

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：基于 Yarn 部署 Flink

学号：10185501402

上机实践日期：2021/5/20

上机实践编号：Lab15

组号：Group5

上机实践时间：

Part 1

实验目的

- (1) 学习在 Yarn 平台上部署 Flink，理解与 Standalone 部署 Flink 的区别
 - (2) 通过在 Yarn 平台上部署的 Flink 上运行程序，体会与 Standalone 部署的 Flink 上运行程序的区别
-

Part 2

实验任务

- (1) 完成 Flink on Yarn 的单机伪分布式部署以及分布式部署。
 - (2) 两种部署方式下启动一个长时间运行的 Flink Session，并向其中提交示例程序。
 - (3) 两种部署方式下在 Yarn 上运行一个单独的 Flink 作业。
 - (4) 额外的实验任务：完成基于 Yarn 同时部署 MapReduce, Spark, Flink 应用
-

Part 3

使用环境

- (1) 操作系统：Ubuntu 18.04
 - (2) JDK 版本：1.8
 - (3) Flink 版本：1.12.1
 - (4) Spark 版本：2.4.7（同时部署 MapReduce,Spark,Flink 应用时使用）
-

Part 4

实验过程

Section 1

单机伪分布式部署

华东师范大学数据科学与工程学院学生实验报告

首先进行单机伪分布式部署，与之前的实验一样先开启 Hadoop，Yarn 以及 Historyserver 三项服务。

```
dase-local@10-23-45-66:~$ ./softwares/hadoop-2.10.1/sbin/start-dfs.sh
Starting namenodes on [localhost]
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:7Izvf1jUbQ/PwIv1HUsCpWTUNX9GJx0tZD9vejtFo.
Are you sure you want to continue connecting (yes/no)? yes
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: starting namenode, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/hadoop-dase-local-namenode-10-23-45-66.out
localhost: starting datanode, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/hadoop-dase-local-datanode-10-23-45-66.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/hadoop-dase-local-secondarynamenode-10-23-45-66.out
dase-local@10-23-45-66:~$ ./softwares/hadoop-2.10.1/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/yarn-dase-local-resourcemanager-10-23-45-66.out
localhost: starting nodemanager, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/yarn-dase-local-nodemanager-10-23-45-66.out
dase-local@10-23-45-66:~$ ./softwares/hadoop-2.10.1/sbin/mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/dase-local/softwares/hadoop-2.10.1/logs/mapred-dase-local-historyserver-10-23-45-66.out
dase-local@10-23-45-66:~$
```

图 1: 开启 Hadoop，Yarn 以及 Historyserver

随后检查各进程是否正常

```
dase-local@10-23-45-66:~$ jps
5072 ResourceManager
4448 NameNode
5250 NodeManager
5683 Jps
5609 JobHistoryServer
4907 SecondaryNameNode
4652 DataNode
dase-local@10-23-45-66:~$
```

图 2: jps 检查各进程是否正常

提交一个词频统计任务，并通过 Netcat 输入待统计的文本

```
2048 MB to make up an integer multiple of its minimum allocation memory (1024 MB
, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 448 MB may n
ot be used by Flink.
2021-05-27 15:31:03,415 INFO org.apache.flink.yarn.YarnClusterDescriptor
[] - The configured TaskManager memory is 1728 MB. YARN will allocate
2048 MB to make up an integer multiple of its minimum allocation memory (1024 M
B, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 320 MB may
not be used by Flink.
2021-05-27 15:31:03,415 INFO org.apache.flink.yarn.YarnClusterDescriptor
[] - Cluster specification: ClusterSpecification{masterMemoryMB=1600,
taskManagerMemoryMB=1728, slotsPerTaskManager=2}
2021-05-27 15:31:09,198 INFO org.apache.flink.yarn.YarnClusterDescriptor
[] - Submitting application master application_1622100209815_0001
2021-05-27 15:31:09,448 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientI
mpl
[] - Submitted application application_1622100209815_0001
2021-05-27 15:31:09,449 INFO org.apache.flink.yarn.YarnclusterDescriptor
[] - Waiting for the cluster to be allocated
2021-05-27 15:31:09,455 INFO org.apache.flink.yarn.YarnclusterDescriptor
[] - Deploying cluster, current state ACCEPTED
2021-05-27 15:31:17,125 INFO org.apache.flink.yarn.YarnclusterDescriptor
[] - YARN application has been deployed successfully.
2021-05-27 15:31:17,126 INFO org.apache.flink.yarn.YarnclusterDescriptor
[] - Found Web Interface 10-23-74-82:35221 of application 'application_1622100209815_0001'.
Job has been submitted with JobID 332f1b3cc30a94519a8130ebbe2a182c
```

图 3: 提交一个词频统计任务

随后访问本地的 WebUI 查看当前正在运行的 Job

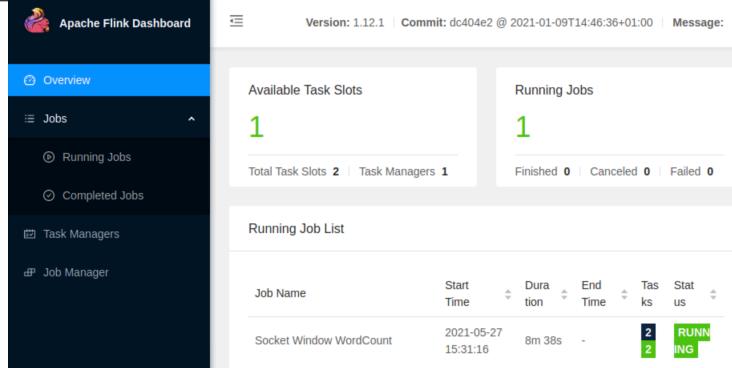


图 4: WebUI 查看进程状态

最后通过日志文件查看输出结果，
.out 中，而非是之前的 flink-1.12.1/log 路径下

```
dase-local@10-23-45-66:~/softwares/hadoop-2.10.1/logs/userlogs/application_1622100209815_0001/container_1622100209815_0001_01_0000025 cat taskmanager.out
00209815_0001/container_1622100209815_0001_01_0000025 cat taskmanager.out
hello : 1
flink : 1
no : 1
me : 1
than : 1
better : 1
flink : 1
loves : 1
one : 1
flink : 1
cool : 1
quite : 1
is : 1
```

图 5: 查看词频统计结果

Section 2

分布式部署

现在开始分布式部署，和往常一样是四台机器，先做一下文件分发并开启服务，流程如图 6，图 7，图 8，图 9 所示。

```
dase-dis@ecnu01:~$ scp .bashrc dase-dis@ecnu02:~
.bashrc                                         100% 3918      7.9MB/s   00:00
dase-dis@ecnu01:~$ scp .bashrc dase-dis@ecnu03:~
.bashrc                                         100% 3918      10.4MB/s   00:00
dase-dis@ecnu01:~$ scp .bashrc dase-dis@ecnu04:~
.bashrc                                         100% 3918      11.1MB/s   00:00
```

图 6: 主节点分发文件

```
dase-dis@ecnu01:~$ ./hadoop-2.10.1/sbin/start-dfs.sh
Starting namenodes on [ecnu01]
ecnu01: starting namenode, logging to /home/dase-dis/hadoop-2.10.1/logs/hadoop-dase-dis-namenode-ecnu01.out
ecnu02: starting datanode, logging to /home/dase-dis/hadoop-2.10.1/logs/hadoop-dase-dis-datanode-10-24-21-148.out
ecnu03: starting datanode, logging to /home/dase-dis/hadoop-2.10.1/logs/hadoop-dase-dis-datanode-ecnu03.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/dase-dis/hadoop-2.10.1/logs/hadoop-dase-dis-secondarynamenode-ecnu01.out
```

