链滴

线上问题排查方法

作者：guobing
原文链接：https://ld246.com/article/1472285294159
来源网站：链滴

<pre class="vditor-yml-front-matter"><code class="language-yaml">###1. top命令详解
输入top命令之后，会打印出如下信息
```

top - 11:50:26 up 167 days, 19:07,  2 users,  load average: 0.00, 0.01, 0.05
Tasks: 100 total,   2 running,  98 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1016656 total,   930924 used,    85732 free,    12816 buffers
KiB Swap:        0 total,        0 used,        0 free.    50268 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND

   25 root      20   0       0      0      0 R  0.3  0.0  31:48.91 rcuos/0

14326 root      20   0  141092  33868   2336 S  0.3  3.3  78:35.81 AliHids

    1 root      20   0   49668   3108   1592 S  0.0  0.3   0:54.62 systemd

    2 root      20   0       0      0      0 S  0.0  0.0   0:00.05 kthreadd

    3 root      20   0       0      0      0 S  0.0  0.0   0:23.01 ksoftirqd/0

    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H

    7 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0

    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh

    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/0

   10 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/1

   11 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/2

   12 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/3

   13 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/4

   14 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/5

   15 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/6

   16 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/7

   17 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/8

   18 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/9

   19 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/10

   20 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/11

   21 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/12

   22 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/13
```

```
   23 root     20   0       0      0      0 S  0.0  0.0   0:00.00 rcuob/14

   24 root     20   0       0      0      0 S  0.0  0.0  16:00.34 rcu_sched

   26 root     20   0       0      0      0 S  0.0  0.0   0:00.00 rcuos/1

   27 root     20   0       0      0      0 S  0.0  0.0   0:00.00 rcuos/2
```
那我们一个个来说明这些信息到底是什么。
第一行，任务队列信息，具体参数如下：
`11:53:36`: 当前系统时间
`up 167 days`: 系统已经运行多长时间
`users `: 当前有几个用户登录
`load average` : load average后面的三个数分别是1分钟、5分钟、15分钟的负载情况。load averag数据是每隔5秒钟检查一次活跃的进程数，然后按特定算法计算出的数值。如果这个数除以逻辑CPU数量，结果高于5的时候就表明系统在超负荷运转了。

第二行，Tasks — 任务（进程），具体信息说明如下：
系统现在共有100个进程，其中处于运行中的有2个，98个在休眠（sleep），stoped状态的有0个，zmbie状态（僵尸）的有0个。

第三行，cpu状态信息，具体属性说明如下：
`5.9%us` — 用户空间占用CPU的百分比。
`3.4% sy` — 内核空间占用CPU的百分比。
`0.0% ni` — 改变过优先级的进程占用CPU的百分比
`90.4% id` — 空闲CPU百分比
`0.0% wa` — IO等待占用CPU的百分比
`0.0% hi` — 硬中断（Hardware IRQ）占用CPU的百分比
`0.2% si` — 软中断（Software Interrupts）占用CPU的百分比

第四行,内存状态，具体信息如下：
`32949016k total` — 物理内存总量（32GB）
`14411180k used` — 使用中的内存总量（14GB）
`18537836k free` — 空闲内存总量（18GB）
`169884k buffers` — 缓存的内存量 （169M）

第五行，swap交换分区信息，具体信息说明如下：
`32764556k total` — 交换区总量（32GB）
`0k used` — 使用的交换区总量（0K）
`32764556k free` — 空闲交换区总量（32GB）
`3612636k cached` — 缓冲的交换区总量（3.6GB）

第六行，空行。
第七行以下：各进程（任务）的状态监控，项目列信息说明如下：
`PID` — 进程id
`USER` — 进程所有者
`PR` — 进程优先级
`NI` — nice值。负值表示高优先级，正值表示低优先级
`VIRT` — 进程使用的虚拟内存总量，单位kb。VIRT=SWAP+RES
`RES` — 进程使用的、未被换出的物理内存大小，单位kb。RES=CODE+DATA
`SHR` — 共享内存大小，单位kb
`S` — 进程状态。D=不可中断的睡眠状态 R=运行 S=睡眠 T=跟踪/停止 Z=僵尸进程

`%CPU` — 上次更新到现在的CPU时间占用百分比
`%MEM` — 进程使用的物理内存百分比
`TIME+` — 进程使用的CPU时间总计，单位1/100秒
`COMMAND` — 进程名称（命令名/命令行）

###2. 线上问题排查
1. 首先使用TOP命令查看每个进程的情况.top命令前面已经讲的很清晰了.
2. 再使用Top的交互命令数字1查看每个CPU的性能数据。
3. 使用Top的交互命令H查看每个线程的性能信息。

在这里可能会出现三种情况：
>* 第一种情况，某个线程一直CPU利用率100%，则说明是这个线程有可能有死循环，那么请记这个PID。
>* 第二种情况，某个线程一直在TOP十的位置，这说明这个线程可能有性能问题。
>* 第三种情况，CPU利用率TOP几的线程在不停变化，说明并不是由某一个线程导致CPU偏高。

如果是第一种情况，也有可能是GC造成，我们可以用jstat命令看下GC情况,看看是不是因为持久代或老代满了，产生Full GC，导致CPU利用率持续飙高，命令如下。
`sudo /opt/java/bin/jstat -gcutil 31177 1000 5`,会打印出这些信息:
`S0 S1 E O P YGC YGCT FGC FGCT GCT`

我们还可以把线程Dump下来，看看究竟是哪个线程，执行什么代码造成的CPU利用率高。执行以下令，把线程dump到文件dump17里。
`sudo -u admin /opt/java/bin/jstack  31177 &gt; /home/innohub.dump/dump17`

dump出来内容的类似下面这段：
```
"http-0.0.0.0-7001-97" daemon prio=10 tid=0x000000004f6a8000 nid=0x555e in Object.wait(
 [0x0000000052423000]
   java.lang.Thread.State: WAITING (on object monitor)
        at java.lang.Object.wait(Native Method)
```
dump出来的线程ID（nid）是十六进制的，而我们用TOP命令看到的线程ID是10进制的，所以我们要rintf命令转换一下进制。然后用16进制的ID去dump里找到对应的线程。
```
printf "%x\n" 31558
 输出：7b46
```

还可以用的命令有:
jps : `java process status`,虚拟机进程状态
jstat : `java statistics monitoring tool`,收集虚拟机运行时数据
jmap : `memery map for java`,内存转储快照(headdump文件)
jhat : `JVM heap Dump browser`,分析headdump文件
jstack : `stack trace for java`,虚拟机的线程快照</code></pre>

参考链接:
http://ifeve.com/find-bug-online/
http://www.cnblogs.com/peida/archive/2012/12/24/2831353.html