

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：Flink 部署 学号：10185501402 上机实践日期：2021/5/6
上机实践编号：Lab13 组号：Group5 上机实践时间：

Part 1

实验目的

- (1) 学习 Flink 的部署，简单实用 Scala Shell
- (2) 查看 Flink 的运行日志，体会与其他分布式应用程序运行过程中日志的区别
- (3) 通过系统部署理解体系结构，体会流计算系统与批处理系统之间的区别

Part 2

实验任务

- (1) 完成 Flink 的单机伪分布式部署以及分布式部署。
- (2) 两种部署方式下分别以默认和托管模式运行示例程序。

Part 3

使用环境

- (1) 操作系统：Ubuntu 18.04
- (2) JDK 版本：1.8
- (3) Flink 版本：1.2.1

Part 4

实验过程

Section 1

单机集中式部署

先做一些准备工作，安装 Flink 后启动 Scala Shell，如图 1 和图 2 所示。

```

dase-local@10-24-21-70: ~/softwares/flink-1.12.1/bin
flink-1.12.1/bin/zookeeper.sh
flink-1.12.1/bin/kubernetes-session.sh
flink-1.12.1/bin/find-flink-home.sh
flink-1.12.1/bin/start-cluster.sh
flink-1.12.1/bin/mesos-taskmanager.sh
flink-1.12.1/bin/taskmanager.sh
dase-local@10-24-21-70:~/softwares$ ls
flink-1.12.1          hadoop-2.10.1-new
flink-1.12.1-bin-scala_2.11.tgz  hadoop-2.10.1.tar.gz
hadoop-1.2.1          jdk-8u171-linux-x64.tar.gz
hadoop-1.2.1.tar.gz   scala-2.11.12.tgz
hadoop-2.10.1          spark-2.4.7-bin-without-hadoop.tgz
dase-local@10-24-21-70:~/softwares$ cd flink-1.12.1/
dase-local@10-24-21-70:~/softwares/flink-1.12.1$ ls
LICENSE README.txt conf lib log plugins
NOTICE bin examples licenses opt
dase-local@10-24-21-70:~/softwares$ cd bin
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$ ls
hash-java-utils.jar  kubernetes-session.sh  start-cluster.sh
config.sh            mesos-appmaster-job.sh  start-scala-shell.sh
find-flink-home.sh  mesos-appmaster.sh    start-zookeeper-quorum.sh
flink               mesos-jobmanager.sh  stop-cluster.sh
flink-console.sh    mesos-taskmanager.sh  stop-zookeeper-quorum.sh
flink-daemon.sh     pyflink-shell.sh    taskmanager.sh
historyserver.sh   sql-client.sh      yarn-session.sh
jobmanager.sh       standalone-job.sh  zookeeper.sh
dase-local@10-24-21-70:~/softwares$ flink-1.12.1/bin$ start-scala-shell.sh local
Starting Flink Shell:

Starting local Flink cluster (host: localhost, port: 8081).

```

图 1: 安装 Flink，并启动 Scala Shell

启动后可以输入 scala 代码

```

FLINK - SCALA - SHELL

NOTE: Use the prebound Execution Environments and Table Environment to implement batch or streaming programs.

Batch - Use the 'benv' and 'btenv' variable
* val dataSet = benv.readTextFile("/path/to/data")
* dataSet.writeAsText("/path/to/output")
* benv.execute("My batch program")
*
* val batchTable = btenv.fromDataSet(dataSet)
* btenv.registerTable("tableName", batchTable)
* val result = btenv.sqlQuery("SELECT * FROM tableName").collect
HINT: You can use print() on a DataSet to print the contents or collect() a SQL query result back to the shell.

Streaming - Use the 'senv' and 'stenv' variable
* val dataStream = senv.fromElements(1, 2, 3, 4)
* dataStream.countWindowAll(2).sum().print()
*
* val streamTable = stenv.fromDataStream(dataStream, 'num')
* val resultTable = streamTable.select('num).where('num % 2 === 1 )
* resultTable.toAppendStream[Row].print()
* senv.execute("My streaming program")
HINT: You can only print a DataStream to the shell in local mode.

scala> 

```

图 2: Scala Shell

Scala 词频统计

```

scala> val textstreaming = senv.fromElements("a a b b c")
textstreaming: org.apache.flink.streaming.api.scala.DataStream[String] = org.apache.flink.streaming.api.scala.DataStream@6d3a56ea

```

图 3: 输入 Scala 代码

```

scala> countsstreaming.print()
res1: org.apache.flink.streaming.api.datastream.DataStreamSink[(String, Int)] = org.apache.flink.streaming.api.datastream.DataStreamSink@667c2066

scala> senv.execute()
(0,1)
(0,2)
(0,1)
(0,2)
(0,1)
res2: org.apache.flink.api.common.JobExecutionResult =
Program execution finished
Job with JobID cf2d3367a394e0101540bc284579b1ca has finished.
Job Runtime: 460 ms

```

图 4: 输入 Scala 代码，查看词频统计结果

然后，通过提交 jar 包运行 DataStream 程序

首先启动 Netcat 监听，随后在该端口输入流数据

```
● ● ●  ~%64 dase-local@10-24-21-70: ~
Last login: Thu May  6 15:36:24 on ttys002
sunqushisunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:36:40 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ nc -lk 8888
```