图 7: 主节点启动 HDFS

```
dase-dis@ecnu01:~$ ./hadoop-2.10.1/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/dase-dis/hadoop-2.10.1/logs/yarn-dase-dis-resourcemanager-ecnu01.out
ecnu02: starting nodemanager, logging to /home/dase-dis/hadoop-2.10.1/logs/yarn-dase-dis-nodemanager-10-24-21-148.out
ecnu03: starting nodemanager, logging to /home/dase-dis/hadoop-2.10.1/logs/yarn-dase-dis-nodemanager-ecnu03.out
localhost: starting nodemanager, logging to /home/dase-dis/hadoop-2.10.1/logs/yarn-dase-dis-nodemanager-ecnu01.out
```

图 8: 主节点启动 Yarn

华东师范大学数据科学与工程学院学生实验报告

```
dase-dis@ecnu01:~$ ./hadoop-2.10.1/sbin/mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/dase-dis/hadoop-2.10.1/logs/mapred-dase-dis-historyserver-ecnu01.out
```

图 9: 主节点启动 HistoryServer

随后主节点检查进程状态，观察是否都成功启动。

```
dase-dis@ecnu01:~$ jps
13537 SecondaryNameNode
14021 JobHistoryServer
13255 NameNode
14074 Jps
13723 ResourceManager
```

图 10: 主节点: 使用 jps 检查进程状态

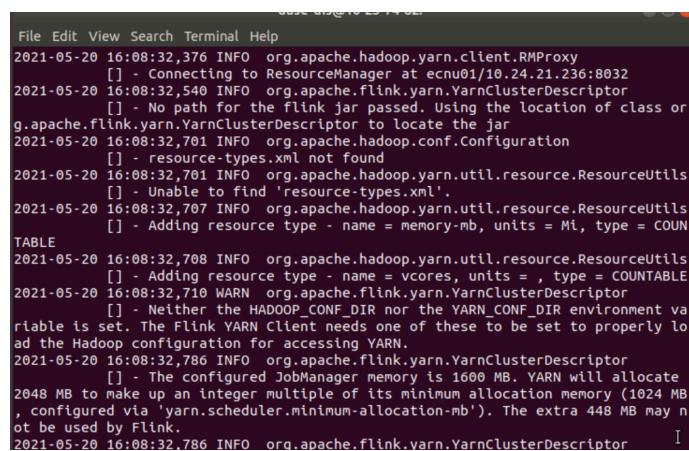
```
[dase-dis@10-24-21-148:~$ jps
3649 Jps
3494 NodeManager
3304 DataNode
```

图 11: 从节点 1: 使用 jps 检查进程状态

```
dase-dis@10-23-136-82:~$ jps
6306 DataNode
6499 NodeManager
6708 Jps
```

图 12: 从节点 2: 使用 jps 检查进程状态

客户端提交 Flink 任务，启动服务



The screenshot shows a terminal window with a black background and white text. It displays the command-line interface for submitting a Flink job to a YARN cluster. The logs include information about connecting to the ResourceManager, finding the YarnClusterDescriptor, and configuring resource types. It also shows a warning about the HADOOP_CONF_DIR environment variable and the configuration of JobManager memory.

```
File Edit View Search Terminal Help
2021-05-20 16:08:32,376 INFO org.apache.hadoop.yarn.client.RMProxy
    [] - Connecting to ResourceManager at ecnu01/10.24.21.236:8032
2021-05-20 16:08:32,540 INFO org.apache.flink.yarn.YarnClusterDescriptor
    [] - No path for the flink jar passed. Using the location of class or
g.apache.flink.yarn.YarnClusterDescriptor to locate the jar
2021-05-20 16:08:32,701 INFO org.apache.hadoop.conf.Configuration
    [] - resource-types.xml not found
2021-05-20 16:08:32,701 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils
    [] - Unable to find 'resource-types.xml'.
2021-05-20 16:08:32,707 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils
    [] - Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
TABLE
2021-05-20 16:08:32,708 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils
    [] - Adding resource type - name = vcores, units = , type = COUNTABLE
2021-05-20 16:08:32,710 WARN org.apache.flink.yarn.YarnClusterDescriptor
    [] - Neither the HADOOP_CONF_DIR nor the YARN_CONF_DIR environment va
riable is set. The Flink YARN Client needs one of these to be set to properly lo
ad the Hadoop configuration for accessing YARN.
2021-05-20 16:08:32,786 INFO org.apache.flink.yarn.YarnClusterDescriptor
    [] - The configured JobManager memory is 1600 MB. YARN will allocate
2048 MB to make up an integer multiple of its minimum allocation memory (1024 MB
, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 448 MB may n
ot be used by Flink.
2021-05-20 16:08:32,786 INFO org.apache.flink.yarn.YarnClusterDescriptor
```

图 13: 客户端: 提交任务

若出现安全模式异常，需要主节点关闭 Safe Mode

```
dase-dis@ecnu01:~$ ./hadoop-2.10.1/bin/dfsadmin -safemode leave
Safe mode is OFF
```

图 14: Safe Mode

随后输入文本用于词频统计

```
dase-dis@10-23-74-82:~$  
dase-dis@10-23-74-82:~$ nc -lk 8888  
i love wengsiyang  
i love dase
```

图 15: 客户端: 输入文本

运行任务后, 主节点处可以查看日志

```
dase-dis@ecnu01:~$ head ./hadoop-2.10.1/logs/yarn-dase-dis-resourcemanager-ecnu01.log  
2021-04-29 16:07:24,887 INFO org.apache.hadoop.yarn.server.resourcemanager.ResourceManager: STARTUP_MSG:  
*****  
STARTUP_MSG: Starting ResourceManager  
STARTUP_MSG: host = ecnu01/10.24.21.95  
STARTUP_MSG: args = []  
STARTUP_MSG: version = 2.10.1  
STARTUP_MSG: classpath = /home/dase-dis/hadoop-2.10.1/etc/hadoop:/home/dase-dis/hadoop-2.10.1/etc/hadoop  
p-2.10.1/etc/hadoop:/home/dase-dis/hadoop-2.10.1/share/hadoop/common/lib/jackson-xc-1.9.13.jar:/home/dase-  
re/hadoop/common/lib/jsr305-3.0.2.jar:/home/dase-dis/hadoop-2.10.1/share/hadoop/common/lib/slf4j-api-1.7.2  
hadoop-2.10.1/share/hadoop/common/lib/avro-1.7.7.jar:/home/dase-dis/hadoop-2.10.1/share/hadoop/common/lib/
```

图 16: 主节点: 查看日志

```
dase-dis@ecnu01:~$ head ./hadoop-2.10.1/logs/yarn-dase-dis-resourcemanager-ecnu01.out  
May 20, 2021 3:56:16 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register  
INFO: Registering org.apache.hadoop.yarn.server.resourcemanager.webapp.JAXBContextResolver as a provider class  
May 20, 2021 3:56:16 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register  
INFO: Registering org.apache.hadoop.yarn.server.resourcemanager.webapp.RMWebServices as a root resource class  
May 20, 2021 3:56:16 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register  
INFO: Registering org.apache.hadoop.yarn.webapp.GenericExceptionHandler as a provider class  
May 20, 2021 3:56:16 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate  
INFO: Initiating Jersey application, version 'Jersey: 1.9 09/02/2011 11:17 AM'  
May 20, 2021 3:56:16 PM com.sun.jersey.spi.container.GuiceComponentProviderFactory getComponentProvider
```

图 17: 主节点: 查看日志

随后查看 Web UI, 因为 Yarn 是基于 Hadoop 的资源管理系统, 显然可以先通过 Hadoop 的 Web UI 来查看程序的运行状况, 再使用 Flink UI 查看

ID	User	Name	Application Type	Queue	Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU vCores	Allocated Memory MB	Allocated GPUs

图 18: 查看 Hadoop UI

ID	Sent	Attempts	Host	Start T	Status	More
0	1		ecnu02	2021-05-20 16:15:00	RUNNING	...

