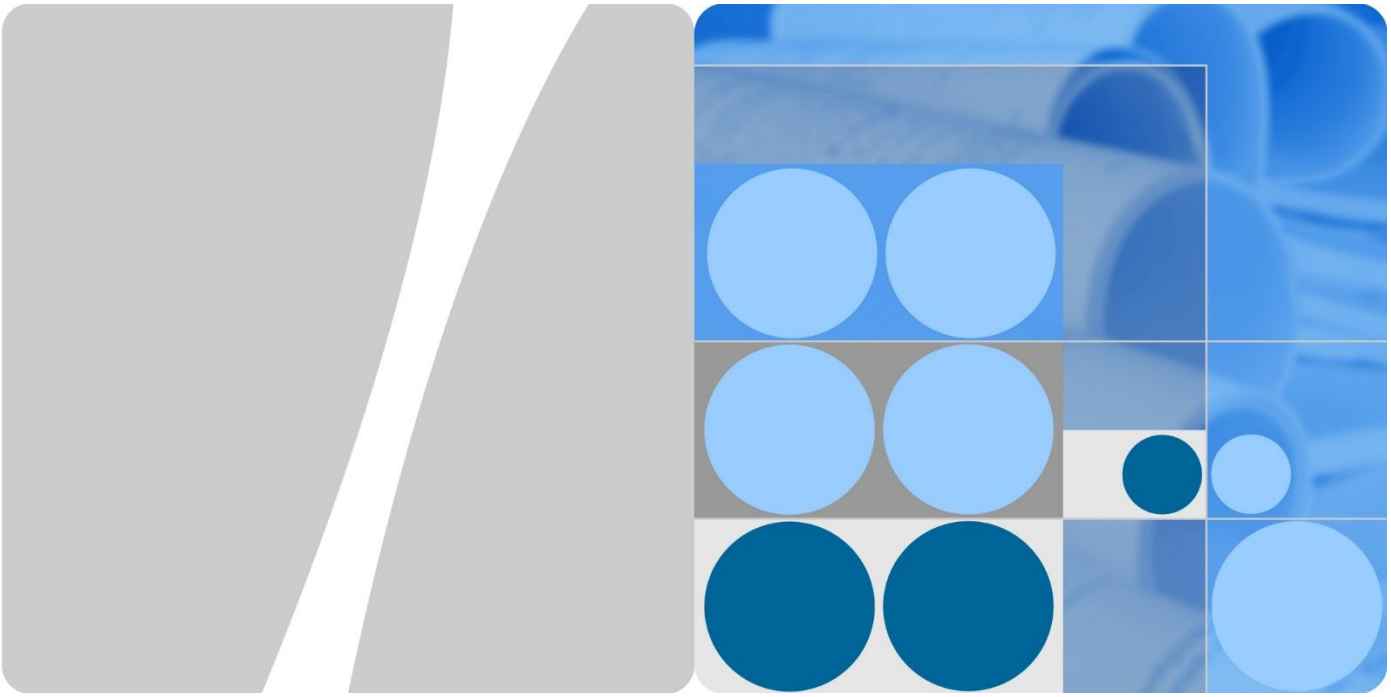


Yarn 常见问题维护手册 V1.0

Yarn 常见问题维护手册 V1.0

文档版本 01

发布日期 2016-03-31

华为技术有限公司

HUAWEI

商标声明


和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： http://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

YARN

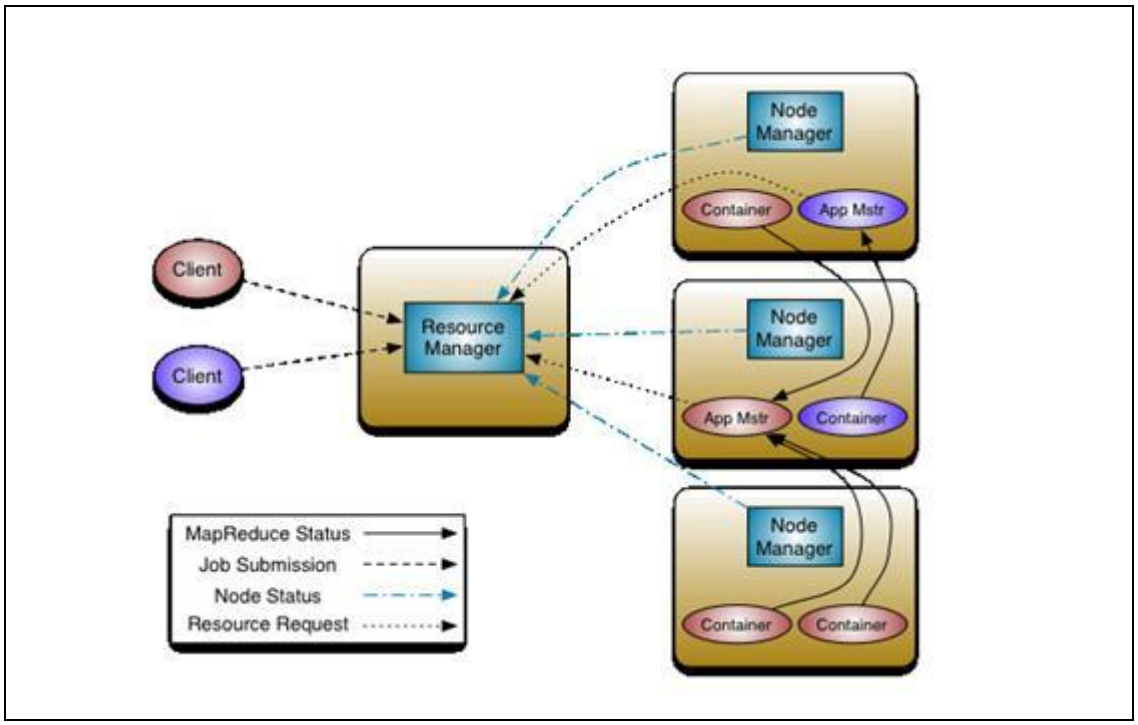
1、基本概念

本章简要介绍 Yarn 基本概念以及 Yarn 日志存放路径，以及常见问题的日志查看方法。

【概述】

Yarn 是 Hadoop 2.x 版本引入的一种资源管理框架。它将统一管理集群的资源，与任务调度分离，可以在其上层搭配不同的计算框架，如 MapReduce、Spark、Tez 等。

Yarn 结构图如下：



Yarn 的主要特点有：

- 将资源管理与任务调度分离，可以在其上层选择不同的分布式计算框架。
- 开放资源调度器的接口，可以选择使用不同的调度策略。

Yarn 各组件说明：

ResourceManager	Yarn 的主服务。主要完成资源申请、分配，任务的提交等调度任务。
NodeManager	以 container 的形式管理本节点上的资源，向 RM 保持心跳连接。

【日志概述】

Yarn 日志

Yarn 日志包括两类，运行日志、审计日志；

日志路径：

运行日志: /var/log/Bigdata/yarn

审计日志: /var/log/Bigdata/audit/yarn

各日志功能如下表：

运行日志	yarn-<SSH_USER>-<process_name>-<hostname>.log	YARN 组件日志，记录 YARN 组件运行时候所产生的大部分日志。
	yarn-<SSH_USER>-<process_name>-<hostname>.out	YARN 运行环境信息日志。
	yarn.log	YARN 客户端操作日志。
	prestart-Detail.log	服务的预启动日志。
	startDetail.log	服务的启动日志。
	stopDetail.log	服务的停止日志。
	hadoop.log	Hadoop 客户端操作日志。
审计日志	yarn-audit-<process_name>.log	YARN 操作审计日志。