图 5: 启动 Netcat 服务

如图 6 所示，我们在开始监听后提交一个 Flink 词频统计任务，并输入一些文本用于 word count 任务，然后在另起的终端中的 scala shell 中可以看到返回的结果

```
● ● ●  ~%5 dase-local@10-24-21-70: ~
Last login: Thu May  6 15:44:58 on ttys003
sunqushisunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:45:01 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ ls
Chinese-Cloze-RC-master  Music      Videos      master.zip  softwares
Desktop                Pictures   examples.desktop myApp    spark-2.4.7
Documents               Public     hadoop-2.10.1  output    tmp-1.2.1
Downloads              Templates  input      pd
dase-local@10-24-21-70:~$ cd ~softwares/flink-1.12.1/bin/
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin $ ./flink run ~/softwares/flink-1.12.1/examples/streaming/SocketWindowWordCount.jar --port 8888
Job has been submitted with JobID 22a0550f1d0c57f98ec5e6cfef0208894
[]

HINT: You can only print a DataStream to the shell in local mode.

scala> hello : 1
flink : 1
hello : 1
dase : 1
I : 1
systems : 1
distributed : 1
love : 1
[]

● ● ●  ~%4 dase-local@10-24-21-70: ~
Last login: Thu May  6 15:36:24 on ttys002
sunqushisunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:36:40 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ nc -lk 8888
hello flink
hello dase
^C
dase-local@10-24-21-70:~$ nc -lk 8888
hello flink
hello dase
I love distributed systems
[]
```

图 6: 词频统计及其结果

执行期间通过 jps 查看进程状态

```
● ● ●  ~%5 dase-local@10-24-21-70: ~
Last login: Thu May  6 15:44:58 on ttys003
sunqushisunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:45:01 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ ls
Chinese-Cloze-RC-master  Music      Videos      master.zip  softwares
Desktop                Pictures   examples.desktop myApp    spark-2.4.7
Documents               Public     hadoop-2.10.1  output    tmp-1.2.1
Downloads              Templates  input      pd
dase-local@10-24-21-70:~$ cd ~softwares/flink-1.12.1/bin/
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin $ ./flink run ~/softwares/flink-1.12.1/examples/streaming/SocketWindowWordCount.jar --port 8888
Job has been submitted with JobID 22a0550f1d0c57f98ec5e6cfef0208894
[]

HINT: You can only print a DataStream to the shell in local mode.

scala> hello : 1
flink : 1
hello : 1
dase : 1
I : 1
systems : 1
distributed : 1
love : 1
[]

● ● ●  ~%4 dase-local@10-24-21-70: ~
Last login: Thu May  6 15:36:24 on ttys002
sunqushisunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:36:40 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ nc -lk 8888
hello flink
3254 Jps
3494 Frontend
3119 FlinkShell
dase-local@10-24-21-70:~$
```

图 7: 任务运行期间进程状态

上述提交方式为默认模式，随后在 detached 模式下提交这个任务

```
dase-local@10-24-21-70:~$ ./flink run -d ~/softwares/flink-1.12.1/bin/flink run -d ~/softwares/flink-1.12.1/examples/streaming/SocketWindowWordCount.jar --port 8888
Job has been submitted with JobID 0151bcdca4bbb02e77a89a91cb5ab7ff
dase-local@10-24-21-70:~$
```

图 8: Detached 模式提交任务

华东师范大学数据科学与工程学院学生实验报告

同样地，输入文本用于词频统计

图 9: Detached 模式提交任务, 输入文本

可以通过 Flink UI 来查看这个已经运行结束的词频统计任务

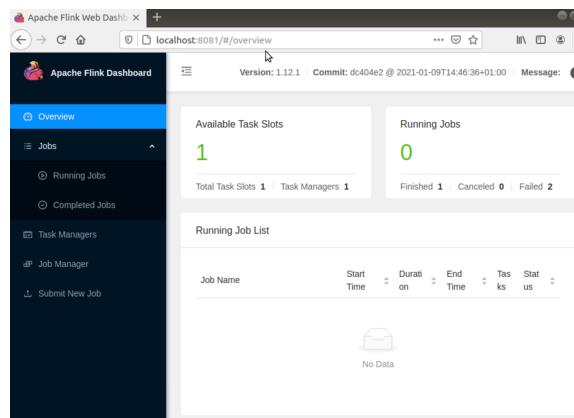


图 10: Flink UI

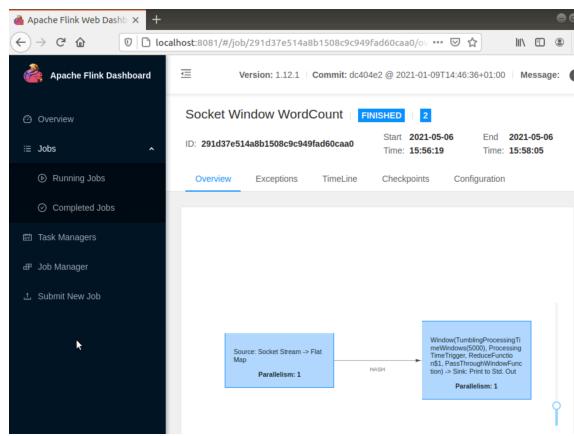


图 11: Flink UI Cont'd

Section 2

单机伪分布式部署

接下来进行单机伪分布式部署，先修改配置文件

```
dase-local@10-24-21-70:~/softwares/flink-1.12.1/conf
The RPC port where the JobManager is reachable.
jobmanager.rpc.port: 6123

The total process memory size for the JobManager.
Note this accounts for all memory usage within the JobManager process, including JVM metaspace and other overhead.
jobmanager.memory.process.size: 1600m

The total process memory size for the TaskManager.
Note this accounts for all memory usage within the TaskManager process, including JVM metaspace and other overhead.
taskmanager.memory.process.size: 1728m

To exclude JVM metaspace and overhead, please, use total Flink memory size instead of 'taskmanager.memory.process.size'.
It is not recommended to set both 'taskmanager.memory.process.size' and Flink memory.
taskmanager.memory.flink.size: 1280m

The number of task slots that each TaskManager offers. Each slot runs one parallel pipeline.
taskmanager.numberofTaskSlots: 2

The parallelism used for programs that did not specify other parallelism.
parallelism.default: 1

the default file system scheme and authority.
```

