YARN资源隔离

Container是资源调度的基础，也是资源隔离的基础。理论上说，不同container之间不应该相互影响。在YARN中ContainerExecutor负责container的初始化、启动以及cleanup。YARN提供了多种ContainerExcecutor，通过属性：yarn.nodemanager.container-executor.class配置：

1. DefaultContainerExecutor，每个Container运行在单独的进程里，但是进程都是由NM的用户启动的，比如NM进程就是用yarn用户启动的，那么所有container进程也是用yarn用户启动。
2. LinuxContainerExecutor，每个contaienr由不同的用户启动，例如A用户提交job的container，由A用户启动。此外支持单独的配置文件及简单的ACL。
3. DockerContainerExecutor，NM将container进程运行在Docker容器中，可以提供自定义，与外部NM隔离的运行环境。

LCE和DCE相比，明显隔离性更好，对于LCE来说：

1）需要Linux Native程序的支持，container-executor程序，使用C来实现。

2）需要一个配置文件container-executor.cfg，这个配置文件和container-executor的二进制文件的相对路径还是固定的。

3）用不同的用户启动container，必须有对应的Linux用户存在，因此增加用户时，在所有的NM节点上都要对应增加

4）container-executor和container-executor.cfg的所有者都必须是root，而且他们的目录上溯到/，所有者必须是root

5）container-executor的文件权限必须是6050（--Sr-s---），因为其执行原理就是setuid/setgid。Group owner必须和启动的NM的用户同组。例如NM由yarn用户启动，yarn用户属于hadoop组，那container-executor必须也是hadoop组。

启动container本质上就是运行一个脚本，可以简单理解DCE是直接*/bin/bash $\**，但是LCE使用的是./container-executor $\*，额外包了一层。

如果直接执行container-executor:

*# ./container-executor*

*Usage: container-executor --checksetup*

*Usage: container-executor --mount-cgroups hierarchy controller=path...*

*Usage: container-executor user yarn-user command command-args*

*Commands:*

*initialize container: 0 appid tokens nm-local-dirs nm-log-dirs cmd app...*

*launch container: 1 appid containerid workdir container-script tokens pidfile nm-local-dirs nm-log-dirs resources*

*signal container: 2 container-pid signal*

*delete as user: 3 relative-path*

container-executor在执行命令是时会从container-executor.cfg读取配置，用于细粒度控制，配置示例如下所示：

*yarn.nodemanager.linux-container-executor.group=hadoop*

*banned.users=bin*

*min.user.id=99*

LCE的资源隔离：

1. 内存隔离

YARN对内存其实没有真正隔离，而是监视container进程的内存使用，超出限制后直接杀掉进程。相关逻辑见ContainersMonitorImpl类，进程监控的逻辑见ProcfsBasedProcessTree类，原理是读取/proc/$pid下面的文件，获取进程的内存占用。

1. cpu隔离

默认情况下，没有没有考虑cpu的隔离

使用cgroup进行资源隔离的设置属性：

*yarn.nodemanager.linux-container-executor.resources-handler.class*

可以配置成：

*org.apache.hadoop.yarn.server.nodemanager.util.CgroupsLCEResourcesHandler*

*https://zh.hortonworks.com/blog/part-5-of-data-lake-3-0-yarn-and-containerization-supporting-docker-and-beyond/*

*http://www.slideshare.net/HadoopSummit/yarn-and-the-docker-container-runtime*