

## 1、调优前景

一般说yarn的调优，就是指Container的调优，而Container是运行在NM节点上的

在NM的节点，一般还可能运行着DN进程，DN+NM两进程不可占据节点的全部内存，一般在75%~85%之间比较合适

对于一台64G内存的机器，生产上可使得DN+NM=54G，DN=4G,NM=50G

## 2、可以调的参数

一般没说生产调则在生产上不要动即可

### A、内存调优：

**yarn.nodemanager.resource.memory-mb**      NM占用机器的内存大小，单位MB 可设置为50\*1024（**生产调**）

**yarn.scheduler.minimum-allocation-mb**      Container包含的最小内存,单位MB 可设置1024（**生产调**）

**yarn.scheduler.maximum-allocation-mb**      Container包含的最大内存,单位MB 可设置8192（**生产调**）

随着任务的运行，计算越来越大，任务消耗的资源越来越多，Container包含的内存也应该逐渐增加，在CDH版本的hadoop中有一个参数可以设置每次内存加大的额度，当内存达到yarn.scheduler.maximum-allocation-mb参数设置的内存还不够时，任务会被yarn杀掉

**yarn.scheduler.increment-allocation-mb**      Container内存不够用时一次性加多少内存 单位MB 可设置2048（**仅CDH**）

**yarn.nodemanager.pmem-check-enabled**      容器是否会执行物理内存限制      默认为True

执行物理内存限制时，当Container达到最大物理内存限制时，改参数为True时,会杀死任务

**yarn.nodemanager.vmem-check-enabled** 容器是否会执行虚拟内存限制 默认为True

执行物理虚拟限制时，当Container达到最大物理虚拟限制时，该参数为True时,会杀死任务

**yarn.nodemanager.vmem-pmem-ratio** 使用虚拟内存和物理内存的比值 默认为2.1

**vm.swappiness** 的值介于0到100之间

**是否使用虚拟内存的场景：**

a.当任务处于高度计算时，不允许慢，允许挂掉后通过各种形式重启任务 这时候将使用虚拟内存的积极性调低 vm.swappiness 的值可设置较大

b.当任务在一般计算时，可以慢，不允许挂，这时候将使用虚拟内存的积极性调高 vm.swappiness 的值可设置较小

生产上一般是允许慢不允许挂，swappiness一般设置为10

**虚拟内存存在linux中的设置情况如下：**

swappiness=0的时候表示最大限度使用物理内存，然后才是 swap空间，swappiness = 100的时候表示积极的使用swap分区，并且把内存上的数据及时的搬运到swap空间里面。linux的基本默认设置为60，具体如下：

```
[root@timeserver ~]# cat /proc/sys/vm/swappiness
```

60

也就是说，你的内存在使用到 $100-60=40\%$ 的时候，就开始会使用虚拟内存。

临时调整（重启失效）：

```
[root@timeserver ~]# sysctl vm.swappiness=10
```

```
vm.swappiness = 10
```

```
[root@timeserver ~]# cat /proc/sys/vm/swappiness
```

```
10
```

永久调整，先完成临时调整，再执行以下操作：

```
[root@timeserver ~]# cat /etc/sysctl.conf
```

```
# Controls the maximum number of shared memory segments, in pages
```

在 “kernel.shmall = 4294967296” 后加上

```
“vm.swappiness=10”
```

```
[root@timeserver ~]# sysctl -p
```

## B、CPU调优

在yarn中，一个物理core等价于两个vcore，这个在CDH版本的hadoop中有一个参数控制（不要轻易调）

### **yarn.nodemanager.resource.cpu-vcores**

可为Container分配的容器数量，这个参数在RM Scheduler为Container分配资源时可以用到。如果它的值被设置为-1，

且参数yarn.nodemanager.resource.detect-hardware-capabilities的值为true,则不限制yarn使用CPU的数量，也可以设置为8

**yarn.nodemanager.resource.detect-hardware-capabilities** 是否自动检测节点的CPU和内存 默认为false

**yarn.scheduler.minimum-allocation-vcores** 为每个Container分配的最少的vcore,小于这个参数设定的值则会抛出异常InvalidResourceRequestException 默认为1

**yarn.scheduler.maximum-allocation-vcores** 为每个Container分配的最多的vcore,大于这个参数设定的值则会抛出异常InvalidResourceRequestException 默认为4(生产调)