

Flink集群部署

Standalone模式

(1) 修改 flink/conf/flink-conf.yaml 文件

```
# JobManager runs.  
jobmanager.rpc.address: hadoop102  
  
# The RPC port where the JobManager is reachable.
```

(2) 修改 /conf/slaves文件

```
hadoop103  
hadoop104
```

听说以后master/slaves都不能用了，你懂我的意思吧

(3) 分发给另外两台机器

```
[zhutian@hadoop102 module]$ xsync flink-1.7.2/
```

(4) 启动

```
[zhutian@hadoop102 flink-1.7.2]$ bin/start-cluster.sh  
Starting cluster.  
Starting standalone session daemon on host hadoop102.  
Starting taskexecutor daemon on host hadoop103.  
Starting taskexecutor daemon on host hadoop104.  
[zhutian@hadoop102 flink-1.7.2]$ j  
----- hadoop102 -----  
3536 StandaloneSessionClusterEntrypoint  
3599 Jps  
----- hadoop103 -----  
3384 Jps  
3327 TaskManagerRunner  
----- hadoop104 -----  
3330 TaskManagerRunner  
3384 Jps  
[zhutian@hadoop102 flink-1.7.2]$
```

(5) 访问Web界面

The screenshot shows the Apache Flink Dashboard web interface. The browser address bar displays 'hadoop102:8081/#/overview'. The interface includes a sidebar with navigation links: Overview, Running Jobs, Completed Jobs, Task Managers, Job Manager, and Submit new Job. The main content area is titled 'Overview' and shows the following information:

- Version: 1.7.2, Commit: cebaf
- Task Managers: 2
- Task Slots: 2
- Available Task Slots: 2
- Total Jobs: 0 Running, 0 Finished, 0 Canceled, 0 Failed
- Running Jobs table with columns: Start Time, End Time, Duration, Job Name, Job ID, Tasks, Status

(6) 任务提交到集群

- 准备好数据文件

```
how are you  
fine thank you  
and you  
i am fine too thank you
```

- 数据文件分发到每台taskmanager中(由于读取数据是从本地磁盘读取，实际任务会被分发到taskmanager的机器中，所以要把目标文件分发。)

```
xsync data.txt
```

- 在集群执行程序

```
/flink run -c com.zhutian.wc.StreamWordCount -p 2 FlinkTutorial-1.0-SNAPSHOT-jar-with-dependencies.jar --host localhost-port 7777
```

