

# Monitoring shared processor pools with lpar2rrd

## Monitor and report on shared processor usage on IBM Power Systems

Chris Gibson

May 18, 2010

This article describes how to use the lpar2rrd tool to monitor and report on shared processor pool usage. It is based on a customer's experience with the tool.

### Introduction

In this article, I will share with you some of the benefits of using the free **lpar2rrd** tool. After using this for several years, I have come to rely on it as an excellent monitoring and reporting tool for shared processor pools on IBM Power based systems. I will briefly describe the lpar2rrd tool and illustrate how I implemented it to monitor the shared processor pool across four Power6 systems.

I love this tool as it allows me to quickly review and report on shared processor pool usage across several systems. Since I started using it almost 3 years ago, I have tried to spread the word about this amazing tool. I mention it to anyone who is willing to listen and I am surprised by how many people are not aware of this tool and what it can do.

I am hoping this article will encourage others to take it up in their environment. Perhaps one day it may be officially supported by IBM, just like the **nmon** utility. One can only hope.

### Overview of lpar2rrd

The lpar2rrd tool has been available for several years now (going back to 2006). When I first looked at it, I liked the idea of what it promised, such as monitoring and reporting for a shared processor pool. However, the tool was still in its infancy, and I had a few issues getting it up and running. When I looked at it 12 months later, things had certainly changed. It had improved substantially, and I was able to get the tool up and running very quickly (in about 30 minutes).

I was amazed by the information it was able to provide. When I considered how I had previously been monitoring and reporting on my shared processor pool, this was a breath of fresh air.

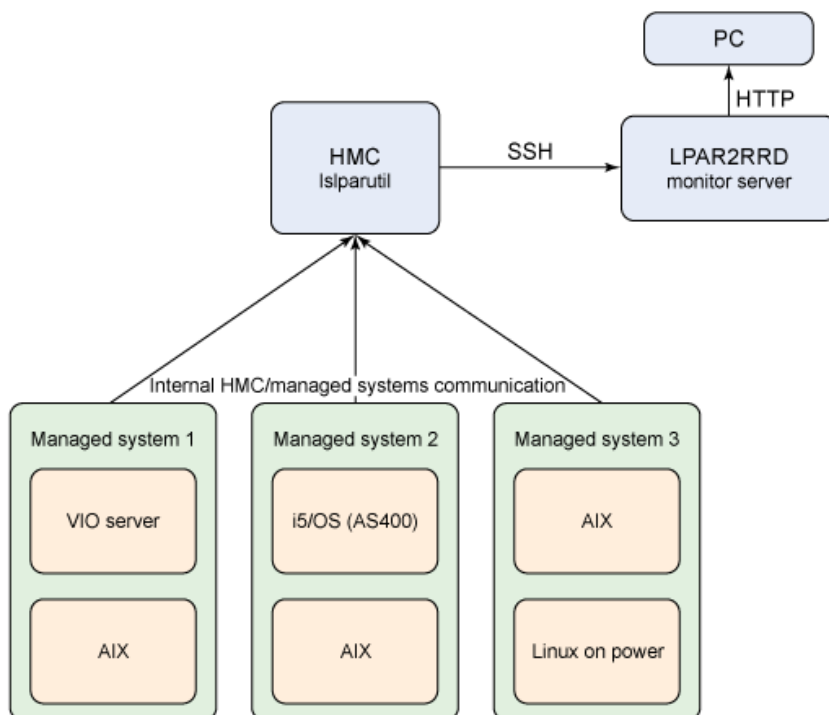
We can thank [Pavel Hampl](#), an IBM employee in the Czech Republic, for this amazing project. He is the creator of the tool, and he has done a great job of pulling it altogether and continuing to develop and improve this utility.

The lpar2rrd tool is capable of collecting historical CPU utilization data for shared processor partitions and systems. It is intended for use with HMC attached, micro-partitioned systems with shared processor pools. It is agentless, so there is no need to install an agent on each LPAR.

Utilization data is collected from the HMC (via [lsparutil](#)). The lpar2rrd utility connects to the nominated HMC(s), via SSH, to collect the performance data. This data is stored in an RRD database on the lpar2rrd system. The data is processed and graphs are generated for shared processor usage for each LPAR and the shared processor pools.

The graphs are then accessible via a Web server (HTTP) running on the lpar2rrd server system. It is a very straightforward architecture, which is detailed in the following diagram (Figure 1) from the lpar2rrd website.

**Figure 1. lpar2rrd architecture overview**



Data is collected at regular intervals (which you can decide). Historical data is available and can provide you with a great view of an LPAR, or an entire system, shared processor usage over days, months, or years.

When you are ready to install the tool, I recommend you visit the [lpar2rrd site](#), download the latest distribution of the code and then follow the detailed installation instructions.

Make sure you have nominated a system that will act as the lpar2rrd server and that you have installed all the pre-requisites. Also, ensure you have a functioning Web server on this system. The installation notes will walk you through the "lpar2rrd to HMC" SSH communications setup.

The lpar2rrd server can be hosted on any UNIX type system, as all that is required is SSH, a Web server, Perl and RRDTool. For me, it made sense to host lpar2rrd on one of my AIX LPARs, in particular one of my NIM servers.

I have used lpar2rrd since version 1.83 and just recently, I upgraded to version 2.51. Version 2 has many new features such as support for *multiple shared processor pools*, not to mention an improved Web front-end. The upgrade from 1.83 to 2.51 was painless. Pavel's documentation provided simple instructions for performing the update.

