

一、Linux 操作系统基本操作题 (每题 1 分, 共 10 分)

1. 登录 linux, 打印自己的主目录, 当前目录;

```
xuan@Hadoop:~$ ls
anaconda2  Documents  examples.desktop  mybash.sh  myjava  Public  Videos
bigdata    Downloads  linux.txt         myc         mypy    software
Desktop    example    Music             mydir       Pictures Templates
xuan@Hadoop:~$ pwd
/home/xuan
```

2. 在自己主目录下, 建立 mydir 文件夹, 在该文件夹中建立 main.py 文件;

```
xuan@Hadoop:~$ mkdir mydir
xuan@Hadoop:~$ cd mydir/
xuan@Hadoop:~/mydir$ touch main.py
xuan@Hadoop:~/mydir$ ls
main.py
```

3. 重命名上面 main.py 文件为 mytest.py, 修改其权限为可执行;

```
xuan@Hadoop:~/mydir$ mv main.py mytest.py
xuan@Hadoop:~/mydir$ ls
mytest.py
xuan@Hadoop:~/mydir$ chmod 777 mytest.py
xuan@Hadoop:~/mydir$ ls
mytest.py
```

4. chmod 777 myfile, chown myfile hadoop 各代表什么意思?

chmod 777 myfile : myfile 文件修改为可读可写可执行文件

chown myfile hadoop: hadoop 的权限路径修改为 myfile 文件

5. 复制主目录下 mydir 文件夹中所有的文件到主目录;

```
xuan@Hadoop:~$ cp -r mydir/. ~
xuan@Hadoop:~$ ls
anaconda2  Documents  examples.desktop  mybash.sh  myjava  Pictures  Templates
bigdata    Downloads  linux.txt         myc         mypy    Public    Videos
Desktop    example    Music             mydir       mytest.py software
```

6. Linux, shell 命令中, /, ~, -, >, >> 着这个符号代表什么意思? 请举例说明;

/ : 在路径表示时代表根目录。

~ : 账号的主目录。

- : (1) 减法; (2) 系统指令的选项符号; (3) 从标准输入中读取资料; (4)

cd - 变更工作目录到"上一次"工作目录。

> : 把命令的输出重定向到文件中, 若文件已经存在, 则清空原有文件。

>> : 把命令的输出重定向到文件中, 若文件已经存在, 则把信息加在原有文件后面。

7. Linux 主目录和根目录符号

主目录: ~ 根目录: /

8.请例举 cat , more , less 的用法;

cat: (1)显示文件内容; (2)创建新文件; (3)合并文件; (4)复制文件内容。

```
xuan@Hadoop:~$ cat ./bigdata/mydata.txt > 1.txt
xuan@Hadoop:~$ cat 1.txt
1001,Tom,101,21,2000
2001,Mary,102,20,2800
3001,Heny,101,22,3200
4001,Alice,102,19,1800xuan@Hadoop:~$ cat 1.txt > 2.txt
xuan@Hadoop:~$ cat 1.txt >> 2.txt
xuan@Hadoop:~$ cat 2.txt
1001,Tom,101,21,2000
2001,Mary,102,20,2800
3001,Heny,101,22,3200
4001,Alice,102,19,18001001,Tom,101,21,2000
2001,Mary,102,20,2800
3001,Heny,101,22,3200
4001,Alice,102,19,1800xuan@Hadoop:~$
```

more: 浏览文件内容。 (1)-f: 计算行数时, 以实际的行数; (2)-p: 先清除屏幕后再显示内容; (3)-c: 先显示内容再清除其他旧资料; (4)-s: 当遇到有连续两行以上的空白行时, 就替换为一行的空白行; (5)-u: 不显示下引号; (6)+n: 从第 n 行开始显示文件内容; (7)-n: 一次显示的行数。

less 查看文件内容。 (1)-N: 显示每行的行号; (2)-S: 行过长时将超出部分舍弃; (3)-e: 当文件显示结束后, 自动离开; (4)-g: 只标志最后搜索到的关键词; (5)-Q:

不使用警告音; (6)-i: 忽略搜索时的大小写; (7)-m: 显示类似 more 命令的百分比; (8)-f: 强迫打开特殊文件; (9)-s: 显示连续空行为一行。

9.在 shell 中如何正确的关机、重启电脑?

关机: `shutdown -h now`

重启: `shutdown -r now`

10.如何在 Linux 环境中运行用户程序。

.sh 文件: `source mybash.sh`

Python: `python hello.py`

C 语言: `gcc hello.c -o hello`

`./hello`

Java : `javac hello.java`

`java hello`

二、简答题 (每题 6 分 共 30 分)

1.大数据的特点有哪一些?