图 12: 修改配置文件

随后启动 Flink，观察启动后的进程

```
dase-local@10-24-21-70:~/softwares/flink-1.12.1$ cd bin
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$ start-cluster.sh
Starting cluster.
Starting standalonesession daemon on host 10-24-21-70.
Starting taskexecutor daemon on host 10-24-21-70.
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$ jps
6130 StandaloneSessionClusterEntryPoint
6501 Jps
6405 TaskManagerRunner
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$
```

图 13: 启动 Flink 服务后存在的进程

最后可以通过访问 8081 端口查看 Flink 程序的运行情况，可以通过 Web UI 来 cancel 掉应用程序。

| Job Name | Start Time | Duration | End Time | Tasks | Status |
|----------|------------|----------|----------|-------|--------|
| | | | | | |

图 14: 查看 Flink UI

```
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$ start-scala-shell.sh remote localhost 8081
Starting Flink Shell:
Connecting to Flink cluster (host: localhost, port: 8081).
```

图 15: 单机伪分布式下启动 Flink Shell

在 Flink Shell 中输入代码

华东师范大学数据科学与工程学院学生实验报告

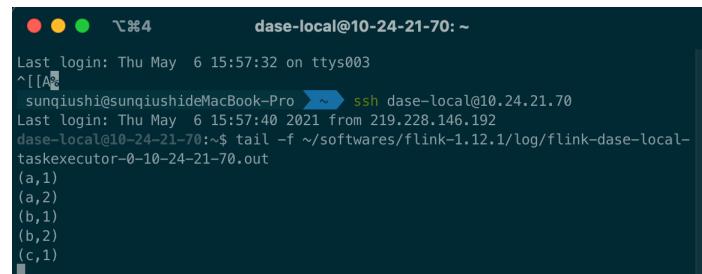
```
scala> val textstreaming = senv.fromElements("a a b b c")
textstreaming: org.apache.flink.streaming.api.scala.DataStream[String] = org.apache.flink.streaming.api.scala.DataStream@799206ad
scala> val countsstreaming = textstreaming.flatMap(_.toLowerCase.split("\\W+")).map{(_,_)}.keyBy(0).sum(1)
warning: 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@37f687c1
scala> countsstreaming.print()
res0: org.apache.flink.streaming.api.datastream.DataStreamSink[(String, Int)] = org.apache.flink.streaming.api.datastream.DataStreamSink@517594dd
scala> senv.execute()
res1: org.apache.flink.api.common.JobExecutionResult =
Program execution finished
Job with JobID 16a3450bd0a4d26ac391ade7e28aeabc has finished.
Job Runtime: 1836 ms
scala>
```

图 16: 输入 Scala 代码

词频统计的结果输出在日志文件中

```
dase-local@10-24-21-70:~/softwares/flink-1.12.1$ cd log
dase-local@10-24-21-70:~/softwares/flink-1.12.1/log$ ls
flink-dase-local-client-10-24-21-70.log          flink-dase-local-standalonesession-0-10-24-21-70.log
flink-dase-local-scala-shell-local-10-24-21-70.log   flink-dase-local-standalonesession-0-10-24-21-70.out
flink-dase-local-scala-shell-local-10-24-21-70.log.1  flink-dase-local-taskexecutor-0-10-24-21-70.log
flink-dase-local-scala-shell-local-10-24-21-70.log.2  flink-dase-local-taskexecutor-0-10-24-21-70.out
flink-dase-local-scala-shell-local-10-24-21-70.log.3
dase-local@10-24-21-70:~/softwares/flink-1.12.1/log$
```

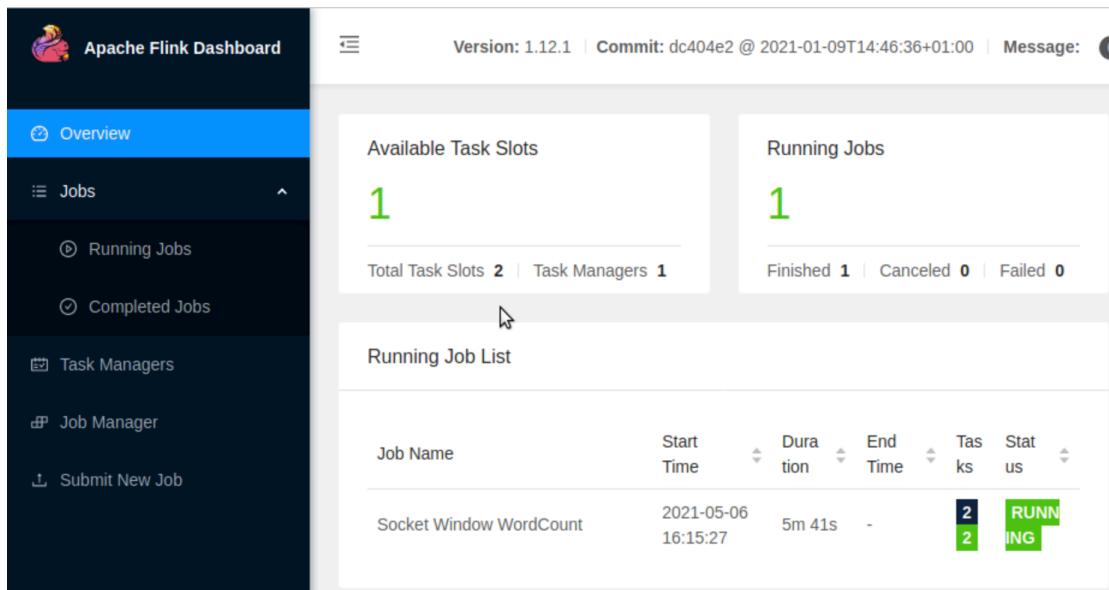
图 17: 日志文件



```
dase-local@10-24-21-70: ~
Last login: Thu May  6 15:57:32 on ttys003
^[[A
sunqushu@sunqushideMacBook-Pro ~ ssh dase-local@10.24.21.70
Last login: Thu May  6 15:57:40 2021 from 219.228.146.192
dase-local@10-24-21-70:~$ tail -f ~/softwares/flink-1.12.1/log/flink-dase-local-
taskexecutor-0-10-24-21-70.out
(a,1)
(a,2)
(b,1)
(b,2)
(c,1)
```

