User Guide and Manual for Project Canary

"Miners used a canary in a cage to detect toxic gases. If the canary stopped singing, the miners fled the caves. This was one of the early predictive monitoring systems."

Last Modified: 1-April-2005

Copyright and Licensing Information

Table of Contents

Chapter 1: Project Canary Overview

Project Canary was originally created to show where load comes from on a Sun Ray Server so that systems administrators could watch for trends and correct performance problems before they developed into larger issues. Project Canary software has also proved capable of exposing load issues on any server running the Solaris Operating System, from version 2.6 through version 10 on both SPARC and x86 platforms as well as machines running the Linux Operating System.

If you do not have Project Canary software installed, please refer now to the *Project Canary Installation and Configuration Guide* and follow the installation instructions. This document, the *User Guide and Manual for Project Canary* will prove worthless without having a working installation of Project Canary software to refer to.

1.1: Design Criteria

The design criteria initially used to create Project Canary software included, but was not limited to:

- No formal installation of software on machines to be monitored
- No requirement to create special accounts on machines to be monitored
- The same software will work without modification on multiple operating systems, hardware platforms, and machine types
  - Currently, Project Canary software works without modification on clients running the Solaris Operating System (both SPARC and x86) and is confirmed to run on and is supported on two flavors of the Linux Operating System, SuSE and Red Hat.
- The software will not interfere with the function of any other client or application software running on a client being monitored
- Root privileges are not required to gather data from machines being monitored
- Crontab entries are not required to run the software on client machines being monitored
- The software must match performance against many of the SE-Toolkit rules so that users can take advantage of those suggestions and recommendations
- The software must display the load generated by Web Browsers, GNOME sessions, the Xsun server, Java applets, and StarOffice sessions, specifically.
- The software must provide network performance graphs showing Input and Output packets per second, error codes, and other statistics and data useful when examining network throughput and performance

1.2: How Project Canary software gathers data from machines it monitors

Servers are probed using a shell script that each machine monitored executes locally. For machines running the Solaris Operating System, this shell script is called `canary_solaris.sh` and for machines running the Linux Operating System, it is called `canary_linux.sh`

Both shell scripts are included with Project Canary software.

The `canary_solaris.sh` and `canary_linux.sh` scripts execute the following 48 commands on every client machine monitored by Project Canary software as shown on the next page:

Data Gathering (continued)

```
1. /usr/bin/showrev
2. /usr/sbin/psrinfo -v | grep operates | sort | uniq -c
3. /usr/sbin/prtconf -p -v | grep banner-name:
```

```
 4. /usr/bin/uptime
 5. /usr/bin/prstat -c -n 9999,99 1 1 | nawk -f prstat1.awk | sort | ...
 6. /usr/bin/ps -eo s,comm,user,pid | egrep ^O|^R | sort
 7. cleanRunaways -m 3 -r -t 2
 8. /usr/bin/prstat -c -n 99,999 -a 1 1
 9. /usr/bin/prstat -c -n 1,1 -Z 1 1
10. /usr/bin/prstat -m -c -n 9999,999 1 1 | nawk -f p_m.awk |sort
11. /usr/sbin/swap -l
12. /usr/bin/netstat -i
13. /usr/bin/netstat -s -P tcp
14. /usr/bin/netstat -s -P udp
15. /usr/bin/netstat -k | nawk ...
16. /usr/bin/gconftool-2 ...
17. shmlist
18. /usr/bin/grep ^set /etc/system
19. /usr/bin/cat /etc/auto_master
20. /usr/bin/grep automount /etc/init.d/autofs
21. /usr/bin/ypcat -k auto.master
22. /usr/bin/grep AUTOMOUNT /etc/default/autofs
23. /usr/bin/cat /etc/release
24. /usr/bin/netstat -anP tcp -f inet | nawk '{print $NF}' | sort |  ...
25. /usr/bin/iostat -xn 30 2 | nawk '{ if ($10 >4) print $0; if ( $1 ...
26. /usr/bin/mpstat 30 2
27. /usr/bin/vmstat 30 2
28. /usr/sbin/ping -s DATA_SERVER 56 3
29. /usr/sbin/traceroute DATA_SERVER
30. /usr/bin/df -lk
31. /usr/bin/sar -k 15 4
32. /usr/bin/sar -a 15 4
33. /usr/bin/sar -g 15 4
34. /usr/bin/sar -u 15 4
35. /usr/bin/sar -d 15 4
36. /usr/bin/sar -r 15 4
37. /usr/bin/ps -ef | grep metacity (or) ps -ef | grep dtwm
38. ce
39. /usr/bin/ps -eo pid,nlwp,comm | nawk ...
40. /usr/bin/who -H | egrep dtlocal|dtremote
41. utwho -c
42. /opt/SUNWut/sbin/utcrypto
43. /usr/bin/ls -1a /tmp | grep Xauth | wc
44. /usr/bin/pkgparam SUNWuto VERSION
45. st.xx files | nawk st.awk
46. /usr/bin/ls -l /tmp/.cleanRunaways/*state
47. /usr/sbin/ifconfig -a
48. /usr/bin/ps -eo pid,user,vsz,rss,s,pri,nice,time,etime,pcpu,nlwp ...
```

The monitoring script uses approximately 19 seconds of CPU time and has an elapsed time (clock time to run) of approximately 90 seconds. The output of this script is stored in the local `/tmp` directory of the client being monitored temporarily before being sent to the central Project Canary monitoring server to be incorporated into the Web display.

All data gathered from clients is sent to the central Project Canary monitoring server and must be viewed using the Web interface. Project Canary software does not support viewing data directly on client machines being monitored. The temporary file generated by the client software is overwritten every time Project Canary gathers data from the client.

The temporary file created by Project Canary software follows a simple naming convention.
It is named `canary.rawdatafile.{$USER}` where $USER is the user name it is executed under.

An example of this file is:

```
client% ls -al /tmp/*rawdatafile*
-rwxrwxrwx 1 canaryusr canary 24058 May 4 06:57 canary.rawdatafile.canaryusr
```

Every raw data file generated will always follow this naming convention regardless of any other software running on the client being monitored by Project Canary.

1.3: How the raw data file is moved to the central data collection server

There are four ways the raw data file can be copied from the server being monitored to the central data collection server:

- NFS copy: The `canary_solaris.sh` or `canary_linux.sh` script will simply copy the raw data file to the central data collection server using `cp(1)`
- Electronic Mail: This option sends the approximately 30kb ASCII raw data file as an email message to the central data collection server using `mailx(1)`
- FTP: The ASCII raw data file is sent to the central data collection server using `ftp(1)`
- SCP: The ASCII raw data file is copied to the central data collection server using `scp(1)`

The particular method used is set in the Project Canary configuration files during installation.

See the *Project Canary Installation and Configuration Guide* for additional details on how data is transferred from clients to the central data collection host and how to determine which method of transferring data from the client to the central data collection host best fits the particular environment Project Canary will be running in.

The *Project Canary Installation and Configuration Guide* has detailed instructions on how to configure data transfers from clients to the server.

1.4: Frequency of client data-gathering script execution

The default interval as Project Canary software comes preconfigured is "once every ten minutes." This is configurable by editing the Project Canary scripts, but should not be set to run more often than once every two minutes. Running the client data gathering scripts more often than the default of once every ten minutes offers little benefit and uses additional resources on the client machine being monitored. In a large installation, this may cause problems with the central monitoring server if that server is not sized properly.

1.5: How Project Canary software runs "periodically" without the use of cron

There are two methods supported and included with Project Canary software:

An `expect` script is supplied with Project Canary software which runs from the central data collection server and will execute a remote login session to each client machine being probed. This `expect` script will execute `canary_solaris.sh` or `canary_linux.sh`, depending, on each

client being monitored. A ten minute delay between data gathering runs is coded into this `expect` script. This method of data collection requires absolutely nothing to be done on the client machine and also requires that no software be installed on the client. Everything, using this method of data collection, is controlled and executed from the central data collection server.

Client scripts (a package of approximately 250kb in size) may be installed on the client machine. This is the recommended approach. The package of client scripts should be installed according to instructions in the *Project Canary Installation and Configuration Guide*. These scripts contain code which simulates a crontab entry on the client machine being monitored.

1.6: How the data is displayed on the data collection server's web page

Two files are sent from the client every ten minutes. The raw data file is the output from the `canary_solaris.sh` or `canary_linux.sh` script. The raw data file is approximately 300 lines long. A parsed version of this file is copied to the `incoming` directory. The parsed file is approximately 20 lines long. The files in the `incoming` directory are moved to the `parsed_data` directory. This is done every two minutes. In the process of moving this parsed data to its final destination, the web page is updated automatically.

It is not possible to view data directly on the client. One must use the Project Canary web page to view all data that has been collected from clients in order to do proper analysis and take advantage of all available features of Project Canary software.

1.7: Viewing Data on the Web Page

There are four tabs at the top of the Project Canary web page. These tabs are:

1.7.1: Home Tab

This tab will always take one to the main "splash" page of the Project Canary Web Page. It is the "root" of the Project Canary web site and the starting point for all navigation through data collected from clients.

1.7.2: Dashboard Tab

This tab will take one to the Project Canary dashboard under which data collected from clients can be viewed in groups or as individual servers. Graphs of data collected from clients can also be found under the Dashboard tab. Some reports, such as lists of runaway processes can be found under the Dashboard tab which are useful in diagnosing load problems with servers. Also, each server's particular configuration and the raw data file collected from each client server may be examined from sub menus underneath the Dashboard tab.

Viewing Data on the Web Page (continued)

1.7.3: Reports Tab

This tab will take you to the Project Canary reports page. The reports available under this tab include load profiles, server drill downs, animated CPU load graphs, the age of a probe, server configuration settings, CPU utilization by user, logins by the same user to multiple servers, application resource utilization histograms, CPU profiles, etc.

1.7.4: Documents Tab

This tab will display all the documentation available for Project Canary software. The installation guide, this user guide, and other documentation for Project Canary software will be found by clicking this tab. Late-breaking news that did not have a chance to make its way into the manuals provided with Project Canary software will also be found here in release notes, errata, and `README` files. It is strongly recommended that users read through all documentation provided with Project Canary software before using it to diagnose and correct load issues with servers being monitored by the software.

1.7.5: Load Test

The Canary Load test has the ability to put the exact same load across Linux, Solaris x86 and Solaris Sparc.

These tabs are the primary methods of navigating the Project Canary web site and accessing the data collected from clients. Each tab brings one to sub menus which are documented in successive sections of this *User Guide and Manual*.

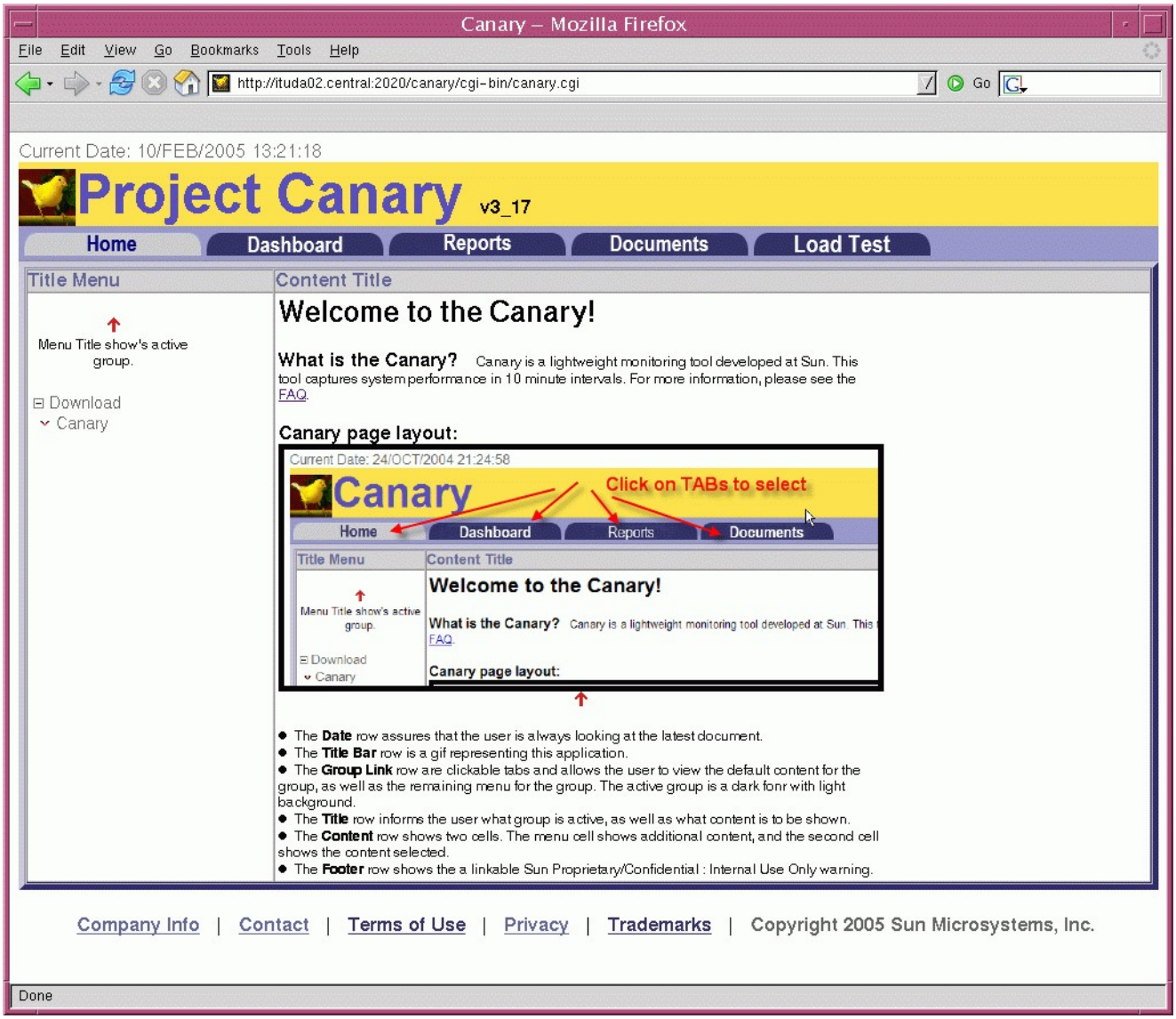
At any point, while navigating the Project Canary website, one may simply use the tabs at the top of the Project Canary web page to jump from one section to another.

Chapter 2: Project Canary Website Overview

2.1: Home Tab

When one first arrives at the Project Canary website, one views the site from the Home Tab by default. Clicking on the Home Tab will always return one to this view.

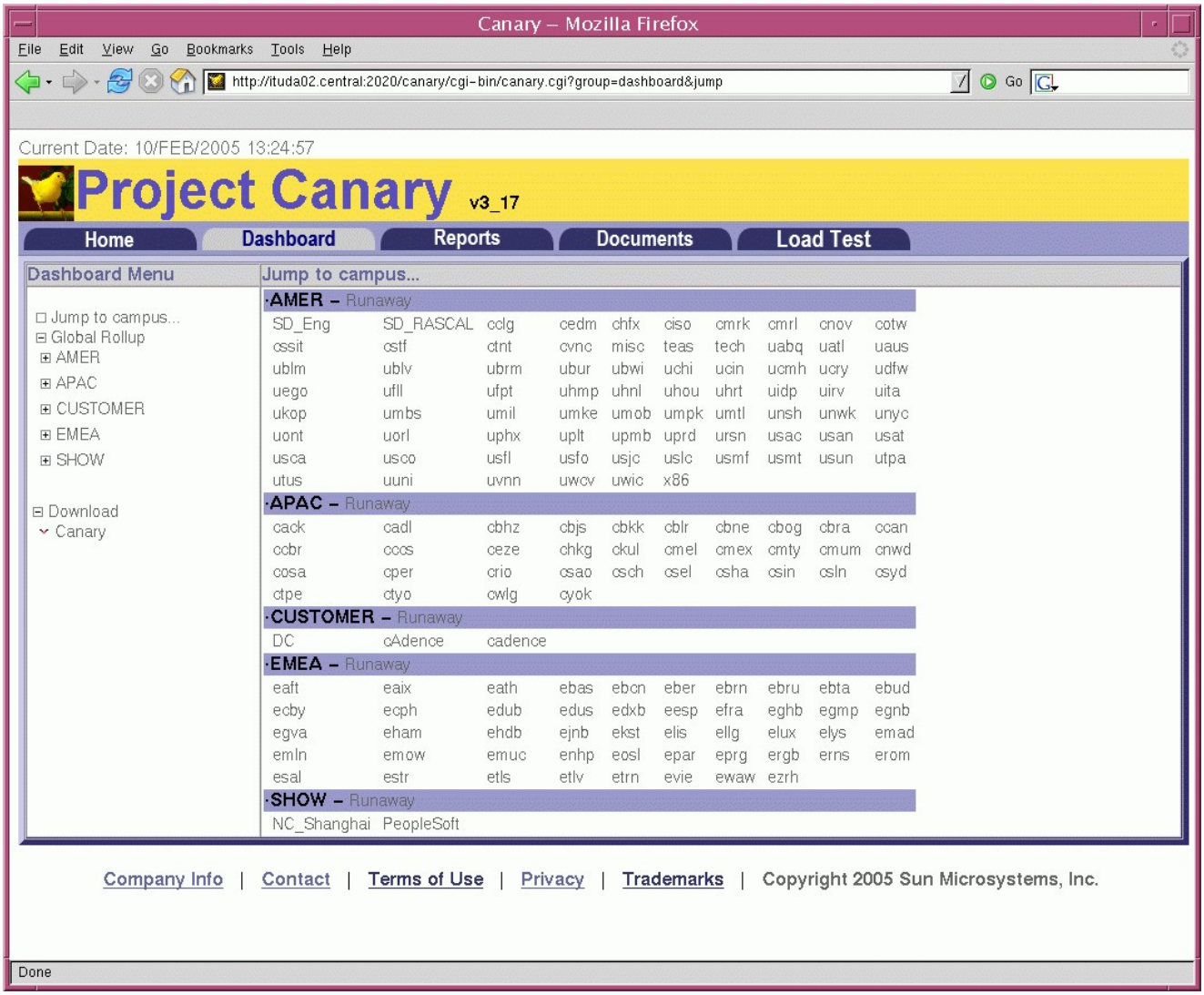
The Project Canary website from the root or Home Tab view looks like this:



On this page, one sees a basic guide for navigating the site as well as the Tabs used to navigate it. On the left-hand side there is a Download Link to download Project Canary software. At the bottom are navigation links showing license information, privacy statements, contact information, and the Copyright notice.

2.2: Dashboard Tab

Clicking on the Dashboard Tab brings one to the Dashboard view:

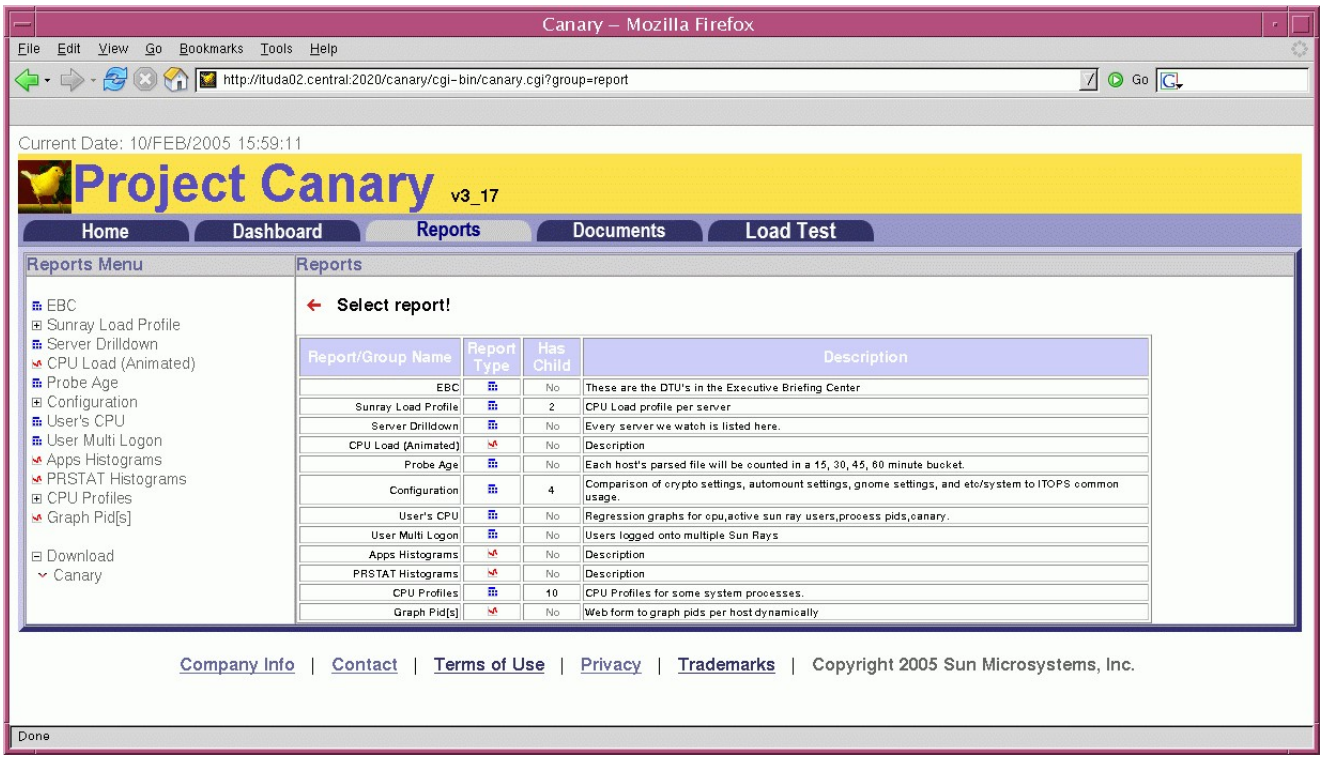


On the main Dashboard page, all servers being monitored are grouped and displayed by their groups. Clicking on a group will display data gathered by Project Canary software. Clicking the "Runaway" link next to the top-level group name will show all processes that Project Canary software think are runaway processes for that entire group of machines being monitored.

There is, again, a "Download" link to the left and the left-hand menu can be expanded or collapsed as needed to drill down and examine data gathered by Project Canary software in greater detail. The main Dashboard page is a gateway to other pages as one goes to examine data gathered from client machines.

2.3: Reports Tab

Clicking on the Reports Tab brings one to a page where various reports compiled by the Project Canary server from data gathered from clients can be viewed:



Here one can access the various reports compiled by the Project Canary server.

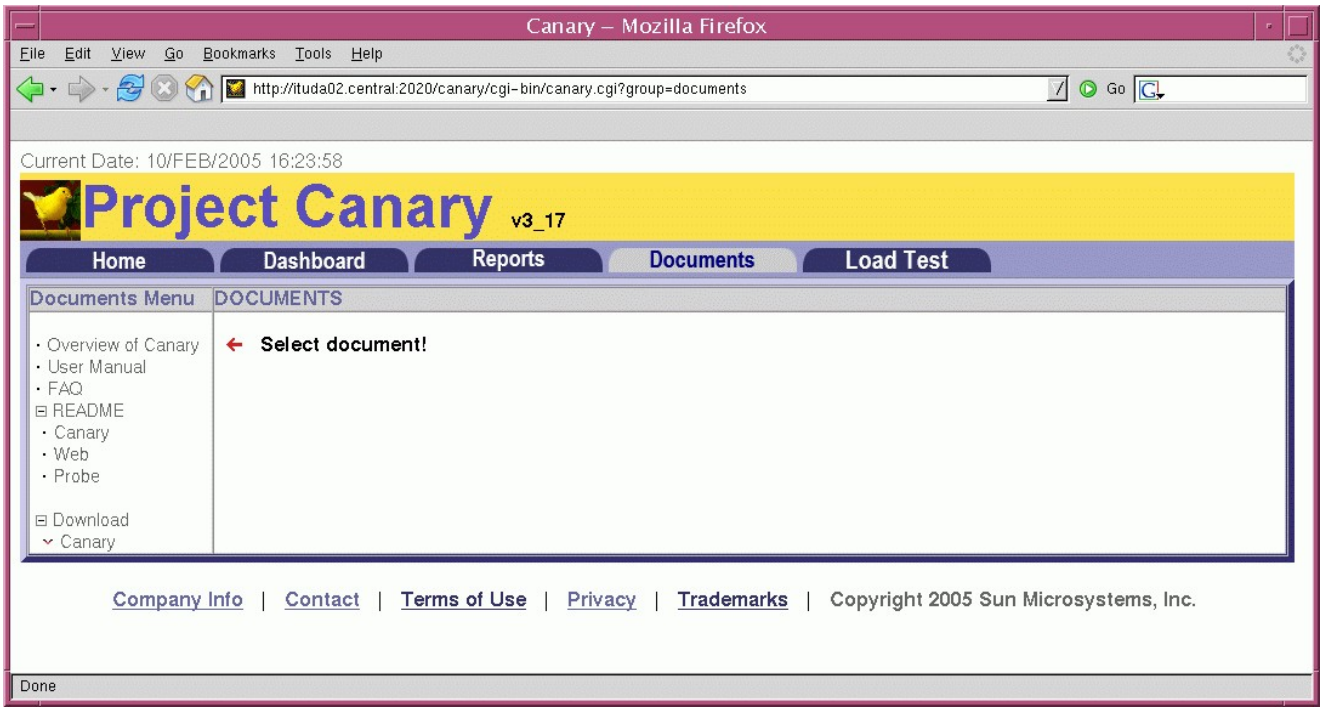
Reports available from this Tab are:

- Sun Ray Load Profile, an expandable menu
- Server Drilldown
- CPU Load (Animated) along with the number of users on the Sun Ray server
- Probe Age
- Configuration, an expandable menu
- User's CPU
- User Multi Login
- Apps Histograms
- PRSTAT Histograms
- CPU Profiles, an expandable menu
- Graph PID[s]

There are two images as well which state in what form a report is given when one clicks on it. The icon with the red slash indicates that the report is graphical, whereas the blue "sheet" icon indicates a report delivered in text.

2.4: Documents Tab

The Documents Tab takes one to a page on the Project Canary server where all the documentation provided with Project Canary software can easily be viewed or downloaded.



The only options available on this page are viewing or downloading Project Canary documentation. The menu on the left lists the documents available for download or viewing in the browser.

Currently, these documents include:

The Project Canary Overview
The Project Canary User Guide and Manual (this document)
The Project Canary FAQ
README files for various portions of the Project Canary software including
    Project Canary server software
    Project Canary client probe software
    Project Canary server-side web software

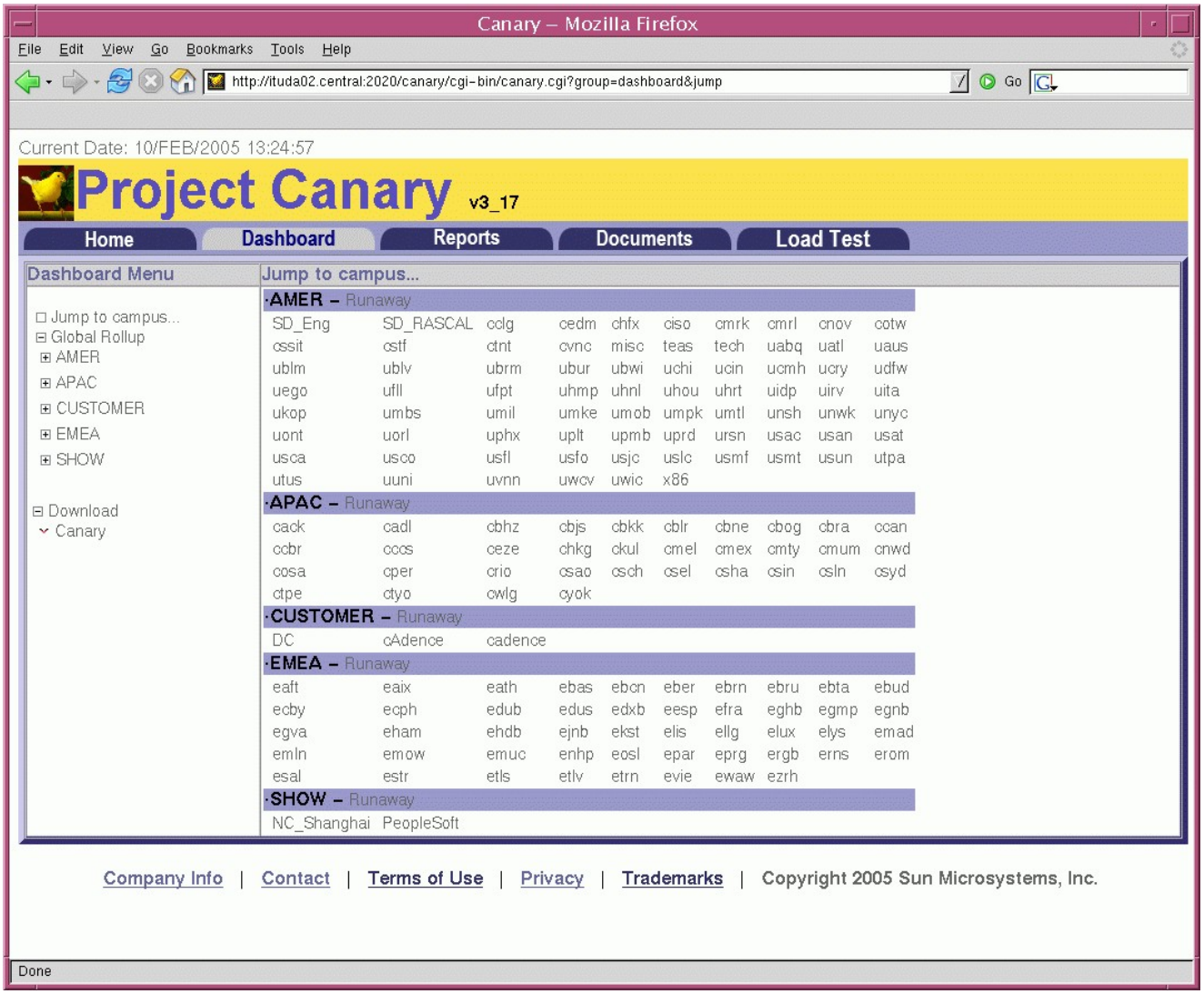
There is also a download link on the left so one can download a copy of Project Canary from this page.

It is strongly recommended to read through all the `README` files for each release of Project Canary software as those will include late-breaking news, release notes, and other errata that may not have had time to make it into the formal Project Canary documentation.

Chapter 3: Dashboard Tab

3.1: Dashboard Tab: Root Page

The Dashboard Tab is where servers are grouped and data on individual servers can be found. The root page of the Dashboard Tab looks like this:

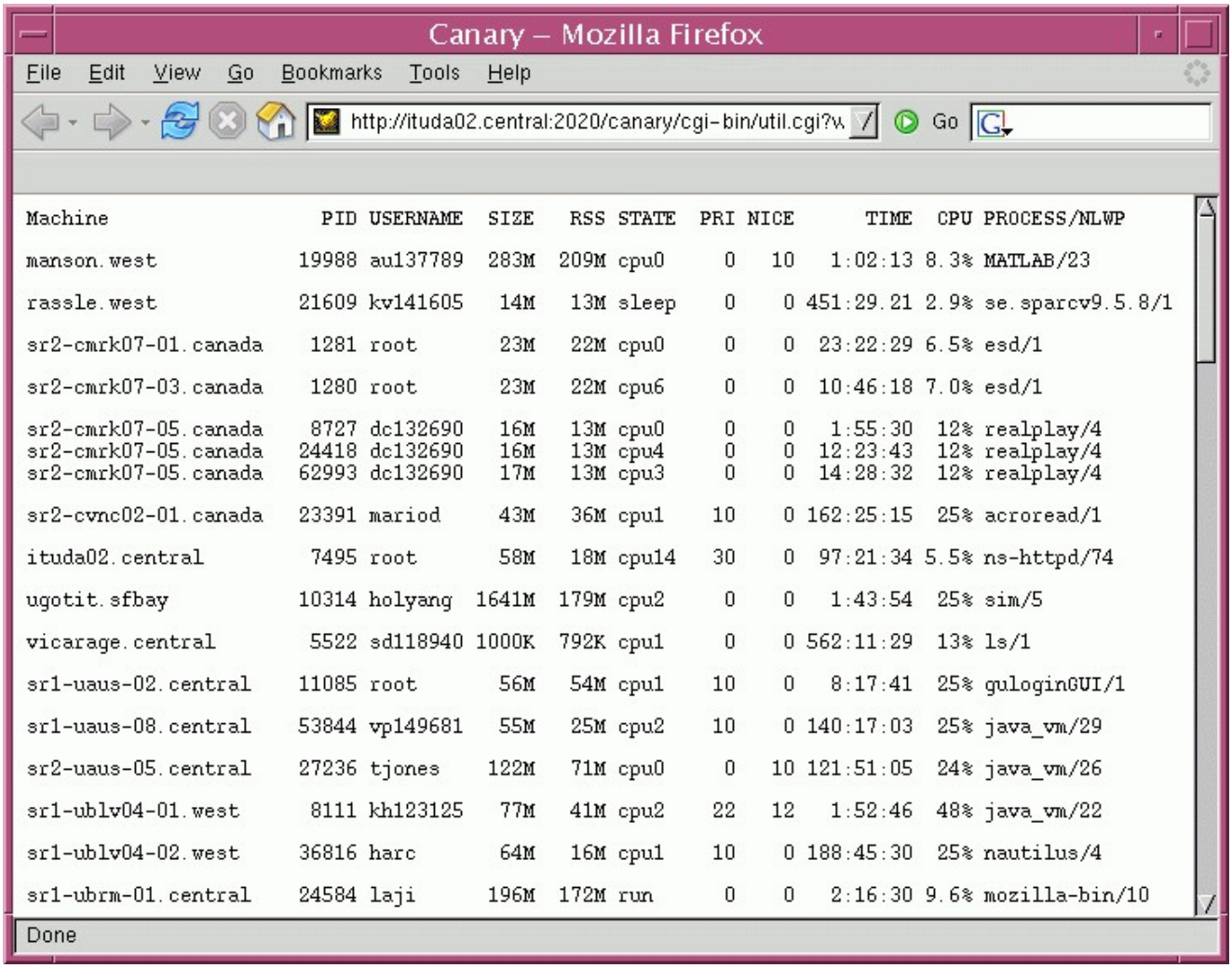


By default, servers are grouped into "Geos" and "Campuses." Multiple "Campuses" make up a "Geo" as is shown above. This is configurable in the `server_to_geo.cfg` file. Details on how to name and/or rename "Geos" and "Campuses" are available in the *Project Canary Installation and Configuration Guide*.

3.2: Dashboard Tab: Runaway Link

Next to the "Geo" name is a "Runaway" link. This link lists all processes that Project Canary software thinks are runaway processes and make good candidates for termination.

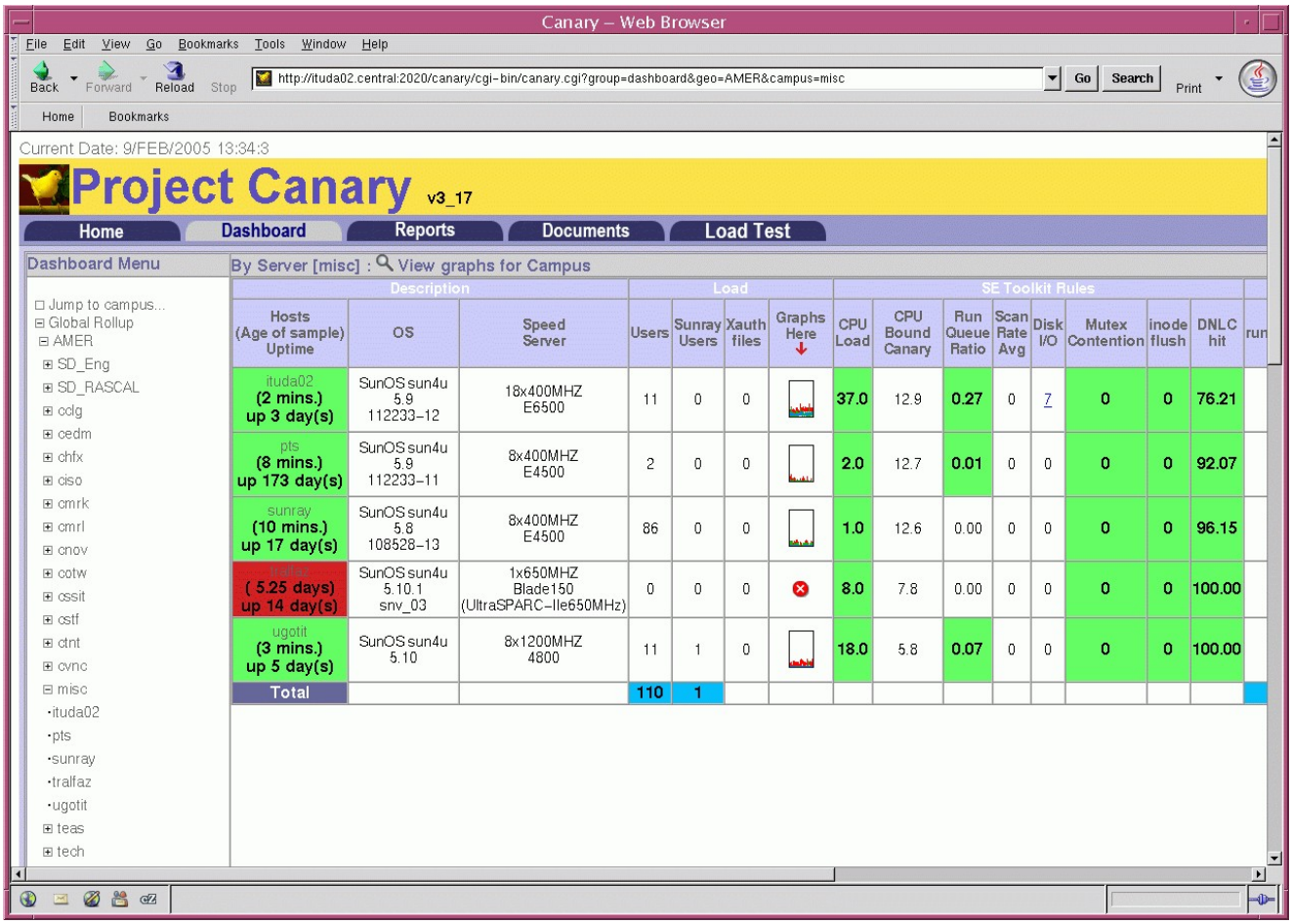
Clicking on the "Runaway" link brings one to a page listing the owner of the alleged runaway process, the server it is running on, how much CPU time it has used, etc. This list is a sorted list based on the output from `ps(1)` which the client software runs at regular intervals as part of its data collection process.



As shown above, the "Runaway" link lists processes that Project Canary software has determined to be potential runaway processes. Project Canary software lists out in this report all the data needed to make a determination as to whether or not these processes are indeed runaway processes. The list includes the machine the process is running on, the process ID number, the username the process is running under, its status, CPU utilization, CPU time used, etc. These should be investigated by the systems administrator and terminated if, indeed, they are runaway processes.

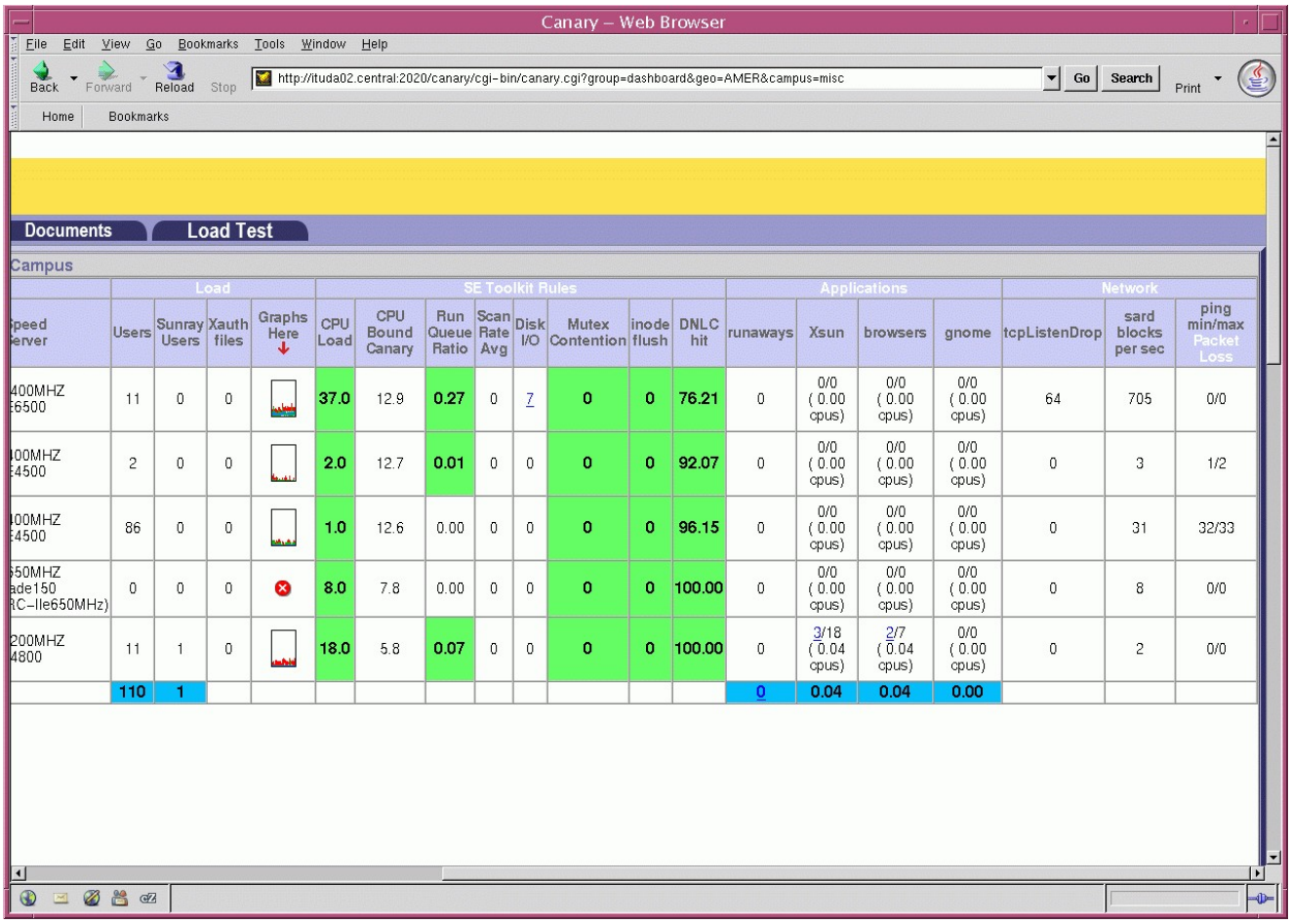
3.3: Dashboard Tab: Campuses

Clicking on one of the "Campuses" brings one to a screen here additional data can be found on servers in that grouping:



As shown above, clicking on a "Campus" grouping displays a page with additional data. This page is quite wide due to the amount of data it displays. It shows the list of hosts within that particular "Campus," the operating system it runs, the CPU speed, number of users on that particular system, as well as other pertinent data. Data is summed at the bottom if possible.

The second part (the right side) of the "Campus" view is shown below:



	Load				SE Toolkit Rules								Applications				Network		
Speed Server	Users	Sunray Users	Xauth files	Graphs Here	CPU Load	CPU Bound Canary	Run Queue Ratio	Scan Rate Avg	Disk I/O	Mutex Contention	inode flush	DNLC hit	runaways	Xsun	browsers	gnome	tcpListenDrop	sard blocks per sec	ping min/max Packet Loss
400MHZ 6500	11	0	0		37.0	12.9	0.27	0	7	0	0	76.21	0	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0/0 (0.00 cpus)	64	705	0/0
400MHZ 4500	2	0	0		2.0	12.7	0.01	0	0	0	0	92.07	0	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0	3	1/2
400MHZ 4500	86	0	0		1.0	12.6	0.00	0	0	0	0	96.15	0	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0	31	32/33
650MHZ ade150 RC-IIe650MHz)	0	0	0		8.0	7.8	0.00	0	0	0	0	100.00	0	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0/0 (0.00 cpus)	0	8	0/0
200MHZ 4800	11	1	0		18.0	5.8	0.07	0	0	0	0	100.00	0	3/18 (0.04 cpus)	2/7 (0.04 cpus)	0/0 (0.00 cpus)	0	2	0/0
	110	1											0	0.04	0.04	0.00			

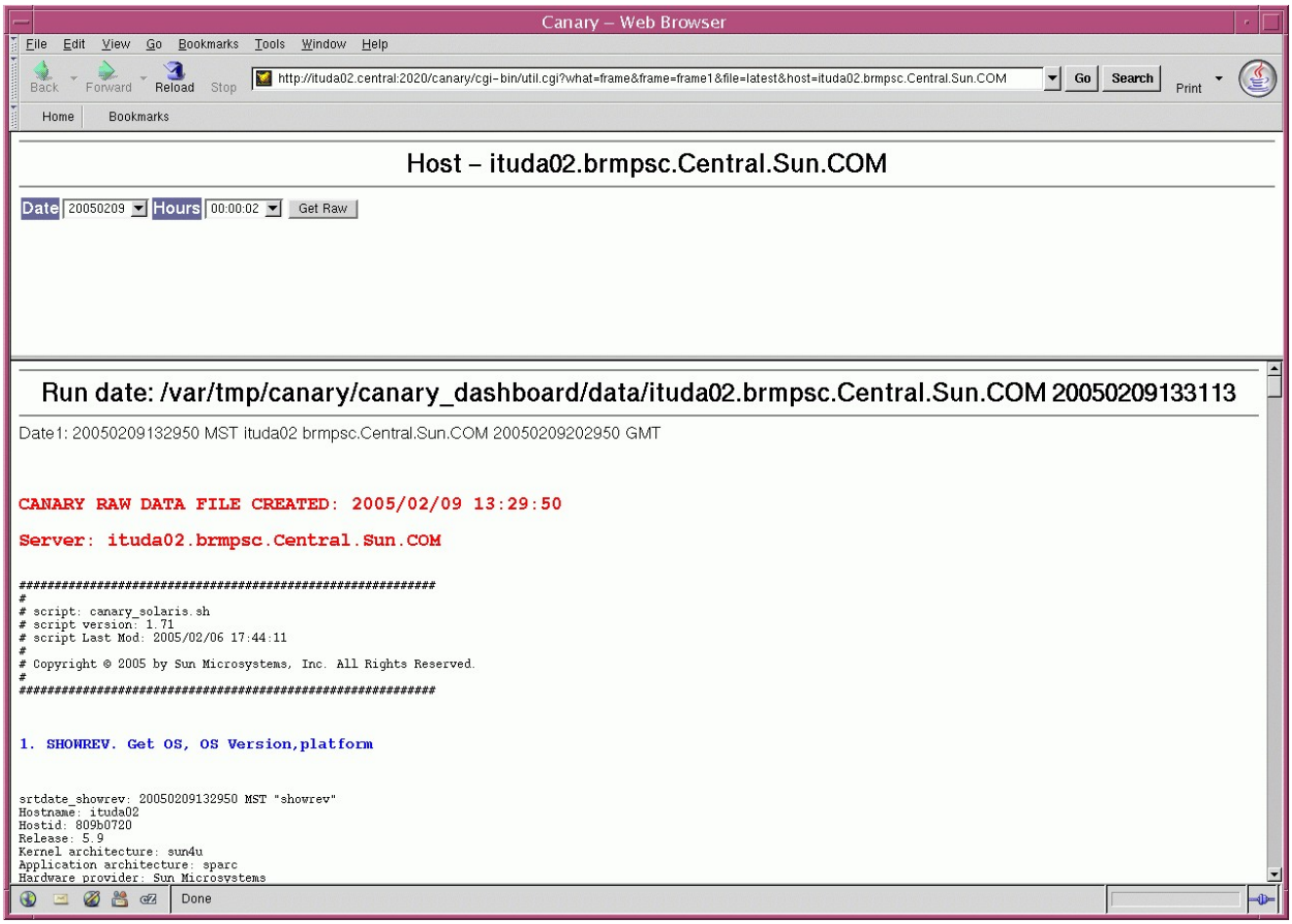
The Campus view includes most kinds of data a systems administrator needs to quickly scan for problems. When a server is sending data to the central Project Canary monitoring server that Project Canary software feels is "out of spec" or problematic, the web page will display that data in red. Yellow signifies "out of compliance with normal but not yet a large problem" and green signifies no issues.

If a server has not reported back in a significant amount of time, the server name will change color as well. Normally there are graphs available for each server that display the statistics gathered. If there is no data available for a particular server, the "small graph" icon will change to a red circle with an X in the middle of it.

Usually, when servers are not sending data back, they are either out of service or there is a misconfiguration in the Project Canary software. If Project Canary software has been installed on a server and there are no graphs or data available for it, check to make sure the client software is running. Failing that, recheck the Project Canary server installation. There may be a mismatch with where clients are sending data and where the server expects it.

3.4: Dashboard Tab: Raw Data File

Clicking on the server name brings one to another page with the raw data file on it:



This page shows when the Project Canary server received the raw datafile, the name of the server as configured in the `server_to_geo.cfg` file, and also, the entire raw data file itself, parsed into various sections. The raw data file contains all the information Project Canary software has gathered about the server, from its hostname to which operating system it is running, a capture of the process stack, its architecture, and all pertinent information to be able to identify the server and examine its performance in detail.

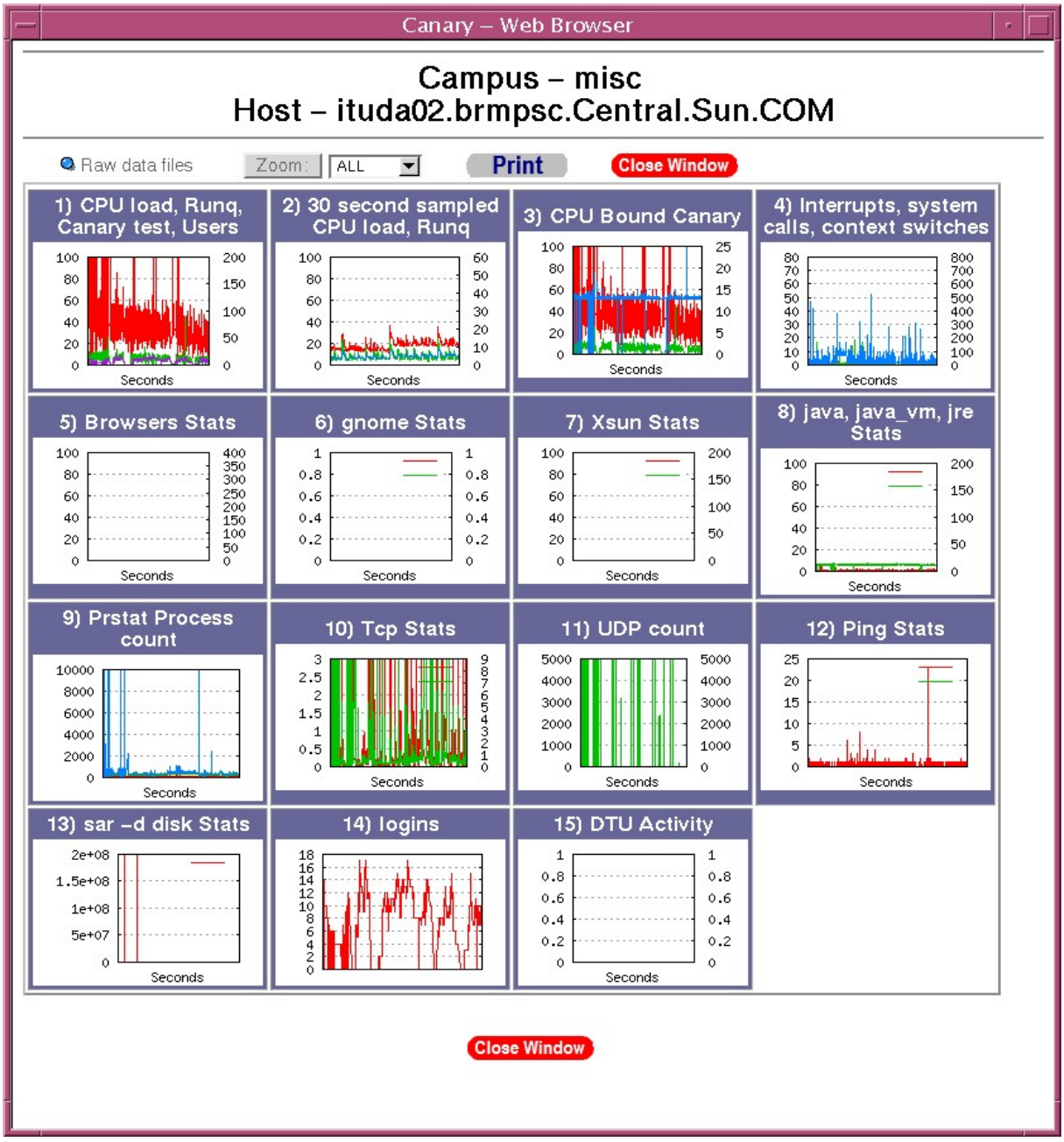
It is generally rare that one would need to refer to the raw data file, however, this data is useful for other tasks, such as taking a server inventory, checking to see how many servers are running a particular operating system, what processes across a cross-section of servers are using the most CPU time, and many other tasks a systems administrator may be asked to gather data for.

Going back to the "Campus" view under the Dashboard Tab, each server also has a clickable link for "Graphs" which display the performance data gathered for the individual server. Most problems can be quickly diagnosed either on the "Campus" page or by using the graphs provided via this link.

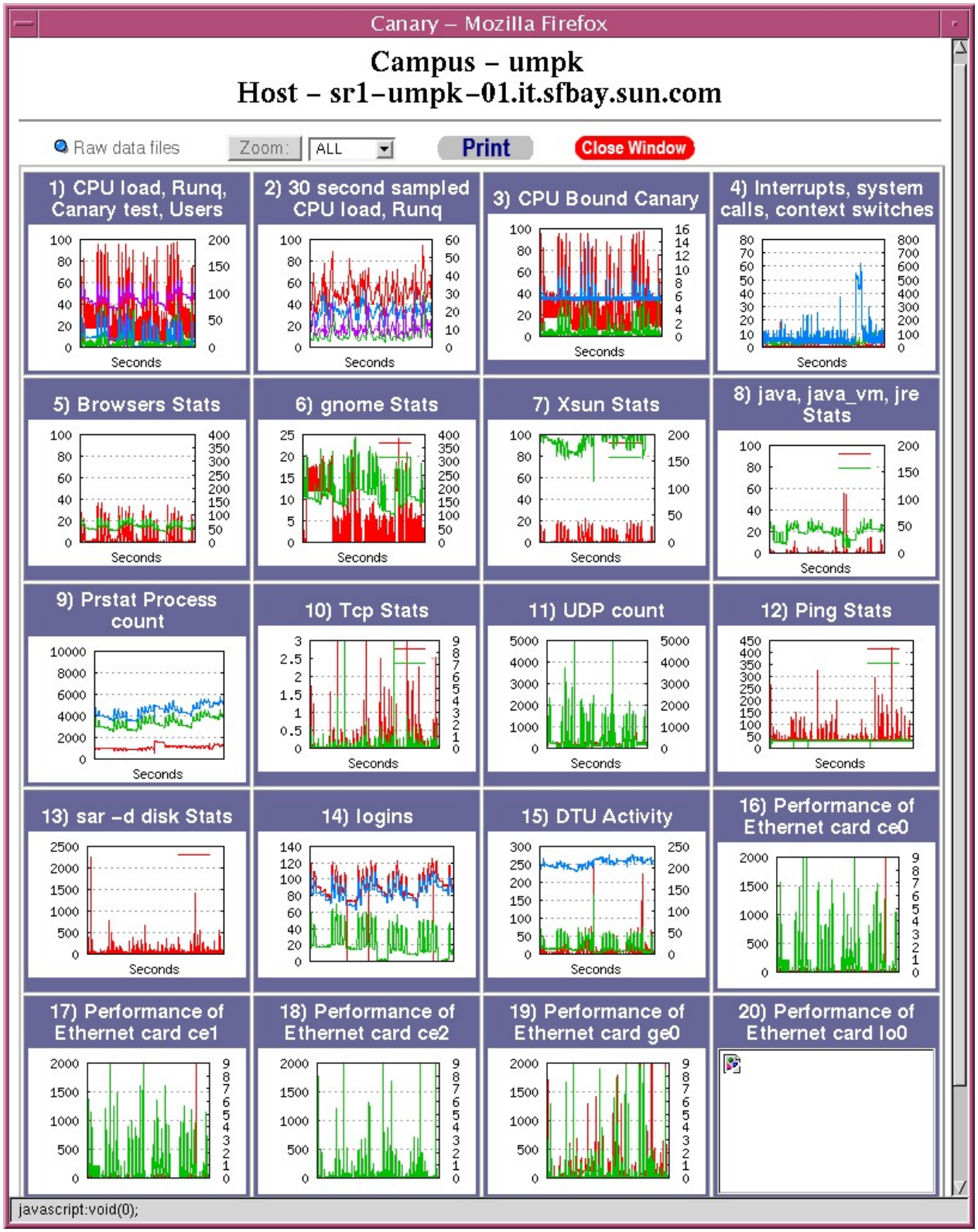
The examples in this *User Guide and Manual* were taken from both an actual data center machine and a production Sun Ray server using an existing production installation of Project Canary software. Graphs and data from both machines will be used as examples to highlight differences when monitoring different types of servers using Project Canary software.

