**[Linux系统中负载较高问题排查思路与解决方法](https://www.cnblogs.com/liuyupen/p/13905967.html" \o "发布于 2020-10-31 13:37)**

Load 就是对计算机干活多少的度量，Load Average 就是一段时间（1分钟、5分钟、15分钟）内平均Load。

**一、Load分析：**

**情况1：CPU高、Load高**

1. 通过top命令查找占用CPU最高的进程PID；
2. 通过top -Hp PID查找占用CPU最高的线程TID;
3. 对于java程序，使用jstack打印线程堆栈信息（可联系业务进行排查定位）；
4. 通过printf %x tid打印出最消耗CPU线程的十六进制；
5. 在堆栈信息中查看该线程的堆栈信息；

**情况2：CPU低、Load高**

1. 通过top命令查看CPU等待IO时间，即%wa；
2. 通过iostat -d -x -m 1 10查看磁盘IO情况；(安装命令 yum install -y sysstat)
3. 通过sar -n DEV 1 10查看网络IO情况；
4. 通过如下命令查找占用IO的程序；

ps -e -L h o state,cmd | awk '{if($1=="R"||$1=="D"){print $0}}' | sort | uniq -c | sort -k 1nr

**二、CPU高、Load高情况分析**

* 使用vmstat 查看系统纬度的 CPU 负载；
* 使用 top 查看进程纬度的 CPU 负载；

**2.1、使用 vmstat 查看系统纬度的 CPU 负载**

可以通过 vmstat 从系统维度查看 CPU 资源的使用情况

格式：vmstat -n 1 -n 1 表示结果一秒刷新一次

[root@k8s-10 ~]# vmstat -n 1

procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----

r b swpd free buff cache si so bi bo in cs us sy id wa st

1 1 0 2798000 2076 6375040 0 0 10 76 10 49 6 2 91 1 0

0 0 0 2798232 2076 6375128 0 0 0 207 7965 12525 7 2 90 2 0

返回结果中的主要数据列说明：

* **r**： 表示系统中 CPU 等待处理的线程。由于 CPU 每次只能处理一个线程，所以，该数值越大，通常表示系统运行越慢。
* **b**： 表示阻塞的进程,这个不多说，进程阻塞，大家懂的。
* **us**： 用户CPU时间，我曾经在一个做加密解密很频繁的服务器上，可以看到us接近100,r运行队列达到80(机器在做压力测试，性能表现不佳)。
* **sy**： 系统CPU时间，如果太高，表示系统调用时间长，例如是IO操作频繁。
* **wa**：IO 等待消耗的 CPU 时间百分比。该值较高时，说明 IO 等待比较严重，这可能磁盘大量作随机访问造成的，也可能是磁盘性能出现了瓶颈。
* **id**：处于空闲状态的 CPU 时间百分比。如果该值持续为 0，同时 sy 是 us 的两倍，则通常说明系统则面临着 CPU 资源的短缺。

**常见问题及解决方法：**

* 如果r经常大于4，且id经常少于40，表示cpu的负荷很重。
* 如果pi，po长期不等于0，表示内存不足。
* 如果disk经常不等于0，且在b中的队列大于3，表示io性能不好。

**2.1、使用 top 查看进程纬度的 CPU 负载**

可以通过 top 从进程纬度来查看其 CPU、内存等资源的使用情况。

[root@k8s-10 ~]# top -c

top - 19:53:49 up 2 days, 7:57, 3 users, load average: 0.76, 0.79, 0.58

Tasks: 282 total, 2 running, 280 sleeping, 0 stopped, 0 zombie

%Cpu(s): 2.4 us, 1.4 sy, 0.0 ni, 95.0 id, 1.2 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 12304204 total, 2800864 free, 3119064 used, 6384276 buff/cache

KiB Swap: 0 total, 0 free, 0 used. 8164632 avail Mem

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

29884 root 20 0 5346580 929332 14556 S 0.0 7.6 6:19.19 /opt/jdk1.8.0\_144/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apach+

875 root 20 0 729524 563424 38612 S 3.1 4.6 93:22.70 kube-apiserver --authorization-mode=Node,RBAC --service-node-port-range=80-60000 --advertise-address=10.68.7.162 --allow-privileged=true -+

3870 nfsnobo+ 20 0 910376 317248 22812 S 1.6 2.6 42:29.59 /bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/prometheus --storage.tsdb.retention=1d --web.enable-life+

默认界面上第三行会显示当前 CPU 资源的总体使用情况，下方会显示各个进程的资源占用情况。

可以直接在界面输入大小字母 P，来使监控结果按 CPU 使用率倒序排列，进而定位系统中占用 CPU 较高的进程。最后，根据系统日志和程序自身相关日志，对相应进程做进一步排查分析，以判断其占用过高 CPU 的原因。

**2.2、strace命令分析**

<https://oa.kedacom.com/confluence/pages/viewpage.action?pageId=77136289>

**三、CPU低、Load高情况分析**

**问题描述**：  
Linux 系统没有业务程序运行，通过 top 观察，类似如下图所示，CPU 很空闲，但是 load average 却非常高：

**处理办法**：

* load average 是对 CPU 负载的评估，其值越高，说明其任务队列越长，处于等待执行的任务越多。
* 出现此种情况时，可能是由于僵死进程导致的。可以通过指令ps -axjf查看是否存在 D 状态进程。
* D 状态是指不可中断的睡眠状态。该状态的进程无法被 kill，也无法自行退出。只能通过恢复其依赖的资源或者重启系统来解决。

等待I/O的进程通过处于uninterruptible sleep或D状态；通过给出这些信息我们就可以简单的查找出处在wait状态的进程

ps -eo state,pid,cmd | grep "^D"; echo "----"

- 查找占用IO的程序

ps -e -L h o state,cmd | awk '{if($1=="R"||$1=="D"){print $0}}' | sort | uniq -c | sort -k 1nr

分类: [运维开发](https://www.cnblogs.com/liuyupen/category/1873615.html)