第一节 CPU性能

CPU性能

运行队列

每个CPU都会维持一个运行队列，理想情况下，调度器会不断让队列中的进程运行。进程不是处在sleep状态就是runable状态。如果CPU过载，就会出现调度器跟不上系统的要求，导致可运行的进程会填满队列。队列愈大，程序执行时间就愈长。
系统平均负载被定义为在特定时间间隔内运行队列中(在CPU上运行或者等待运行多少进程)的平均进程树，通常通过load average体现系统的平均负载。

Load Average

CPU的 Load，它所包含的信息不是 CPU的使用率状况，而是在一段时间内 CPU正在处理以及等待 CPU处理的进程数之和的统计信息，也就是CPU运行队列长度的统计信息。Load Average是反应系统压力的一个很重要的指标。
查看Load Average的常用命令有：

- Top：top - 15:13:55 up 39 days, 23:49, 9 users, load average: 7.17, 7.37, 6.22
- Uptime：15:14:17 up 39 days, 23:49, 9 users, load average: 6.77, 7.26, 6.21

Top/uptime命令中load average显示的是最近1分钟、5分钟和15分钟的系统平均负载。评估系统性能时，不仅仅看load average的数值，而且应该结合系统的CPU核数，系统的负载均值是基于内核的数量决定的，例如该服务器有24核，则单核的CPU负载为7/24。
依据经验分析，单核CPU负载<2时，系统性能是良好的，当单核CPU负载>5时，那么就表明这个机器存在严重的性能问题。

用户态user 系统态system

在CPU的所有指令中，有一些指令是非常危险的，如果错用，将导致整个系统崩溃。比如：清内存、设置时钟等。所以，CPU将指令分为特权指令和非特权指令，对于那些危险的指令，只允许操作系统及其相关模块使用，普通的应用程序只能使用那些不会造成灾难的指令。
系统态（内核态）：当一个任务（进程）执行系统调用而陷入内核代码中执行时，我们就称进程处于内核运行态（或简称为内核态）。此时处理器处于特权级最高的（0级）内核代码中执行。当进程处于内核态时，执行的内核代码会使用当前进程的内核栈。每个进程都有自己的内核栈。
用户态：当进程在执行用户自己的代码时，则称其处于用户运行态（用户态）。即此时处理器在特权级最低的（3级）用户代码中运行。
用户态与内核态的区别：

1. 内核态可以执行任何指令程序，用户态只能执行特定指令
2. 内核态代码可以自由访问任何有效地址，用户态的代码则要受到处理器的诸多检查，它们只能访问映射其地址空间的页表项中规定的在用户态下可访问页面的虚拟地址

用户态与内核态转换：
在内核态下CPU可执行任何指令，在用户态下CPU只能执行非特权指令。当CPU处于内核态，可以随意进入用户态；
而当CPU处于用户态时，用户从用户态切换到内核态只有在系统调用和中断两种情况下发生，一般程序一开始都是运行于用户态，当程序需要使用系统资源时，就必须通过调用软中断进入内核态。
CPU使用率可以通过Top，vmstat，mpstat等命令查看，其中us(r)%表示用户态使用的CPU百分比，sy(s)%表示内核态使用的CPU百分比：

- TOP：

```
%Cpu(s):  0.9 us,  0.6 sy,  0.0 ni, 98.4 id,  0.1 wa,  0.0 hi,  0.0
si,  0.0 st
```

- Vmstat：

```
procs -----------memory---------- ---swap-- -----io---- -system--
----cpu----
r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy
id wa
1  0 1061592 15986 395216 91660    0    0     3     9    0    0  5  1
94  0
```

- Mpstat：

```
CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest
%idle
  all    0.92    0.00    0.50    0.17    0.00    0.00    0.00    4.14
94.27
```

程序用户态、内核态分布：
如果CPU在满负荷运行，应该符合下列分布，
a) User Time：65%～70%
b) System Time：30%～35%
c) Idle：0%～5%
http://www.cnblogs.com/cobbliu/archive/2012/11/07/2758184.html
http://blog.csdn.net/skywalkzf/article/details/5185442