MapReduce 日志

日志路径：

MapReduce 任务在**运行的过程**中，Map 和 Reduce 任务所运行的 Container 日志位于该任务所在节点的以下路径：

/srv/BigData/hadoop/data1/nm/containerlogs/<application_id>/<container_id>/

Container 日志	syslog	MapReduce 程序中使用 log4j 打印的日志信息。
	stdout	MapReduce 程序中使用标准输出流打印的信息。
	stderr	MapReduce 程序中使用标准错误流打印的信息。

MapReduce 任务**运行完成**后，相关日志会被拷贝至 HDFS 的以下路径，并且上述的三个子日志被合并为同一个文件：

/tmp/logs/<user_name>/logs/<application_id>/<hostname_port>

各个日志里都存放了什么信息？

审计日志

对于 Yarn 来说，审计日志主要记录了某个用户从哪个 IP 向 Yarn（主要是 ResourceManager）提交的任务信息，以及 AppMaster 向 ResourceManager 的注册信息。

如下的 RM 审计日志，表示在 2016-03-22 10:12:21 用户 admin 向 RM 提交了一个 MapReduce 任务，客户端 IP 为 xxx.xxx.1.18。其中主要包括了任务提交，任务的 APPMaster 注册，任务的结果三条记录。

```
2016-03-22 10:12:21,056 | INFO  | IPC Server handler 13 on 26004 | USER=admin   IP=xxx.xxx.1.18 OPERATION=Submit Application Request   TARGET=ClientRMService  RESULT=SUCCESS  APPID=application_1458609629121_0001 | RMAuditLogger.java:91
2016-03-22 10:12:35,812 | INFO  | Socket Reader #1 for port 26002 | Auth successful for appattempt_1458609629121_0001_000001 (auth:SIMPLE) | Server.java:1334
2016-03-22 10:12:35,885 | INFO  | IPC Server handler 8 on 26002 | USER=admin   IP=xxx.xxx.1.18 OPERATION=Register App Master   TARGET=ApplicationMasterService RESULT=SUCCESS  APPID=application_1458609629121_0001      APPATTEMPTID=appattempt_1458609629121_0001_000001 | RMAuditLogger.java:91
2016-03-22 10:13:01,233 | INFO  | Socket Reader #1 for port 26004 | Auth successful for hive/hadoop.hadoop.com@HADOOP.COM (auth:KERBEROS) | Server.java:1334
2016-03-22 10:13:05,722 | INFO  | Socket Reader #1 for port 26004 | Auth successful for mapred/hadoop.hadoop.com@HADOOP.COM (auth:KERBEROS) | Server.java:1334
2016-03-22 10:13:12,236 | INFO  | Socket Reader #1 for port 26004 | Auth successful for hive/hadoop.hadoop.com@HADOOP.COM (auth:KERBEROS) | Server.java:1334
2016-03-22 10:13:34,882 | INFO  | Socket Reader #1 for port 26004 | Auth successful for mapred/hadoop.hadoop.com@HADOOP.COM (auth:KERBEROS) | Server.java:1334
2016-03-22 10:13:52,133 | INFO  | AsyncDispatcher event handler | USER=admin   OPERATION=Application Finished - Succeeded   TARGET=RMAppManager     RESULT=SUCCESS  APPID=application_1458609629121_0001 | RMAuditLogger.java:91
```

审计日志就是对这些流程动作进行了一下记录，对 Yarn 的问题定位来说，较少来查看这些信息。

ResourceManager 运行日志

ResourceManager 负责接收客户端请求（MR/Spark 等任务），然后为任务分配第一个资源启动 APPMaster 并管理 APPMaster 的生命周期，接受来自 APPMaster 的资源申请。ResourceManager 运行日志中将记录这些信息。

