Using Solaris™ Resource Manager With Sun Ray™

Marcel Guerin, Systems Engineer, Global Sales Organization (GSO)

Sun BluePrints™ OnLine—June 2004

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-6617-10
Revision .01, 5/27/04
Edition: June 2004

Please
Recycle

Using Solaris™ Resource Manager With Sun Ray™

This article describes best practices for managing system resources for Sun Ray™ users. Based on an actual customer scenario, the solution in this article provides recommendations for integrating and using the Solaris™ Resource Manager software to distribute system resources fairly when users insert and remove their smart cards from Sun Ray desktop units (DTUs). Sample scripts associated with this article are available from the Sun BluePrints™ web site via the Sun Download Center (SDLC) download service.

This article does not provide introductions to the Sun Ray DTU and the Solaris Resource Manager software. For an introduction to Sun Ray, refer to the *Sun Ray Server Software 2.0 Administrator's Guide*, available from Sun's web site. For an introduction to Solaris Resource Manager software, refer to Sun Blueprints OnLine article titled "Resource Management in the Solaris 9 Operating Environment."

This article contains the following topics:

Understanding the Background and Case Scenario

It's no fun sharing a room with a slob. Inevitably, the person’s mess creeps into your side of the room. Likewise, when consolidating potentially hundreds of users onto a single Sun Ray server, run-away processes and users who use too much of system resources can diminish the desktop experience for all users. Streaming media to a Sun Ray could consume a significant portion of a 400-MegaHertz processor. Unfair! What can be done?

Users should be prevented from consuming more than their fair share of system resources. Furthermore, users sitting in front of their Sun Ray desktops should have higher priority than those not at their desktops, so that they have the resources they need when they need them.

We recently solved this problem when providing session mobility for the Internet Café of a customer's annual sales management conference. We provided session mobility for about 600 users on 70 Sun Rays. In essence, the solution required Solaris Resource Manager software constraints to be modified whenever users insert or remove their smart cards from a Sun Ray DTU. Key elements of the solution include:

- A functional Sun Ray 2.0 server, running Solaris 9 Operating Environment.
- The Fair Share Scheduler (FSS) as the default scheduling class for the system.
- A project in `/etc/project` for each user, with certain resource constraints.
- Scripts to run when users insert or remove their cards.
- Modifications to the default X session configuration.

Integrating Solaris Resource Manager Software

Integrating Solaris Resource Manager software with Sun Ray requires the Fair Share Scheduler (FSS) as the system-wide default scheduler. The following method requires a reboot, although the reboot is not strictly necessary.

Note – The process for migrating a running system into the FSS without rebooting is documented in the Sun Blueprints OnLine article titled “Resource Management in the Solaris 9 Operating Environment.” Also documented in the “Resource Management in the Solaris 9 Operating Environment” article are important details about when to run other scheduling classes simultaneously with the FSS.

```
Sun Ray1# dispadmin -d FSS

Sun Ray1# init 6
```

Aggregate User Processes

Users initiate many processes during their normal workflow. For example, a simple GNOME session typically employs about forty processes. While these processes work together to present users with an integrated desktop environment, Solaris Operating System (Solaris OS) treats each process as an independent entity. To meaningfully control consumption of key system resources, the system must treat all of the processes belonging to a single user as a single entity.

In Solaris Resource Manager software, the project container provides a convenient mechanism for aggregating user processes. Although technically a project might contain the processes of multiple users, placing each user within a separate project enables the defined resource constraints to individually operate on users.

The simplest way to define a project is to add it to the local project repository `/etc/project`. Each Sun Ray user needs a dedicated project, that is, a project for which the user is the sole user. By convention, projects for users are prefixed with user (`user-dot`).

```
Sun Ray1# projadd -c "Smith, Mary" -U ms12345 -p 84855 user.ms12345
```

The parameters to `projadd` have a similar context to those of `useradd`:

- `-c` argument specifies a comment.
- `-U` designates the usernames belonging to the project.
- `-p` gives a unique identifier for the project (similar to a UID).
- Like `useradd`, the last parameter denotes a common name for the project entry.

By default a project is not resource constrained. However, common Solaris utilities such as `prstat`(1M) can aggregate their output by project.

To add resource controls to the project entry in `/etc/project`, edit the file and append the resource controls without line breaks to the appropriate entries. An automated script is provided to accomplish this without manually editing `/etc/project`.

Control CPU Shares

Controlling the CPU shares of Sun Ray users requires the presence of a `project.cpu-shares` constraint. Users should not be able to modify the number of CPU shares allotted to their projects, and the share limit should be enforced when a project hits the threshold. Appending the following to a project entry specifies these limits.

```
project.cpu-shares=(privileged,1,deny)
```

Generically, resource controls are specified by triplets (type, value, and action). The triplet in the code example `(privileged,1,deny)` sets a control on the project's CPU shares with a value of 1 (1 share) that denies further shares when the 1 share threshold is reached; superuser (or privileged) permissions are required to change this value. Refer to the project manual page `project`(4) for more details

Multiple resource constraints of type `project.cpu-shares` might exist for a single project. However, this solution assumes that only a single `project.cpu-shares` constraint exists for each project; additional constraints on `cpu-shares` might lead to unexpected results.

