华东师范大学数据科学与工程学院实验报告

课程名称：分布式编程模型与系统	**年级：**2021	**上机实践成绩：**
指导教师：徐辰	**姓名：**彭一珅	**学号：**10215501412
上机实践名称： Flink 部署	**上机实践日期：**	2024.6.5

一、实验目的

学习 Flink 的部署，简单实用 Scala Shell

查看 Flink 的运行日志，体会与其他系统运行过程中查看日志方式的区别

通过系统部署理解体系架构，体会流计算系统与批处理系统之间的区别

二、实验任务

完成 Flink 的单机伪分布式部署以及分布式部署

两种部署方式下分别以 Attached 和 Detached 提交方式运行示例程序

三、实验环境

操作系统 Ubuntu18.04

JDK 版本 1.8

Flink 版本 1.12.1

四、实验过程

单机集中式部署

登录用户，下载并安装 Flink，使用 shell 运行 DataStream 程序，在 scala>后输入以下指令

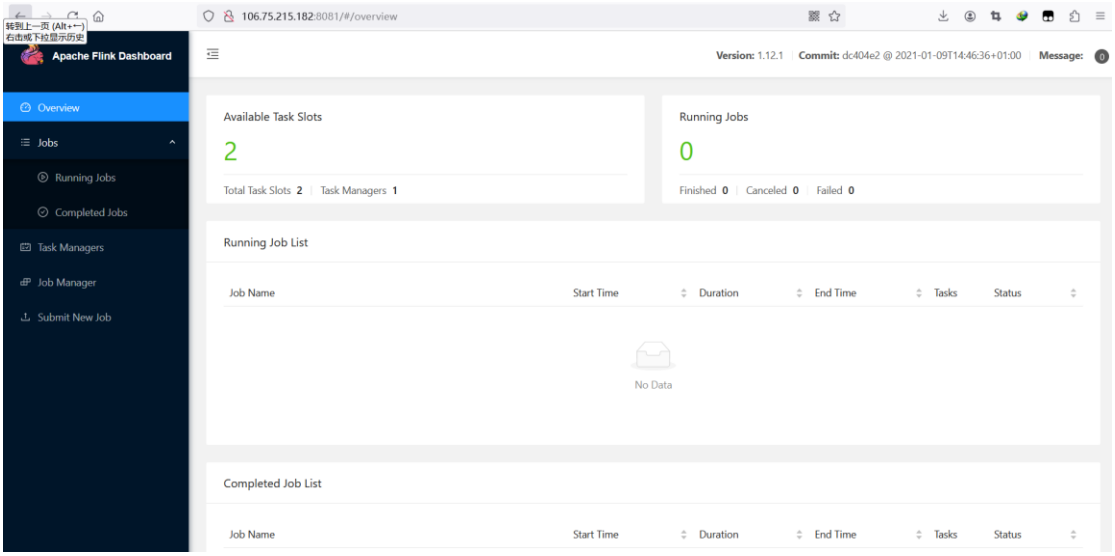
```
scala> val countsstreaming=textstreaming.flatMap { _.toLowerCase.split("\\W+") }.map { (_, 1) }.key
By(0).sum(1)
warning: there was one deprecation warning; re-run with -deprecation for details
countsstreaming: org.apache.flink.streaming.api.scala.DataStream[(String, Int)] = org.apache.flink.
streaming.api.scala.DataStream@3644d305

scala> countsstreaming.print()
res0: org.apache.flink.streaming.api.datastream.DataStreamSink[(String, Int)] = org.apache.flink.st
reaming.api.datastream.DataStreamSink@17bda3

scala> senv.execute()
(a,1)
(a,2)
(b,1)
(b,2)
(c,1)
res1: org.apache.flink.api.common.JobExecutionResult =
Program execution finished
Job with JobID 491ce76e02117f39aadb0255e0c3218d has finished.
Job Runtime: 1495 ms

scala>
```

单机伪分布式部署

使用 jps 查看进程，启动服务后输出如下

```
ubuntu@10-23-3-42:~$ jps
2252743 Jps
2252423 StandaloneSessionClusterEntrypoint
2252687 TaskManagerRunner
ubuntu@10-23-3-42:~$
```

查看 flinkUI 界面，每个 taskManager 有两个 slot



运行 flink dataStream 程序，结果如下所示

```
Last login: Thu May 23 20:47:48 2024 from 180.160.46.171
ubuntu@10-23-3-42:~$ tail -f ~/flink-1.12.1/log/flink-ubuntu-taskexecutor-0-10-23-3-42.out
(a,1)
(a,2)
(b,1)
(b,2)
(c,1)
```