以下为截取的 ResourceManager 日志，从内部状态机的角度展示了 RM 中 APP 的运行过程。

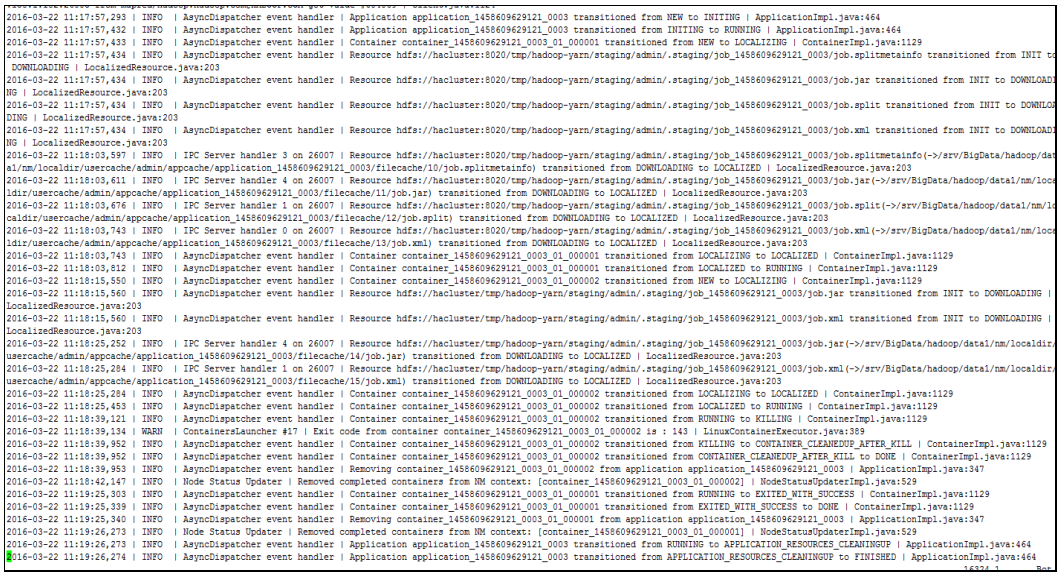
```
2016-03-22 11:09:14,059 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from NEW to NEW_SAVING | RMAppImpl.java:753
2016-03-22 11:09:14,072 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from NEW_SAVING to SUBMITTED | RMAppImpl.java:753
2016-03-22 11:09:14,074 | INFO  | ResourceManager Event Processor | Accepted application application_1458609629121_0002 from user: admin, in queue: default | CapacityScheduler.java:772
2016-03-22 11:09:14,074 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from SUBMITTED to ACCEPTED | RMAppImpl.java:753
2016-03-22 11:09:14,075 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from NEW to SUBMITTED | RMAppAttemptImpl.java:800
2016-03-22 11:09:14,075 | INFO  | ResourceManager Event Processor | Application application_1458609629121_0002 from user: admin activated in queue: default | LeafQueue.java:662
2016-03-22 11:09:14,075 | INFO  | ResourceManager Event Processor | Added Application Attempt appattempt_1458609629121_0002_000001 to scheduler from user admin in queue default | Capac
802
2016-03-22 11:09:14,078 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from SUBMITTED to SCHEDULED | RMAppAttemptImpl.java:800
2016-03-22 11:09:14,510 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000001 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:14,527 | INFO  | AsyncDispatcher event handler | container_1458609629121_0002_01_000001 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:14,528 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from SCHEDULED to ALLOCATED_SAVING | RMAppAttemptImpl.java:800
2016-03-22 11:09:14,546 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from ALLOCATED_SAVING to ALLOCATED | RMAppAttemptImpl.java:800
2016-03-22 11:09:14,583 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from ALLOCATED to LAUNCHED | RMAppAttemptImpl.java:800
2016-03-22 11:09:15,514 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000001 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:36,342 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from LAUNCHED to RUNNING | RMAppAttemptImpl.java:800
2016-03-22 11:09:36,342 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from ACCEPTED to RUNNING | RMAppImpl.java:753
2016-03-22 11:09:37,700 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000002 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:37,712 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000003 Container Transitioned from NEW to RESERVED | RMContainerImpl.java:394
2016-03-22 11:09:38,028 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000004 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:38,030 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000005 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:38,032 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000006 Container Transitioned from NEW to RESERVED | RMContainerImpl.java:394
2016-03-22 11:09:38,052 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000007 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:38,055 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000008 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:38,491 | INFO  | IPC Server handler 29 on 26002 | container_1458609629121_0002_01_000002 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:38,493 | INFO  | IPC Server handler 29 on 26002 | container_1458609629121_0002_01_000004 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:38,496 | INFO  | IPC Server handler 29 on 26002 | container_1458609629121_0002_01_000005 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:38,498 | INFO  | IPC Server handler 29 on 26002 | container_1458609629121_0002_01_000007 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:38,498 | INFO  | IPC Server handler 29 on 26002 | container_1458609629121_0002_01_000008 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:09:39,044 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000004 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:39,047 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000005 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:39,726 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000002 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:40,080 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000007 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:40,081 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000008 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:09:59,195 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000005 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:09:59,201 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000004 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:09:59,208 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000009 Container Transitioned from NEW to ALLOCATED | RMContainerImpl.java:394
2016-03-22 11:09:59,876 | INFO  | IPC Server handler 27 on 26002 | container_1458609629121_0002_01_000009 Container Transitioned from ALLOCATED to ACQUIRED | RMContainerImpl.java:394
2016-03-22 11:10:03,251 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000009 Container Transitioned from ACQUIRED to RUNNING | RMContainerImpl.java:394
2016-03-22 11:10:09,974 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000002 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:10:12,386 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000009 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
```

```
2016-03-22 11:10:50,021 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000008 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:11:01,063 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000010 Container Transitioned from NEW to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:11:01,905 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from RUNNING to FINAL_SAVING | RMAppAttemptImpl.java:800
2016-03-22 11:11:01,922 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from RUNNING to FINAL_SAVING | RMAppImpl.java:753
2016-03-22 11:11:01,922 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from FINAL_SAVING to FINISHING | RMAppAttemptImpl.java:800
2016-03-22 11:11:01,971 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from FINAL_SAVING to FINISHING | RMAppImpl.java:753
2016-03-22 11:11:14,288 | INFO  | ResourceManager Event Processor | container_1458609629121_0002_01_000001 Container Transitioned from RUNNING to COMPLETED | RMContainerImpl.java:394
2016-03-22 11:11:14,289 | INFO  | AsyncDispatcher event handler | appattempt_1458609629121_0002_000001 State change from FINISHING to FINISHED | RMAppAttemptImpl.java:800
2016-03-22 11:11:14,289 | INFO  | AsyncDispatcher event handler | application_1458609629121_0002 State change from FINISHING to FINISHED | RMAppImpl.java:753
```

NodeManager 运行日志

NodeManager 管理本节点的所有资源，向 RM 保持心跳并领取和启动对应的任务。NodeManager 运行日志将记录这些信息。

以下为截取的 NodeManager 日志，从内部状态机的角度展示了 RM 中 APP 的运行过程。

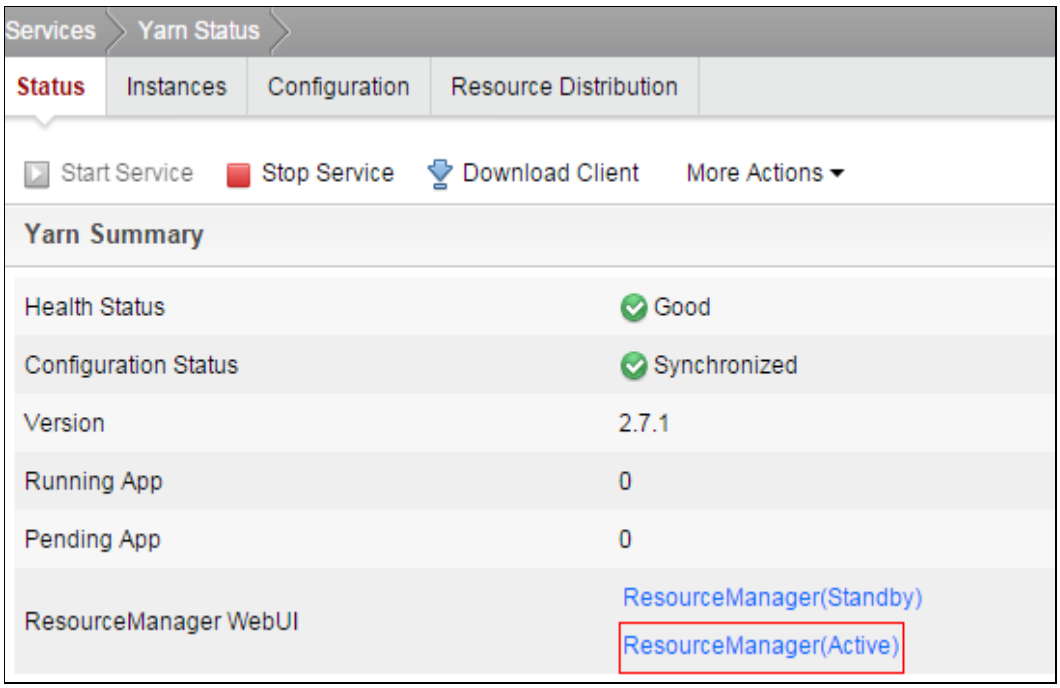


GC 日志

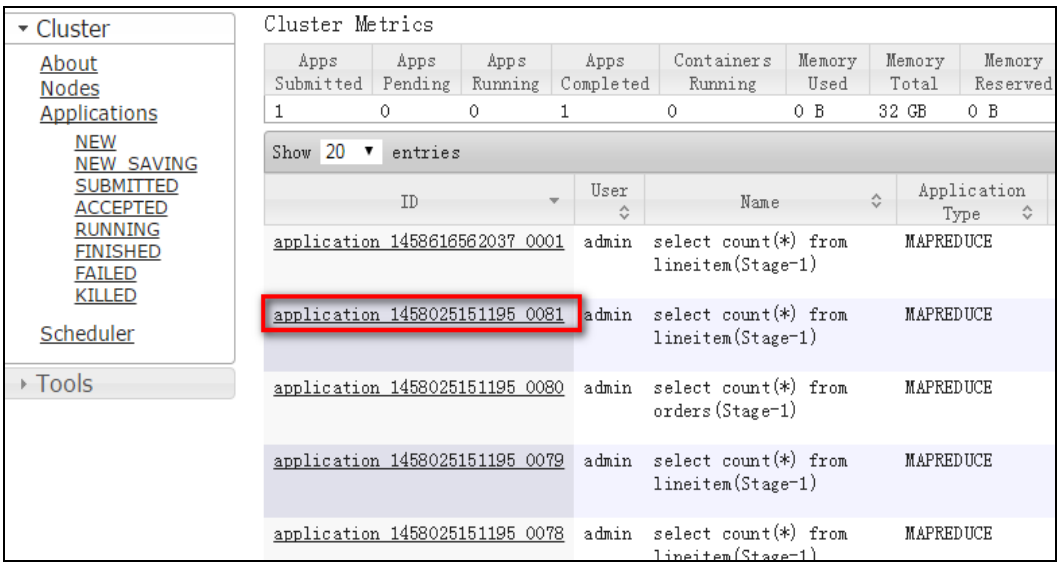
ResourceManager 和 NodeManager 均有 GC 日志，当我们遇到 GC 问题可以 GC 日志以快速定位是否是 GC 导致。

