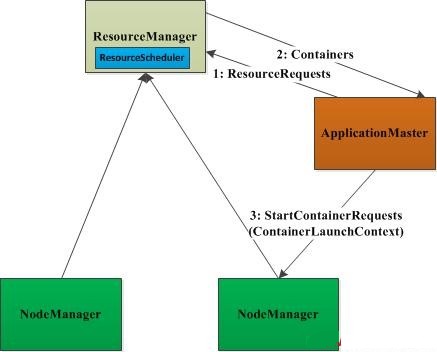
Container资源管理及控制

# 1.Yarn Container简介

用户将应用程序提交到ResourceManager上，首先为ApplicationMaster申请资源，并与某个NodeManager通信，以启动ApplicationMaster。AM与RM通信，为内部要执行的任务申请资源，一旦得到资源后，将于NodeManager通信，以启动对应的任务。其中AM与Task均运行在Container中。

ResourceManager中的资源调度器负责资源的管理和调度，其中分配的单位是Container，实现类为ContainerPBImpl，每个Container都有分配的memory和vcores。

下图是Resource在整个运行流程中的使用：



1）资源的初始化

AM向资源调度器申请资源，需要向它发送一个ResourceRequest列表，其中，每个ResourceRequest描述了一个资源单元的详细需求，而资源调度器则为之返回分配到的资源描述Container。

2）资源的调度/分配

RM根据集群整体的资源使用情况，分配Containers给ApplicationMaster，AM根据程序特点，在不同的Container中启动相应的服务。

3）Container实例化

Container的启动由ContainerLauncher服务来完成，YARN提供了DefaultContainerExec-

utror及LinuxContainerExecutor。ContainerImpl向ContainersLauncher服务发送

LAUNCH\_CONTAINER事件，请求启动Container，为Container创建一个Callable类型的对象ContainerLaunch，为Container创建Tokens文件和执行脚本Launch\_container.sh，通过ShellCommandExecutor.execute执行。

# 2.Container内存隔离

应用程序的内存监控是由服务ContainersMonitorImpl实现的，它保存了每个Container进程的pid，内部的MonitorThread线程每隔一段时间扫描正在运行的Container进程树，并检查使用的内存量是否超过上限值。

Container的内存限制由ContainerImpl#sendContainerStartEvent来设置，源码如下所示：

dispatcher.getEventHandler().handle(new ContainerStartMonitoringEvent(containerId,

vmemBytes, pmemBytes, cpuVcores));

超过限制，将container kill掉

if (isVmemCheckEnabled()&& isProcessTreeOverLimit(containerId.toString(),

currentVmemUsage, curMemUsageOfAgedProcesses, vmemLimit))

isMemoryOverLimit = true;

containerExitStatus = ContainerExitStatus.KILLED\_EXCEEDED\_VMEM;

}

超过限制，会发送ContainerKillEvent，错误信息为：

String.format("Container [pid=%s,containerID=%s] is running beyond %s memory limits. ",

pId, containerId, memTypeExceeded) +"Current usage: " +

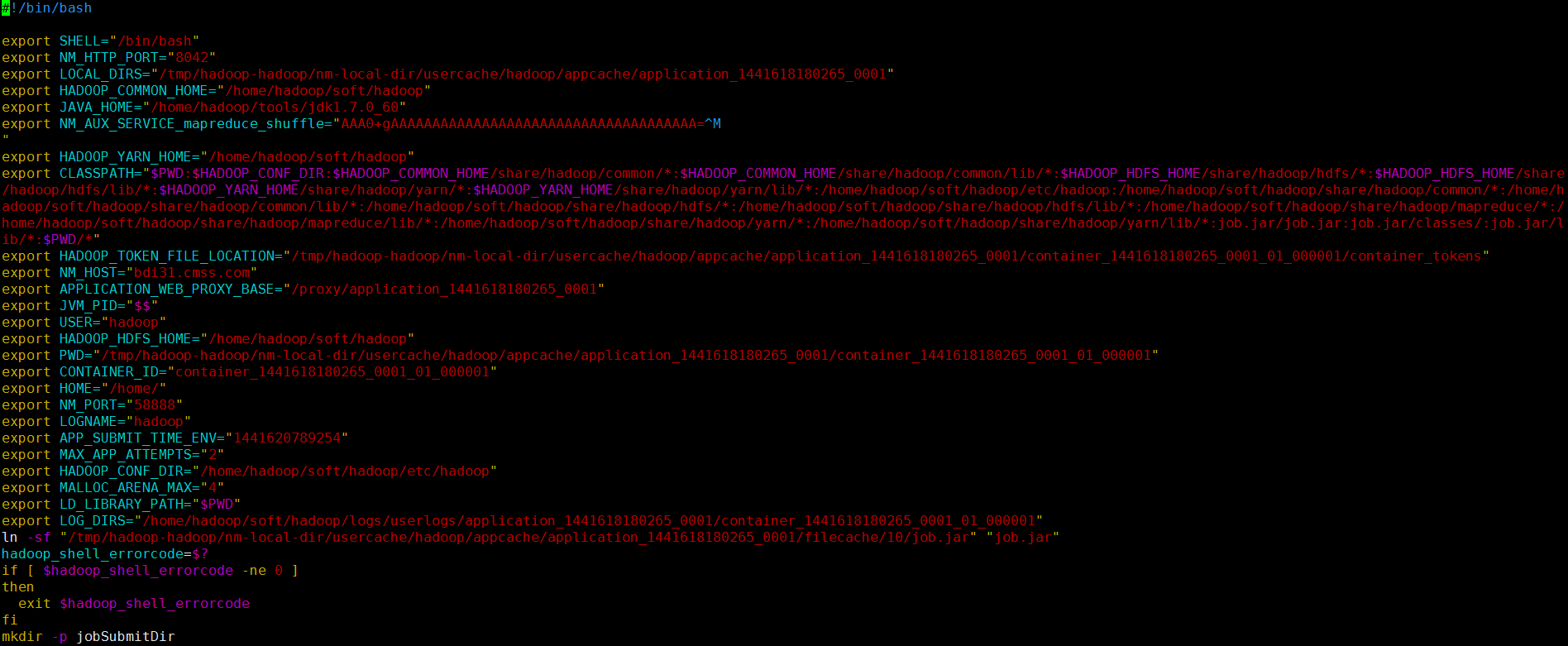
formatUsageString(currentVmemUsage, vmemLimit, currentPmemUsage, pmemLimit) +". Killing container.\n" +"Dump of the process-tree for " + containerId + " :\n"