通过提交 jar 包运行 DataStream 程序：

```
ubuntu@10-23-3-42:~$ tail -f ~/flink-1.12.1/log/flink-ubuntu-taskexecutor-0-10-23-3-42.out
(a,1)
(a,2)
(b,1)
(b,2)
(c,1)
hellp : 1
dase : 1
: 1
hello : 1
ecnu : 1
hello : 2
ecnu : 1
dase : 1
```

启动 netcat 服务，监听 8888 端口

```
ubuntu@10-23-3-42:~$ nc -lk 8888
hello dase
hello ecnu
hello dase
hello ecnu
```

程序运行中存在的进程

```
ubuntu@10-23-3-42:~$ jps
2253984 TaskManagerRunner
2254507 CliFrontend
2253720 StandaloneSessionClusterEntrypoint
2254941 Jps
ubuntu@10-23-3-42:~$
```

输入命令提交 jar 包

```
ubuntu@10-23-3-42:~$ ~/flink-1.12.1/bin/flink run -d ~/flink-1.12.1/examples/streaming/SocketWindow
WordCount.jar --port 8888
```

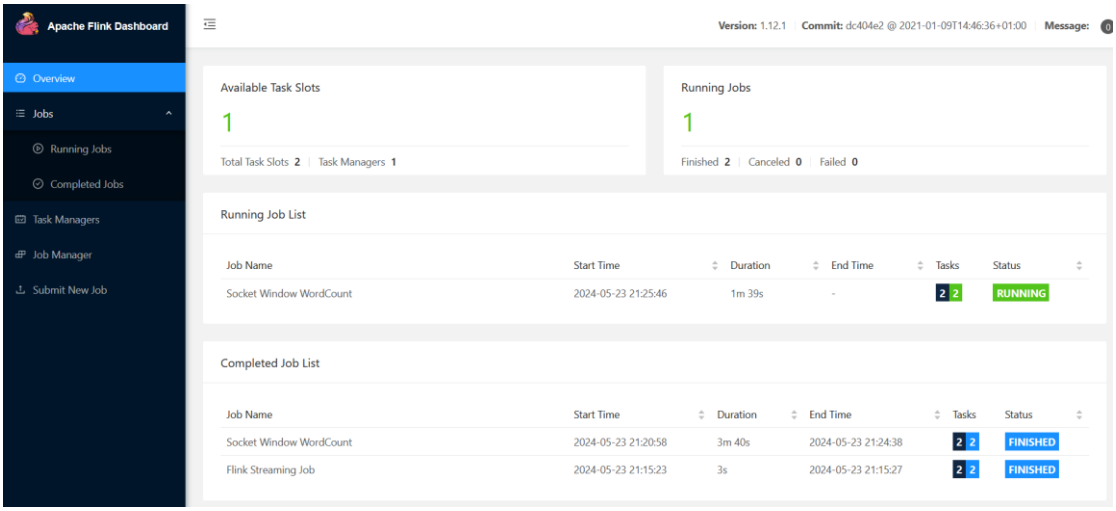
Detached 方式提交示例程序时的进程

```
ubuntu@10-23-3-42:~$ jps
2253984 TaskManagerRunner
2255127 Jps
2253720 StandaloneSessionClusterEntrypoint
```

输出结果：

```
hello : 1
ecnu : 1
hello : 2
ecnu : 1
dase : 1
hello : 1
dase : 1
```

应用程序提交历史记录可以看到右下角的两个 finished



五、实验总结

在本次实验中，完成了 flink 的部署和基本示例程序的运行，通过 attached 和 detached 两种方式进行提交。

华东师范大学数据科学与工程学院实验报告

课程名称：分布式编程模型与系统	**年级：**2021	**上机实践成绩：**
指导教师：徐辰	**姓名：**彭一珅	**学号：**10215501412
上机实践名称： Flink 编程	**上机实践日期：**	2024.6.10

一、实验目的

学习编写简单的基于 DataStream API 的 Flink 程序

掌握在 IDEA 中调试 Flink 相关程序，以及在单机伪分布式、分布式部署方式下提交运行 Flink 程序的方法

二、实验任务

完成 WordCount 示例程序的编写

分别在单机伪分布式和分布式环境下运行 WordCount 示例程序

三、实验环境

操作系统：ubuntu20.04
Jdk 版本：1.8
Hadoop 版本：2.5.1
Flink 版本：1.12.1
Scala 版本：2.11.12

四、实验过程

编写 Flink 应用程序

编写完程序之后，打包成 jar 包上传到云主机，启动 flink 和 netcat 服务

```
ubuntu@10-23-3-42:~$ mkdir ~/flink-1.12.1/myApp
ubuntu@10-23-3-42:~$ cp FlinkWordCount.jar ~/flink-1.12.1/myApp/
ubuntu@10-23-3-42:~$ ~/flink-1.12.1/bin/start-cluster.sh
Starting cluster.
Starting standalonesession daemon on host 10-23-3-42.
Starting taskexecutor daemon on host 10-23-3-42.
ubuntu@10-23-3-42:~$ nc -lk 8888
hello flink
hello dase
hello ecnu
```

