华东师范大学数据学院上机实践报告

课程名称：分布式模型与编程	年级：2018	上机实践成绩:
指导教师：徐辰		姓名：孙秋实
上机实践名称：Spark 部署	学号：10185501402	上机实践日期：2021/4/8
上机实践编号：Lab7	组号：Group5	上机实践时间:

Part 1
实验目的

(1) 学习 Spark 的部署，简单实用 Spark-Shell

(2) 查看 Spark 的运行日志，体会与 MapReduce 运行过程中日志的区别

(3) 通过系统部署理解体系架构，体会 Spark 与 MapReduce 之间的区别

Part 2
实验任务

(1) 完成 Spark 的单机集中式部署、单机伪分布式部署以及分布式部署

(2) 在上述部署方式下分别以 Client 和 Cluster 模式运行示例程序

Part 3
使用环境

(1) 操作系统：Ubuntu 18.04

(2) JDK 版本：1.8

(3) Spark 版本：2.4.7

(4) Hadoop 版本：2.10.1

Part 4
实验过程

Section 1
单机集中式部署

在实验开始前，先登录 dase-local 用户，并且修改 *.bashrc* 文件

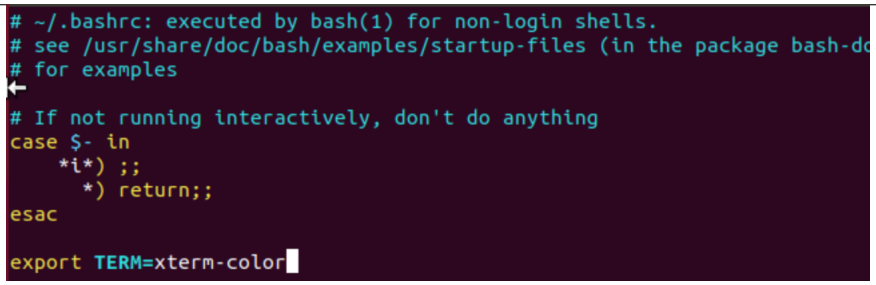


图 1: 添加 xterm-color

随后使用命令 *source /.bashrc* 使之生效

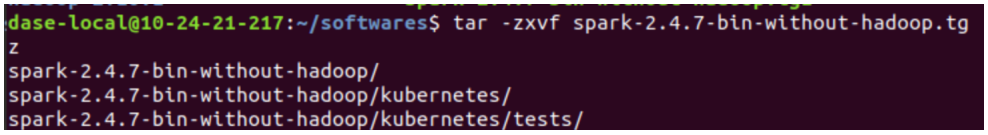


图 2: 使用本地安装包安装 Spark

修改配置文件，如下所示

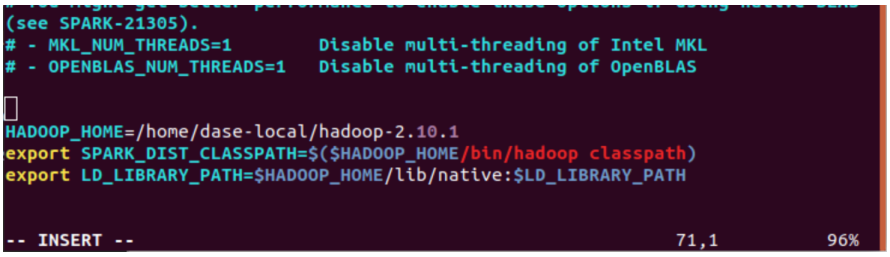


图 3: 修改配置文件 spark-env.sh

随后，使用命令 *spark-shell –masterlocal* 进入 Spark Shell

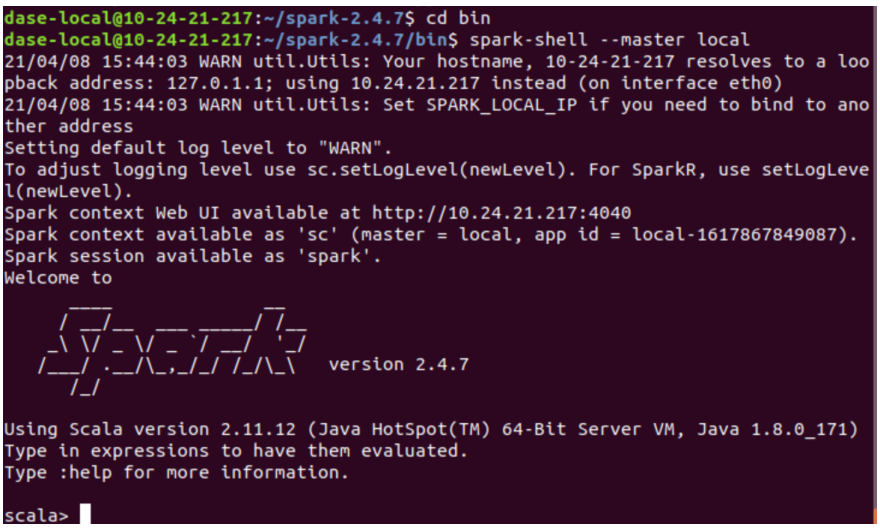


图 4: 进入 Spark-Shell

输入 Scala 代码，运行状态如下：

```
(TID 1) in 27 ms on localhost (executor driver) (2/2)
21/04/08 15:53:34 INFO scheduler.DAGScheduler: ResultStage 0 (reduce at SparkPi.
scala:38) finished in 0.524 s
21/04/08 15:53:34 INFO scheduler.DAGScheduler: Job 0 finished: reduce at SparkPi
.scala:38, took 0.613876 s
Pi is roughly 3.1399556997784988
21/04/08 15:53:34 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose t
asks have all completed, from pool
21/04/08 15:53:34 INFO server.AbstractConnector: Stopped Spark@3153ddfc{HTTP/1.1
,[http/1.1]}{0.0.0.0:4040}
21/04/08 15:53:34 INFO ui.SparkUI: Stopped Spark web UI at http://10.24.21.217:4
040
```

图 5: 任务执行过程

我们可以直接在命令行看到词频统计的结果

```
scala> sc.textFile("file:///home/dase-local/spark-2.4.7/RELEASE").flatMap(_.spli
t(" ")).map((_,1)).reduceByKey(_+_).collect
res1: Array[(String, Int)] = Array((-Psparkr,1), (-B,1), (Spark,1), (-Pkubernete
s,1), (-Pyarn,1), (revision,1), (Build,1), (built,1), (-DzincPort=3038,1), (-Pfl
ume,1), ((git,1), (2.6.5,1), (flags:,1), (-Pmesos,1), (for,1), (-Pkafka-0-8,1),
(-Phadoop-provided,1), (14211a1),1), (2.4.7,1), (Hadoop,1))

scala>
```

