Sunday, March 16. 2008

Less known Solaris Features: Resource Management - Part 4: Limiting CPU usage

Okay, but your system wasn´t purchased for students at night. You´ve bought it for computational science at night. The Large Hadron Collider project spend 75% of the costs, and the Small Hadron Collider project spend 25%. Thus the compute power should be distributed in a similar manner.
Without Resource MangementBut when you start two compute intensive processes, both get roundabout 50 percent of processor. Okay, we have an super-duper simulation script called /opt/bombs/cpuhog.pl:

```
#! /usr/bin/perl
while (1) { my $res = ( 3.3333 / 3.14 ) }
```

Let´s login as user einstein:

```
# su einstein
Password:
$ /opt/bombs/cpuhog.pl &
$ /opt/bombs/cpuhog.pl &
```

After a few moments the system will stabilize at aprox. 50% CPU resources for both processes. Just look at the first column:

```
bash-3.2$ ps -o pcpu,project,args
%CPU  PROJECT COMMAND
 0.0 user.einstein -sh
 0.3 user.einstein bash
47.3 user.einstein /usr/bin/perl /opt/bombs/cpuhog.pl
48.0 user.einstein /usr/bin/perl /opt/bombs/cpuhog.pl
 0.2 user.einstein ps -o pcpu,project,args
```

This isn´t the intended behaviour.

Using the Fair Share SchedulerSolaris can solve this problem. There is something called Fair Share Scheduler in Solaris. This scheduler is capable to distribute the compute power of the system based on a configuration. The FFS isn´t activated by default. You have to login as root to activate it:

```
# dispadmin -d FSS
```

There are ways to enable this scheduler with a running system, but it´s easier to reboot the system now.

When the system has startet, we get root provileges by using su. At first we create an additional project for the SHC project. We created the other project (lhcproject) before:

```
# projadd shcproject
#projmod -U einstein shcproject
```

Now we configure an resource limit on the projects.

```
# projmod -K
"project.cpu-shares=(privileged,150,none)" lhcproject
# projmod -K "project.cpu-shares=(privileged,50,none)" shcproject
```

. We´ve used the resource control project.cpu-shares. With this control we can assign an amount of CPU power to an project. We´ve defined an privileged limit, thus only root can change this limit later on.

Shares Okay, what is the meaning of these numbers 150 and 50 in the last commands ? Where are the 25% and the 75%? Well, the resource management isn´t configured with percentages, it´s configured in a unit called shares.

It´s like with stocks. The person with the most stocks owns most of the company. The project with the most shares owns most of the CPU. In our example we divided the CPU in 200 shares. Every share represents 1/200 of the CPU. Project shcproject owns 50 shares. Project lhcproject owns 150. I think, you already saw it: 150 is 75% of 200 and 50 is 25% of 200. Here we find our planed partitioning of the CPU we´ve planed before. By the way: I deliberatly choose 150/50 instead of 75/25 to show you that these share definitions are not scaled in percent.

Okay, but what happens when you add a third project and you give this project 200 shares (For example because a new project gave money for buying another processor board). Then the percentages are different. In total you have 400 shares on the system. The 200 shares of the new project are 50%, thus the project gets 50% of the compute power. The 150 shares of the lhcproject are 37.5 percent. This project gets 37.5 of the computing power. And the 50 shares of the shcproject are now 12.5 percent and thus the project get this part of the CPU power.

Behaviour of processes with Resource ManagementOkay, now let´s start a process unter the control of the new resource limitation. Login as user einstein and type the following command:

```
$ newtask -p shcproject
/opt/bombs/cpuhog.pl &
```

We start at first only one of the cpuhog.pl processes .

```
bash-3.2$ ps -o pcpu,project,args
%CPU  PROJECT COMMAND^
 0.0 user.einstein -sh
 0.3 user.einstein bash
 0.2 user.einstein ps -o pcpu,project,args
```

```
95.9 shcproject /usr/bin/perl /opt/bombs/cpuhog.pl
```
Wait ... the process gets 95.9 percent? An error ? No. It makes no sense to slow down the process, when there is no other process needing the compute power.

Now we start the second process, this time as a task in the lhcproject:
```
bash-3.2$ newtask -p lhcproject /opt/bombs/cpuhog.pl &
[2] 784
```
We look in the process table again.
```
bash-3.2$ ps -o pcpu,project,args
%CPU  PROJECT COMMAND
 0.0 user.einstein -sh
 0.1 user.einstein bash
72.5 lhcproject /usr/bin/perl /opt/bombs/cpuhog.pl
25.6 shcproject /usr/bin/perl /opt/bombs/cpuhog.pl
 0.2 user.einstein ps -o pcpu,project,args
```
Voila, each of our compute processes get their configured part of the compute power. It isn´t exactly 75%/25% all the time but in average the distribution will be this way.

A few days later, the Dean of the department comes into the office and tells you that we need the results of the SHC project earlier, as important persons want to see them soon to spend more money. So you have to change the ratio of the shares. We can do this without restarting the processes at runtime. But as we´ve defined the limits as privileged before, we have to login as root:
```
# prctl -n project.cpu-shares -r -v 150 -i project shcproject
# prctl -n project.cpu-shares -r -v 50 -i project lhcproject
```
Let´s look after the processes again after a few moments:
```
bash-3.2$ ps -o pcpu,project,args
%CPU  PROJECT COMMAND
 0.0 user.einstein -sh
 0.1 user.einstein bash
25.7 lhcproject /usr/bin/perl /opt/bombs/cpuhog.pl
72.9 shcproject /usr/bin/perl /opt/bombs/cpuhog.pl
 0.2 user.einstein ps -o pcpu,project,args
```
The ratio has changed to new settings. It´s important to know that only the settings in /etc/projects is boot-persistent. Everything you set via prctl is lost at the boot.

Posted by Joerg Moellenkamp in English at 22:48