3.5: Dashboard Tab: Graphs

The "Graphs" link on the Dashboard page brings up a popup window that looks like this on a typical data center server:



The graphs for a typical Sun Ray (or other multiuser) server look quite different:

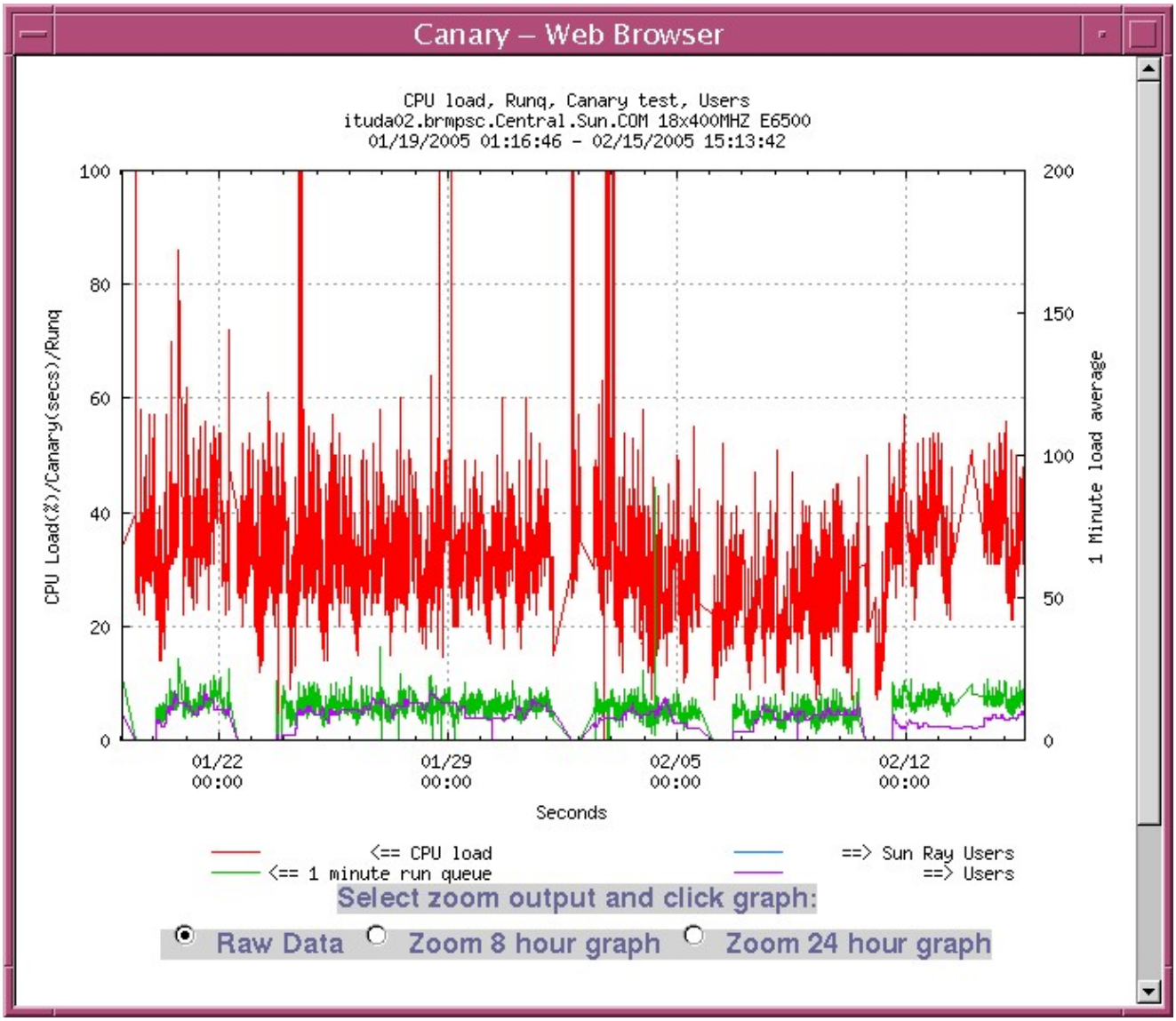


Each of the graphs shown in the previous images are clickable and will bring up another popup window with a zoomed-in view of that particular graph. Raw datafiles are available by clicking the "Raw Data Files" link or

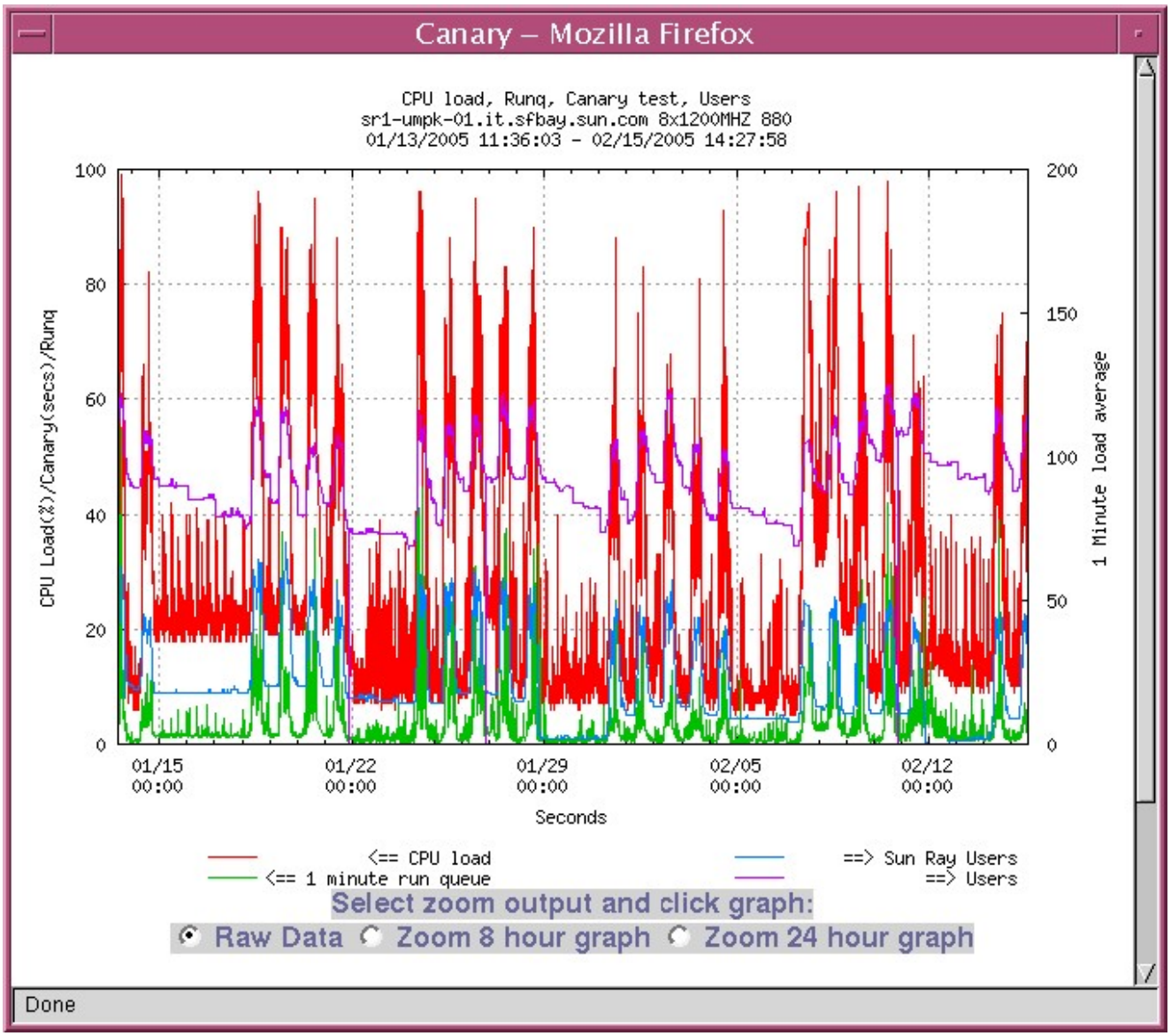
by simply clicking the graph itself. Examining the raw data files is useful when investigating a particular problem that the graphs show.

3.5.1: CPU Load, RunQ, Canary Test, Users Graph

The CPU Load/RunQ/Users/Canary Test graph looks like this on a typical data center server:



On a Sun Ray server with multiple users running applications during business hours, the graph looks quite different:



Scrolling down on the page will reveal a "Print" button which will send the graph to the default printer as well as a "Close Window" button which will cause the popup window to close.

These graphs display the CPU load as well as the run queue and the number of users on the server. Note that the load (the red lines on the graph) rise during business hours when users are actively using the server and dip down again at night. On weekends, the load will drop as well. On a Sun Ray server or any server with users running applications, watch the CPU load on the local Project Canary installation carefully. If the red lines do not fall when the server should be idle, that is a good fast indicator that there are either runaway processes on the server or processes that are hard at work when the may not need to be. On multiuser servers or Sun Ray servers where users are running productivity applications, browsers left open viewing webpages with animated images, Java applets that cycle images or the page itself, or flash animation will cause the browser to use cycles even when the user is not actively using the browser. On the Sun Ray server graph, note that CPU load and number of users tend to rise and fall together. If the run queue drops and the CPU load does not, then that is also another good indicator that runaway processes or processes that are using an inordinate number of

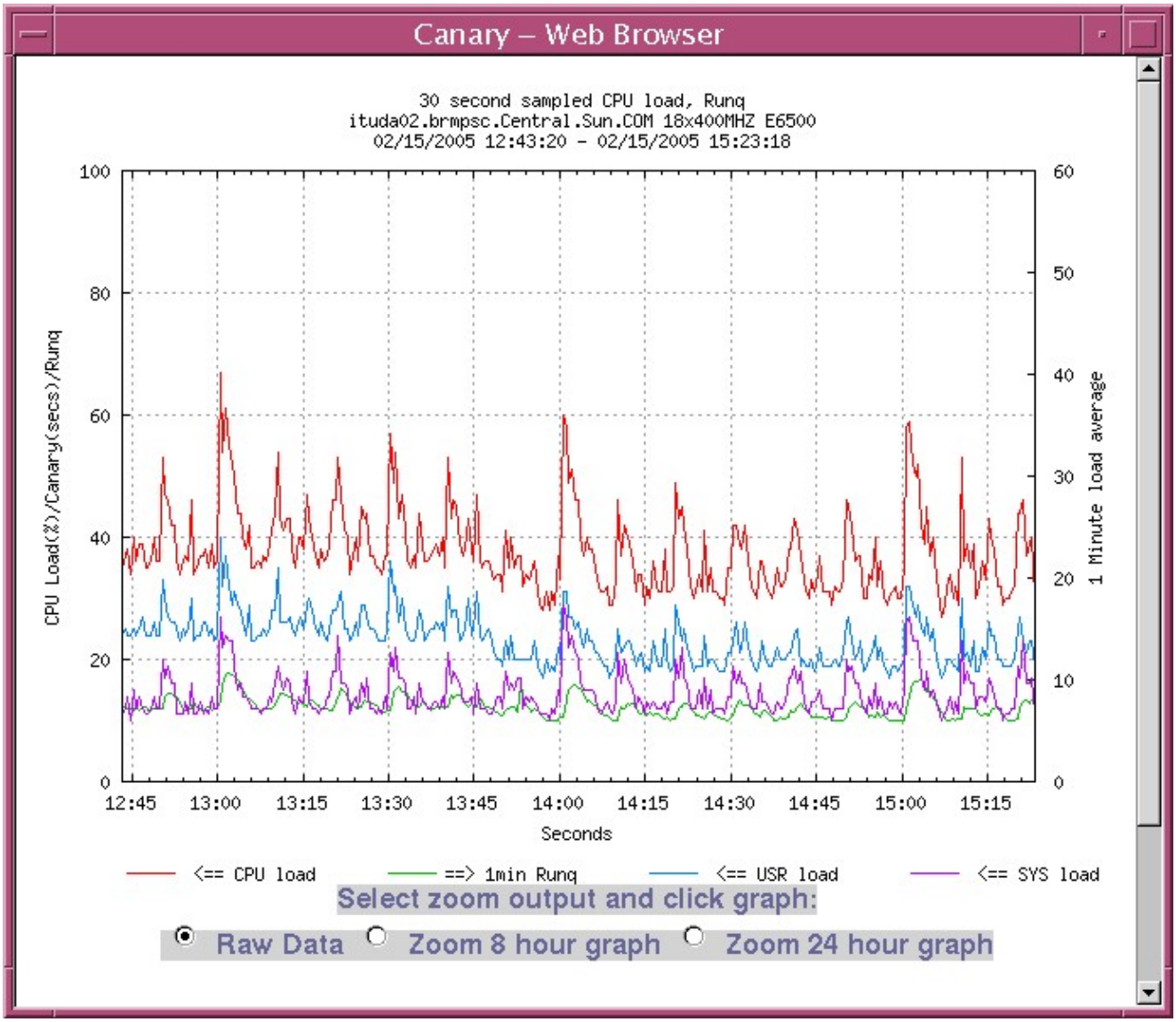
CPU cycles are running. Scrutinize the process table carefully and be certain a process indeed is a runaway before terminating it.

There are also two options on the graph which can be selected to see more specific data. Select the "Zoom 8 hour graph" button to view only the last eight hours' worth of data. Select the "Zoom 24 hour graph" to view the last day's worth of data. By default, the last four weeks' worth of data (or however much data is available up until that time) will be shown.

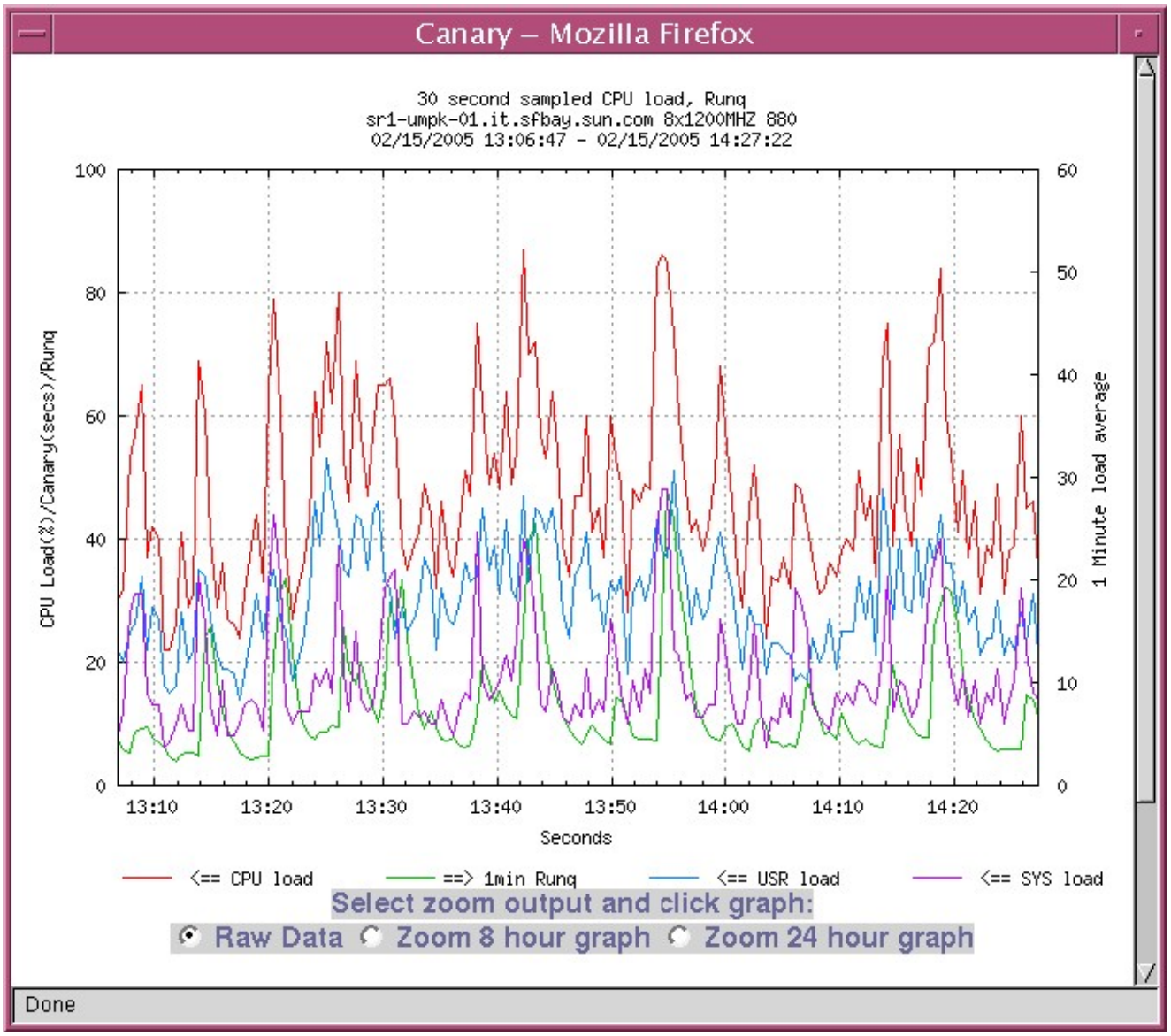
All graphs in this section have the same Zoom options.

3.5.2: 30 Second Sampled CPU Load, RunQ Graph

An example of a graph taken from a typical data center server is shown below:



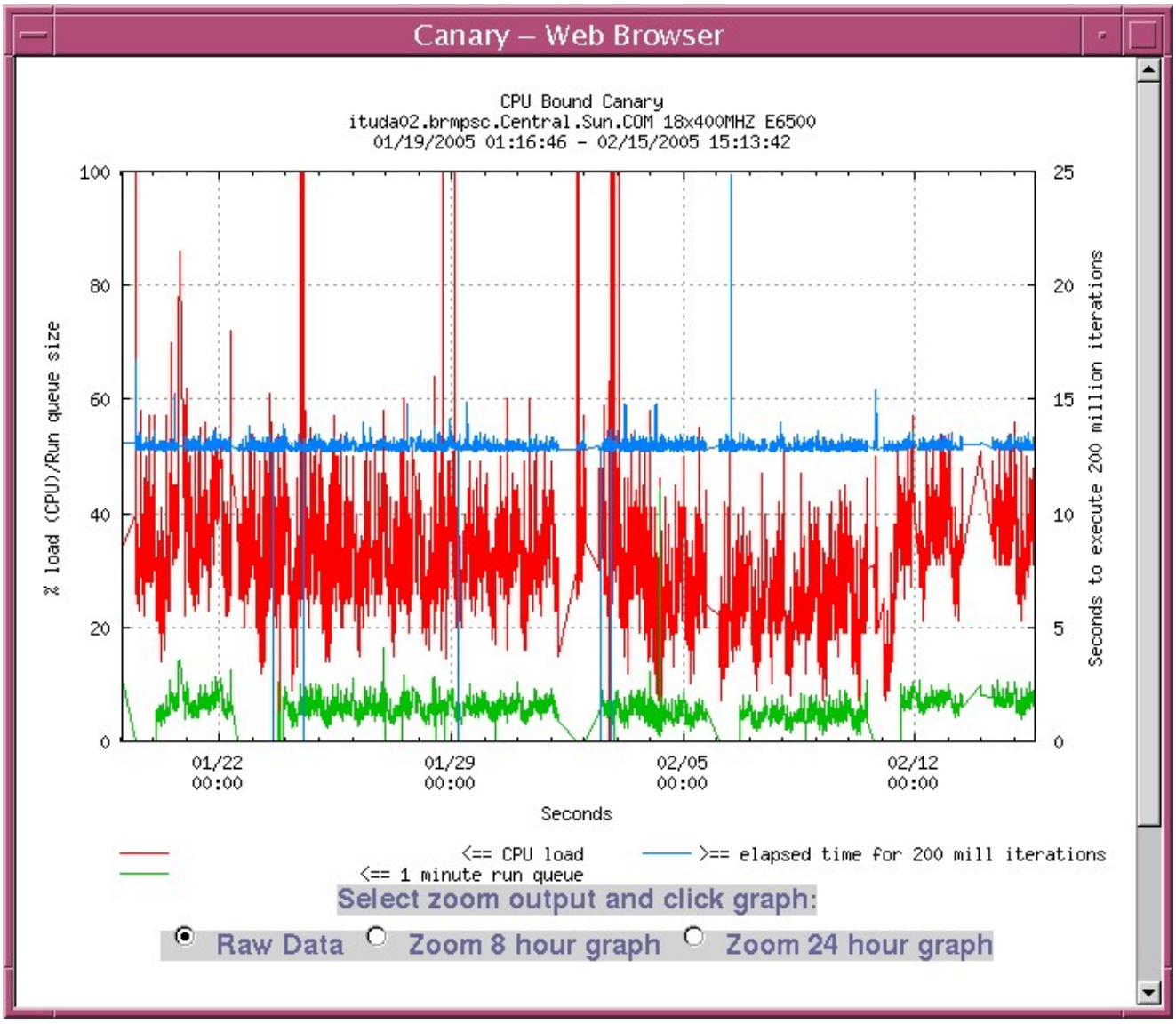
Below is a graph taken from a typical Sun Ray server:



Note the differences between the Data Center server and the Sun Ray server in terms of peaks and valleys in terms of how load is generated during the day. The Data Center server will generally show a more steady load whereas the Sun Ray server, which has users logging in and out and running applications on it will show many more peaks and valleys which can be plotted against the workday clock.

3.5.3: CPU Bound Canary Graph

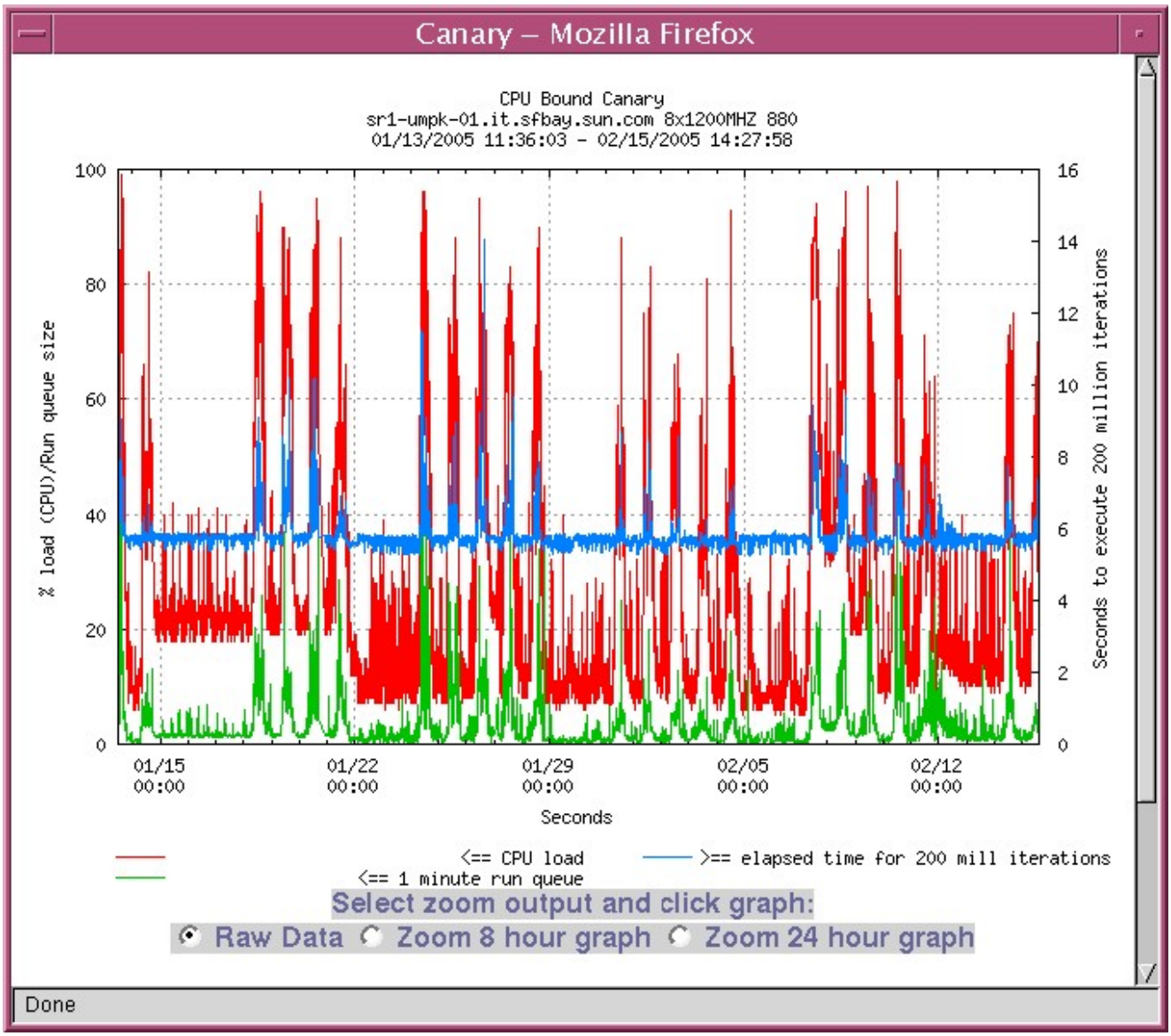
The next graph is the CPU Bound Canary. This test runs a CPU-bound process for ten seconds and then records statistics about that process.



The Blue line shows how many iterations the CPU Bound Canary was able to run in ten seconds. The Red line shows CPU load on the server as a whole and the Green line shows the one-minute run queue for the server as a whole.

This is useful when attempting to diagnose whether there is a process running on a server that is consuming more CPU than it should. Where the Blue line dips, the test is unable to get CPU time to run. On an idle machine, the Blue line should be flat and the CPU Bound Canary will get as much CPU time as it needs. Dips in the blue line indicate load spikes on the server. If the server is not performing as expected, then the times when the Blue line dips should be investigated by examining the raw data files gathered by Project Canary software to determine what caused the performance problem.

The Sun Ray server graph of the CPU Bound Canary is shown below:



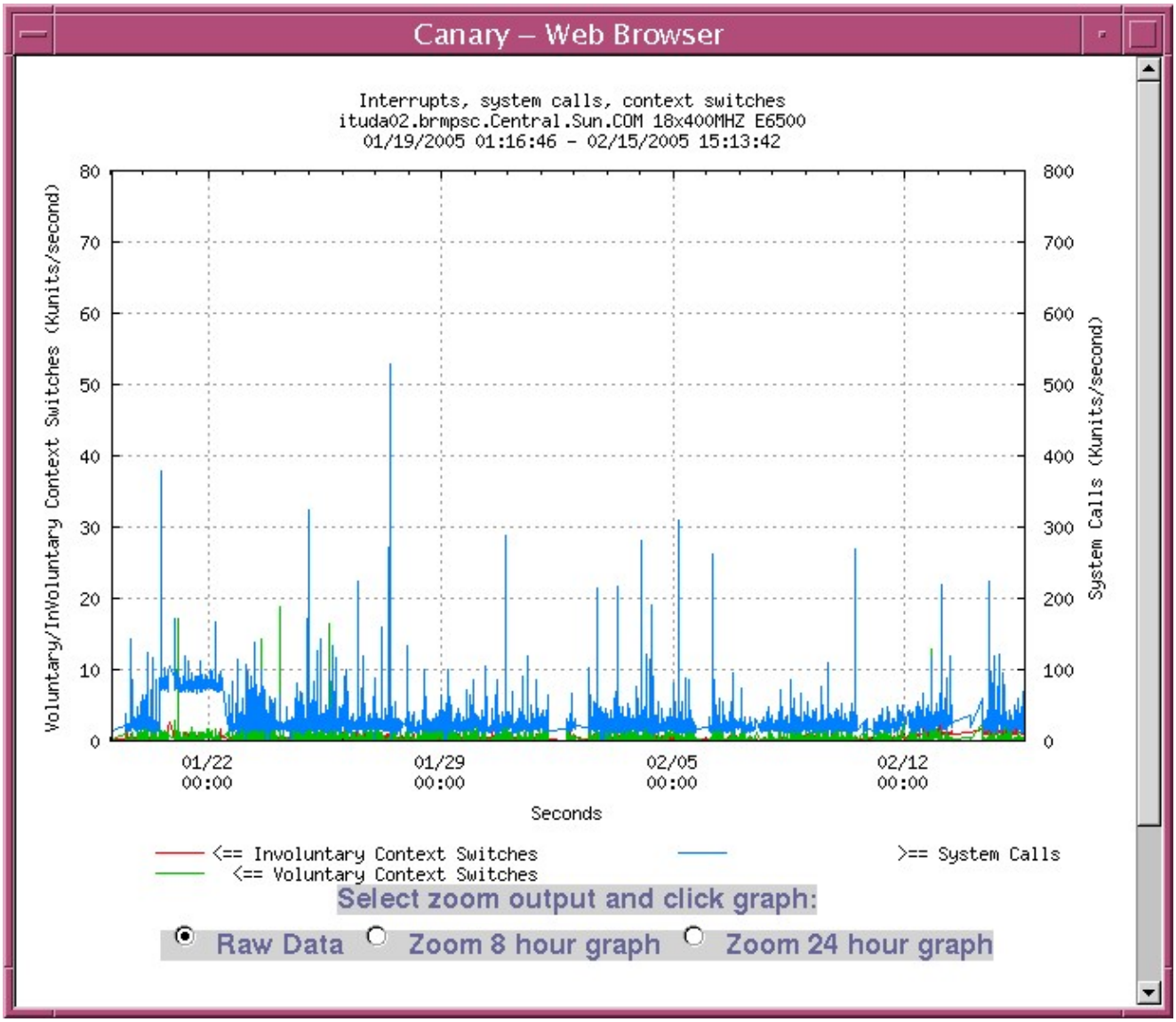
Note the CPU speed difference between the Data Center server and the Sun Ray server.

Note also that the CPU Bound Canary on the Sun Ray server, which is more susceptible to user load, shows the CPU Bound Canary taking much longer to execute than on the Data Center server graph shown previously.

3.5.4: Interrupts, System Calls, Context Switches Graph

The next graph shows CPU interrupts, system calls, and context switches.

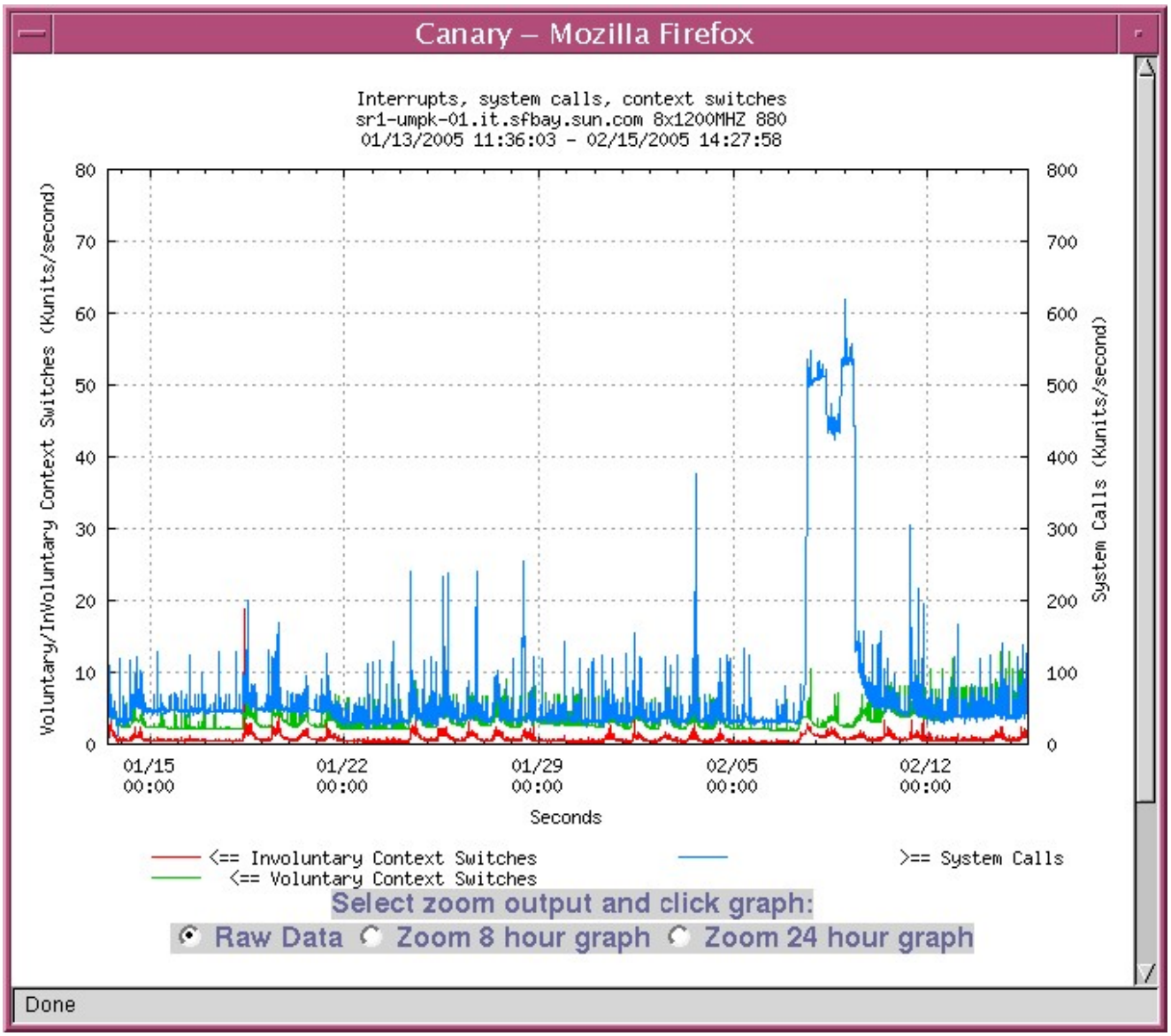
On a typical Data Center server, it looks like this:



This graph shows how many interrupts, system calls, and context switches occur every second on the server being monitored by Project Canary software. High values should be investigated by going to the raw data file for the server and investigating what the server was doing at the time. Context switches are the sum of VCX + ICX. System calls are SCL.

Note that the graph for the typical data center server has system call spikes at the beginning and end of the work day. Since processing is generally done "off-hours" for scheduled jobs like backups or other network-intensive operations, this makes sense. If the graph shows rises during times when work is being done on the server or when intensive jobs are scheduled, it likes does not need to be investigated further. The "blank" sections in the graph above show service outages where a disk was knocked offline or an application abruptly terminated.

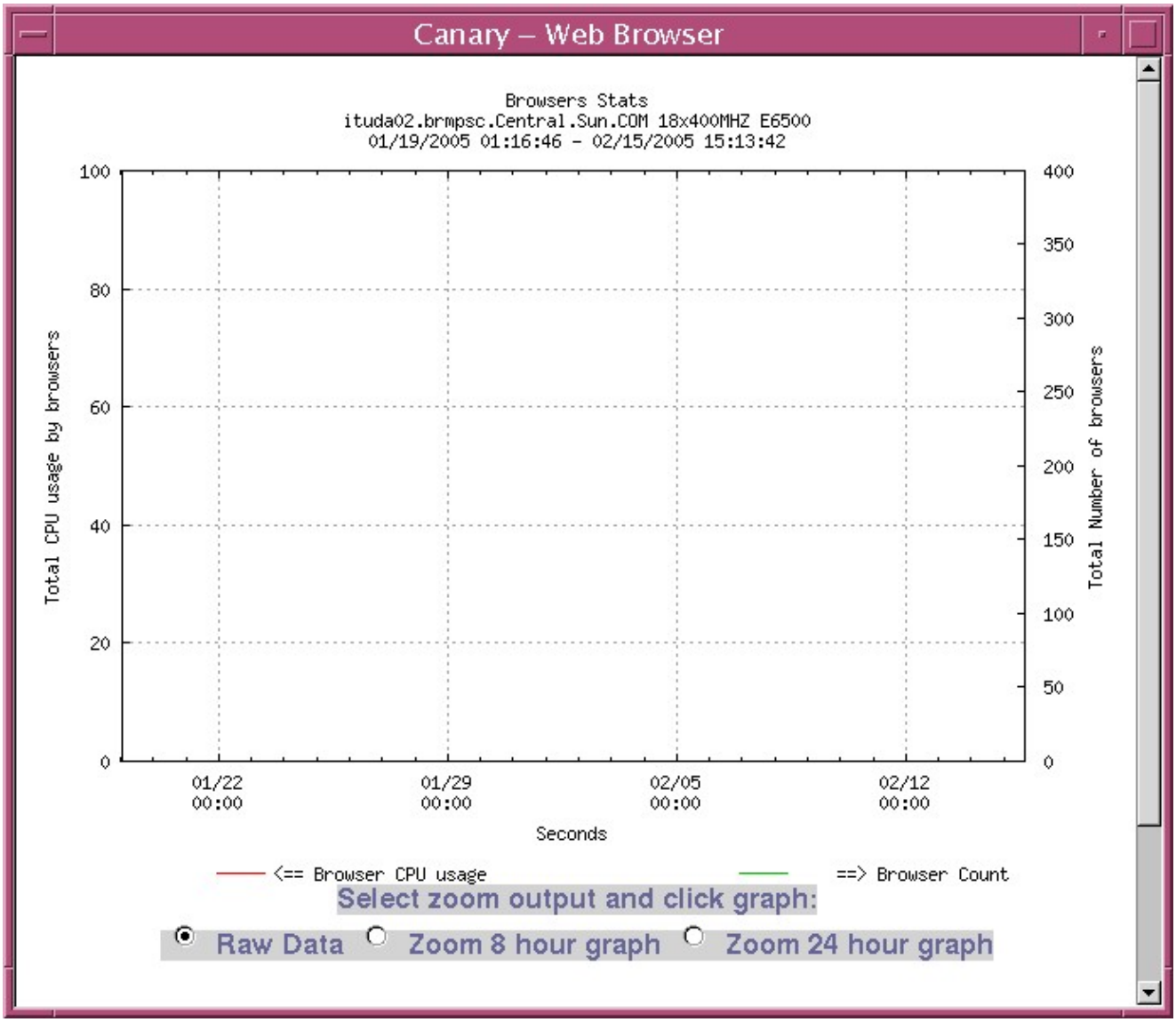
On a typical Sun Ray server, the same graph looks like this:



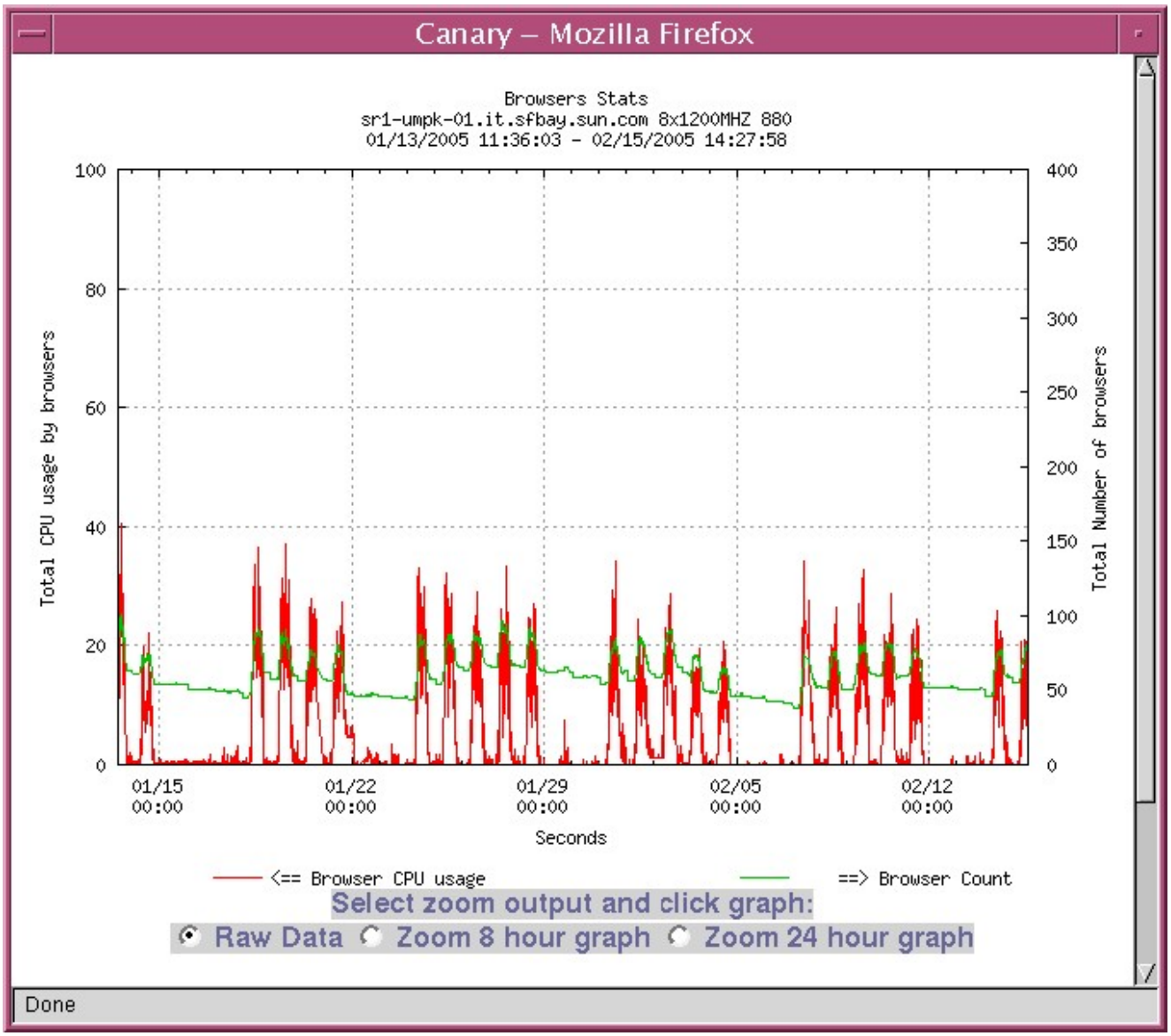
Note the large spike near the end of the graph. This is an anomaly and should be investigated.

3.5.5: Browser Statistics Graph

The next graph shows statistics for Web Browsers. On a typical data center server with no users on it running productivity applications, the graph will be blank and look like this:



On a Sun Ray server with multiple users running Web Browsers, the graph looks like this:



The Browsers Stats graph shows how much CPU load is being generated by users running Web Browsers. Project Canary watches for both the Netscape and Mozilla browsers. Other browsers are not yet being monitored.

The Green line is a simple count of the number of Web Browsers running on the server and the Red line shows the CPU utilization of those same Web Browsers.

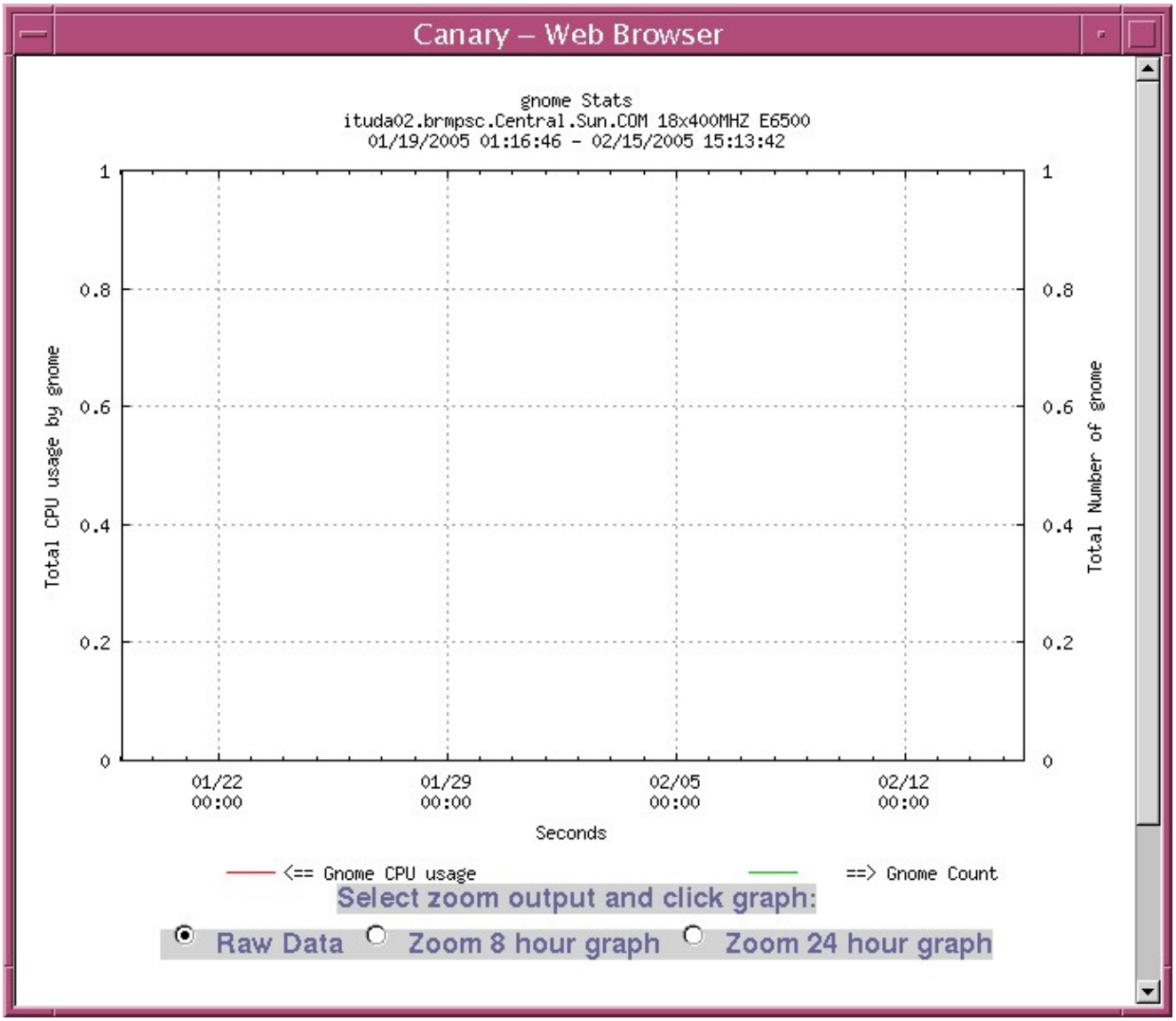
Note that the both the red line and the green line rise and fall together; as more browsers are run on the server, the load increases.

Also note that CPU utilization varies over the business day. In the morning, when users arrive and start new browsers, CPU utilization increases. During typical meeting times, lunchtime, and when users leave work, utilization decreases. The red line should drop during off-hours. If it does not, this may indicate a runaway browser process or a browser that is using CPU cycles for animation or page refreshes when the user is not

using it. If the red CPU utilization line plateaus, this is a clear indicator of a runaway browser process.

3.5.6: GNOME Statistics Graph

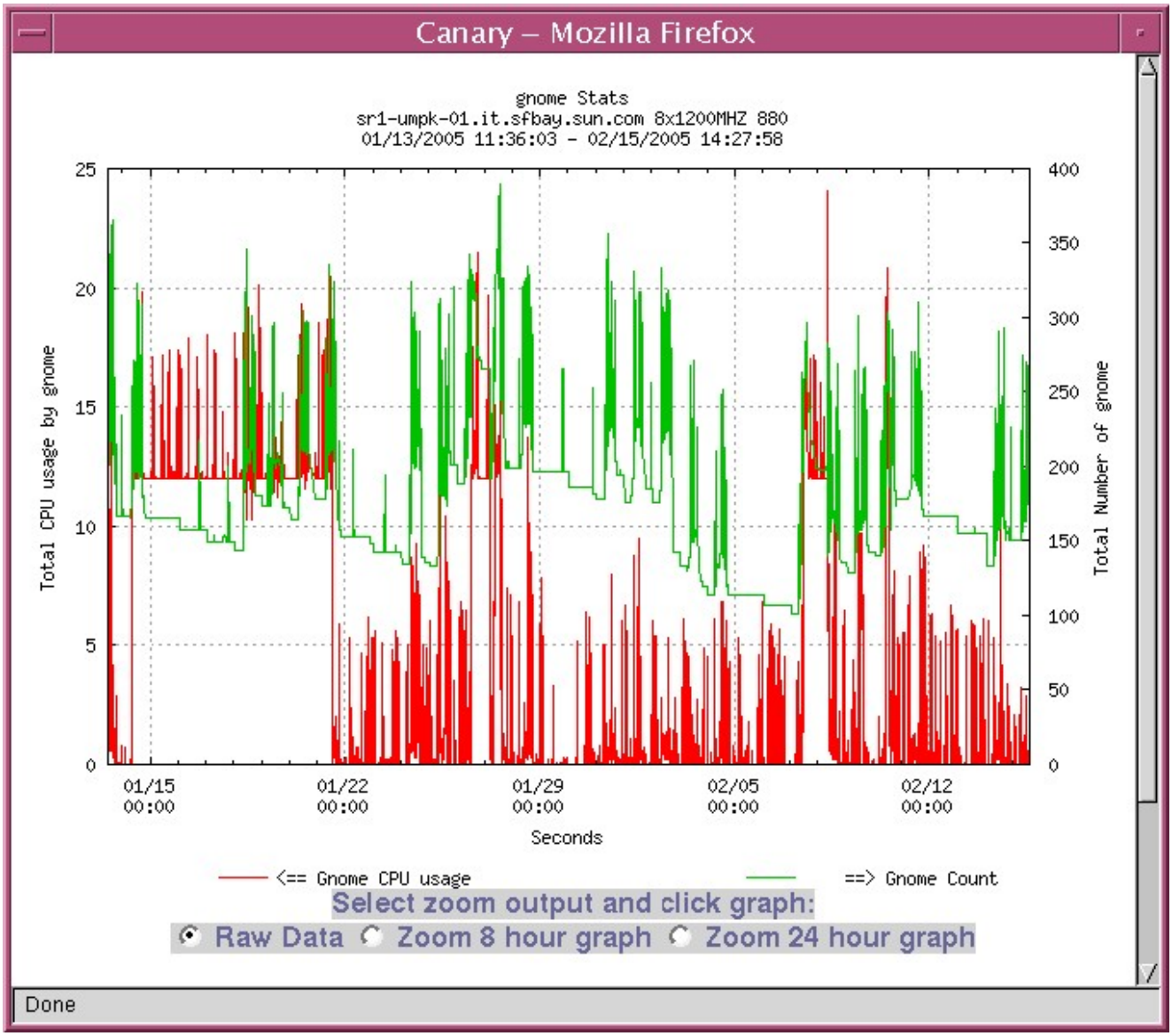
Project Canary also collects statistics on GNOME sessions running on servers. A graph from a typical data center server is blank, as in this example:



On the Sun Ray server, however, with users running real desktops, the graph shows quite a bit of data. The red lines show the CPU utilization that GNOME sessions use and the green line shows the number of GNOME sessions.

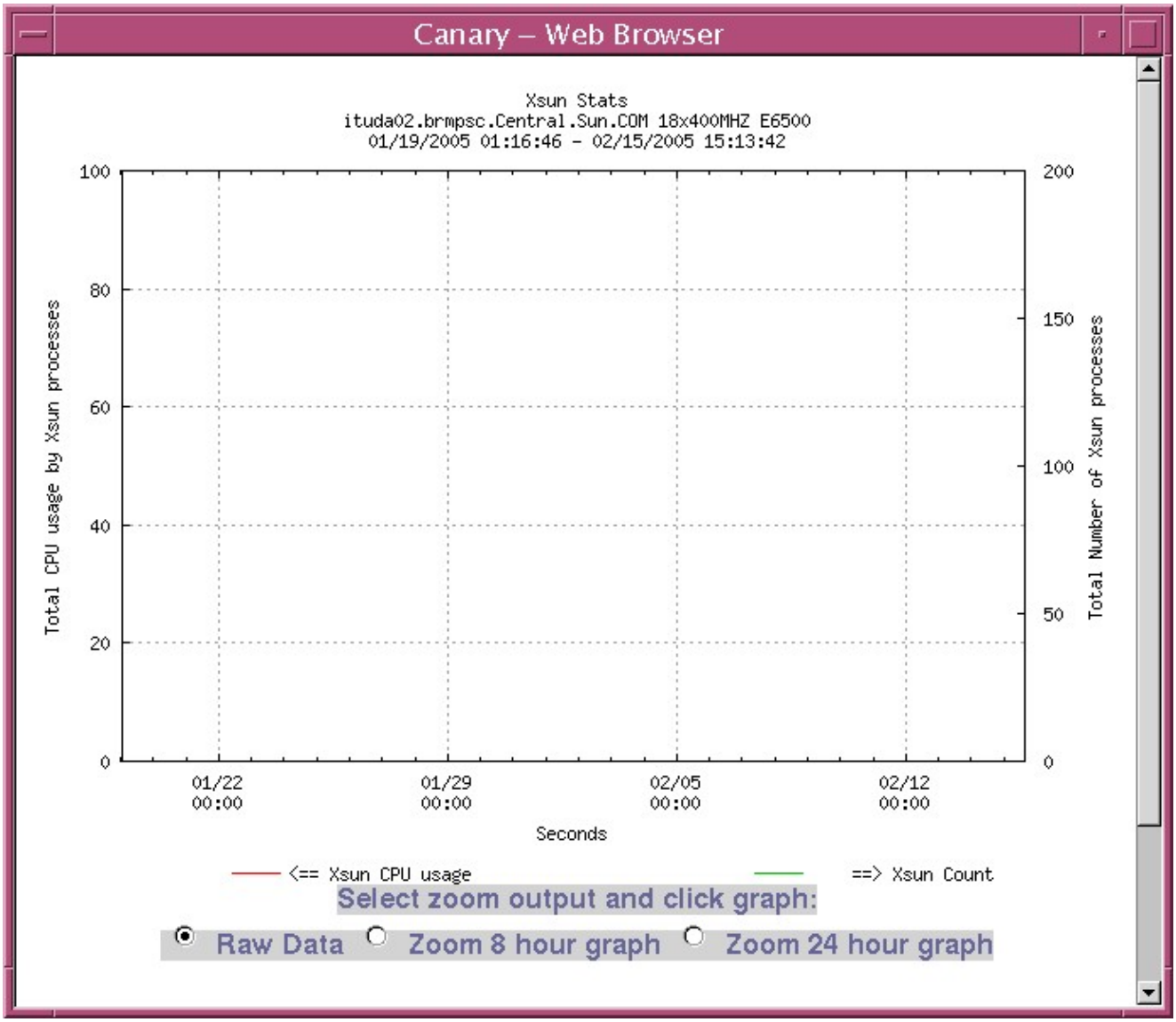
Under normal usage, the red and green lines should rise and fall together. A large discrepancy between the

two, this indicates a problem that requires investigation.

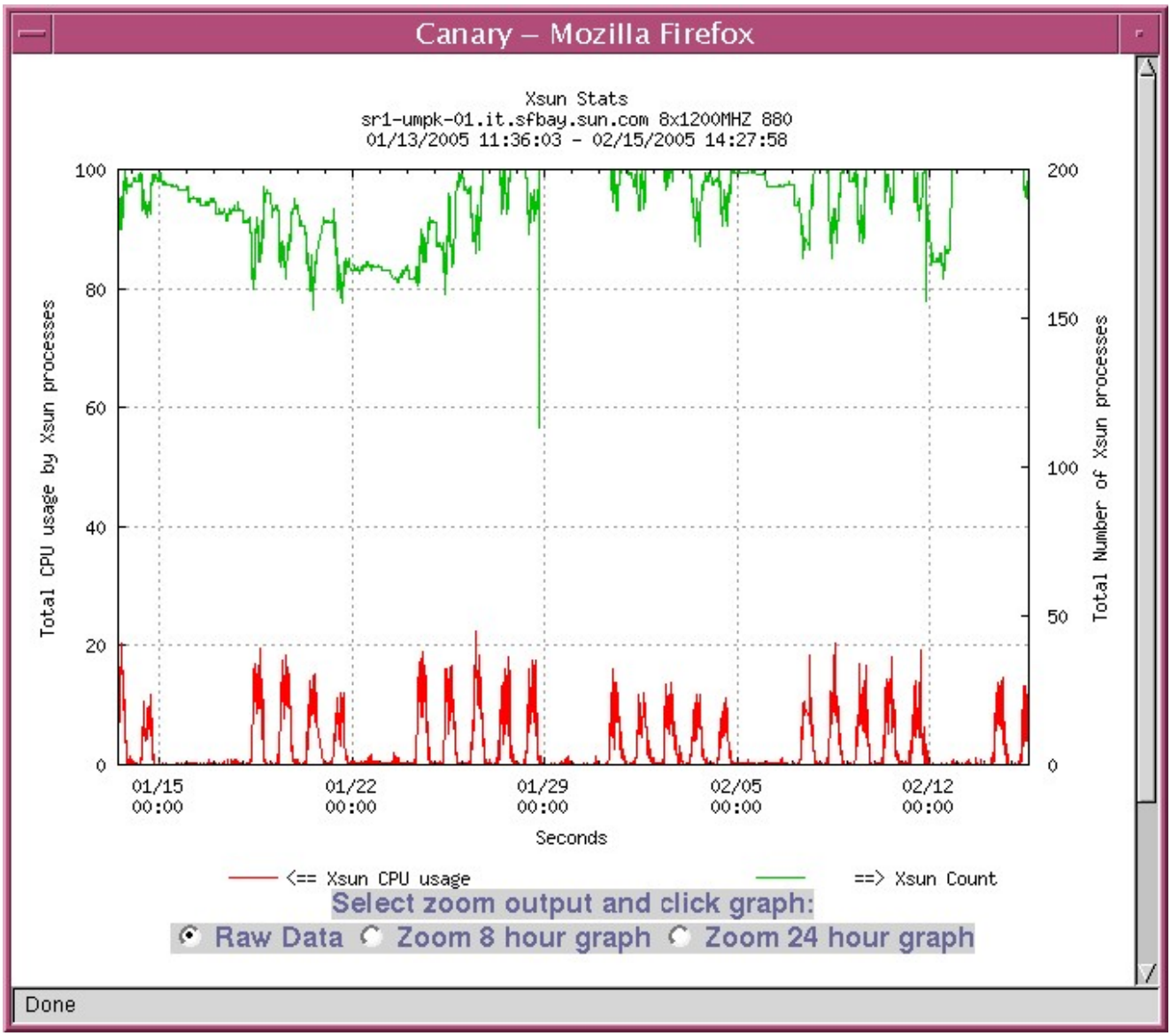


3.5.7: Xsun Statistics Graph

Xsun statistics are also gathered by Project Canary software. A typical data center server will not have X servers running on it unless it has a console so the graph will normally be empty:



A Sun Ray server or diskless workstation server that must display X sessions remotely will have quite a different graph:



The red lines denote CPU utilization and the green lines denote the number of Xsun processes running. Typically, there is one Xsun process for each display being managed.

Note that the red lines move up and down during the workday. This particular Sun Ray server is on an engineering campus so the day "starts" a little later and "ends" a little later than it would in a standard office environment with more rigid working hours. In the off-hours and at usual "meeting times" and lunchtimes, the red lines should drop significantly. If they do not, then there is a potential runaway to be investigated.