图 18: 访问文件查看词频统计结果

Detached 模式提交后，所有终端如下所示



The screenshot shows the Apache Flink Dashboard interface. On the left, a sidebar menu includes 'Overview', 'Jobs' (with 'Running Jobs' and 'Completed Jobs' sub-options), 'Task Managers', 'Job Manager', and 'Submit New Job'. The main content area displays the following information:

- Available Task Slots:** 1
- Running Jobs:** 1
- Running Job List:** A table showing the job 'Socket Window WordCount' with the following details:

| Job Name | Start Time | Duration | End Time | Tasks | Status |
|-------------------------|---------------------|----------|----------|-------|---------|
| Socket Window WordCount | 2021-05-06 16:15:27 | 5m 41s | - | 2 | RUNNING |

图 19: Flink UI 查看 Job 状态

华东师范大学数据科学与工程学院学生实验报告

此时启动的 Terminal 状况如下所示



```
love : 1  
dase-local@10-24-21-70:~$ tail ~/softwares/flink-1.12.1/log/flink-dase-local-tas  
keexecutor-0-10-24-21-70.out  
dase : 1  
love : 1  
i : 1  
science : 1  
data : 1  
love : 1  
this : 1  
mode : 1  
detached : 1  
is : 1  
dase-local@10-24-21-70:~$  
  
dase-local@10-24-21-70:~/softwares/flink-1.12.1/bin$ nc -lk 8888  
i love distributed systems  
i love dase  
i love data science  
this is detached mode  
dase-local@10-24-21-70:~$  
dase-local@10-24-21-70:~$ jps  
6130 StandaloneSessionClusterEntryPoint  
7443 Jps  
6405 TaskManagerRunner  
dase-local@10-24-21-70:~$
```

图 20: Terminals

在 Flink UI 中，可以看到更详细的程序执行细节，如 Parallelism（并行度）以及所分配的资源

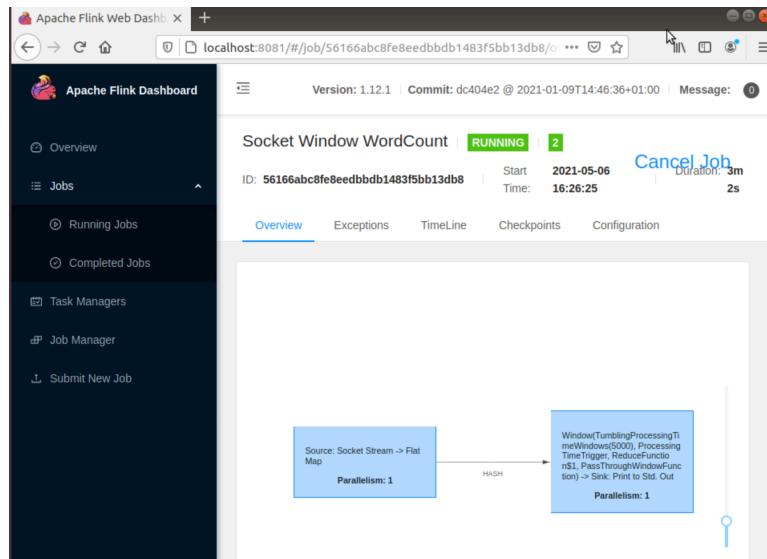
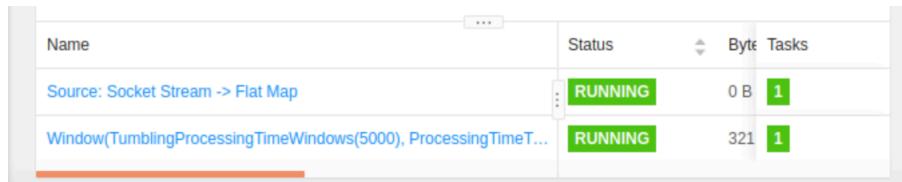


图 21: Flink UI



| Name | Status | Byte | Tasks |
|--|---------|------|-------|
| Source: Socket Stream -> Flat Map | RUNNING | 0 B | 1 |
| Window(TumblingProcessingTimeWindows(5000), ProcessingTimeT... | RUNNING | 321 | 1 |