图 6: 统计结果

运行中使用 jps 查看进程如下，可以看到出现了一个 SparkSubmit 进程，它会在程序运行结束后消失

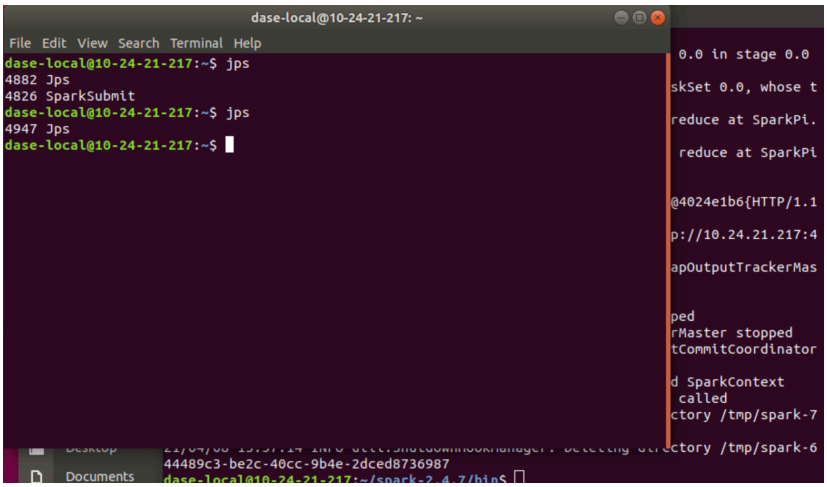


图 7: local 模式下运行过程中存在的进程

Section 2

单机伪分布式部署

在开始单机伪分布式部署前，我们先修改相应的配置文件

```
#
spark.eventLog.enabled=true
spark.eventLog.dir=hdfs://localhost:9000/tmp/spark_history
spark.history.fs.logDirectory=hdfs://localhost:9000/tmp/spark_history
#
```

图 8: 修改 spark-defaults.sh 文件

```
export SPARK_CONF_DIR="${SPARK_CONF_DIR:-"${SPARK_HOME}/conf"}"
# Add the PySpark classes to the PYTHONPATH:
if [ -z "${PYSPARK_PYTHONPATH_SET}" ]; then
  export PYTHONPATH="${SPARK_HOME}/python:${PYTHONPATH}"
  export PYTHONPATH="${SPARK_HOME}/python/lib/py4j-0.10.7-src.zip:${PYTHONPATH}"
  export PYSPARK_PYTHONPATH_SET=1
fi


export JAVA_HOME = /usr/local/jdk1.8
                                                          37,1          Bot
```

图 9: 修改 spark-config.sh 文件

tips: 这里最后是错误示范！sh 文件对空格敏感，务必把 JAVA_HOME = /usr/local/jdk1.8 等号两侧的空格删除

接着在 HDFS 中建立 Spark History 目录

```
dase-local@10-24-21-217:~/spark-2.4.7/bin$ ~/softwares/hadoop-2.10.1/bin/hdfs df
s -mkdir -p /tmp/spark_history
dase-local@10-24-21-217:~/spark-2.4.7/bin$
```

图 10: 在 HDFS 中建立 spark 历史目录

启动 Spark 服务，并且使用 jps 命令检查服务启动后的进程状态

```
dase-local@10-24-21-217:~/spark-2.4.7/sbin$ jps
9553 HistoryServer
9605 Jps
8710 DataNode
8505 NameNode
9273 Master
8958 SecondaryNameNode
9454 Worker
dase-local@10-24-21-217:~/spark-2.4.7/sbin$
```

图 11: 启动了 Spark 服务后的进程状态

我们使用 Web UI 来查看 Spark Master 的状况

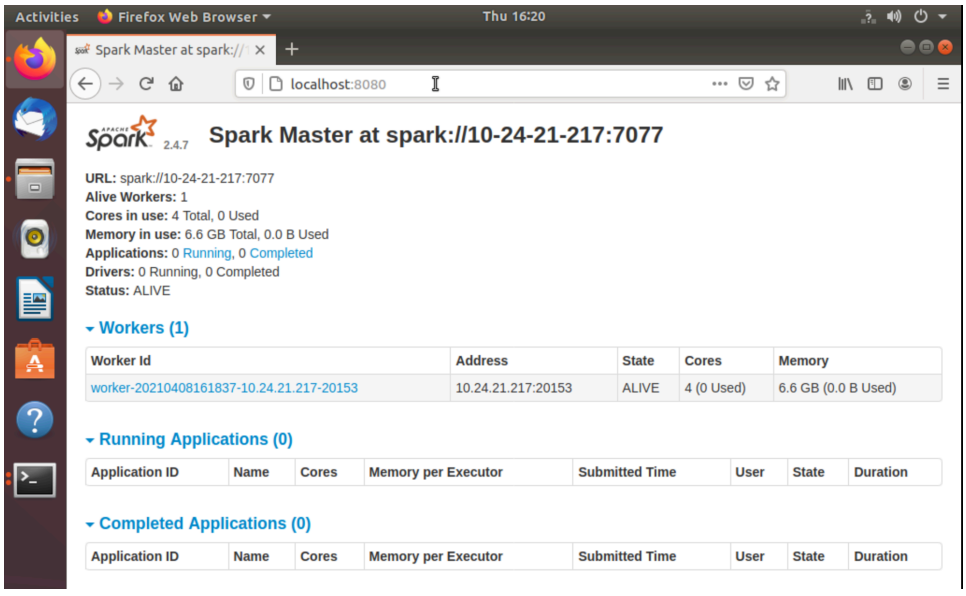


图 12: Spark Master

```
dase-local@10-24-21-217:~/spark-2.4.7/sbin$ ~/softwares/hadoop-2.10.1/bin/hdfs d
fs -mkdir -p spark_input
dase-local@10-24-21-217:~/spark-2.4.7/sbin$ ~/softwares/hadoop-2.10.1/bin/hdfs d
fs -put ~/spark-2.4.7/RELEASE spark_input/
dase-local@10-24-21-217:~/spark-2.4.7/sbin$
```

图 13: 准备输入文件

在 scala> 后输入 Scala 代码，统计 RELEASE 文件中的单词数量

```
File  Edit  View  Search  Terminal  Help
l(newLevel).
Spark context Web UI available at http://10.24.21.217:4040
Spark context available as 'sc' (master = spark://localhost:7077, app id = app-2
0210408165729-0000).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.7
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_171)
Type in expressions to have them evaluated.
Type :help for more information.

scala> sc.textFile("hdfs://localhost:9000/user/dase-local/spark_input/RELEASE").
flatMap(_.split(" ")).map((_,1)).reduceByKey(_+_).collect
res0: Array[(String, Int)] = Array((-Psparkr,1), (Build,1), (built,1), (-Pflume,
1), ((git,1), (-Pmesos,1), (-Phadoop-provided,1), (14211a1),1), (-B,1), (Spark,1
), (-Pkubernetes,1), (-Pyarn,1), (revision,1), (-DzincPort=3038,1), (2.6.5,1), (
flags:,1), (for,1), (-Pkafka-0-8,1), (2.4.7,1), (Hadoop,1))

scala>
```