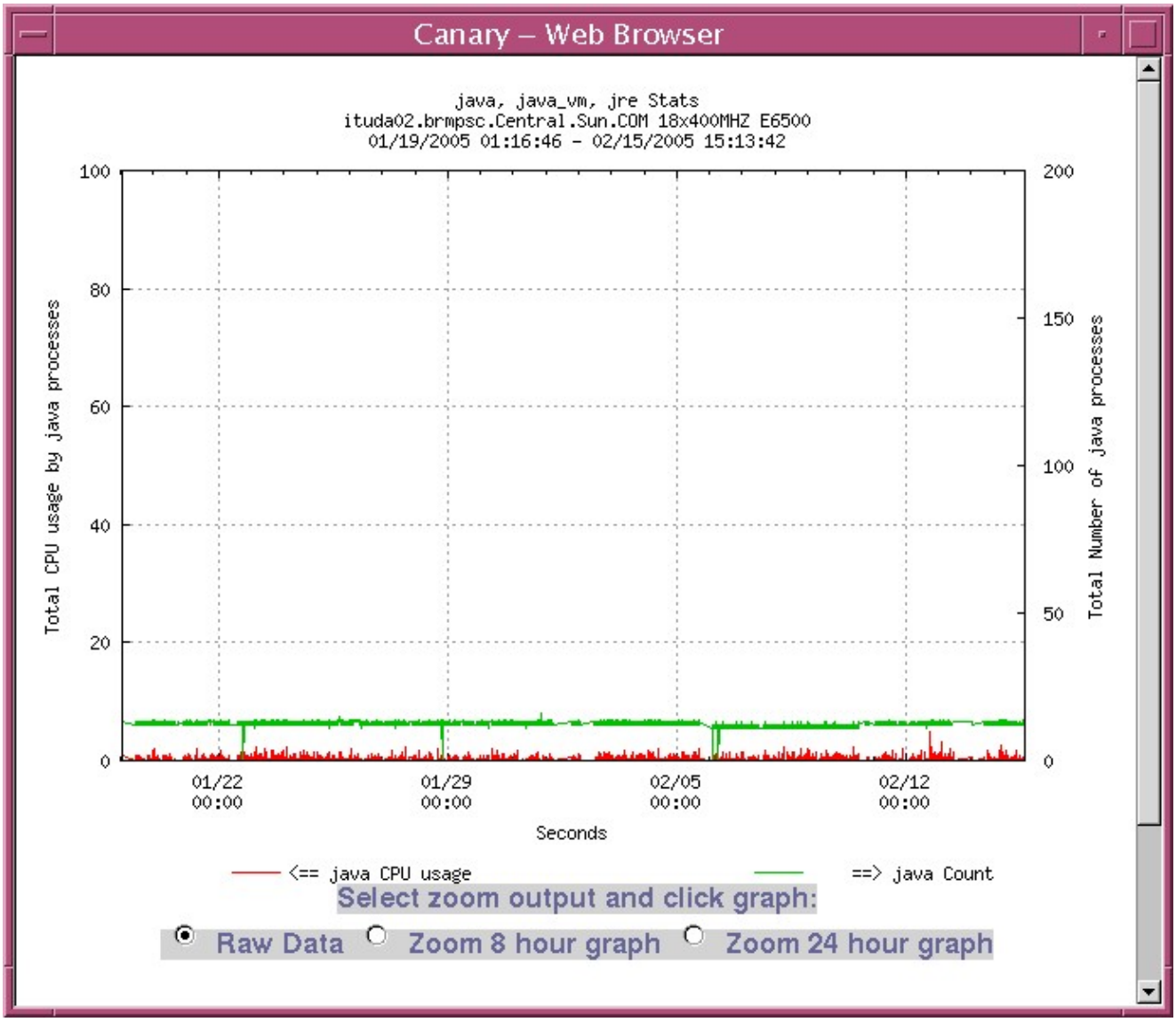
Note the large drop in the graph above where the number of Xsun processes dropped significantly on a normally busy server. One may infer that there were service outages of one form or another on that server from this graph or that many users may have been "idled out" and forcibly logged out of the Sun Ray server.

Project Canary software also tracks Java applications and the resources consumed by the Java Virtual machines running. This is useful on all types of servers since many new applications use Java as part of their operation. This is especially important to track on data center servers which may be using Java to connect applications or

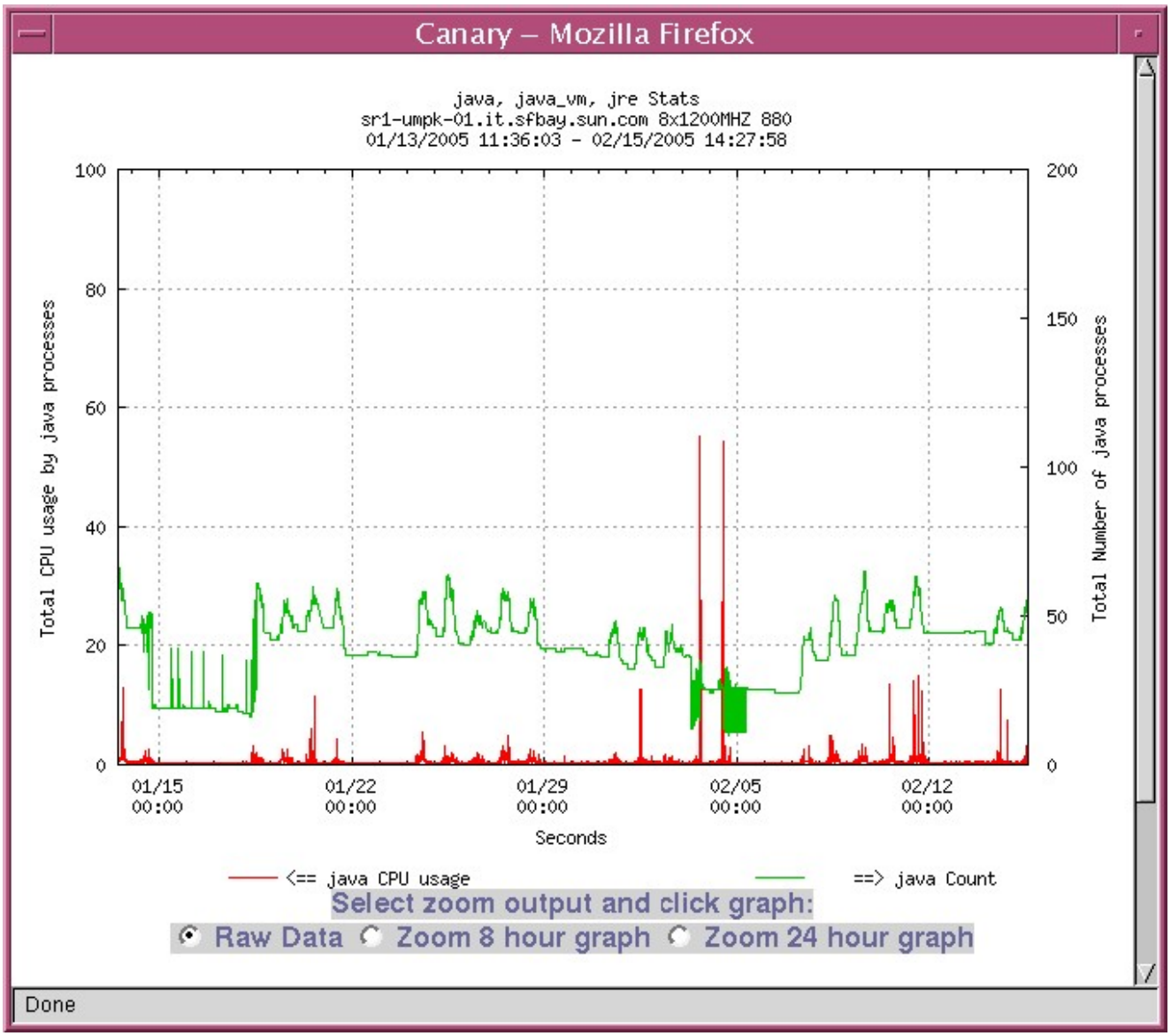
perform cross-architecture communication.

3.5.8: Java, Java Virtual Machine, and JRE Statistics Graph

A graph of Java on a typical data center server looks like this:



The red lines signify CPU utilization and the green lines indicate the number of Java applications running. Note that on the data center server, Java use is low.
On the Sun Ray server, with users running Java applications and downloading applets to execute through web browsers, the Java graph is startlingly different, as shown below:

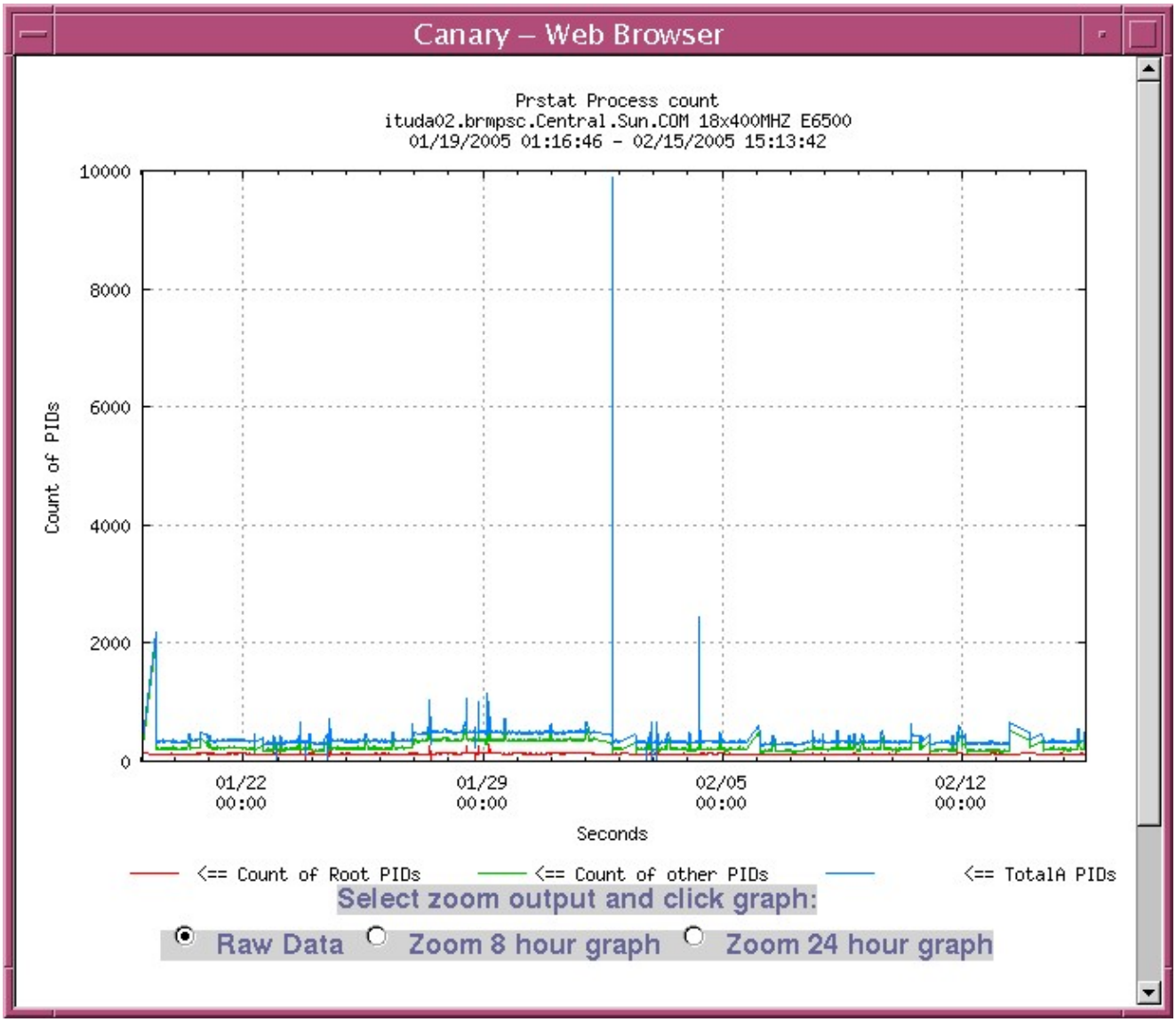


Again, the red lines indicate CPU utilization and the green lines indicate the number of Java processes running. Since many more user applications use Java, from custom in-house applications to access various databases to StarOffice and OpenOffice, the graph of a Sun Ray server will almost always show more Java processes than a typical data center server.

Note the two spikes where Java CPU utilization skyrocketed. This indicates a potential runaway process that should be investigated.

3.5.9: PRSTAT Process Count Graph

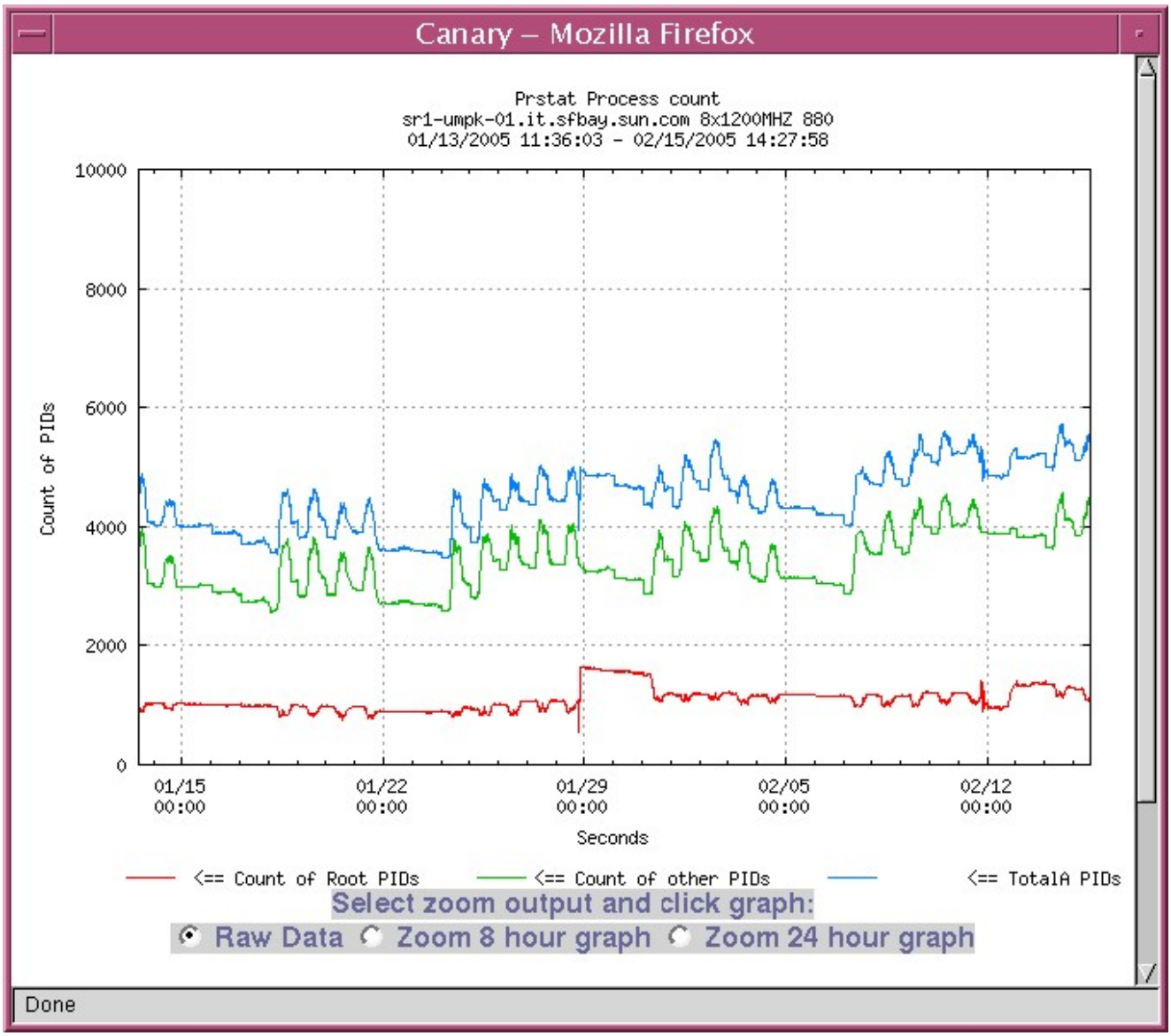
This graph shows the PRSTAT Process Count for a data center server, as shown below:



The X-axis shows the number of processes on the server and the Y axis shows time. The red line shows root-owned processes and the green line shows processes owned by "other than root" users. The blue line is the sum of both the red and green lines and shows a total of all processes running on the server.

Note the two spikes in the middle of the graph. These spikes are indicative of a process that went out of control and spawned multiple child processes.

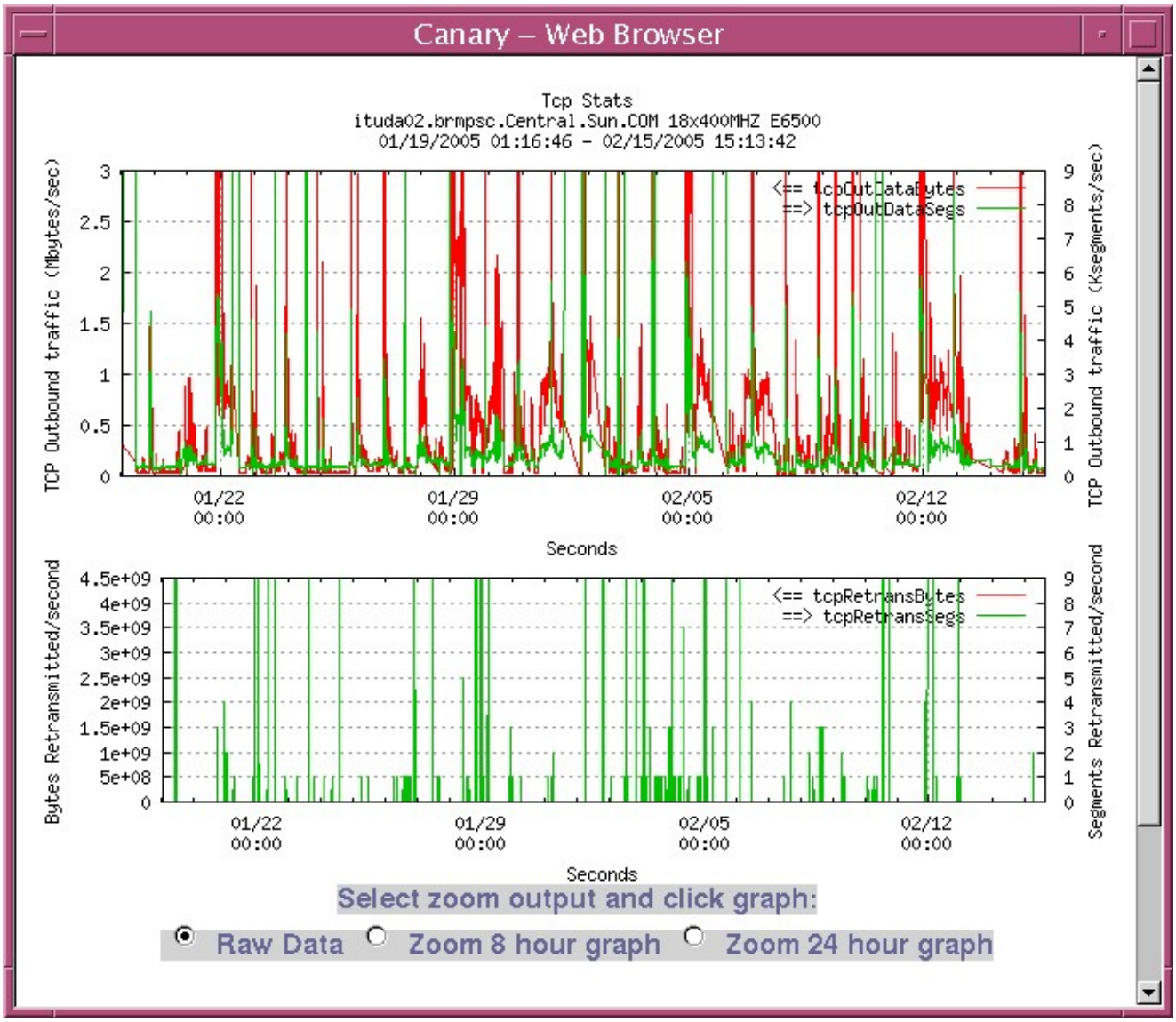
The graph for the Sun Ray server shows many more processes. This is mostly due to the fact that users are utilizing the Sun Ray server to drive their desktops.



Note the number of root-owned processes. Each Sun Ray client attached to the Sun Ray server requires an entire desktop environment be launched for them including X server processes, some of which are owned by root on the Sun Ray server. Note also the green line denoting user-level processes rising in the morning as users arrive at their offices, dipping at "typical meeting times" as well as at lunchtime, and then falling again as users log out of their desktops and go home for the evening.

3.5.10: TCP Statistics Graph

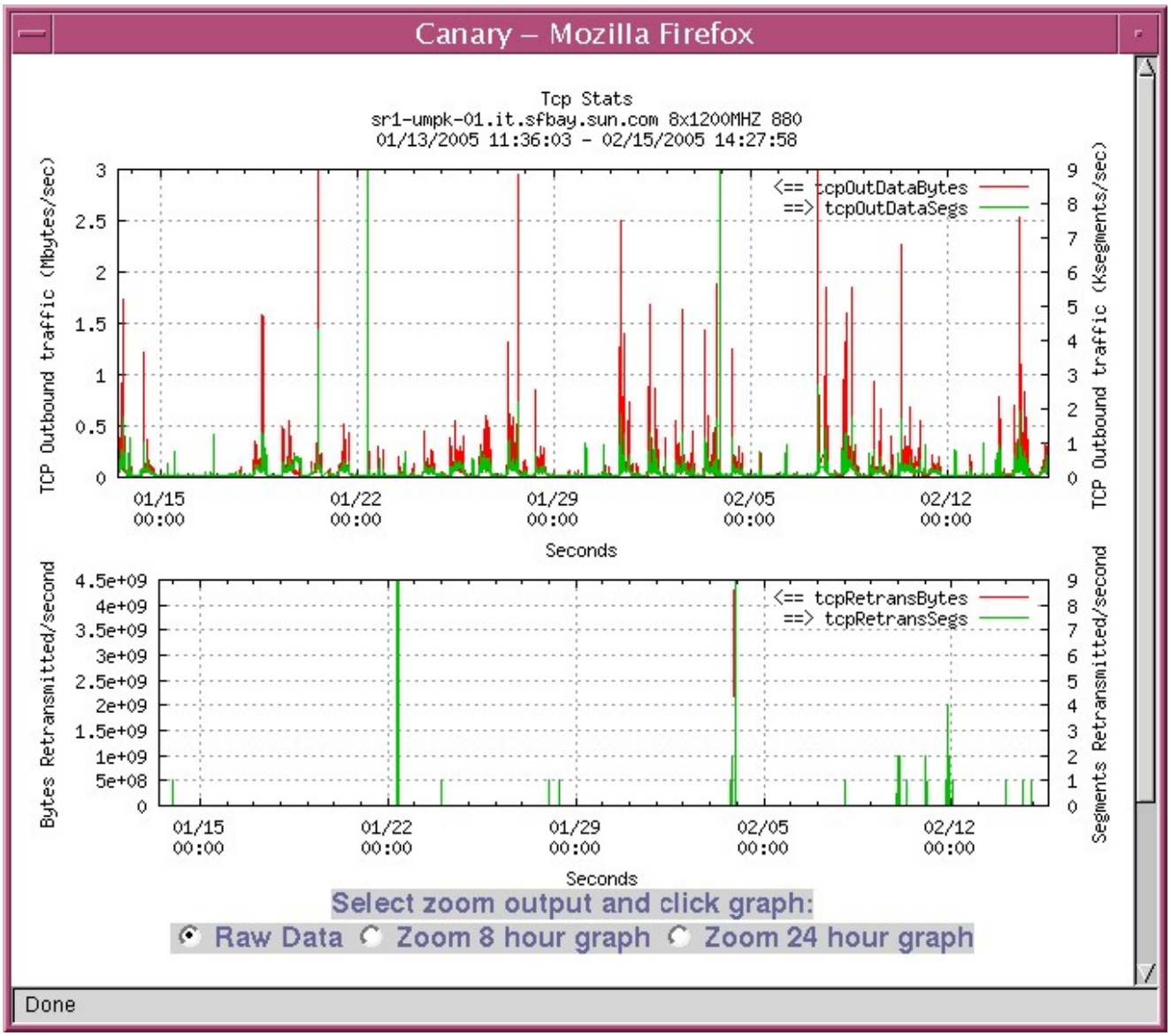
The graph shown below displays TCP statistics for the example data center server:



These statistics are collected from `netstat(1M)` and other tools. The top graph plots outbound network traffic and the bottom graph plots the amount of data resent by the data center server.

Traffic appears heaviest at night, typically when backups are done, and retransmits are also heavy at that time.

The TCP Statistics Graph for the example Sun Ray server looks slightly different:

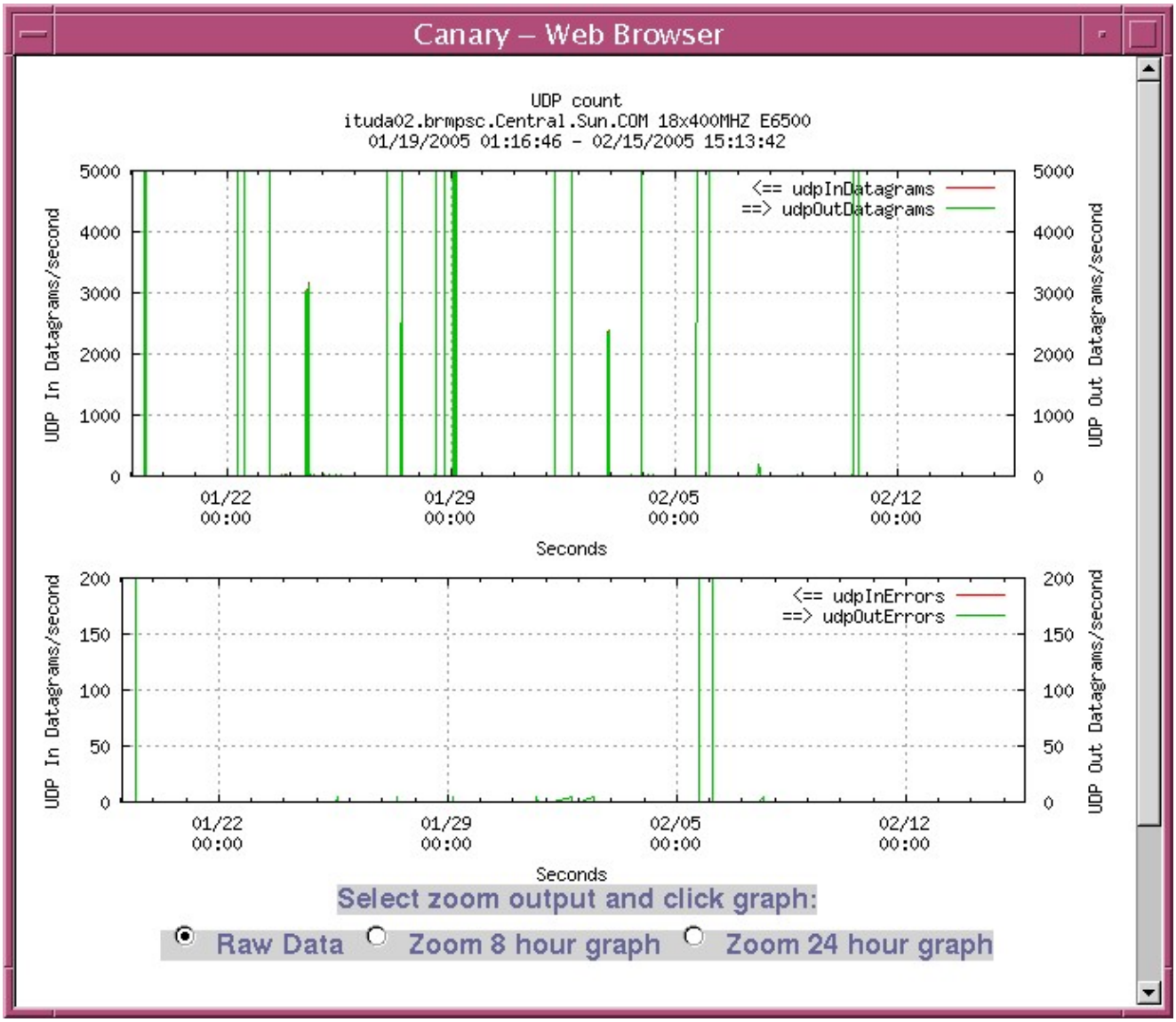


Most Sun Ray servers do not have storage attached to them so they do not show network activity spikes at night when backups run. However, it appears from this graph that around midnight, there is some network activity that causes the Sun Ray server to resend a number of packets. This could be because other servers are being backed up over the network, however, this would require an investigation to come to a definite conclusion. The green spikes in the second graph, however, show that there is a problem that only occurs at night which should be investigated.

Note also that the traffic in the top graph parallels the work day. With the graphs provided by Project Canary software, a user profile could be built of Sun Ray users; when do they arrive in the office? When do they go home? When are they using the server the most? All of this data can be inferred from the graphs.

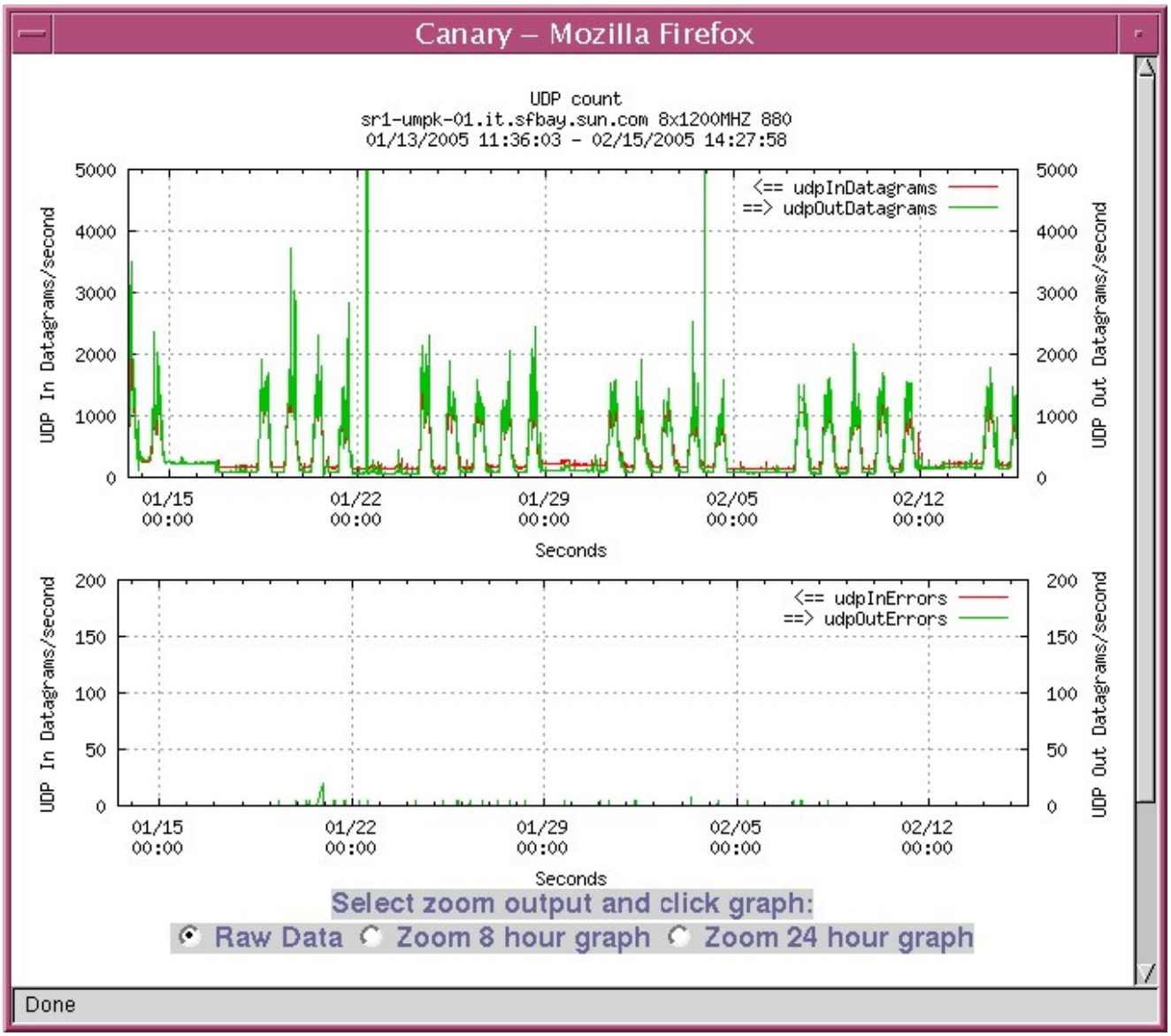
3.5.11: UDP Count Graph

Below is a graph showing the UDP count for the Data Center Server:



Note again, there are spikes at night, which is typically when backups are run. Also, errors spike upwards at that time, showing that there may be too much traffic on the network at those hours.

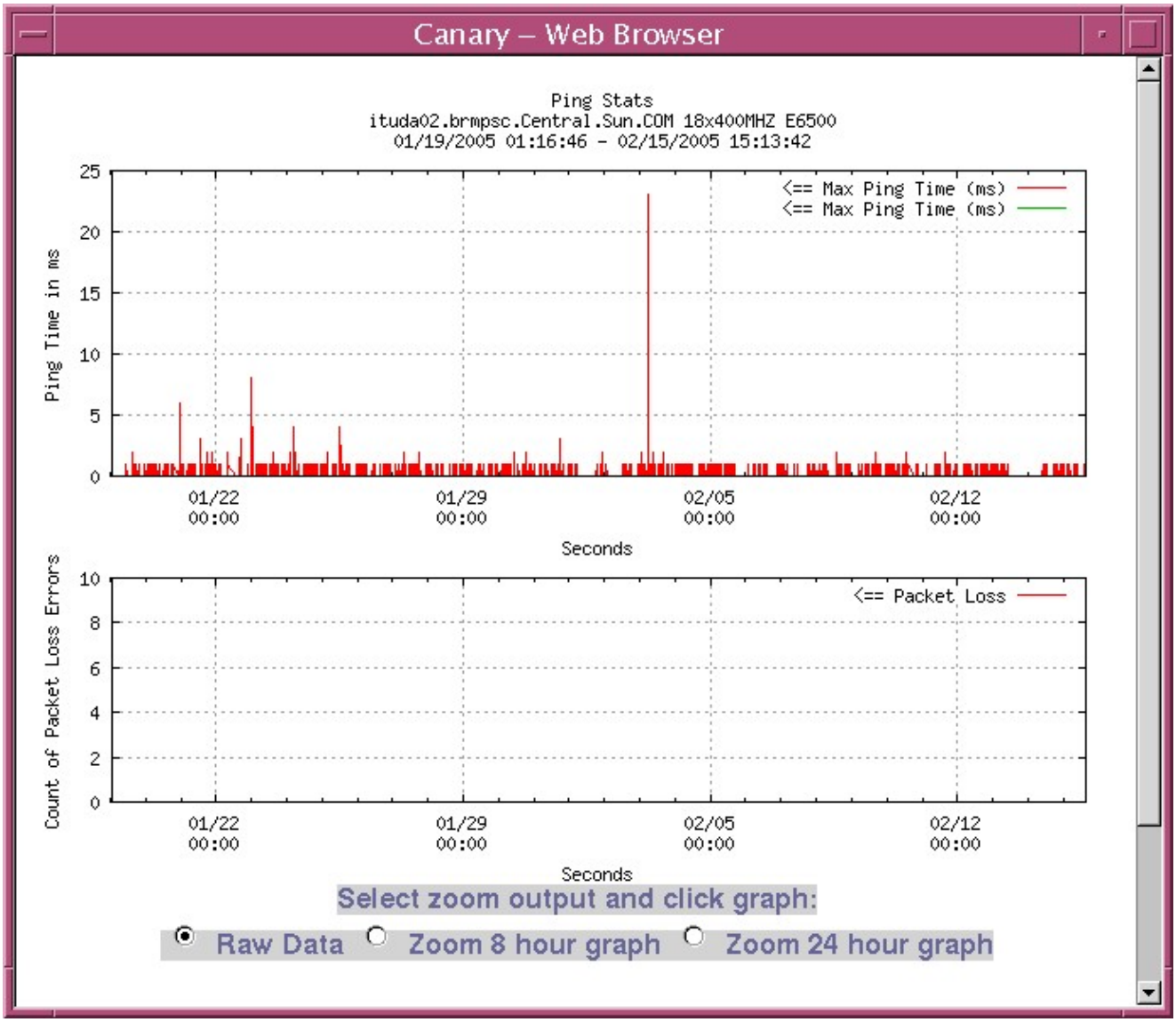
The UDP count graph for the Sun Ray server again parallels the work day:



Note that there is very little traffic at night when users are at home sleeping. Most of the traffic comes in bursts during the day, an hour with traffic rising followed by an hour with little network traffic.

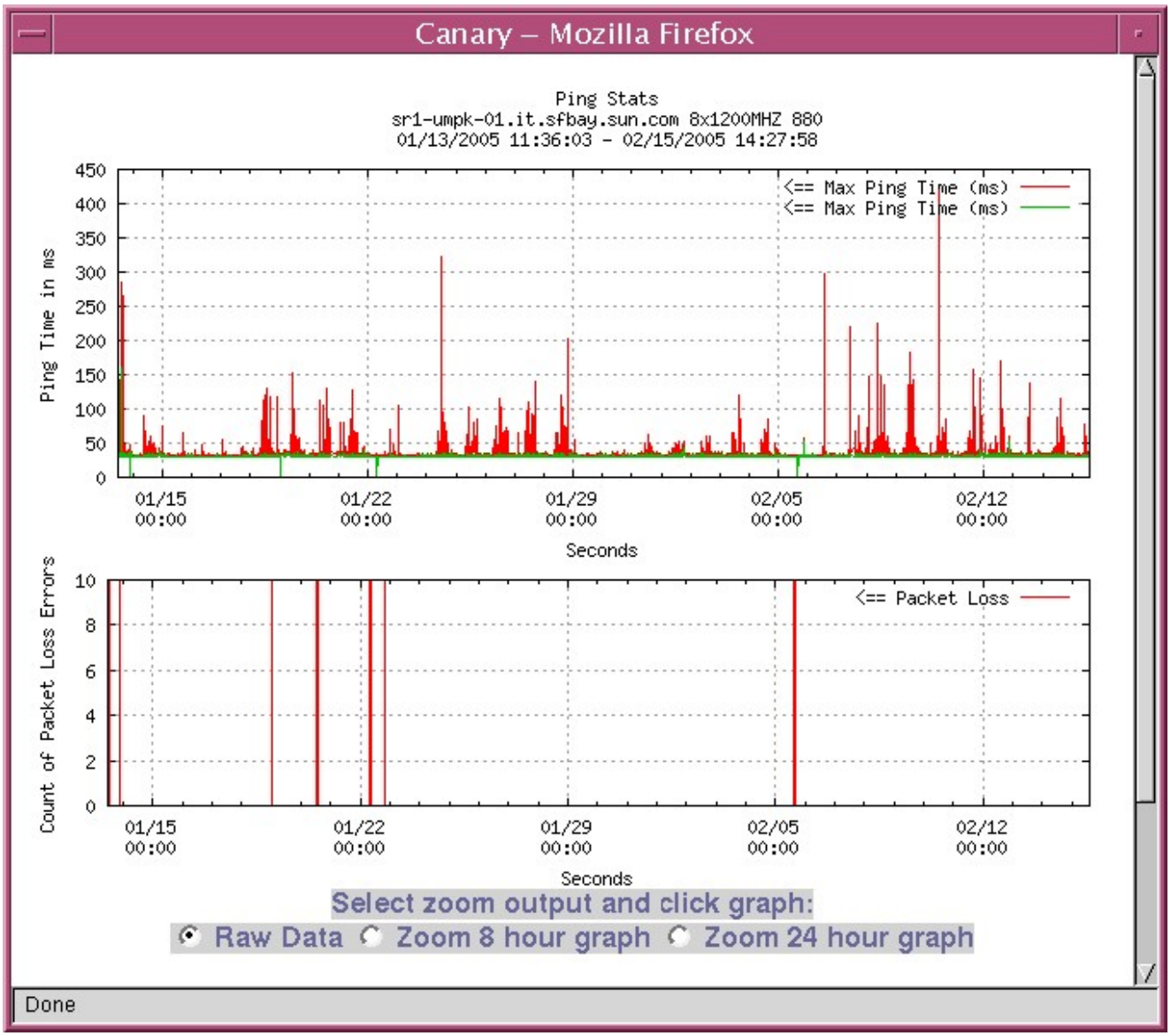
3.5.12: Ping Statistics Graph

The graph below shows ping statistics for the Data Center server:



The network appears to be fairly speedy and there is no packet loss, however, there is a spike in the ping time graph that indicates there was a network glitch as the client somehow took a long time to reach the Project Canary central data server around February 04, 2005. This indicates a possible network glitch at that time. Other graphs in the Dashboard view could help confirm or deny this.

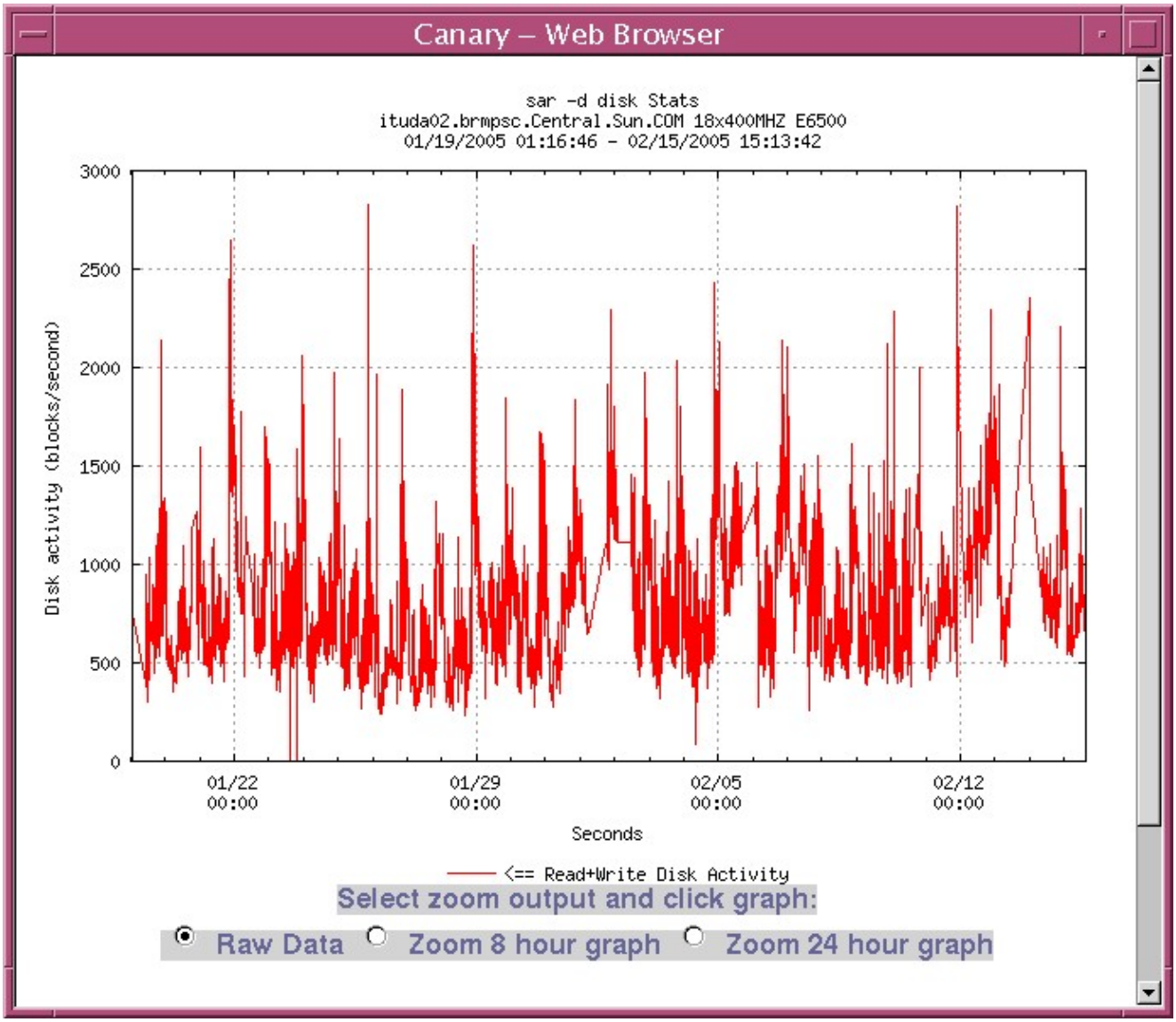
The ping statistics graph for the Sun Ray server is shown below:



This graph shows that during the work day, when multiple users are attached to the Sun Ray server, there is increased traffic which leads to longer ping times. Note that the increased ping times almost directly correlate to graphs shown earlier in this *User Guide and Manual*.

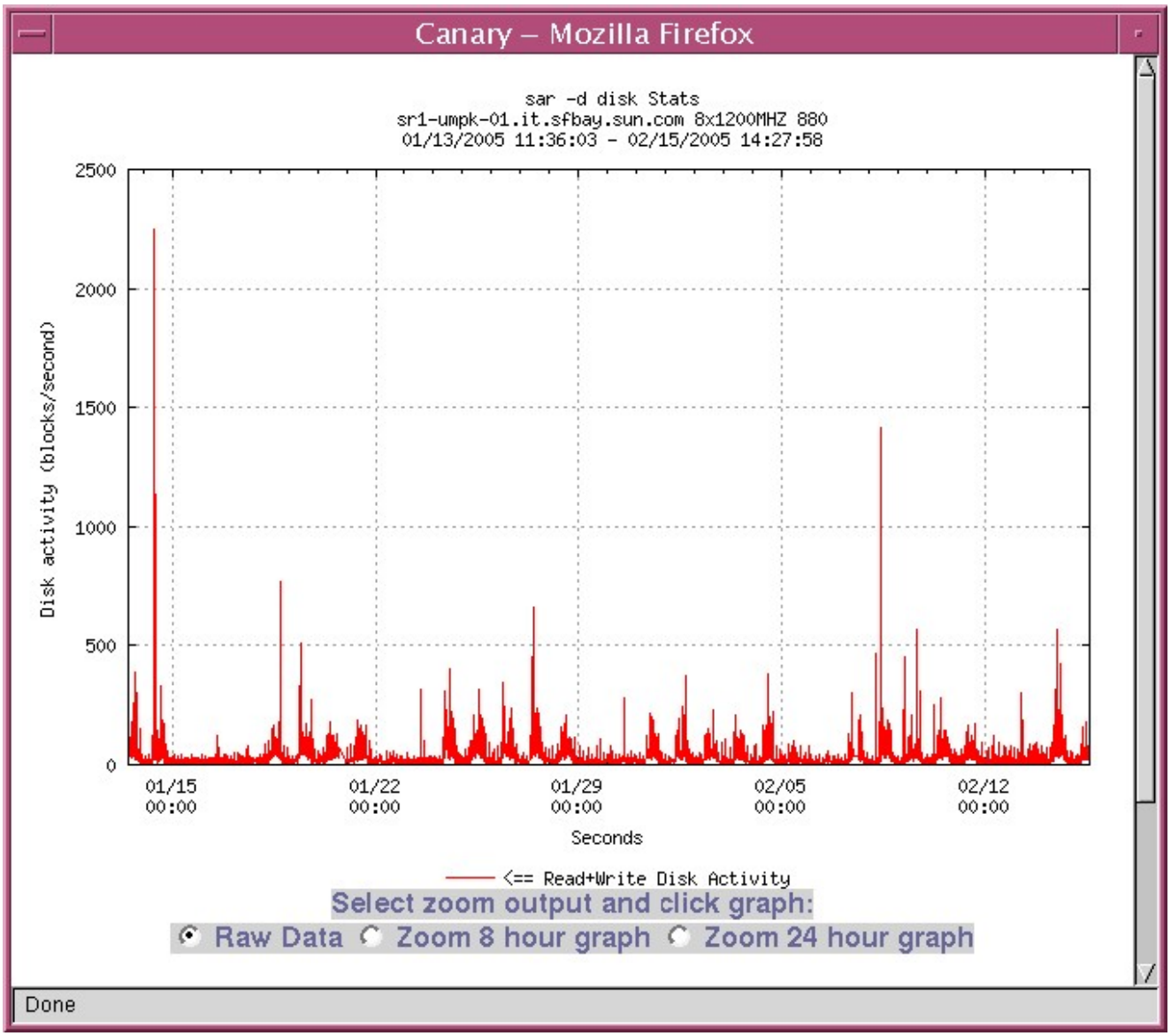
3.5.13: "sar -d" Disk Statistics Graph

The graph below shows disk statistics for the example data center server:



From this graph, it would appear there there are spikes in disk activity every few hours and that at night, disk activity spikes. As mentioned before in this User Guide, that could simply indicate that backups are running.

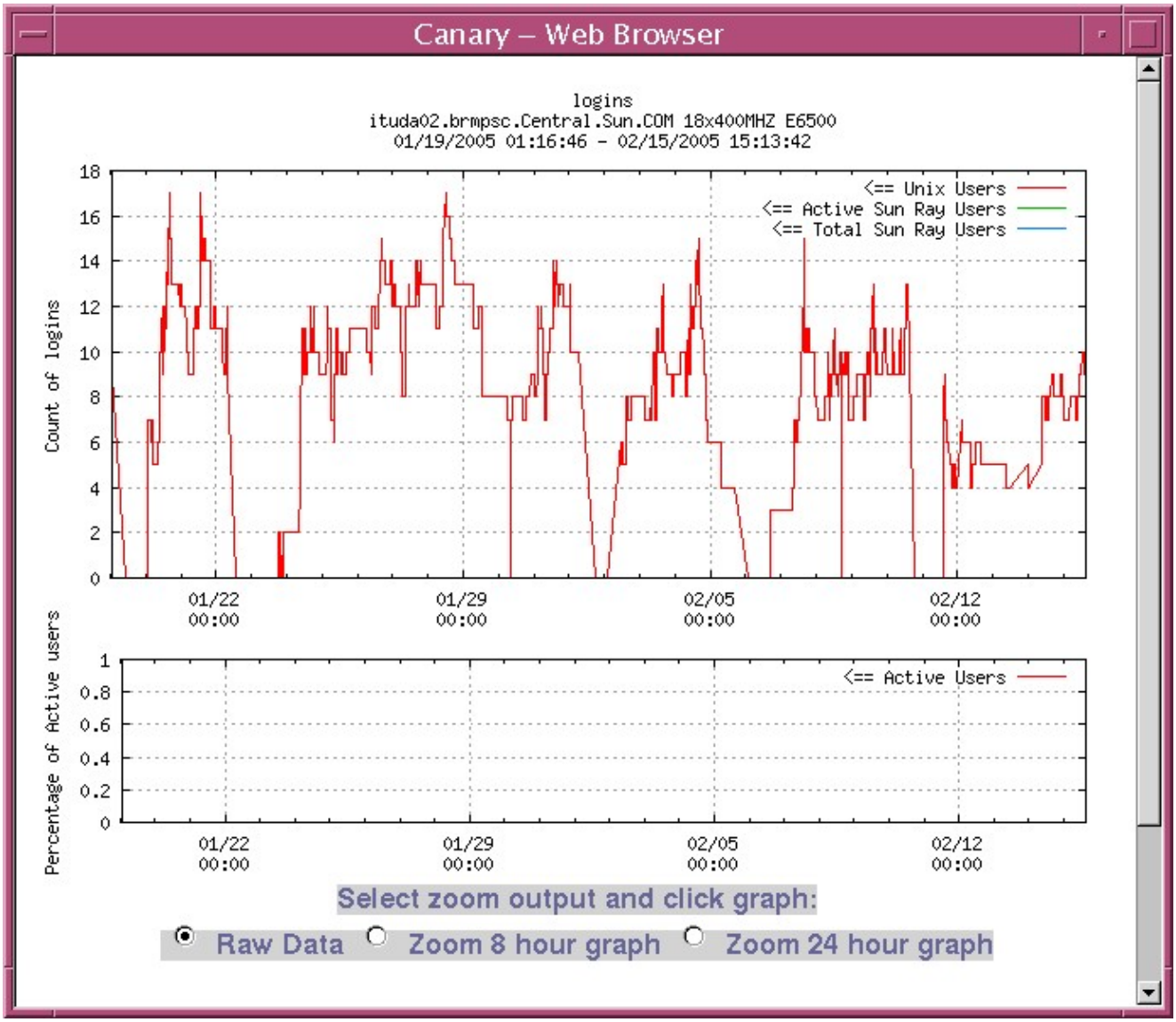
The disk statistics graph for the Sun Ray server looks quite different:



Most of this activity is the Sun Ray server utilizing swap. This particular Sun Ray server does not do double-duty as a fileserver. It is dedicated to users. Note, again, the spikes in the graph that correspond to previous graphs, indicating when users do their work during the day.

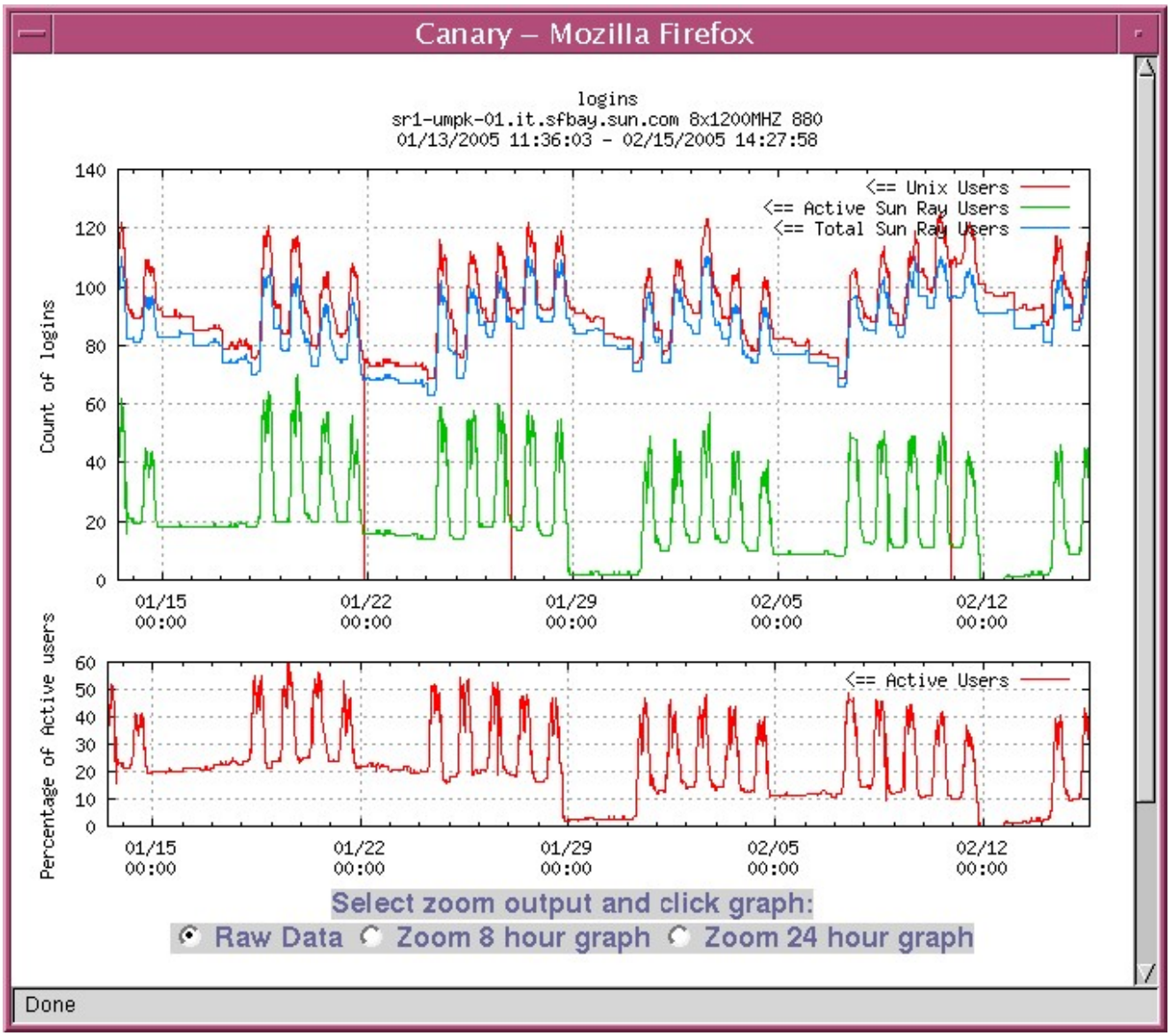
3.5.14: Logins Graph

Below is a graph of logins to the example Data Center server:



The Data Center server is not running Sun Ray server software so there will be no Sun Ray logins shown on this graph. However, there are a fair number of users logging into this machine at various times during the day. Logins on this graph also correspond to other Sun Ray graphs of a typical work day.

Below is a graph showing the number of logins on an example Sun Ray server:



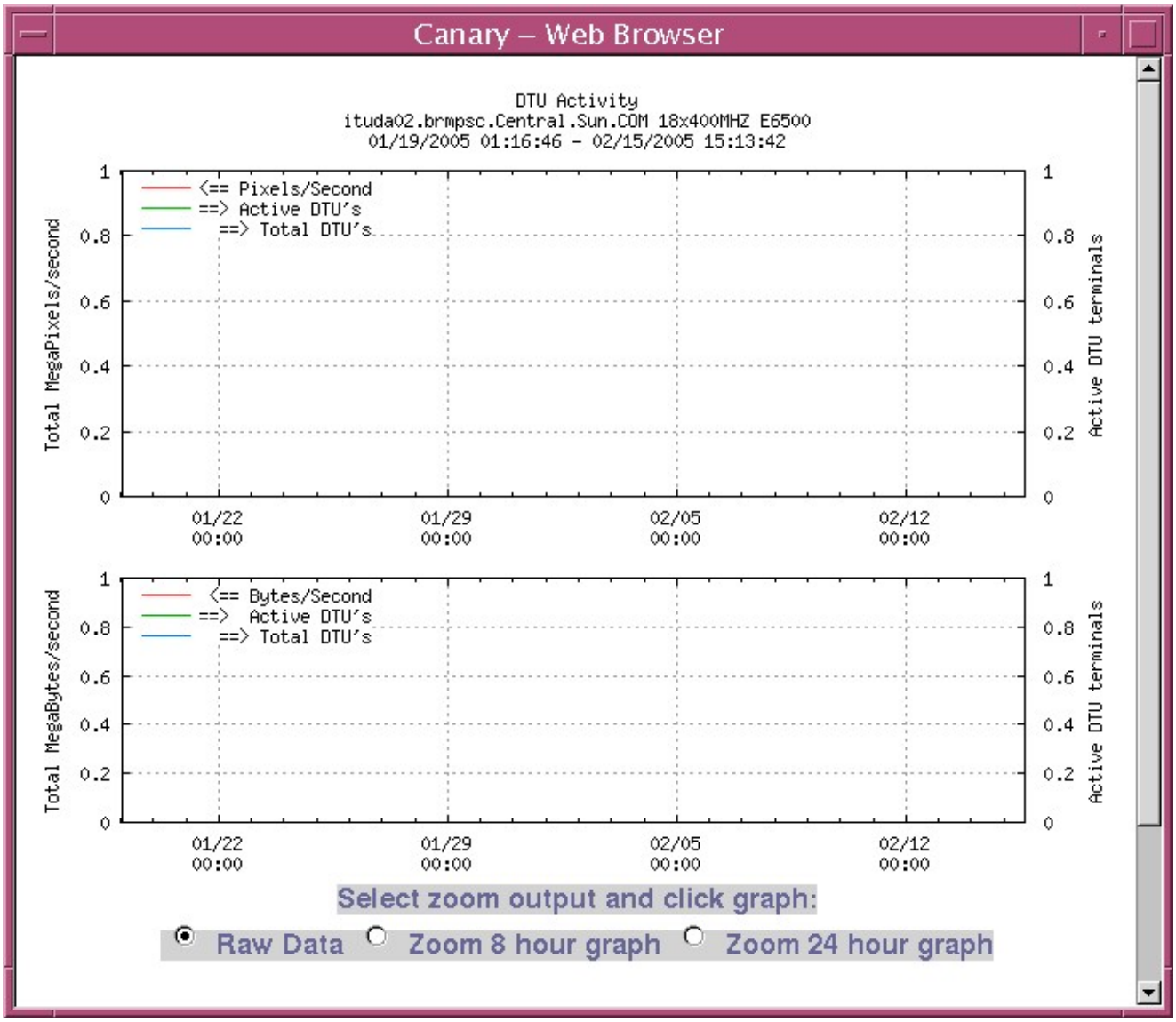
Again, as in previous graphs, logins to the Sun Ray server follow a "work day" pattern. Interestingly, there are users logging into the Sun Ray server who do not appear to be Sun Ray users.