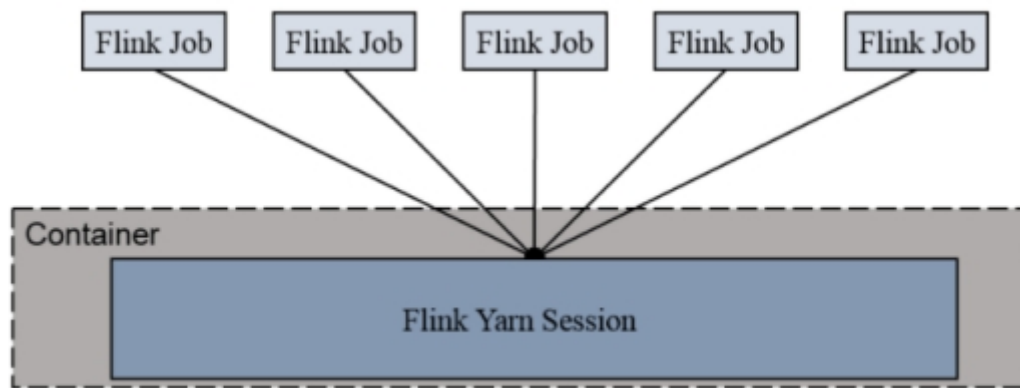
- 然后就可以去Web控制台查看结果了

也可以通过Web界面提交任务哦

Yarn模式

以Yarn模式部署Flink任务时，要求Flink是有Hadoop支持的版本，Hadoop环境需要保证版本在2.2以上，并且集群中安装有HDFS服务。

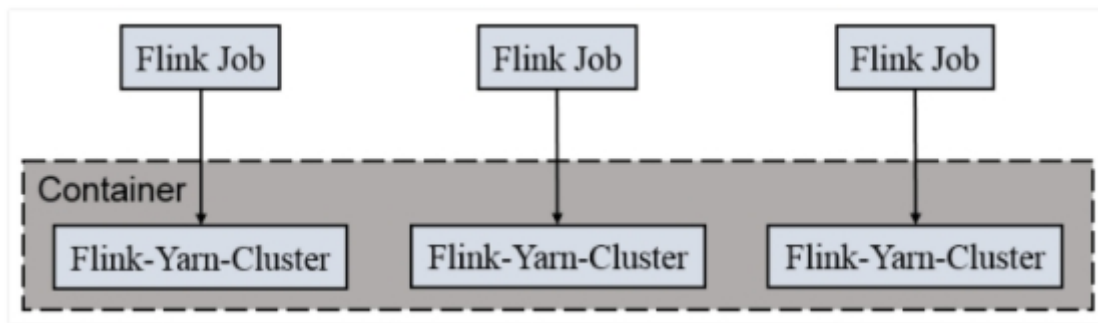
1.Session-cluster模式



Session-Cluster模式需要先启动集群，然后再提交作业，接着会向yarn申请一块空间后，**资源永远保持不变**。如果资源满了，下一个作业就无法提交，只能等到yarn中的其中一个作业执行完成后，释放了资源，下个作业才会正常提交。**所有作业共享Dispatcher和ResourceManager**；共享资源；适合规模小执行时间短的作业。

在yarn中初始化一个flink集群，开辟指定的资源，以后提交任务都向这里提交。这个flink集群会常驻在yarn集群中，除非手工停止。

2.Per-Job-Cluster模式



一个Job会对应一个集群，每提交一个作业会根据自身的情况，都会单独向yarn申请资源，直到作业执行完成，一个作业的失败与否并不会影响下一个作业的正常提交和运行。独享Dispatcher和ResourceManager，按需接受资源申请；适合规模大长时间运行的作业。

每次提交都会创建一个新的flink集群，任务之间互相独立，互不影响，方便管理。任务执行完成之后创建的集群也会消失。

SessionCluster部署

(1) 启动Hadoop集群

startHadoop.sh和我都是自己写的脚本，【有问题都可以私聊我WX: focusbigdata，或者关注我的公众号: FocusBigData，注意大小写】

```
[zhutian@hadoop102 flink-1.7.2]$ startHadoop.sh
Starting namenodes on [hadoop102]
hadoop102: starting namenode, logging to /opt/module/hadoop-2.7.2/logs/hadoop-zhutian-namenode-hadoop102.out
hadoop102: starting datanode, logging to /opt/module/hadoop-2.7.2/logs/hadoop-zhutian-datanode-hadoop102.out
hadoop104: starting datanode, logging to /opt/module/hadoop-2.7.2/logs/hadoop-zhutian-datanode-hadoop104.out
hadoop103: starting datanode, logging to /opt/module/hadoop-2.7.2/logs/hadoop-zhutian-datanode-hadoop103.out
Starting secondary namenodes [hadoop104]
hadoop104: starting secondarynamenode, logging to /opt/module/hadoop-2.7.2/logs/hadoop-zhutian-secondarynamenode-hadoop104.out
starting yarn daemons
starting resourcemanager, logging to /opt/module/hadoop-2.7.2/logs/yarn-zhutian-resourcemanager-hadoop103.out
hadoop103: starting nodemanager, logging to /opt/module/hadoop-2.7.2/logs/yarn-zhutian-nodemanager-hadoop103.out
hadoop104: starting nodemanager, logging to /opt/module/hadoop-2.7.2/logs/yarn-zhutian-nodemanager-hadoop104.out
hadoop102: starting nodemanager, logging to /opt/module/hadoop-2.7.2/logs/yarn-zhutian-nodemanager-hadoop102.out
starting historyserver, logging to /opt/module/hadoop-2.7.2/logs/mapred-zhutian-historyserver-hadoop102.out
[zhutian@hadoop102 flink-1.7.2]$ j
----- hadoop102 -----
5011 Jps
4809 NodeManager
4876 JobHistoryServer
4397 NameNode
4541 DataNode
----- hadoop103 -----
4499 Jps
4295 NodeManager
4056 ResourceManager
3917 DataNode
----- hadoop104 -----
3922 DataNode
4039 SecondaryNameNode
4264 Jps
4127 NodeManager
```

(2) 启动yarn-session

```
[zhutian@hadoop102 flink-1.7.2]$ bin/yarn-session.sh -n 2 -s 2 -jm 1024 -tm 1024 -nm test -d
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/hadoop/yarn/exceptions/YarnException
    at java.lang.Class.getDeclaredMethods0(Native Method)
    at java.lang.Class.privateGetDeclaredMethods(Class.java:2701)
    at java.lang.Class.privateGetMethodRecursive(Class.java:3048)
    at java.lang.Class.getMethod0(Class.java:3018)
    at java.lang.Class.getMethod(Class.java:1784)
    at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.java:544)
    at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:526)
Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.yarn.exceptions.YarnException
    at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:335)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    ... 7 more
```

