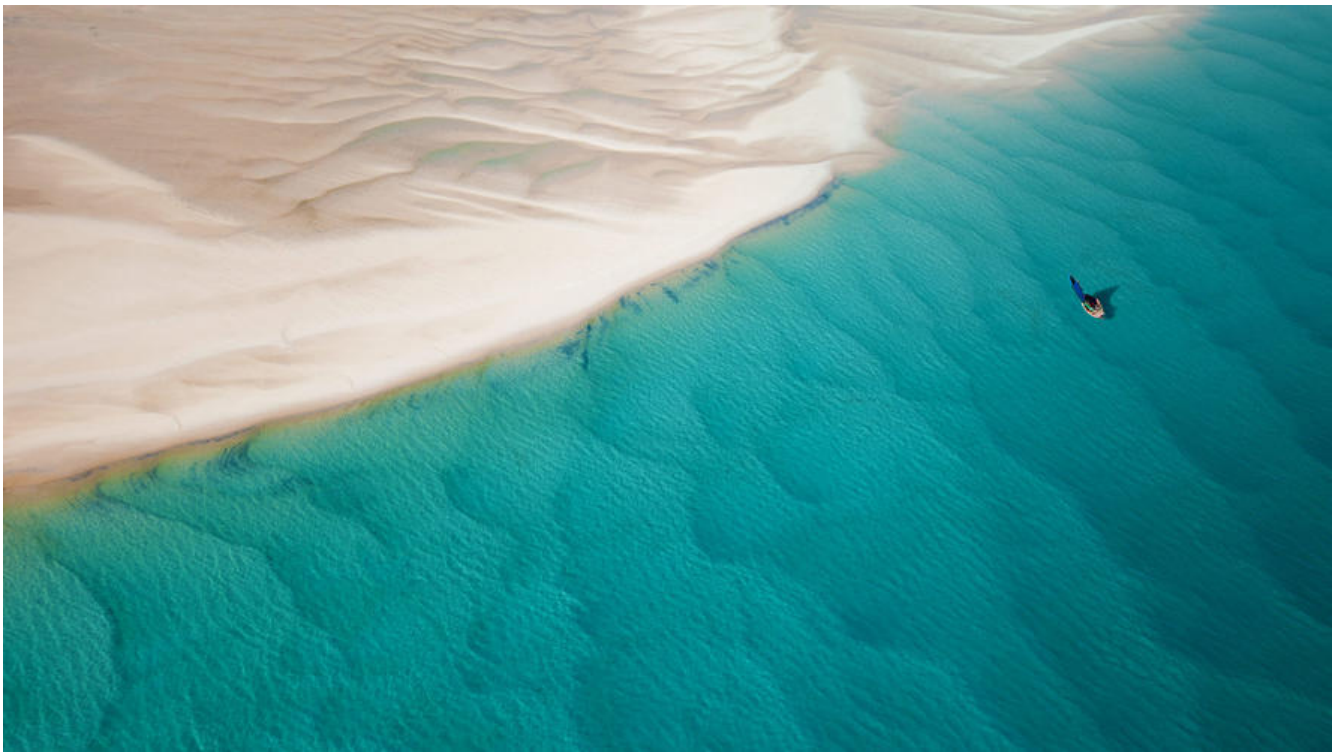
链滴

一次故障的查找记录

作者：someone33881
原文链接：https://ld246.com/article/1547619484329
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)



●

背景

    在一次某个项目sp-ja
运行过程中，总是在过一段时间后该jar进程总是会“无缘无故”（怎么可能是无缘无故呢，一切都是原因的，只不过自己还不知道而已）结束掉或者被杀死，因此决定进行一次进行查找以自我地提升，竟在很小的一个以外包demo服务为导向的非技术性公司，这样的机会也是非常少的！

    既然，难得有这样的
战机会，就不应该再仅仅停留在理论上对各类故障定位解决、JVM分析的理论层面上!于是，决定进行番探索！

1、系统日志

● 首先，在该应用运行过程中，记录其运行的PID号为24296

● 在系统日志/var/log/messages中定位原因，执行命令

cd /var/log

cat messages | grep 24296

```
[root@tst-wj-dwhqjy-linux01 log]# cat messages | grep 24296
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [24296]     0 24296  2070217   465017    1180        0             0 java
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Out of memory: Kill process 24296 (java) score 225 or sacrifice child
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Killed process 24296 (java) total-vm:8280868kB, anon-rss:1860068kB, file-rss:0kB, shmem-rss:0kB
```