The complete line in `/etc/project` should look similar to the following. Note that the entry should be constrained to a single line in the file, without line continuation.

```
user.mary:12345:Smith, Smith:marys::project.cpu-
shares=(privileged,1,deny)
```

Now when the individual Mary Smith (username marys) logs into the system, or starts a process via cron, the `user.marys` project will have 1 share of the CPU. If Smith has existing processes, they will inherit this limit.

Using Scripts to Allocate Shares Automatically

With only a little scripting, we can automatically increase a user's shares whenever the user inserts a smart card into a Sun Ray DTU. The Sun Ray command `utaction` is the mechanism for specifying commands to execute when users insert or remove cards at a DTU. With `utaction`, administrators can either specify a time delay for the action, or alternatively specify that the action should be taken immediately.

An unmodified Sun Ray server installation starts two `utaction` processes for each card user when a user removes the card. The first process immediately locks the screen via `utxlock`. (This can be modified to specify a time delay, for example, two minutes, so if a user re-inserts the card within two minutes of removing it, the user does not need to unlock the screen.) The second process performs housecleaning after a user logs out of a session.

Because `utaction` operates on card insert and remove events, it cannot be used to control non-card Sun Ray users. However, with the resource constrained project(s) just created, the commands specified to `utaction` increase or decrease a user's CPU shares appropriately for card-based users.

Let's start with the scripts that actually increase or decrease a user's CPU shares. For now, assume the `utaction` command was invoked properly, and it provides us with the user name (sometimes referred to as the user ID, however, not the numeric ID). The `connectme.sh` and `disconnectme.sh` scripts operate on this user name, accepted as a single argument.

When a user inserts a smart card, we want to run the `connectme.sh` script.

```
#!/bin/sh
# M. Smith, December 2003
# connectme.sh - when a user inserts their card, increase
# the number of CPU shares for their project
#
# requires $1 to be a valid username, and
# a user.$1 project in /etc/project, with a project.cpu-shares
constraint

prctl -t privileged -e deny -n project.cpu-shares -r -v 50 -i
project user.$1
```

When user Smith inserts her card, or connects to her Sun Ray session, this script sets the CPU shares for her project to 50. The call to the `prctl` command replaces (`-r`) the first resource control (identified by the `-n` argument, here `project.cpu-shares`) with the new triplet specified by the type (`-t`), value (`-v`), and action (`-e`). The final arguments to `prctl` indicate to what entity the new resource control should apply, namely, a project entity (`-i project`) identified by `user.$1` (in this example, `user.marysmarys`).

Similarly, when Smith removes her card from a Sun Ray, her shares should return to 1, as shown in the following `disconnectme.sh` script.

```
#!/bin/sh
# M. Smith, December 2003
# disconnectme.sh - when a user removes their card, decrease
# the number of CPU shares for their project
#
# requires $1 to be a valid username, and
# a user.$1 project in /etc/project, with a project.cpu-shares
constraint

prctl -t privileged -e deny -n project.cpu-shares -r -v 1 -i
project user.$1
```

Can you see the difference between the two scripts? The value (`-v`) argument to the `prctl` command is the only difference. Therefore, we can consolidate both into a single script named `adjustshares.sh` and pass a new argument specifying the number of shares (depending on whether the user inserts or removes the card).

```
#!/bin/sh
# M. Smith, February 2004
# adjustshares.sh - adjust a user's shares whenever they insert or
# remove their smartcard from a Sun Ray
#
# requires $1 to be a valid username,
# requires $2 to be a valid number of cpu-shares for the user's
project, and
# a user.$1 project in /etc/project, with a project.cpu-shares
constraint

prctl -t privileged -e deny -n project.cpu-shares -r -v $2 -I
project user.$1
```

Now that the `adjustshares.sh` script is defined, we turn our attention to invoking `utaction`. For controlling a user's CPU shares, the `utaction` process must be active throughout the lifetime of the desktop session.

Preventing Users From Modifying Share Limits

To prevent a user from circumventing the `utaction` process and keeping shares increased even though the card is removed, `utaction` must be run as root user. Solaris OS provides a handy mechanism for doing this. When a user successfully authenticates to the system, the system must perform certain tasks that the user does not normally have permission to do. For example, when a user logs into a system console, the system grants to the user read and write permissions for the console device. Similarly, these changes must be undone when the user logs out. These changes are symmetrically made in `Xstartup` and `Xreset`.

Note – An additional script, `Xsetup`, can be used to set up the X server before the authentication screen is displayed. Additional files exist; refer to the `FILES` section of the `dtlogin`(1X) manual page for more details.

FIGURE 1 depicts the time line of a user session.