图 22: 查看 Job 的状态

Section 3

分布式部署

接下来开始分布式部署，一台机器 ecnu01 作为主节点，两台机器 ecnu02, ecnu03 作为从节点，一台机器 ecnu04 作为客户端。

同样先实现 4 台机器间的免密登录并且，将 dase-local 中的压缩包转移到 dase-dis 再进行对从节点和客户端的分发

由于是第一次进行 Flink 实验，需要先解压文件并且修改配置，如图 23 和图 24 所示

```
[dase-dis@ecnu01:~]$ scp dase-dis@localhost:~/softwares/flink-1.12.1-bin-scala_2
11.tgz ~/
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:4Hoq6ENrOBhwftG8mLqH3cW123tuxKKVXWyr59aw04.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
dase-dis@localhost's password:
flink-1.12.1-bin-scala_2.11.tgz          100% 319MB 142.8MB/s  00:02
[dase-dis@ecnu01:~]$ ls
Desktop  examples_desktop      input    spark-2.4.7
Documents flink-1.12.1-bin-scala_2.11.tgz  Music   Templates
Download  hadoop-2.10.1           Pictures  Videos
Downloads hadoop-2.10.1.tar.gz     Public
[dase-dis@ecnu01:~]$ tar -zvxf flink-1.12.1-bin-scala_2.11.tgz
flink-1.12.1/
flink-1.12.1/LICENSE
flink-1.12.1/bin/
flink-1.12.1/licenses/
flink-1.12.1/plugins/
...[redacted]
```

图 23: 主节点拷贝到当前用户目录下解压

```
# by specifying the --host <hostname> parameter of the bin/jobmanager.sh executable.
# In high availability mode, if you use the bin/start-cluster.sh script and set up
# the conf/masters file, this will be taken care of automatically. Yarn/Mesos
# automatically configure the host name based on the hostname of the node where
# JobManager runs.

jobmanager.rpc.address: ecnu01
# The RPC port where the JobManager is reachable.

jobmanager.rpc.port: 6123

# The total process memory size for the JobManager.
# Note this accounts for all memory usage within the JobManager process, including
# JVM metaspace and other overhead.

jobmanager.memory.process.size: 1600m
```

图 24: 修改 Flink 配置文件

主节点将 ecnu02 和 ecnu03 添加到 workers 中，并且拷贝相应文件到其他节点上，如图 25 和图 26 所示

图 25: 添加从节点 ecnu02,ecnu03

图 26: 将配置好的文件拷贝到其他节点

主节点上启动 Flink，然后使用 jps 查看进程状态

图 27: 主节点启动 Flink

图 28: 主节点使用 jps 查看进程状态

随后在从节点上查看进程状态，确认已启动。

```
dase-dis@10-23-204-186:~$ jps
3176 Jps
3100 TaskManagerRunner
```

图 29: 从节点使用 jps 查看进程状态

随后可以查看 Web UI

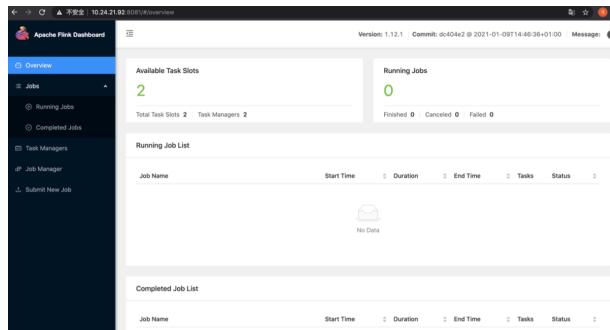


图 30: Web UI 查看信息

接下来开始由客户端提交任务

在客户端执行如下操作，运行 DataStream 程序

首先是通过 jar 包运行 DataStream 程序

默认模式提交

```
dase-dis@10-24-21-156:~$ ~/flink-1.12.1/bin/flink run ~/flink-1.12.1//examples/streaming/SocketWindowWordCount.jar --hostname ecnu04 --port 8888
Job has been submitted with JobID a7776b8c3c24f32fa5ff3a66a469ee53
[...]
dase-dis@10-24-21-156:~/flink-1.12.1/log$ nc -lk 8888
[...]
```

图 31: 客户端提交任务并开启 Netcat 监听

```
dase-dis@10-24-21-156:~/flink-1.12.1/log$ nc -lk 8888
i love distributed computing systems
i love data science and engineering
no one loves china better than me
[...]
```

图 32: 输入用于 WordCount 的文本

在客户端使用 jps 命令查看进程状态

```
sunqiuishi@sunqiuishiMacBook-Pro ~ ssh dase-dis@10.24.21.156
Last login: Tue May 11 19:34:34 2021 from 10.24.21.156
dase-dis@10-24-21-156:~$ jps
4526 Jps
4367 CliFrontend
dase-dis@10-24-21-156:~$
```

图 33: 客户端使用 jps 查看进程状态

主节点可以访问 Web UI 查看进程信息

访问 ecnu01:8081，选中对应的正在运行中的应用程序，进入到应用程序详情

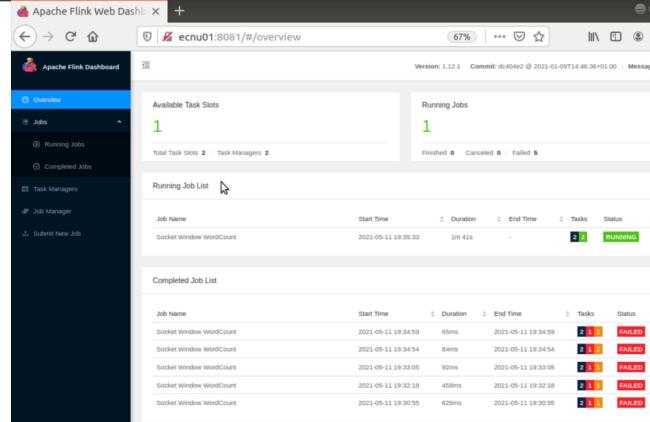


图 34: Web UI 查看信息

(这里前面有几个失败的任务，是因为我提交的时候把命令打错了导致的，排查出错误后运行正常)