数据体量巨大,大数据的采集、计算、存储量都非常庞大;种类众多,来源多样化;价值密度相对较低;数据增长、处理和获取的速度都很快。

2.在大数据应用平台框架中,你了解的工具具有哪一些,请至少例举3个以上及其功能。

(1) Hadoop: Hdfs 存储数据, MapReduce 计算数据, Yarn 调度资源, 实现对大规模数据的高效处理。

(2) Spark: 具有改进的数据流处理的批处理框架, 通过内存计算, 实现对大批量实时数据的处理, 基于 Hadoop 架构, 弥补了 Hadoop 在实时数据处理上的不足。

(3) Storm: 作为一个实时处理流式数据的计算框架, 可以简单、高效、可靠地处理流式数据并支持多种语言, 它能与多种系统进行整合, 从而开发出更强大的实时计算系统。

3.Hadoop 中 HDFS 是什么? 请简单叙述其工作原理。

HDFS 即 Hadoop 分布式文件系统, 以流式数据访问模式来存储超大文件, 运行于商用硬件集群上, 是管理网络中跨多台计算机存储的文件系统。

HDFS 支持在计算节点之间快速传输数据。在开始阶段, 它与 MapReduce 紧密耦合。当 HDFS 接收数据时, 会将信息分解为单独的块, 并将它们分布到集群中的不同节点, 从而支持高效的并行处理。

4.请叙述 MapReduce 编程模型。

MapReduce 的思想就是“分而治之”。

Mapper 负责“分”, 即把复杂的任务分解为若干个“简单的任务”来处理。“简单的任务”包含三层含义: 一是数据或计算的规模相对原任务要大大缩小; 二是就近计算原则, 即任务会分配到存放着所需数据的节点上进行计算; 三是这些小任务可以并行计算, 彼此间几乎没有依赖关系。

Reducer 负责对 map 阶段的结果进行汇总。

5.请叙述 Spark 主要的数据对象。

(1) RDD 是一个分布式的对象集合, 本质上是一个不可修改的只读的对象集合, 声明的类型是 `JavaRDD<T>`, 其基本单位是一个对象, 不同的 Work 节点上分布着不同的对象。

(2) DataFrame 将数据组织为类似的关系型数据库中表的形式(行列形式), 并且

是带约束的，明确每列的数据类型。

(3) DataSet 的基本形式是 DataSet<T>，而 DataFrame 就可以看成一个特殊的 DataSet，即 DataSet<Row>，他可以完成 DataFrame 的所有功能。

三、HDFS 操作题 (共 20 分，每题 2 分)

1.如何利用 hadoop 命令上传文件 /user/root/data.txt 上传到 HDFS 的 /user/root/目录下? (2 分);

```
hadoop fs -put /user/root/data.txt /user/root
```

2.如何利用 hadoop 命令复制 HDFS 文件/user/root/data.txt 到 HDFS 的 /user/root/tmp 目录下? (2 分);

```
hadoop fs -copyFromLocal /user/root/data.txt /user/root/tmp
```

3.如何利用 hadoop 命令移动 HDFS 文件/user/root/data.txt 到 HDFS 的 /user/root/tmp 目录下? (2 分);

```
hadoop fs -mv /user/root/data.txt /user/root/tmp
```

4.如何利用 hadoop 命令下载 HDFS 文件/user/root/anaconda-ks.cfs 到本地目

录/user/root/tmp? (2 分);

```
hadoop fs -copyToLocal /user/root/anaconda-ks.cfs /user/root/tmp或
```

```
hadoop fs -get/user/root/anaconda-ks.cfs /user/root/tmp
```

5.如何利用 hadoop 命令查看 HDFS 目录/user/root/tmp/所有文件? (2 分);

```
hadoop fs -ls /user/root/tmp
```

6.如何利用 hadoop 命令删除 HDFS 目录/user/root/tmp/下所有文件? (2 分);

```
hadoop fs -rm -r /user/root/tmp
```

7.如何利用 hadoop 命令新建文件夹 mydir? (2 分);

```
hadoop dfs -mkdir /mydir
```

8.如何利用 hadoop 命令删除文件夹/mydir? (2 分);

```
Hadoop fs -rm -r /mydir
```

9.如何利用 hadoop 命令查看 HDFS 文件/user/root/tmp/data.txt 下所有内容?
(2 分);

```
hadoop fs -cat /user/root/tmp/data.txt
```

10.如何利用 hadoop 命令查看文件或文件夹属性? (2 分)

```
hadoop fs -ls -R /
```

四、Mapreduce 程序题目 (共 20 分, 每一题 10 分)

1. 已知文件 mydata.txt 其内容为:

1001, Tom, 101, 21, 2000

2001, Mary, 102, 20, 2800

3001, Henry, 101, 22, 3200

4001, Alice, 102, 19, 1800

列分别代表: 工号(id)、姓名(name)、部门号(Dept.), 年龄(age)、薪水(salary);