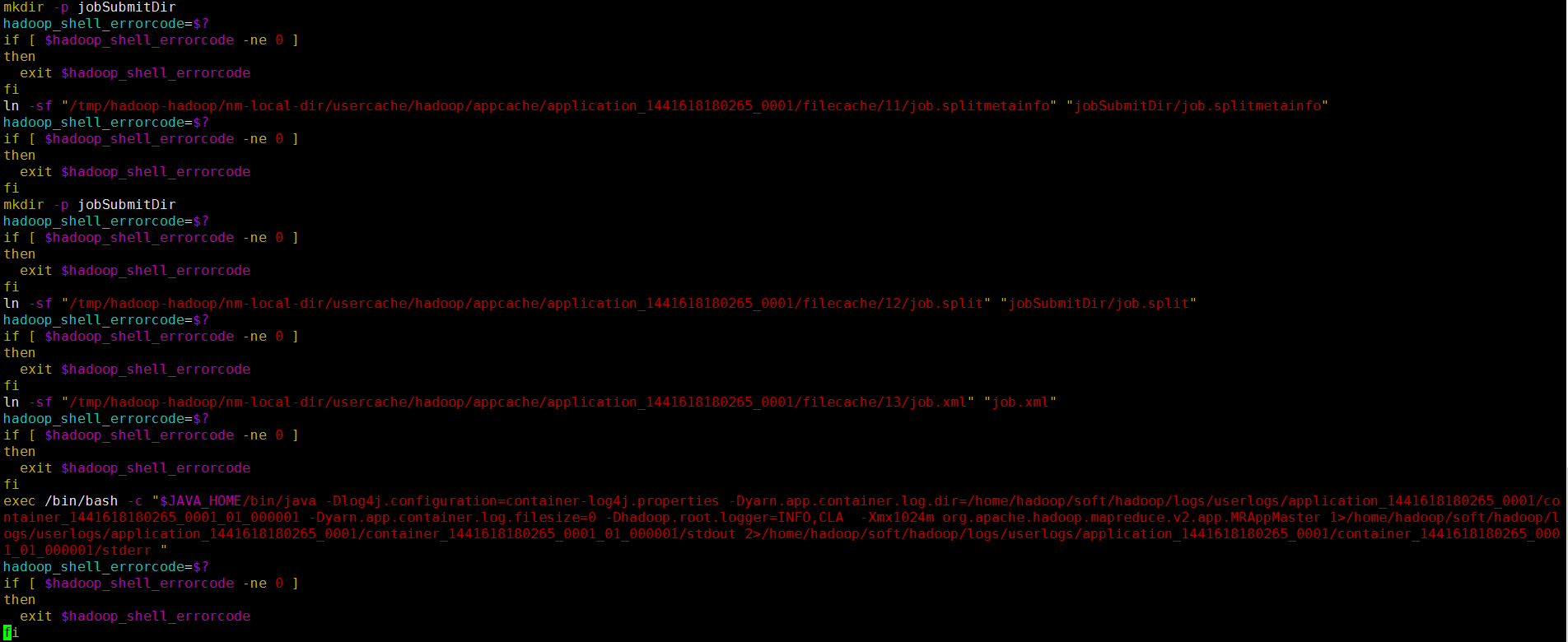
# 3.Container的进程树

在Container的内存隔离中，ContainersMonitorImpl根据ContainerID的进程树检测其内存的限制，源码如下：

ResourceCalculatorProcessTree pTree = ptInfo.getProcessTree();

下面是launch\_container.sh的示例程序：





核心程序是通过exec /bin/bash -c "$JAVA\_HOME/bin/java启动AppMaster程序，在container中启动的程序是通过ContainerLaunchContext的setCommand来设置。