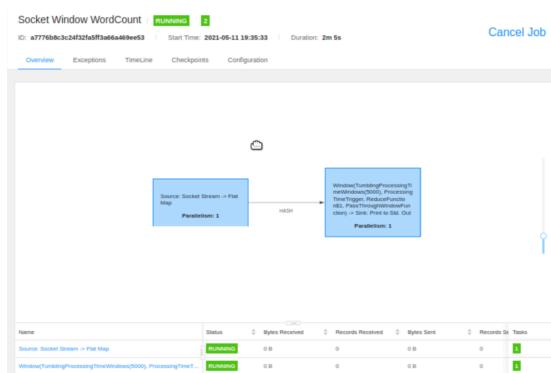


图 35: Web UI 查看信息 Socket Window

可以在 Web UI 上看到这个任务执行在哪个节点上

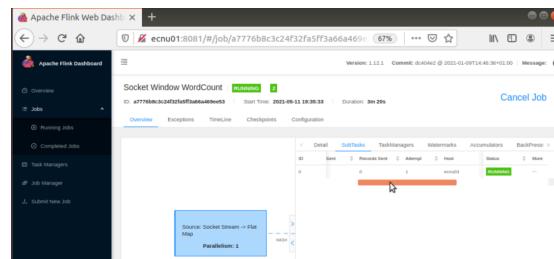


图 36: Web UI 查看信息 Socket Window

可见是在从节点 ecnu03 上执行，查看 ecnu03 上的日志如下

```
dase-dis@ecnu03:~$ cat flink-1.12.1/log/flink-dase-dis-taskexecutor-0-ecnu03.out
i : 1
systems : 1
computing : 1
distributed : 1
love : 1
i : 1
engineering : 1
and : 1
science : 1
data : 1
love : 1
no : 1
me : 1
than : 1
better : 1
china : 1
loves : 1
one : 1
```

图 37: 日志文件查看词频统计结果

Detached 模式提交

我们接下来执行 Detached 模式下提交一个任务，同样启动 Netcat 监听，并输入文本

```
dase-dis@10-24-21-156:~/flink-1.12.1/bin$ flink run -d ~/flink-1.12.1//examples/streaming/SocketWindowWordCount.jar --hostname ecnu04 --port 8888
Job has been submitted with JobID 6ab35cea45084f0a068eea0814e39dec
dase-dis@10-24-21-156:~$
```

图 38: Detached 模式下运行

```
dase-dis@10-24-21-156:~/flink-1.12.1/log$ nc -lk 8888
i love flink
flink is cool
dase is cool
```

图 39: 输入用于 WordCount 的文本

同样我们通过访问 Web UI 确定其运行在哪个从节点

| ID | Records Sent | Attempt | Host | Status |
|----|--------------|---------|--------|---------|
| 0 | 9 | 1 | ecnu02 | RUNNING |

图 40: 主节点通过 Web UI 查看

在 ecnu02 的日志文件里可以看到词频统计文件

```
dase-dis@ecnu02:~/flink-1.12.1/log$ tail ./flink-dase-dis-taskexecutor-0-ecnu02.out
out
i : 1
cool : 1
is : 1
flink : 2
love : 1
dase : 1
cool : 1
is : 1
```

图 41: 日志文件查看词频统计结果

主节点通过 Web UI 查看 flink 程序运行信息，可以实时查看 Flink 进程

华东师范大学数据科学与工程学院学生实验报告

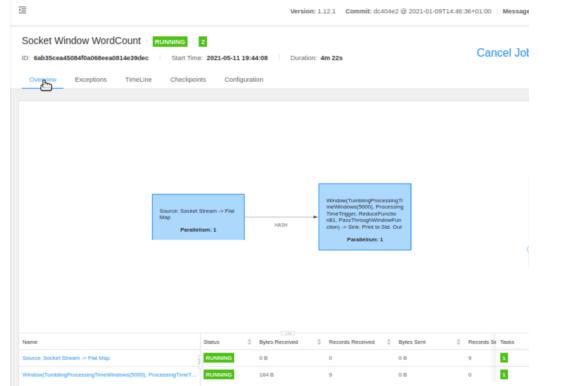


图 42: Flink UI

之行结束后使用 cancel 命令退出

```
dase-dis@10-24-21-156:~$ ~/flink-1.12.1/bin/flink list
Waiting for response...
-----  
Running/Restarting Jobs -----
11.05.2021 19:44:08 : 6ab35cea45084f0a068eea0814e39dec : Socket Window WordCount (RUNNING)

No scheduled jobs.
dase-dis@10-24-21-156:~$ ~/flink-1.12.1/bin/flink cancel 6ab35cea45084f0a068eea0814e39dec
 Cancelling job 6ab35cea45084f0a068eea0814e39dec.
Cancelled job 6ab35cea45084f0a068eea0814e39dec.
dase-dis@10-24-21-156:~$
```