MapReduce 日志

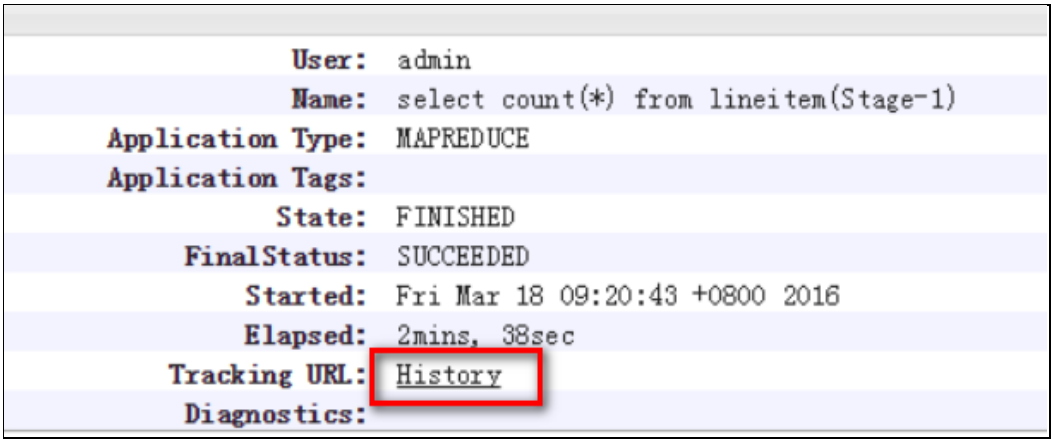
进入 OM 页面 Services->Yarn->ResourceManager,如图:



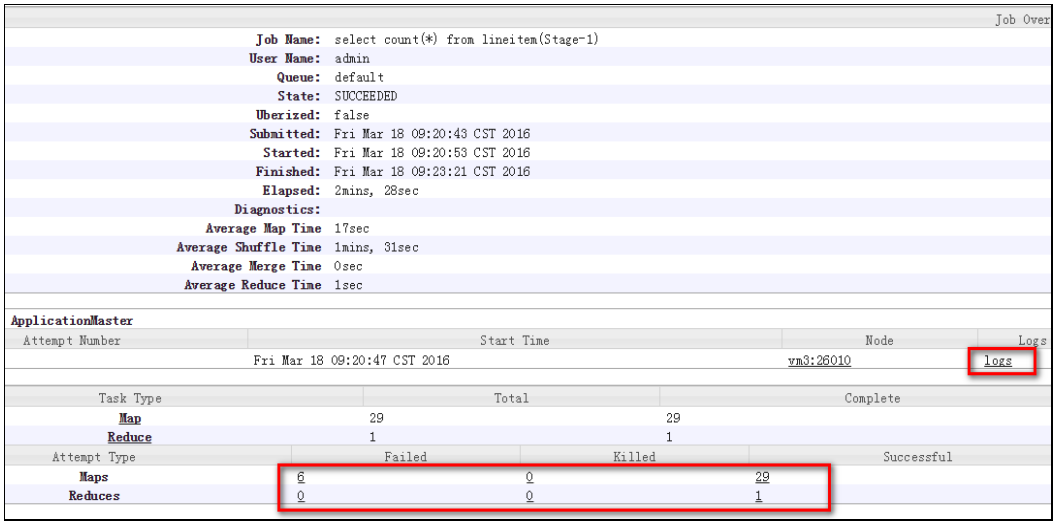
进入 Yarn 的页面后，找到我们刚才找到的 jobId，如下:



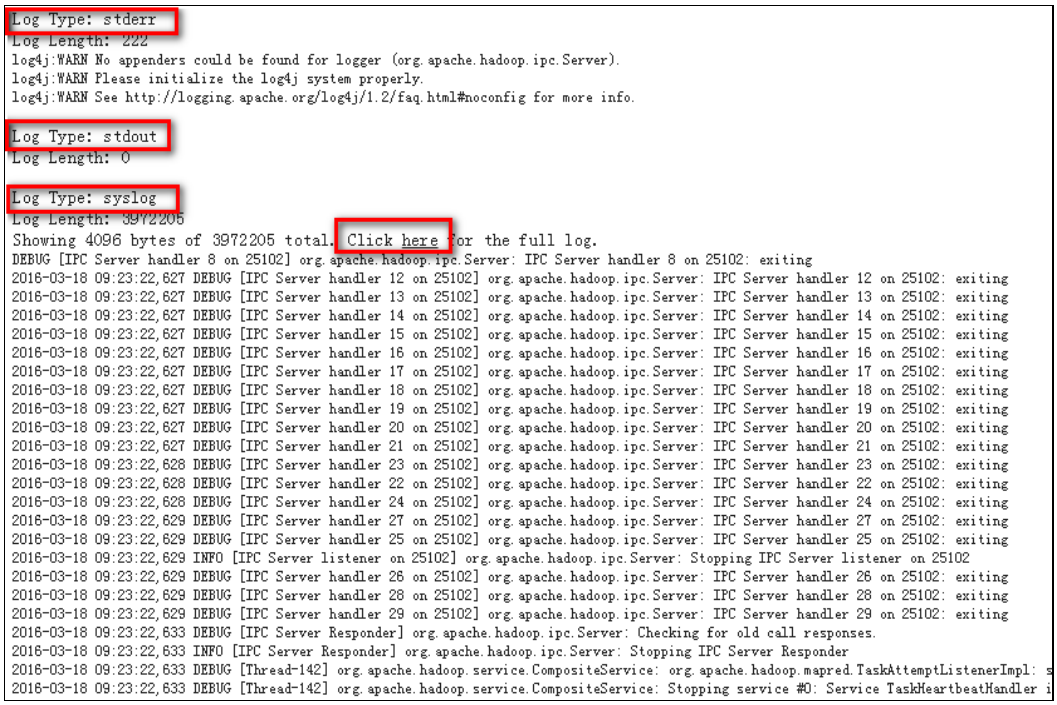
然后可以看见如下页面，点击 History。



进入下面页面，可以点击打开 AM 日志，以及任意一个 map 或者 reduce 的日志。



最终可以看到 MapReduce 的执行日志：



2、常见问题

注意：案例中 IP 地址信息，请根据实际情况进行修改。

【任务提交异常】

[YARN-10001] 在 Windows 环境下提交 MR 任务失败。

【问题背景与现象】

在 windows 上开发 MR 应用程序后失败：

```
40014 [main] INFO  org.apache.hadoop.mapreduce.Job  - Job job_1458609629121_0009 failed with state FAILED
For more detailed output, check the application tracking page:https://160-138-1-182:26001/cluster/app/app
Diagnostics: Exception from container-launch.
Container id: container_1458609629121_0009_02_000001
Exit code: 1
Stack trace: ExitCodeException exitCode=1:
        at org.apache.hadoop.util.Shell.runCommand(Shell.java:556)
        at org.apache.hadoop.util.Shell.run(Shell.java:467)
        at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:733)
        at org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor.launchContainer(LinuxContaine
        at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.ContainerLaunch.call(Conta
        at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.ContainerLaunch.call(Conta
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)

Shell output: main : command provided 1
main : run as user is omm
main : requested yarn user is omm

Container exited with a non-zero exit code 1
Failing this attempt. Failing the application.
40075 [main] INFO  org.apache.hadoop.mapreduce.Job  - Counters: 0
```