This graph can be used to help enforce policies if needed; if a machine is supposed to be dedicated to Sun Ray usage and other users are logging into it to use CPU time, this graph would show that so that a login policy could be enforced.

Note that there are far more users using the Sun Ray server than on the data center server.

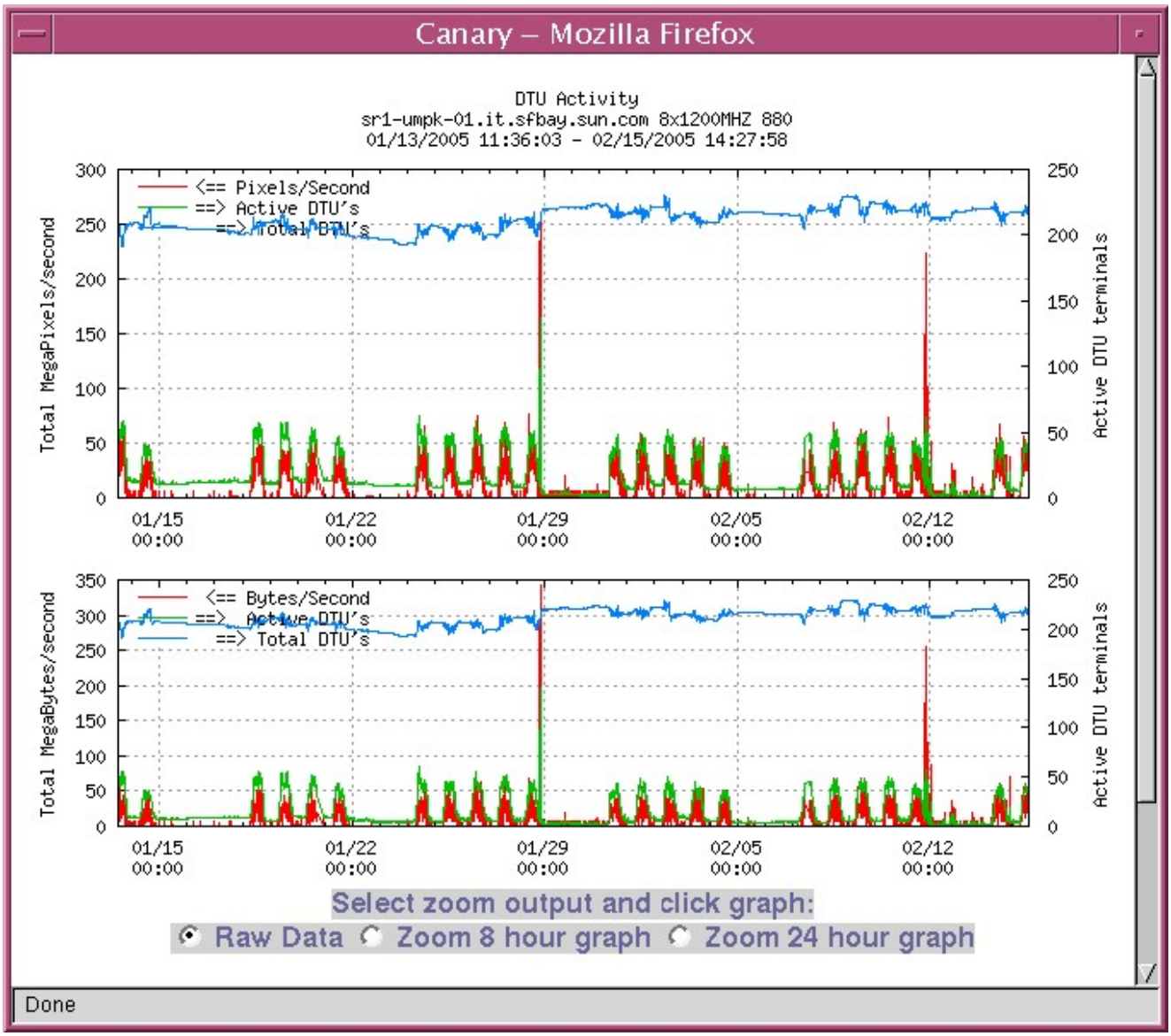
3.5.15: DTU Activity Graph

Below is a graph of DTU activity on the example data center server:



Since the data center server is not running Sun Ray Server Software, there will be no activity.

Below is a graph of DTU activity on the example Sun Ray server:



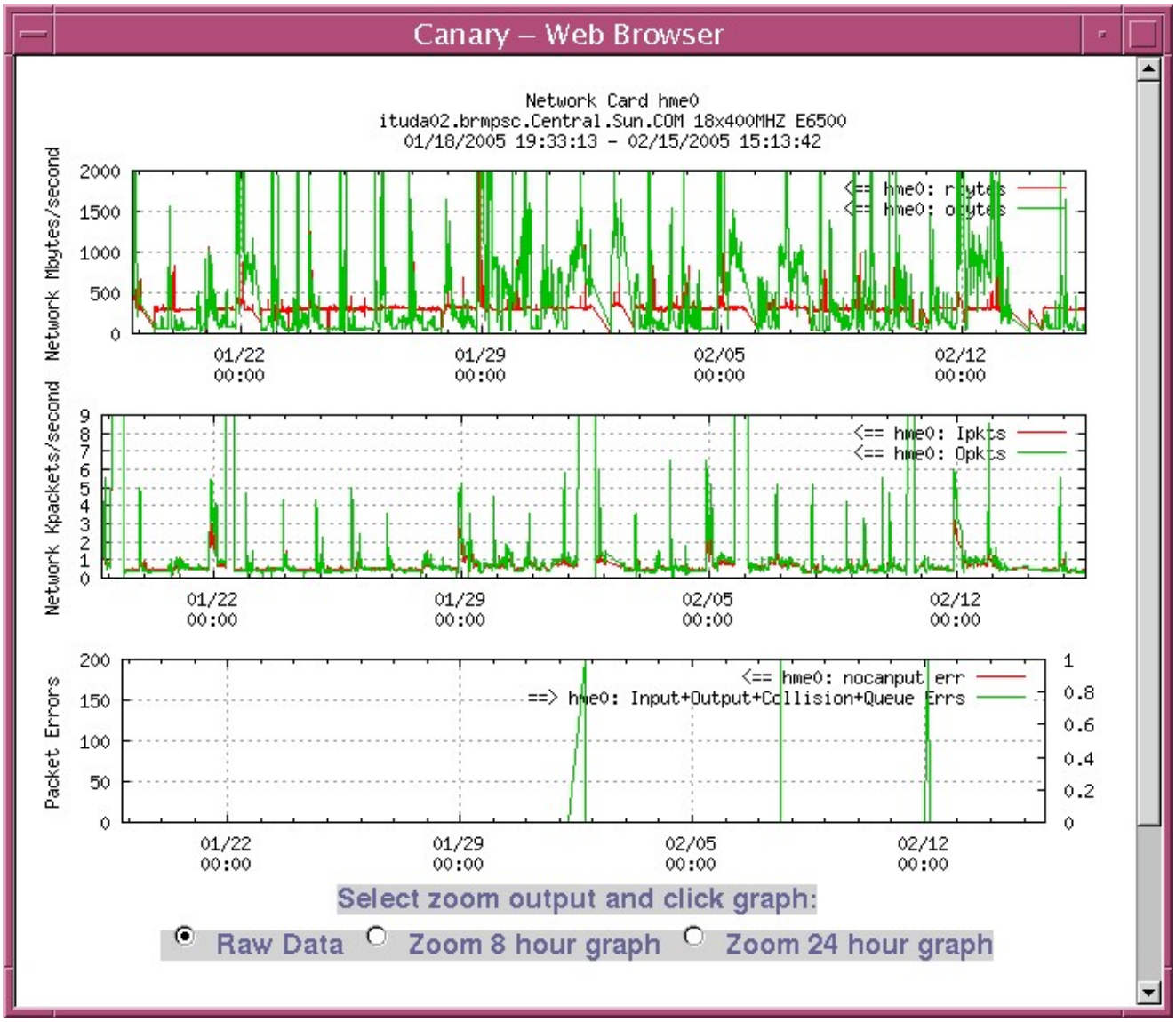
Note that again, activity follows the work day for users of this server.

The top graph shows how many DTU terminals are open (on the right side, using the green line) and the red line shows how many pixels per second are being drawn on Sun Ray terminals. The blue line shows the total number of DTUs.

The bottom graph is the same as the top graph, only instead of showing how many pixels per second are being drawn, the red line shows how many megabytes of data per second the Sun Ray server is sending to Sun Ray terminals to be drawn on the screen.

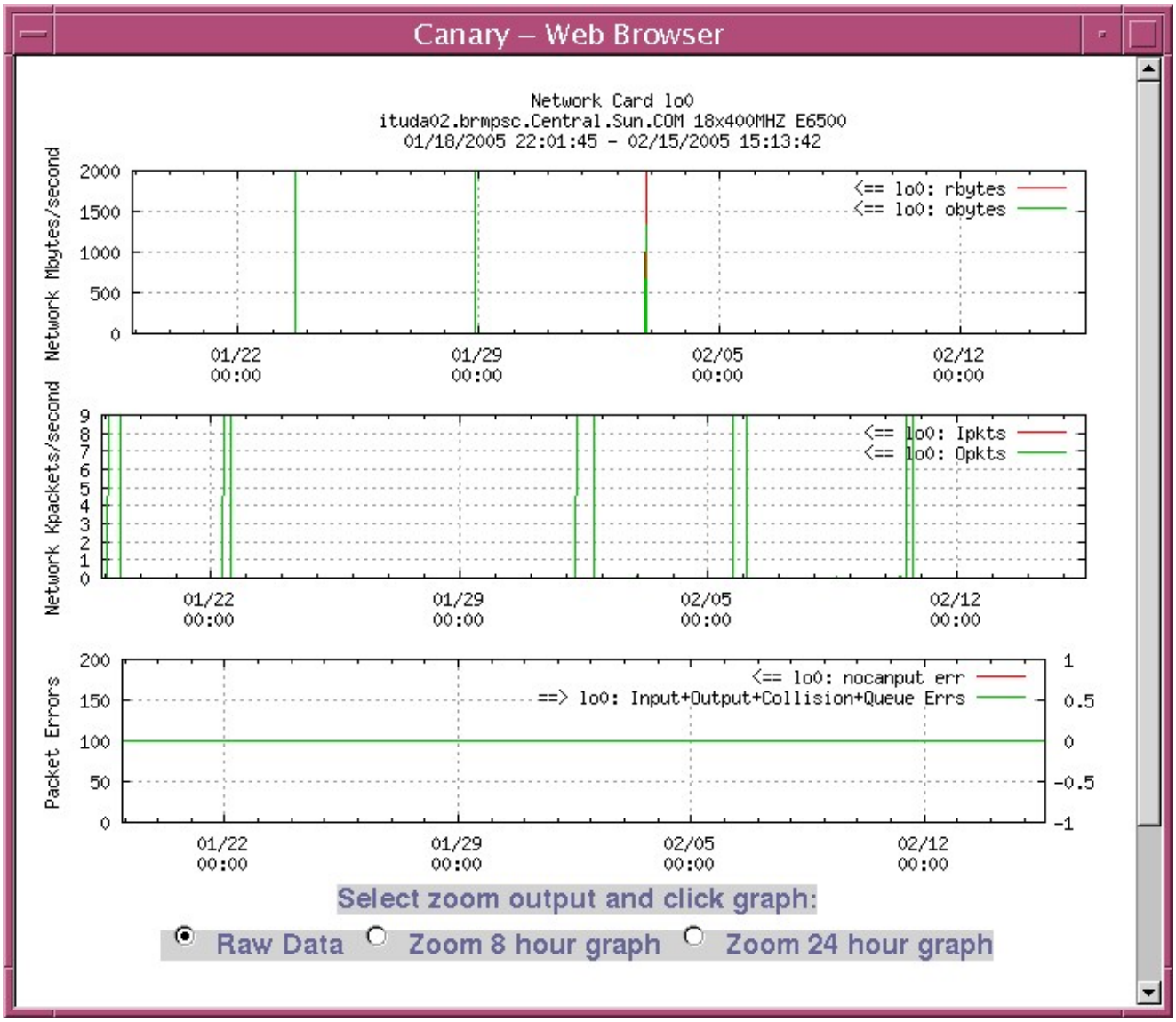
3.5.16: Ethernet Card Performance Graphs

The graph below shows network card performance for the example data center server:



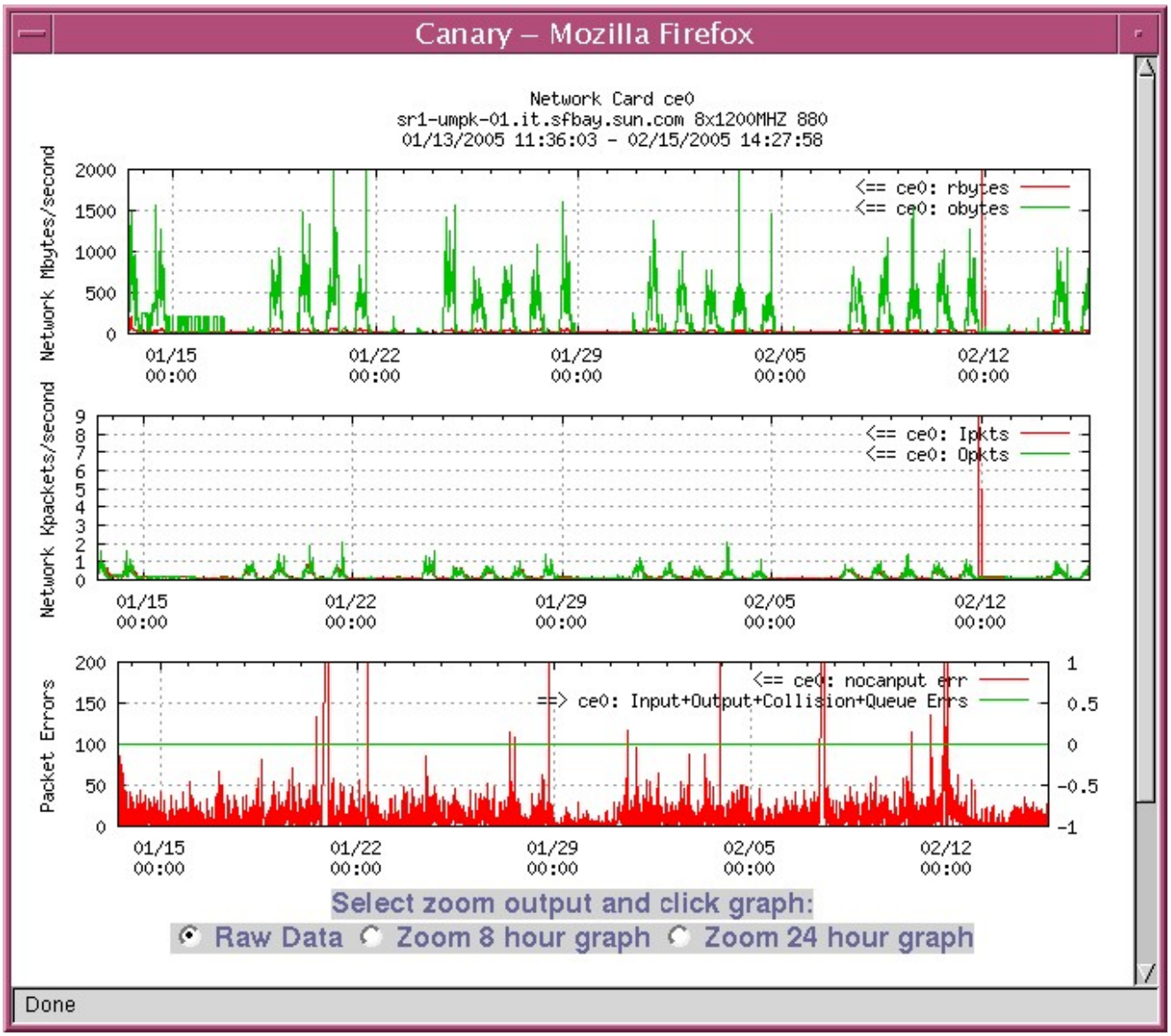
The graph above shows that the data center server is sending a lot of data every few hours somewhere.

Below is a graph of the loopback interface on the example data center server:



There should be no collisions on the loopback interface. Interestingly, there are traffic spikes across it.

The graph below shows network card performance on the example Sun Ray server:

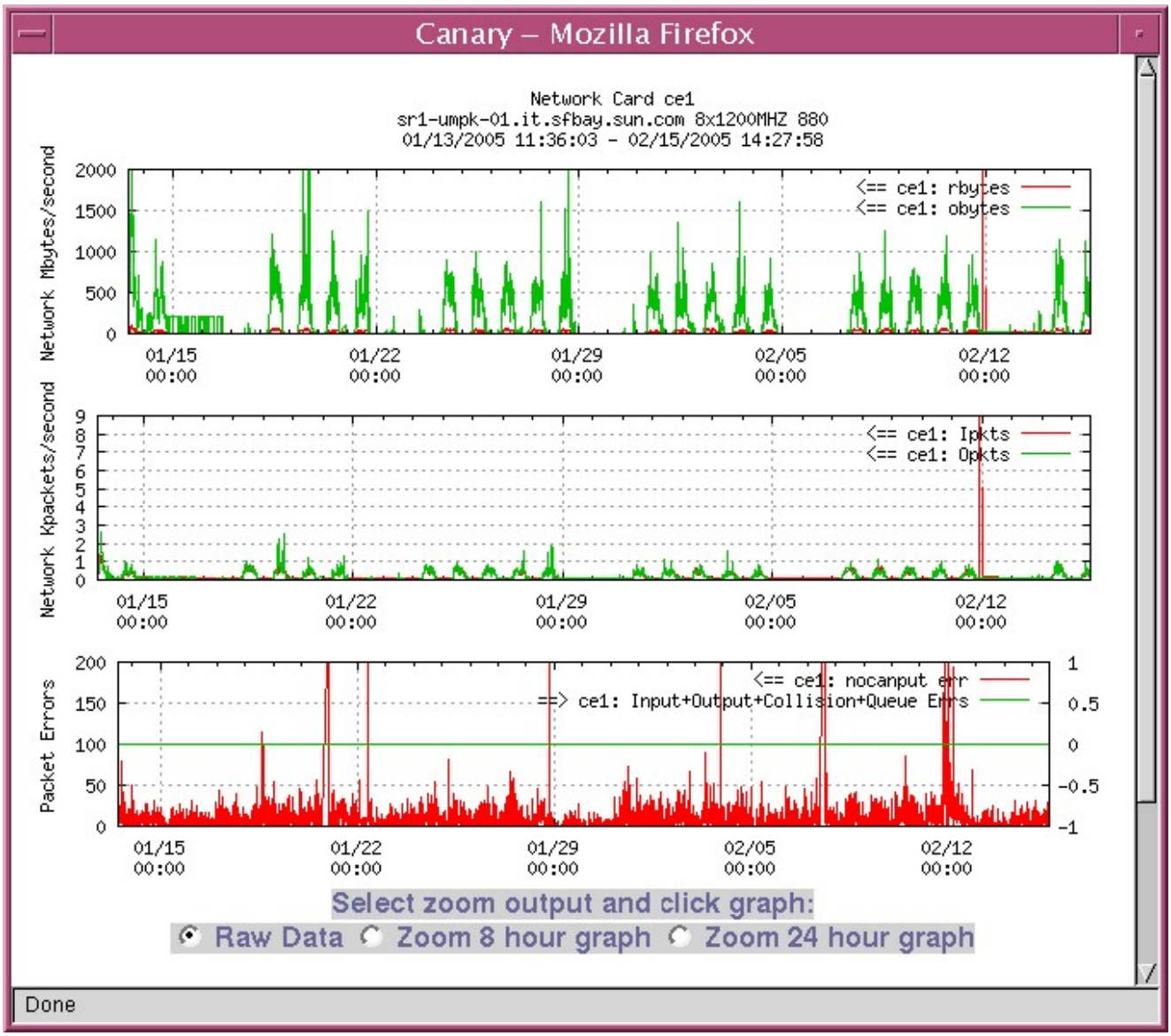


As the previous graphs have shown, network traffic spikes and falls according to the workload placed on it by the users utilizing the server.

There is a nasty spike on February 12 around midnight that should be investigated.

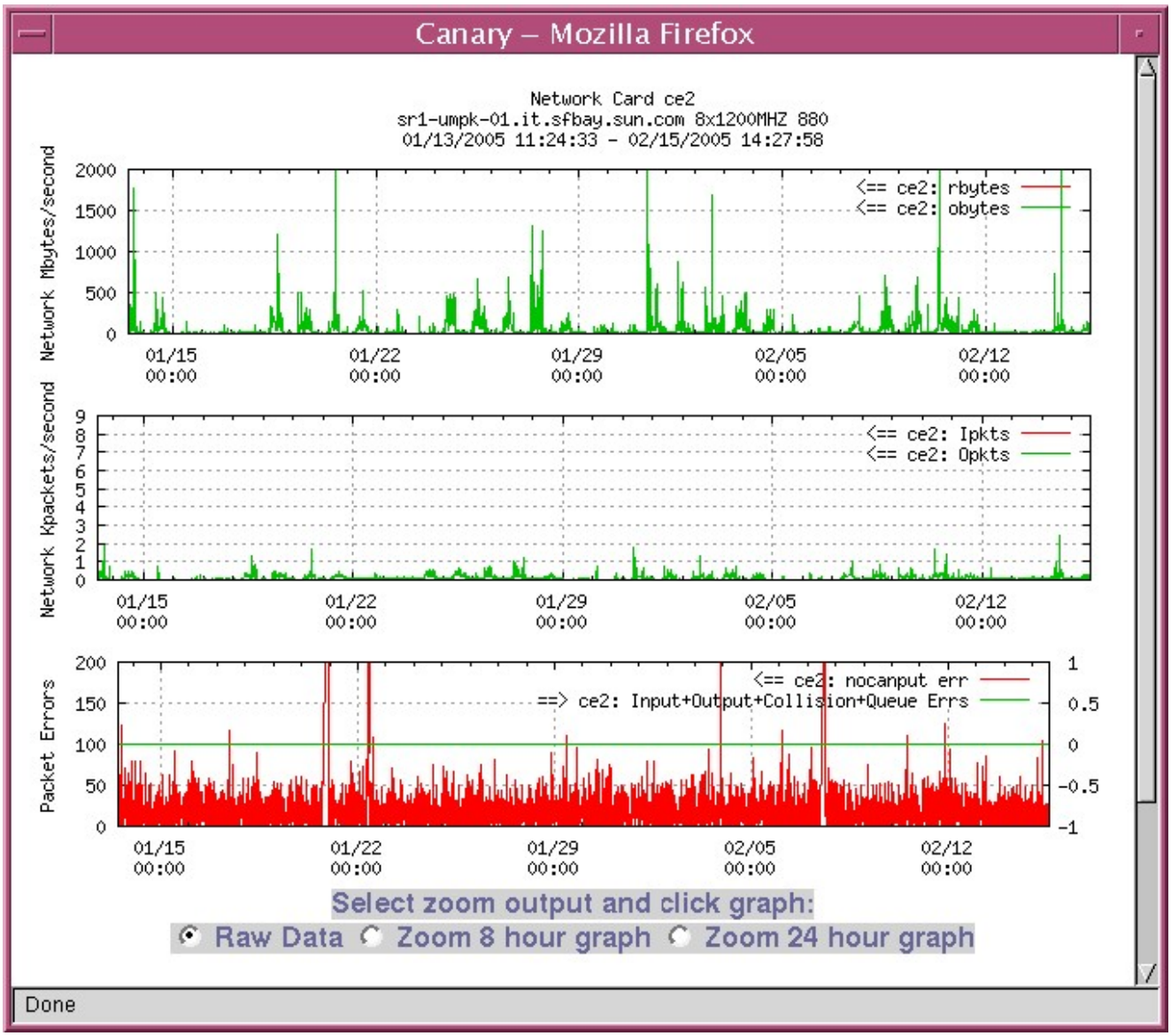
Note that the Sun Ray server shows a lot of packet errors, but traffic is generally fairly light on this interface.

Below is another network interface on the example Sun Ray server:



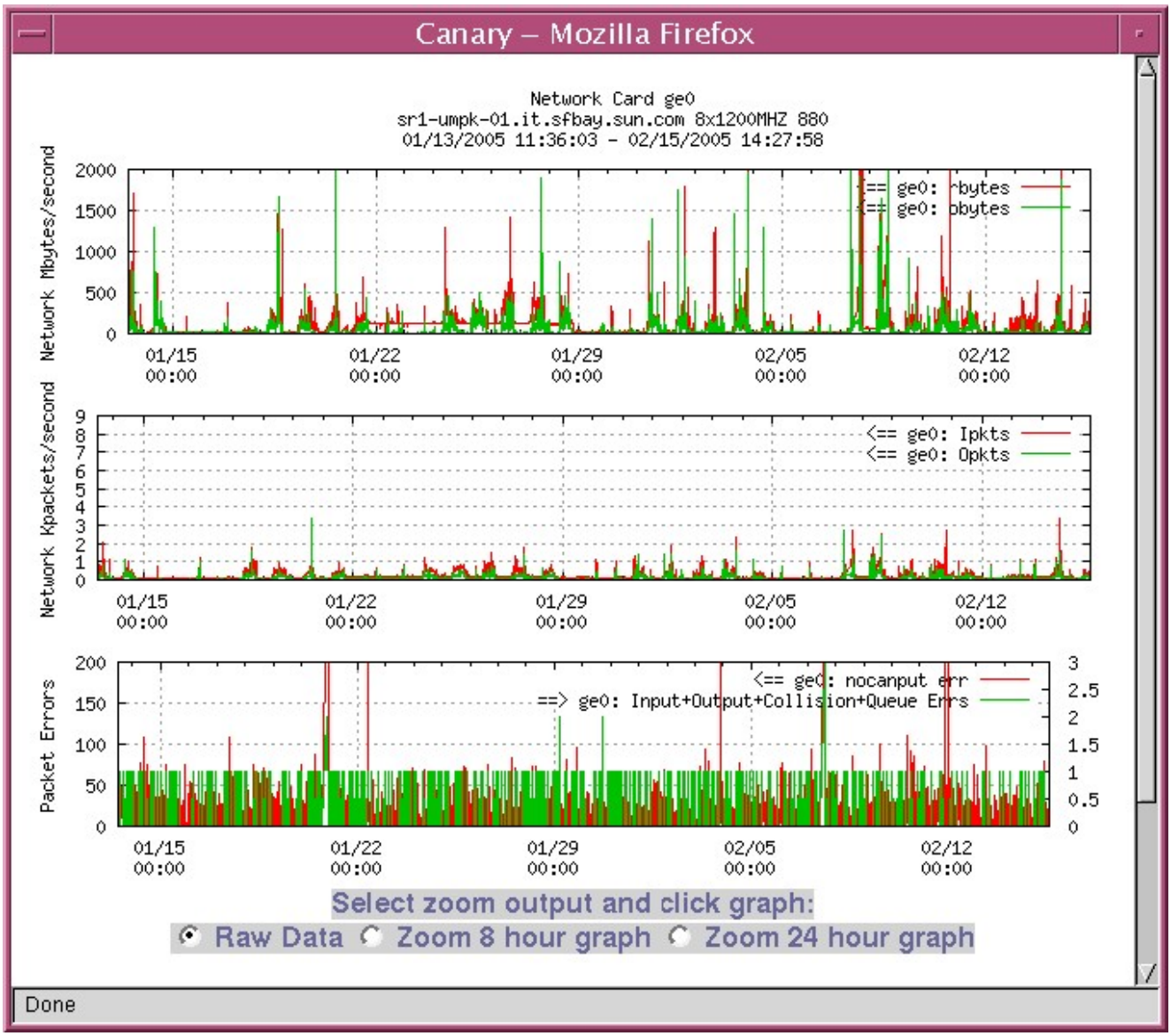
This graph shows spikes similar to the previous one.

A graph of the example Sun Ray server's third network interface is shown below:



Older versions of Sun Ray Server Software required a private network to operate Sun Ray terminals on. Newer versions of Sun Ray Server Software permit Sun Ray Desktop Appliances to coexist on a standard network with other desktop machines.

Below is a graph of a gigabit network interface on the example Sun Ray server:



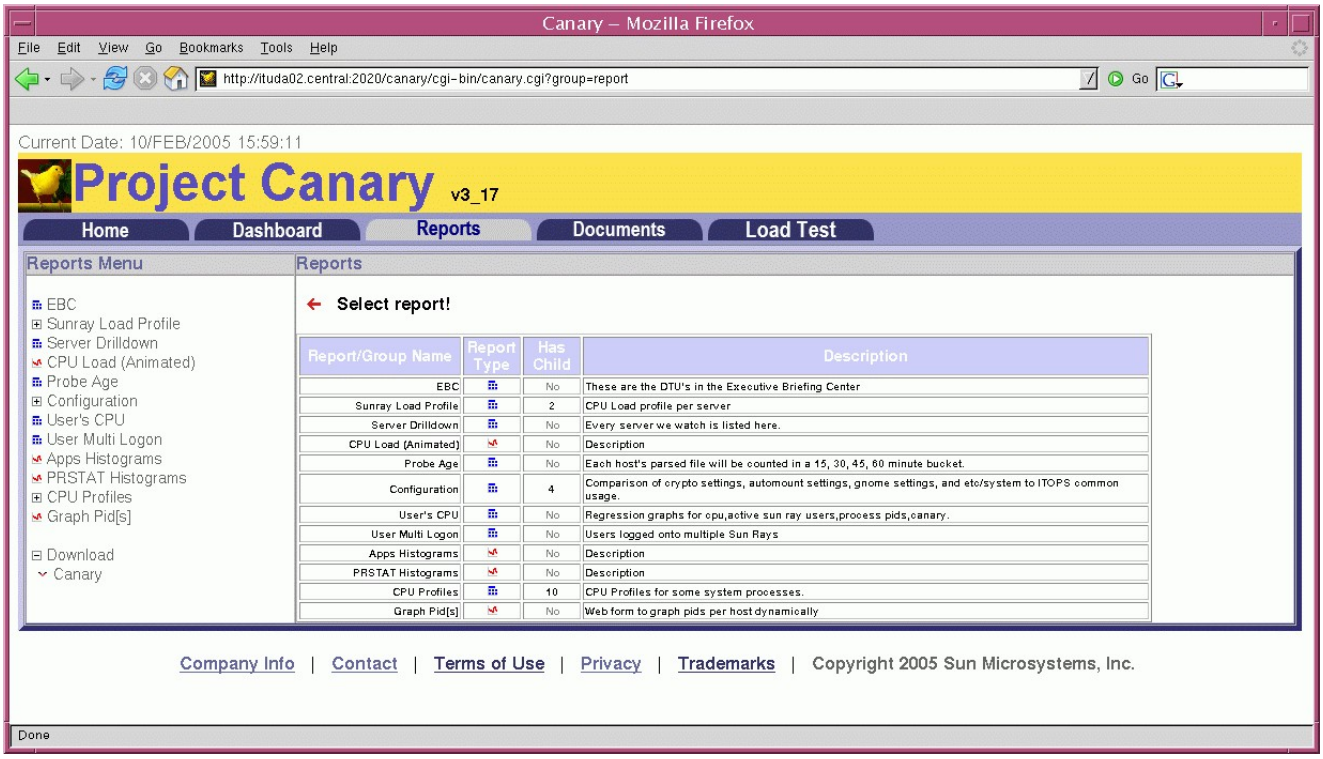
This particular interface is plugged into a backbone network. Traffic that needs to get from this particular Sun Ray server to another campus, building, or network goes over this interface. As a result, there are more packet errors and also simply more traffic than on other network interfaces.

Chapter 4: Reports Tab

4.1: Reports Tab: Root Screen

When clicking on the Reports Tab, one is taken to a screen where various reports can be generated. Some of these reports are overviews of all systems being monitored and some allow the user to drill down into specifics for either particular groups of machines or individual machines.

The screen one arrives at when clicking on the Reports Tab is shown below:



On the left-hand side there is a menu which will take one to a particular report. In the middle is a listing of the various reports along with information as to whether or not there is a "child" of that report, meaning there are more specific reports available underneath that particular one (note the [+] icon next to the reports on the left-hand side menu), what type of report it is, the name of the report, and a description of the report.

The blue icon indicates a text-based report and the red icon indicates a graph-based report.

These reports are useful when looking at a problem and work in concert with the data under the Dashboard Tab. Data under the Reports Tab should be consulted first when attempting to get a "feel" for what is going on with servers being monitored by Project Canary software. They are also useful for inclusion in presentations or when trying to look at how an entire network of servers is performing when doing capacity planning or tracking down specific problems a particular user or group of servers may be having.

The Probe Age report is particularly useful when upgrading Project Canary software as it will give the entire list of servers and when the last bits of data was collected from them.
Available reports under the Reports Tab are:

- Sun Ray Load Profile: This report gives the CPU load profile on a per-server or per-"campus" basis
- Server Drilldown: This report lists out all the different servers being monitored and gives details about those servers

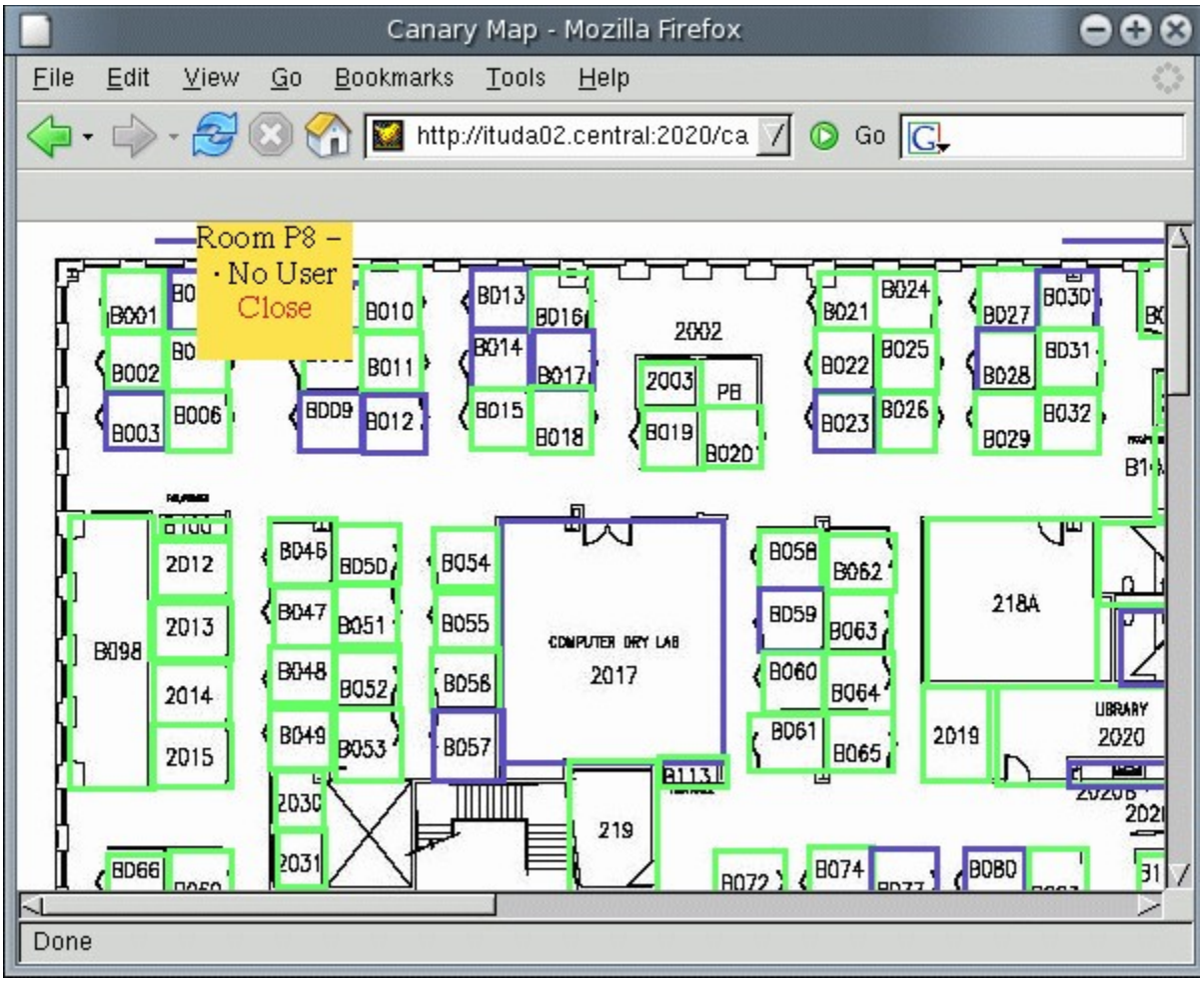
- CPU Load (Animated): This report displays animated graphs showing how users and load rise and fall over the course of a day.
- Probe Age: The Probe Age report shows when the last report was sent to the Project Canary central data collection server from a client being monitored. This is broken down into 15, 30, 45, and 60 minute time intervals.
- Configuration: Gives a comparison of crypto settings, automount settings, GNOME settings, and kernel settings and compares them to common usage.
- User's CPU: Displays regression graphs for CPU, active Sun Ray users, process Ids, and the Project Canary software itself
- User Multi Login: Shows which users are logged onto multiple Sun Ray servers. This report can also be used to show which users are simply logged into multiple servers
- Apps Histograms: Profile all applications running on a server; compute statistics, and draw a histogram
- PRSTAT Histograms: Show statistics on a per-user basis, specifically: memory, PIDs, CPU usage
- CPU Profiles: Gives the CPU profiles for some system processes, in particular, `automount`, `esd` (part of SunMC), `ncsd` (the name service cache daemon), `fsflush`, and `utdsd`
- Graph PID[s]: Displays a web form where one can graph process ID numbers per host dynamically

All of these reports are generated by Project Canary software "on the fly" meaning that they are not created and stored, thereby reducing load on the Project Canary server as the server is not spending time generating and storing reports that are not used right away.

The reports under the reports tab should be cross-checked against data under the Dashboard tab.

4.2: DTU Map

The first report available on the Reports Tab is the DTU Map. This report shows all of the Sun Ray terminals in use on Sun Ray Servers being monitored by Project Canary software. An example of a DTU Map is shown below:



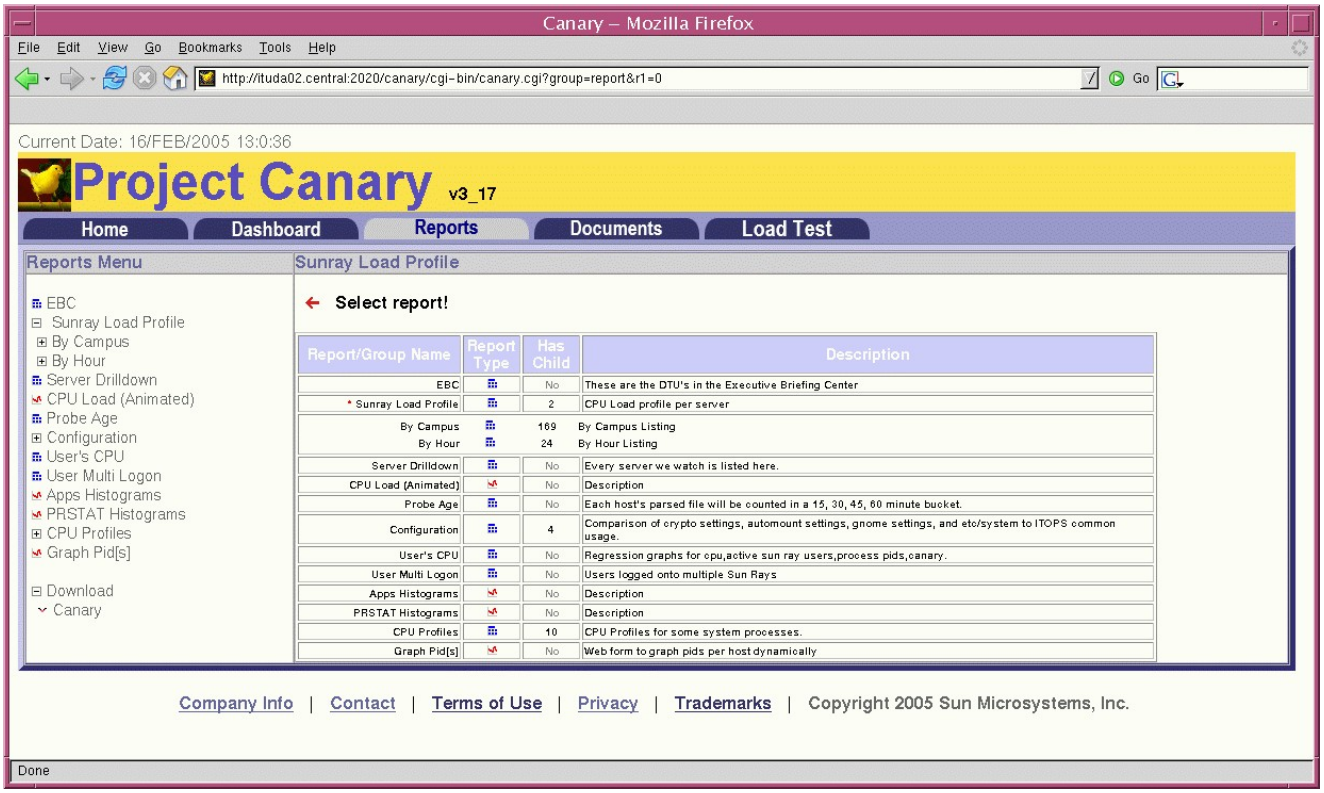
The above is an example. To create a DTU map for a building, one must first get a scanned image of a building map. The best size for such an image is an 800x600 image that will fit in the browser nicely. The software used to "map out" the rooms in the image above was purchased from www.mapedit.com for about $10. To use that software to create the needed data files to put a building map into Project Canary software, simply open the map in the tool and use the mouse to draw a box around each room needing to be mapped. The mapping software records the pixel locations of where the room is on the map image. This can also be done by hand but is cumbersome and time consuming. Save the data file when it is completed.

To get the map to display using Project Canary software, one also needs the MAC address of the Sun Ray terminal in each room. This can be had by walking up to the terminal and pressing the three "audio" buttons in the upper right-hand quadrant of the keyboard at the same time. The Sun Ray terminal will display its MAC address which can be written down and added to the datafile generated by the MapEdit software.
Once the map image and the MapEdit file are created, the make_map.sh script included with Project Canary software will produce a file with the room number, MAC address of the Sun Ray terminal inside it, description of the room and Sun Ray server, and the X/Y coordinates on the map and will generate the appropriate web pages for display.

4.3: Sun Ray Load Profile Report

The second report available on the Reports Tab is the Sun Ray Load profile. Clicking on that link on the left-

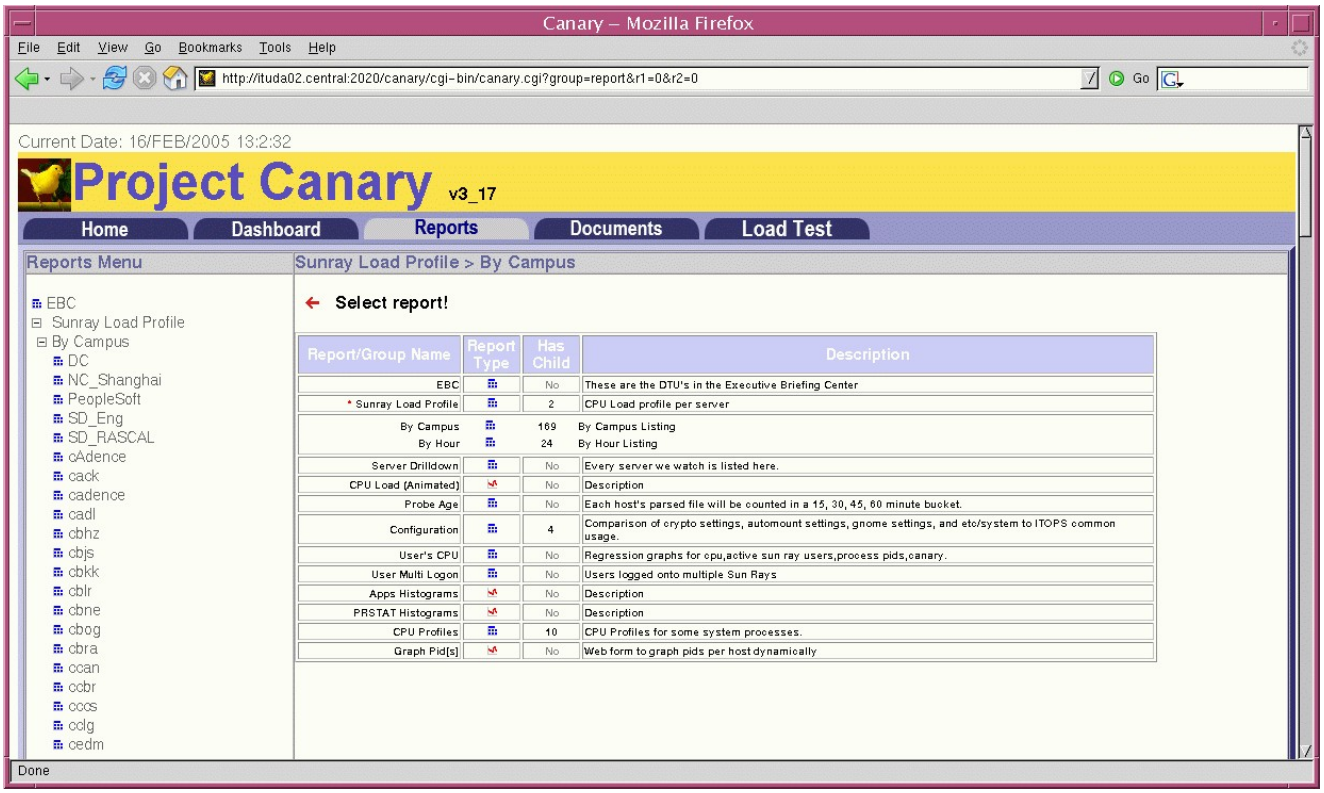
hand menu brings one to a submenu as shown below:



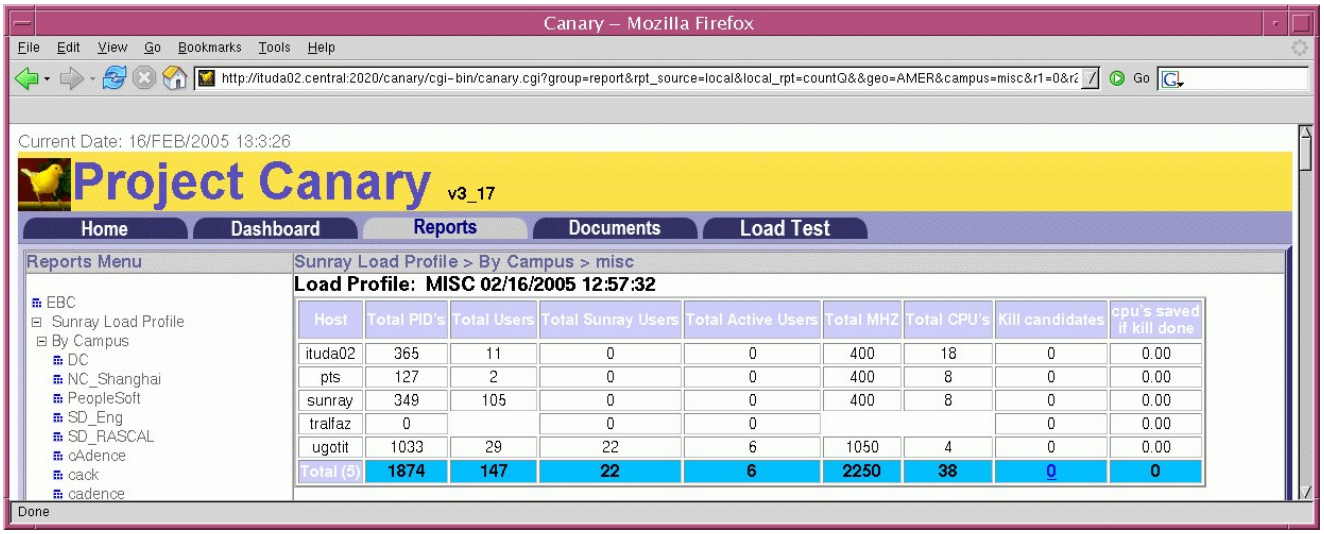
This submenu allows for two choices; running a report to retrieve the Sun Ray Load Profile Report by "Campus" or retrieving a Sun Ray Load Profile Report for all servers "By Hour."

Both reports display data as of the last ten minutes and leave out root-owned processes. The Sun Ray Load Profile report is primarily concerned with user-level processes and assumes that root-owned processes need not be displayed in this particular report. Other reports and the data shown under the Dashboard Tab display all processes running on the server being monitored.

These two reports are useful for determining which processes generate the most load so those processes can be actively monitored. Generally, on servers where users are running standard productivity tools, the processes which will float to the top and use the most CPU time are Web Browsers, Office Productivity Suites, Java, GNOME sessions, Acroread, and other tools users use commonly and frequently.
The Sun Ray Load Profile Report "By Campus" is shown below for a sample "Campus" of hosts actively monitored by Project Canary Software is shown on the next page.



From this screen, one must select a "Campus" from the left-hand menu. Once a "Campus" is selected, one comes to a screen with data specifically for that "Campus." There are two reports displayed; one is a basic load profile per server as shown below:



This view shows the server name, the total number of processes running on it, total number of users, total number of Sun Ray users, total number of active users, CPU clock speed, number of CPUs, processes which are candidates to kill, and the number of CPUs saved if runaway processes are terminated. All data is summed at the bottom of the report to give an overview as to activity in this "Campus" server grouping.

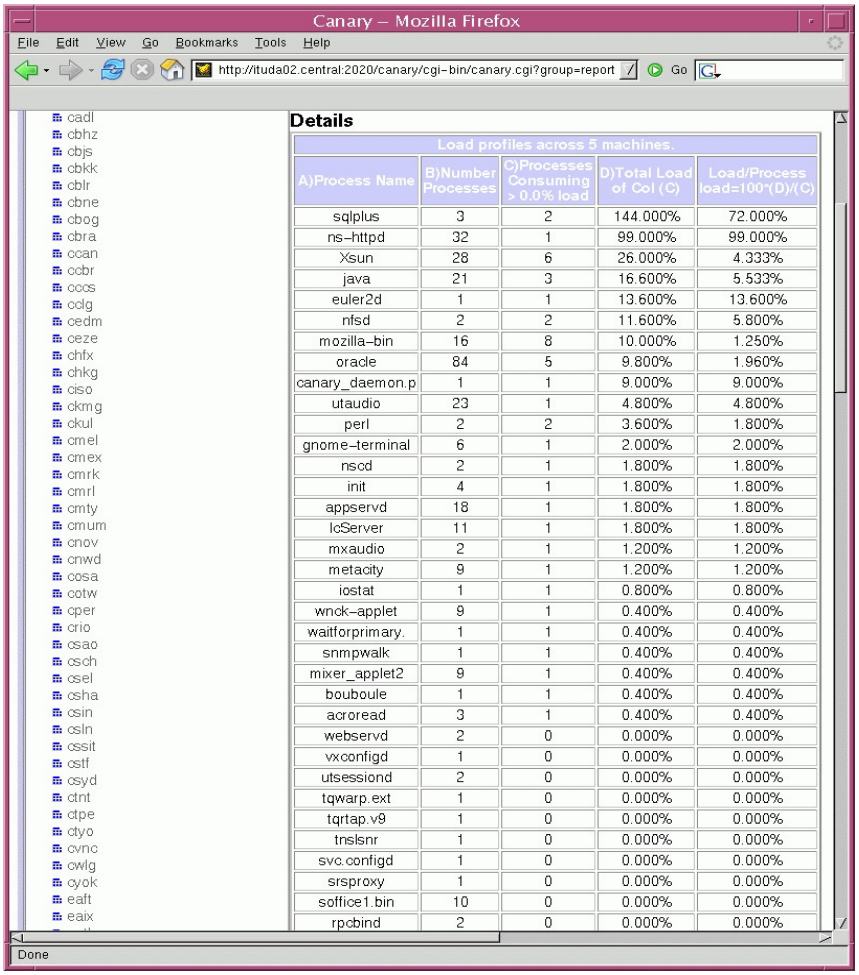
The view "by Campus" shown on the previous page in showing the CPU load and the kill candidates along with the number of CPUs saved if suspected runaway processes are terminated is one way to recover resources that can be put to use for other tasks.

The CPU utilization of the "Kill Candidates" is pulled from the `ps(1)`command. The `ps(1)` command lists the percentage of a CPU that is being used by each process. Suspected runaway processes are listed and the amount of CPU time they use is added together to give the "CPUs saved if kill done" column its data. In the screen capture shown on the previous page, there were no runaway processes that were kill candidates. However, if there were a few runaway processes that were kill candidates, they would have been shown.

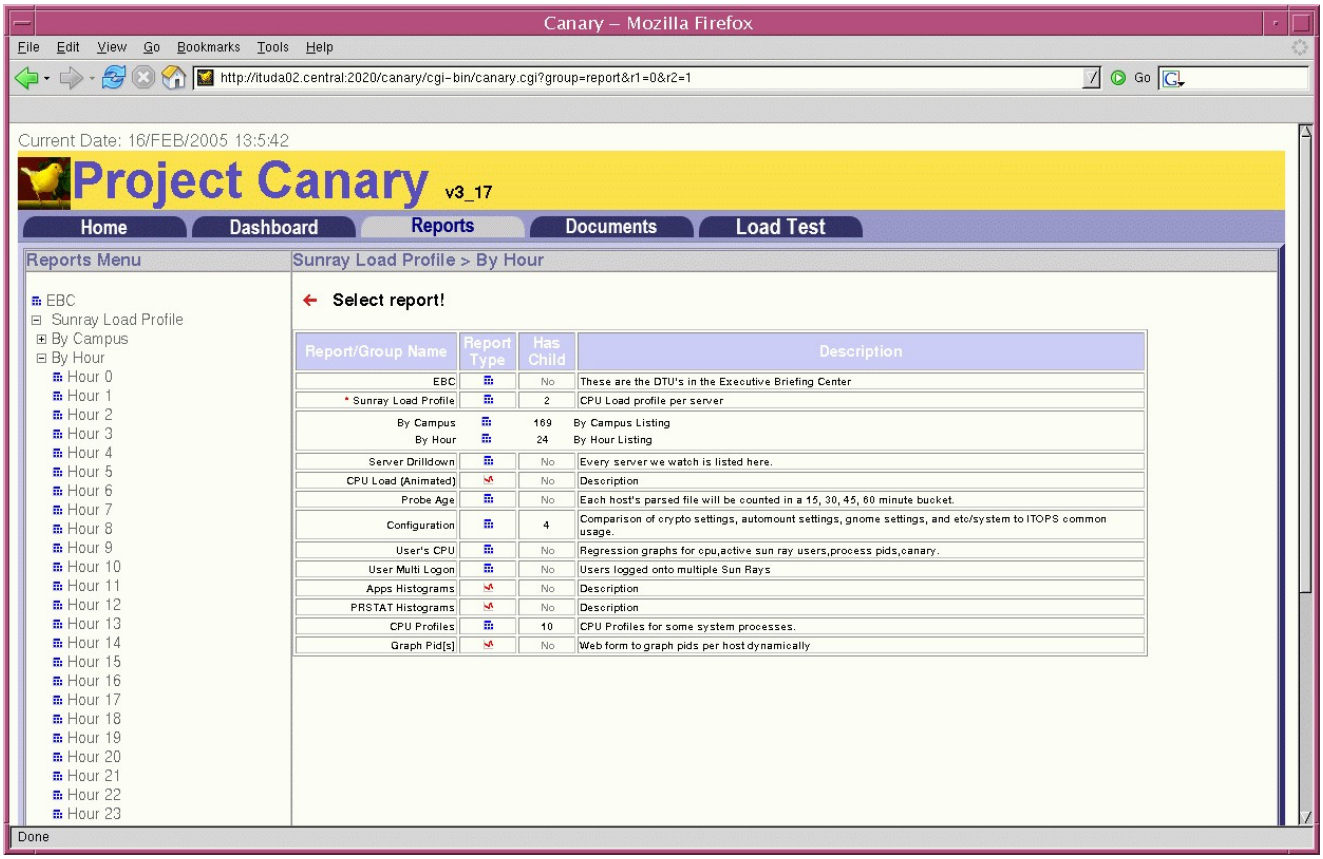
Assume that there are four "runaway" processes, one per machine in a four-machine "Campus" grouping. If each of those "runaway" processes was consuming 25% of a CPU, then terminating all four of those "runaway" processes would recover one full CPU from that server grouping.

The second table in the "Campus View" of the Sun Ray Load Profile is a listing of individual processes in that server grouping. It is ranked in descending order with the processes at the top of the table being the ones which are consuming the most CPU time.

An example is shown on the next page:



As described previously, this table simply lists processes in descending order of CPU utilization.