通过提交 jar 包运行应用程序

```
ubuntu@10-23-3-42:~$ ~/flink-1.12.1/bin/flink run -c cn.edu.ecnu.flink.example.java.wordcou
ount ~/flink-1.12.1/myApp/FlinkWordCount.jar localhost 8888
Job has been submitted with JobID af2e65845c151a72824160de540f40ea
```

另起一个终端查看输出结果：

```
ubuntu@10-23-3-42:~$ tail -f ~/flink-1.12.1/log/flink-ubuntu-taskexecutor-0-10-23-3-42.out
('hello,1)
(flink,1)
(hello,1)
(dase,1)
(hello,2)
(ecnu,1)
```

五、实验总结

在本次实验中，通过实际操作，掌握了 Flink 应用程序的编写和调试方法，并成功在单机伪分布式和分布式环境下运行了 WordCount 示例程序。这不仅加深了对流计算系统的理解，还提高了分布式编程的实践能力。

华东师范大学数据科学与工程学院实验报告

课程名称：分布式编程模型与系统	**年级**：2021	**上机实践成绩**：
指导教师：徐辰	**姓名**：彭一珅	**学号**：10215501412
上机实践名称：基于 Yarn 部署 Flink	**上机实践日期**：	2024.6.20

一、实验目的

学习在 Yarn 平台上部署 Flink，理解 Standalone 模式与 Yarn 模式的区别

通过 Yarn 模式运行 Flink 程序，体会与 Standalone 模式下运行 Flink 程序的区别

了解 Attached 和 Detached 两者提交方式之间的差异

二、实验任务

完成 Flink on Yarn 的单机伪分布式部署以及分布式部署

完成单机伪分布式部署方式狭隘在 Per-Job 运行模式中分别按照 Attached 和 detached 方式提交应用程序

完成分布部署方式狭隘在 Per-job 运行模式中分别按照 Attached 和 Detached 方式提交应用程序

三、实验环境

操作系统：ubuntu18.04
Jdk 版本：1.8
Hadoop 版本：2.5.1
Flink 版本：1.12.1
Scala 版本：2.11.12

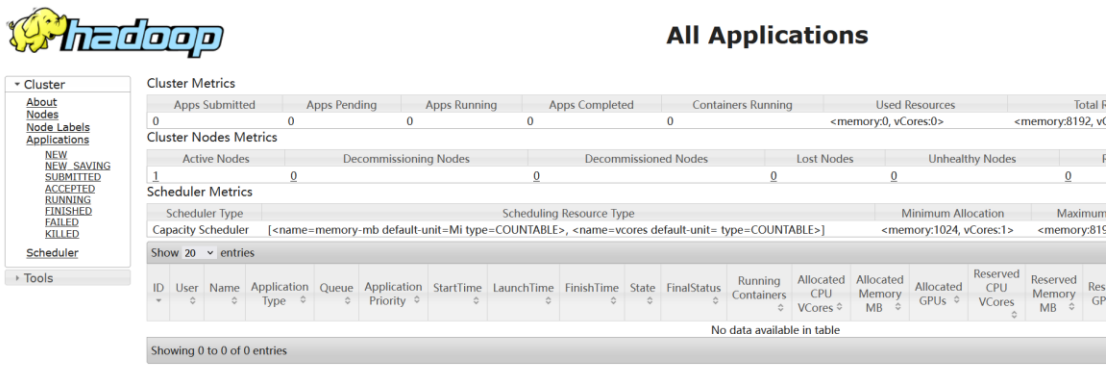
四、实验过程

单机伪分布式部署

启动 yarn 服务，使用 jps 查看进程

```
ubuntu@10-23-3-42:~$ jps
2912 JobHistoryServer
2960 Jps
2417 ResourceManager
2549 NodeManager
2057 DataNode
1883 NameNode
2268 SecondaryNameNode
ubuntu@10-23-3-42:~$
```

访问 yarn web 界面



以默认方式提交 jar 包

```
2024-06-18 13:41:27,416 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - Dep
loying cluster, current state ACCEPTED
2024-06-18 13:41:37,669 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - YAR
N application has been deployed successfully.
2024-06-18 13:41:37,670 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - Fou
nd Web Interface 10-23-142-76:8081 of application 'application_1718688756786_0002'.
Job has been submitted with JobID 5e7d45db86fff35295d3cc3acaa89dd2
```