上下文切换（Content Switch）

每个进程都会分到CPU的时间片来运行，当一个进程用完时间片或者被更高优先级的进程抢占后，它会*备份*到CPU的运行队列中，同时其他进程在CPU上运行。这个进程切换的过程被称作上下文切换。过多的上下文切换会造成系统很大的开销。
上下文切换过程中执行的操作：

- 挂起一个进程，并储存该进程当时在内存中所反映出的状态
- 从内存中恢复下一个要执行的进程，恢复该进程原来的状态到寄存器，返回到其上次暂停的执行代码然后继续执行

Context
Switch大体上由两个部分组成：中断和进程(包括线程)切换，一次中断（Interrupt）会引起一次切换，进程（线程）的创建、激活之类的也会引起一次切换。
上下文切换可以通过vmstat命令查看：

- Vmstat中的cs字段表示每秒产生的上下文切换次数：

```
procs -----------memory---------- ---swap-- -----io---- -system--
----cpu----
r  b   swpd   free   buff  cache   si   so    bi    bo   in
{color:#ff0000}cs{color} us sy id wa
1  0 1061592 15986 395216 91660    0    0     3     9    0
{color:#ff0000}0{color}  5  1 94  0
```

对于上下文切换要结合CPU使用率来看，如果CPU使用满足上面所讲的分布，大量的上下文切换也是可以接受的。
http://www.6san.com/469/
http://space.itpub.net/24435147/viewspace-694469

中断interrupt

中断是指由于接收到来自外围硬件（相对于中央处理器和内存）的异步信号或来自软件的同步信号，而进行相应的硬件/软件处理。发出这样的信号称为进行中断请求（interrupt request，IRQ）。
硬件中断：外围硬件发给CPU或者内存的异步信号就称之为硬中断，简言之就是外设对CPU的中断。
软件中断：由软件系统本身发给操作系统内核的中断信号，称之为软中断。通常是由硬中断处理程序或进程调度程序对操作系统内核的中断，也就是我们常说的系统调用(System Call)。
硬中断与软中断之区别与联系：

1. 硬中断是有外设硬件发出的，需要有中断控制器之参与。其过程是外设侦测到变化，告知中断控制器，中断控制器通过CPU或内存的中断脚通知CPU，然后硬件进行程序计数器及堆栈寄存器之现场保存工作（引发上下文切换），并根据中断向量调用硬中断处理程序进行中断处理。
2. 软中断则通常是由硬中断处理程序或者进程调度程序等软件程序发出的中断信号，无需中断控制器之参与，直接以一个CPU指令之形式指示CPU进行程序计数器及堆栈寄存器之现场保存工作(亦会引发上下文切换)，并调用相应的软中断处理程序进行中断处理(即我们通常所言之系统调用)。
3. 硬中断直接以硬件的方式引发，处理速度快。软中断以软件指令之方式适合于对响应速度要求不是特别严格的场景。
4. 硬中断通过设置CPU的屏蔽位可进行屏蔽，软中断则由于是指令之方式给出，不能屏蔽。
5. 硬中断发生后，通常会在硬中断处理程序中调用一个软中断来进行后续工作的处理。
6. 硬中断和软中断均会引起上下文切换(进程/线程之切换)，进程切换的过程是差不多的。

查看中断的命令：

- Linux下中断来源可以从 /proc/interrupts 中了解到:
- 软中断可以从 /proc/softirqs/ 中了解到:
- 总的中断次数可以从vmstat或者dstat了解到，vmstat中的in参数表示每秒中的中断次数:

```
procs -----------memory---------- ---swap-- -----io---- -system--
----cpu----
r  b   swpd   free   buff  cache   si   so    bi    bo
{color:#ff0000}in{color}   cs us sy id wa
1  0 1061592 15986 395216 91660    0    0     3     9
{color:#ff0000}0{color}    0  5  1 94  0
```

http://blog.csdn.net/pxz_002/article/details/7327668
http://blog.yufeng.info/archives/1062

指标汇总分析

指标名称	监控方式	参考阀值	瓶颈分析
Load Average	top uptime	小于CPU个数的2倍	
%urs	top Vmstat Mpstat	<75%	用户态使用CPU过多，需要优化用户程序
%sys		<30%	系统态使用CPU过多，可能是过多的中断以及上下文切换导致
%id		>5%	
cs	Vmstat dstat		
in			