【可能原因】

1. 未设置跨平台提交任务的配置参数 mapreduce.app-submission.cross-platform。

【解决办法】

1. 将客户端参数 mapreduce.app-submission.cross-platform 参数设置为 true。
   参数添加方式请参阅[Yarn-40001]如何添加客户端参数。

[YARN-10002] 怎样在 Yarn 提交任务的时候指定队列。

【问题背景与现象】

如何在 Yarn 提交任务的时候指定队列？

【解决办法】

1. 将 mapreduce.job.queuename 设置为想要提交的队列名。
   参数添加方式请参阅[Yarn-40001]如何添加客户端参数。

[YARN-10003] 如何让 MR 任务输出的压缩文件。

【问题背景与现象】

如何让 MR 任务输出的压缩文件？

【解决办法】

1. 当前 Yarn 支持以下几种压缩格式：

```
org.apache.hadoop.io.compress.BZip2Codec
org.apache.hadoop.io.compress.Lz4Codec
org.apache.hadoop.io.compress.DeflateCodec
org.apache.hadoop.io.compress.SnappyCodec
org.apache.hadoop.io.compress.GzipCodec
```

2. 以 gz 为例：通过在客户端代码或者 mapred-sit.xml 中指定如下参数可以让 MR 任务输出为 gz 压缩：

```
mapreduce.output.fileoutputformat.compress=true
mapreduce.output.fileoutputformat.compress.type=BLOCK
mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.GzipCodec
```

3. 可以仅配置 map 的输出是否为压缩文件

```
mapreduce.map.output.compress=true
mapreduce.map.output.compress.codec= org.apache.hadoop.io.compress.GzipCodec
```

4. 参数配置方式请参阅[Yarn-40001]如何添加客户端参数。

【任务运行异常】

[YARN-20001]MR 任务运行失败，报虚拟内存不足

【问题背景与现象】

MR 任务执行失败。

【原因分析】

1. 查看 AM 日志中发现 beyond virtual memory limits，具体信息如下：

```
Container[pid=41884,containerID=container_1405950053048_0016_01_000284] is running beyond virtual memory limits. Current usage: 314.6 MB of 2.9 GB physical memory used; 8.7 GB of 6.2 GB virtual memory used. Killing container.
```

2. 分配给 container 的虚拟内存不足

【解决办法】

1. 增加 container 能使用的虚拟内存大小。
   虚拟内存的配置是由物理内存以及虚拟内存和物理内存的使用比例得到的 yarn.nodemanager.resource.memory-mb * yarn.nodemanager.vmem-pmem-ratio，因此，可以通过增加这两个参数的方式进行处理。其中 yarn.nodemanager.vmem-pmem-ratio 是服务端参数。

【扩展介绍】

1. **MR 的内存参数介绍**

yarn.app.mapreduce.am.resource.mb	MR ApplicationMaster 占用的内存量
yarn.app.mapreduce.am.command-opts	AM 进程的 JVM 参数，其中-Xmx 最大不能超过 yarn.app.mapreduce.am.resource.mb
mapreduce.map.memory.mb	每个 Map Task 需要的内存量
mapreduce.map.java.opts	Map 的 JVM 参数，其中-Xmx 最多不能超过 mapreduce.map.memory.mb * 0.75
mapreduce.reduce.memory.mb	每个 Reduce Task 需要的内存量
mapreduce.reduce.java.opts	Reduce 的 JVM 参数，其中-Xmx 最多不能超过 mapreduce.reduce.memory.mb * 0.75
yarn.nodemanager.vmem-pmem-ratio	Container 能使用的虚拟内存和物理内存的比例
yarn.scheduler.maximum-allocation-mb	单个 Container 能申请的最大内存
yarn.scheduler.minimum-allocation-mb	单个 Container 能申请的最小内存

[YARN-20002]MR 任务运行失败，报 OOM 异常

【问题背景与现象】

MR 任务执行失败。

【原因分析】

1. 查看 AM 日志中发现 beyond virtual memory limits，具体信息如下：

   Container[pid=41884,containerID=container_1405950053048_0016_01_000284] is running beyond virtual memory limits. Current usage: 314.6 MB of 2.9 GB physical memory used; 8.7 GB of 6.2 GB virtual memory used. Killing container.

2. 分配给 container 的内存不足

【解决办法】

1. 增加 container 能使用的内存大小。
   增加 mapreduce.map.memory.mb 或 mapreduce.map.java.opts 的值。这两者是有关联关系的，通常默认是 0.8，建议设置为 0.75，可由 mapreduce.job.heap.memory-mb.ratio 进行指定。

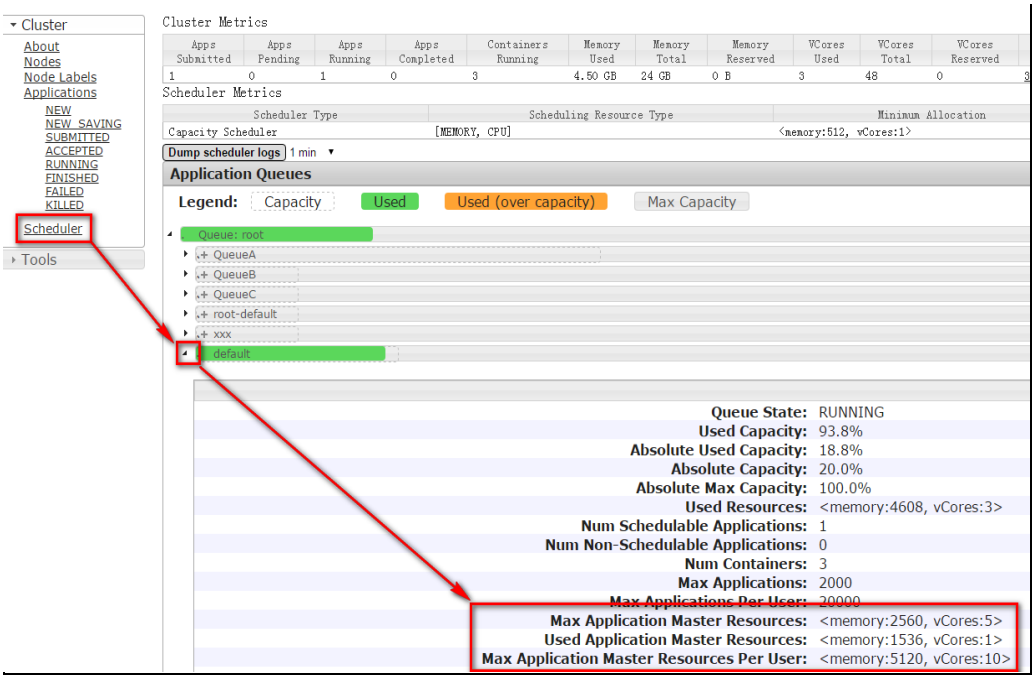
[YARN-20003] 集群资源足够时，仍然有大量任务处于 Accepted 状态