图 19: 查看 Flink UI

通过 Flink UI 查看运行信息

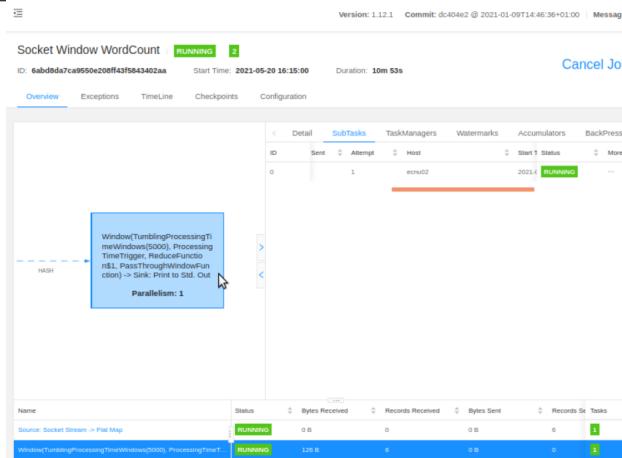


图 20: 查看 Flink UI

可以看到是在 ecnu02 上运行，因此我们查看 ecnu02 节点上的日志

```
[dase-dis@10-24-21-148:~/hadoop-2.10.1/logs/userlogs/application_1621498351698_0002/container_1621498351698_0002_01_000002$ ls
prelaunch.err prelaunch.out taskmanager.err taskmanager.log taskmanager.out
[dase-dis@10-24-21-148:~/hadoop-2.10.1/logs/userlogs/application_1621498351698_0002/container_1621498351698_0002_01_000002$ tail taskmanager.out
i : 1
wengsiyang : 1
love : 1
i : 1
dase : 1
love : 1
```

图 21: 查看位于从节点的输出

Part 5

思考题

Section 1

除了 Per-Job 运行模式之外，Flink on Yarn 还支持 Application 运行模式，请采用 Application 运行模式来运行 Flink 程序，并通过 jps 命令观察 Application 运行模式与 Per-Job 运行模式下的进程有何不同。

我们使用 application mode 启动一次词频统计程序，并查看其细节

```
[dase-local@10-24-21-138:~/flink-1.12.1/bin]$ ./flink run-application -t yarn-application ~/flink-1.12.1/examples/streaming/SocketWindowWordCount.jar --port 8888
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [/home/dase-local/flink-1.12.1/lib/log4j-slf4j-impl-2.12.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [/home/dase-local/hadoop-2.10.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory].
2021-05-27 16:30:51.941 WARN org.apache.flink.yarn.configuration.YarnLogConfigUtil [] - The configuration directory ('/home/dase-local/flink-1.12.1/conf') already contains a LOG4J config file. If you want to use logback, then please delete or rename the log configuration file.
2021-05-27 16:30:51.973 INFO org.apache.hadoop.yarn.client.RMProxy [] - Connecting to ResourceManager at /0.0.0.0:8032
2021-05-27 16:30:52.167 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - No path for the flink jar passed. Using the location of class org.apache.flink.yarn.YarnClusterDescriptor to locate the jar
2021-05-27 16:30:52.288 INFO org.apache.hadoop.conf.Configuration [] - resource-types.xml not found
2021-05-27 16:30:52.288 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils [] - Unable to find 'resource-types.xml'.
2021-05-27 16:30:52.294 INFO org.apache.hadoop.util.resource.ResourceUtils [] - Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
```

图 22: 使用 Application Mode 提交任务

由图 23 和图 24 可以看出，在单机伪分布式下分别运行 application 和 per-job 模式的 flink 程序，在所启动的进程上有所差别，前者的作业管理进程为 YarnApplicationClusterEntryPoint，后者的作业管理进程为 YarnJobClusterEntryPoint。

```
[dase-local@10-24-21-130:~]$ jps
7872 NameNode
8081 DataNode
10961 YarnTaskExecutorRunner
8502 ResourceManager
10839 YarnApplicationClusterEntryPoint
9225 JobHistoryServer
11114 Jps
8333 SecondaryNameNode
8687 NodeManager
```

图 23: Application Mode 进程状态

```
[dase-local@10-24-21-130:~]$ jps
2900 DataNode
3492 SecondaryNameNode
5476 Jps
5045 YarnJobClusterEntrypoint
3669 ResourceManager
5174 YarnTaskExecutorRunner
4199 JobHistoryServer
2696 NameNode
3849 NodeManager
4540 CliFrontend
[dase-local@10-24-21-130:~]$
```

图 24: Per-Job Mode 进程状态

在此基础上，我们通过查阅资料，可以知道 Per job 模式下，Flink 任务对 jar 包的解析以及生成 JobGraph 是在 Client 上执行的，然后将生成的 Job Graph 提交到 Flink 集群。而 Yarn Application 模式下会在客户端将运行任务需要的依赖都上传到 Flink Master，然后在 Master 端进行任务的提交。简而言之，Application 模式让 Master 将解析程序和下载 jar 包等工作完成，相对于其他提交方式减轻了客户端的任务负担。

对减轻 Client 负担的具体的表述可以在 flink 的文档中找到： *Despite their differences, both session and per-job modes execute the application's main() method, i.e. the pre-flight phase, on the client side*

根据文档的解释，这么处理的优点是，当 Flink 的客户端和集群分离时，采用传统方法需要传输相关的依赖，对通讯带宽压力大。采用 Application 模式则由集群下载依赖，对客户端和集群间的带宽依赖较小。这种模式的结构图如下所示（图源：Flink 官方文档）

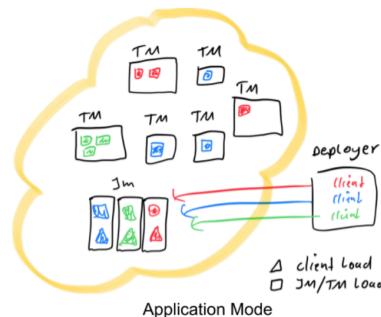


图 25: Flink Application Mode

参考资料：

- Application Deployment in Flink: Current State and the new Application Mode
- flink 教程-详解 flink 1.11 中的新部署模式

Section 2

上述 Flink 程序中仅包含一个 Flink 作业，然而一个 Flink 程序中允许包含多个 Flink 作业，附录 A 是一个包含 2 个作业的 Flink 程序，请分别使用 Per-Job 运行模式和 Application 运行模式来运行该 Flink 程序，并结合 Yarn 和 Flink 的 Web UI 观察两种运行模式分别启动了多少个 Flink 集群。

运行思考题 2 所提供的程序，我们发现虽然代码中有两处 run 语句分别监听 8888 和 8889 端口，但运行程序时并不会并行地运行两个 Flink 作业。我们观察到只有在断开 Netcat 监听的第一个端口之后，程序才会开始监听第二个端口的输入，此前在第二个端口的输入都不会产生任何输出，如图 26 和图 27 所示。