The Sun Ray Load Profile "By Hour" view is shown on the next page:

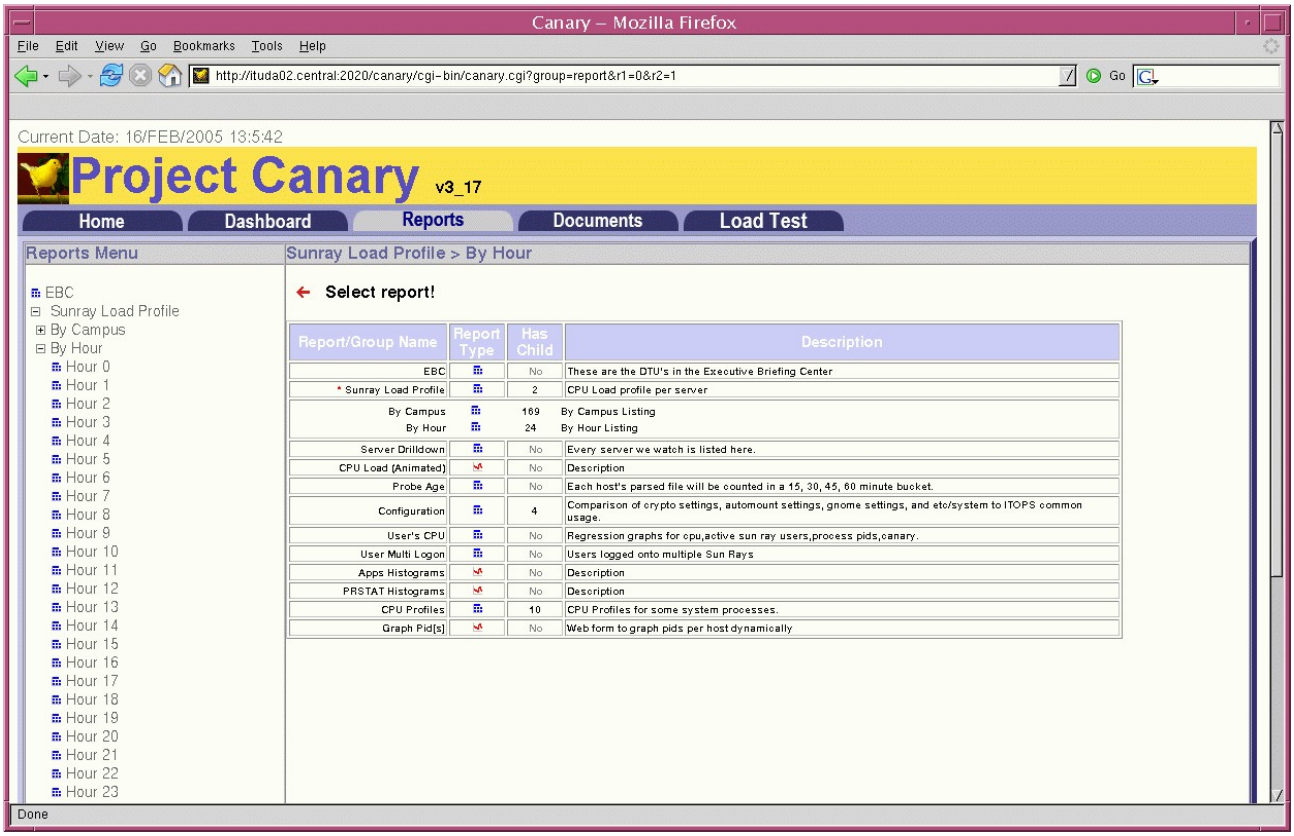
The left-hand side menu gives a listing of hours based on the time zone that the Project Canary Data Collection Server resides in. This is known as "Canary Central Time." A report can be generated giving the Load Profile for any hour of the day in military time. "Hour 0" corresponds to the "midnight to 1am" time frame; "Hour 12" corresponds to the "noon to 1pm time frame" etc.

Clicking on any hour gives a load profile for all servers Project Canary software is monitoring for that particular hour. This report may take a little while to generate and load in the browser, depending on how many servers are being monitored.

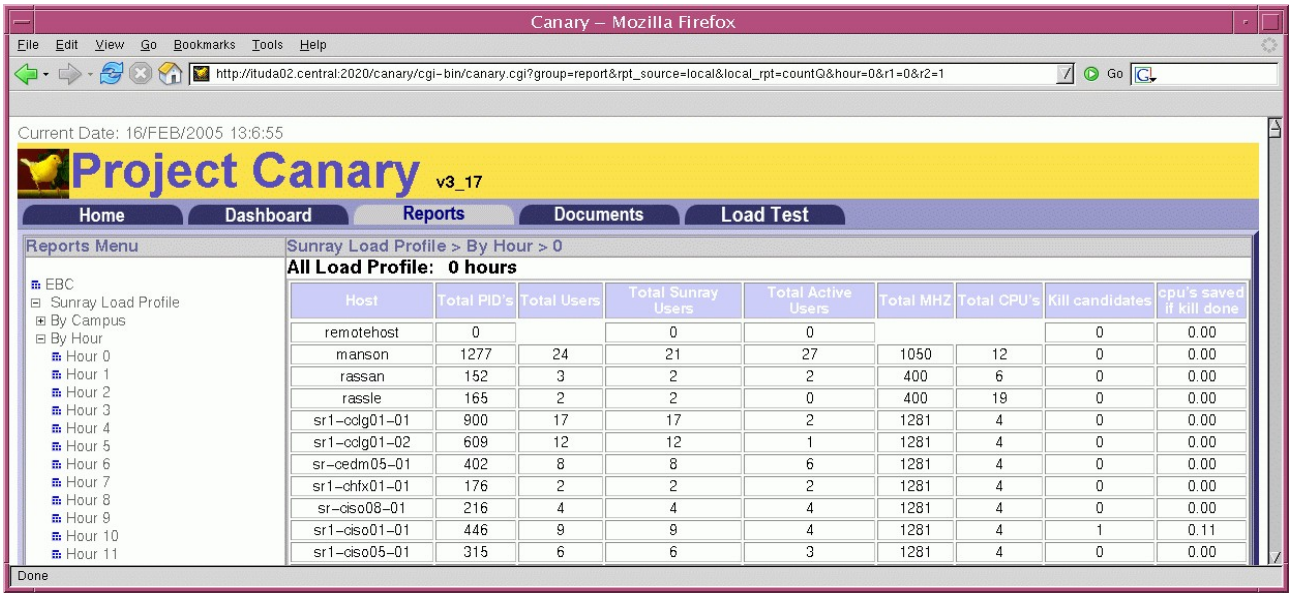
Like the "Campus" view, the Sun Ray Load Profile By Hour report is split into two different tables.

The top table lists all the servers Project Canary software is monitoring in the same manner that the "By Campus" report lists them, along with a list of totals at the bottom of the table. Likewise, in the "Details" table, the Sun Ray Load Profile "By Hour" report shows, in descending order, which processes are consuming the most CPU time.

The "top" of the Sun Ray Load Profile "By Hour" screen is shown on the next page:



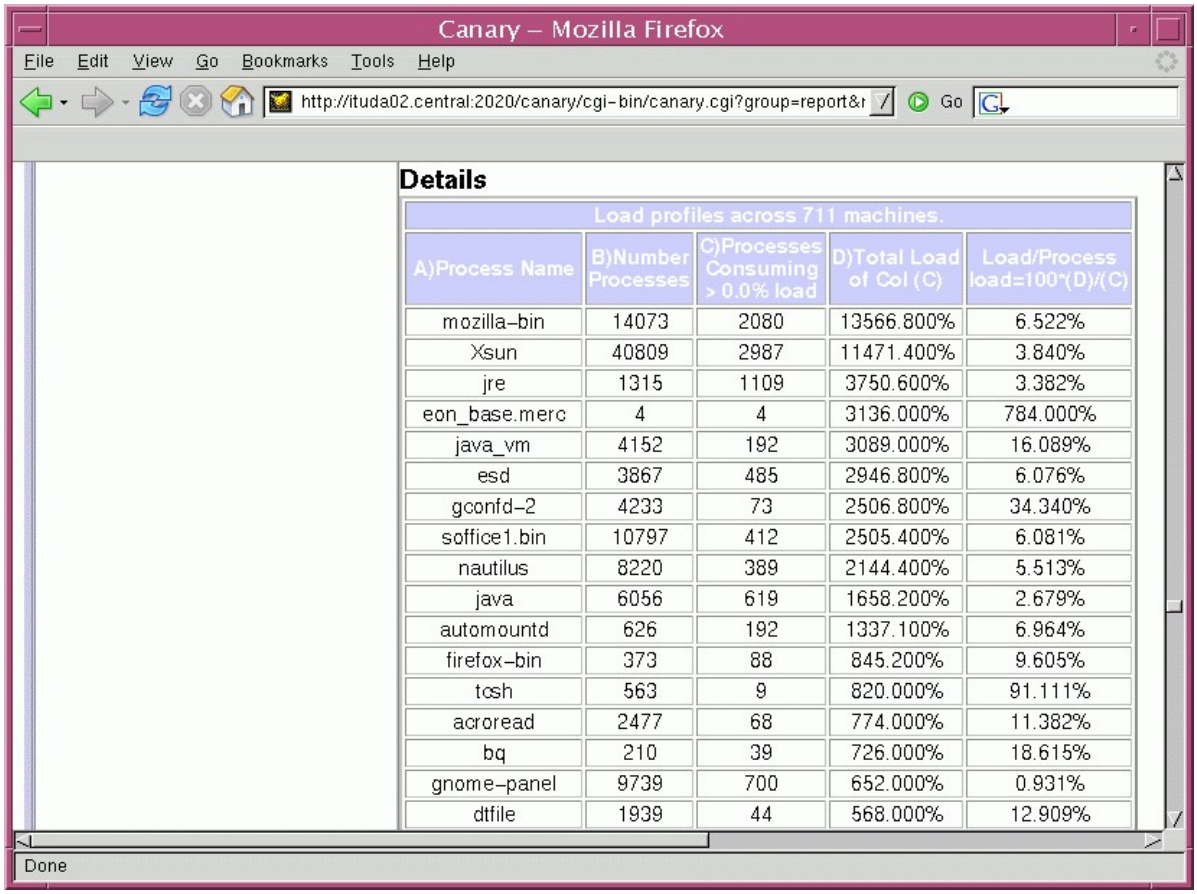
The left-hand menu gives each hour according to "Canary Central Time," or the time that the Project Canary central data server operates in. Clicking on one of the "Hour" links in the left-hand menu brings one to a report for that particular hour, as shown below:



This report is identical to the "By Campus" Sun Ray Load Profile report in data delivery, however, unlike the "By Campus" report, this report lists all servers being monitored by Project Canary software.

Note that the example on the previous page shows a "Kill Candidate" process that, if terminated, would free up .11 CPUs for other tasks.

The Details table in the Sun Ray Load Profile By Hour report is identical to the report in the Sun Ray Load Profile By Campus report described earlier, again, with the exception that it lists all servers being monitored by Project Canary software. An example of the Details table in the Sun Ray Load Profile By Hour report is shown below:



Details

Load profiles across 711 machines.

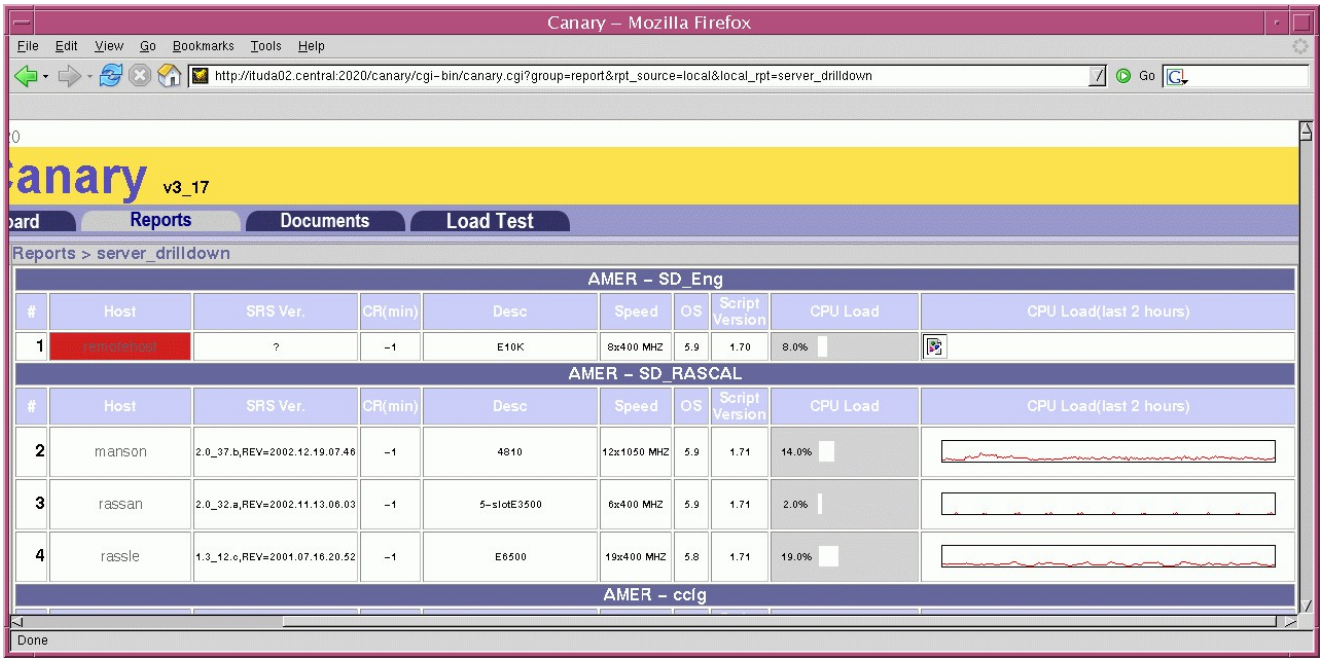
A)Process Name	B)Number Processes	C)Processes Consuming > 0.0% load	D)Total Load of Col (C)	Load/Process load=100*(D)/(C)
mozilla-bin	14073	2080	13566.800%	6.522%
Xsun	40809	2987	11471.400%	3.840%
jre	1315	1109	3750.600%	3.382%
eon_base.merc	4	4	3136.000%	784.000%
java_vm	4152	192	3089.000%	16.089%
esd	3867	485	2946.800%	6.076%
gconfd-2	4233	73	2506.800%	34.340%
soffice1.bin	10797	412	2505.400%	6.081%
nautilus	8220	389	2144.400%	5.513%
java	6056	619	1658.200%	2.679%
automountd	626	192	1337.100%	6.964%
firefox-bin	373	88	845.200%	9.605%
tcsh	563	9	820.000%	91.111%
acroread	2477	68	774.000%	11.382%
bq	210	39	726.000%	18.615%
gnome-panel	9739	700	652.000%	0.931%
dtfile	1939	44	568.000%	12.909%

This table excerpt shows only the "top" CPU consuming processes and is quite long. Note that the processes taking up the majority of CPU time are browsers, `esd`, the SunMC monitoring daemon, Java, GNOME applications, office productivity software, and other user productivity software.

Column "A" shows the name of the process. Column "B" shows the number of those processes running across all server being monitored by Project Canary software. Column "C" shows the number of active processes; this number is derived from the number of processes actually using CPU time. Processes in a sleep state are not counted as part of this number. Column "D" adds together the amount of CPU utilized by each process on each server. The percentage of CPU used shown in this column is a total from all CPUs being monitored. The last column shows the average CPU utilization per CPU that each instance of that process.

4.4: Server Drilldown Report

The Server Drilldown report shows all servers along with configuration information, current version of Project Canary client probe software running, and offers a load graph. An example of the Server Drilldown Report is shown below:



Servers whose name is listed in red have no current data associated with them, but have been left in the configuration file so that the Project Canary server still expects data to be delivered from them.

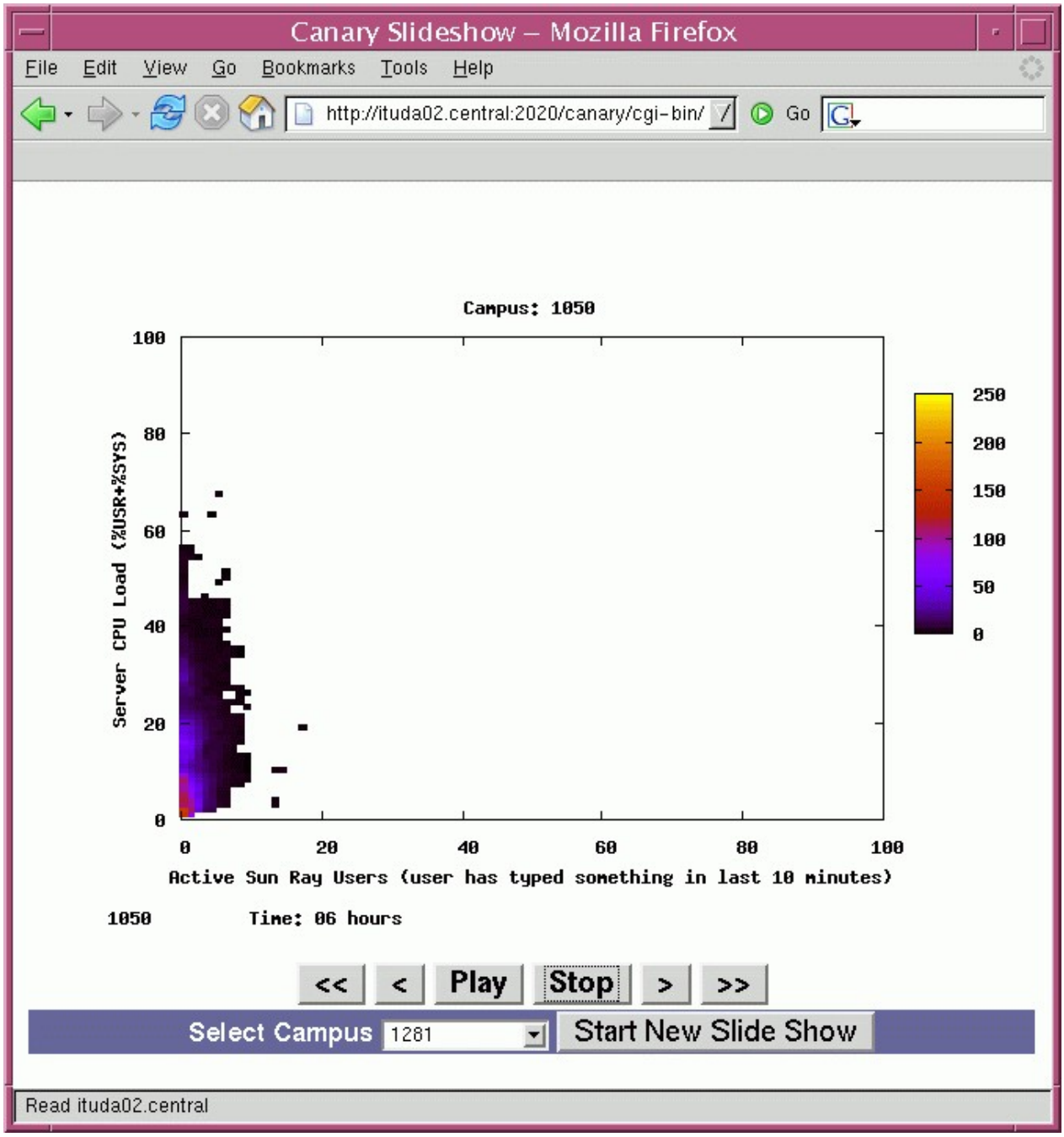
As shown above, servers are listed by "Campus" and the data offered on this report is a quick overview of all servers being monitored. This report's primary purpose is to show what version of the Project Canary client probe software is running on each client being monitored, but can also be used to get a quick snapshot of the network being monitored.

The columns displayed in this report offer the following data:

Column 1: Hostname of the machine being monitored by Project Canary
Column 2: The version of SRS (Sun Remote Services) software installed, if any.
Column 3: The frequency at which the cleanRunaways script runs to collect data about and identify potential runaway processes.
Note: A "-1" in this column means that the cleanRunaways script is not active
Column 4: A description of the server's hardware or model number
Column 5: The number and speed of the CPUs in the server being monitored
Column 6: The version of the Solaris Operating System or Linux Operating System running on the server
Column 7: The version of Project Canary client probe software running on the server
Column 8: The current CPU load on the server being monitored
Column 9: A graph of the CPU load of the server being monitored over the last two hours

4.5: CPU Load Report (animated)

The CPU Load (animated) report shows the server load vs the number of active users on the server. This is used to predict future load and to do capacity and sizing planning for servers. An example of one frame from the animated CPU load graph is shown below:



The CPU Load (animated) report shows an animated histogram plotting the number of active users vs the server load over the course of a 24-hour period. The data used to plot these graphs is captured six times per

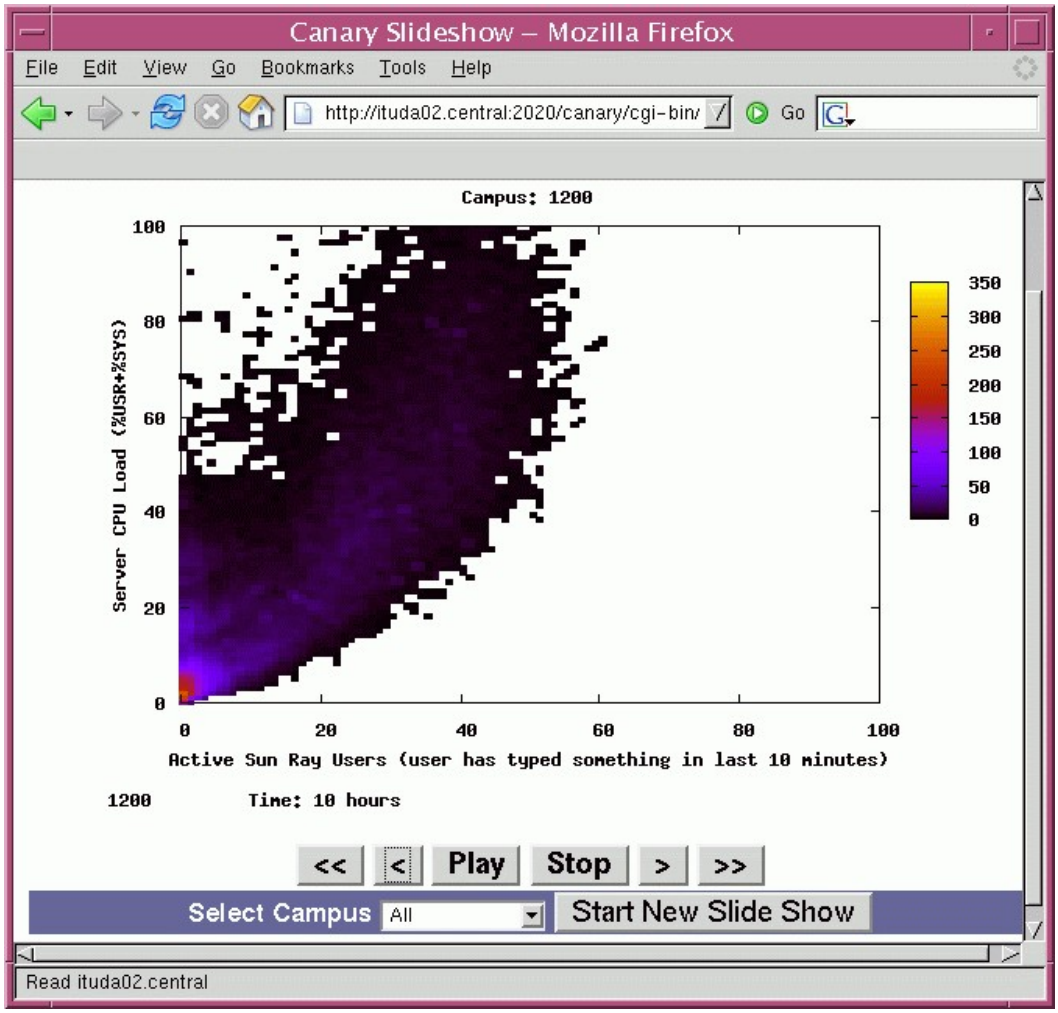
hour by Project Canary client probe software.

The X axis of the graph is the number of active users and the Y axis is the server load. The Z axis is a color denoted on the right side of the graph with the number of data samples used to plot that particular point.

When reading this graph, it is important to note that the horizontal axis signifies the number of people logging in, the vertical axis, the CPU load on the server being graphed, and "hot spots," meaning the section of the graph that is not black, but, rather colored in bright colors, is where most of the data points lie.

As users log in to the server and begin performing their daily tasks, note that the graph will rise to the Northeast as the workday starts and goes on and then collapses at night. This is expected behavior. If the graph simply rises and does not fall, this shows that the server has either a number of runaway processes that need to be investigated and terminated or the server is running a number of extremely long CPU intensive jobs.

See the image below for an example of a server with a "flag pattern" plotted rising to the Northeast quadrant of the graph:



A vertical spread in the graph will denote that a number of runaway processes are on the server and the

systems administrator is advised to check the Runaways graph under the Dashboard tab to locate, evaluate, and terminate these processes.

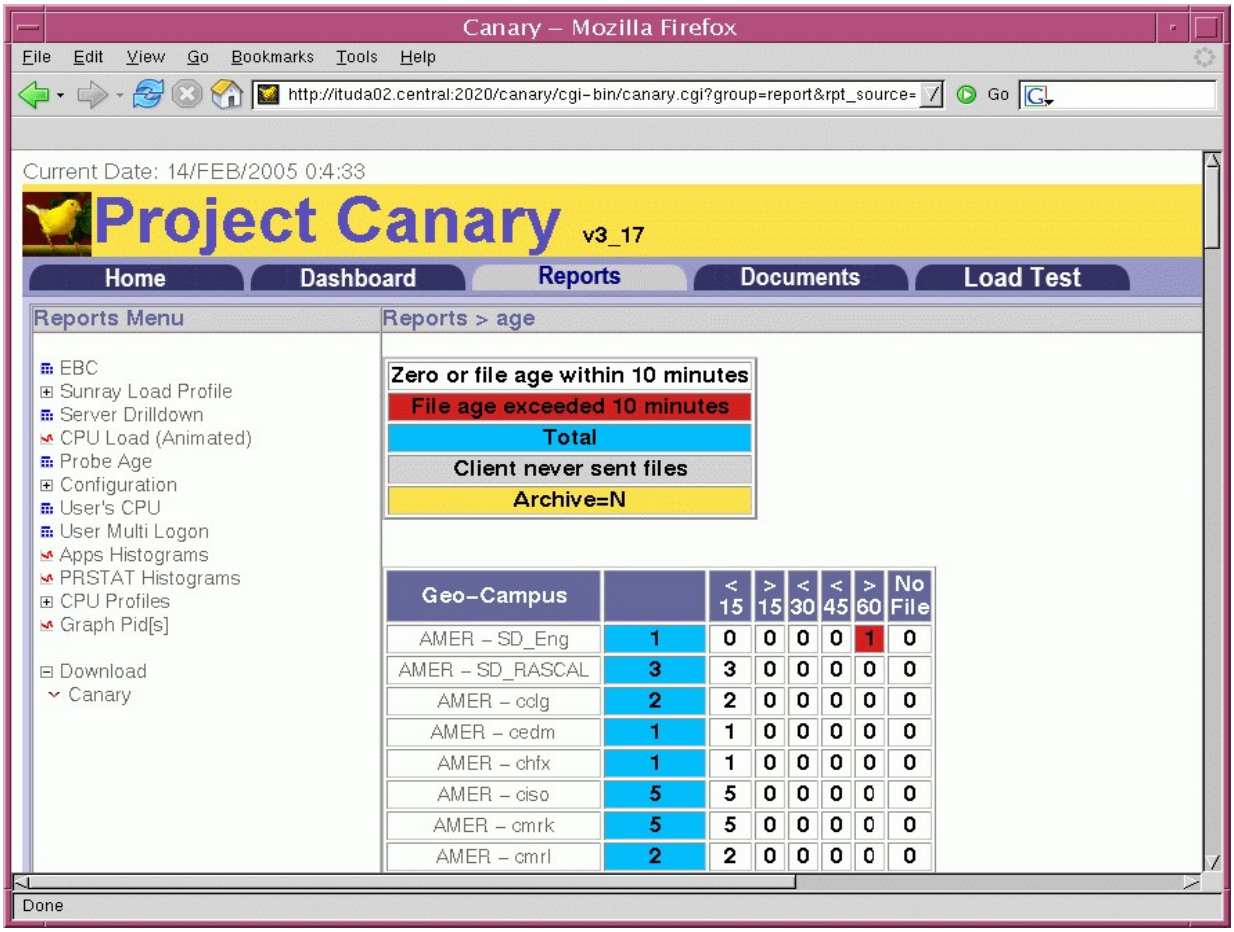
A horizontal spread in the animated graph shows that additional users are logging into the server. If the graph shows an extreme tilt toward the horizontal, it simply shows that users are selecting that particular server to log into in order to do their work. In a Sun Ray environment, servers should be balanced so that no single server receives the bulk of users. If this is the case, then the Sun Ray server software needs to be adjusted to balance users more equitably among the pool of servers available. Switching users from one server to another manually may be worth considering if one server is receiving the bulk of logins.

The CPU Load (animated) report is a predictive report. Systems Administrators can use this report to predict at which point, given an average workload, a server will "max out," or reach 100% utilization. This is useful for proactive analysis and capacity planning. This report can help Systems Administrators and IT managers determine approximately when additional servers will be required in order to spread load evenly among a pool of servers or to set thresholds at which additional user logins become inadvisable.

4.6: Probe Age Report

The Probe Age report, shown below, is useful mostly for detecting dead Canaries on client systems being

monitored, as shown below:

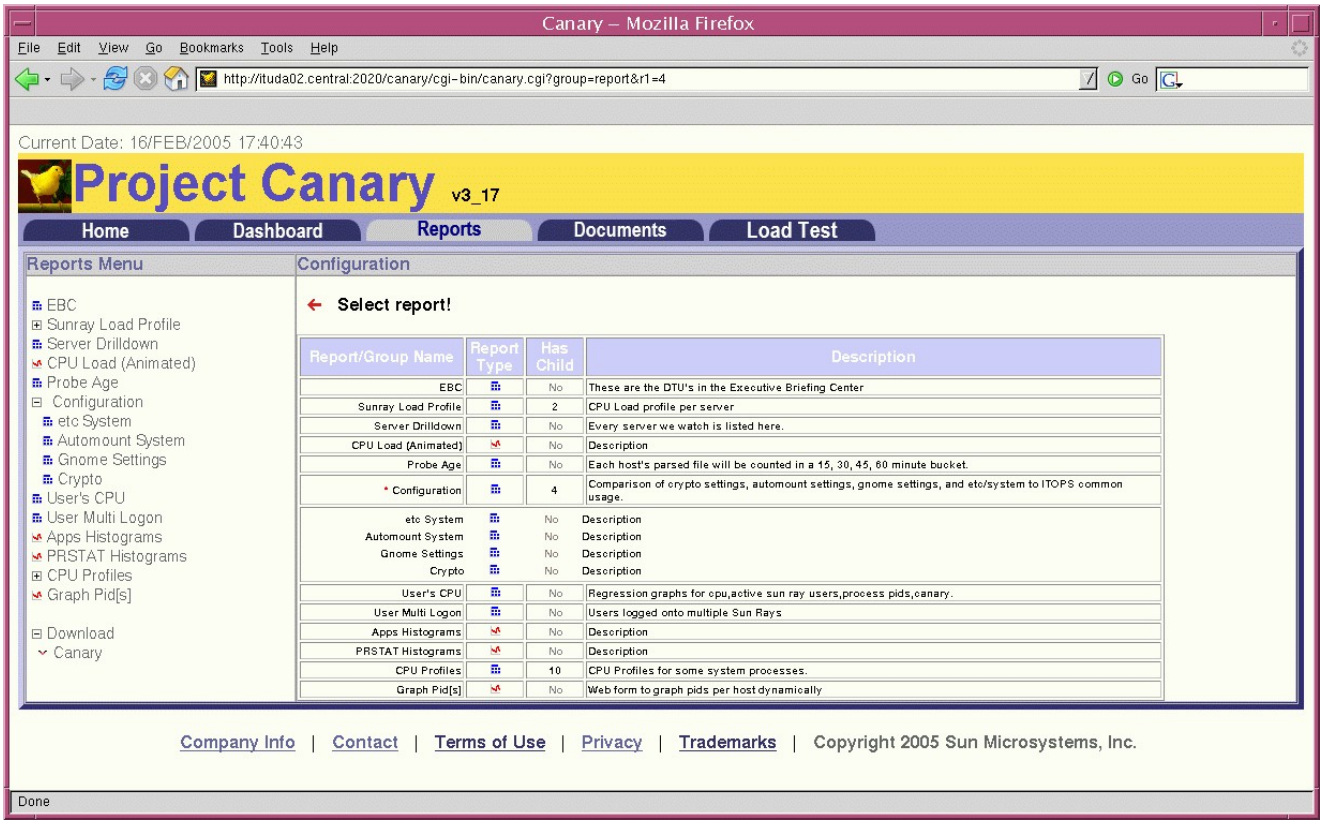


This report lists all "Campuses" being monitored by Project Canary software and simply shows when the last report was received by the central monitoring host from clients being monitored. Of primary concern are campuses that have servers listed in red. Clicking on the "Campus" listed in the left-most column of the report takes one to the Dashboard view for that particular campus.

At the bottom of this report are summaries of the number of clients configured in the server's `server_to_geo.cfg` file as being monitored, the number of servers actively being monitored, as well as a sum of the rightmost columns at the bottom.

4.7: Configuration Report

The Configuration report shows how servers being monitored by Project Canary software compare against standards coded into the Project Canary software. An example of this screen is shown below:



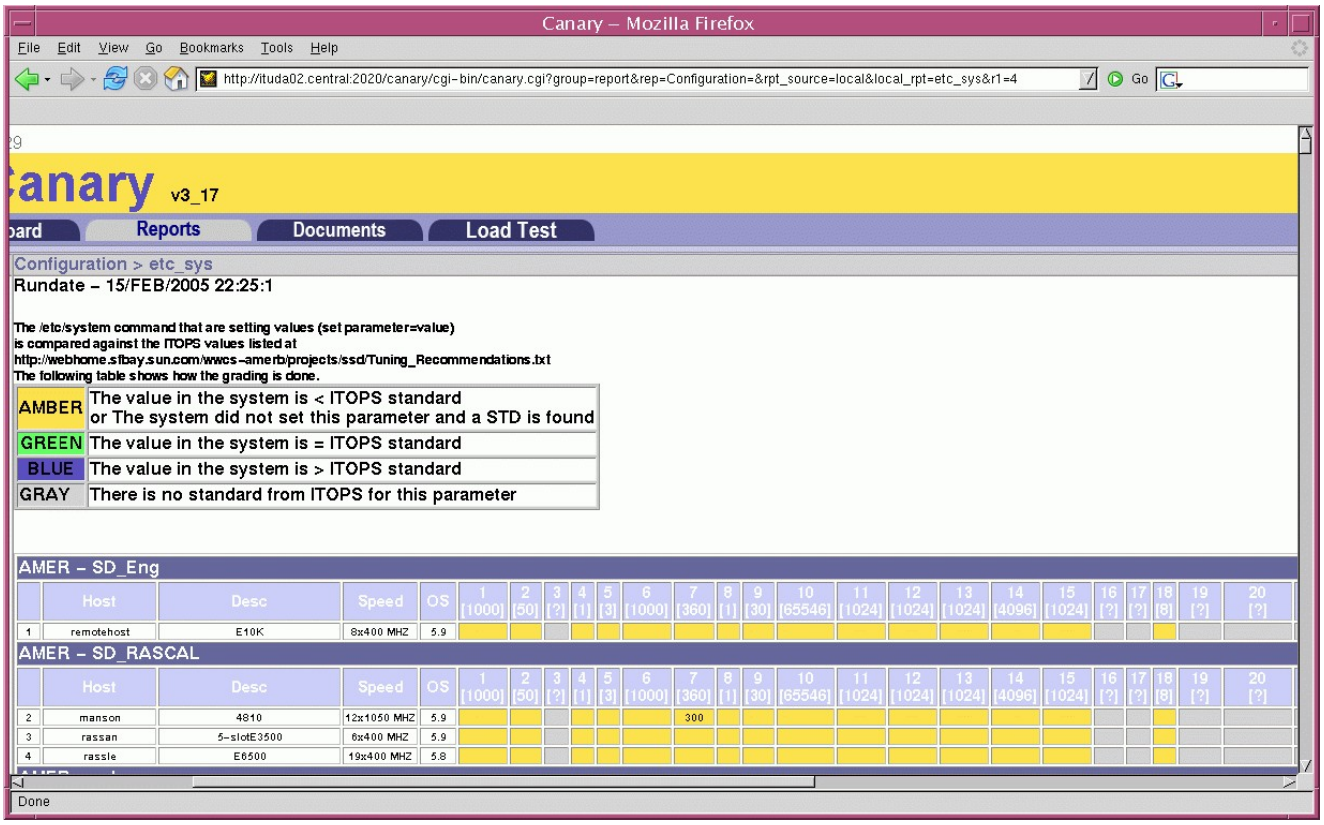
There are four reports on various configuration matters Project Canary software monitors for and collects data about which can be brought up from the Configuration Report page.

- etc System: This report examines the kernel parameters on the server Project Canary software is reporting on found in the server's `/etc/system` file and compares it against Sun IT Operations Standards.
- Automount System: This report primarily looks for the "nobrowse" option set in the server's `/etc/auto_master` on servers for the `/home` mount point.
- GNOME Settings: This report compares a Sun Ray server's GNOME settings against standards coded into the Project Canary software. In particular, it checks the default "background image" size against the standard.
- Crypto: This report checks to make sure that the appropriate Crypto settings for Sun Ray software are correctly set in both the uplink and downlink directions.

These reports are useful when planning and executing a Standards Audit to ensure compliance with IT standards. The settings checked for are coded within the Project Canary server software. Management can quickly scan these reports to determine which Sun Ray servers are running "according to standard" and which are not. The next page(s) show examples of the reports available from this submenu:

4.7.1: etc System Report

Following is a screen capture of the etc System report:



This report compares kernel values set in the Solaris `/etc/system` file to those required in Sun IT Operations standards.

As shown above, servers can be quickly scanned for compliance issues with Sun IT standards simply by looking at the colors displayed in the various fields of the report.

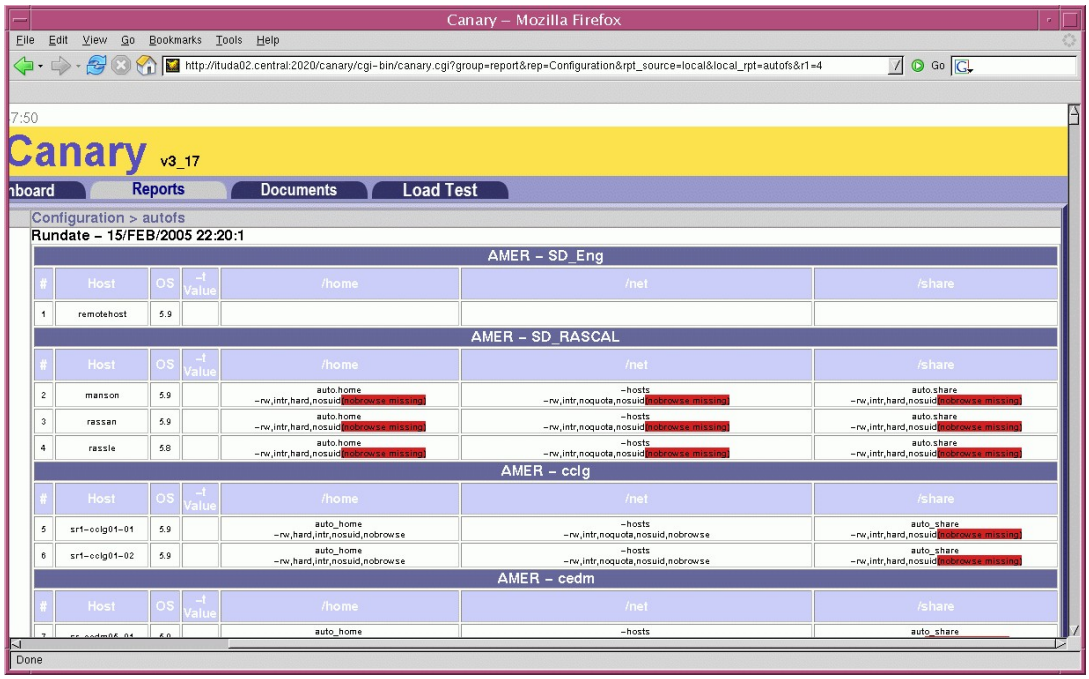
To be considered "in 100% compliance" a server should have all fields either green or gray. Servers that are out of compliance should have an exemption on file with IT management.

Currently, Project Canary software implements Sun IT Operations standards, however, this is configurable by modifying the Project Canary scripts. Examine the source code to see how these standards are set in Project Canary software. Modification of the scripts to suit an individual shop's requirements is a task for the local IT management and Systems Administrators to perform.

Not all IT shops will wish to adhere to Sun's internal standards for server compliance. This screen can be extended to whatever needs any particular shop has. The rest of the reports available under the Configuration submenu are described in detail on the next page:

4.7.2: Automount System Report

The Automount System report from the Configuration menu is shown below:



As described earlier in this document, the Automount System report compares the settings in `/etc/auto_master` on the server to those required by Sun IT Operations standards.

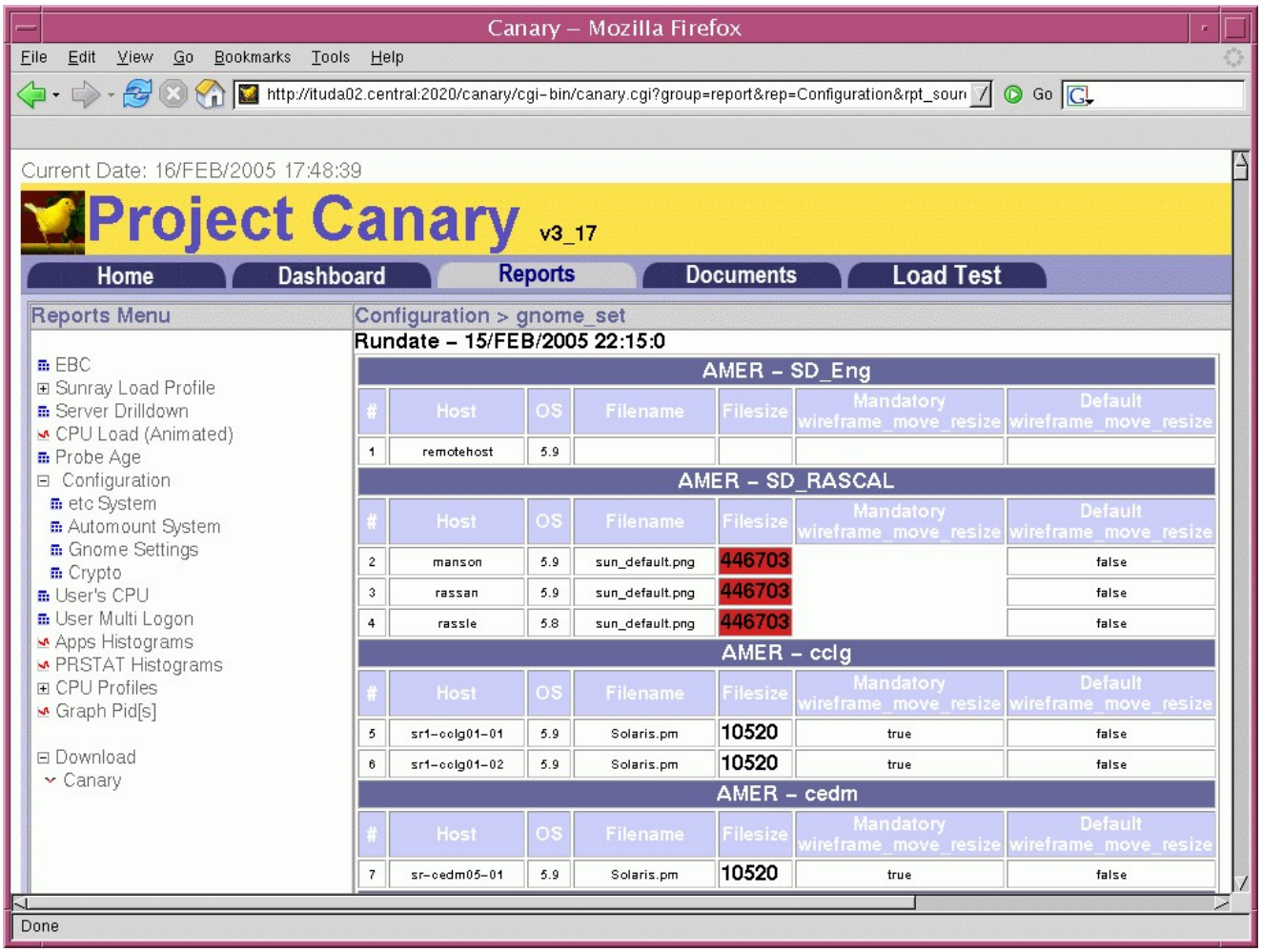
Servers whose settings are shown in red on this page are deemed to be out of compliance with Sun IT Operations standards. This report serves primarily as a standards compliance auditing mechanism so that management can quickly bring up a web page to determine which servers are complying with required IT standards and which are not.

In particular, this report looks to see that the `-nobrowse` option is set on the `/home` automount point.

As stated under the etc System report, not all IT shops will wish to comply to Sun IT Operations' standards. If an IT shop wishes to enforce their own standards, those standards will need to be edited into the Project Canary code.

4.7.3: GNOME Settings Report

The GNOME Settings report from the Configuration report menu lists all servers monitored by Project Canary software and produces a report showing which servers comply with Sun IT Operations' standards and which do not. An example of this report is shown below:



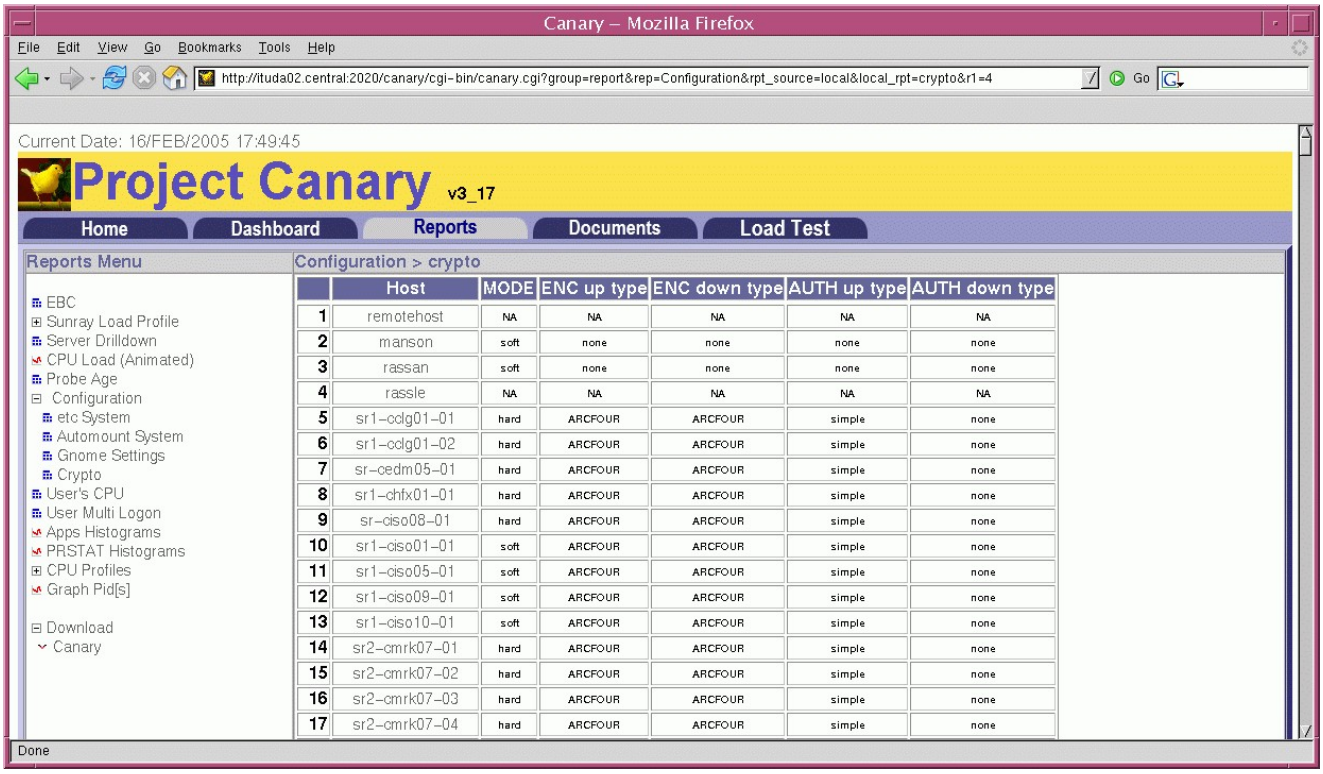
This report is primarily concerned with the background image loaded onto Sun Ray desktops via the GNOME desktop environment. Servers that have a background image that is out of compliance with Sun IT Operations' standards have the part of their GNOME configuration which is out of compliance labeled in red, with an example of this shown in the screen capture above.

As stated under the etc System report, not all IT shops will wish to comply to Sun IT Operations' standards. If an IT shop wishes to enforce their own standards, those standards will need to be edited into the Project Canary code.

4.7.4: Crypto Settings Report

The Crypto Settings report concerns itself with ensuring that the uplink and downlink crypto settings on a Sun Ray server are set appropriately.

An example of this report is shown below:



As with all reports available from the Configuration report menu, this report is a standards compliance report so management can quickly see if servers are following Sun IT Operations' standards or not. Servers that have crypto settings that are out of compliance with Sun IT Operations' standards will be shown in red.

As stated under the etc System report, not all IT shops will wish to comply to Sun IT Operations' standards. If an IT shop wishes to enforce their own standards, those standards will need to be edited into the Project Canary code.

4.8: User's CPU Report

The User's CPU report displays four graphs which statistically analyze the CPU load users place on a Sun Ray server. This report is an attempt to show the following graphically:

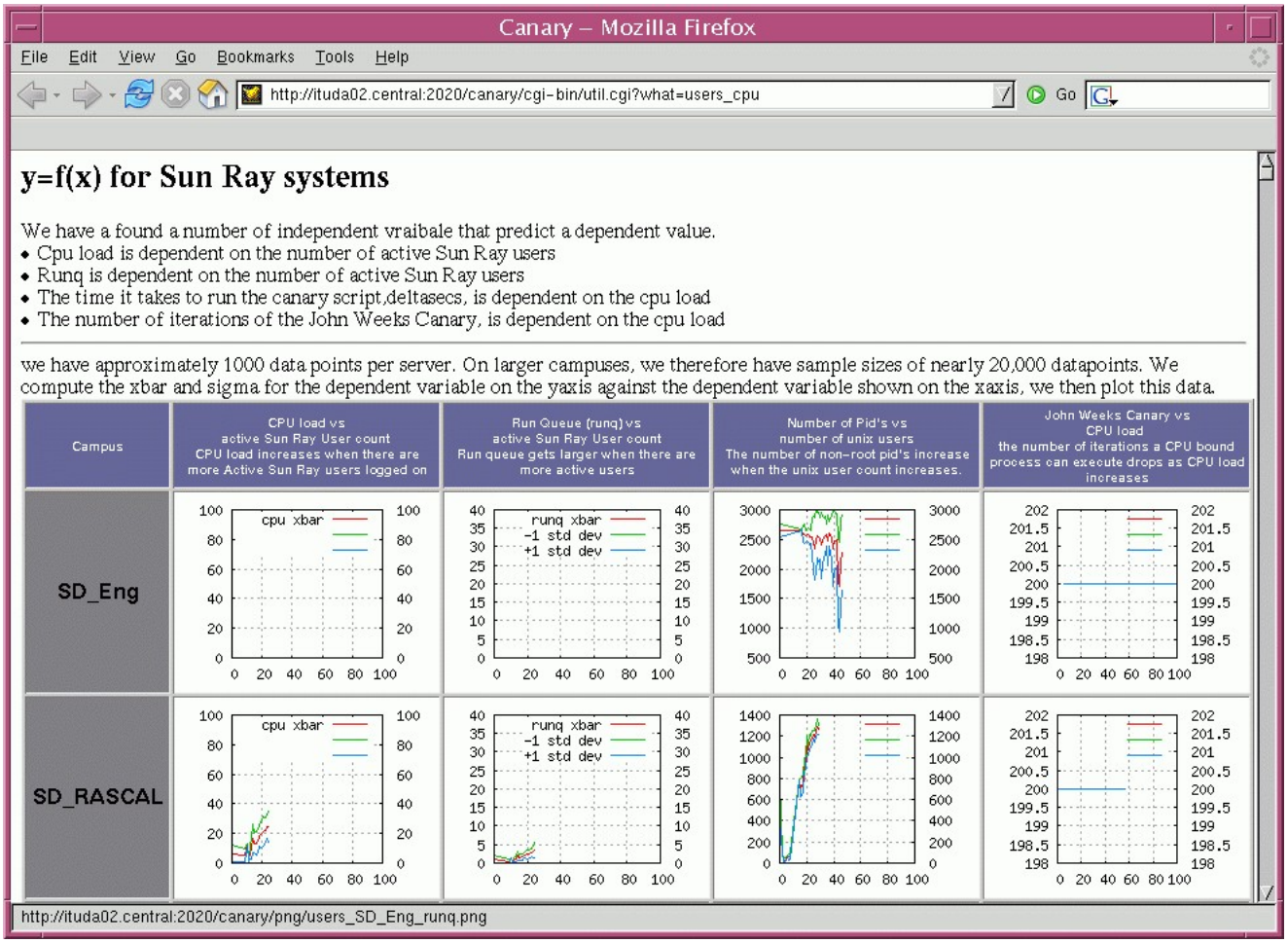
First graph: CPU Load vs Active Sun Ray User Count
Shows that CPU load increases when there are more Active Sun Ray users logged in
Second Graph: CPU Run Queue vs Active Sun Ray User Count
Shows that the Run Queue gets larger when there are more active users
Third Graph: Number of Process Ids vs the number of Unix users on a Sun Ray server

Shows that the number of processes on a system increases with additional users

Fourth Graph: CPU Bound Canary vs CPU Load

Shows that the number of iterations a CPU bound process can execute drops as CPU load increases

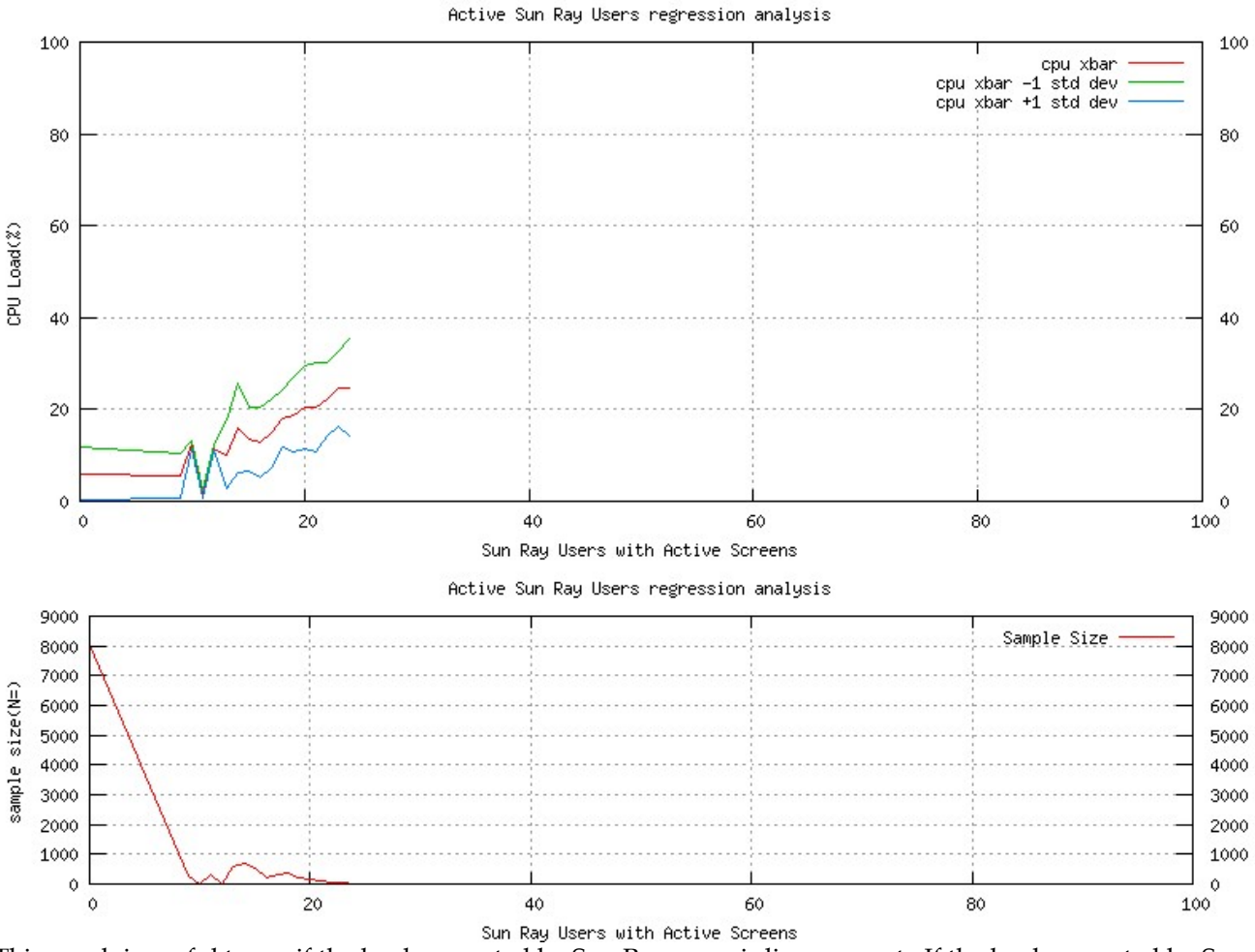
An example of this report is shown below:



Clicking on a particular graph will bring it up in a separate window.

4.8.1: CPU Load vs Active Sun Ray User Count

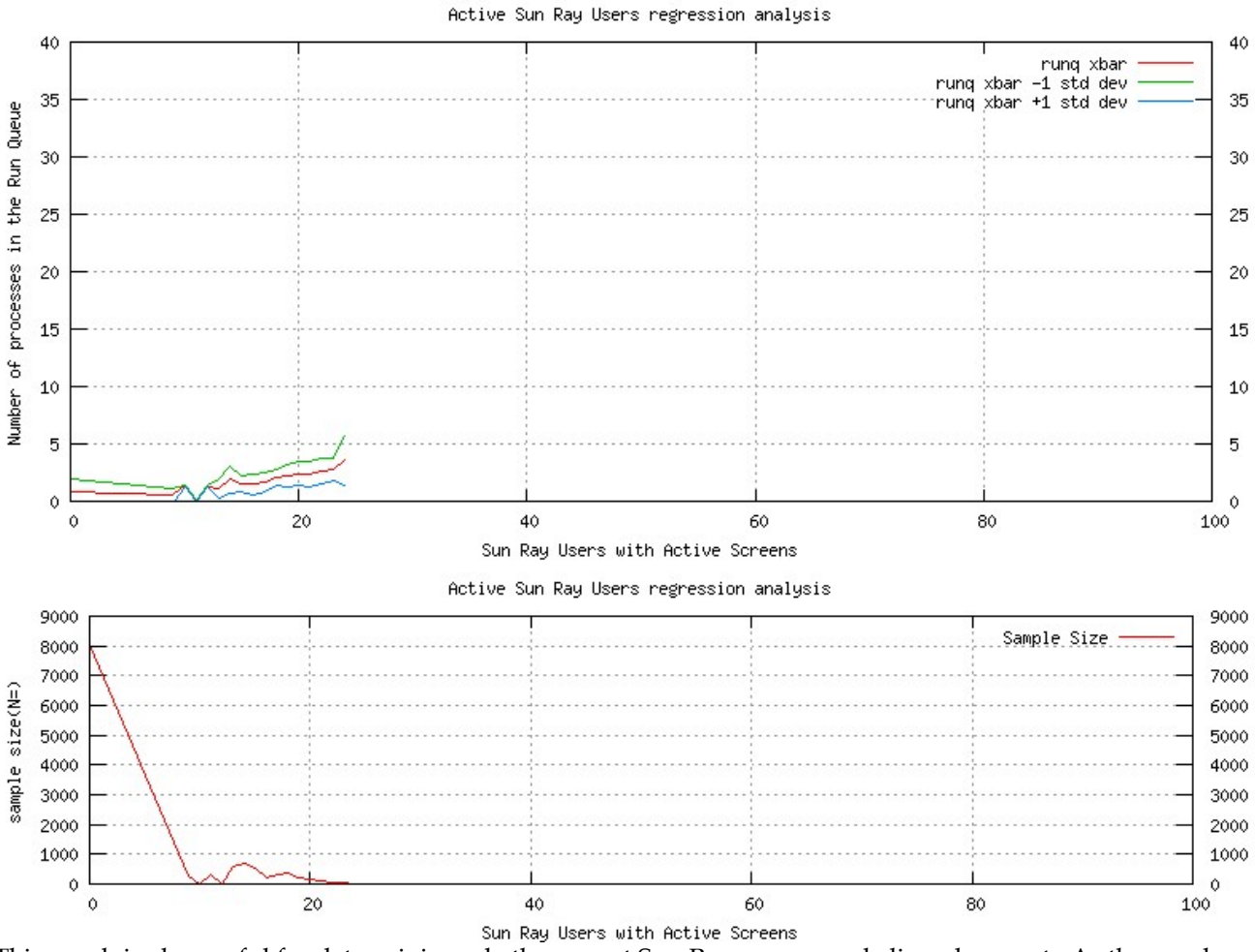
The First graph from this report page, the CPU Load vs Active Sun Ray User Count is shown below:



This graph is useful to see if the load generated by Sun Ray users is linear or not. If the load generated by Sun Ray users is not linear, then user activities should be investigated.