报错了，有点慌，这个提示很猜想到少了依赖包，注意版本，拷贝到lib目录下即可

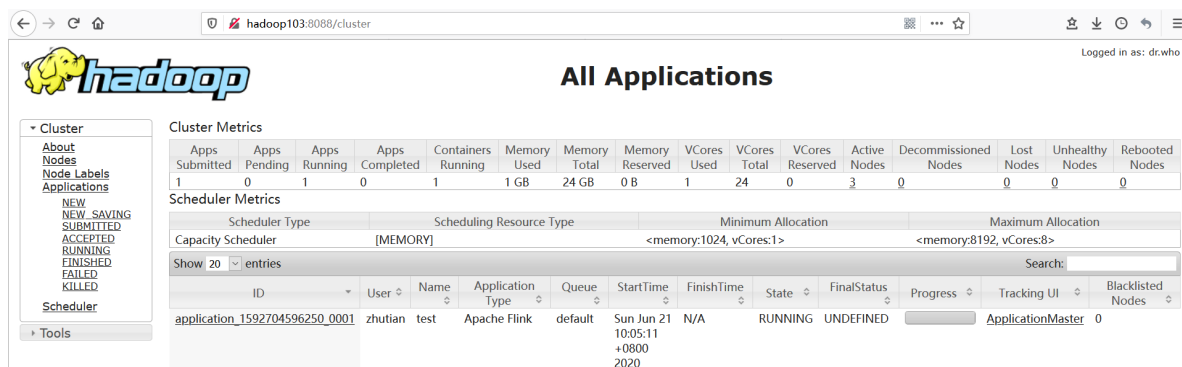
flink-shaded-hadoop2-uber-1.7.2.jar

```
[zhutian@hadoop102 flink-1.7.2]$ bin/yarn-session.sh -n 2 -s 2 -jm 1024 -tm 1024 -nm test -d
2020-06-21 10:05:06,239 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: jobmanager.replicat
ess, hadoop102
2020-06-21 10:05:06,240 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: jobmanager.replicat
, 6123
2020-06-21 10:05:06,240 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: jobmanager.heap
e, 1024m
2020-06-21 10:05:06,240 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: taskmanager.heap
ze, 1024m
2020-06-21 10:05:06,240 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: taskmanager.numberOfTaskSlots, 1
2020-06-21 10:05:06,241 INFO org.apache.flink.configuration.GlobalConfiguration - Loading configuration property: parallelism.default
```

(3) 执行任务

```
./flink run -c com.zhutian.wc.StreamWordCount FlinkTutorial-1.0-SNAPSHOT-jar-with-dependencies.jar --host localhost -port 7777
```

(4) 去yarn控制台查看任务状态



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	1	1 GB	24 GB	0 B	1	24	0	3	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Application List

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
application_1592704596250_0001	zhutian	test	Apache Flink	default	Sun Jun 21 10:05:11 +0800 2020	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster	0

(5) 只能手动取消flink集群

mode. In order to stop Flink on YARN, use the following command or a YARN web interface to stop it:
`yarn application -kill application_1592704596250_0001`
 Please also note that the temporary files of the YARN session in the home directory will not be removed.

集群启动的时候，会有上面如何停止它的命令

Per-Job-Cluster部署

(1) 启动Hadoop集群

(2) 不启动yarn-session，直接执行job

```
/flink run -m yarn-cluster -c com.zhutian.wc.StreamWordCount FlinkTutorial-1.0-SNAPSHOT-jar-with-dependencies.jar --host localhost -port 7777
```

K8s部署

了解就行，不需要自己操作，公司运维都会替你搭好。容器化部署时目前业界很流行的一项技术，基于Docker镜像运行能够让用户更加方便地对应用进行管理和运维。容器管理工具中最为流行的就是Kubernetes (k8s)，而Flink也在最近的版本中支持了k8s部署模式。

(1) 搭建Kubernetes集群 (略)

(2) 配置各组件的yaml文件

在k8s上构建Flink Session Cluster，需要将Flink集群的组件对应的docker镜像分别在k8s上启动，包括JobManager、TaskManager、JobManagerService三个镜像服务。每个镜像服务都可以从中央镜像仓库中获取。

(3) 启动Flink Session Cluster

```
// 启动jobmanager-service 服务
kubectl create -f jobmanager-service.yaml
// 启动jobmanager-deployment服务
kubectl create -f jobmanager-deployment.yaml
// 启动taskmanager-deployment服务
kubectl create -f taskmanager-deployment.yaml
```

(4) 访问Flink UI页面

集群启动后，就可以通过JobManagerServices中配置的WebUI端口，用浏览器输入以下url来访问Flink UI页面了

```
http://{JobManagerHost:Port}/api/v1/namespaces/default/services/flink-
jobmanager:ui/proxy
```