此外，在使用 Per-job 模式运行程序时，第一阶段结束后，程序会重新启动一个 Job，此时无法访问第一个 Job 对应的 Flink Web UI，刷新后会退回到 Yarn 的 Web UI。并且，两个 Job 都分别拥有自己的 Job ID。因此，两个 Job 的输出并不会被写到同一个输出文件中。

```

2021-05-27 19:52:28,502 INFO org.apache.flink.yarn.YarnClusterDescriptor
105022187_0019'.
Job has been submitted with JobID 03c18dc607c2f761316919c653299309
Program execution finished
Job with JobID 03c18dc607c2f761316919c653299309 has finished.
Job Runtime: 84254 ms

2021-05-27 19:53:52,649 WARN org.apache.flink.yarn.configuration.YarnLogConfigU
12.1/conf') already contains a LOG4J config file.If you want to use logback, the
2021-05-27 19:53:52,664 INFO org.apache.hadoop.yarn.client.RMProxy
2021-05-27 19:53:52,666 INFO org.apache.flink.yarn.YarnClusterDescriptor
.apache.flink.yarn.YarnClusterDescriptor to locate the jar
2021-05-27 19:53:52,691 WARN org.apache.flink.yarn.YarnClusterDescriptor
iable is set. The Flink YARN Client needs one of these to be set to properly loa
2021-05-27 19:53:52,695 INFO org.apache.flink.yarn.YarnClusterDescriptor
048 MB to make up an integer multiple of its minimum allocation memory (1024 MB,
be used by Flink.
2021-05-27 19:53:52,696 INFO org.apache.flink.yarn.YarnClusterDescriptor
taskManagerMemoryMB=1728, slotsPerTaskManager=2]
2021-05-27 19:53:55,878 INFO org.apache.flink.yarn.YarnClusterDescriptor
2021-05-27 19:53:56,086 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientI
2021-05-27 19:53:56,086 INFO org.apache.flink.yarn.YarnClusterDescriptor
2021-05-27 19:53:56,088 INFO org.apache.flink.yarn.YarnClusterDescriptor
2021-05-27 19:54:06,220 INFO org.apache.flink.yarn.YarnClusterDescriptor
2021-05-27 19:54:06,221 INFO org.apache.flink.yarn.YarnClusterDescriptor
105022187_0020'.
Job has been submitted with JobID 24f5ff7c4ab3446e24527c1698b72c4f
Program execution finished
Job with JobID 24f5ff7c4ab3446e24527c1698b72c4f has finished.
Job Runtime: 1293942 ms

```

图 26: 思考题 2-1

Running Job List						
Job Name	Start Time	Duration	End Time	Tasks	Status	
DemoWith2Env2Exec8889	2021-05-27 20:21:55	8s	-	2 2	RUNNING	
Completed Job List						
Job Name	Start Time	Duration	End Time	Tasks	Status	
DemoWith2Env2Exec8888	2021-05-27 20:20:17	1m 37s	2021-05-27 20:21:54	2 2	FINISHED	

图 27: 思考题 2-2

```

dase-local:9~24~212:~/softwares/flink-1.12.1/bin$ ./flink run-application -t yarn-application exp14.jar
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/dase-local/softwares/flink-1.12.1/lib/log4j-slf4j-impl-2.12.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dase-local/softwares/hadoop-2.18.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2021-05-27 20:20:09,355 WARN org.apache.flink.yarn.configuration.YarnLogConfigUtil [] - The configuration directory ('/home/dase-local/softwares/flink-1.12.1/lib/log4j-slf4j-impl-2.12.1.jar') contains a YARN log configuration file. If you want to use logback, then please delete or rename the log configuration file.
2021-05-27 20:20:09,410 INFO org.apache.hadoop.yarn.client.RMProxy [] - Connecting to ResourceManager at /0.0.0.0:8082
2021-05-27 20:20:09,709 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - No path for the Flink jar passed. Using the location of class org.apache.flink.yarn.YarnClusterDescriptor to locate the jar
2021-05-27 20:20:09,881 INFO org.apache.hadoop.conf.Configuration [] - resource-types.xml not found
2021-05-27 20:20:09,882 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils [] - Unable to find resource-types.xml.
2021-05-27 20:20:09,888 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils []
[] - Adding resource type - name = memory-MB, units = Mi, type = COUNTABLE
2021-05-27 20:20:09,889 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils [] - Adding resource type - name = vcores, units = , type = COUNTABLE
2021-05-27 20:20:09,892 WARN org.apache.flink.yarn.YarnClusterDescriptor [] - Neither the HADOOP_CONF_DIR nor the YARN_CONF_DIR environment variable is set. The Flink YARN Client needs one of these to be set to properly load the Hadoop configuration for accessing YARN.
2021-05-27 20:20:09,936 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - The configured JobManager memory is 1600 MB. YARN will allocate 2048 MB to make up an integer multiple of its minimum allocation memory (1024 MB, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 448 MB may not be used by Flink.
2021-05-27 20:20:09,937 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - The configured TaskManager memory is 1728 MB. YARN will allocate 2048 MB to make up an integer multiple of its minimum allocation memory (1024 MB, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 320 MB may be used by Flink.
2021-05-27 20:20:09,937 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - Cluster specification: ClusterSpecification(masterMemoryMB=1600, taskManagerMemoryMB=1728, slotsPerTaskManagers=2)
2021-05-27 20:20:12,525 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - Submitting application master application_1622105022187_0021
2021-05-27 20:20:12,563 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl [] - Submitted application application_1622105022187_0021
2021-05-27 20:20:12,564 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - Waiting for the cluster to be allocated
2021-05-27 20:20:12,566 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - Deploying cluster, current state ACCEPTED
2021-05-27 20:20:18,129 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - YARN application has been deployed successfully.
2021-05-27 20:20:18,130 INFO org.apache.flink.yarn.YarnClusterDescriptor [] - Found Web Interface ecnu01:10545 of application 'application_1622105022187_0021'.

```

图 28: 思考题 2-3

application_1622105022187_0021	dase-local	Flink Application Cluster	Apache Flink	default	0	Thu May 27 20:20:12 +0800 2021	Thu May 27 20:23:26 +0800 2021	FINISHED	SUCCEEDED	N/A
application_1622105022187_0020	dase-local	Flink per-job cluster	Apache Flink	default	0	Thu May 27 19:53:56 +0800 2021	Thu May 27 20:15:40 +0800 2021	FINISHED	SUCCEEDED	N/A
application_1622105022187_0019	dase-local	Flink per-job cluster	Apache Flink	default	0	Thu May 27 19:52:22 +0800 2021	Thu May 27 19:52:33 +0800 2021	FINISHED	SUCCEEDED	N/A

图 29: 思考题 2-4

在 Per job 模式中，两个阶段有不同的 Job ID，被 Flink 视为不同集群的 Job，被 Yarn 视为不同的 application。

在 application 模式中两个阶段共享 Job ID，被 Flink 视为同一集群中的两个 Job，被 Yarn 视为同一个 application。因此可以认为前者启动了两个集群，后者启动了一个集群。因为这部分实验是在单机伪分布式下完成的，无法观测到节点的变化，接下来在分布式模式下重新做一次这个实验。

首先，在 Application 模式下，有一个 task manager。且在两个阶段，会复用其中的一个 container，在图 30 中可以看出，一共用了 3 个 container，其中 container***001 是在两个阶段均使用的，推测其为 Job Manager