4.8.2: Run Queue vs Active Sun Ray User Count

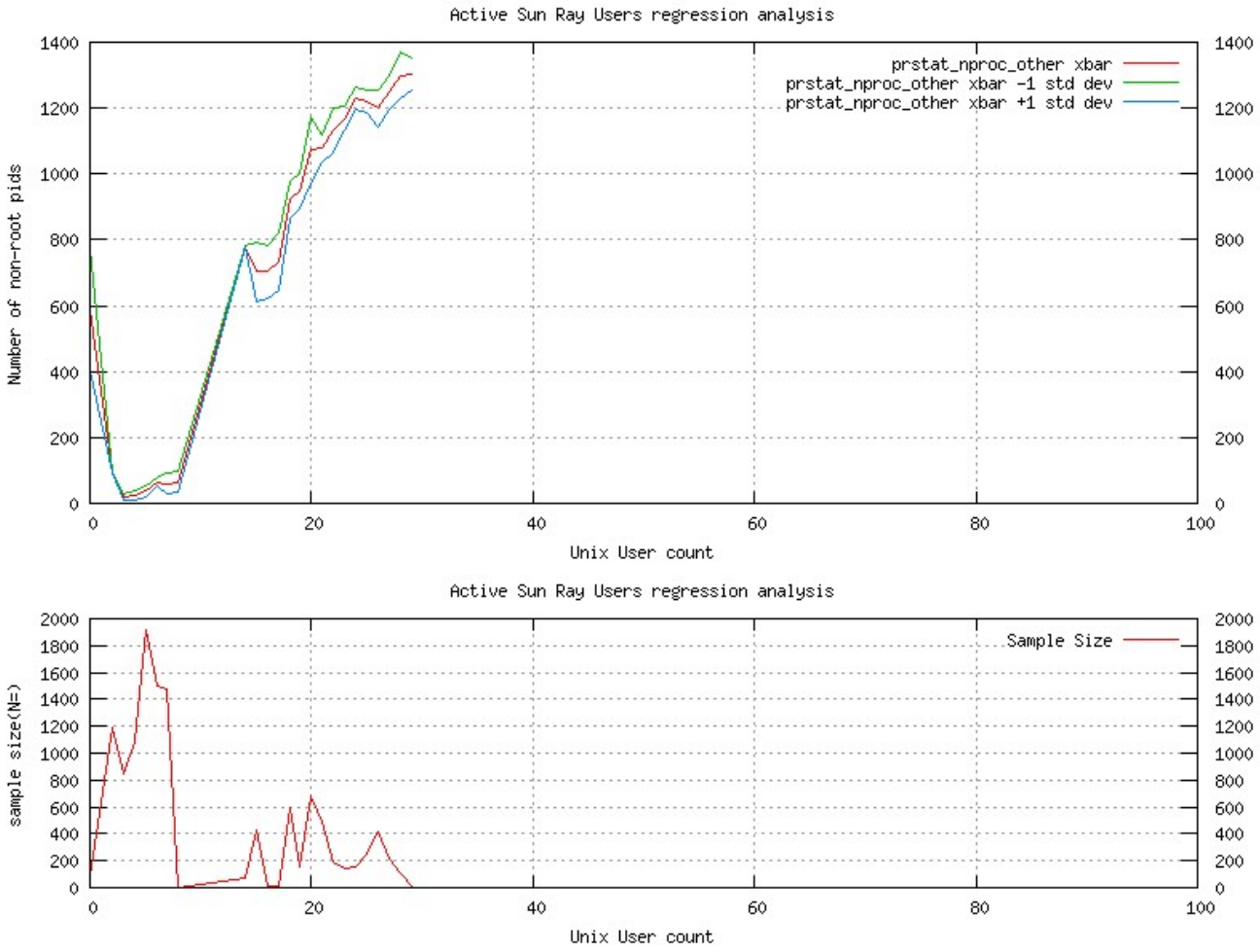
The second graph from the Users CPU report page, the Run Queue vs active Sun Ray user count is shown below:



This graph is also useful for determining whether or not Sun Ray servers scale linearly or not. As the number of users on a Sun Ray server increase, the CPU Run Queue should increase at a reasonable rate. If there are spikes in this chart, then user activities should be investigated.

4.8.3: Number of Process Ids vs Number of Unix Users

The third graph from the User's CPU report page, the Number of PIDs vs Number of Unix Users report, is shown below:

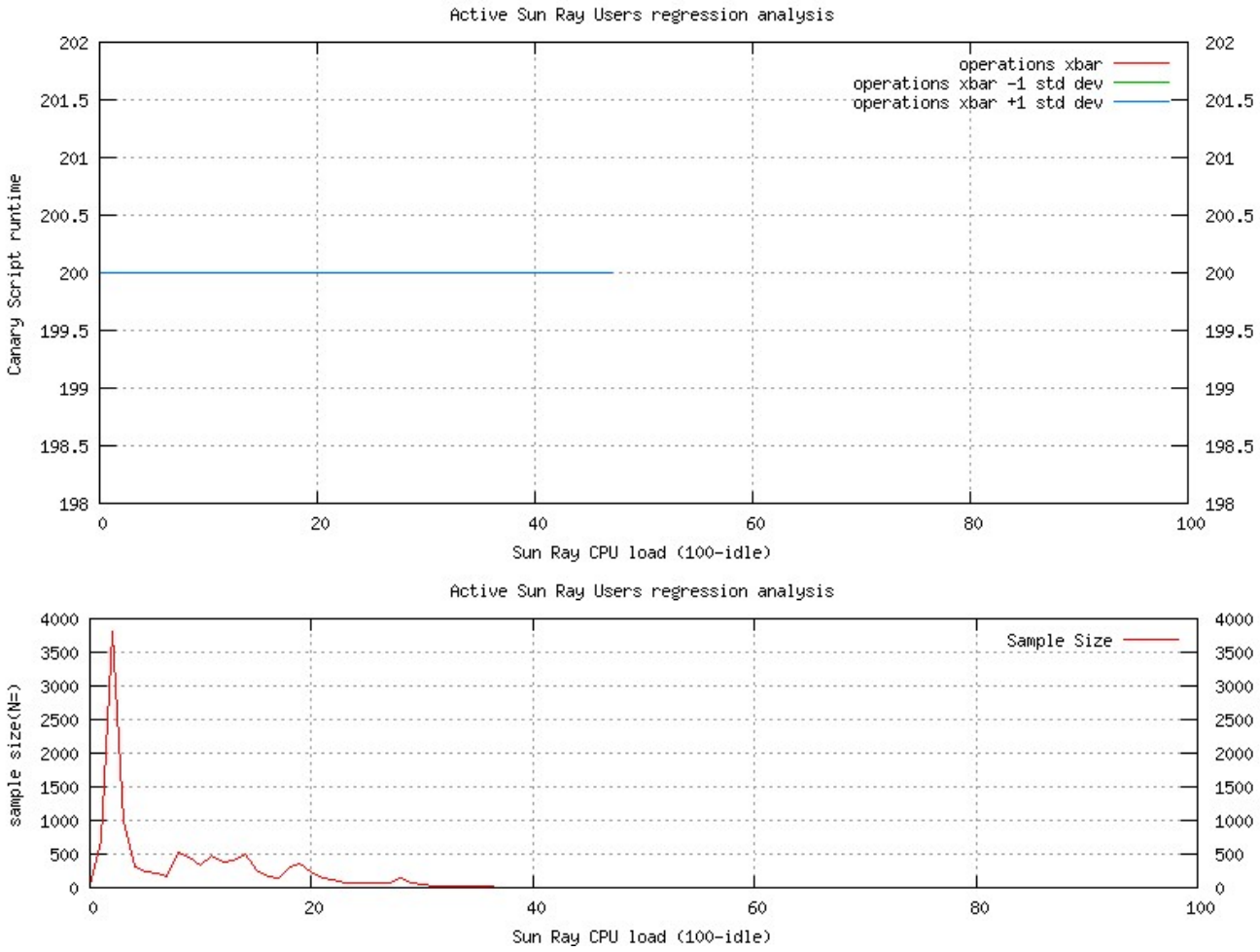


This chart is useful for monitoring user activities on a Sun Ray server. As with the other charts available from this report page, it should show a relationship between the number of users and the number of process Ids on a Sun Ray server.

As the number of users increases, so should the number of processes running on the Sun Ray server. Spikes or dips occurring constantly show that users are either spawning more processes than usual or not and should be investigated.

4.8.4: CPU Bound Canary vs CPU Load

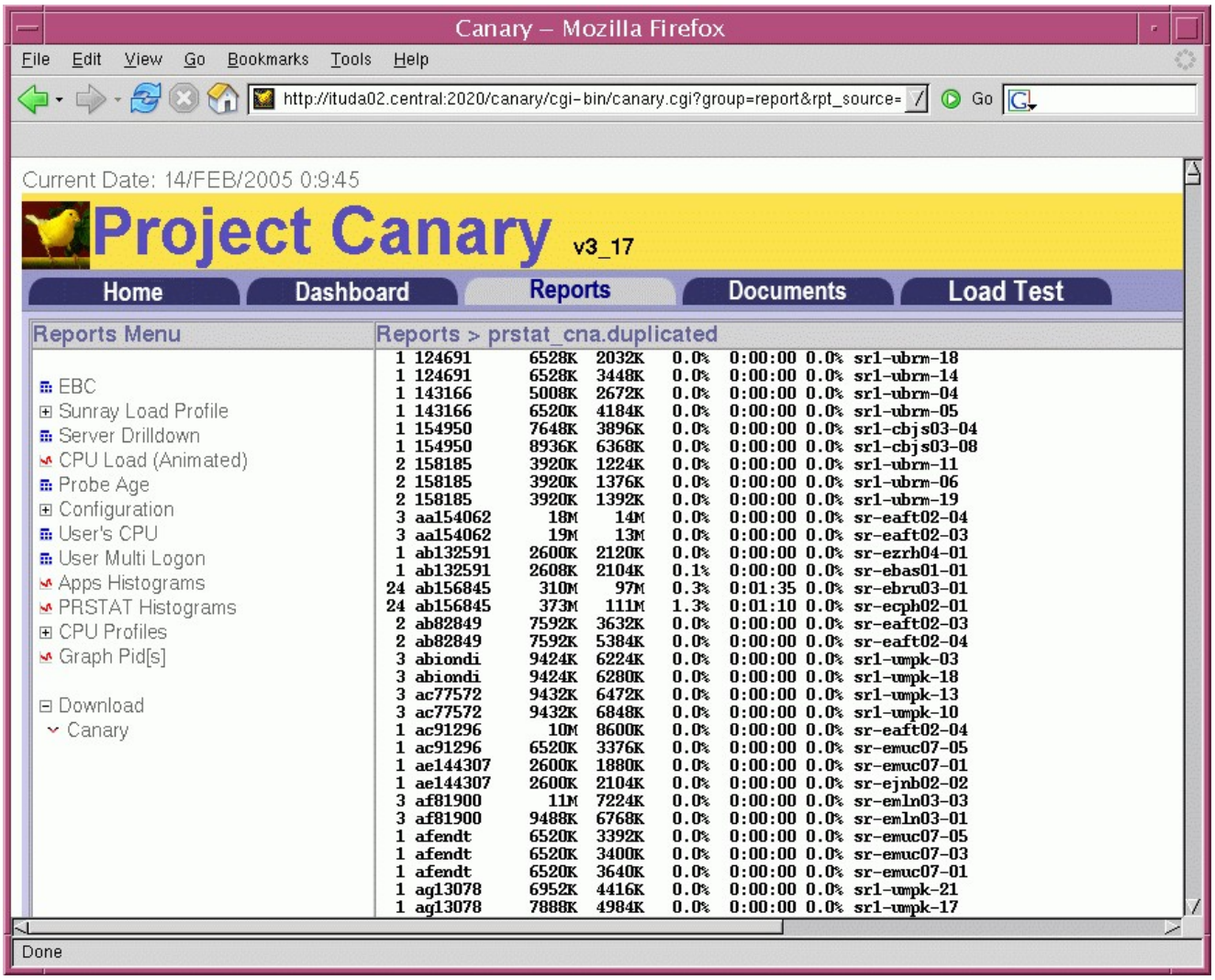
The fourth and final report available from the Users CPU page, the CPU bound process vs CPU load graph is shown below:



On a relatively idle Sun Ray server, this graph should remain relatively flat. As a Sun Ray server becomes loaded down with many users or runaway processes that consume an inordinate amount of CPU time, the graph may fluctuate. Fluctuations in the graph indicate that the server is not sized properly for the load it is bearing.

4.9: User Multi Logon Report

The User Multi Logon report is useful to see which users are using more than one Sun Ray server to do their work on. An example of this report is shown below:



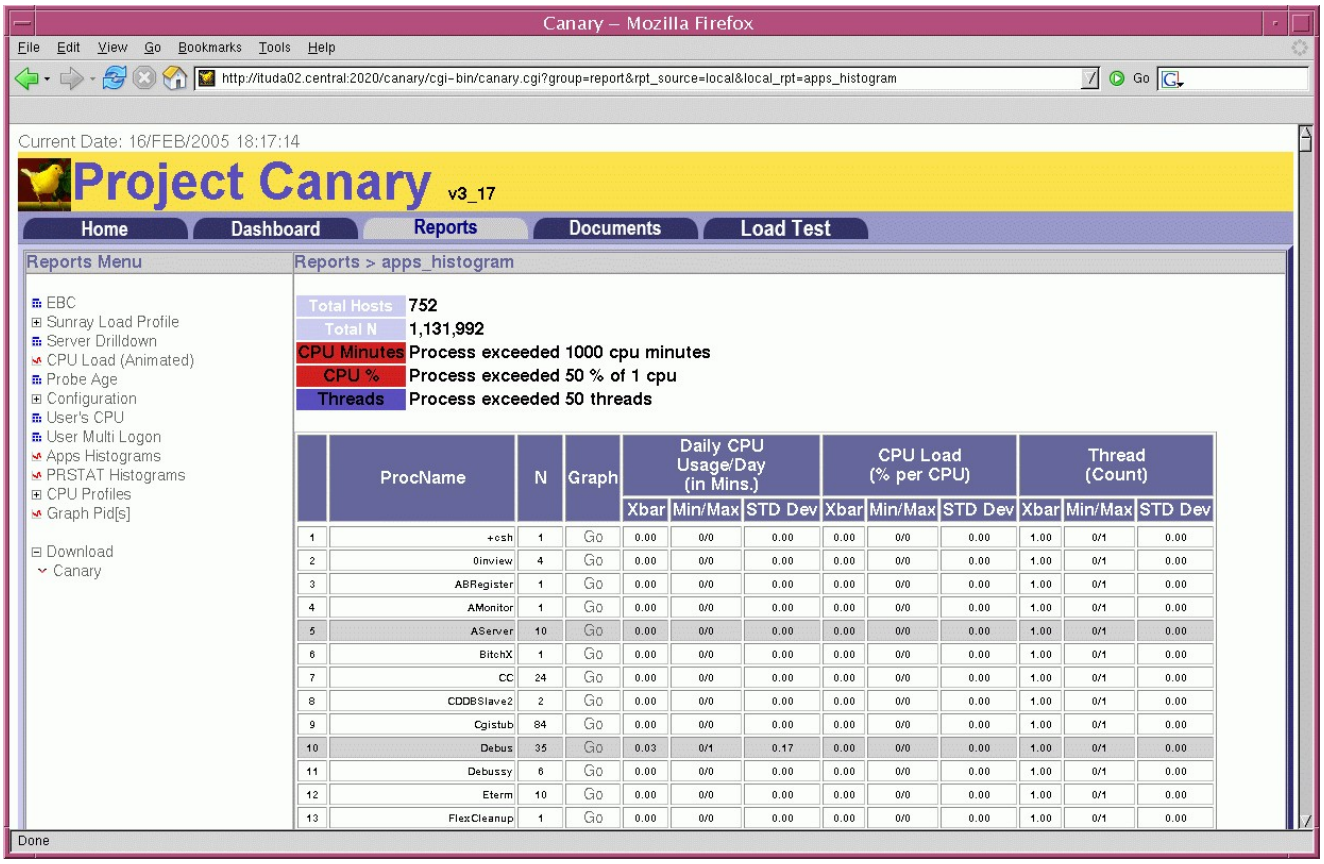
As shown above, users who have login session on multiple Sun Ray servers are listed along with their CPU utilization on the machines they have login sessions running on and which Sun Ray servers they have logged in to. This is useful for tracking down users who frequently let processes get into a "runaway" state or who use more CPU time than is normal.

Many users will change their Sun Ray server using `utswitch(1)` if the Sun Ray server they are using becomes bogged down. If there are a lot of users with login sessions on a couple of different Sun Ray servers, this may indicate a problem that needs to be investigated.

4.10: Apps Histograms

The Apps Histograms report is useful for examining the entire network being monitored by Project Canary software and identifying those processes which tend to either consume a significant amount of CPU resources or which spawn a large number of threads. It also produces a report which attempts to statistically predict what that particular process is likely to do in the future based on its past behavior as determined by data collected by Project Canary software.

An example of the Apps Histogram report is shown below:

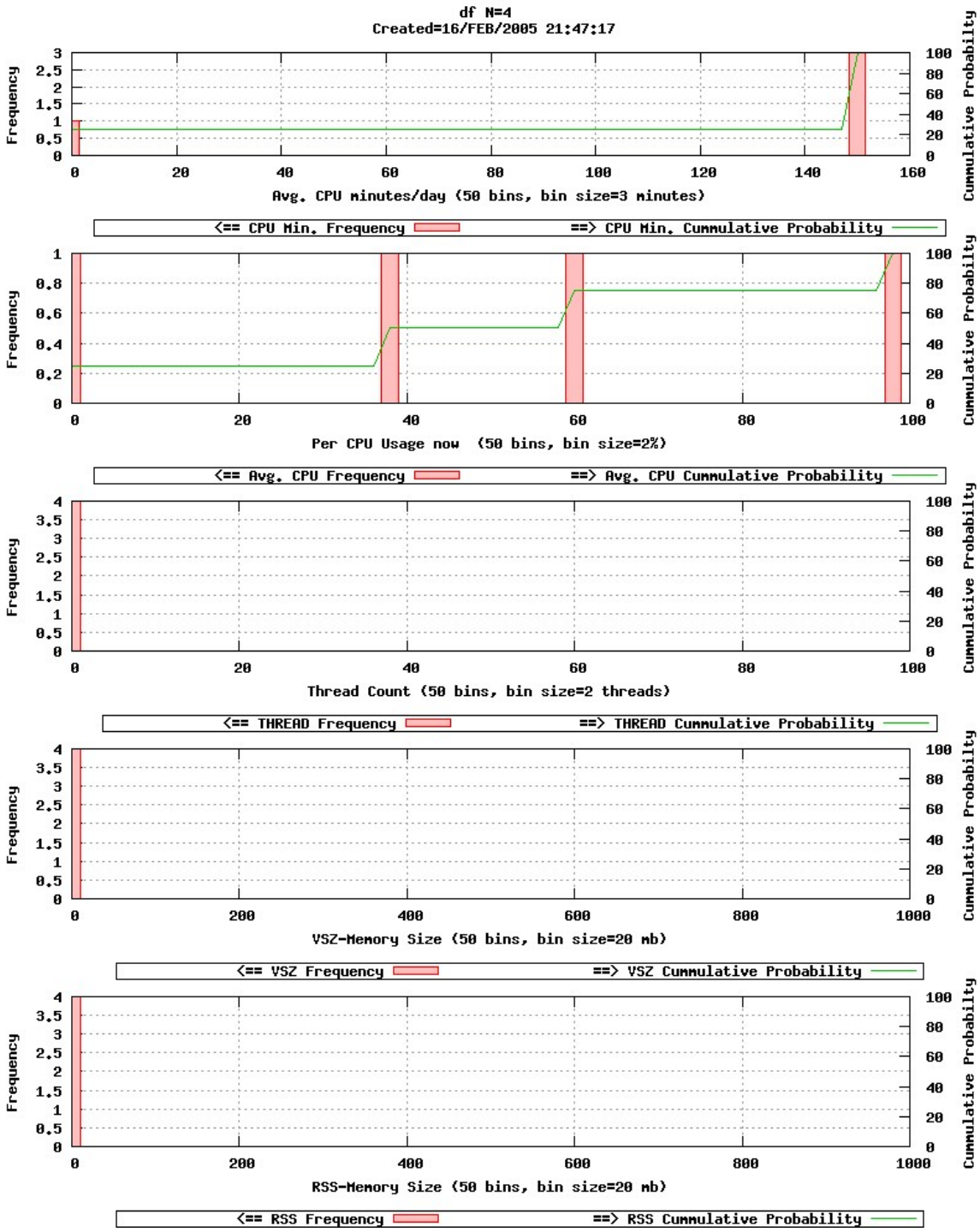


Note that processes are not listed in order of resource consumption, but, rather, are listed alphabetically. Processes which consume resources above and beyond the thresholds shown on this web page will have that utilization marked on the page in red as described in the legend above the data table.

The data table counts the number of instances of a process (N) and has a graph available. Click the "Go" link in the Graph column to see the chart for that process. Project Canary software also calculates the standard deviation of resource consumption. This can be quickly consulted to determine which processes are consumers of large amounts of resources and which are not. This table can be used for resource planning. If applications which consume a large amount of CPU resources are critical to operations, then servers for those applications can be sized properly and allocated for using the data provided in this report.

4.10.1: Apps Histograms Graph

The Apps Histograms report also generates a graph which attempts to predict resource consumption and behavior patterns of applications by examining their past behavior. An example of such a graph is shown below, for the `df(1)`, a commonly used application:



From the graph shown on the previous page, one can see clearly that `df(1)` uses quite a bit of CPU time but not much in terms of memory utilization, nor does `df(1)` spawn multiple threads. Different applications behave differently and this graph can help a Systems Administrator generate a profile for an application so its behavior is better understood.

This report is particularly useful on mixed-use servers where large numbers of different applications are being

used. Different applications have different requirements and examining the behavior of these applications together paints a distinctive picture of the server's operation and is useful for capacity planning and tuning.

Ideally, a server should be able to run a mix of applications which utilize its resources to their maximum without impacting performance. Examining the resource utilization patterns of various applications can help maximize computing potential while minimizing cost. An analysis of application "signatures" on a server might indicate that some applications "play well" together while others do not.

Balancing applications on a server or migrating some applications to different servers where the applications utilize resources in complimentary manners as opposed to competing for the same resources constantly may help reduce cost and complexity and maximize the return on investment in computing resources.

For example, if two applications appear to be CPU bound, and two others are found to be memory bound, it may be useful to mix them so that resource utilization is maximized as opposed to adding additional hardware resources to the server being monitored. This may allow for a cost savings, but really, that all depends on the particular characteristics of the applications being used.

Additionally, identifying applications which utilize a significant amount of CPU time will be of help when determining which server to place an application on. Since most computing environments contain a mixture of CPU speeds, applications which spawn multiple threads are better placed on servers with multiple CPUs. With most multithreaded applications, CPU speed is less important than the ability to run multiple threads simultaneously. Conversely, monolithic applications which are not multithreaded should be placed on servers with faster CPUs.

Each application has unique characteristics and only an analysis of the application's unique behavior will permit a systems administrator to extract the maximum potential from the computing environment. Project Canary software is uniquely equipped to assist with this task.

4.11: PRSTAT Histograms

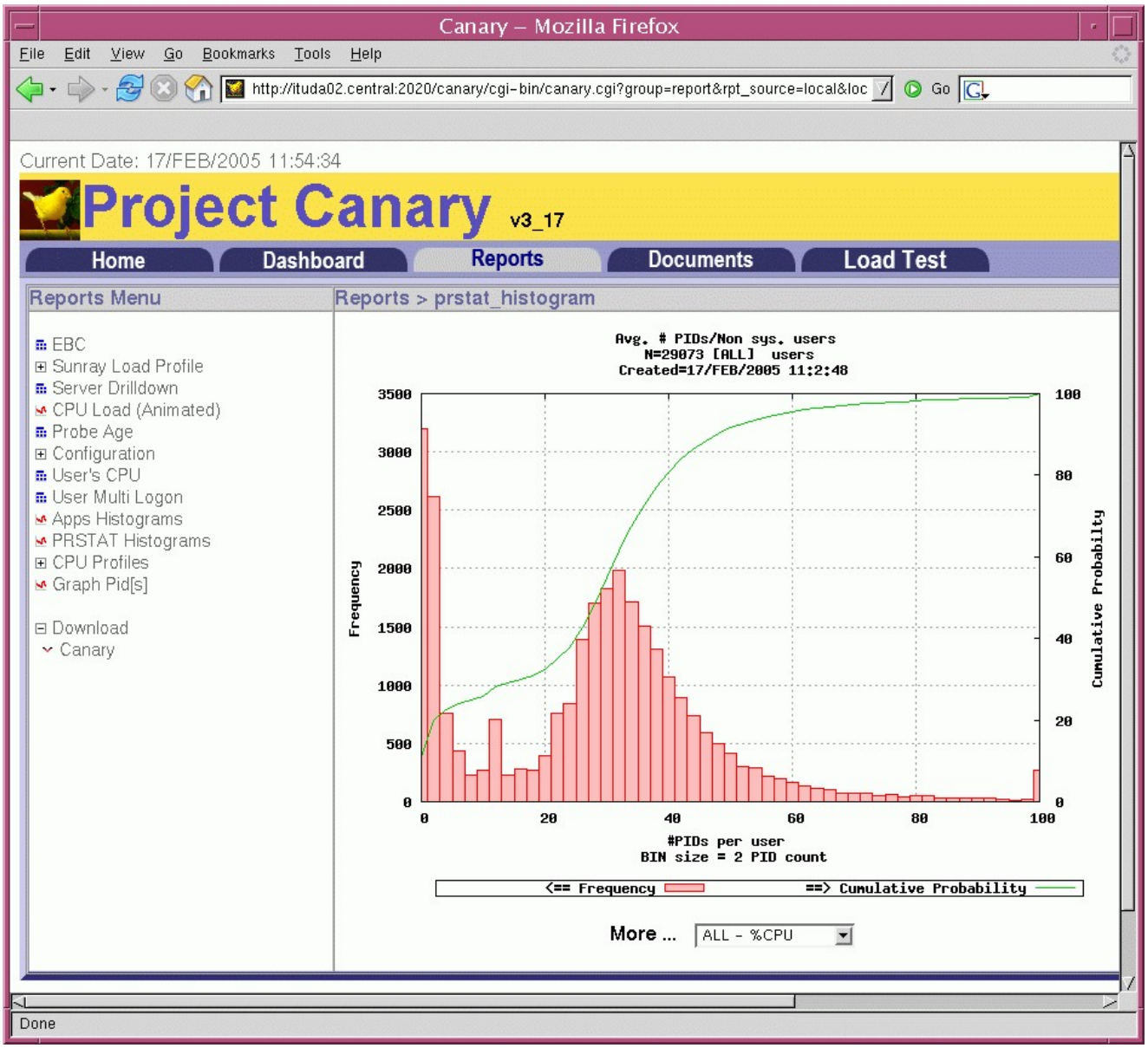
4.11.1: Process Ids per User

The PRSTAT Histograms report displays multiple graphs selected from a drop-down menu and displays statistics on a per-user basis. Specifically, the PRSTAT Histograms show memory utilization, number of processes spawned, and CPU utilization.

Graphs available from this report profile "all" processes run and also separate out certain user processes, such

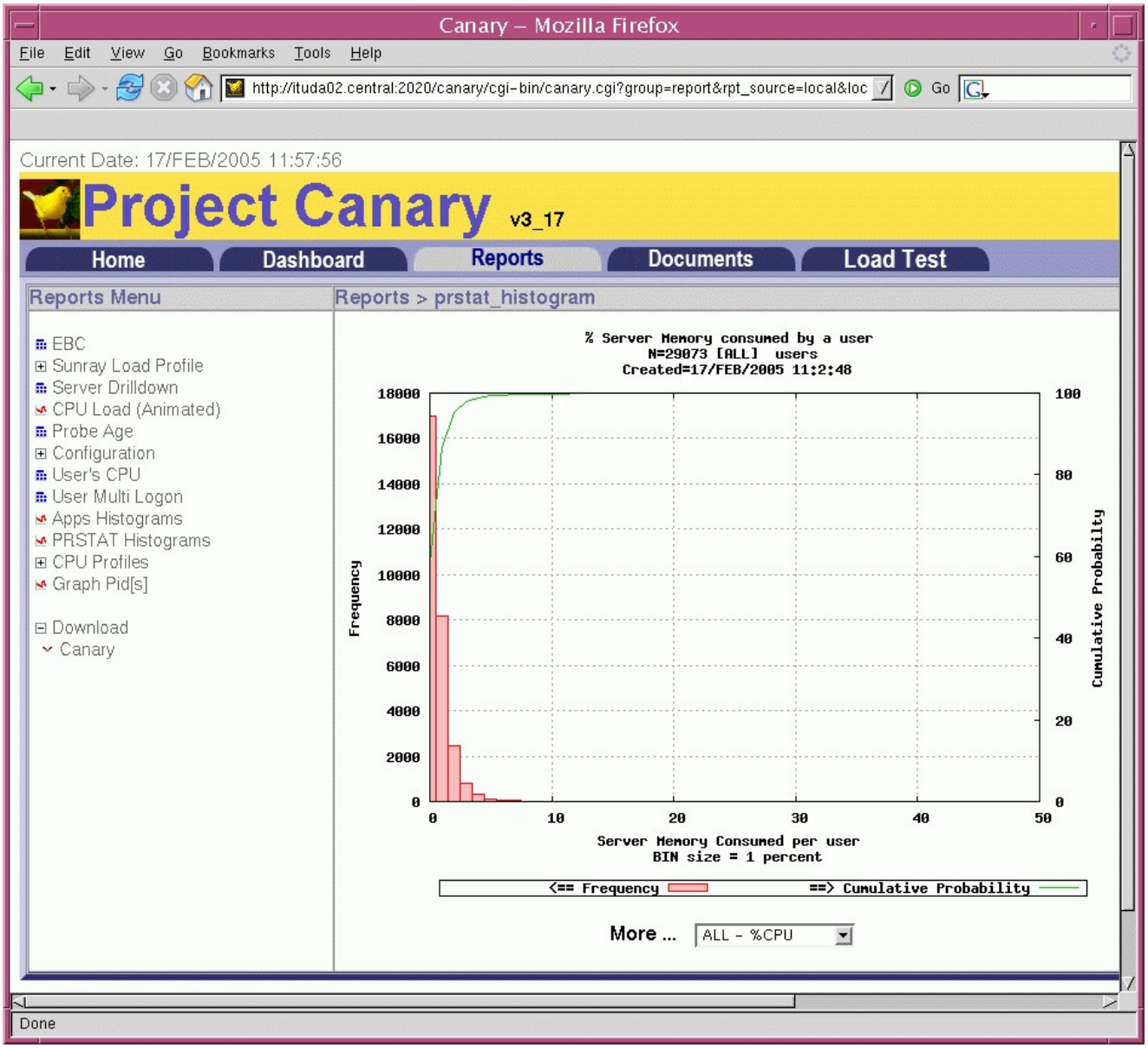
as CDE and GNOME which have been separated out to allow for extra scrutiny.

When the PRSTAT Histograms report is selected, the first graph shown gives the average number of processes run on a per-user basis as shown below:



4.11.2: Percentage of System Memory Consumed per User:

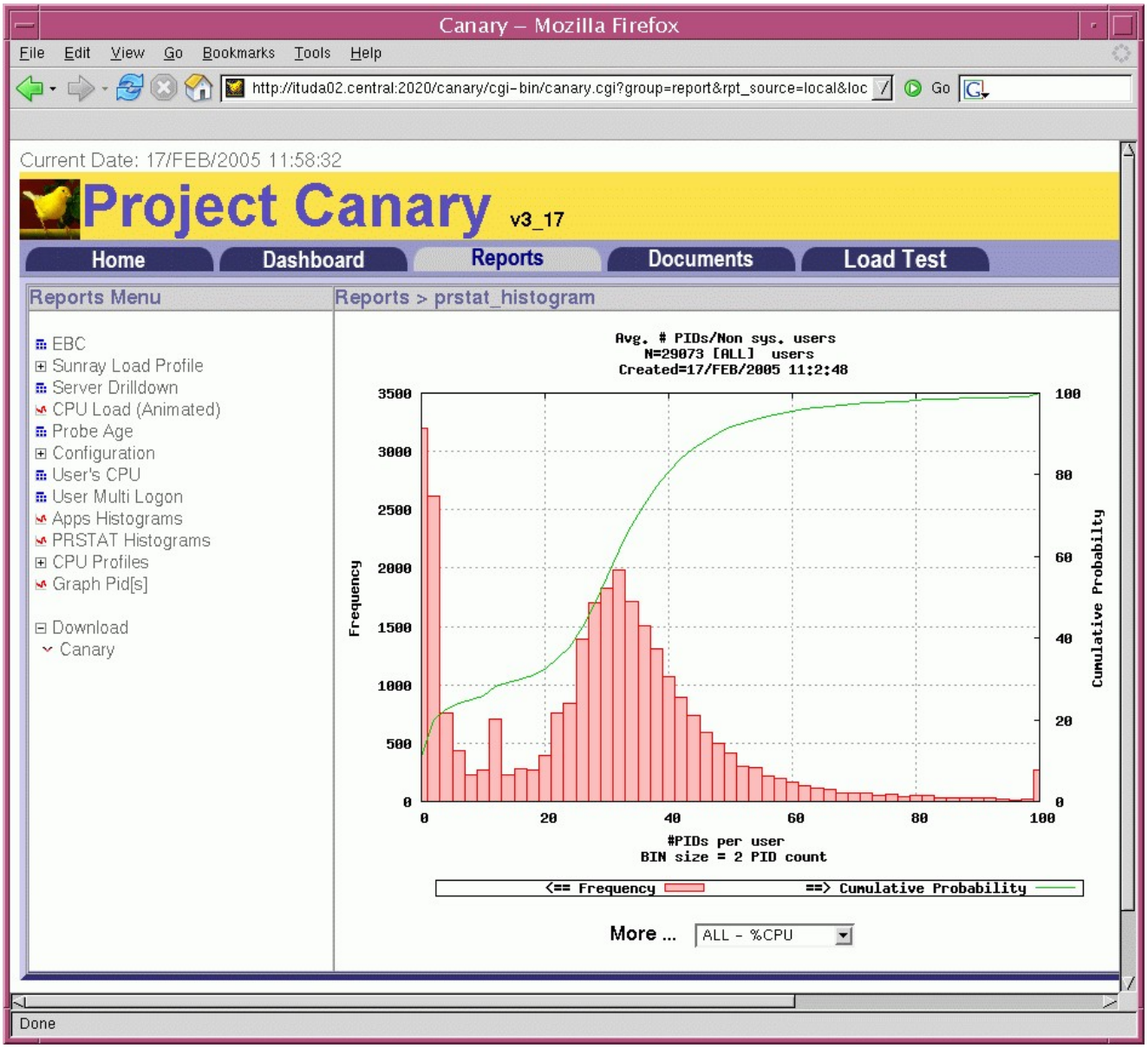
The second report available from the drop-down menu underneath the graph shows the percentage of memory consumed by a user, again, on a per-user basis:



This graph shows that the majority of users are not consuming more than one or two percent of a server's memory, on average. If utilization suddenly spikes in the middle of the graph, then that should be investigated.

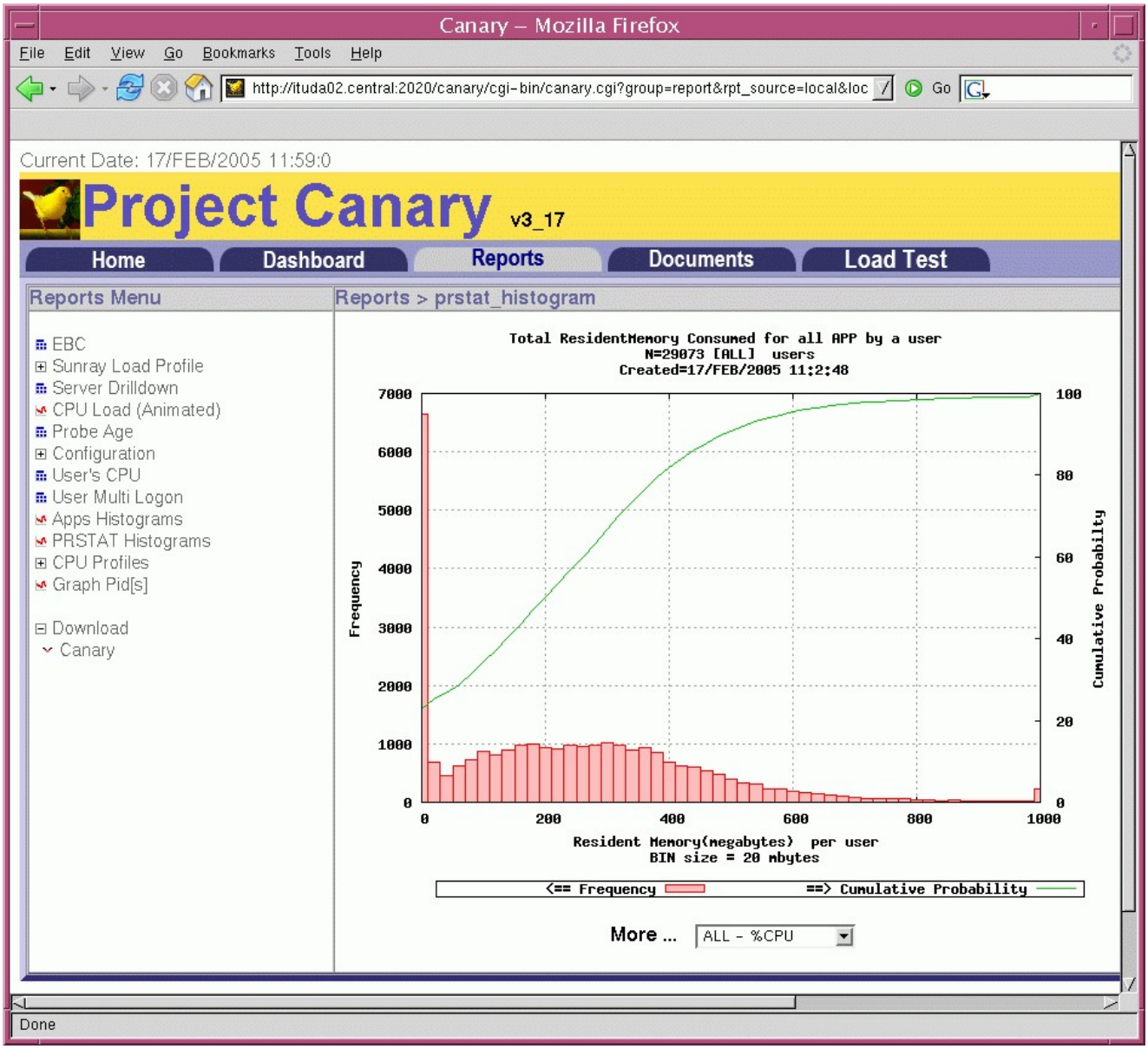
4.11.3: Average Number of Processes by Non-System Users

This graph is a duplicate of the first graph.



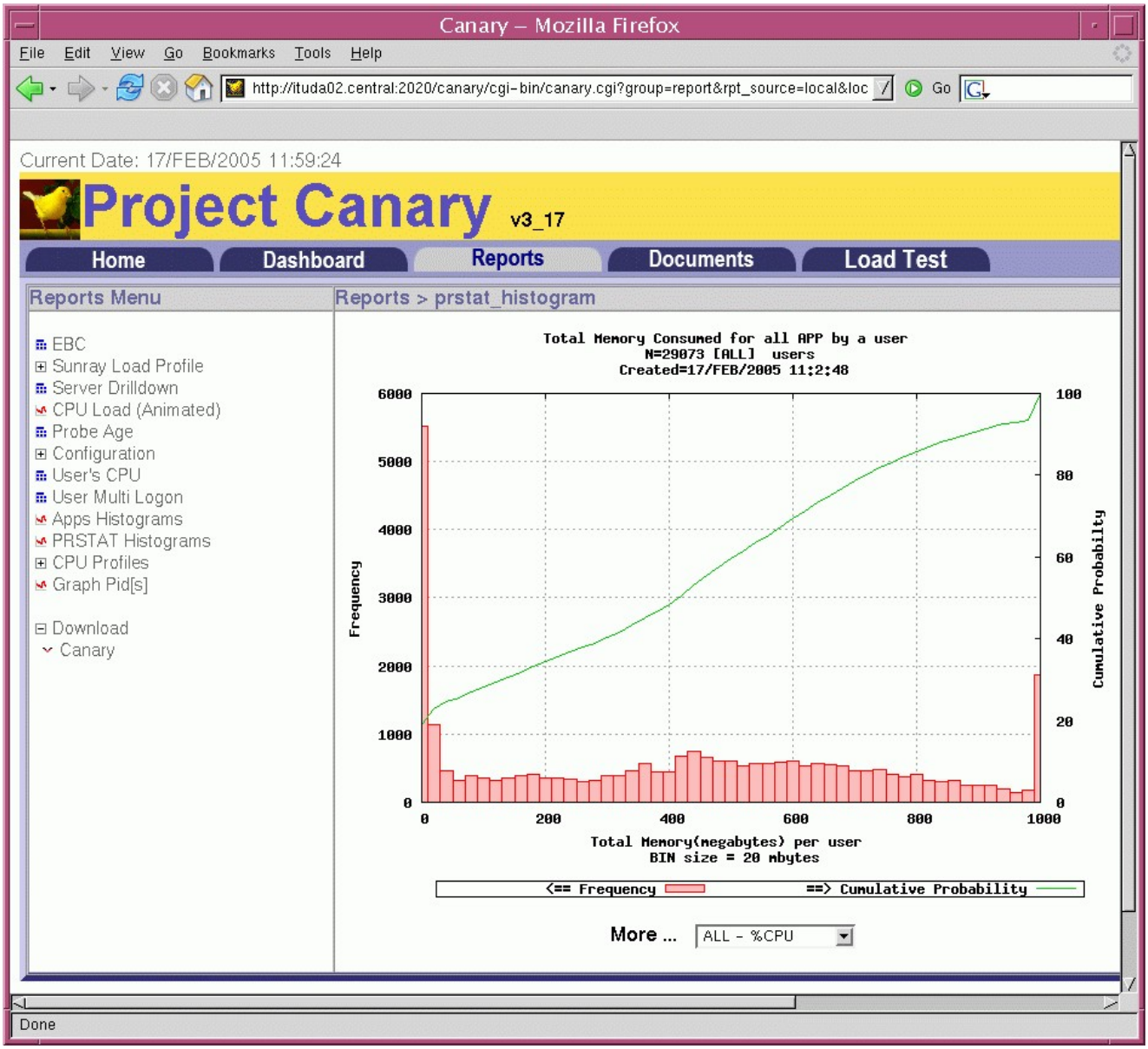
4.11.4: Total Resident Memory Consumed for All Applications by a User

This graph shows approximately how much real memory is being consumed by applications run by users on the server. Note that the majority of applications require very little real memory.



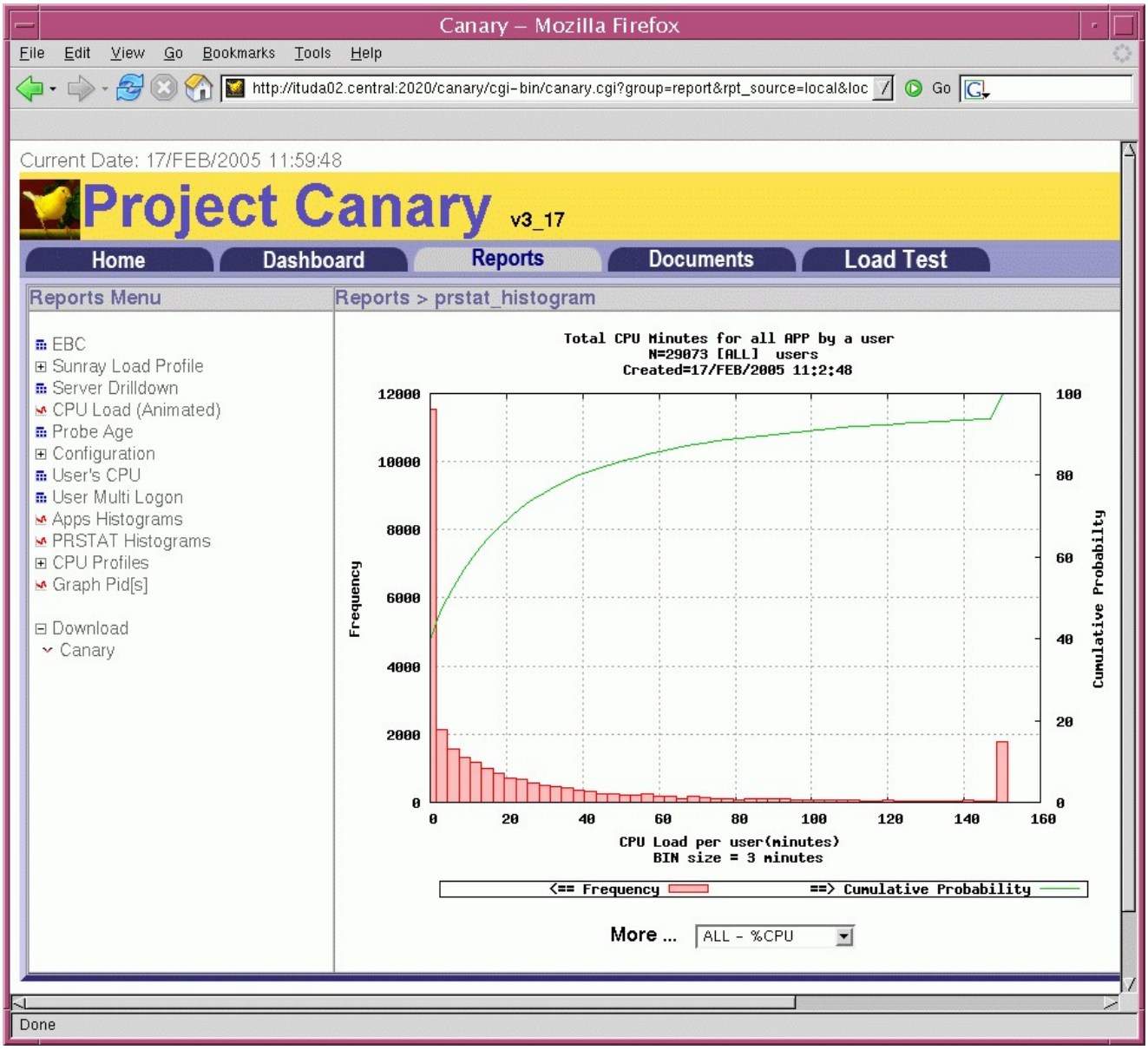
4.11.5: Total Memory Consumed for all Applications by Users

This graph shows the total memory utilization, both virtual and real by all applications run by users on a server. Note again, that the majority of applications require very little memory, however, there are certain applications that require a large amount of memory.



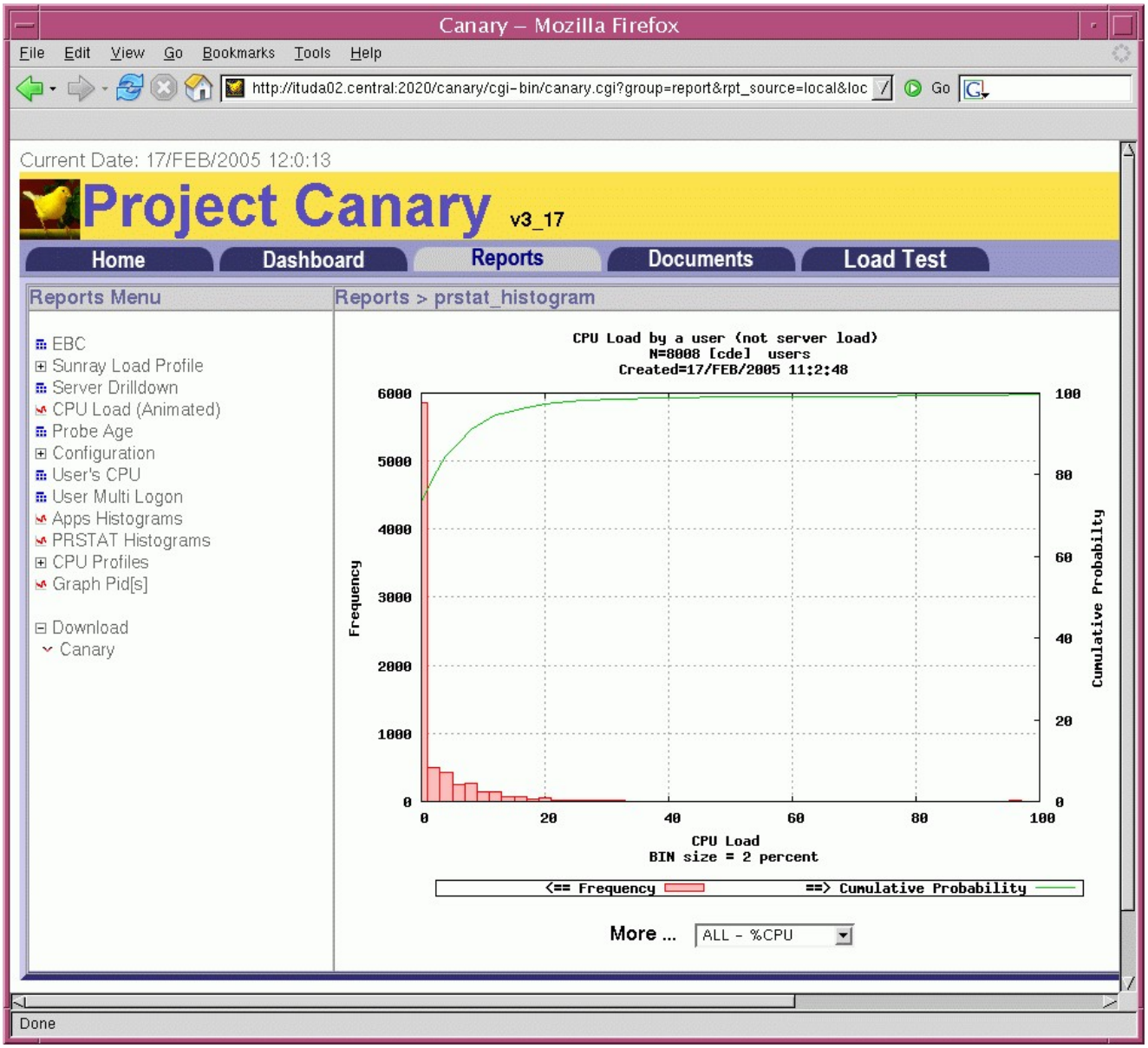
4.11.6: Total CPU Minutes Used for all Applications by Users

This graph shows the amount of total CPU time a user will consume on a server. Note that the majority of users consume very few CPU minutes and that the amount of CPU time used per user tapers off and that there are a few "power users" who use an inordinate amount of CPU power.



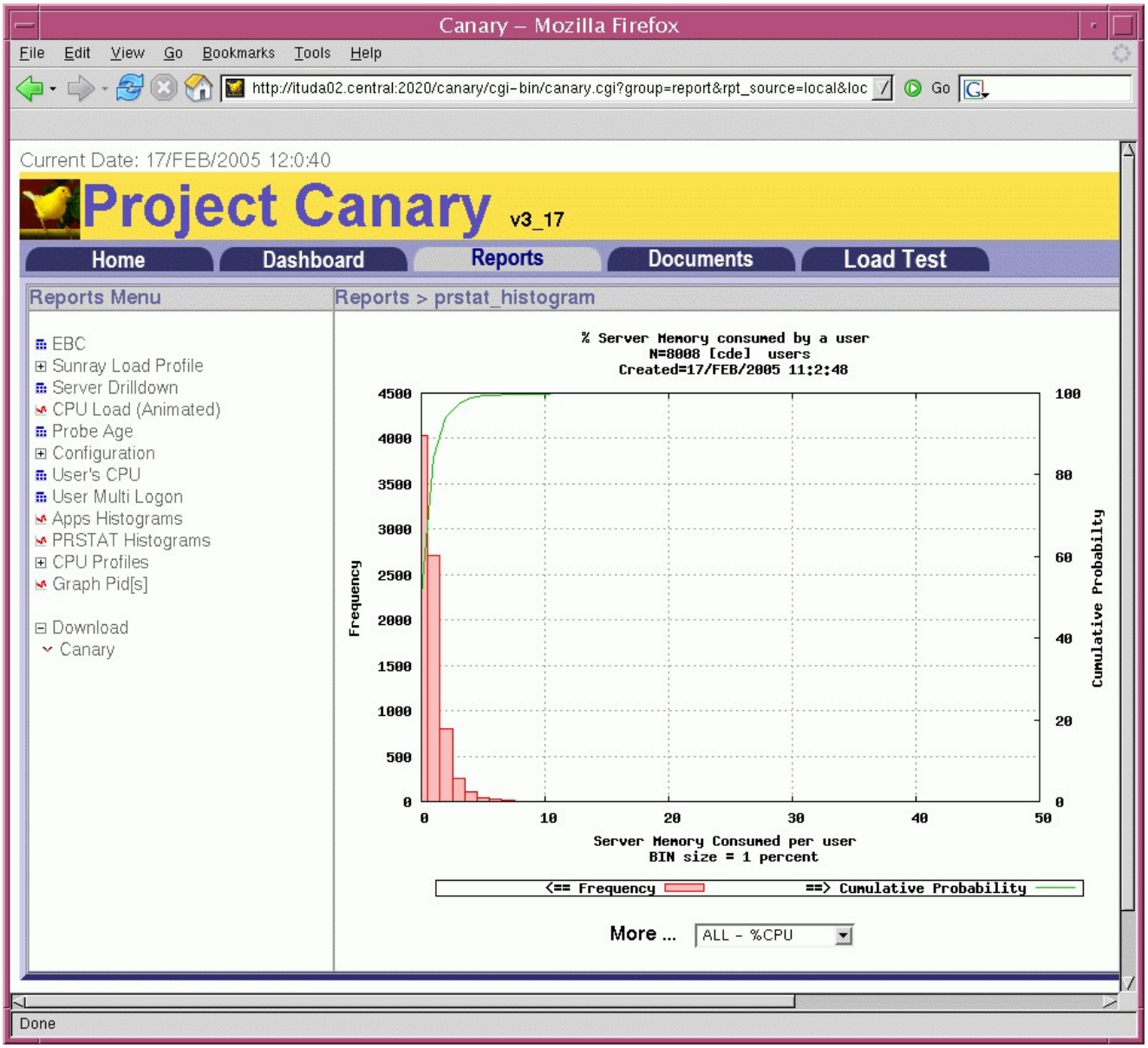
4.11.7: CPU Load by a User (not server load)

This graph shows the CPU load generated by the CDE desktop. Note that the majority of users generate very little load on the server and that the load tapers off quickly in the graph below:



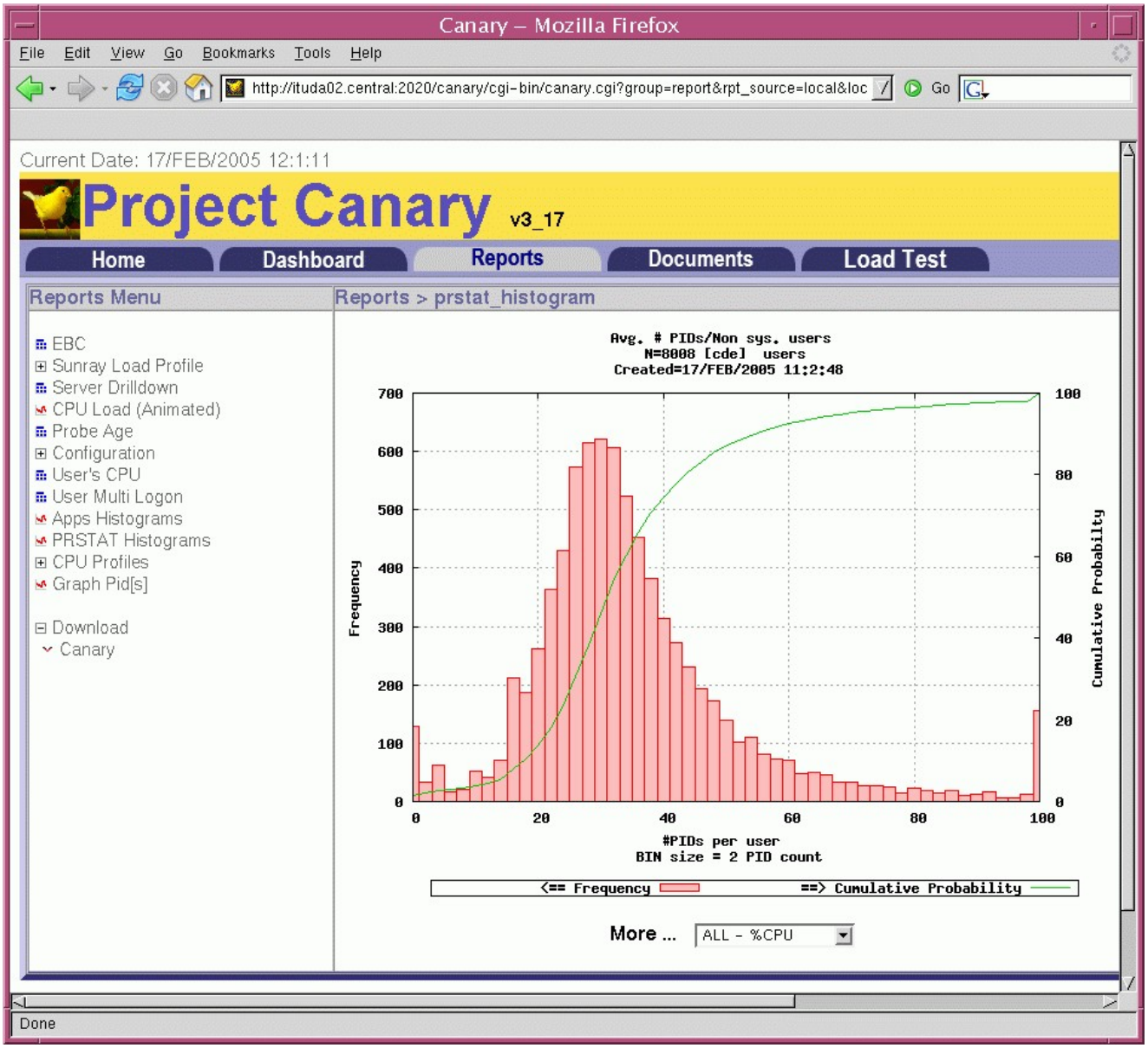
4.11.8: Percentage of memory consumed by a user

This graph shows the percentage of memory consumed by the CDE desktop. Note that operating the CDE desktop uses very little memory for the majority of users.



