
4836 Jps
4569 JobTracker
4476 SecondaryNameNode
4750 TaskTracker
4879 NameNode
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$
```

Figure 6: hadoop运行状态

4 Hadoop基本操作

4.1 启动hadoop

进入hadoop的bin目录下,运行start-all.sh文件,启动成功之后,使用JPS查看hadoop运行状态,如果出现上图几个进程启动标识说明启动成功.

4.2 基本命令测试

4.3 下周任务规划

- 1.尝试在多节点上搭建分布式文件系统
- 2.学习并测试使用mapreduce编程模型

```

Most commands print help when invoked w/o parameters.
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$ hadoop fs -ls
Warning: $HADOOP_HOME is deprecated.
Found 1 items
drwxr-xr-x  - liujie supergroup          0 2017-04-15 00:27 /user/liujie/one
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$

```

Figure 7: ls命令

```

usage: java hadoop-1.2.1/bin/hadoop fs -ls
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$ hadoop fs -mkdir hello
Warning: $HADOOP_HOME is deprecated.
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$ hadoop fs -ls
Warning: $HADOOP_HOME is deprecated.
Found 2 items
drwxr-xr-x  - liujie supergroup          0 2017-04-17 22:14 /user/liujie/hello
drwxr-xr-x  - liujie supergroup          0 2017-04-15 00:27 /user/liujie/one
liujie@liujie-Lenovo-Y430P:~/software/hadoop-1.2.1/bin$

```

Figure 8: mkdir命令