【问题背景与现象】

集群资源足够时，仍然有大量任务处于 Accepted 状态。

【原因分析】

1. 查看该队列的详细信息，查看 AM 最大允许资源配置是否已经超过。



【解决办法】

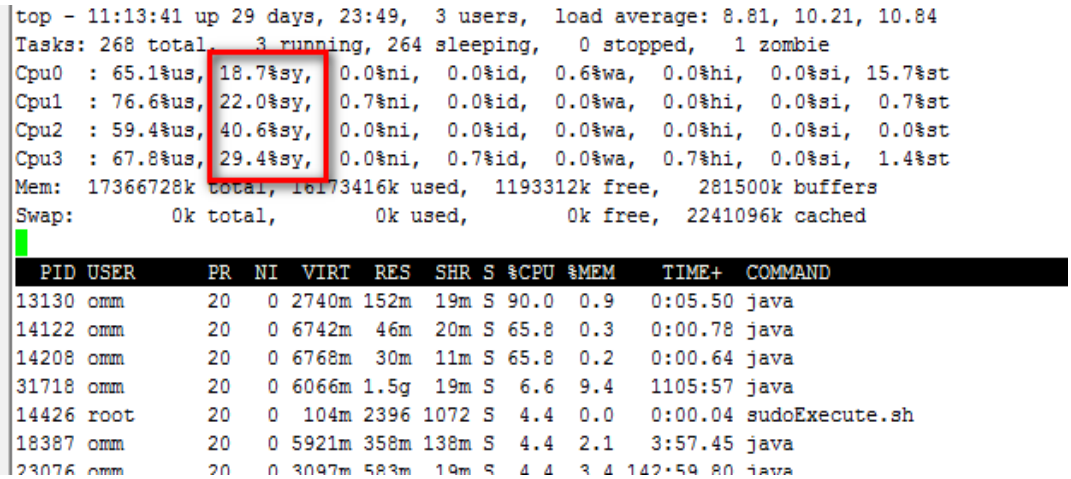
1. 如果 Used Application Master Resources 已经超过 Max Application Master Resources，

则需要等待某个任务结束后才能继续启动下一个任务。可以调整 yarn.scheduler.capacity.maximum-am-resource-percent 参数，最大允许 AM 占用集群总资源的比例。

[YARN-20004] OS 开启透明大页功能导致 MR 缓慢

【问题背景与现象】

Map/reduce 任务时间特别长，并且 top 命令发现 cpu 占用率的 sy 项特别高（红框值超过60%以上），OS 运行缓慢。



【原因分析】

1. OS 开启透明大页功能。

【解决办法】

1. 执行以下命令关闭透明大页功能：

vi /etc/rc.d/rc.local

echo "never" > /sys/kernel/mm/redhat_transparent_hugepage/enabled

echo "never" > /sys/kernel/mm/redhat_transparent_hugepage/defrag

echo "no" > /sys/kernel/mm/redhat_transparent_hugepage/khugepaged/defrag

【监控状态异常】

[YARN-30001] NodeManager 启动时出现 Concerning。

【问题背景与现象】

启动 Yarn 后，发现个别 NodeManager 出现 Concerning 状态。

【可能原因】

1. NodeManager 启动内存过小
2. NodeManager 依赖文件权限错误
3. NodeManager 读写磁盘错误

【原因分析】

1. 参看 NodeManager 的运行日志，出现 doesn't satisfy minimum allocations 错误，具体内容如下：

```
2016-03-03 16:53:35,102 | WARN  | main | USER=mapred OPERATION=nmStartup
 TARGET=NodeManager RESULT=FAILURE DESCRIPTION=Exception occurred during
startup | NMAuditLogger.java:288
2016-03-03 16:53:35,102 | FATAL | main | Error starting NodeManager |
NodeManager.java:552
org.apache.hadoop.yarn.exceptions.YarnRuntimeException:
org.apache.hadoop.yarn.exceptions.YarnRuntimeException: Recieved SHUTDOWN signal
from Resourcemanager, Registration of NodeManager failed, Message from
ResourceManager: NodeManager from  xxx-xxx-1-182 doesn't satisfy minimum
allocations, Sending SHUTDOWN signal to the NodeManager.
```

说明设置的 NodeManager 启动内存过小，小于 NodeManager 所需的最小内存解决方法参考 NODEMANAGER 内存过小

2. 检查

opt/huawei/Bigdata/FusionInsight-Hadoop-2.7.1/hadoop/bin/container-executor 权限，发现被修改

```
total 268
---Sr-s---. 1 root wheel 114035 Sep 30 00:28 container-executor
-rw-------. 1 root root      2 Feb 27 16:34 drop_caches
-r-x------. 1 omm  wheel   6524 Sep 30 00:28 hadoop
-r-x------. 1 omm  wheel   8514 Sep 30 00:28 hadoop.cmd
-rwx------. 1 omm  wheel   2375 Oct 16 03:19 hadoop-oam.sh
-r-x------. 1 omm  wheel  12523 Sep 30 00:28 hdfs
-rwx------. 1 omm  wheel   1133 Oct 16 03:19 hdfs-backup.sh
```

需要将权限修改正确，解决方法参考 NODEMANAGER 文件权限错误

3. 查看/etc/fatab 中是否包含该项目

```
#
# /etc/fstab
# Created by anaconda on Fri Apr 24 16:31:37 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/vg_160138118-lv_root /                       ext4    defaults,noatime,errors=panic       1 1
UUID=53befa2b-f7ac-43be-b013-3a1fbca267d2 /boot                   ext4    defaults,noatime,errors=panic       1 2
/dev/mapper/vg_160138118-lv_home /home                   ext4    defaults,noatime,errors=panic       1 2
/dev/mapper/vg_160138118-lv_swap swap                    swap    defaults        0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
/dev/sdb1       /opt        ext4    defaults,nosuid,noatime,errors=panic        0 0
/dev/sda1       /srv/BigData        ext4    defaults,noatime        0 0
~
~
```