● 如查询结果，显然知道是出现了OOM异常，被系统杀死了；接下来，就是定位是应用中哪段代码致的该异常

分析插曲：

毕竟第一次摸索着实践，也没啥有经验人士面基，因此分析以为：

● 第一次以为，在应用发生关闭的时候，是系统内存超了；而这个时候，系统会自动查找一个内存占最大的进程也即sp给杀死（=>但是sp被杀死，并不一定就是因为sp这个应用而导致的系统内存超出，也有可能是其他的应用导致的系统内存超出，然后系统去杀死了内存占用最大的这个sp）

● =>，因此决定去查看上面那一段报错之前的一段log信息内容，只能vim messages加上搜索时间”Jan 16 02:34:34 “，去查看详细的时间点的内容如下，

```
Jan 16 02:01:01 tst-wj-dwhqjy-linux01 systemd: Starting Session 5779 of user root.
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: java invoked oom-killer: gfp_mask=0x280da, order=0, oom_score_adj=0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: java cpuset=/ mems_allowed=0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: CPU: 1 PID: 6411 Comm: java Not tainted 3.10.0-693.5.2.el7.x86_64 #1
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Hardware name: Fedora Project OpenStack Nova, BIOS seabios-1.7.5-11.el7 04/01/2014
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: ffff8801cdeebf40 0000000098426cf6 ffff8800af7dba48 ffffffff816a3e51
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: ffff8800af7dbad8 ffffffff8169f246 ffff8800af7dbae0 ffffffff812b7d1b
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: ffff8801d8618688 0000000000000206 ffffffffaf7dbb28 ffffeffff00000000
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Call Trace:
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff816a3e51>] dump_stack+0x19/0x1b
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff8169f246>] dump_header+0x90/0x229
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff812b7d1b>] ? cred_has_capability+0x6b/0x120
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff811863a4>] oom_kill_process+0x254/0x3d0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff812b7efe>] ? selinux_capable+0x2e/0x40
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff81186be6>] out_of_memory+0x4b6/0x4f0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff8169fd4a>] __alloc_pages_slowpath+0x5d6/0x724
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff8118cdb5>] __alloc_pages_nodemask+0x405/0x420
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff811d40a5>] alloc_pages_vma+0xb5/0x200
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff811b2350>] handle_mm_fault+0xb60/0xfa0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff816b0074>] __do_page_fault+0x154/0x450
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff811ba02a>] ? mprotect_fixup+0x14a/0x240
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff816b0456>] trace_do_page_fault+0x56/0x150
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff816afaea>] do_async_page_fault+0x1a/0xd0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: [<ffffffff816ac5f8>] async_page_fault+0x28/0x30
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Mem-Info:
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: active_anon:1897780 inactive_anon:19814 isolated_anon:0#012 active_file:1087 inactive_file:1295 isolated_file:124#012 unevictable:0 dirty:26
writeback:0 unstable:0#012 slab_reclaimable:7305 slab_unreclaimable:15131#012 mapped:1672 shmem:31699 pagetables:8347 bounce:0#012 free:25939 free_pcp:483 free_cma:0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 DMA free:15908kB min:132kB low:164kB high:196kB active_anon:0kB inactive_anon:0kB active_file:0kB inactive_file:0kB unevictable:0kB is
olated(anon):0kB isolated(file):0kB present:15992kB managed:15908kB mlocked:0kB dirty:0kB writeback:0kB mapped:0kB shmem:0kB slab_reclaimable:0kB slab_unreclaimable:0kB kernel_stack:0kB p
agetables:0kB unstable:0kB bounce:0kB free_pcp:0kB local_pcp:0kB free_cma:0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? yes
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: lowmem_reserve[]: 0 2814 7804 7804
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 DMA32 free:46492kB min:24324kB low:30404kB high:36484kB active_anon:2764192kB inactive_anon:16672kB active_file:1720kB inactive_file:1
816kB unevictable:0kB isolated(anon):0kB isolated(file):0kB present:3129212kB managed:2884228kB mlocked:0kB dirty:0kB writeback:8kB mapped:688kB shmem:25972kB slab_reclaimable:8364kB slab
_unreclaimable:18136kB kernel_stack:9936kB pagetables:11188kB unstable:0kB bounce:0kB free_pcp:744kB local_pcp:0kB free_cma:0kB writeback_tmp:0kB pages_scanned:1992 all_unreclaimable? yes
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: lowmem_reserve[]: 0 0 4989 4989
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 Normal free:42864kB min:43120kB low:53900kB high:64680kB active_anon:4826580kB inactive_anon:62584kB active_file:4132kB inactive_file:
4140kB unevictable:0kB isolated(anon):0kB isolated(file):112kB present:5242880kB managed:5109528kB mlocked:0kB dirty:4kB writeback:4kB mapped:6812kB shmem:100824kB slab_reclaimable:20856k
B slab_unreclaimable:42452kB kernel_stack:21600kB pagetables:22196kB unstable:0kB bounce:0kB free_pcp:1228kB local_pcp:0kB free_cma:0kB writeback_tmp:0kB pages_scanned:10819 all_unreclaim
able? yes
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: lowmem_reserve[]: 0 0 0 0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 DMA: 1*4kB (U) 0*8kB 0*16kB 1*32kB (U) 2*64kB (U) 1*128kB (U) 1*256kB (U) 0*512kB 1*1024kB (U) 1*2048kB (M) 3*4096kB (M) = 15908kB
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 DMA32: 1309*4kB (UE) 635*8kB (UEM) 226*16kB (UEM) 186*32kB (UEM) 127*64kB (UEM) 62*128kB (UEM) 10*256kB (UE) 1*512kB (M) 6*1024kB (UM)
 0*2048kB 0*4096kB = 45164kB
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 Normal: 1500*4kB (UE) 319*8kB (UEM) 118*16kB (UEM) 184*32kB (UEM) 146*64kB (UEM) 83*128kB (UEM) 12*256kB (UE) 3*512kB (EM) 2*1024kB (E
M) 0*2048kB 0*4096kB = 42952kB
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Node 0 hugepages_total=0 hugepages_free=0 hugepages_surp=0 hugepages_size=2048kB
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: 34723 total pagecache pages
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: 0 pages in swap cache
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Swap cache stats: add 0, delete 0, find 0/0
Jan 16 02:34:34 tst-wj-dwhqjy-linux01 kernel: Free swap  = 0kB
                                                                                                                                                                    901,1         84%
```