第二阶段，可以看到统计的 container 是 3 所以 Application 模式下允许提交由多个 Job 组成的应用程序，图 30 中的两个阶段 job 会被归为一个 application，在一个集群里执行。

Total Allocated Containers: 3			
Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.			
	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	0		
Num Rack Local Containers (satisfied by)	0	0	
Num Off Switch Containers (satisfied by)	0	0	3

Show 20 > entries				Search:
Container ID	Node	Container Exit Status	Logs	
container_1622105022187_0005_01_000003	http://ecnu02:8042	0	Logs	
container_1622105022187_0005_01_000001	http://ecnu02:8042	0	Logs	

图 30: 思考题 2-5

而在 Per job 环境下，两个阶段分别起两个 container。可以看到两个阶段甚至所在的节点都不一样了，如图 33 和图 31 中所示，这里显示节点分别为 ecnu02 和 ecnu04，而 JobManager 相当于整个集群的 Master 节点，JobManager 负责整个集群的任务管理和资源管理。

对于 Per job 模式下不同 Job 被分配给了不同的 JobManager，Flink 会为每个任务单独创建一个集群，当任务执行完毕，集群会自动关闭，回收资源，实现资源隔离。

华东师范大学数据科学与工程学院学生实验报告

Show 20 entries					
Attempt ID	Started	Node	Logs	Nodes blacklisted	
appattempt_1622190753050_0003_000001	Fri May 28 16:33:46 +0800 2021	http://ecnu02:8042	Logs	0	
Showing 1 to 1 of 1 entries					

图 31: 思考题 2-6

Show 20 entries		
Container ID	Node	
container_1622190753050_0003_01_000002	http://ecnu02:8042	0
container_1622190753050_0003_01_000001	http://ecnu02:8042	0
Showing 1 to 2 of 2 entries		

图 32: 思考题 2-7

Show 20 entries					
Attempt ID	Started	Node	Logs		
appattempt_1622190753050_0004_000001	Fri May 28 16:36:08 +0800 2021	http://ecnu04:8042	Logs	0	
Showing 1 to 1 of 1 entries					

图 33: 思考题 2-8

简而言之，经过我们小组的实验，我们发现这个差异是体现在 Per Job 开了两个集群，而在 Application 模式下开了一个集群，因为没有重启一个新的 Job Manager。

最后得到如下结论

- Per Job 模式下 flink 为每个提交的 Job 启动一个 Flink 集群。Job 完成后，集群将关闭，所有残留的资源（例如文件）也将被清除。在 Per Job 模式可以更好地隔离资源，因为任何出现异常的 Job 不会影响任何其他 Job。与 Per Job 模式相比，Application 模式允许提交由多个 Job 组成的应用程序，属于同一应用程序的 Job 应该被关联起来，并视为一个单元。在 JobManager 中执行 main() 方法，减轻了客户端的资源消耗。
- Per Job 模式下程序代码都是在客户端编译完成。这里的客户端就是我们执行 flink run 启动的程序，也就是通过 jps 查看到的 CliFrontend 进程。而在 Application 模式下，main() 方法在集群上而不是在客户端执行，客户端负载被转移到每个应用程序的 JobManager。

参考资料：[Flink-1.11 新的部署模型 application mode](#)

Part 6

实验补充：基于 Yarn 同时部署 Flink，Spark 和 MapReduce 应用

Yarn 是一个资源调度平台，负责为运算程序提供服务器运算资源，而 MapReduce 等则运行于其上的应用程序，如图 34 所示。

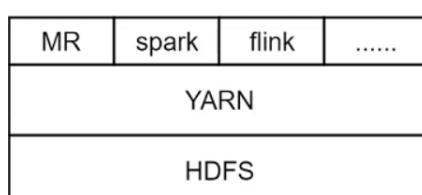


图 34: Yarn Structure

华东师范大学数据科学与工程学院学生实验报告

使用 Yarn 是为了对集群资源进行整合，让我们资源得到最大化利用，同一套硬件资源集群上可以运行多个任务，但我们的实验中目前只涉及了在 Yarn 上部署 Flink 任务。我们现在尝试一下如何在分布式模式下基于 Yarn 同时部署 Flink, Spark 和 MapReduce 三个分布式计算系统的应用。

Remark：实验中遇到了多个与内存分配相关的错误，先进行简单的罗列，最后再给出解决所有报错后的部署情况。

实验开始时经过了一系列调整，我们将客户端和主节点同时添加为从节点后。

在 spark 启动之后成功启动了 flink，但是 Resource Manager 因为内存不足而无法启动，但通过 WebUI 的信息看出，目前节点的空闲资源非常充足，尤其是在将四个节点同时启动为从节点之后

Nodes											Containers		Memory			Virtual CPU			GPU		
Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	Vcores Used	Vcores Avail	GPUs Used	GPUs Avail	Version							
/default-rack	RUNNING	ecnu04:14691	ecnu04:8042	Thu May 20 20:54:45 +0800 2021	0	0 B	8 GB	0	8	0	0	0	0	2.10.1							
/default-rack	RUNNING	ecnu02:33871	ecnu02:8042	Thu May 20 20:54:45 +0800 2021	1	2 GB	6 GB	1	7	0	0	0	0	2.10.1							
/default-rack	RUNNING	ecnu03:21989	ecnu03:8042	Thu May 20 20:54:45 +0800 2021	1	2 GB	6 GB	1	7	0	0	0	0	2.10.1							
/default-rack	RUNNING	ecnu01:10439	ecnu01:8042	Thu May 20 20:54:46 +0800 2021	3	5 GB	3 GB	3	5	0	0	0	0	2.10.1							

图 35: 各节点资源均没有被完全利用

于是怀疑这里的错误是队列数量不够导致的，所以我们小组先尝试了增加队列

```
<property>
  <name>yarn.scheduler.capacity.root.queues</name>
  <value>a,default</value>
  <description>
    The queues at the this level (root is the root queue).
  </description>
</property>
```

图 36: 增加队列

发现修改之后无法启动 Resource Manager，查看日志，发现报错信息如下：

```
urceManager: Error starting ResourceManager
java.lang.IllegalArgumentException: Illegal capacity of -1.0 for queue root.a
  at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capa
citySchedulerConfiguration.getNonLabeledQueueCapacity(CapacitySchedulerConfigura
tion.java:418)
  at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CSQu
eueUtils.loadCapacitiesByLabelsFromConf(CSQueueUtils.java:126)
  at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CSQu
```

图 37: Resource Manager 错误日志

查询了这篇博客：[Hadoop 启动 Resource Manager 失败的解决办法](#)，在各个节点添加了如下配置

```

</property>
<property>
  <name>yarn.scheduler.maximum-allocation-vcores</name>
  <value>7</value>
</property>
<property>
  <name>yarn.scheduler.capacity..maximum-am-resource-percent</name>
  <value>50</value>
</property>
<property>
  <name>yarn.nodemanager.pmem-check-enabled</name>
  <value>>false</value>
</property>