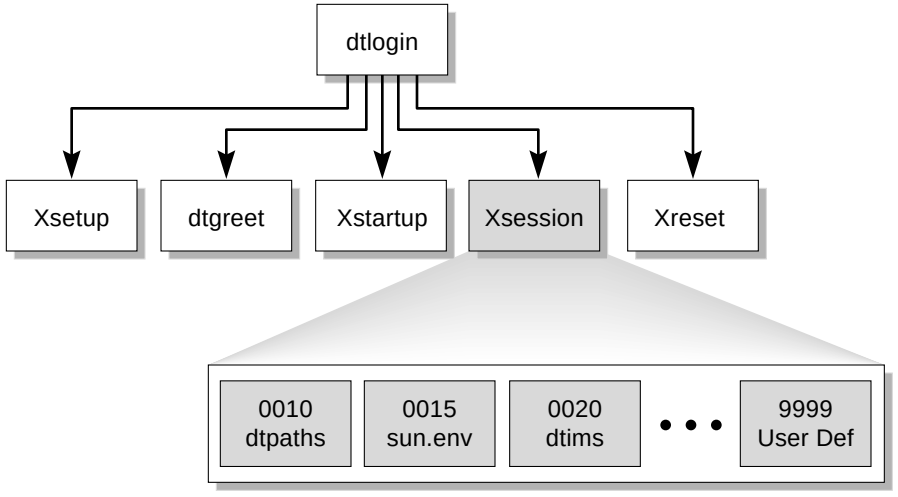

FIGURE 1 Structure of an X11 Session

In FIGURE 1, the grey boxes represent functions that execute with root permissions immediately before and after a user's session. The system default `Xstartup` and `Xreset` scripts are maintained in `/usr/dt/config`.

Preserving Share Limits Before Patches and Upgrades

Because the scripts in `/usr/dt/config` might be changed by patches or Solaris upgrades, it is considered best practice to copy these files into `/etc/dt/config`, and modify them there. Be sure to make the script executable, otherwise the X session will not start correctly (the `dtlogin` screen will reappear).

Alternatively, the default scripts in `/usr/dt/config` could be sourced from the new scripts in `/etc/dt/config`, which would automatically incorporate any changes to the default scripts.

The `dtlogin` process searches the `/etc/dt/config` directory first, and uses the scripts there if they exist with the proper permissions. The `/etc/dt/config` does not exist by default, so you might need to create it.

```
Sun Ray# mkdir -p /etc/dt/config
Sun Ray# cp /usr/dt/config/Xstartup /etc/dt/config
Sun Ray# cp /usr/dt/config/Xreset /etc/dt/config
```

For simplicity, this article sources the default scripts `/etc/dt/config/Xstartup`.

```
#!/bin/ksh

# the corresponding script in /usr may be modified by patches, etc.
# get the latest version
. /usr/dt/config/Xstartup

# then invoke utaction (place in background)
/opt/SUNWut/bin/utaction -i -c "adjustshares.sh $USER 50" -d
"adjustshares.sh $USER 1" &
```

Adjusting Background Processor Shares

The `utaction` command terminates when the user's session is destroyed; it does not run the disconnect command before doing so. If the user logs out, or the session is reaped according to administrative policies, the user still has 50 shares. To rectify this, run the `adjustshares.sh` script in the `Xreset` script.

Now any background processes started by the user will have fewer shares of the processor `/etc/dt/config/Xreset`.

```
#! /bin/ksh

# the corresponding script in /usr may be modified by patches, etc.
# get the latest version
. /usr/dt/config/Xreset

# then invoke the adjust shares script
adjustshares.sh $USER 1
```

About the Author

Marcel Guerin joined the Global Sales Organization in 1998 as a Systems Engineer, providing technical pre-sales advice for major oil and gas companies. Marcel has set up and assisted with several Sun Ray trade-show implementations. In the course of his role as a pre-sales engineer, he helped Sun customers improve application performance, evaluate upcoming hardware, and learn about and adopt Sun technologies.

Marcel has advised customers on technical issues ranging from multi-threaded programming to architecture and implementation of their SPARC systems, in environments ranging from SAP to performance-sensitive database systems.

Prior to joining Sun, Marcel provided programming and application integration support to a Houston-based energy company. Marcel received his Bachelor of Science in computer science from the University of Houston in 1994.

Acknowledgements

The author would like to recognize Craig Bender, Staff Engineer, SunONE Desktop, Sun Ray Engineering group, for his contribution to the content of this article.

Related Resources

- Lawson, Stuart. "Resource Management in the Solaris 9 Operating Environment" Sun BluePrints OnLine, September 2002, `http://www.sun.com/solutions/blueprints/816-7753-10.pdf`.
- *Sun Ray Server Software 2.0 Administrator's Guide*, Sun Microsystems, Inc., Part Number 816-6753-10, February 2003, `http://www.sun.com/products-n-solutions/hardware/docs/pdf/0902/816-6753.pdf`.
- *System Administration Guide: Resource Management and Network Services*, from *Solaris 9 System Administrator Collection*, Part Number 806-4076-10, `http://docs.sun.com`.

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`