图 43: 使用 cancel 命令退出

最后分别查看主节点、从节点和客户端的应用运行日志。

```
[flink@cn001:~/Flink-1.12.1]$ tail flink-dose-dis-standalone-session-0-cn001.log
05-11 19:23:43.288 INFO org.apache.flink.runtime.dispatcher.DispatcherRestEndpoint [] - Web frontend listening at http://cn001:8081.
05-11 19:23:43.233 INFO org.apache.flink.runtime.rpc.akka.AkkaRpcService [] - Starting RPC endpoint for org.apache.flink.runtime.resourcemanager.StandaloneResourceManager at akka://flink/rpc/resourcecanager_0.
05-11 19:23:43.258 INFO org.apache.flink.runtime.dispatcher.runner.SessionDispatcherLeaderProcess [] - Start SessionDispatcher leader process.
05-11 19:23:43.264 INFO org.apache.flink.runtime.resourcemanager.StandaloneResourceManager [] - ResourceManager akka.tcp://flink@cn001:6123/user/rpc/resourcecanager_0 was granted leadership with token 00000000000000000000000000000000.
05-11 19:23:43.263 INFO org.apache.flink.runtime.dispatcher.runner.SessionDispatcherLeaderProcess [] - Recover all persisted job graphs.
05-11 19:23:43.261 INFO org.apache.flink.runtime.dispatcher.runner.SessionDispatcherLeaderProcess [] - Successfully recovered 0 persisted job graphs.
05-11 19:23:43.271 INFO org.apache.flink.runtime.resourcemanager.slotmanager.SlotManagerImpl [] - Starting the SlotManager.
05-11 19:23:43.275 INFO org.apache.flink.runtime.rpc.akka.AkkaRpcService [] - Starting RPC endpoint for org.apache.flink.runtime.dispatcher.StandaloneDispatcher at akka://flink/user/rpc/rtcher_1.
05-11 19:23:44.446 INFO org.apache.flink.runtime.resourcemanager.StandaloneResourceManager [] - Registering TaskManager with ResourceID 10.24.21.39:18547->0@ac2 (akka.tcp://flink@10.24.21.39:18547/u
05-11 19:23:46.463 INFO org.apache.flink.runtime.resourcemanager.StandaloneResourceManager [] - Registering TaskManager with ResourceID 10.23.204.186:27267->0@f726 (akka.tcp://flink@10.23.204.186:272
05-11 19:23:46.463 INFO org.apache.flink.runtime.resourcemanager.StandaloneResourceManager [] - Registering TaskManager with ResourceID 10.23.204.186:27267->0@ResourceManager
[flink@cn001:~/Flink-1.12.1]$ log$
```

图 44: 主节点查看日志

```
dase-dis@10-23-204-186:~/flink-1.12.1/log$ tail flink-dase-dis-taskexecutor-0-ec  
nu03.log  
2021-05-11 19:23:46,169 INFO org.apache.flink.runtime.io.network.NettyClient  
[...] - Successful initialization (took 51 ms).  
2021-05-11 19:23:46,173 INFO org.apache.flink.runtime.io.network.NettyServer  
[...] - Transport type 'auto': using EPOLL.  
2021-05-11 19:23:46,204 INFO org.apache.flink.runtime.io.network.NettyServer  
[...] - Successful initialization (took 34 ms). Listening on SocketAddress  
/0:0:0:0:0:0:0:0:0:22525.  
2021-05-11 19:23:46,205 INFO org.apache.flink.runtime.taskexecutor.KvStateService  
[...] - Starting the kvstate service and its components.  
2021-05-11 19:23:46,220 INFO org.apache.flink.runtime.rpc.akka.AkkaRpcService  
[...] - Starting RPC endpoint for org.apache.flink.runtime.taskexecutor.  
TaskExecutor at akka://flink/user/rpc/taskmanager_0.
```

图 45: 从节点查看日志

华东师范大学数据科学与工程学院学生实验报告

图 46: 客户端查看日志

最后通过 Web UI 查看历史记录

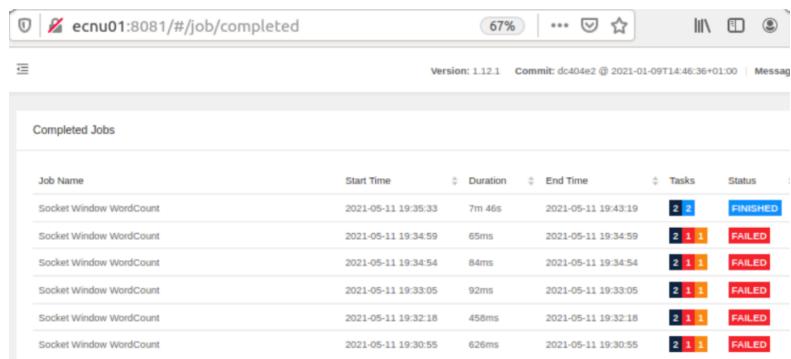


图 47: 通过 Web UI 查看历史记录

结束实验，主节点关闭服务

```
Stopping taskexecutor daemon on host ecnu01.
dase-dis@ecnu01:~/flink-1.12.1/log$ ~/hadoop-2.10.1/sbin/stop-dfs.sh
Stopping namenodes on [ecnu01]
The authenticity of host 'ecnu01 (10.24.21.92)' can't be established.
ECDSA key fingerprint is SHA256:4Hoq6EqNroBhwFTG8mLqHJcw123TuXKkVXWyr59aw04.
Are you sure you want to continue connecting (yes/no)? yes
ecnu01: Warning: Permanently added 'ecnu01,10.24.21.92' (ECDSA) to the list of known hosts.
ecnu01: no namenode to stop
ecnu02: no datanode to stop
ecnu03: no datanode to stop
Stopping secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:4Hoq6EqNroBhwFTG8mLqHJcw123TuXKkVXWyr59aw04.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: no secondarynamenode to stop
dase-dis@ecnu01:~/flink-1.12.1/bin$ ./flink-1.12.1/bin/stop-cluster.sh
Stopping taskexecutor daemon (pid: 5062) on host ecnu02.
Stopping taskexecutor daemon (pid: 4778) on host ecnu03.
Stopping standalonesession daemon (pid: 5609) on host ecnu01.
dase-dis@ecnu01:~/flink-1.12.1/log$ jps
6528 Jps
dase-dis@ecnu01:~/flink-1.12.1/log$
```