```

图 38: 修改 Maximum Capacity

修改后，可以正常启动 Resource Manager，并在 WebUI 界面可以看到启动的队列数量

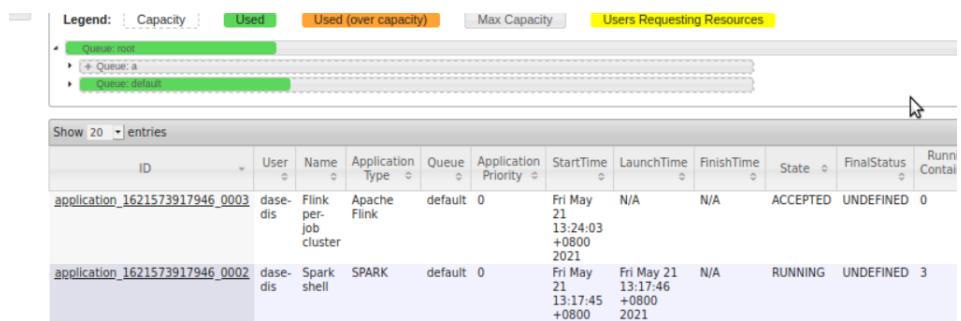


图 39: 队列状态

但是，居然还是没成功启动，再次查看队列的使用情况

我们发现：Yarn 不会自动在多个队列间进行分配，因此进行多队列的配置意义不大。相反的是，因为所有的队列资源都是来源于 root 队列，这也就意味着刚刚的改动反而导致单个队列的资源不足，导致只能启动一个服务，比进行调整前的性能反而下降了！

因此重新观察配置，查阅资料后发现 maximum-am-resource-percent 这个属性需要修改

```

<property>
  <name>yarn.scheduler.capacity.maximum-am-resource-percent</name>
  <value>0.5</value>
  <description>
    Maximum percent of resources in the cluster which can be used to run
    application masters i.e. controls number of concurrent running
    applications.
  </description>
</property>

```

图 40: 修改 maximum-am-resource-percent

修改后重启服务，发现在多队列的情况下，可以成功同时启动 Spark 和 Flink 应用。

华东师范大学数据科学与工程学院学生实验报告

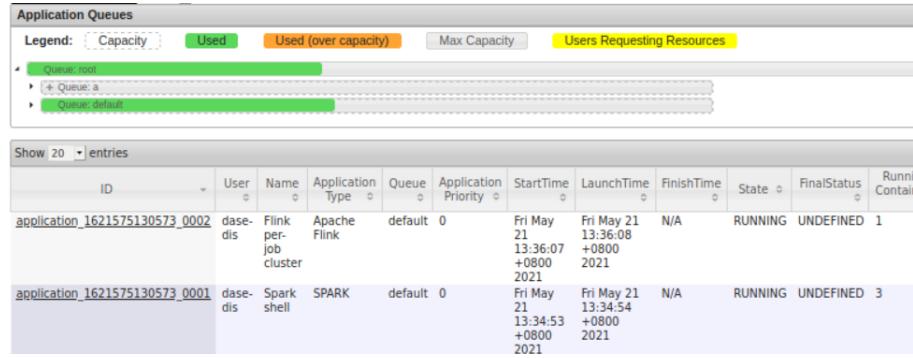


图 41: 同时启动 Flink 和 Spark 成功

即便如此配置之后能够同时启动，Yarn 依然会把它们放在同一个队列中。在此基础上启动 MR，可以成功运行并结束。

The screenshot shows the 'Application Queues' interface. At the top, there's a legend with four items: 'Capacity' (green bar), 'Used' (green bar), 'Used (over capacity)' (orange bar), and 'Max Capacity' (grey bar). Below the legend, a tree view shows 'Queue root' expanded to show 'Queue: a' and 'Queue: default'. A large green bar indicates usage across these queues.

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Runnig Contain
application_1621514085901_0002	dase-dis	Flink per-job cluster	Apache Flink	default	0	Thu May 20 20:37:03 +0800 2021	Thu May 20 20:37:03 +0800 2021	N/A	RUNNING	UNDEFINED	2
application_1621514085901_0001	dase-dis	Spark shell	SPARK	default	0	Thu May 20 20:35:37 +0800 2021	Thu May 20 20:35:38 +0800 2021	N/A	RUNNING	UNDEFINED	3

图 42: 运行 Spark Flink

The screenshot shows the 'Application Queues' interface. At the top, there's a legend with four items: 'Capacity' (green bar), 'Used' (green bar), 'Used (over capacity)' (orange bar), and 'Max Capacity' (grey bar). Below the legend, a tree view shows 'Queue root' expanded to show 'Queue: a' and 'Queue: default'. A large green bar indicates usage across these queues.

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Runnig Conta
application_1621514085901_0003	dase-dis	grep-search	MAPREDUCE	default	0	Thu May 20 20:54:27 +0800 2021	N/A	N/A	ACCEPTED	UNDEFINED	0
application_1621514085901_0002	dase-dis	Flink per-job cluster	Apache Flink	default	0	Thu May 20 20:37:03 +0800 2021	Thu May 20 20:37:03 +0800 2021	N/A	RUNNING	UNDEFINED	2
application_1621514085901_0001	dase-dis	Spark shell	SPARK	default	0	Thu May 20 20:35:37 +0800 2021	Thu May 20 20:35:38 +0800 2021	N/A	RUNNING	UNDEFINED	3

图 43: 运行 Spark Flink MapReduce

最后再看一下队列情况

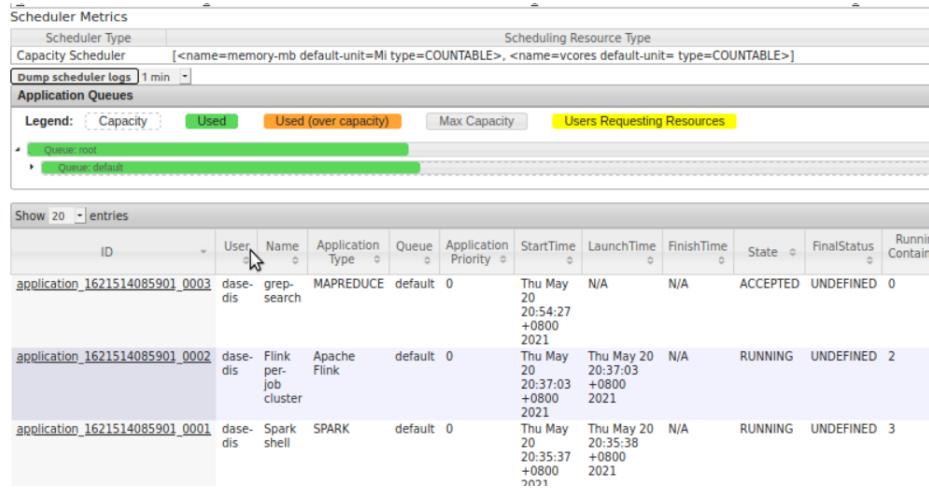


图 44: 运行 Spark Flink MapReduce 后的队列

即便如此配置之后能够同时启动，yarn 依然会把多个服务放在同一个队列中。在此基础上启动 MapReduce，可以成功运行并结束，剩下的操作没有再出现报错，具体流程如下列图片所示。