图 14: 进入 Spark-Shell 并开始单词数量统计

通过提交 jar 包运行应用程序，计算 π 的大小

(Remark：如使用 localhost 无法正常启动的话可将 localhost 改为 127.0.1.1)

ClientClient 提交模式（默认）

此模式下 Driver 运行在客户端，可以在客户端看到应用程序运行过程中的信息。

```
cutors
21/04/08 19:07:03 INFO cluster.CoarseGrainedSchedulerBackend$DriverEndpoint: Ask
ing each executor to shut down
21/04/08 19:07:03 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMas
terEndpoint stopped!
21/04/08 19:07:03 INFO memory.MemoryStore: MemoryStore cleared
21/04/08 19:07:03 INFO storage.BlockManager: BlockManager stopped
21/04/08 19:07:03 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
21/04/08 19:07:03 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinator
Endpoint: OutputCommitCoordinator stopped!
21/04/08 19:07:03 INFO spark.SparkContext: Successfully stopped SparkContext
21/04/08 19:07:03 INFO util.ShutdownHookManager: Shutdown hook called
21/04/08 19:07:03 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-0
22cd3e8-6e99-457c-930f-7635733ffea0
21/04/08 19:07:03 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-e
aabee51-e298-456a-ade6-2e9bb49061e0
dase-local@10-24-21-217:~/spark-2.4.7/bin$
dase-local@10-24-21-217: ~
File  Edit  View  Search  Terminal  Help
6678 SparkSubmit
3274 Master
2988 SecondaryNameNode
6828 Jps
3548 HistoryServer
2526 NameNode
6783 CoarseGrainedExecutorBackend
dase-local@10-24-21-217:~$
```

图 15: Client 模式下运行过程中的进程状态

Cluster 提交模式

此模式下 Master 会随机选取一个 Worker 节点启动 Driver，故在客户端看不到应用程序运行过程中的信息。

```
dase-local@10-24-21-217:~$ jps
2736 DataNode
3461 Worker
7128 SparkSubmit
3274 Master
7274 CoarseGrainedExecutorBackend
2988 SecondaryNameNode
7196 DriverWrapper
7292 Jps
3548 HistoryServer
2526 NameNode
```

图 16: Cluster 模式下运行过程中存在的进程

在运行过程中另起一个终端执行 jps 查看进程，如图所示。在 Cluster 提交模式下，还可以看到一个 DriverWrapper 进程。

可以看到，二者存在一个 CoarseGrainedExecutorBackend 进程，负责创建及维护 Executor 对象。

tips: CoarseGrainedExecutorBackend 是什么？CoarseGrainedExecutorBackend 维护 Executor 对象，Executor 负责计算任务，即执行 task，CoarseGrainedExecutorBackend 在 spark 运行期是一个单独的进程，在 Worker 节点可以通过 Java 的 jps 命令查看

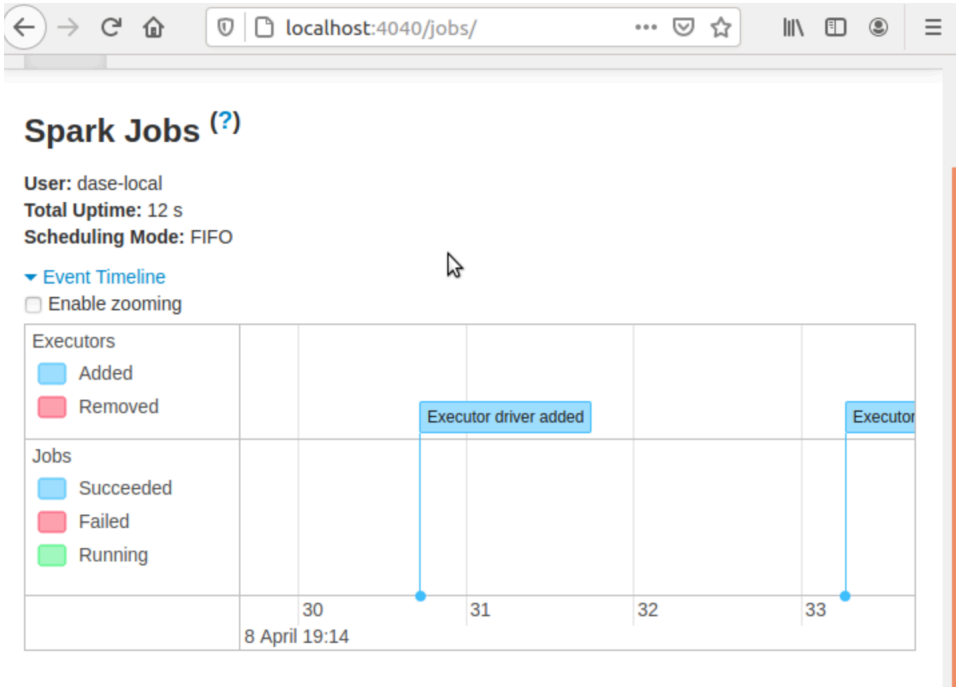
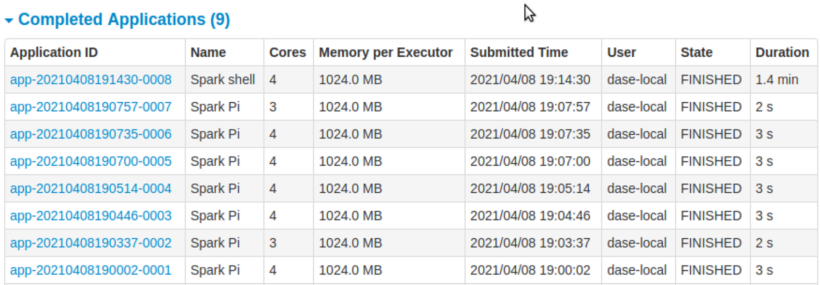