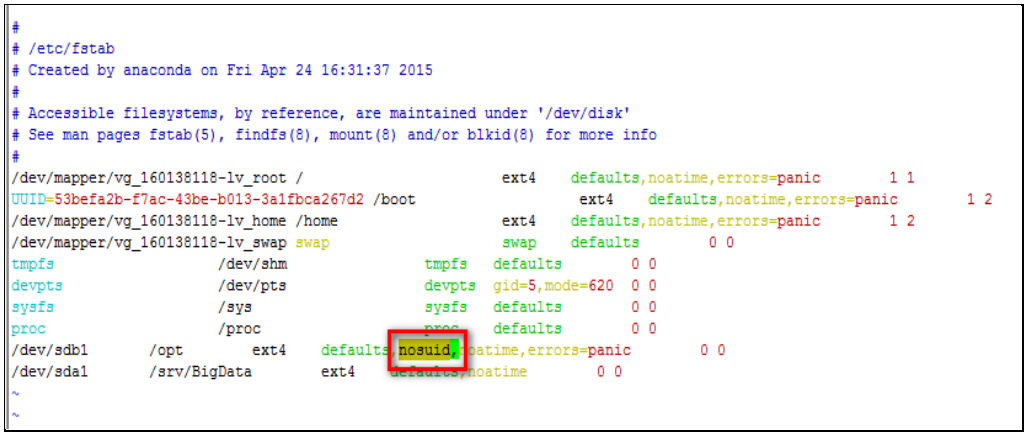
解决方法参考 NODEMANAGER 磁盘挂载错误

【解决办法】

1. NodeManager 内存过小
   修改 yarn.nodemanager.resource.memory-mb 参数，至少需要大于 yarn.scheduler.minimum-allocation-mb 这个参数

2. NodeManager 文件权限错误
   chmod 050 container-executor
   chmod +s container-executor

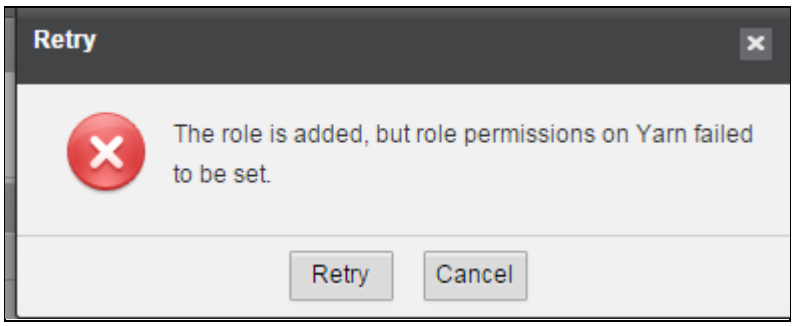
3. NodeManager 磁盘挂载错误
   确保停止该磁盘上所有文件的读写，然后删除红框中的项，执行 mount –a 命令重新挂载磁盘。



[Yarn-30002]在 OM 页面上添加 Yarn 的权限失败

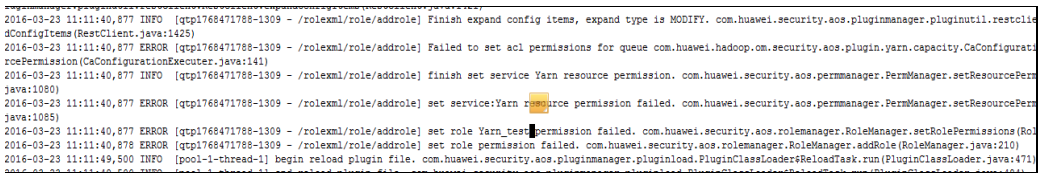
【问题背景与现象】

在 OM 页面创建角色时，添加 Yarn 权限失败：



【原因分析】

1. 查看/var/log/Bigdata/controller/aos.log，发现 set role permission failed 报，具体内容如下：



文件权限错误导致添加用户失败。

【解决办法】

在 RM 主备节点上使用

1. lsof /opt/huawei/Bigdata/etc/*_*_ResourceManager/capacity-scheduler.xml 查看是否有其他进程打开这个文件
2. lsof /opt/huawei/Bigdata/etc/*_*_ResourceManager/.capacity-scheduler.xml.swp 查看是否有其他进程打开这个文件
3. 如果有，关闭这个进程。