(1) 请把如上文件建立上传到 Hadoop HDFS 系统, 并且导入到 SPARK RDD 中

```
source mybash.sh
```

```
$HADOOP_HOME/sbin/start-all.sh
```

```
hadoop fs -copyFromLocal ~/bigdata/mydata.txt /bigdata
```

```
xuan@Hadoop:~$ hadoop fs -ls /bigdata/
Found 2 items
-rw-r--r-- 1 xuan supergroup          90 2022-06-01 20:39 /bigdata/mydata.txt
-rw-r--r-- 1 xuan supergroup    236344 2022-05-07 19:25 /bigdata/u.item
```

```
pyspark
```

```
data=sc.textFile("hdfs:http://192.168.56.102:9000/bigdata/mydata.txt")
```

(2) 利用 PySpark MapReduce 程序统计每个部门的平均薪水。

```
data1=data.map(lambda x:x.split(","))
```

```
data2=data1.map(lambda x:(x[2],x[4]))
```

```
data3=data2.map(lambda x:(x[0],(int(x[1]),1)))
```

```
data4=data3.reduceByKey(lambda x,y:(x[0]+y[0],x[1]+y[1]))
```

```
data5=data4.map(lambda x:(x[0],float(x[1][0])/x[1][1]))
```

```
data5.collect()
```

结果: [(u'102', 2300.0), (u'101', 2600.0)]

2、(1) 利用请利用 PySpark MapReduce 程序统计某个文本含有字母 "a" 的单词个数;

```
data=sc.textFile("/home/xuan/bigdata/The Old Man and the Sea.txt")
```

```
data1=data.flatMap(lambda line:line.split(" "))
```

```
data2=data1.filter(lambda x:'a' in x)
```

```
data2.count()
```

结果：7174

(2) 请举出 PySpark MapReduce 程序的身边一个应用背景，给出核心代码，并解释其含义。

`data=sc.textFile("/home/xuan/bigdata/u.data")`读取文件，文件格式：（用户编号，电影编号，评分，.....）。

`data1=data.map(lambda x:x.split("\t"))`将数据以空格划分为列表。

`data2=data1.map(lambda x:(x[1],x[2]))`将（电影编号，评分）组成元组。

`data3=data2.map(lambda x:(x[0],(int(x[1]),1)))`将数据变为（电影编号，（评分，1）），作好统计准备。

`data4=data3.reduceByKey(lambda x,y:(x[0]+y[0],x[1]+y[1]))`将电影编号相同的 value 进行相加。

`data5=data4.map(lambda x:(x[0],float(x[1][0])/x[1][1]))`评分/个数，即为每个电影的平均评分。

如果想计算每个人评分的平均值，则将（人，评分）组成元组

`data2=data1.map(lambda x:(x[0],x[2]))`其余代码不变。

五、请给出 Spark 在机器学习,推荐算法用，给出流程，给出相应核心代码。(20 分)

如 推荐系统（详细可以参考教材）

ALS 推荐算法：

导入 ALS 的库：

`from pyspark.mllib.recommendation import ALS`

导入数据，格式（用户序号，电影编号，评分，.....）：

`rawdata=sc.textFile("/home/xuan/bigdata/u.data")`

将数据按空格进行划分，提取前三列组成新数据：

`ratingdata=rawdata.map(lambda x:x.split("\t")[0:3])`

将数据组成（用户序号，电影编号，评分）的元组：

`trainingdata=ratingdata.map(lambda x:(x[0],x[1],x[2]))`

训练 ALS 模型：

`from pyspark.mllib.recommendation import Rating`

`model=ALS.train(trainingdata,10,10,0.001)`

应用模型：rating 越高代表系统越加有限向此用户推荐此产品

(1) model.recommendProducts(用户编号, 推荐电影数)

例如向用户 “100” 推荐 5 部电影

```
>>> model.recommendProducts(100,5)
[Rating(user=100, product=34, rating=7.239473870863672), Rating(user=100, product=791, rating=6.9850381672768), Rating(user=100, product=909, rating=6.895647648917226), Rating(user=100, product=1183, rating=6.326411679477529), Rating(user=100, product=1218, rating=6.1959932863511975)]
```

(2) model.predict(用户编号, 电影编号)

例如预测用户 “100” 对电影 “1141” 的评分

```
>>> model.predict(100,1141)
3.0992282922961594
```

(3) model.recommendUsers(电影编号, 用户数)

例如推荐电影 “1141” 给 5 位用户

```
>>> model.recommendUsers(1141,5)
[Rating(user=93, product=1141, rating=18.25368290339115), Rating(user=475, product=1141, rating=14.003104420619199), Rating(user=309, product=1141, rating=12.264546258655532), Rating(user=898, product=1141, rating=11.86330250413172), Rating(user=681, product=1141, rating=11.583405478654019)]
```