图 17: Spark 界面



Completed Applications (9)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20210408191430-0008	Spark shell	4	1024.0 MB	2021/04/08 19:14:30	dase-local	FINISHED	1.4 min
app-20210408190757-0007	Spark Pi	3	1024.0 MB	2021/04/08 19:07:57	dase-local	FINISHED	2 s
app-20210408190735-0006	Spark Pi	4	1024.0 MB	2021/04/08 19:07:35	dase-local	FINISHED	3 s
app-20210408190700-0005	Spark Pi	4	1024.0 MB	2021/04/08 19:07:00	dase-local	FINISHED	3 s
app-20210408190514-0004	Spark Pi	4	1024.0 MB	2021/04/08 19:05:14	dase-local	FINISHED	3 s
app-20210408190446-0003	Spark Pi	4	1024.0 MB	2021/04/08 19:04:46	dase-local	FINISHED	3 s
app-20210408190337-0002	Spark Pi	3	1024.0 MB	2021/04/08 19:03:37	dase-local	FINISHED	2 s
app-20210408190002-0001	Spark Pi	4	1024.0 MB	2021/04/08 19:00:02	dase-local	FINISHED	3 s

图 18: Spark 的 Application 界面

通过 Web UI 查看 Spark 程序运行信息

1. 实时查看应用运行情况在应用运行过程中（如进入 Spark-Shell 之后），访问 http://localhost:4040. Web 界面如图所示。

2. 查看 Spark 应用程序日志在提交一个应用程序后，在 /spark-2.4.7/work 下会出现应用程序运行日志文件夹。访问 http://localhost:8080, 点击 Completed Applications 下的对应应用程序的 Application ID

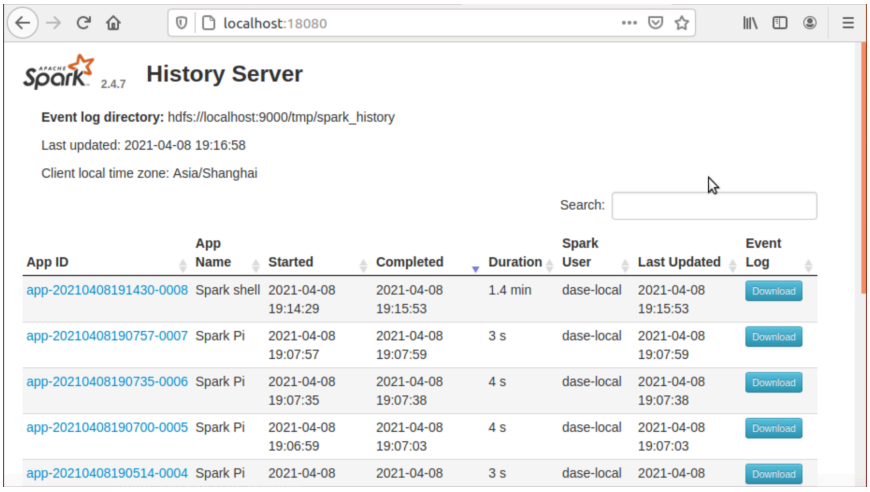


图 19: Spark History 界面

最后我们停止 Spark 服务，如下

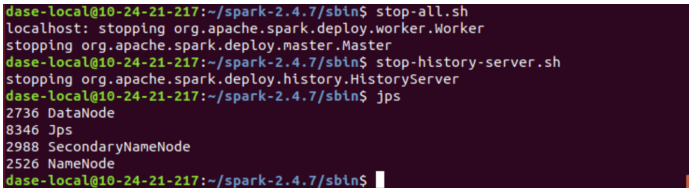


图 20: 停止 Spark 和日志服务器

记得还要在实验结束后把 HDFS 也关闭，否则影响分布式部署。

Section 3
分布式部署

接下来开始分布式部署基于 Spark 的各项任务，同样的，是由一个主节点，一个客户端（本机）和两个从节点构成。主节点负责接受任务后进行处理，从节点辅助其进行计算，客户端提交任务。

在实验开始前，还是需要四台机器设置免密登陆。

(1) 准备工作

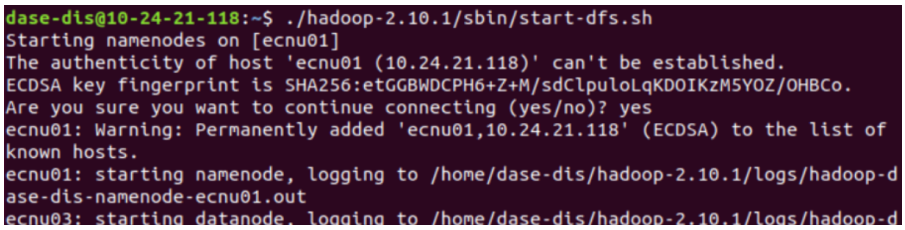


图 21: 准备工作: 开启 HDFS

```
dase-dis@10-24-21-118:~$ jps
4208 SecondaryNameNode
4486 Jps
3928 NameNode
```

图 22: 查看启动 HDFS 后的进程状态

(2) 修改配置

主节点上修改：spark-env.sh,slaves,spark-defaults.conf,spark-config.sh 文件

Remark：我们组的主节点修改 slaves 文件时，发现其中有内容 localhost，在此基础上追加 ecnu02，ecnu03。导致的结果是后期启动服务时主节点也同样充当了 worker，但并不影响实验的结果和完整性，详情见后续截图。

在主节点修改过配置文件后，将配置好的 Spark 拷贝到其他节点：

```
dase-dis@10-24-21-118:~$ scp  -r ./spark-2.4.7/ dase-dis@ecnu02:~/
LICENSE                                       100%   21KB  34.0MB/s   00:00
spark-env.sh                                  100% 4456    27.2MB/s   00:00
docker.properties.template                    100%  996     7.7MB/s   00:00
spark-defaults.conf.template                  100% 1292    10.3MB/s   00:00
spark-defaults.conf                           100% 1447    11.7MB/s   00:00
fairscheduler.xml.template                    100% 1105     6.9MB/s   00:00
slaves                                        100%  880     7.3MB/s   00:00
log4j.properties.template                     100% 2025    13.2MB/s   00:00
metrics.properties.template                   100% 7801    42.9MB/s   00:00
```

图 23: 主节点分发 Spark

(3) 启动服务

在主节点上执行启动命令，启动 Spark 和历史信息服务器。

```
dase-dis@10-24-21-118:~$ ./spark-2.4.7/sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /home/dase-dis/spark-
2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.master.Master-1-ecnu01.out
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known h
osts.
ecnu02: starting org.apache.spark.deploy.worker.Worker, logging to /home/dase-di
s/spark-2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.worker.Worker-1-ecnu02
.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /home/dase
-dis/spark-2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.worker.Worker-1-ecn
u01.out
ecnu03: starting org.apache.spark.deploy.worker.Worker, logging to /home/dase-di
s/spark-2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.worker.Worker-1-ecnu03
.out
```