完整且正确的部署流程如下

首先在客户端启动 Spark Shell

tips: 这里上面出现了一些和主机名相关的 Warning 信息，我们查询了一些资料后认为是分布式模式下启动服务的时候 ip 地址解析有出错，但是并没有影响到实验的正常进行。

```

dase-dis@10-23-74-82: ~
at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:929)
at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

21/05/21 13:17:42 WARN yarn.Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
21/05/21 13:17:43 WARN hdfs.DataStreamer: Abandoning BP-1547215439-10.24.21.138-161660711504:b1_k_1073742310_1490
21/05/21 13:17:43 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[10.23.74.82:50010,DS-548cd792-9d9e-4939-8dcc-fad9763b41d7,DISK]
21/05/21 13:17:43 WARN hdfs.DataStreamer: Abandoning BP-1547215439-10.24.21.138-161660711504:b1_k_1073742313_1493
21/05/21 13:17:45 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[10.23.74.82:50010,DS-548cd792-9d9e-4939-8dcc-fad9763b41d7,DISK]
21/05/21 13:17:52 WARN hdfs.DataStreamer: Abandoning BP-1547215439-10.24.21.138-161660711504:b1_k_1073742315_1495
21/05/21 13:17:52 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[10.23.74.82:50010,DS-548cd792-9d9e-4939-8dcc-fad9763b41d7,DISK]
Spark context Web UI available at http://encu04:4040
Spark context available as `sc` (master = yarn, app id = application_1621573917946_0002).
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> 

```

图 45: 客户端：启动 Spark Shell

tips: 因为镜像版本的不同，这里的 Spark Shell 可能会一直被卡着起不来，出现“Failed to construct terminal”的报错，这个时候要检查一下之前第一次 Spark 实验的 bashrc 配置文件是否正确

Spark Shell 启动卡住问题参考一篇 StackOverflow 上的帖子：[SBT Error: “Failed to construct terminal; falling back to unsupported…”](#)

其实就是一句话的事：

```
export TERM=xterm-color
```

Spark Shell 成功启动后，再启动 Flink 词频统计应用程序，输入一些文本

华东师范大学数据科学与工程学院学生实验报告

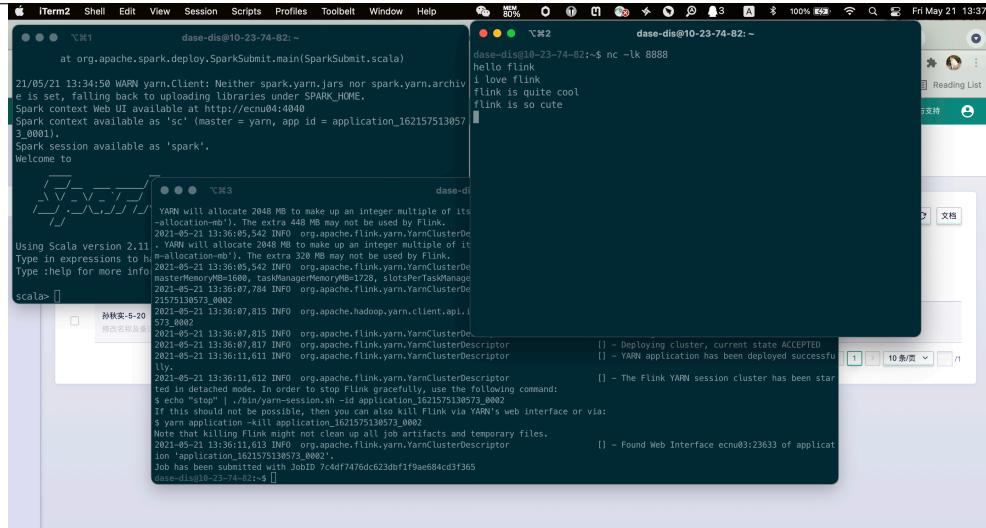


图 46: 客户端: 启动 Flink 应用程序

```
dase-dis@10-23-74-82:~/hadoop-2.10.1$ ./bin/yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.10.1.jar grep input/grep output
grep 'dfe[a-z.]+'
```

21/05/21 13:46:54 INFO client.RMProxy: Connecting to ResourceManager at ecnu01/10.24.21.236:8023
21/05/21 13:46:55 INFO input.FileInputFormat: Total input files to process : 29
21/05/21 13:46:55 INFO mapreduce.JobSubmitter: number of splits:29
21/05/21 13:46:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621575130573_0003
21/05/21 13:46:56 INFO conf.Configuration: resource-types.xml not found
21/05/21 13:46:56 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
21/05/21 13:46:56 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
21/05/21 13:46:56 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
21/05/21 13:46:56 INFO impl.YarnClientImpl: Submitted application application_1621575130573_0003
21/05/21 13:46:56 INFO mapreduce.Job: The url to track the job: http://ecnu01:8088/proxy/application_1621575130573_0003
21/05/21 13:46:56 INFO mapreduce.Job: Running job: job_1621575130573_0003
21/05/21 13:47:01 INFO mapreduce.Job: Job job_1621575130573_0003 running in uber mode : false
21/05/21 13:47:01 INFO mapreduce.Job: map 0% reduce 0%
21/05/21 13:47:09 INFO mapreduce.Job: map 17% reduce 0%
21/05/21 13:47:14 INFO mapreduce.Job: map 21% reduce 0%
21/05/21 13:47:15 INFO mapreduce.Job: map 34% reduce 0%
21/05/21 13:47:20 INFO mapreduce.Job: map 48% reduce 0%
21/05/21 13:47:25 INFO mapreduce.Job: map 55% reduce 0%
21/05/21 13:47:26 INFO mapreduce.Job: map 62% reduce 0%
21/05/21 13:47:30 INFO mapreduce.Job: map 72% reduce 0%

图 47: 客户端: MapReduce 任务执行中

当前客户端状况如图 48 所示, Spark Shell 正在运行, Flink 词频统计程序正在执行, MapReduce 词频统计任务正在执行, 没有再出现之前 MapReduce 进程因内存不足被卡死的情况。

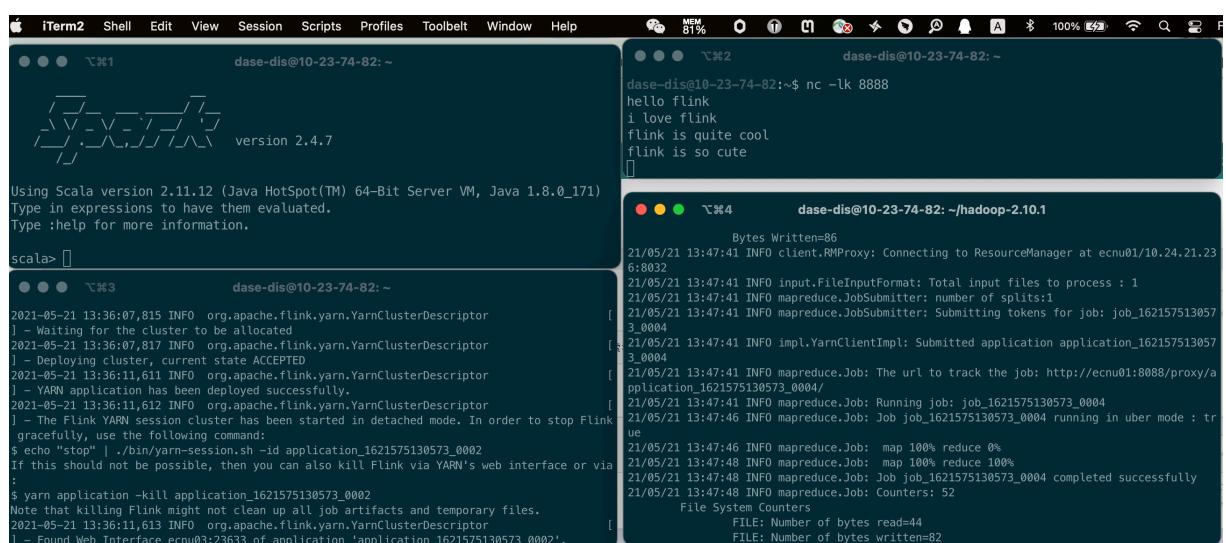
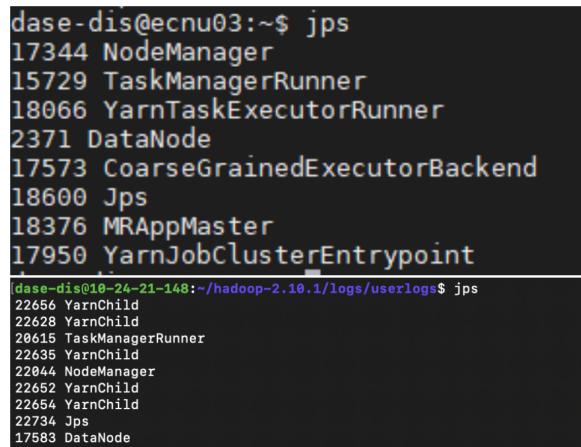


