

Flink HA 配置文档-V1.6.1

作者：徐葳

Flink Standalone 集群 HA 配置

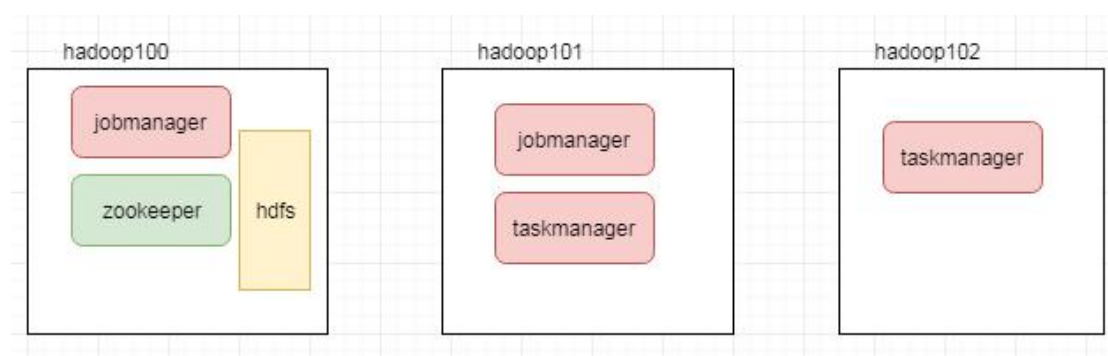
1. HA 集群环境规划

使用三台节点实现两主两从集群(由于笔记本性能限制，不能开启太多虚拟机，其实使用三台和四台机器在安装配置上没有本质区别)

Jobmanager: hadoop100 hadoop101 【一个 active，一个 standby】

Taskmanager: hadoop101 hadoop102

zookeeper: hadoop100 【建议使用外置 zk 集群，在这里我使用单节点 zk 来代替】



注意：

要启用 JobManager 高可用性，必须将高可用性模式设置为 zookeeper，配置一个 ZooKeeper quorum，并配置一个 masters 文件 存储所有 JobManager hostname 及其 Web UI 端口号。

Flink 利用 ZooKeeper 实现运行中的 JobManager 节点之间的分布式协调。ZooKeeper 是独立于 Flink 的服务，它通过领导选举制和轻量级状态一致性存储来提供高度可靠的分布式协调。

2. 开始配置+启动

集群内所有节点的配置都一样，所以先从第一台机器 hadoop100 开始配置

ssh hadoop100

#首先按照之前配置 standalone 的参数进行修改

vi conf/flink-conf.yaml

jobmanager.rpc.address: hadoop100

vi conf/slaves

hadoop101

hadoop102

然后修改配置 HA 需要的参数

vi conf/masters

```

hadoop100:8081
hadoop101:8081

vi conf/flink-conf.yaml
high-availability: zookeeper
high-availability.zookeeper.quorum: hadoop100:2181
# ZooKeeper 节点根目录，其下放置所有集群节点的 namespace
high-availability.zookeeper.path.root: /flink
# ZooKeeper 节点集群 id，其中放置了集群所需的所有协调数据
high-availability.cluster-id: /cluster_one
# 建议指定 hdfs 的全路径。如果某个 flink 节点没有配置 hdfs 的话，不指定全路径无法识别
# storageDir 存储了恢复一个 JobManager 所需的所有元数据。
high-availability.storageDir: hdfs://hadoop100:9000/flink/ha

# 把 hadoop100 节点上修改好配置的 flink 安装目录拷贝到其他节点
cd /data/soft/
scp -rq flink-1.6.1 hadoop101:/data/soft
scp -rq flink-1.6.1 hadoop102:/data/soft

# 【先启动 hadoop 服务】
sbin/start-all.sh
# 【先启动 zk 服务】
bin/zkServer.sh start
# 启动 flink standalone HA 集群，在 hadoop100 节点上启动如下命令
bin/start-cluster.sh

# 启动之后会显示如下日志信息
Starting HA cluster with 2 masters.
Starting standalone-session daemon on host hadoop100.
Starting standalone-session daemon on host hadoop101.
Starting taskexecutor daemon on host hadoop101.
Starting taskexecutor daemon on host hadoop102.