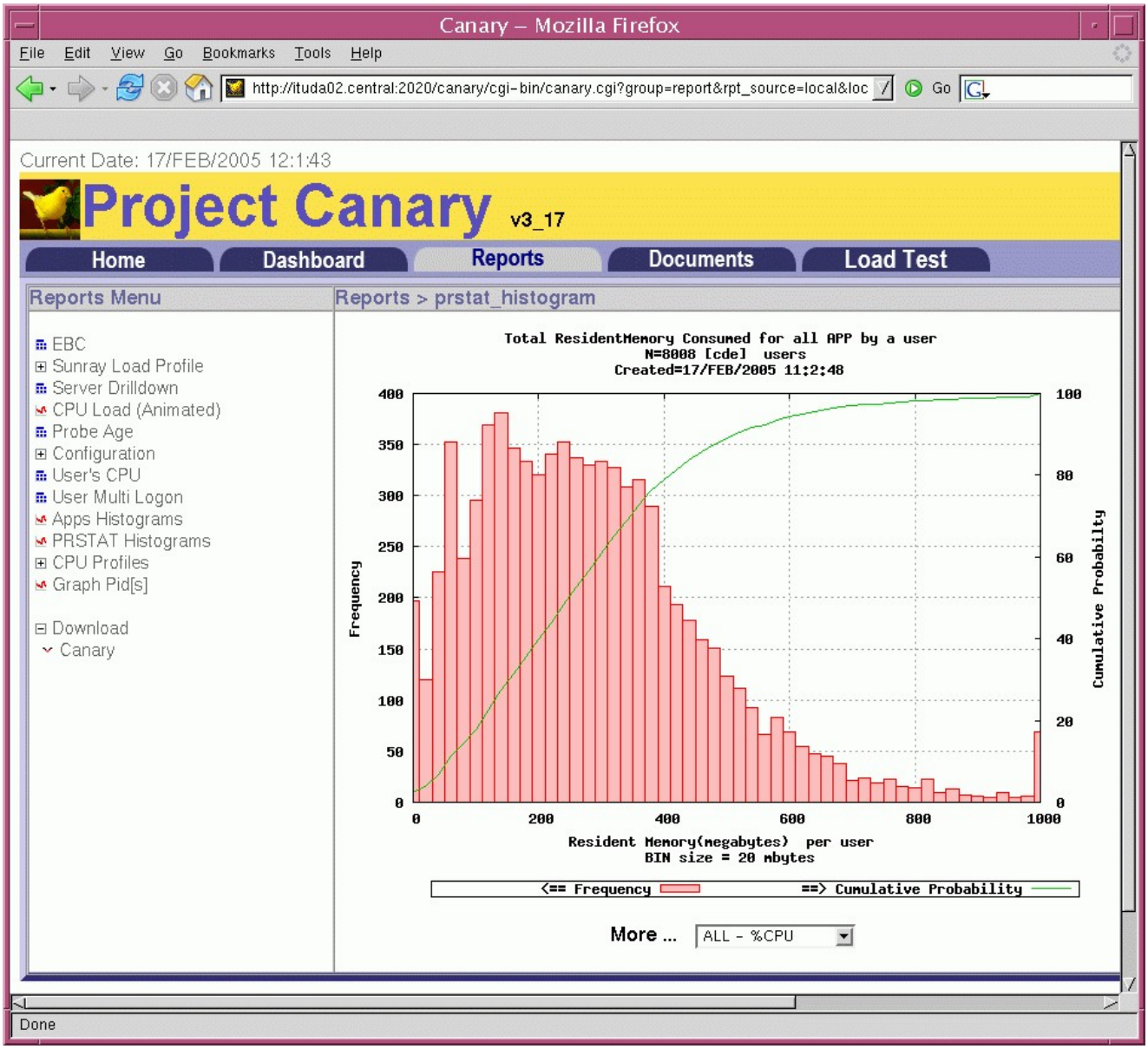
4.11.9: Average number of processes per non-system user

This graph shows the number of processes spawned by CDE desktop users to operate their environment. Note that usage spikes, then tapers off and only a few users are "driving" their desktop environment.



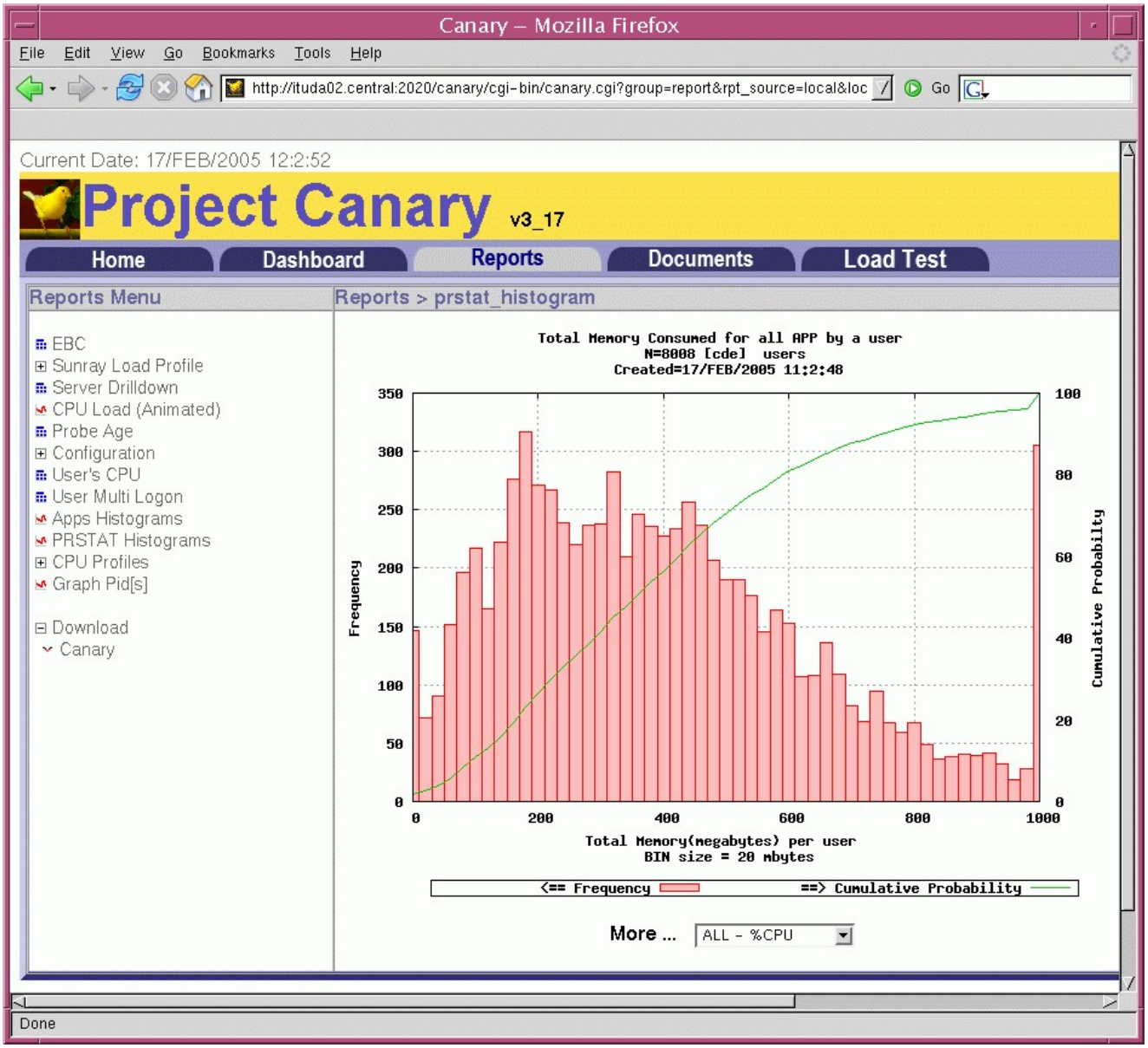
4.11.10: Total resident memory consumed for all applications by a user

This graph shows the amount of resident memory consumed for all applications by a user who is using the CDE desktop. Note that at around 400mb per user, utilization drops rapidly with the exception of the few power-users at the end of the graph.



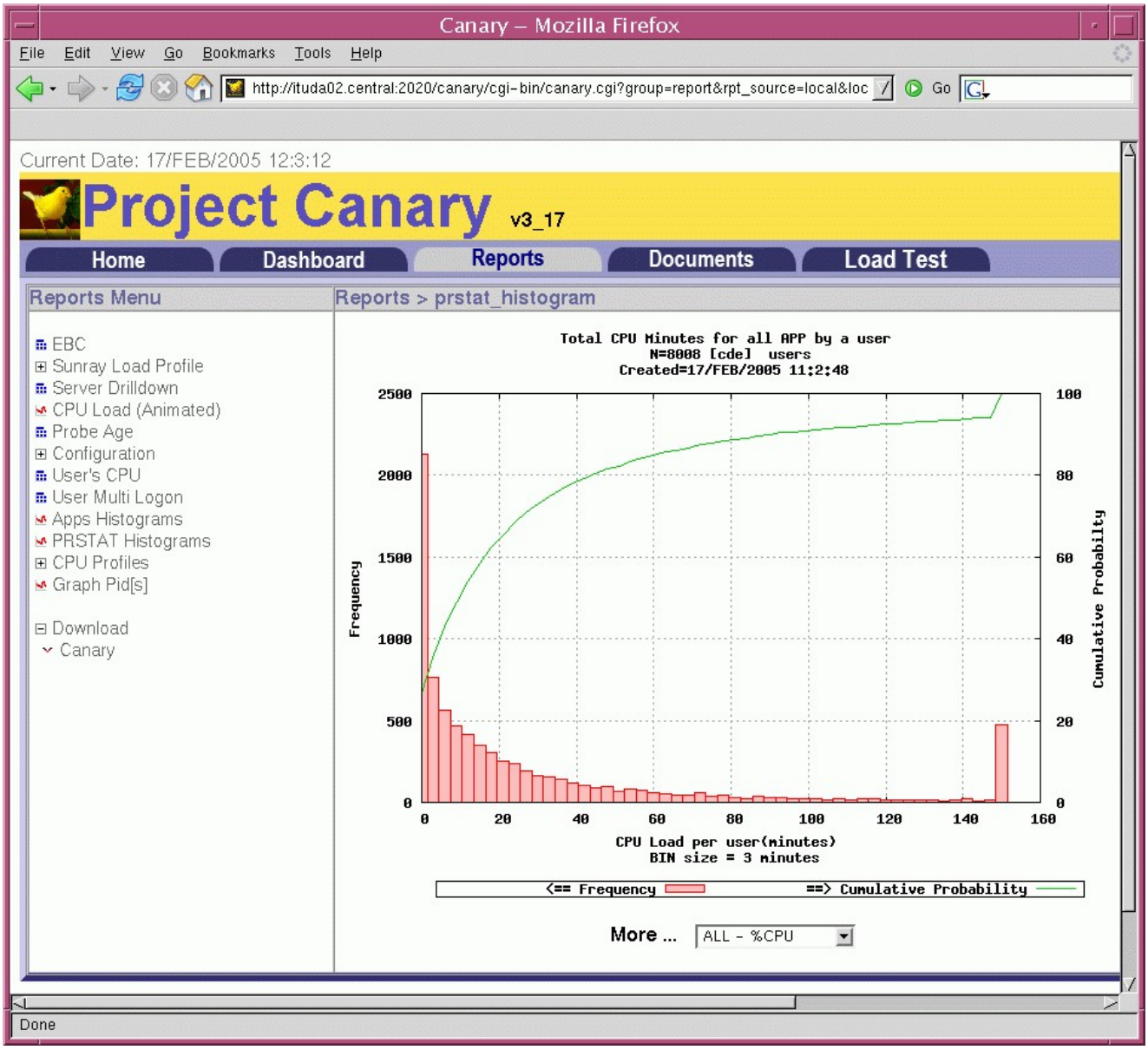
4.11.11: Total memory consumed for all applications by a user

This graph breaks down the total amount of memory, both virtual and real, by all users. Note that usage tapers off as in previous graphs, with the exceptions of the power-users toward the end of the graph.



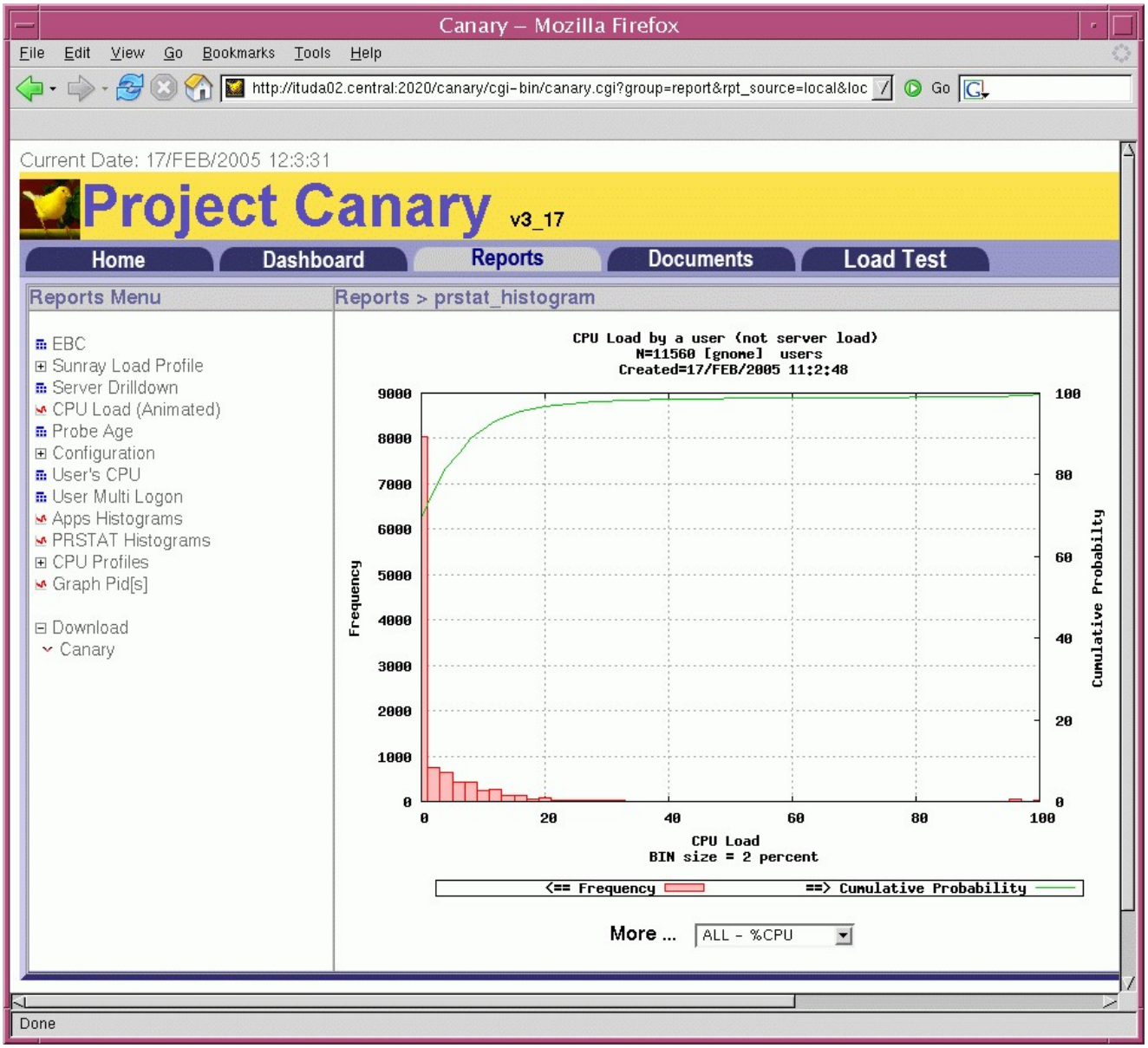
4.11.12: Total CPU minutes consumed for all applications by a user

This graph displays the total amount of CPU time consumed by CDE desktop users. Note that most users consume very little CPU and that usage tapers off and spikes again with the power-users at the end of the graph.



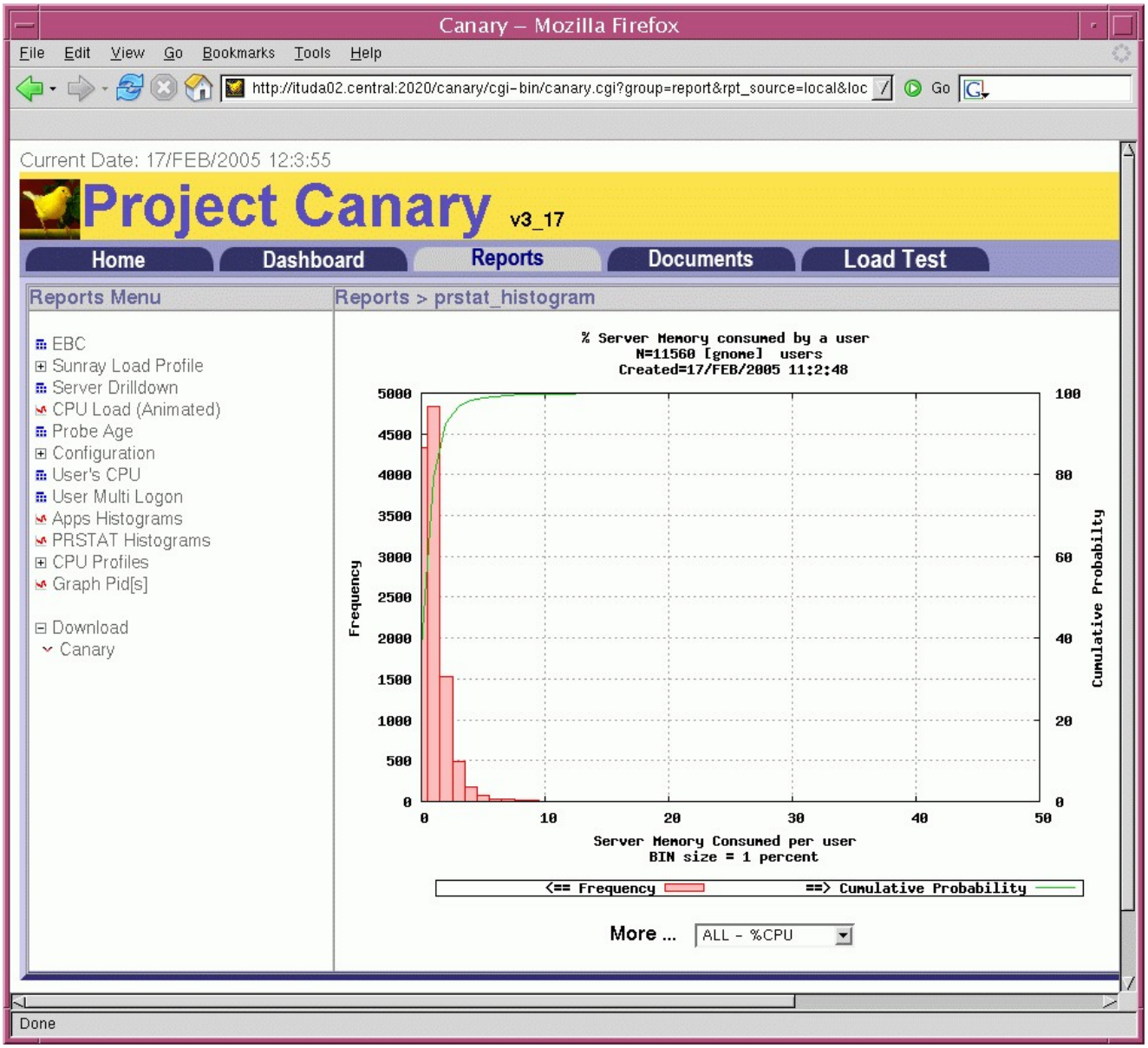
4.11.13: CPU load by a user (not server load)

This graph shows GNOME users and their CPU utilization. Note that the majority of users utilize very little CPU with the exception of a power-user toward the end of the graph.



4.11.14: Percentage of server memory consumed by a user

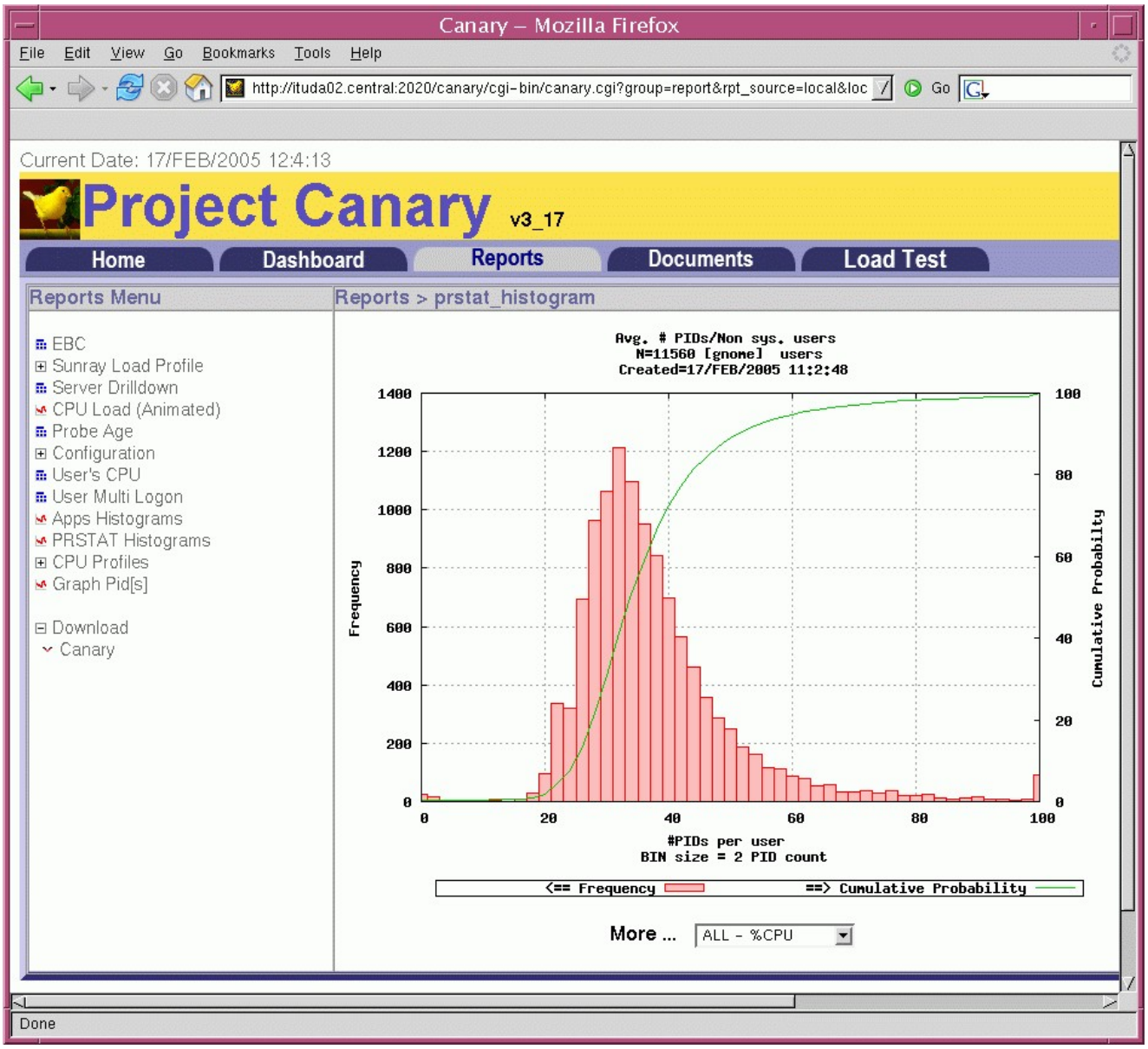
This graph breaks out GNOME users as opposed to CDE users on a Sun Ray server. Note that the majority of GNOME desktops use very little memory and that utilization spikes, then falls rapidly.



4.11.15: Average number of processes per user for non-system users

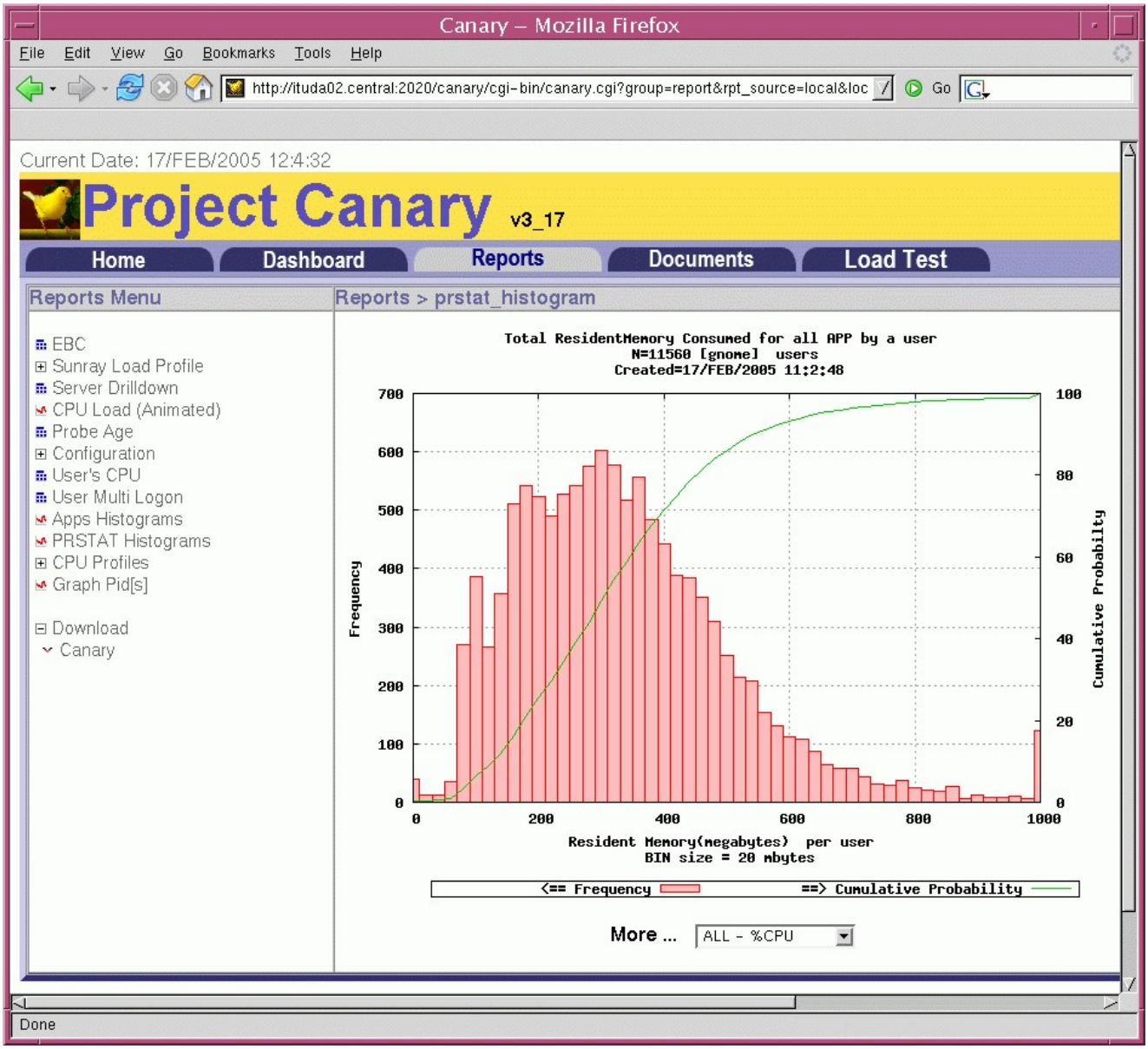
This graph separates GNOME desktop users from CDE desktop users. Note that the "average" GNOME desktop user spawns approximately 30 processes and that the number of processes spawned per user rises from zero rapidly, reaches an apex, then drops.

Also note the power-user at the end of the graph who is driving the Sun Ray server.



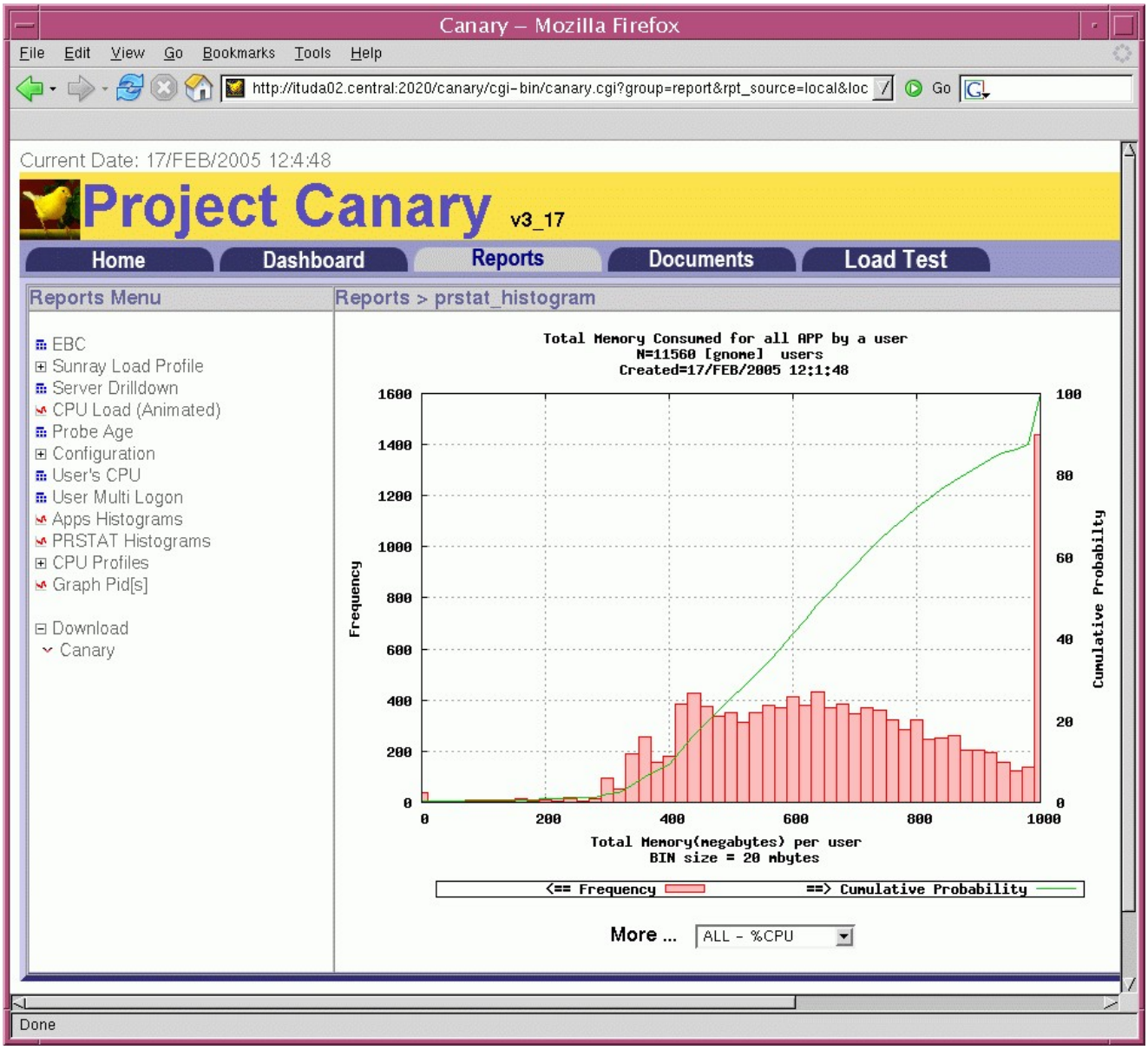
4.11.16: Total resident memory consumed for all applications by a user

Again, this graph breaks GNOME users out and identifies them specifically. Note that users average around 300mb of resident memory usage. Also, note the power user at the end of the graph consuming about a gigabyte of real memory.



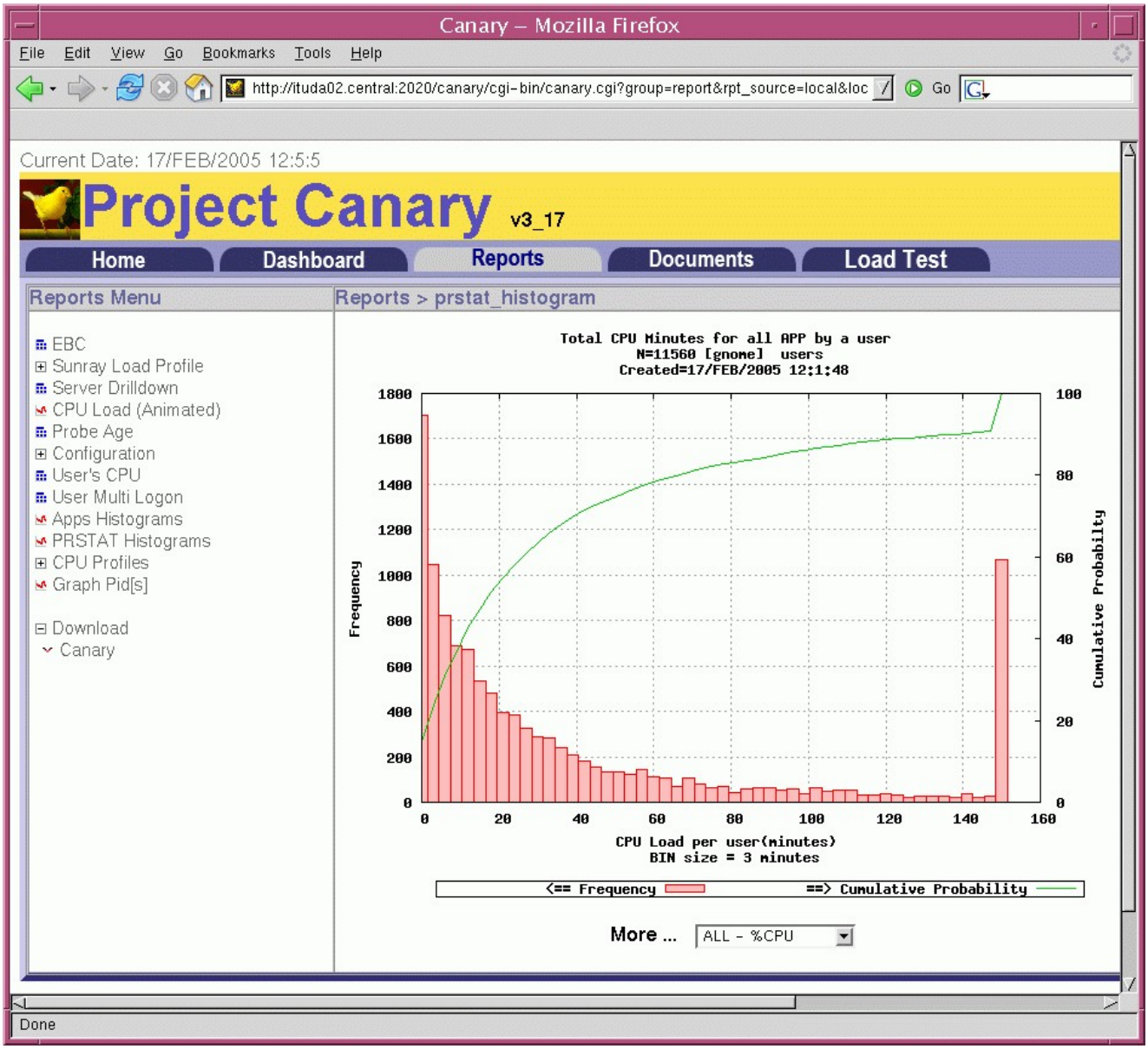
4.11.17: Total memory consumed for all applications by a user

This graph shows that GNOME desktop users tend to use more total memory (both virtual and real) than CDE desktop users. Memory utilization rises, then falls in a bell-shaped curve with the exception of the power-user at the end of the graph.



4.11.18: Total CPU minutes for all applications by a user

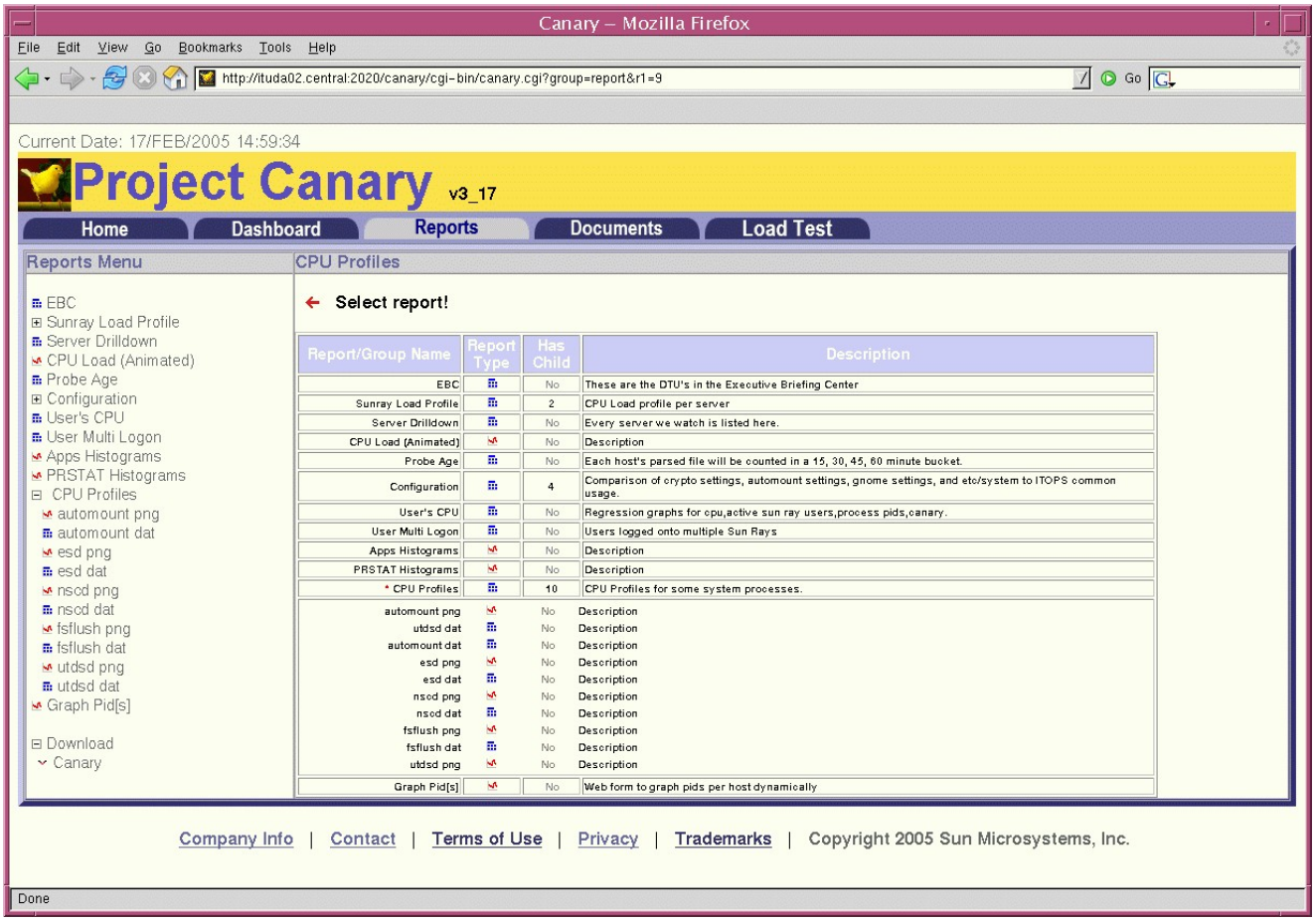
This graph again breaks GNOME desktop users off from CDE desktop users and examines their CPU utilization separately. Note that the majority of GNOME users use very little CPU and that utilization tapers off until the power-user is seen at the end of the graph.



4.12: CPU Profiles Report

The CPU Profiles report gives the CPU profiles for some system processes. In particular, this report gives data on the `automount`, `esd`, `ncsd`, `fsflush`, and `utdsd` processes.

The CPU Profiles report page looks like this:



On the CPU Profiles page, there is a submenu to the left. This submenu has a number of choices of reports which can be displayed in the web browser.

There are two types of reports available from the submenu on the left. Reports labeled with the red icon denote a graphical report whereas the blue icon denotes a text report.

Each report under the CPU Profiles menu has both a graph and a data file. The data file is needed to note where on each graph a particular machine is plotted. Machines are grouped by "Campus" as described earlier in this document and are plotted on the graph. Typically, groups of machines tend to be configured similarly so they will appear on the graph with similar CPU usage profiles.

These graphs are useful mostly to compare different groups of machines against each other.

The available reports under the CPU Profiles report submenu are:

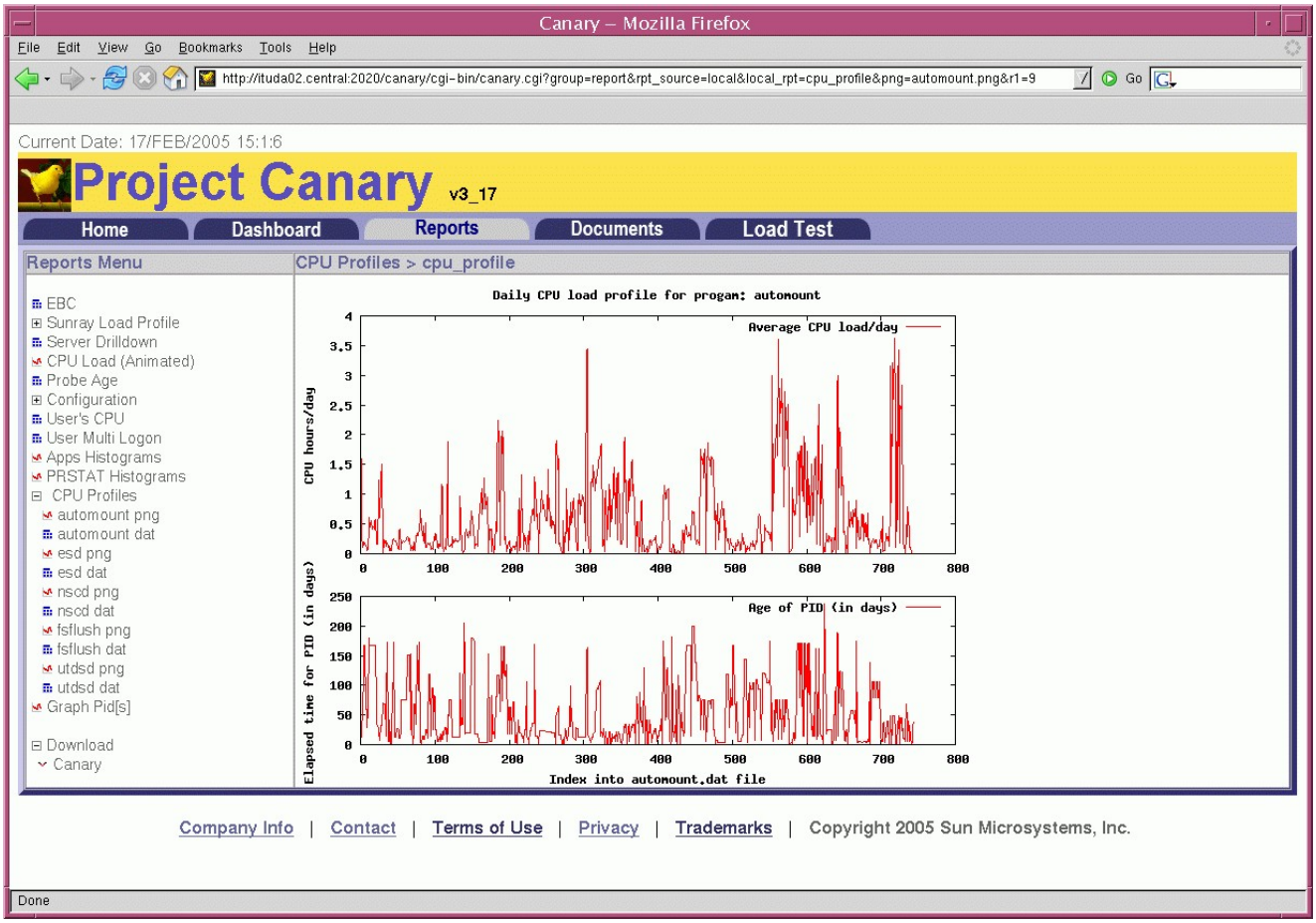
- automount png: This is a graph showing the CPU utilization of the `automount(1M)` daemon across all servers monitored by Project Canary software
- automount dat: The raw data file used to generate the automount png graph
- esd png: A graph showing the CPU utilization of the `esd` (SunMC) daemon across all servers being

monitored
- esd dat: The raw data file used to generate the esd png graph
- ncsd png: A graph showing the CPU utilization of the `ncsd` (name service cache daemon) across all servers being monitored
- ncsd dat: The raw data file used to generate the ncsd png graph
- fsflush png: A graph showing the CPU utilization of the `fsflush` daemon; a daemon which flushes buffers to disk approximately every 30 seconds
- fsflush dat: The raw data file used to generate the fsflush png graph
- utdsd png: A graph showing the CPU utilization of the `utdsd` daemon
- utdsd dat: The raw data file used to generate the utdsd png graph

All of these reports compare data across all servers being monitored by Project Canary software. The goal in producing these graphs is to determine if machines are running in production with different configurations. Servers configured identically should show little to no variation between them in terms of performance under a similar load. Servers configured differently should show large variations in terms of performance under a similar load.

4.12.1: automount png Report

The automount png graph compares the CPU utilization of the `automount(1M)` daemon across all servers monitored by Project Canary software. An example of this graph is shown below:



Note that peaks and valleys appear "connected." This is because machines are grouped when graphed.

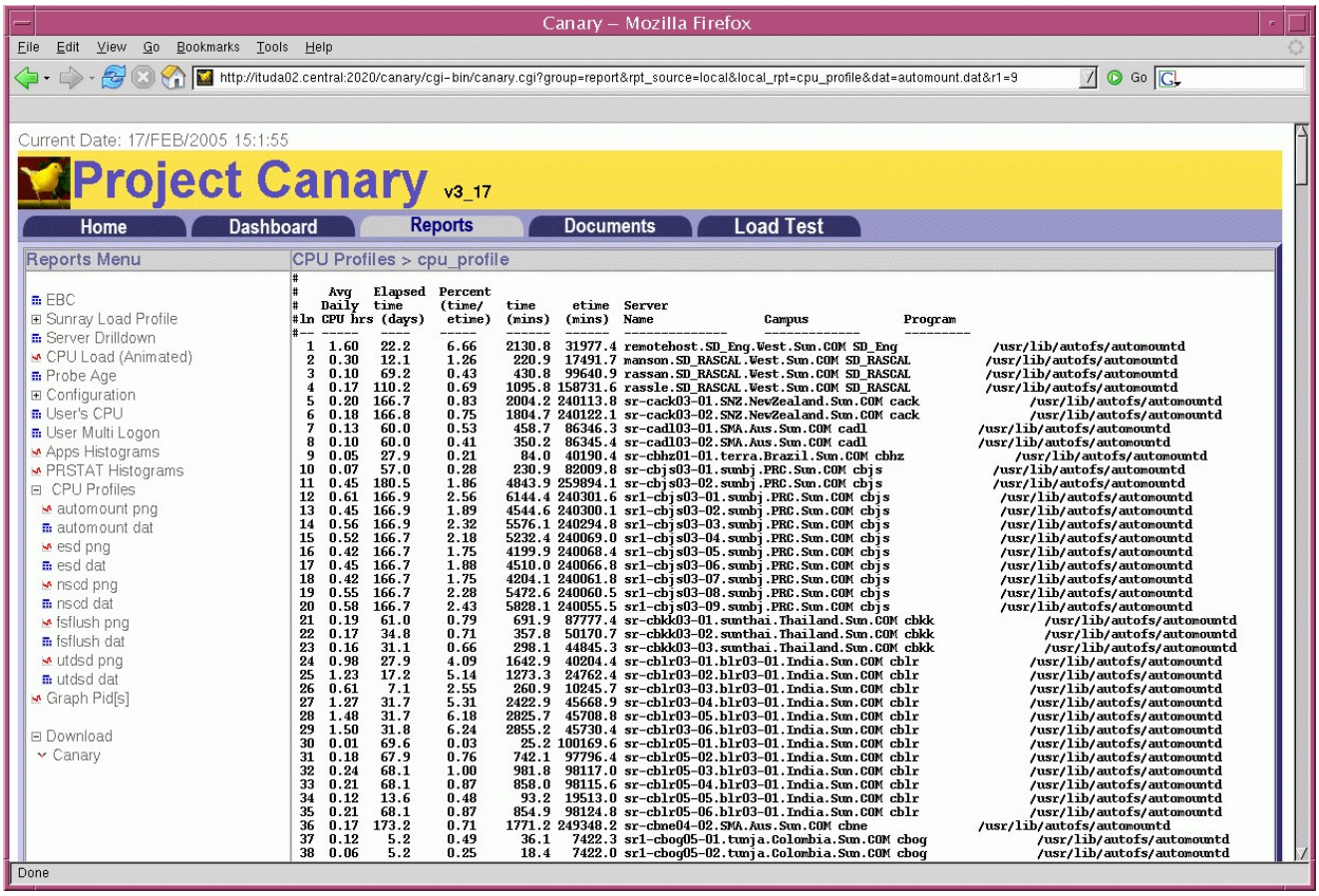
The automount dat file holds the raw data used to draw the graph shown above. The X, or horizontal axis of the top graph is drawn by the "number" assigned to a particular server in the automount dat file and the automount dat file is displayed in numerical order starting with "Server #1" and going through all servers monitored by Project Canary software.

The Y, or vertical axis on the first graph shows the amount of CPU time the `automount(1M)` daemon has consumed on each server.

The second graph shows how long the `automount(1M)` daemon has been running on that particular server. This can be used as a quick measurement of server availability, as generally, the `automount(1M)` daemon is not restarted on a server unless there is a configuration change to the `auto_master` file.

4.12.2: automount dat Report

The automount dat file shown below is the raw data source for the automount png graph:

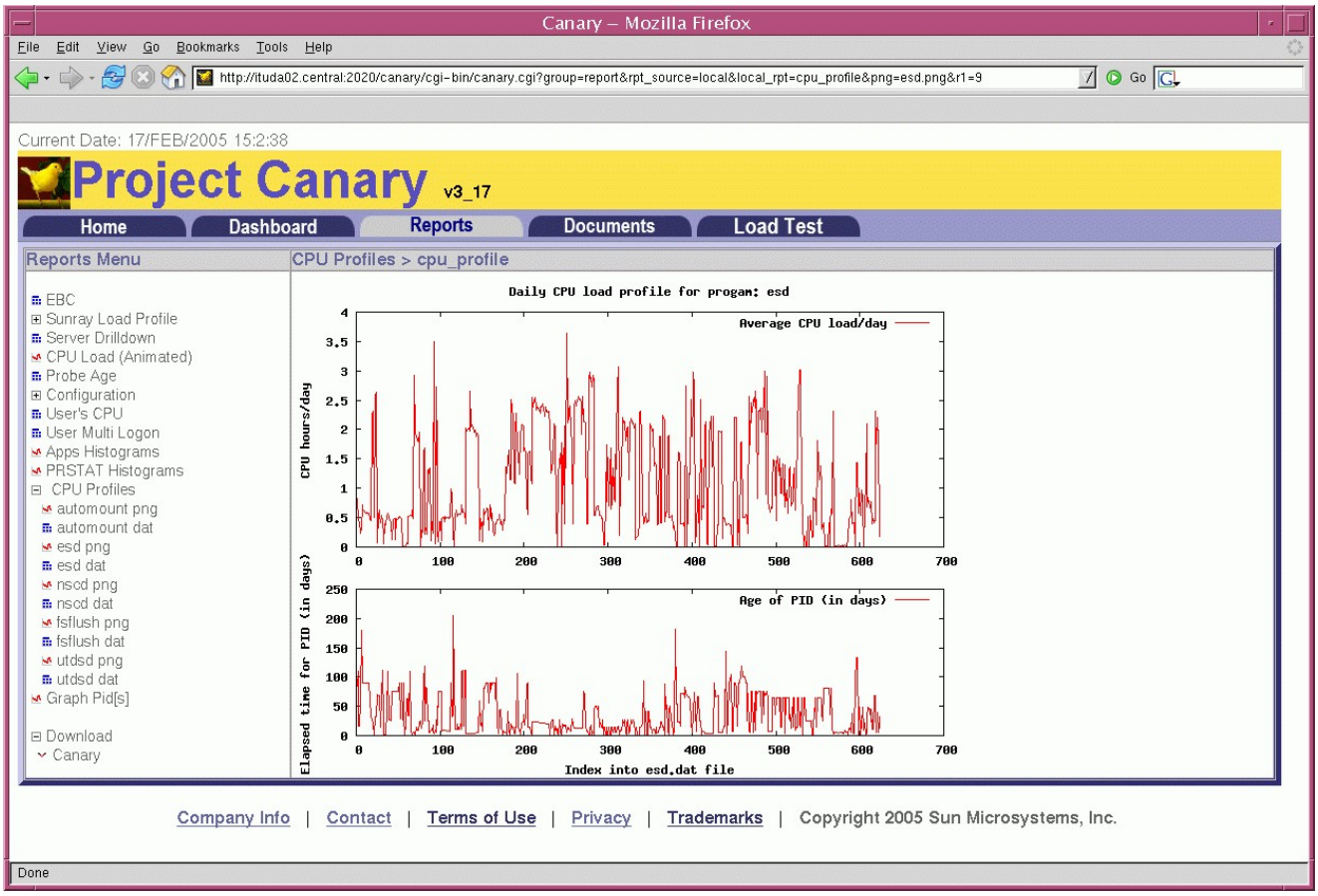


Note that the "domain names" of the servers listed in the above report are alike. This report is useful for not only taking a look at the raw data captured to draw the automount png graph, but also to see if groups of servers are configured alike or not. In the example automount dat file shown above, the raw data shows that different servers in different groupings must be configured differently (or be used in quite different manners) as the data is "alike" for most servers in a single group.

When reading the automount png graph it is a good idea to keep a second browser window open to the automount dat report to compare the graph to the raw data in order to get a better feel for what the graph is trying to say.

4.12.3: esd png Report

The esd png graph shows the CPU utilization for the `esd` daemon across all servers monitored by Project Canary software and is shown below:



As with the automount png graph described earlier in this document, servers are grouped in the esd dat file and plotted on the two graphs shown in the above screen capture.

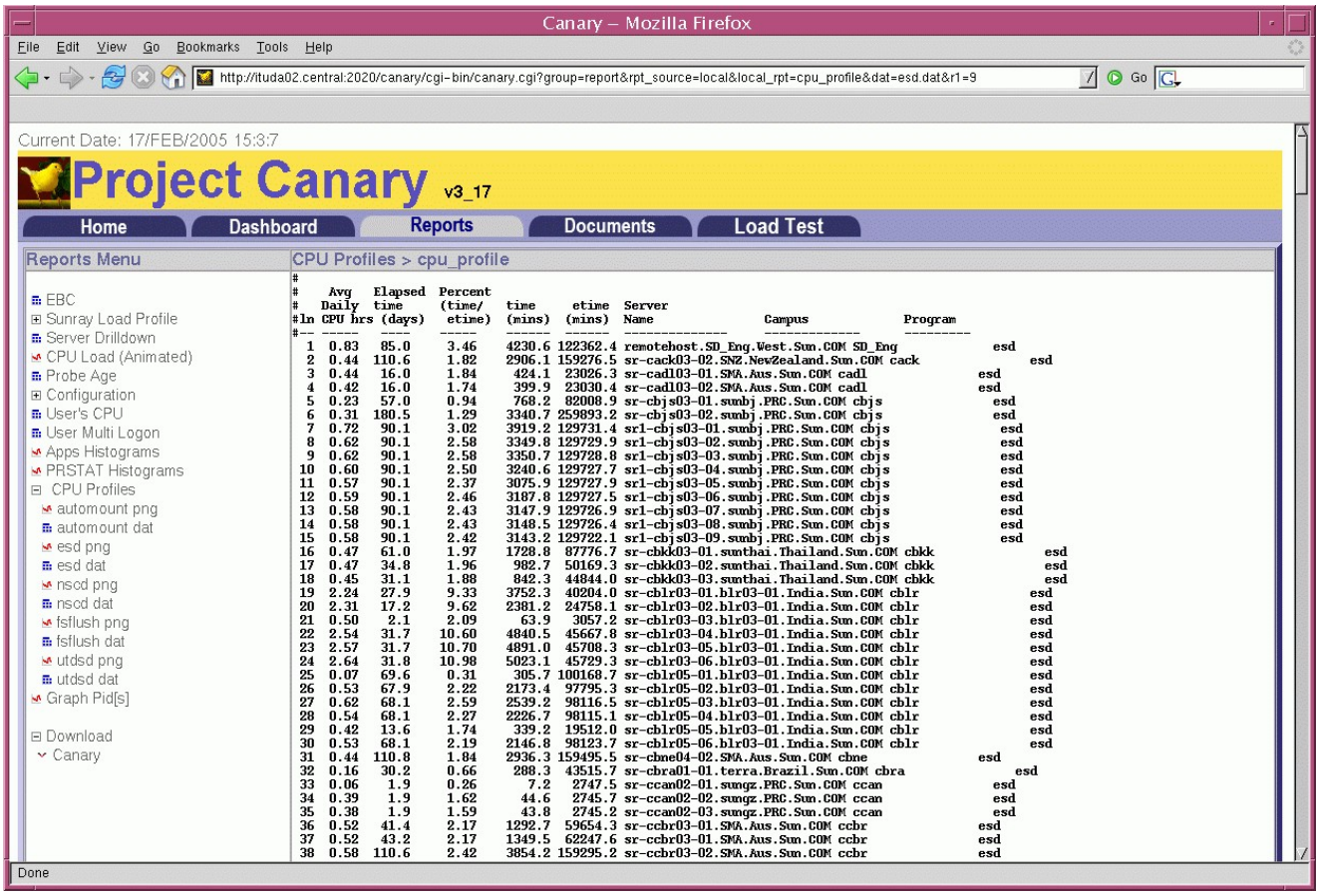
As `esd` is part of the SunMC monitoring suite, this daemon may not have the longevity of `automount(1M)` on a server and may be restarted more often.