图 48: 主节点关闭服务

Part 5

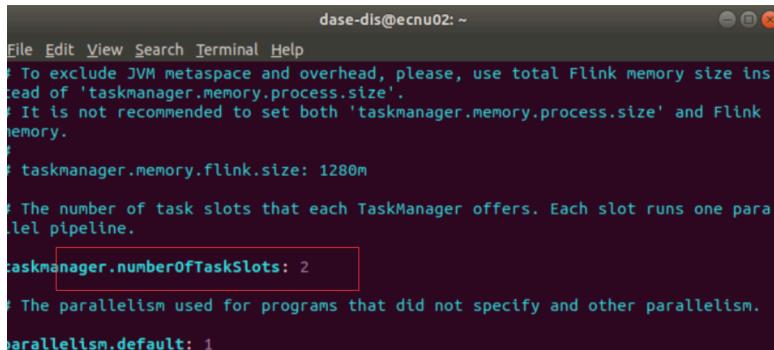
思考题

Section 1

在部署 flink 时，如何指定每个 TaskManager 中的 slot 数量？

在每个从节点的配置文件中，与伪分布式部署下相同，可以修改 `numberOfTaskSlots` 这一参数对当前从节点生效。

我们对从节点 ecnu02 的配置文件修改如下，将 Slot 的数量由 1 改为 2



```
dase-dis@ecnu02: ~
File Edit View Search Terminal Help
# To exclude JVM metaspace and overhead, please, use total Flink memory size instead of 'taskmanager.memory.process.size'.
# It is not recommended to set both 'taskmanager.memory.process.size' and Flink memory.
#
# taskmanager.memory.flink.size: 1280M
#
# The number of task slots that each TaskManager offers. Each slot runs one parallel pipeline.
taskmanager.numberOfTaskSlots: 2
#
# The parallelism used for programs that did not specify any other parallelism.
parallelism.default: 1
```

图 49: 思考题 1-1

重启服务后，我们可以看到 Slot 数量由 2 变为 3，可以在如下截图中看到：

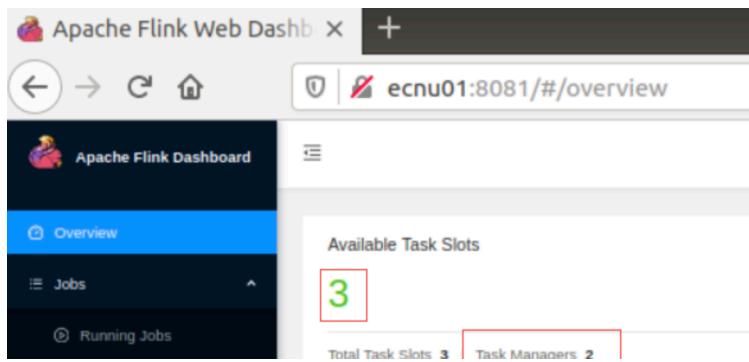


图 50: 思考题 1-2

Section 2

一个 Flink 流计算应用程序何时停止，如何停止？请通过与 Spark 批处理应用程序进行对比来阐述。

Flink 应用程序的停止与否只能由用户控制，停止的方式包括关闭 Natcat 服务、使用 Web UI 停止和在命令行通过 JobID 停止（在单机模式下使用了停止 Netcat 服务的方式，在分布式模式下使用了命令行通过 cancel 掉 Job 停止）。而 Spark 应用程序会在完成任务后自动停止。其区别来自于：

- Flink 程序接受我们认为“无界”的流数据，Flink 计算任务会长期驻留在计算节点，并且持续地更新。因此，程序无法确定何时输出结束，因此不会主动停止。
- Spark 接受有界静态数据，确定有限的完成时间，Spark 会主动结束运行完成的程序。

Part 6

思考与探索

Slot 的概念

在前面的思考题中修改了 slot 的数量，但是对这个抽象概念并没有细致的了解，现在仔细研究一下 TaskManager 和 slot 的关系，以及它们如何利用资源。

每个 Flink TaskManager 在集群中提供 slot，如图 51 所示。slot 的数量通常与每个 TaskManager 的可用 CPU 内核数成比例。一般情况下 slot 数是你每个 TaskManager 的 CPU 的核数，简而言之 slot 代表的是 taskmanager 基于它所拥有的计算资源的并发执行能力。

Slot 代表一个可用线程，该线程的内存分配固定（对 CPU 不隔离）。默认情况下，Flink 允许子任务共享 Slot，即使它们是不同 task 的 subtask，只要它们来自相同的 job，就可以共享。这种共享方式可以更有效利用资源。

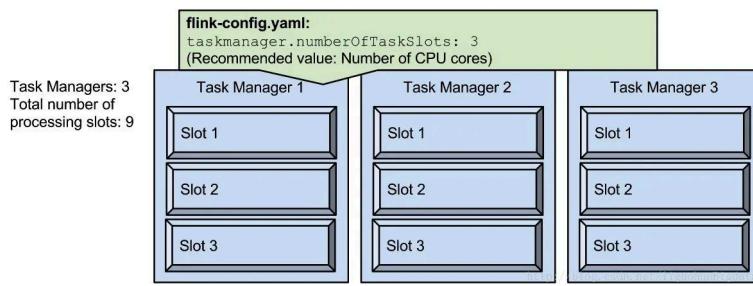


图 51: Flink-Taskmanager

参考资料: [Flink-对 parallelism 和 slot 的理解](#)

Part 7

参考资料

- (1) 阿里云 Flink 部署解读: [Flink 原理与实现：架构和拓扑概览](#)
- (2) Netcat 入门: [nc 命令用法举例](#)
- (3) Apache Flink 文档: [Deployment & Operations](#)
- (4) Slot 与并行度概念: [Flink-对 parallelism 和 slot 的理解](#)