图 48: 客户端: 当前所有终端状况

华东师范大学数据科学与工程学院学生实验报告

在从节点查看进程状态，在图 49 中可以观察到，从节点上同时出现了分别属于 MapReduce，Spark 和 Flink 的任务。

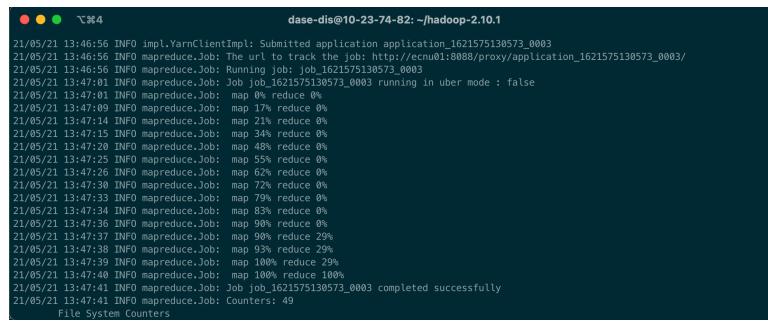


```
dase-dis@ecnu03:~$ jps
17344 NodeManager
15729 TaskManagerRunner
18066 YarnTaskExecutorRunner
2371 DataNode
17573 CoarseGrainedExecutorBackend
18600 Jps
18376 MRAppMaster
17950 YarnJobClusterEntrypoint

[dase-dis@10-24-21-148:~/hadoop-2.10.1/logs/userlogs$ jps
22656 YarnChild
22628 YarnChild
20615 TaskManagerRunner
22635 YarnChild
22044 NodeManager
22652 YarnChild
22654 YarnChild
22734 Jps
17583 DataNode
```

图 49: 从节点: 查看当前进程状态

接着等待 MapReduce 应用执行完毕，如图 50 和 51 所示



```
● ● ● TextEdit dase-dis@10-23-74-82:~/hadoop-2.10.1
21/05/21 13:46:56 INFO impl.YarnClientImpl: Submitted application application_1621575130573_0003
21/05/21 13:46:56 INFO mapreduce.Job: The url to track the job: http://ecnu01:8088/proxy/application_1621575130573_0003
21/05/21 13:47:00 INFO mapreduce.Job: Running job: job_1621575130573_0003
21/05/21 13:47:00 INFO mapreduce.Job: Job job_1621575130573_0003 running in uber mode : false
21/05/21 13:47:09 INFO mapreduce.Job: map 0% reduce 0%
21/05/21 13:47:09 INFO mapreduce.Job: map 17% reduce 0%
21/05/21 13:47:14 INFO mapreduce.Job: map 21% reduce 0%
21/05/21 13:47:15 INFO mapreduce.Job: map 34% reduce 0%
21/05/21 13:47:20 INFO mapreduce.Job: map 48% reduce 0%
21/05/21 13:47:25 INFO mapreduce.Job: map 55% reduce 0%
21/05/21 13:47:26 INFO mapreduce.Job: map 62% reduce 0%
21/05/21 13:47:30 INFO mapreduce.Job: map 72% reduce 0%
21/05/21 13:47:33 INFO mapreduce.Job: map 79% reduce 0%
21/05/21 13:47:34 INFO mapreduce.Job: map 83% reduce 0%
21/05/21 13:47:38 INFO mapreduce.Job: map 96% reduce 0%
21/05/21 13:47:41 INFO mapreduce.Job: map 99% reduce 0%
21/05/21 13:47:43 INFO mapreduce.Job: map 100% reduce 0%
21/05/21 13:47:48 INFO mapreduce.Job: map 100% reduce 29%
21/05/21 13:47:49 INFO mapreduce.Job: map 100% reduce 100%
21/05/21 13:47:49 INFO mapreduce.Job: Job job_1621575130573_0003 completed successfully
21/05/21 13:47:49 INFO mapreduce.Job: Counters: 49
File System Counters
```

图 50: 客户端: MapReduce 任务执行完毕



```
● ● ● TextEdit dase-dis@10-23-74-82:~/hadoop-2.10.1
Reduce output records=0
Spilled Records=0
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=24
CPU time spent (ms)=780
Physical memory (bytes) snapshot=882442240
Virtual memory (bytes) snapshot=5917265920
Total committed heap usage (bytes)=608174080
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=86
File Output Format Counters
  Bytes Written=0
dase-dis@10-23-74-82:~/hadoop-2.10.1$ ips
```

图 51: 客户端: MapReduce 任务执行完毕 Cont'd

执行完后，三者的最终状态如图 52 所示

华东师范大学数据科学与工程学院学生实验报告

The screenshot shows four terminal windows side-by-side:

- Terminal 1 (dase-dis@10-23-74-82: ~): Displays the Flink logo and "version 2.4.7".
- Terminal 2 (dase-dis@10-23-74-82: ~): Shows a nc -lk 8888 session with messages: "hello flink", "i love flink", "flink is quite cool", and "flink is so cute".
- Terminal 3 (dase-dis@10-23-74-82: ~): A Scala REPL session with the command "scala> []".
- Terminal 4 (dase-dis@10-23-74-82: ~/hadoop-2.10.1): Displays Hadoop metrics: Reduce output records=0, Spilled Records=0, Shuffled Maps =1, Failed Shuffles=0, Merged Map outputs=1, GC time elapsed (ms)=24, CPU time spent (ms)=780, Physical memory (bytes) snapshot=882442240, Virtual memory (bytes) snapshot=5917265920, Total committed heap usage (bytes)=606174080. It also lists Shuffle Errors: BAD_ID=0, CONNECTION=0, IO_ERROR=0, WRONG_LENGTH=0, WRONG_MAP=0, WRONG_REDUCE=0, File Input Format Counters: Bytes Read=86, File Output Format Counters: Bytes Written=0.

图 52: 客户端: 最终状态

到此为止，本小组成功基于 Yarn 同时部署 Flink，Spark 和 MapReduce 应用。

Part 7

参考资料（链接）

- (1) 解决 Spark 无法创建终端问题: SBT Error: “Failed to construct terminal; falling back to unsupported…”
- (2) 解决因资源不足导致 RM 启动失败: Hadoop 启动 Resource Manager 失败的解决办法
- (3) 解决 Flink 报错内存不足: flink1.10 在 yarn 上运行 job 报内存超出
- (4) Flink 内存空间设置参考: talk about the memory size setting of flink taskmanager
- (5) Application-mode 参考 1: Application Deployment in Flink: Current State and the new Application Mode
- (6) Application-mode 参考 2: flink 教程-详解 flink 1.11 中的新部署模式