```

3. 验证 HA 集群进程

查看机器进程会发现如下情况【此处只列出 flink 自身的进程信息，不包含 zk，hadoop 进程信息】

```

登录 hadoop100 节点
执行 jps:
20159 StandaloneSessionClusterEntrypoint

登录 hadoop101 节点
执行 jps:
7795 StandaloneSessionClusterEntrypoint

```

8156 TaskManagerRunner

登录 hadoop102 节点

执行 jps:

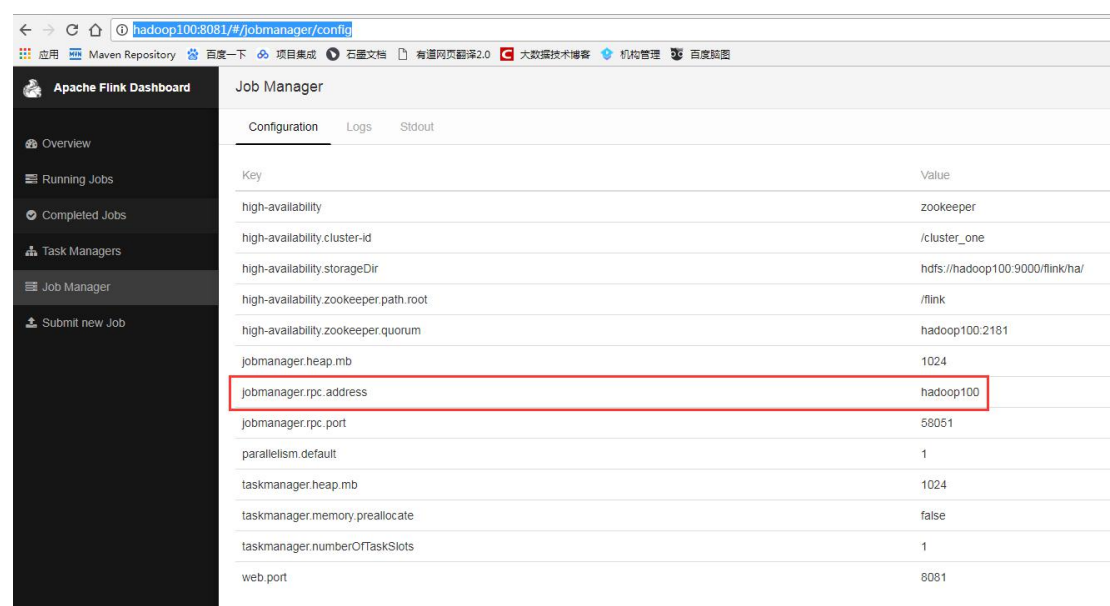
5046 TaskManagerRunner

因为 jobmanager 节点都会启动 web 服务，也可以通过 web 界面进行验证

访问 <http://hadoop100:8081/#/jobmanager/config>

发现以下信息:

注意: 此时就算是访问 hadoop101:8081 也会跳转回 hadoop100:8081 因为现在 hadoop100 是 active 的 jobmanager。从下图中也可以看出, 点击 jobmanager 查看, 显示哪个节点, 就表示哪个节点现在是 active 的。



Key	Value
high-availability	zookeeper
high-availability.cluster-id	/cluster_one
high-availability.storageDir	hdfs://hadoop100:9000/flink/ha/
high-availability.zookeeper.path.root	/flink
high-availability.zookeeper.quorum	hadoop100:2181
jobmanager.heap.mb	1024
jobmanager.rpc.address	hadoop100
jobmanager.rpc.port	58051
parallelism.default	1
taskmanager.heap.mb	1024
taskmanager.memory.preallocate	false
taskmanager.numberOfTaskSlots	1
web.port	8081