图 24: 启动 Spark

启动 Spark 历史信息服务器

```
dase-dis@10-24-21-118:~$ ./spark-2.4.7/sbin/start-history-server.sh
starting org.apache.spark.deploy.history.HistoryServer, logging to /home/dase-di
s/spark-2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.history.HistoryServer-
1-ecnu01.out
```

图 25: 开启 Spark 历史信息服务器

(4) 查看 Spark 服务信息

使用 jps 查看进程，验证是否成功启动服务

在主节点查看信息，可见由于 **slaves 文件中存在 localhost**，主节点也启动了 worker

```
dase-dis@10-24-21-118:~$ jps
4208 SecondaryNameNode
4980 HistoryServer
3928 NameNode
4713 Worker
5034 Jps
4526 Master
```

图 26: 主节点进程状态

从节点进程信息如下

```
dase-dis@ecnu02:~$ jps
4005 Worker
4135 Jps
3623 DataNode
```

图 27: 从节点 ecnu02 进程状态

```
dase-dis@ecnu03:~$ jps
3782 Worker
3942 Jps
3465 DataNode
dase-dis@ecnu03:~$
```

图 28: 从节点 ecnu03 进程状态

在主节点查看 master 日志

```
dase-dis@10-24-21-118:~$ tail ./spark-2.4.7/logs/spark-dase-dis-org.apache.spark
.deploy.master.Master-1-ecnu01.out
21/04/12 16:26:04 INFO master.Master: Launching executor app-20210412162604-0000
/0 on worker worker-20210412162525-10.24.21.220-17343
21/04/12 16:26:04 INFO master.Master: Launching executor app-20210412162604-0000
/1 on worker worker-20210412162525-10.24.21.118-28049
21/04/12 16:26:04 INFO master.Master: Launching executor app-20210412162604-0000
/2 on worker worker-20210412162525-10.24.21.47-13083
21/04/12 16:26:05 INFO master.Master: Received unregister request from applicati
on app-20210412162604-0000
21/04/12 16:26:05 INFO master.Master: Removing app app-20210412162604-0000
21/04/12 16:26:05 INFO master.Master: 10.23.79.29:25742 got disassociated, remov
ing it.
21/04/12 16:26:05 INFO master.Master: ecnu04:36593 got disassociated, removing i
t.
21/04/12 16:26:05 WARN master.Master: Got status update for unknown executor app
-20210412162604-0000/2
21/04/12 16:26:05 WARN master.Master: Got status update for unknown executor app
-20210412162604-0000/1
21/04/12 16:26:05 WARN master.Master: Got status update for unknown executor app
-20210412162604-0000/0
```

图 29: Master 日志

再查看 worker 日志

```
dase-dis@10-24-21-118:~$ head ./spark-2.4.7/logs/spark-dase-dis-org.apache.spark
.deploy.worker.Worker-1-ecnu01.out
Spark Command: /usr/local/jdk1.8/bin/java -cp /home/dase-dis/spark-2.4.7/conf/:/
home/dase-dis/spark-2.4.7/jars/*:/home/dase-dis/hadoop-2.10.1/etc/hadoop/:/home/
dase-dis/hadoop-2.10.1/share/hadoop/common/lib/*:/home/dase-dis/hadoop-2.10.1/sh
are/hadoop/common/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/hdfs/:/home/dase-d
is/hadoop-2.10.1/share/hadoop/hdfs/lib/*:/home/dase-dis/hadoop-2.10.1/share/hado
op/hdfs/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/yarn/:/home/dase-dis/hadoop-
2.10.1/share/hadoop/yarn/lib/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/yarn/*:
/home/dase-dis/hadoop-2.10.1/share/hadoop/mapreduce/lib/*:/home/dase-dis/hadoop-
2.10.1/share/hadoop/mapreduce/*:/home/dase-dis/hadoop-2.10.1/contrib/capacity-sc
heduler/*.jar -Xmx1g org.apache.spark.deploy.worker.Worker --webui-port 8081 spa
rk://ecnu01:7077
========================================
21/04/12 16:25:24 INFO worker.Worker: Started daemon with process name: 4713@ecn
u01
21/04/12 16:25:24 INFO util.SignalUtils: Registered signal handler for TERM
21/04/12 16:25:24 INFO util.SignalUtils: Registered signal handler for HUP
21/04/12 16:25:24 INFO util.SignalUtils: Registered signal handler for INT
21/04/12 16:25:24 INFO spark.SecurityManager: Changing view acls to: dase-dis
21/04/12 16:25:24 INFO spark.SecurityManager: Changing modify acls to: dase-dis
21/04/12 16:25:24 INFO spark.SecurityManager: Changing view acls groups to:
21/04/12 16:25:24 INFO spark.SecurityManager: Changing modify acls groups to:
```

图 30: Worker 日志

最后查看 historyserver 日志

```
dase-dis@10-24-21-118:~$ head ./spark-2.4.7/logs/spark-dase-dis-org.apache.spark
.deploy.history.HistoryServer-1-ecnu01.out
Spark Command: /usr/local/jdk1.8/bin/java -cp /home/dase-dis/spark-2.4.7/conf/:/
home/dase-dis/spark-2.4.7/jars/*:/home/dase-dis/hadoop-2.10.1/etc/hadoop/:/home/
dase-dis/hadoop-2.10.1/share/hadoop/common/lib/*:/home/dase-dis/hadoop-2.10.1/sh
are/hadoop/common/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/hdfs/:/home/dase-d
is/hadoop-2.10.1/share/hadoop/hdfs/lib/*:/home/dase-dis/hadoop-2.10.1/share/hado
op/hdfs/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/yarn/:/home/dase-dis/hadoop-
2.10.1/share/hadoop/yarn/lib/*:/home/dase-dis/hadoop-2.10.1/share/hadoop/yarn/*:
/home/dase-dis/hadoop-2.10.1/share/hadoop/mapreduce/lib/*:/home/dase-dis/hadoop-
2.10.1/share/hadoop/mapreduce/*:/home/dase-dis/hadoop-2.10.1/contrib/capacity-sc
heduler/*.jar -Xmx1g org.apache.spark.deploy.history.HistoryServer
========================================
21/04/12 16:30:57 INFO history.HistoryServer: Started daemon with process name:
4980@ecnu01
21/04/12 16:30:57 INFO util.SignalUtils: Registered signal handler for TERM
21/04/12 16:30:57 INFO util.SignalUtils: Registered signal handler for HUP
21/04/12 16:30:57 INFO util.SignalUtils: Registered signal handler for INT
21/04/12 16:30:57 INFO spark.SecurityManager: Changing view acls to: dase-dis
21/04/12 16:30:57 INFO spark.SecurityManager: Changing modify acls to: dase-dis
21/04/12 16:30:57 INFO spark.SecurityManager: Changing view acls groups to:
21/04/12 16:30:57 INFO spark.SecurityManager: Changing modify acls groups to:
```