监听 8888 端口

```
ubuntu@10-23-3-42:~$ nc -lk 8888
Hello Per-Job Mode
Hello Default
Hello Flink on yarn
```

Taskmanager.out 文件输出了结果：

```
ubuntu@10-23-3-42:~/hadoop-2.10.1/logs/userlogs/application_1718688756786_0002/container_1718688756
786_0002_01_000002$ cat taskmanager.out
Hello : 1
Mode : 1
Per-Job : 1
Hello : 1
Default : 1
Hello : 1
yarn : 1
on : 1
flink : 1
```

查看进程，可以看到 CliFrontend、YarnTaskExecutorRunner、YarnJobClusterEntrypoint 进程

```
2912 JobHistoryServer
4081 Jps
2417 ResourceManager
3699 YarnJobClusterEntrypoint
3828 YarnTaskExecutorRunner
2549 NodeManager
2057 DataNode
1883 NameNode
2268 SecondaryNameNode
3196 CliFrontend
```

Detached 方式提交

```
ubuntu@10-23-3-42:~$ ~/flink-1.12.1/bin/flink run -t yarn-per-job --detached ~/flink-1.12.1/example
s/streaming/SocketWindowWordCount.jar --port 8888
```

运行完毕截图

```
2024-06-18 14:05:27,026 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - Dep
loying cluster, current state ACCEPTED
2024-06-18 14:05:31,845 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - YAR
N application has been deployed successfully.
2024-06-18 14:05:31,846 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - The
 Flink YARN session cluster has been started in detached mode. In order to stop Flink gracefully, u
se the following command:
$ echo "stop" | ./bin/yarn-session.sh -id application_1718690565863_0001
If this should not be possible, then you can also kill Flink via YARN's web interface or via:
$ yarn application -kill application_1718690565863_0001
Note that killing Flink might not clean up all job artifacts and temporary files.
2024-06-18 14:05:31,846 INFO  org.apache.flink.yarn.YarnClusterDescriptor                  [] - Fou
nd Web Interface 10-23-142-76:8081 of application 'application_1718690565863_0001'.
Job has been submitted with JobID e0805393a8c032d41a410c482f046205
```

监听端口。输入 Detached

```
ubuntu@10-23-3-42:~$ nc -lk 8888
Hello Per-Job Mode
Hello Detached Hello flink on yarn
```

输出结果如下：

```
ubuntu@10-23-3-42:~/hadoop-2.10.1/logs/userlogs/application_1718692730590_0001/container_17186927305
90_0001_01_000002$ cat taskmanager.out
Hello : 1
Mode : 1
Per-Job : 1
Hello : 2
yarn : 1
on : 1
flink : 1
Detached : 1
ubuntu@10-23-3-42:~/hadoop-2.10.1/logs/userlogs/application_1718692730590_0001/container_17186927305
90_0001_01_000002$
```

此时的进程

```
2912 JobHistoryServer
4081 Jps
2417 ResourceManager
3699 YarnJobClusterEntrypoint
3828 YarnTaskExecutorRunner
2549 NodeManager
2057 DataNode
1883 NameNode
2268 SecondaryNameNode
3196 CliFrontend
```

（因为两个提交方式之间我重启过主机，因此这里只有一个日志记录）

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB
application_1718692730590_0001	ubuntu	Flink per-job cluster	Apache Flink	default	0	Tue Jun 18 14:45:07 +0800 2024	Tue Jun 18 14:45:09 +0800 2024	N/A	RUNNING	UNDEFINED	2	2	4096

Showing 1 to 1 of 1 entries

五、实验总结

通过本次实验，我们成功地在 Yarn 平台上部署了 Flink，并在单机伪分布式和分布式环境下分别以 Attached 和 Detached 方式提交了 Flink 应用程序。实验结果表明 Standalone 模式和 Yarn 模式有显著区别：Standalone 模式下，Flink 集群独立运行，不依赖其他资源管理平台，适用于资源较为固定的环境；而 Yarn 模式下，Flink 作业由 Yarn 管理，具有更好的资源调度和管理能力，适用于动态资源需求的环境。通过这两种模式的比较，我们理解了它们各自的优缺点和适用场景。

在实验过程中，我们还体会到 Attached 和 Detached 提交方式的不同。Attached 方式提交应用程序后，客户端会等待作业执行完成，并在终端输出结果，适用于需要实时监控作业执行状态的场景。Detached 方式则在提交应用程序后，客户端立即返回，作业在后台继续执行，适用于不需要实时监控作业状态的场景。通过对比这两种提交方式，我们更深入地理解了 Flink 在不同运行模式下的行为特点，为后续在实际生产环境中部署和运行 Flink 应用程序提供了宝贵的经验和参考。