4. 模拟 jobmanager 进程挂掉

现在 hadoop100 节点上的 jobmanager 是 active 的。我们手工把这个进程 kill 掉, 模拟进程挂掉的情况, 来验证 hadoop101 上的 standby 状态的 jobmanager 是否可以正常切换到 active。

ssh hadoop100

执行 jps:

20159 StandaloneSessionClusterEntrypoint

kill 20159

5. 验证 HA 切换

hadoop100 节点上的 jobmanager 进程被手工 kill 掉了, 然后 hadoop101 上的 jobmanager 会自动切换为 active, 中间需要有一个时间差, 稍微等一下

访问 <http://hadoop101:8081/#/jobmanager/config>

如果可以正常访问并且能看到 jobmanager 的信息变为 hadoop101, 则表示 jobmanager 节点切换成功

Key	Value
high-availability	zookeeper
high-availability.cluster-id	/cluster_one
high-availability.storageDir	hdfs://hadoop100:9000/flink/ha/
high-availability.zookeeper.path.root	/flink
high-availability.zookeeper.quorum	hadoop100:2181
jobmanager.heap.mb	1024
jobmanager.rpc.address	hadoop101
jobmanager.rpc.port	45230
parallelism.default	1
taskmanager.heap.mb	1024
taskmanager.memory.preallocate	false
taskmanager.numberOfTaskSlots	1
web.port	8081

6. 重启之前 kill 掉的 jobmanager

进入到 hadoop100 机器

ssh hadoop100

执行下面命令启动 jobmanager

```
bin/jobmanager.sh start
```

启动成功之后，可以访问 <http://hadoop100:8081/#/jobmanager/config>

这个节点重启启动之后，就变为 standby 了。hadoop101 还是 active。

Key	Value
high-availability	zookeeper
high-availability.cluster-id	/cluster_one
high-availability.storageDir	hdfs://hadoop100:9000/flink/ha/
high-availability.zookeeper.path.root	/flink
high-availability.zookeeper.quorum	hadoop100:2181
jobmanager.heap.mb	1024
jobmanager.rpc.address	hadoop101
jobmanager.rpc.port	42626
parallelism.default	1
taskmanager.heap.mb	1024
taskmanager.memory.preallocate	false
taskmanager.numberOfTaskSlots	1
web.port	8081

Flink on yarn 集群 HA 配置

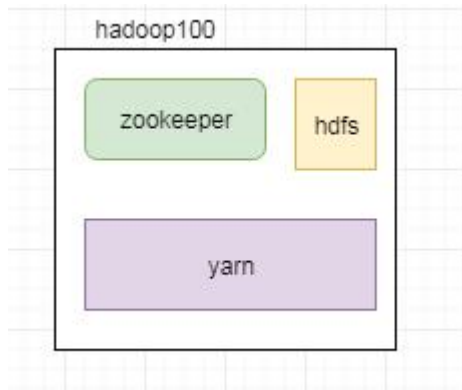
1. HA 集群环境规划

flink on yarn 的 HA 其实是利用 yarn 自己的恢复机制。

在这需要用到 zk，主要是因为虽然 flink-on-yarn cluster HA 依赖于 Yarn 自己的集群机制，但是 Flink Job 在恢复时，需要依赖检查点产生的快照，而这些快照虽然配置在 hdfs，但是其元数据信息保存在 zookeeper 中，所以我们还要配置 zookeeper 的信息

hadoop 搭建的集群，在 hadoop100，hadoop101，hadoop102 节点上面【flink on yarn 使用伪分布 hadoop 集群和真正分布式 hadoop 集群，在操作上没有区别】

zookeeper 服务也在 hadoop100 节点上



2. 开始配置+启动

主要在 hadoop100 这个节点上配置即可

首先需要修改 hadoop 中 yarn-site.xml 中的配置，设置提交应用程序的最大尝试次数

```
<property>
  <name>yarn.resourcemanager.am.max-attempts</name>
  <value>4</value>
  <description>
    The maximum number of application master execution attempts.
  </description>
</property>
```