图 31: HistoryServer 日志

通过 Spark Web UI 界面，可以看到三个 worker 和正在运行的应用信息和已完成的应用的历史信息

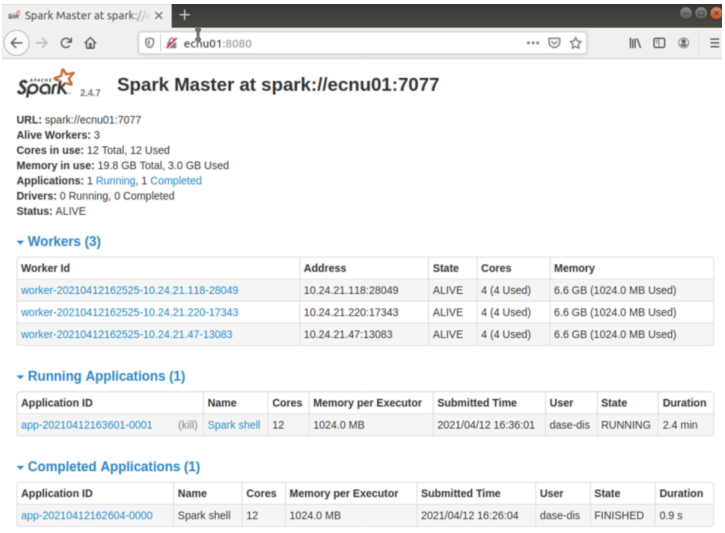


图 32: Web UI 查看主节点信息

(5) 运行 Spark 应用程序

客户端进入 Spark-Shell，开始实验

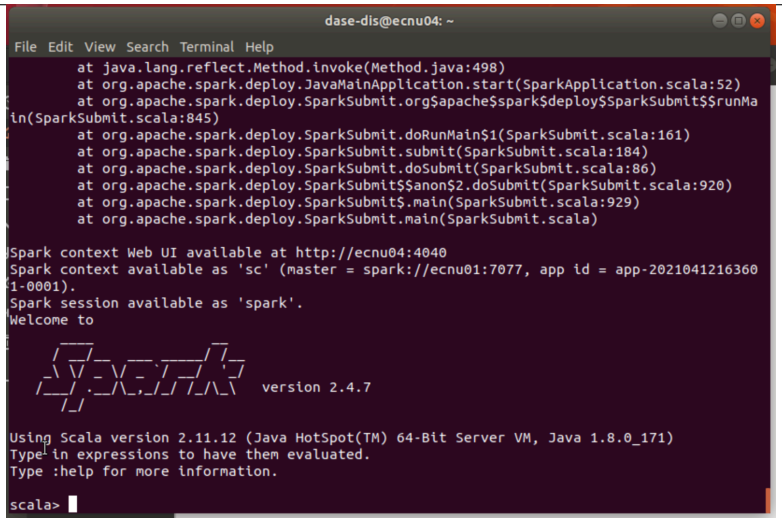


图 33: Spark-Shell

在 *scala>* 后输入 Scala 代码，用于统计 RELEASE 文件的单词数量

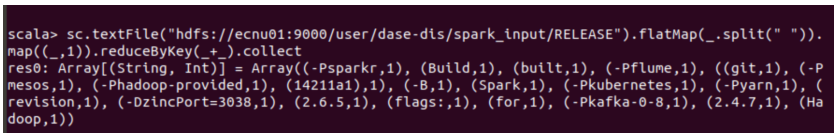


图 34: 统计 RELEASE 单词数量

通过提交 jar 包运行应用程序首先是在 Client 提交模式下执行:

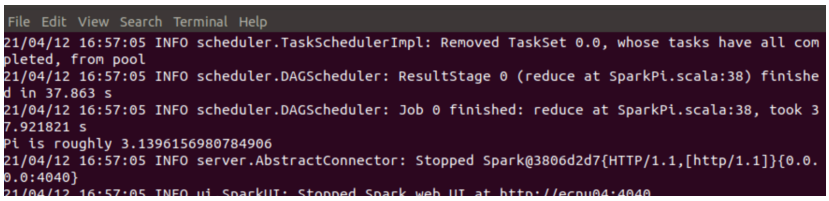


图 35: 计算 π 任务执行中

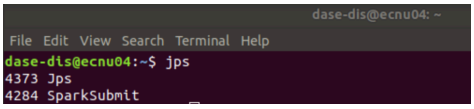


图 36: 客户端进程状态

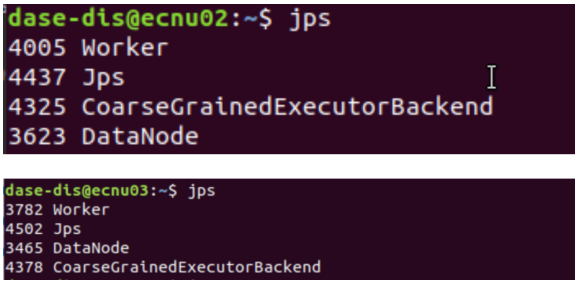


图 37: 两个从节点进程状态

接下来是在 Cluster 提交模式，这种模式不会在命令行工具中看到显式的执行结果

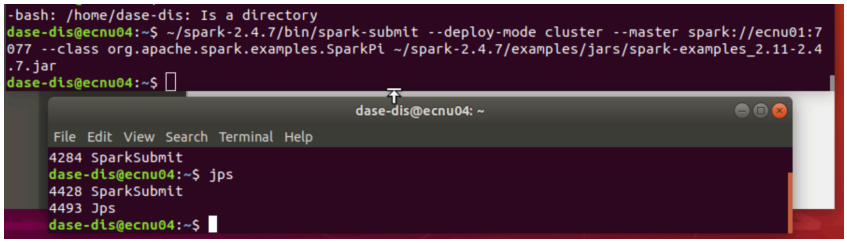


图 38: Cluster 模式下计算 π 并查看客户端进程状态

观察从节点状态，可以看到一个从节点被 Spark 选为了 Driver，这是集群模式下非常明显的一个特征

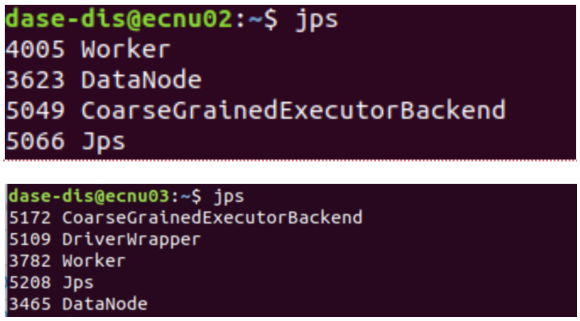


图 39: 观察从节点状态，**注意 DriverWrapper**

(6) 查看 Spark 程序运行信息

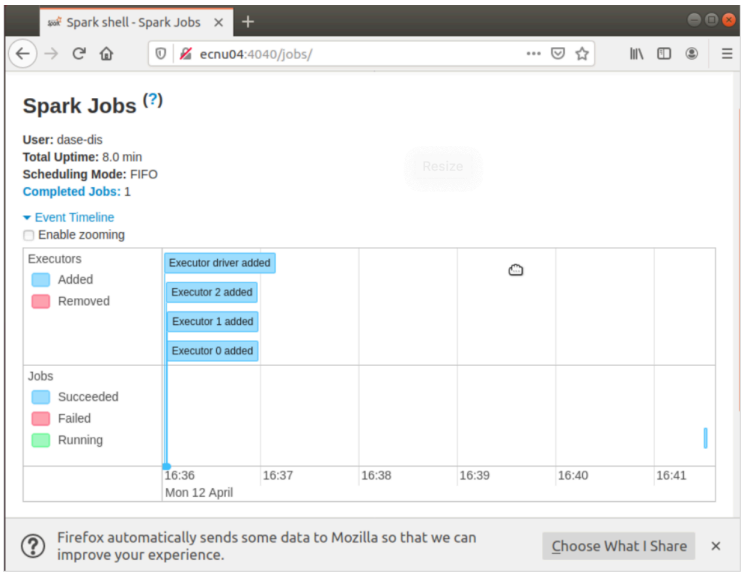


图 40: Spark Jobs

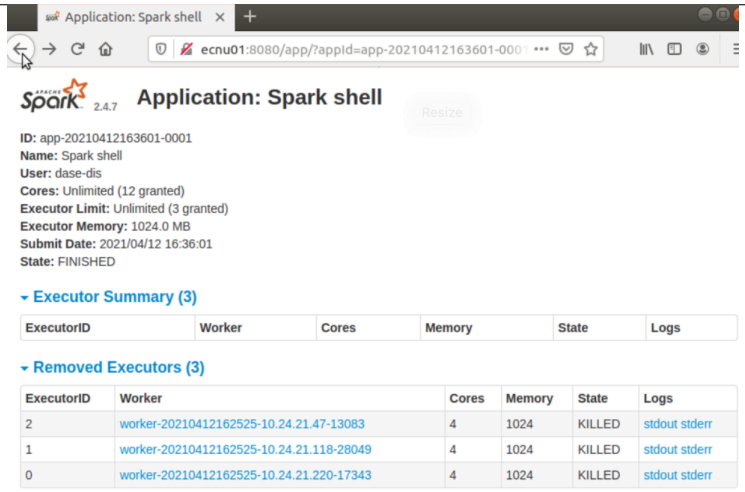


图 41: Spark Shell Web UI

(注意，这里必须在 Spark Shell 运行的情况下才可以查看)

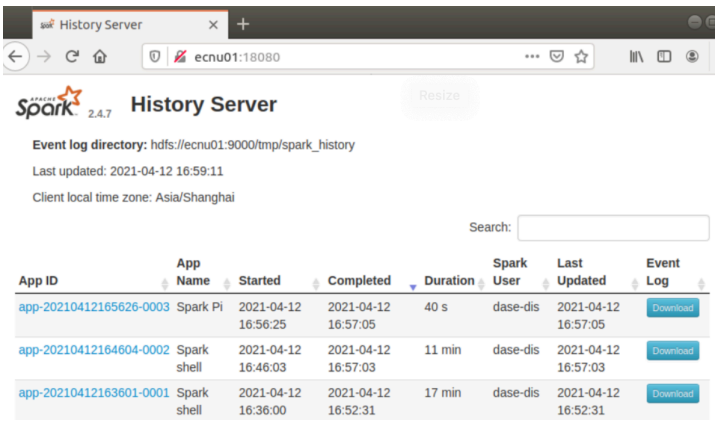


图 42: 查看 History Server

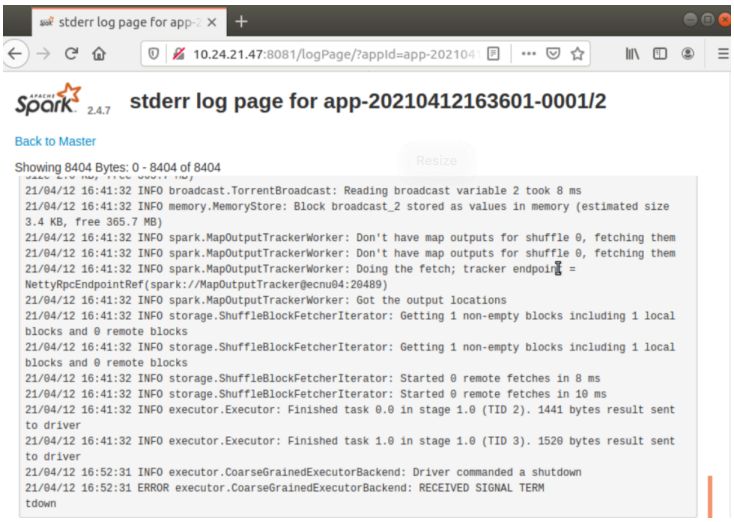


图 43: stderr-log

(7) 停止服务

关闭 Spark 及其相关服务

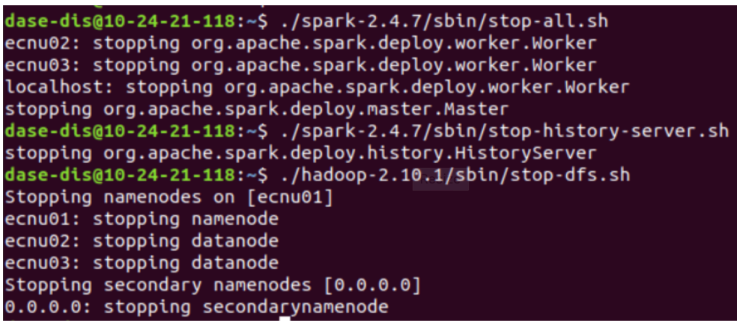


图 44: 关闭 Spark 服务和 dfs

检查主节点状态，不再有任何与 Spark 相关的进程了。

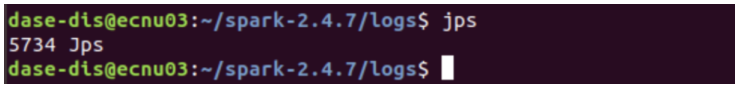


图 45: 结束后从节点状态

Part 5

思考题

Section 1

对于分布式部署的 spark 集群，如果在集群启动过程中某个从节点的 Worker 节点没有启动成功，那么应该在哪个节点上查看哪个路径下的什么日志?