Note as well that the graphs shown above appear to have similarly configured servers grouped together.

The esd dat report will confirm or deny this. Again, like the automount png graphical report, it is a good idea to have browser windows open to both the graph and the raw data file so one may be cross-referenced against the other.

4.12.4: esd dat Report

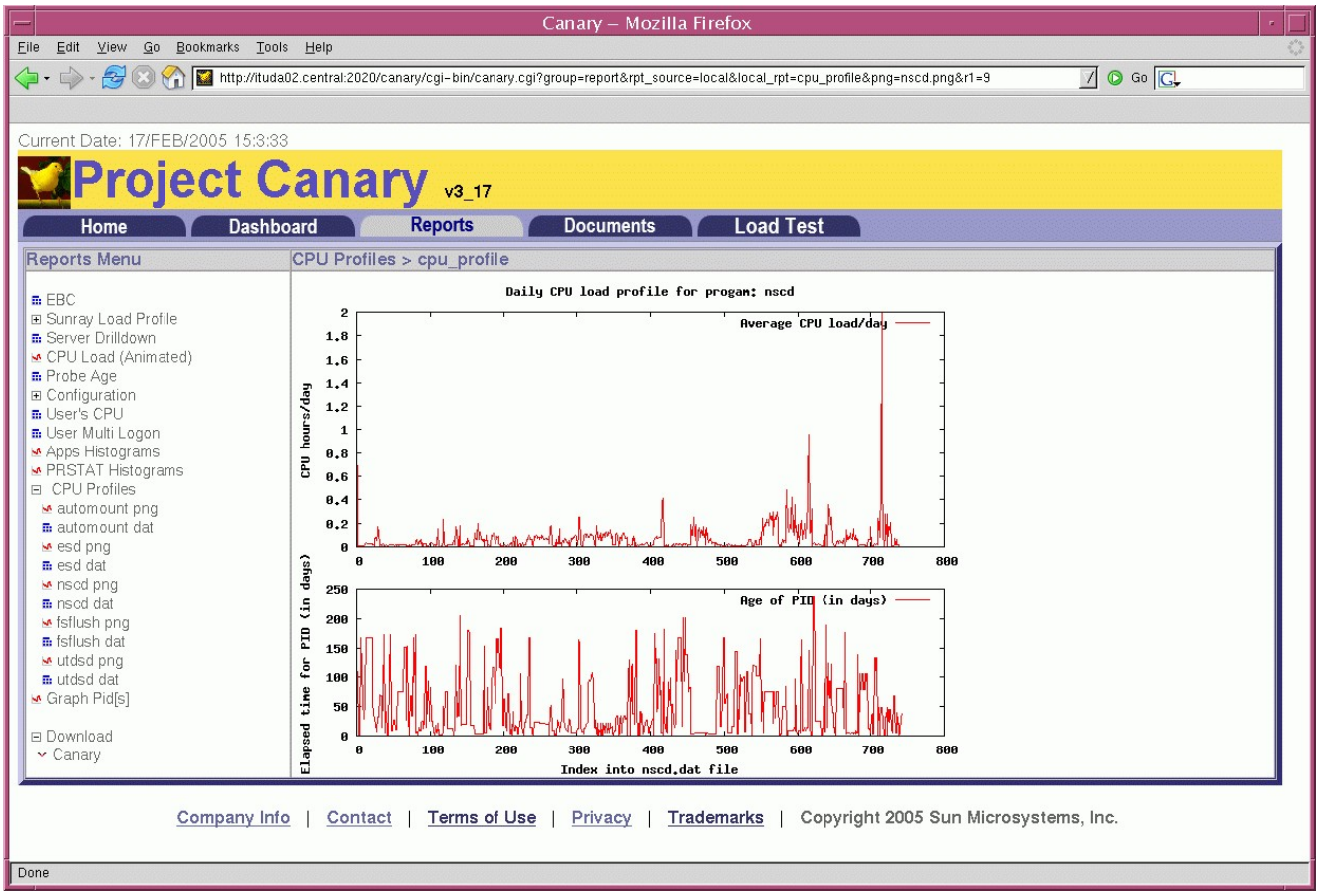
Below, the esd dat report is shown. This is the raw data file used to draw the esd png graphical report.



#ln	Avg Daily CPU hrs	Elapsed time (days)	Percent (time/ etime)	time (mins)	etime (mins)	Server Name	Campus	Program
1	0.83	85.0	3.46	4230.6	122362.4	remotehost.SD_Eng.West.Sun.COM	SD_Eng	esd
2	0.44	110.6	1.82	2906.1	159276.5	sr-cack03-02.SNZ.NewZealand.Sun.COM	cack	esd
3	0.44	16.0	1.84	424.1	23026.3	sr-cadl03-01.SMA.Aus.Sun.COM	cadl	esd
4	0.42	16.0	1.74	399.9	23030.4	sr-cadl03-02.SMA.Aus.Sun.COM	cadl	esd
5	0.23	57.0	0.94	768.2	82008.9	sr-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	esd
6	0.31	180.5	1.29	3340.7	259893.2	sr-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	esd
7	0.72	90.1	3.02	3919.2	129731.4	sr1-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	esd
8	0.62	90.1	2.58	3349.8	129729.9	sr1-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	esd
9	0.62	90.1	2.58	3350.7	129728.8	sr1-cbjs03-03.sunbj.PRC.Sun.COM	cbjs	esd
10	0.60	90.1	2.50	3240.6	129727.7	sr1-cbjs03-04.sunbj.PRC.Sun.COM	cbjs	esd
11	0.57	90.1	2.37	3075.9	129727.9	sr1-cbjs03-05.sunbj.PRC.Sun.COM	cbjs	esd
12	0.59	90.1	2.46	3187.8	129727.5	sr1-cbjs03-06.sunbj.PRC.Sun.COM	cbjs	esd
13	0.58	90.1	2.43	3147.9	129726.9	sr1-cbjs03-07.sunbj.PRC.Sun.COM	cbjs	esd
14	0.58	90.1	2.43	3148.5	129726.4	sr1-cbjs03-08.sunbj.PRC.Sun.COM	cbjs	esd
15	0.58	90.1	2.42	3143.2	129722.1	sr1-cbjs03-09.sunbj.PRC.Sun.COM	cbjs	esd
16	0.47	61.0	1.97	1728.8	87776.7	sr-cbkk03-01.sunthai.Thailand.Sun.COM	cbkk	esd
17	0.47	34.8	1.96	982.7	50169.3	sr-cbkk03-02.sunthai.Thailand.Sun.COM	cbkk	esd
18	0.45	31.1	1.88	842.3	44844.0	sr-cbkk03-03.sunthai.Thailand.Sun.COM	cbkk	esd
19	2.24	27.9	9.33	3752.3	40204.0	sr-cblr03-01.blr03-01.India.Sun.COM	cblr	esd
20	2.31	17.2	9.62	2381.2	24758.1	sr-cblr03-02.blr03-01.India.Sun.COM	cblr	esd
21	0.50	2.1	2.09	63.9	3057.2	sr-cblr03-03.blr03-01.India.Sun.COM	cblr	esd
22	2.54	31.7	10.60	4840.5	45667.8	sr-cblr03-04.blr03-01.India.Sun.COM	cblr	esd
23	2.57	31.7	10.70	4891.0	45708.3	sr-cblr03-05.blr03-01.India.Sun.COM	cblr	esd
24	2.64	31.8	10.98	5023.1	45729.3	sr-cblr03-06.blr03-01.India.Sun.COM	cblr	esd
25	0.07	69.6	0.31	305.7	100168.7	sr-cblr05-01.blr03-01.India.Sun.COM	cblr	esd
26	0.53	67.9	2.22	2173.4	97795.3	sr-cblr05-02.blr03-01.India.Sun.COM	cblr	esd
27	0.62	68.1	2.59	2539.2	98116.5	sr-cblr05-03.blr03-01.India.Sun.COM	cblr	esd
28	0.54	68.1	2.27	2226.7	98115.1	sr-cblr05-04.blr03-01.India.Sun.COM	cblr	esd
29	0.42	13.6	1.74	339.2	19512.0	sr-cblr05-05.blr03-01.India.Sun.COM	cblr	esd
30	0.53	68.1	2.19	2146.8	98123.7	sr-cblr05-06.blr03-01.India.Sun.COM	cblr	esd
31	0.44	110.8	1.84	2936.3	159495.5	sr-cbne04-02.SMA.Aus.Sun.COM	cbne	esd
32	0.16	30.2	0.66	288.3	43515.7	sr-cbra01-01.terra.Brazil.Sun.COM	cbra	esd
33	0.06	1.9	0.26	7.2	2747.5	sr-ccan02-01.sungz.PRC.Sun.COM	ccan	esd
34	0.39	1.9	1.62	44.6	2745.7	sr-ccan02-02.sungz.PRC.Sun.COM	ccan	esd
35	0.38	1.9	1.59	43.8	2745.2	sr-ccan02-03.sungz.PRC.Sun.COM	ccan	esd
36	0.52	41.4	2.17	1292.7	59654.3	sr-ccbr03-01.SMA.Aus.Sun.COM	ccbr	esd
37	0.52	43.2	2.17	1349.5	62247.6	sr-ccbr03-01.SMA.Aus.Sun.COM	ccbr	esd
38	0.58	110.6	2.42	3854.2	159295.2	sr-ccbr03-02.SMA.Aus.Sun.COM	ccbr	esd

4.12.5: ncsd png Report

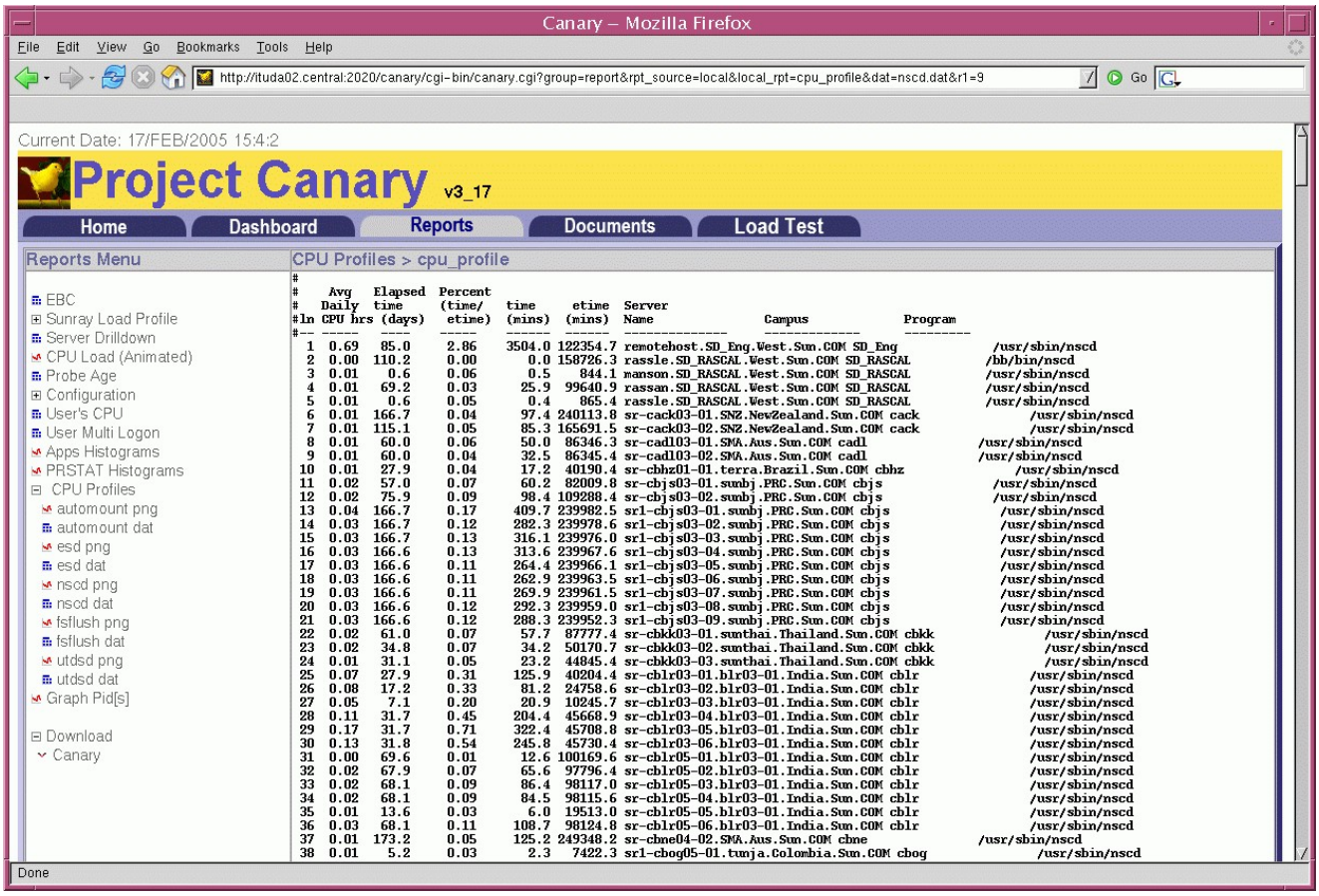
Below is the graphical report for `ncsd`, the name service cache daemon:



Note that while `ncsd` doesn't appear to consume a lot of CPU time and is not restarted very often, there are spikes in the graph showing that `ncsd` had had some problems. These should be checked against in the ncsd dat file and investigated.

4.12.6: ncsd dat

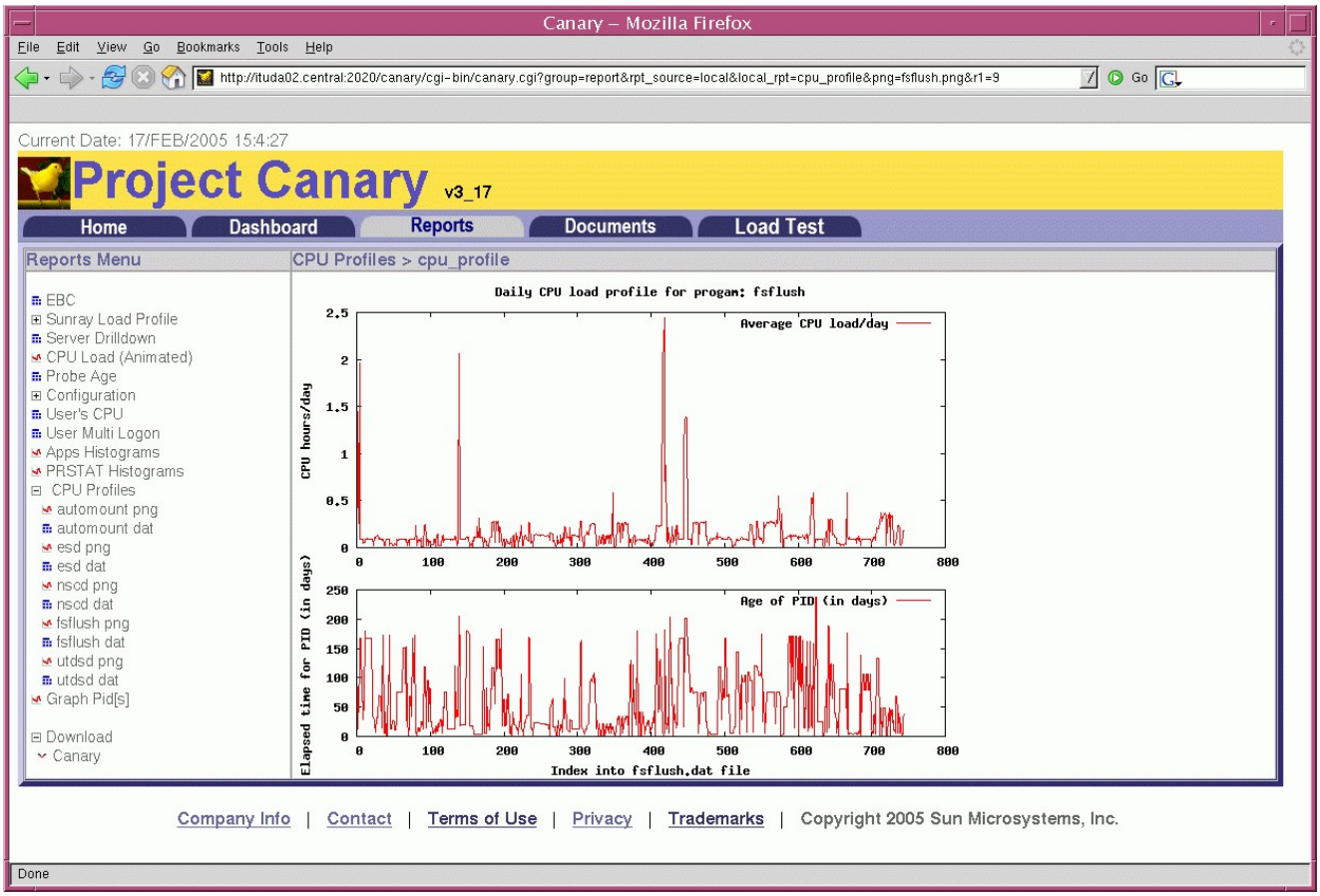
Below is the ncsd dat report, the raw data used to draw the ncsd png graph:



#ln	Avg Daily CPU hrs	Elapsed time (days)	Percent (time/ etime)	time (mins)	etime (mins)	Server Name	Campus	Program
1	0.69	85.0	2.86	3504.0	122354.7	remotehost.SD_Eng.West.Sun.COM	SD_Eng	/usr/sbin/nscd
2	0.00	110.2	0.00	0.0	158726.3	rassle.SD_RASCAL.West.Sun.COM	SD_RASCAL	/bb/bin/nscd
3	0.01	0.6	0.06	0.5	844.1	manson.SD_RASCAL.West.Sun.COM	SD_RASCAL	/usr/sbin/nscd
4	0.01	69.2	0.03	25.9	99640.9	rassan.SD_RASCAL.West.Sun.COM	SD_RASCAL	/usr/sbin/nscd
5	0.01	0.6	0.05	0.4	865.4	rassle.SD_RASCAL.West.Sun.COM	SD_RASCAL	/usr/sbin/nscd
6	0.01	166.7	0.04	97.4	240113.8	sr-cack03-01.SNZ.NewZealand.Sun.COM	cack	/usr/sbin/nscd
7	0.01	115.1	0.05	85.3	165691.5	sr-cack03-02.SNZ.NewZealand.Sun.COM	cack	/usr/sbin/nscd
8	0.01	60.0	0.06	50.0	86346.3	sr-cadl03-01.SMA.Aus.Sun.COM	cadl	/usr/sbin/nscd
9	0.01	60.0	0.04	32.5	86345.4	sr-cadl03-02.SMA.Aus.Sun.COM	cadl	/usr/sbin/nscd
10	0.01	27.9	0.04	17.2	40190.4	sr-cbhz01-01.terra.Brazil.Sun.COM	cbhz	/usr/sbin/nscd
11	0.02	57.0	0.07	60.2	82009.8	sr-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
12	0.02	75.9	0.09	98.4	109288.4	sr-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
13	0.04	166.7	0.17	409.7	239982.5	sr1-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
14	0.03	166.7	0.12	282.3	239978.6	sr1-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
15	0.03	166.7	0.13	316.1	239976.0	sr1-cbjs03-03.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
16	0.03	166.6	0.13	313.6	239967.6	sr1-cbjs03-04.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
17	0.03	166.6	0.11	264.4	239966.1	sr1-cbjs03-05.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
18	0.03	166.6	0.11	262.9	239963.5	sr1-cbjs03-06.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
19	0.03	166.6	0.11	269.9	239961.5	sr1-cbjs03-07.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
20	0.03	166.6	0.12	292.3	239959.0	sr1-cbjs03-08.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
21	0.03	166.6	0.12	288.3	239952.3	sr1-cbjs03-09.sunbj.PRC.Sun.COM	cbjs	/usr/sbin/nscd
22	0.02	61.0	0.07	57.7	87777.4	sr-cbkk03-01.sunthai.Thailand.Sun.COM	cbkk	/usr/sbin/nscd
23	0.02	34.8	0.07	34.2	50170.7	sr-cbkk03-02.sunthai.Thailand.Sun.COM	cbkk	/usr/sbin/nscd
24	0.01	31.1	0.05	23.2	44845.4	sr-cbkk03-03.sunthai.Thailand.Sun.COM	cbkk	/usr/sbin/nscd
25	0.07	27.9	0.31	125.9	40204.4	sr-cblr03-01.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
26	0.08	17.2	0.33	81.2	24758.6	sr-cblr03-02.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
27	0.05	7.1	0.20	20.9	10245.7	sr-cblr03-03.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
28	0.11	31.7	0.45	204.4	45668.9	sr-cblr03-04.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
29	0.17	31.7	0.71	322.4	45708.8	sr-cblr03-05.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
30	0.13	31.8	0.54	245.8	45730.4	sr-cblr03-06.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
31	0.00	69.6	0.01	12.6	100169.6	sr-cblr05-01.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
32	0.02	67.9	0.07	65.6	97796.4	sr-cblr05-02.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
33	0.02	68.1	0.09	86.4	98117.0	sr-cblr05-03.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
34	0.02	68.1	0.09	84.5	98115.6	sr-cblr05-04.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
35	0.01	13.6	0.03	6.0	19513.0	sr-cblr05-05.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
36	0.03	68.1	0.11	108.7	98124.8	sr-cblr05-06.blr03-01.India.Sun.COM	cblr	/usr/sbin/nscd
37	0.01	173.2	0.05	125.2	249348.2	sr-cbne04-02.SMA.Aus.Sun.COM	cbne	/usr/sbin/nscd
38	0.01	5.2	0.03	2.3	7422.3	sr1-cbog05-01.tunja.Colombia.Sun.COM	cbog	/usr/sbin/nscd

4.12.7: fsflush png Report

Below is the graphical report of CPU utilization across all servers for `fsflush`, the filesystem flush daemon:

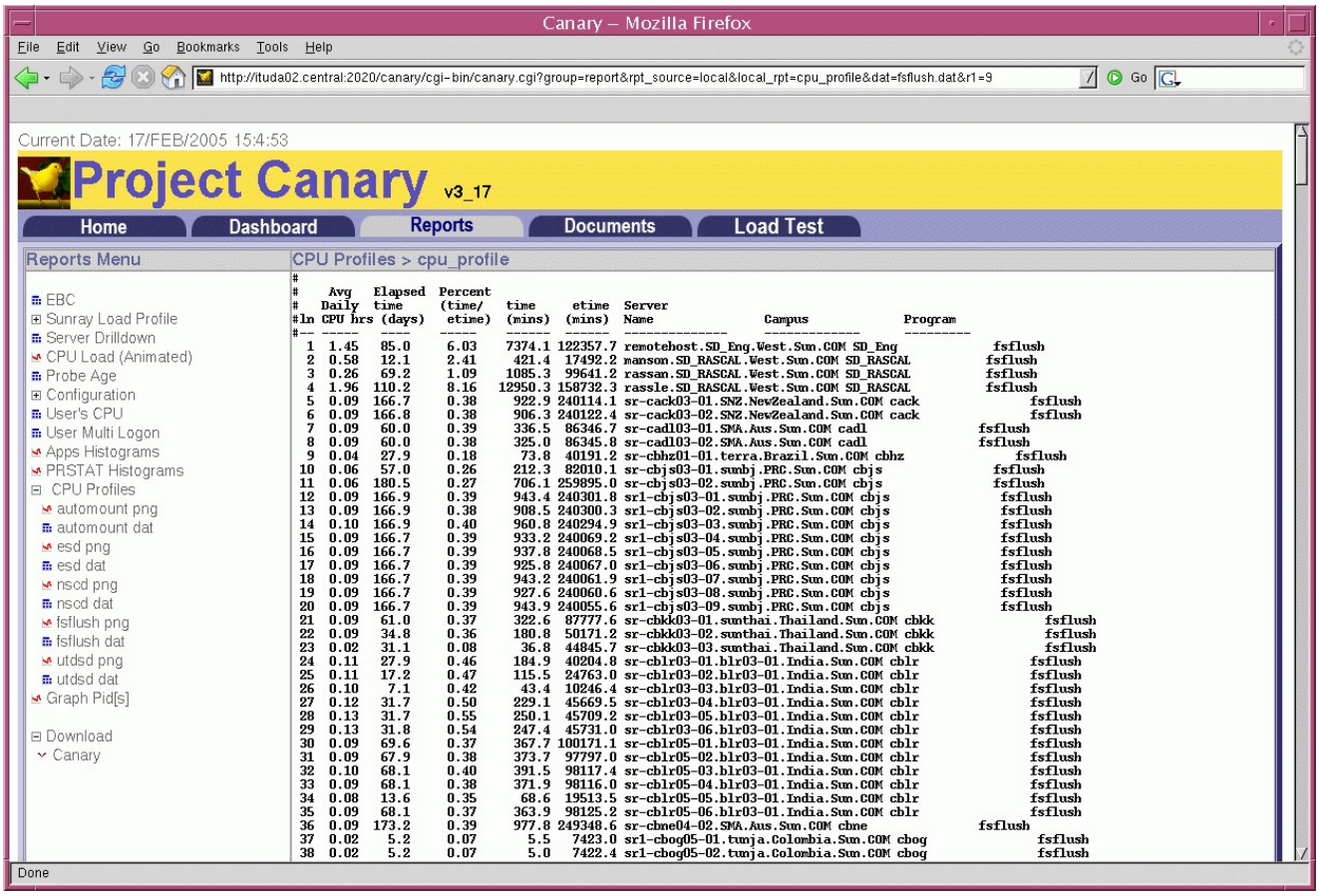


Here on this graph, it is shown that `fsflush` generally is not restarted; usually, `fsflush` is only restarted when a server is rebooted. Also, note that `fsflush` normally does not consume a large amount of CPU time, however, there are spikes in the graph shown in the example.

These spikes show potential problems with the server; perhaps a failing drive where `fsflush` was unable to write to disk or the server might have been I/O bound at the time and unable to write to disk. These spikes should be investigated and corrective action should be taken.

4.12.8: fsflush dat Report

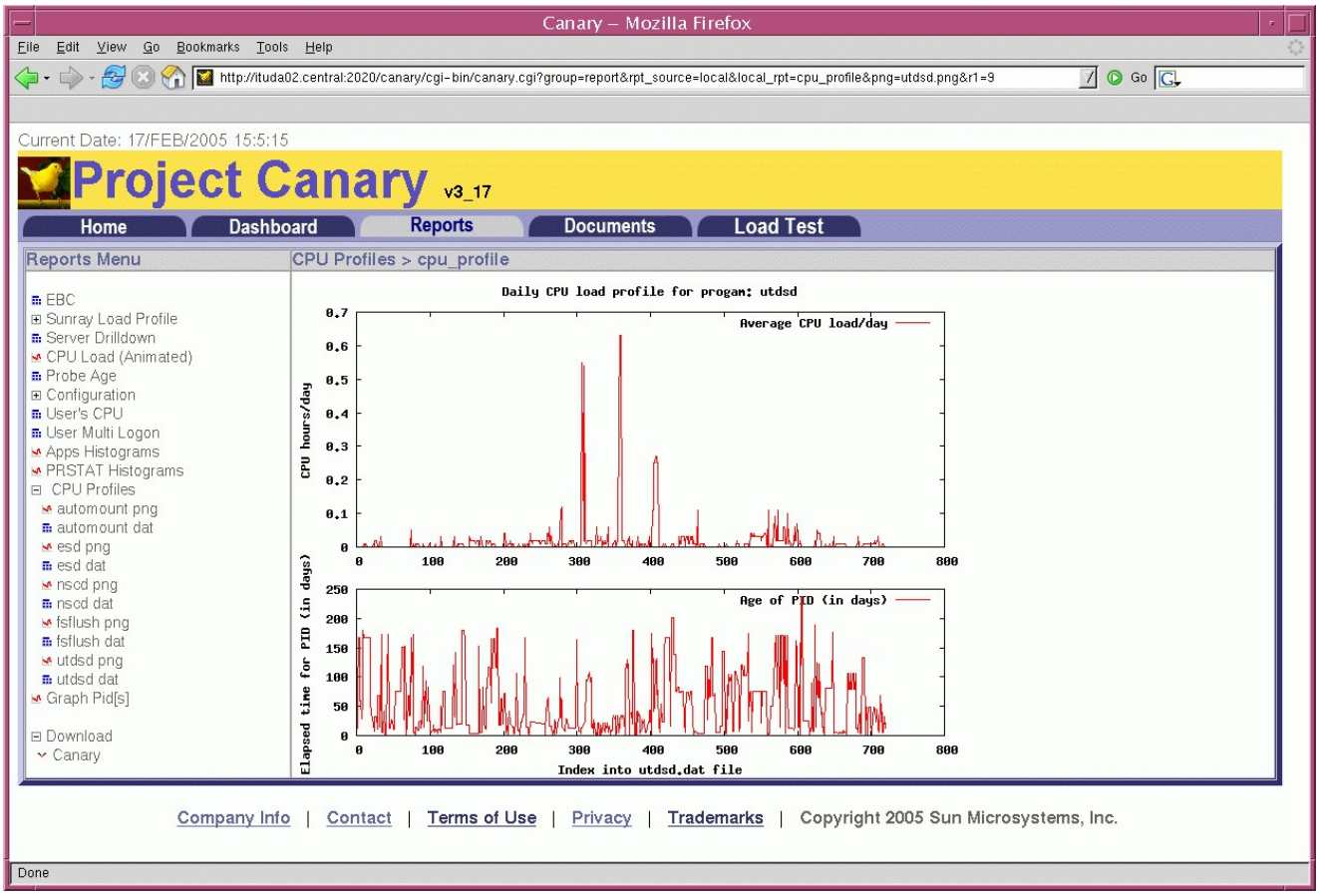
Below is the fsflush dat report, the raw data file used to draw the fsflush png graph shown on the previous page:



#ln	Avg Daily CPU hrs	Elapsed time (days)	Percent (time/ etime)	time (mins)	etime (mins)	Server Name	Campus	Program
1	1.45	85.0	6.03	7374.1	122357.7	remotehost.SD_Eng.West.Sun.COM	SD_Eng	fsflush
2	0.58	12.1	2.41	421.4	17492.2	manson.SD_RASCAL.West.Sun.COM	SD_RASCAL	fsflush
3	0.26	69.2	1.09	1085.3	99641.2	rassan.SD_RASCAL.West.Sun.COM	SD_RASCAL	fsflush
4	1.96	110.2	8.16	12950.3	158732.3	rassle.SD_RASCAL.West.Sun.COM	SD_RASCAL	fsflush
5	0.09	166.7	0.38	922.9	240114.1	sr-cack03-01.SNZ.NewZealand.Sun.COM	cack	fsflush
6	0.09	166.8	0.38	906.3	240122.4	sr-cack03-02.SNZ.NewZealand.Sun.COM	cack	fsflush
7	0.09	60.0	0.39	336.5	86346.7	sr-cadl03-01.SMA.Aus.Sun.COM	cadl	fsflush
8	0.09	60.0	0.38	325.0	86345.8	sr-cadl03-02.SMA.Aus.Sun.COM	cadl	fsflush
9	0.04	27.9	0.18	73.8	40191.2	sr-cbhz01-01.terra.Brazil.Sun.COM	cbhz	fsflush
10	0.06	57.0	0.26	212.3	82010.1	sr-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	fsflush
11	0.06	180.5	0.27	706.1	259895.0	sr-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	fsflush
12	0.09	166.9	0.39	943.4	240301.8	sr1-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	fsflush
13	0.09	166.9	0.38	908.5	240300.3	sr1-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	fsflush
14	0.10	166.9	0.40	960.8	240294.9	sr1-cbjs03-03.sunbj.PRC.Sun.COM	cbjs	fsflush
15	0.09	166.7	0.39	933.2	240069.2	sr1-cbjs03-04.sunbj.PRC.Sun.COM	cbjs	fsflush
16	0.09	166.7	0.39	937.8	240068.5	sr1-cbjs03-05.sunbj.PRC.Sun.COM	cbjs	fsflush
17	0.09	166.7	0.39	925.8	240067.0	sr1-cbjs03-06.sunbj.PRC.Sun.COM	cbjs	fsflush
18	0.09	166.7	0.39	943.2	240061.9	sr1-cbjs03-07.sunbj.PRC.Sun.COM	cbjs	fsflush
19	0.09	166.7	0.39	927.6	240060.6	sr1-cbjs03-08.sunbj.PRC.Sun.COM	cbjs	fsflush
20	0.09	166.7	0.39	943.9	240055.6	sr1-cbjs03-09.sunbj.PRC.Sun.COM	cbjs	fsflush
21	0.09	61.0	0.37	322.6	87777.6	sr-cbkk03-01.sunthai.Thailand.Sun.COM	cbkk	fsflush
22	0.09	34.8	0.36	180.8	50171.2	sr-cbkk03-02.sunthai.Thailand.Sun.COM	cbkk	fsflush
23	0.02	31.1	0.08	36.8	44845.7	sr-cbkk03-03.sunthai.Thailand.Sun.COM	cbkk	fsflush
24	0.11	27.9	0.46	184.9	40204.8	sr-cblr03-01.blr03-01.India.Sun.COM	cblr	fsflush
25	0.11	17.2	0.47	115.5	24763.0	sr-cblr03-02.blr03-01.India.Sun.COM	cblr	fsflush
26	0.10	7.1	0.42	43.4	10246.4	sr-cblr03-03.blr03-01.India.Sun.COM	cblr	fsflush
27	0.12	31.7	0.50	229.1	45669.5	sr-cblr03-04.blr03-01.India.Sun.COM	cblr	fsflush
28	0.13	31.7	0.55	250.1	45709.2	sr-cblr03-05.blr03-01.India.Sun.COM	cblr	fsflush
29	0.13	31.8	0.54	247.4	45731.0	sr-cblr03-06.blr03-01.India.Sun.COM	cblr	fsflush
30	0.09	69.6	0.37	367.7	100171.1	sr-cblr05-01.blr03-01.India.Sun.COM	cblr	fsflush
31	0.09	67.9	0.38	373.7	97797.0	sr-cblr05-02.blr03-01.India.Sun.COM	cblr	fsflush
32	0.10	68.1	0.40	391.5	98117.4	sr-cblr05-03.blr03-01.India.Sun.COM	cblr	fsflush
33	0.09	68.1	0.38	371.9	98116.0	sr-cblr05-04.blr03-01.India.Sun.COM	cblr	fsflush
34	0.08	13.6	0.35	68.6	19513.5	sr-cblr05-05.blr03-01.India.Sun.COM	cblr	fsflush
35	0.09	68.1	0.37	363.9	98125.2	sr-cblr05-06.blr03-01.India.Sun.COM	cblr	fsflush
36	0.09	173.2	0.39	977.8	249348.6	sr-cbne04-02.SMA.Aus.Sun.COM	cbne	fsflush
37	0.02	5.2	0.07	5.5	7423.0	sr1-cbog05-01.tunja.Colombia.Sun.COM	cbog	fsflush
38	0.02	5.2	0.07	5.0	7422.4	sr1-cbog05-02.tunja.Colombia.Sun.COM	cbog	fsflush

4.12.9: utdsd png Report

This is the graphical report for CPU utilization for the `utdsd` daemon, part of the Sun Ray Server Software package. `utdsd` is the Sun Ray Data Store Daemon.

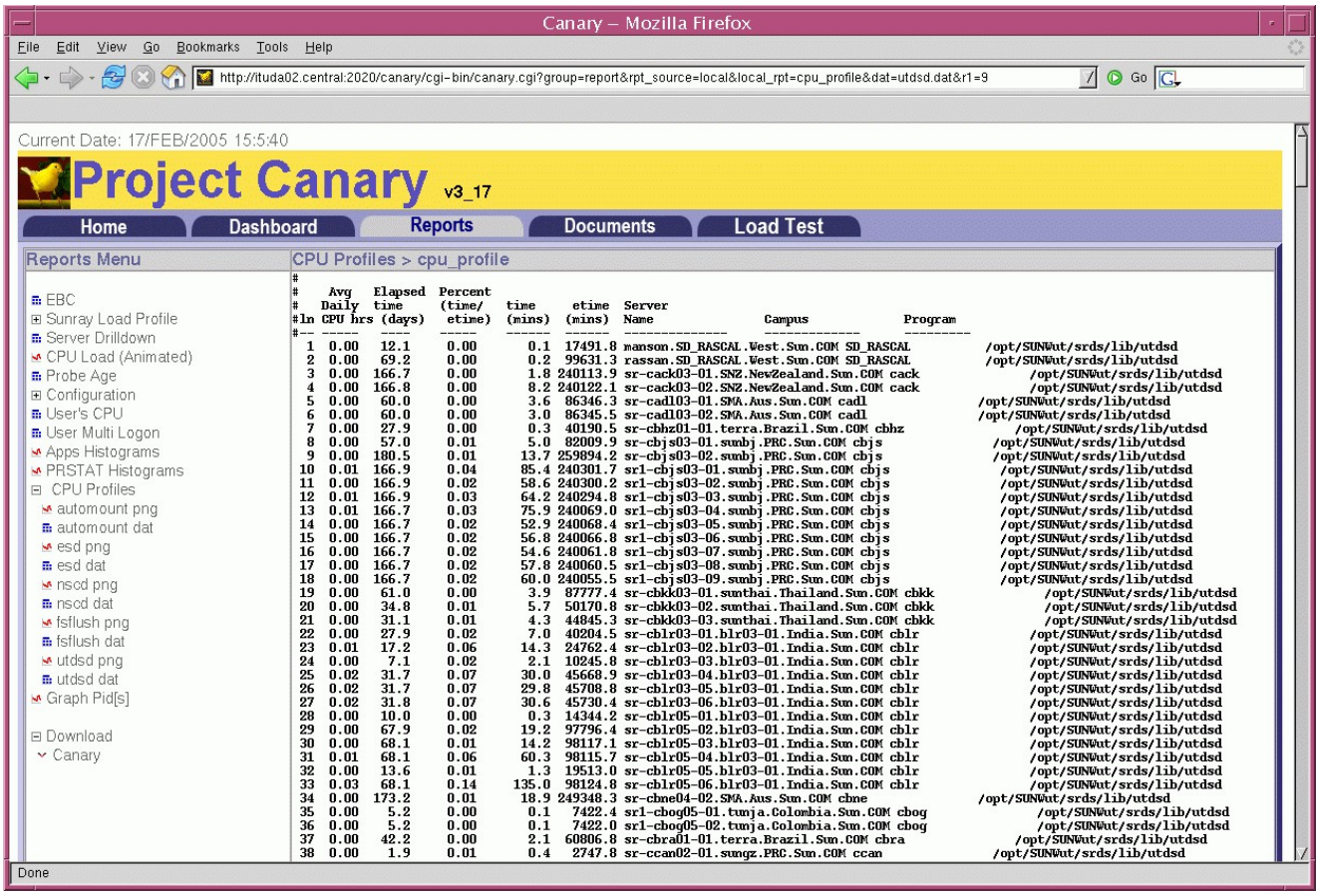


`utdsd`, like other system service daemons, does not usually consume a lot of CPU time. However, like other system services, occasionally a problem can crop up which causes `utdsd` to have problems. Note the spikes in the graph in the CPU Profiles report shown above. These are not normal and should be investigated by examining the utdsd dat report.

An example of the utdsd dat report is shown on the next page.

4.12.10: utdsd dat Report

Following is the utdsd dat report, the raw data file used to draw the utdsd graph shown on the previous page of this *User Guide and Manual*.



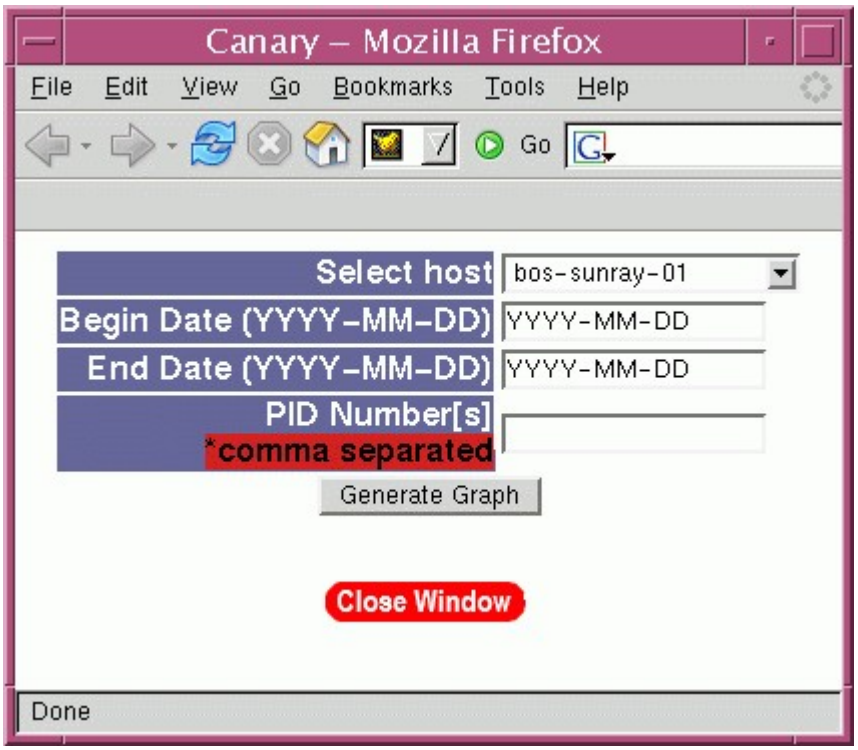
#	Avg Daily CPU hrs	Elapsed time (days)	Percent (time/ etime)	time (mins)	etime (mins)	Server Name	Campus	Program
1	0.00	12.1	0.00	0.1	17491.8	manson.SD_RASCAL.West.Sun.COM	SD_RASCAL	/opt/SUNWut/srds/lib/utdsd
2	0.00	69.2	0.00	0.2	99631.3	rassan.SD_RASCAL.West.Sun.COM	SD_RASCAL	/opt/SUNWut/srds/lib/utdsd
3	0.00	166.7	0.00	1.8	240113.9	sr-cack03-01.SNZ.NewZealand.Sun.COM	cack	/opt/SUNWut/srds/lib/utdsd
4	0.00	166.8	0.00	8.2	240122.1	sr-cack03-02.SNZ.NewZealand.Sun.COM	cack	/opt/SUNWut/srds/lib/utdsd
5	0.00	60.0	0.00	3.6	86346.3	sr-cadl03-01.SMA.Aus.Sun.COM	cadl	/opt/SUNWut/srds/lib/utdsd
6	0.00	60.0	0.00	3.0	86345.5	sr-cadl03-02.SMA.Aus.Sun.COM	cadl	/opt/SUNWut/srds/lib/utdsd
7	0.00	27.9	0.00	0.3	40190.5	sr-cbhz01-01.terra.Brazil.Sun.COM	cbhz	/opt/SUNWut/srds/lib/utdsd
8	0.00	57.0	0.01	5.0	82009.9	sr-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
9	0.00	180.5	0.01	13.7	259894.2	sr-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
10	0.01	166.9	0.04	85.4	240301.7	sr1-cbjs03-01.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
11	0.00	166.9	0.02	58.6	240300.2	sr1-cbjs03-02.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
12	0.01	166.9	0.03	64.2	240294.8	sr1-cbjs03-03.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
13	0.01	166.7	0.03	75.9	240069.0	sr1-cbjs03-04.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
14	0.00	166.7	0.02	52.9	240068.4	sr1-cbjs03-05.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
15	0.00	166.7	0.02	56.8	240066.8	sr1-cbjs03-06.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
16	0.00	166.7	0.02	54.6	240061.8	sr1-cbjs03-07.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
17	0.00	166.7	0.02	57.8	240060.5	sr1-cbjs03-08.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
18	0.00	166.7	0.02	60.0	240055.5	sr1-cbjs03-09.sunbj.PRC.Sun.COM	cbjs	/opt/SUNWut/srds/lib/utdsd
19	0.00	61.0	0.00	3.9	87777.4	sr-cbkk03-01.sunthai.Thailand.Sun.COM	cbkk	/opt/SUNWut/srds/lib/utdsd
20	0.00	34.8	0.01	5.7	50170.8	sr-cbkk03-02.sunthai.Thailand.Sun.COM	cbkk	/opt/SUNWut/srds/lib/utdsd
21	0.00	31.1	0.01	4.3	44845.3	sr-cbkk03-03.sunthai.Thailand.Sun.COM	cbkk	/opt/SUNWut/srds/lib/utdsd
22	0.00	27.9	0.02	7.0	40204.5	sr-cblr03-01.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
23	0.01	17.2	0.06	14.3	24762.4	sr-cblr03-02.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
24	0.00	7.1	0.02	2.1	10245.8	sr-cblr03-03.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
25	0.02	31.7	0.07	30.0	45668.9	sr-cblr03-04.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
26	0.02	31.7	0.07	29.8	45708.8	sr-cblr03-05.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
27	0.02	31.8	0.07	30.6	45730.4	sr-cblr03-06.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
28	0.00	10.0	0.00	0.3	14344.2	sr-cblr05-01.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
29	0.00	67.9	0.02	19.2	97796.4	sr-cblr05-02.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
30	0.00	68.1	0.01	14.2	98117.1	sr-cblr05-03.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
31	0.01	68.1	0.06	60.3	98115.7	sr-cblr05-04.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
32	0.00	13.6	0.01	1.3	19513.0	sr-cblr05-05.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
33	0.03	68.1	0.14	135.0	98124.8	sr-cblr05-06.blr03-01.India.Sun.COM	cblr	/opt/SUNWut/srds/lib/utdsd
34	0.00	173.2	0.01	18.9	249348.3	sr-cbne04-02.SMA.Aus.Sun.COM	cbne	/opt/SUNWut/srds/lib/utdsd
35	0.00	5.2	0.00	0.1	7422.4	sr1-cbog05-01.tunja.Colombia.Sun.COM	cbog	/opt/SUNWut/srds/lib/utdsd
36	0.00	5.2	0.00	0.1	7422.0	sr1-cbog05-02.tunja.Colombia.Sun.COM	cbog	/opt/SUNWut/srds/lib/utdsd
37	0.00	42.2	0.00	2.1	60806.8	sr-cbra01-01.terra.Brazil.Sun.COM	cbra	/opt/SUNWut/srds/lib/utdsd
38	0.00	1.9	0.01	0.4	2747.8	sr-ccan02-01.sungz.PRC.Sun.COM	ccan	/opt/SUNWut/srds/lib/utdsd

Like the other raw data files described in this section of the *User Guide and Manual*, the utdsd dat file numbers servers, shows the CPU usage of the process Project Canary software is monitoring and graphing, and groups machines according to entries in the `server_to_geo.cfg` file described earlier in this manual and also in the *Project Canary Installation and Configuration Guide*.

4.13: Graph PID[s] Report

The Graph PID[s] report displays a web form where one can graph process ID numbers on a per-host basis dynamically. This can be useful for tracking long-running jobs.

Clicking on the Graph PID[s] link brings up another window with a menu that looks like this:



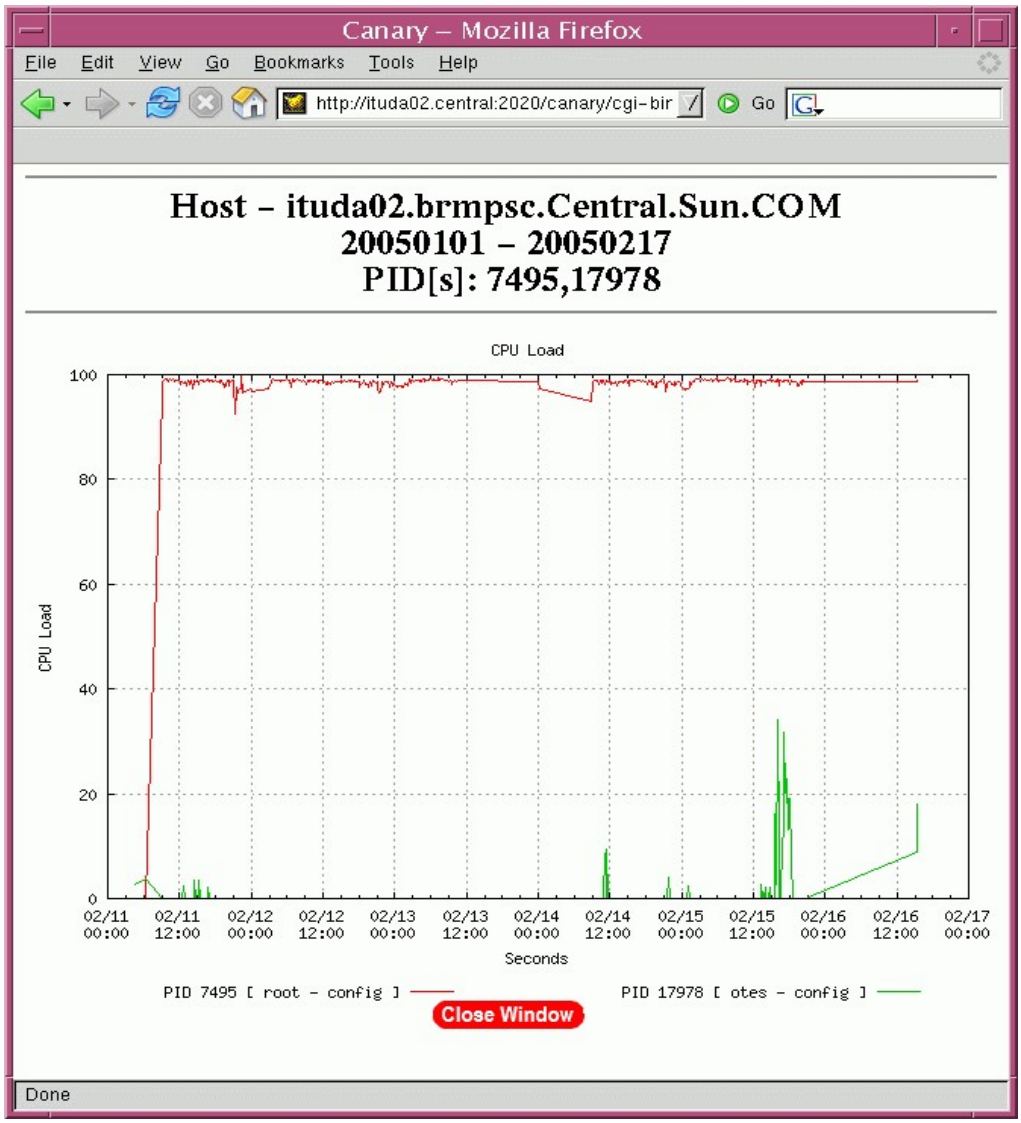
The main purpose of this report is to take a closer look at a long-running process to see exactly how much CPU time that process has used.

The upper limit of the number of processes Project Canary software can record data on is approximately one million processes per day and all of those can be graphed at the same time. There are a total of 460 million possible combinations of process ID numbers that can be graphed together. Over a five week period, between 10,000,000 and 100,000,000 processes are monitored and data stored by Project Canary software. This is a good tool for examining long-running processes on a single host.

To examine a process, first select the host the process either was running on or is running on using the drop-down menu at the top of the page. This drop-down menu will list every machine Project Canary software is monitoring.

Then, select the dates to graph data for the process to be examined and then the processes to be examined. Clicking on the "Generate Graph" button will generate a graph of the process(es) to be examined.

An example of two processes being graphed is shown on the next page:
Below is an example of two processes being graphed using the Graph PID[s] report function:



Each process is graphed in a different color; in this example, the colors are red and green. Note that one of the processes is producing a large amount of CPU load and the other process occasionally spikes, but otherwise consumes very little CPU time.

The process that is shown with the red line is an `apache(1M)` server process and the process shown with the green line is a database server process. One can infer from this graph that this is a heavily used web server and that it has occasional demands on a back-end database. The graph was drawn from January 01, 2005 to February 17, 2005 and graphs both processes.

When there is no data available before a certain date, Project Canary software draws the graph from the earliest available date. In this case, the earliest date with data available to graph for the web server process was February 11, 2005.

Chapter 5: Documents Tab

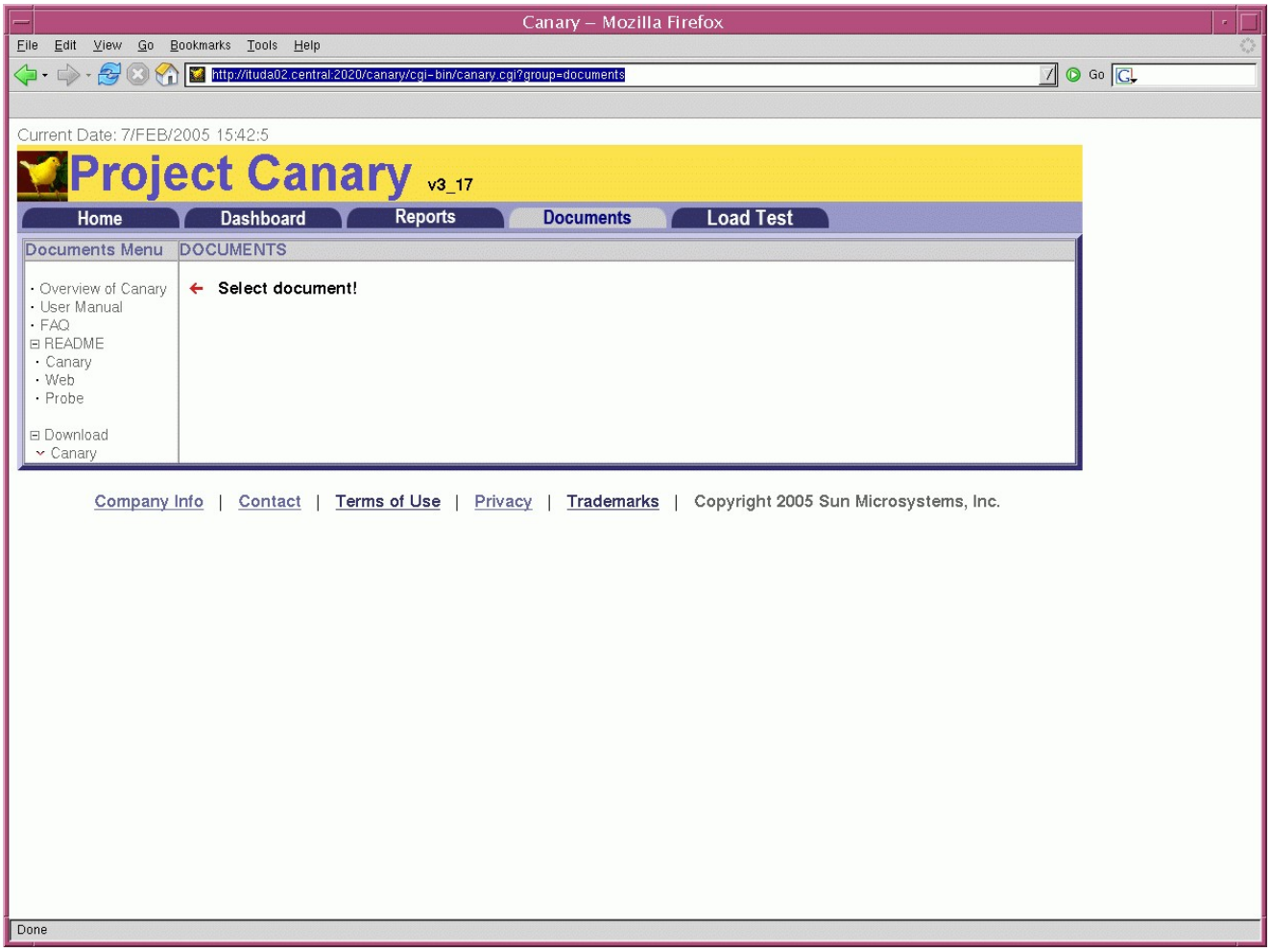
The Documents Tab takes one to a page where all documentation for Project Canary software is available. Currently, these documents can be downloaded from this tab:

Overview of Project Canary Software
Project Canary User Guide and Manual
Project Canary Frequently Asked Questions

Three README files:

Project Canary Software README
Web Server Configuration README
Project Canary Probe Software README

The Documents Tab screen is shown below:



All documentation for Project Canary software is either read from or downloaded from the Documents Tab.

Appendix 1: Glossary

automount(1): utility which installs automatic mount points
browser: program used to view web pages
canary: a small yellow songbird

ftp(1):	file transfer program
load:	the work a computer is doing, also, the number of processes asking for CPU time at a given moment (the load average)
loopback:	a virtual network interface where packets "loop back" to that particular machine
mailx(1):	a simple mail user agent
scp(1):	secure copy (remote file copy program)
Sun Ray:	a "thin client" that has no CPU and relies on a back-end server for processing power and resources

Appendix 2: Frequently Asked Questions

A2.1: What is the difference between Project Canary and SunMC?

SunMC mostly monitors hardware. It also allows one to manipulate the machine being monitored from the SunMC console. Project Canary measures the load generated by software on a particular machine. One cannot manipulate client machines from the Project Canary web page. One must make any changes on the client directly. Project Canary is a monitoring package only, not a full-featured

management system.

A2.2: I don't want to install any software on clients I wish to monitor.

There are ways to configure the software so that there is no need to install software on client systems. This approach has some performance and security concerns but will function adequately in scenarios where one is monitoring a small number of systems. Instructions on how to do this are detailed in the *Project Canary Installation and Configuration Guide*.

A2.3: I killed all the runaway programs but the CPU load indicator is still red. Why?

It takes a while for Project Canary software to cycle through its update/processing cycle. One should see the indicator change within five to ten minutes if the change fixed the problem. The graphs take somewhat longer to update and may require up to 30 minutes to be redrawn with new data.

A2.4: Is this just a tool for monitoring Sun Ray servers?

Project Canary was designed and "tuned" to monitor Sun Ray servers in particular, however, it can be used to monitor the performance of any Solaris or Linux machine. With some modification, Project Canary could be extended to run on HP-UX and AIX as well as any other Unix variant.

A2.5: What is the significance of the number of users logged into multiple Sun Ray servers?

Due to some software issues currently being addressed, one could log out of a Sun Ray session but not have all of one's processes terminated upon logout. These left-over processes wind up consuming CPU cycles and producing additional load on the Sun Ray server. Users are generally unaware that this happens. The multiple Sun Ray user report helps a systems administrator as well as a user to see how many different Sun Ray servers he/she is logged into.

A2.6: Why show anything dealing with the automounter?

If the `-nobrowse` option is not set in `/etc/init.d/autofs` a server could inadvertently be mounting many directories that are not needed. Compare the graphs for a system during the week and then over the weekend. If the CPU load and number of users doesn't drop to around zero, then there are other issues to investigate. Look for abnormal network traffic on either the individual network cards or on the TCP Statistics Graph.

Index