[Yarn-30003]通过后台命令动态刷新队列配置失败

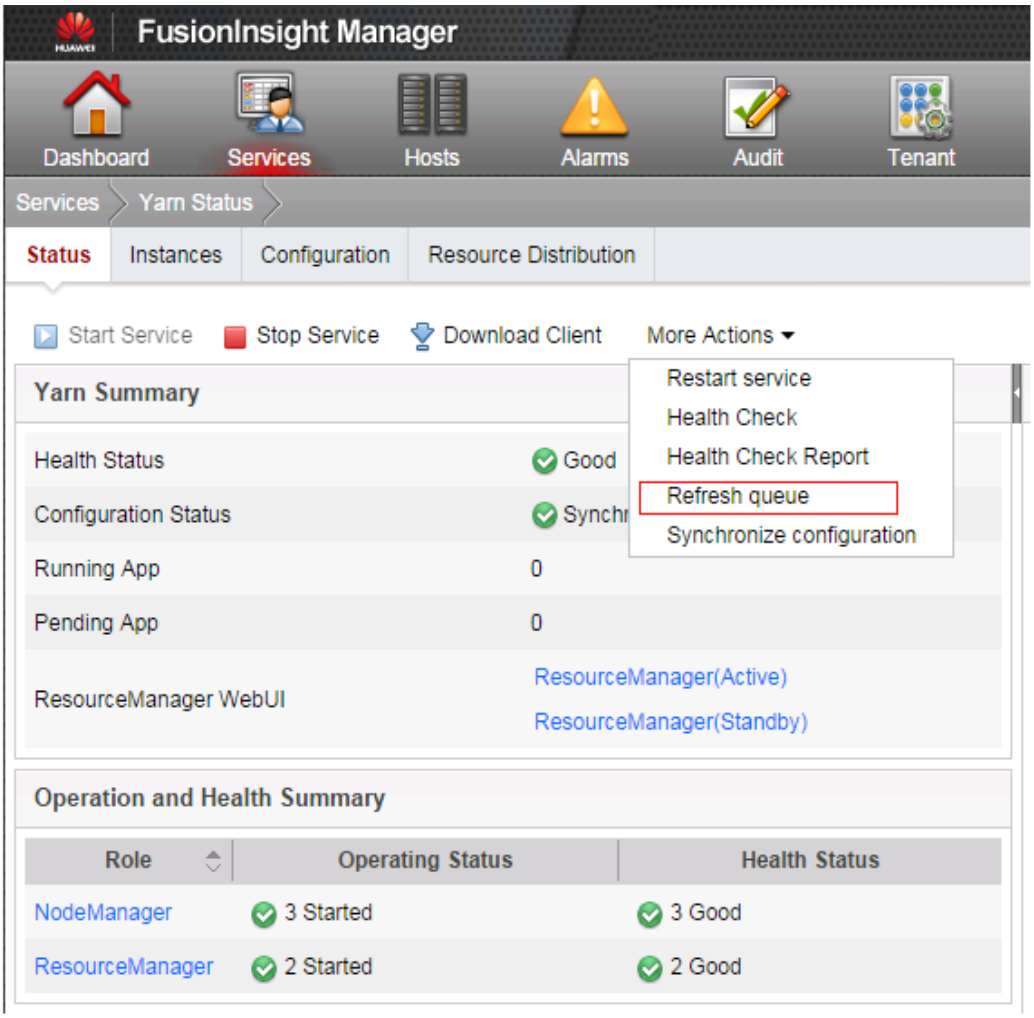
【问题背景与现象】

在 OM 页面创建角色时，添加 Yarn 权限失败：

```
sdp@dkfzxsmis0app57:~> source /datafs2/sdp/FusionInsight_Client/bigdata_env
sdp@dkfzxsmis0app57:~> kinit -k -t /datafs2/sdp/FusionInsight_Client/keytab/SDP_BATCH.keytab SDP_BATCH
sdp@dkfzxsmis0app57:~> $HW_CLIENT_NAME/HDFS/hadoop/bin/yarn rmadmin -refreshQueues
No GC_PROFILE is given. Defaults to medium.
Exception in thread "main" java.lang.IllegalArgumentException: Can't get Kerberos realm
        at org.apache.hadoop.security.HadoopKerberosName.setConfiguration(HadoopKerberosName.java:65)
        at org.apache.hadoop.security.UserGroupInformation.initialize(UserGroupInformation.java:261)
        at org.apache.hadoop.security.UserGroupInformation.ensureInitialized(UserGroupInformation.java:246)
        at org.apache.hadoop.security.UserGroupInformation.loginUserFromSubject(UserGroupInformation.java:746)
        at org.apache.hadoop.security.UserGroupInformation.getLoginUser(UserGroupInformation.java:731)
        at org.apache.hadoop.security.UserGroupInformation.getCurrentUser(UserGroupInformation.java:633)
        at org.apache.hadoop.hdfs.tools.GetConf.run(GetConf.java:315)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:75)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
        at org.apache.hadoop.hdfs.tools.GetConf.main(GetConf.java:332)
Caused by: java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.security.authentication.util.KerberosUtil.getDefaultRealm(KerberosUtil.java:75)
        at org.apache.hadoop.security.HadoopKerberosName.setConfiguration(HadoopKerberosName.java:63)
        ... 9 more
Caused by: KrbException: Cannot locate default realm
        at sun.security.krb5.Config.getDefaultRealm(Config.java:1183)
        ... 15 more
Caused by: KrbException: Generic error (description in e-text) (60) - Unable to locate Kerberos realm
        at sun.security.krb5.Config.getRealmFromDNS(Config.java:1279)
        at sun.security.krb5.Config.getDefaultRealm(Config.java:1164)
        ... 15 more
```

【解决办法】

FusionInsight 提供了图形化刷新队列的方案，不支持后台命令行刷新，可以使用如下操作刷新队列配置。



【扩展介绍】

Yarn 默认资源调度器主要参数介绍

yarn.scheduler.capacity.<queue-path>.capacity	队列的资源容量（百分比），所有队列的该配置加起来不能超过 100
yarn.scheduler.capacity.<queue-path>.maximum-capacity	队列的资源使用上限（百分比）。由于存在资源共享，因此一个队列使用的资源量可能超过其容量，而最多使用资源量可通过该参数限制。
yarn.scheduler.capacity.<queue-path>.maximum-applications	集群或者队列中同时处于等待和运行状态的应用程序数目上限，这是一个强限制，一旦集群中应用程序数目超过该上限，后续提交的应用程序将被拒绝，默认值为 10000
yarn.scheduler.capacity.<queue-path>.maximum-am-resource-percent	集群中用于运行应用程序 ApplicationMaster 的资源比例上限，该参数通常用于限制处于活动状态的应用程序数目。该参数类型为浮点型，默认是 0.1，表

	示 10%。
yarn.scheduler.capacity.<queue-path>.user-limit-factor	每个用户最多可使用的资源量（百分比）

[Yarn-30004]通过 FI Manager 管理页面动态刷新队列配置失败

【问题背景与现象】

FusionInsight Manager 管理页面刷新队列失败。

【原因分析】

RM 日志中打印 Illegal capacity 错误，红框处的值大于 1

```
s=0, numContainers=0 | CapacityScheduler.java:656
2016-03-26 10:56:22,546 | WARN  | IPC Server handler 0 on 26005 | Exception refresh queues. | AdminService.java:710
java.io.IOException: Failed to re-init queues
        at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler.reinitialize(CapacityScheduler.java:390)
        at org.apache.hadoop.yarn.server.resourcemanager.AdminService.refreshQueues(AdminService.java:376)
        at org.apache.hadoop.yarn.server.api.impl.pb.service.ResourceManagerAdministrationProtocolPBServiceImpl.refreshQueues(ResourceMan
        at org.apache.hadoop.yarn.proto.ResourceManagerAdministrationProtocol$ResourceManagerAdministrationProtocolService$2.callBlocking
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:616)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:973)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2088)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2084)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1672)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2082)
Caused by: java.lang.IllegalArgumentException: Illegal capacity of 1.4000001 for children of queue root
        at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.ParentQueue.setChildQueues(ParentQueue.java:145)
        at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler.parseQueue(CapacityScheduler.java:646)
        at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler.reinitializeQueues(CapacityScheduler.java:5
        at org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler.reinitialize(CapacityScheduler.java:386)
        ... 11 more
2016-03-26 10:56:22,554 | WARN  | IPC Server handler 0 on 26005 | USER=mapred   IP=160.138.1.181       OPERATION=refreshQueues TARGET=Ad
es.   PERMISSIONS= | RMAuditLogger.java:51
```

【解决办法】

所有的 yarn.scheduler.capacity.<queue-path>.capacity 配置项的值加起来不得超过 100。

【咨询】

[Yarn-40001]如何添加客户端参数

【解决办法】

方式一：为客户端添加 mapred-site.xml 配置文件，在其中配置需要的属性，如下：

```
<property>
    <name>mapreduce.app-submission.cross-platform</name>
    <value>true</value>
</property>
```

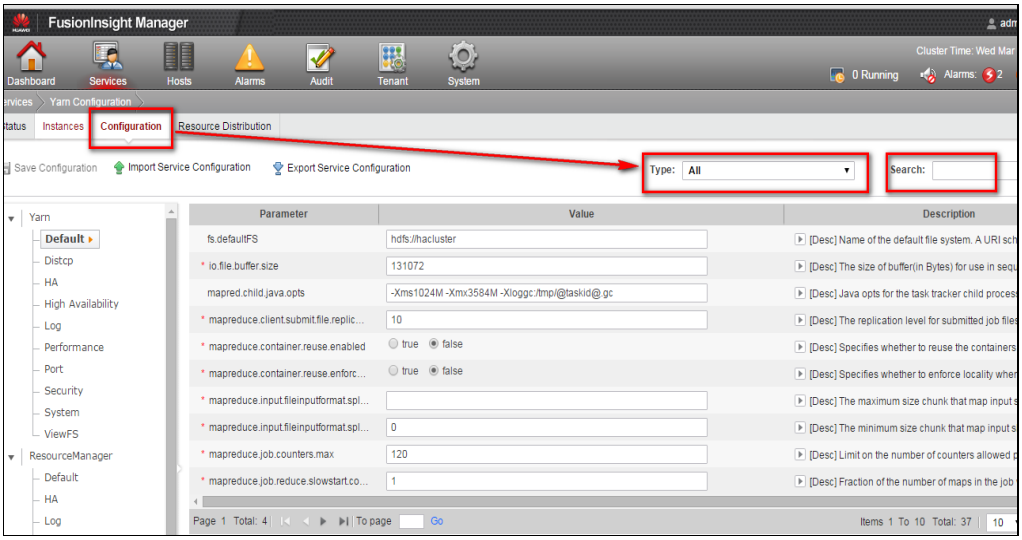
方式二：在客户端代码中添加参数设置代码，例如：

conf.set(“mapreduce.app-submission.cross-platform”, "true");

[Yarn-40002]如何添加服务端参数

【解决办法】

点击 Services->Yarn->Configuration，并选择 Type 为 ALL，如下：



在搜索框内搜索需要修改的参数并修改。