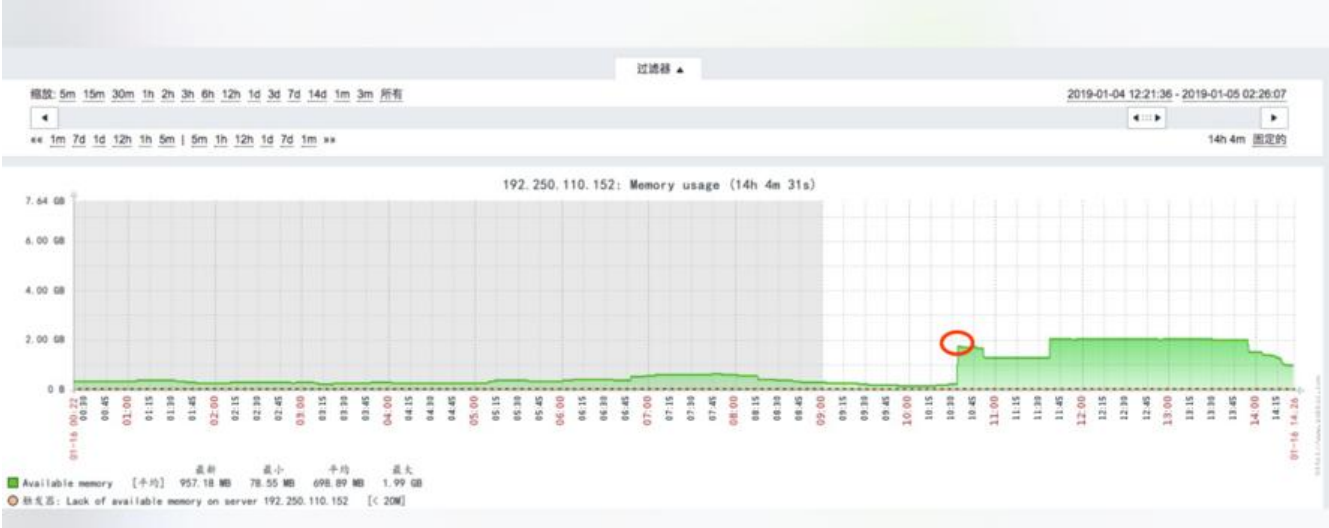
虽然也看不太懂（汗颜）

● 再次分析，通过zabbix上系统的内存监控，发现系统内存好像并没有超出系统最大值，因此上面的为是系统内存超出了而杀死的占用最大的内存进程，说法不一定是对的

(其实，zabbix上的时间内存飙升的那一个时间点刚好上02:34:34 + 8差不多这个时间)，

=>另外疑惑的是，在这一时刻，系统内存确实是在飙升的，但是根据监控图显示来说的系统内存也是有的，仍然有6G多的内存可供使用！一方面系统中被使用的内存确实突增，另一方面，确实使用也有爆掉整个系统的内存

=>更进一步地思考，观察zabbix中该系统使用内存的走势来说，进一步发现被使用的最多也就是达到G！因此，考虑：或者操作系统本身安全考虑导致的某个地方的配置内存上限，或者JVM中的内存最参数Xmx设置



- 然后，因此也有可能该sp这个应用本身的最大可用内存超了，即重新设置Xmx？？？？？？

原文链接：一次故障的查找记录