把修改后的配置文件同步到 hadoop 集群的其他节点

```
scp -rq etc/hadoop/yarn-site.xml hadoop101:/data/soft/hadoop-2.7.5/etc/hadoop/
scp -rq etc/hadoop/yarn-site.xml hadoop102:/data/soft/hadoop-2.7.5/etc/hadoop/
```

然后修改 flink 部分相关配置

可以解压一份新的 flink-1.6.1 安装包

```
tar -zxvf flink-1.6.1-bin-hadoop27-scala_2.11.tgz
```

修改配置文件【标红的目录名称建议和 standalone HA 中的配置区分开】

```
vi conf/flink-conf.yaml
```

```
high-availability: zookeeper
```

```
high-availability.zookeeper.quorum: hadoop100:2181
```

```
high-availability.storageDir: hdfs://hadoop100:9000/flink/ha-yarn
```

```
high-availability.zookeeper.path.root: /flink-yarn
```

```
yarn.application-attempts: 10
```

3. 启动 flink on yarn, 测试 HA

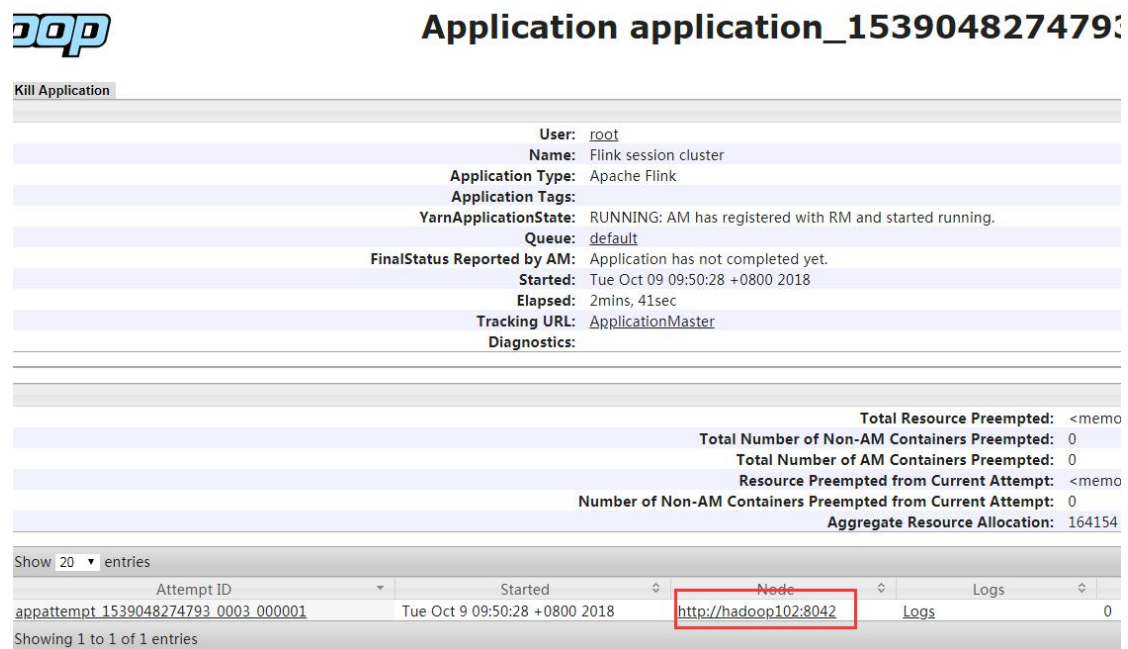
先启动 hadoop100 上的 zookeeper 和 hadoop

```
bin/zkServer.sh start
sbin/start-all.sh
```

在 hadoop100 上启动 Flink 集群

```
cd /data/soft/flink-1.6.1
bin/yarn-session.sh -n 2
```

到 resoucemanager 的 web 界面上查看对应的 flink 集群在哪个节点上



The screenshot shows the Hadoop ResourceManager web interface. At the top, there's a logo and the title 'Application application_1539048274793'. Below this, there's a 'Kill Application' button. The main section displays application details in a table-like format:

User:	root
Name:	Flink session cluster
Application Type:	Apache Flink
Application Tags:	
YarnApplicationState:	RUNNING: AM has registered with RM and started running.
Queue:	default
FinalStatus Reported by AM:	Application has not completed yet.
Started:	Tue Oct 9 09:50:28 +0800 2018
Elapsed:	2mins, 41sec
Tracking URL:	ApplicationMaster
Diagnostics:	

Below the details, there's a section for resource preemption statistics:

Total Resource Preempted:	<memo
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memo
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	164154

At the bottom, there's a table showing application attempts:

Attempt ID	Started	Node	Logs
appattempt_1539048274793_0003_000001	Tue Oct 9 09:50:28 +0800 2018	http://hadoop102:8042	Logs

The 'Node' column for the first attempt is highlighted with a red box, showing the URL 'http://hadoop102:8042'.

jobmanager 进程就在对应的节点的(YarnSessionClusterEntrypoint)进程里面

所以想要测试 jobmanager 的 HA 情况, 只需要拿 YarnSessionClusterEntrypoint 这个进程进行测试即可。

执行下面命令手工模拟 kill 掉 jobmanager (YarnSessionClusterEntrypoint)、

ssh hadoop102

jps

5325 YarnSessionClusterEntrypoint

kill 5325

然后去 yarn 的 web 界面进行查看:



All Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
1	0	1	0	1	1 GB	16 GB	0 B	1	16

Scheduler Metrics

Scheduler Type		Scheduling Resource Type				Minimum Allocation	
Capacity Scheduler		[MEMORY]				<memory:1024, vCores:1>	
Show 20 entries							
ID	User	Name	Application Type	Queue	StartTime	FinishTime	State
application_1539048274793_0003	root	Flink session cluster	Apache Flink	default	Tue Oct 9 09:50:28 +0800 2018	N/A	RUNNING
Showing 1 to 1 of 1 entries							

发现这个程序的 AttemptId 变为 00002 了

Kill Application	
User: root	
Name: Flink session cluster	
Application Type: Apache Flink	
Application Tags:	
YarnApplicationState: RUNNING: AM has registered with RM and started running.	
Queue: default	
FinalStatus Reported by AM: Application has not completed yet.	
Started: Tue Oct 9 09:50:28 +0800 2018	
Elapsed: 8mins, 39sec	
Tracking URL: ApplicationMaster	
Diagnostics:	
Total Resource Preempted: <me	
Total Number of Non-AM Containers Preempted: 0	
Total Number of AM Containers Preempted: 0	
Resource Preempted from Current Attempt: <me	
Number of Non-AM Containers Preempted from Current Attempt: 0	
Aggregate Resource Allocation: 5292	
Show 20 entries	
Attempt ID	Started
appattempt_1539048274793_0003_000002	Tue Oct 9 09:56:38 +0800 2018
appattempt_1539048274793_0003_000001	Tue Oct 9 09:50:28 +0800 2018
Showing 1 to 2 of 2 entries	

如果想查看 jobmanager 的 webui 界面可以点击下面链接:



All Applications

Cluster

About NodesNode LabelsApplicationsNEWNEW SAVINGSUBMITTEDACCEPTEDRUNNINGFINISHEDFAILEDKILLED

Scheduler

Tools

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
1	0	1	0	1	1 GB	16 GB	0 B	1	16	0	2	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation							
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>							
Show 20 entries										
ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1539048274793_0003	root	Flink session cluster	Apache Flink	default	Tue Oct 9 09:50:28 +0800 2018	N/A	RUNNING	UNDEFINED		ApplicationMaster

Showing 1 to 1 of 1 entries

注意: 针对上面配置文件中的一些配置参数的详细介绍信息可以参考此文章

<https://blog.csdn.net/xu470438000/article/details/79633824>

但是需要注意一点, 此链接文章中使用的 flink 版本是 1.4.2。我们本课程中使用的 flink 版本是 1.6.1, 这两个版本中的一些参数名称会有细微不同。但是参数的含义基本没有什么变化。