我们主动制造一个配置 Worker 节点时的错误，为了模拟从节点未能启动成功的情形，我们对一个从节点的配置文件进行修改，将其 HADOOP_HOME 变量故意修改为一个错误值，随后通过以下方式查看

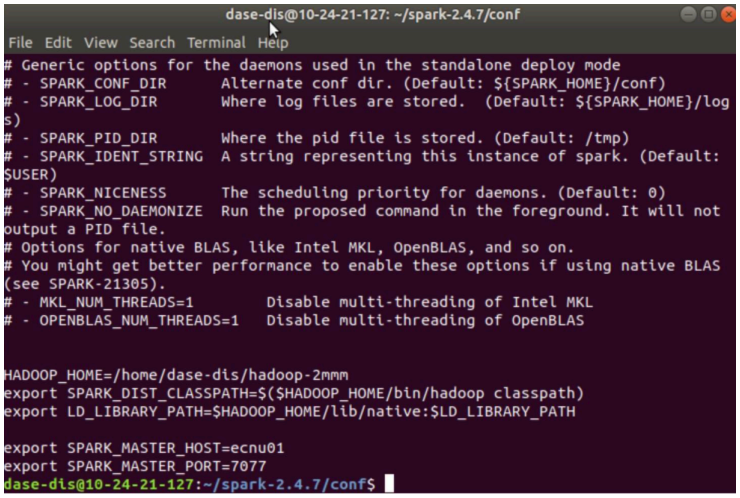


图 46: 思考题 1-制造配置错误

随后我们可以在在从节点的 logs 目录里查看日志

主节点启动时的报错信息如下，错误原因的确为修改了 HADOOP_HOME 之后运行时找不到 Hadoop 的目录

图 47: 思考题 1- 报错信息

主节点启动时的报错信息如下，可以看到，错误原因的确为修改了 HADOOP_HOME 之后运行时无法正确找到 Hadoop 的目录

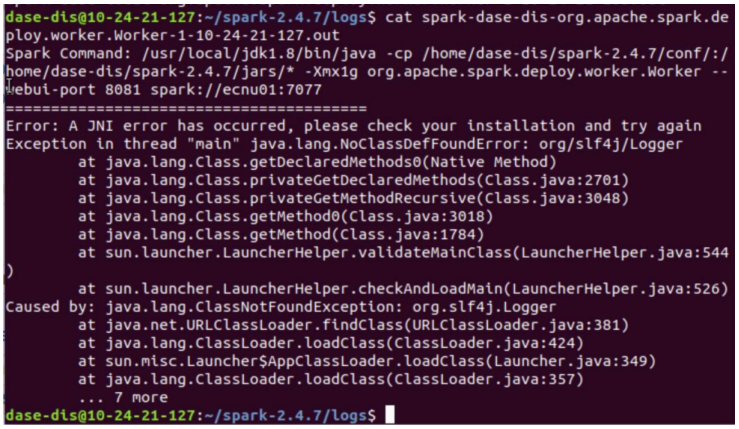


图 48: 思考题 1-主节点日志

我们也可以看到从节点同样出现了报错。对比在主节点上的报错信息，可以发现抛出的异常是一样的，如下图所示

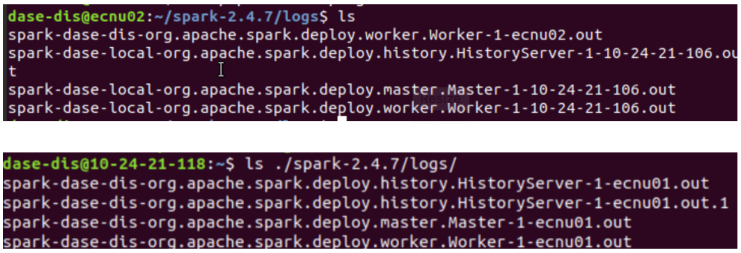


图 49: 思考题 1-从节点日志

```
ecnu03: /home/dase-dis/spark-2.4.7/conf/spark-env.sh: line 73: /home/dase-dis/ha
doop-2mmm/bin/hadoop: No such file or directory
ecnu03: starting org.apache.spark.deploy.worker.Worker, logging to /home/dase-di
s/spark-2.4.7/logs/spark-dase-dis-org.apache.spark.deploy.worker.Worker-1-10-24-
21-127.out
ecnu03: failed to launch: nice -n 0 /home/dase-dis/spark-2.4.7/bin/spark-class o
rg.apache.spark.deploy.worker.Worker --webui-port 8081 spark://ecnu01:7077
ecnu03:         at java.lang.Class.getMethod0(Class.java:3018)
ecnu03:         at java.lang.Class.getMethod(Class.java:1784)
ecnu03:         at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.
java:544)
ecnu03:         at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.j
ava:526)
ecnu03:   Caused by: java.lang.ClassNotFoundException: org.slf4j.Logger
ecnu03:         at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
ecnu03:         at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
ecnu03:         at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)
ecnu03:         at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
ecnu03:         ... 7 more
ecnu03: full log in /home/dase-dis/spark-2.4.7/logs/spark-dase-dis-org.apache.sp
ark.deploy.worker.Worker-1-10-24-21-127.out
dase-dis@ecnu01:~$
```

图 50: 思考题 1-从节点日志 Cont'd

Section 2

既然采用 Cluster 提交模式在客户端看不到程序运行过程中的信息，那么这些运行过程中的信息如何查看？（提示：结合 HistoryServer 的 Web UI 进行阐述）

可以在 HistoryServer 的 Web UI 中进行查看，运行过程中的信息通过 stderr 流进行输出，因此我们查看 stderr，若要知道运行结果，应该查看 stdout

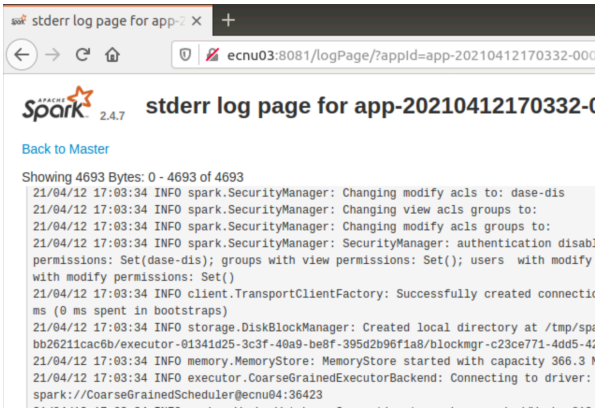


图 51: 思考题 2 Web UI-stderr