The very latest version, 2.59 (as of 02/26/2010) now has support for LEA/HEA network statistics, IVM support, and real-time refresh of LPAR/pool usage graphs. And according to the wiki, there are also future plans to support the very latest technologies on Power such as *Active Memory Sharing*.

Aside from CPU statistics, the tool can also provide data relating to memory allocation and LPAR configuration and state change history.

IBM does not officially support this tool. However, if you have a genuine issue with lpar2rrd, you could contact its creator via e-mail and ask for assistance. For example, if you review the release notes on the lpar2rrd site, you will find this entry:

```
1.85 (16-May-2008)
fixed a problem with LANG variable on HMC set to other than en_US with date format
different from MM/DD/YYYY
```

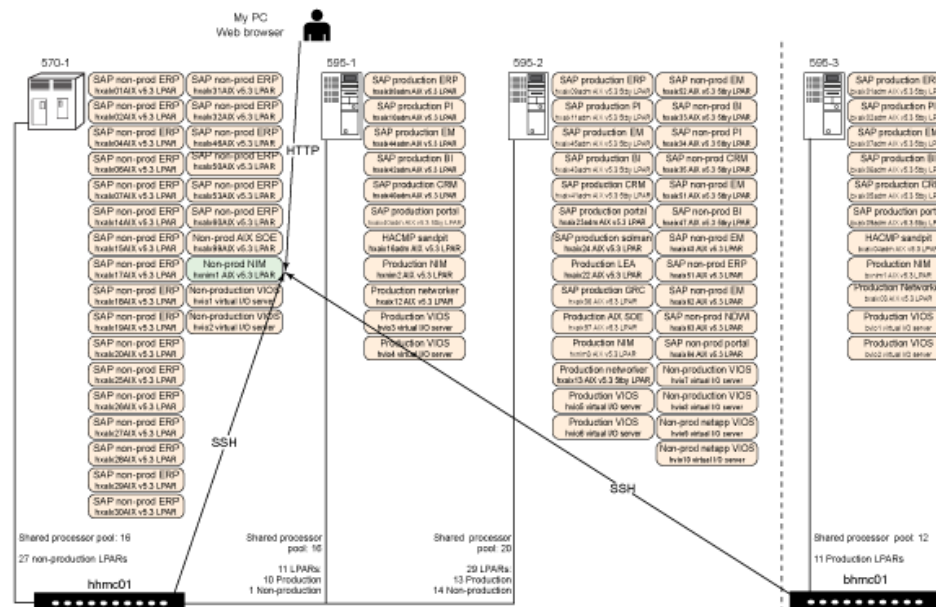
That was me! I had an issue with the tool and I contacted Pavel. He responded very quickly with a resolution to my problem and then set about updating lpar2rrd with a permanent fix. Keep in mind that this is a project that is only conducted during "free time" and is not sponsored by IBM. So please do not inundate the developer with e-mail's requesting support!

## Using lpar2rrd

In my environment, we have four IBM POWER6 shared processor pools that need to be monitored. We have one p6 570 (9117-MMA) and three p6 595s (9119-FHA). Each system has a shared processor pool. We use lpar2rrd tool to collect usage data for each of the frames and then use the resulting graphs for reporting on overall processor pool usage. We also use it to monitor and report on individual LPAR (shared) processor usage.

In Figure 2, there are two HMCs from which lpar2rrd is collecting utilization data. We have an HMC (hhmc01) at one site connected to 570-1, 595-1 and 595-2. The other HMC (bhmc01) is connected to 595-3, located at our disaster recovery site.

**Figure 2. The customers POWER6 environment**

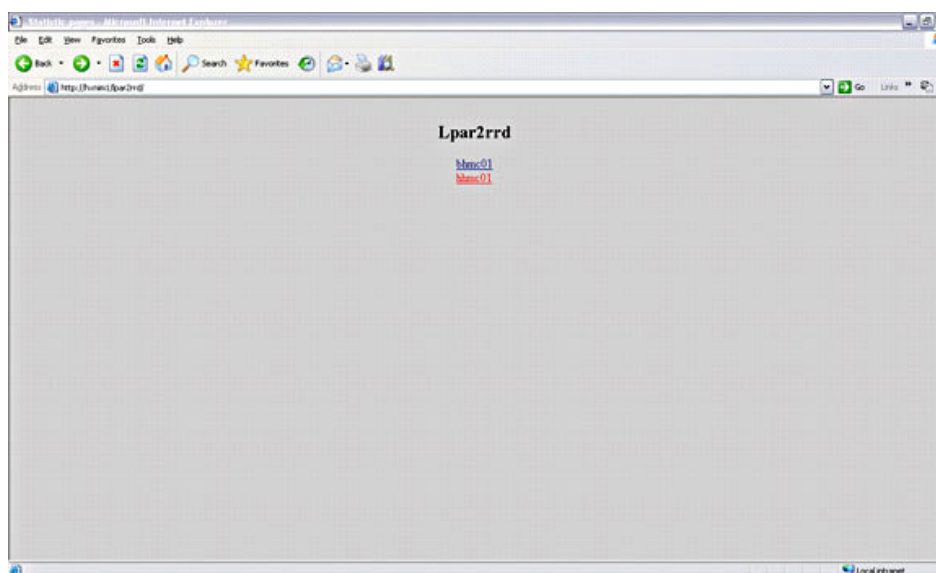


To give you an idea of how we use lpar2rrd and the benefits it provided to us, I have included some screen shots.

The lpar2rrd code is installed on one of our NIM masters (hxnim1) on 570-1. Apache is also installed on the NIM master. To access the tool, I simply point my Web browser at the URL, <http://hxnim1/lpar2rrd>.

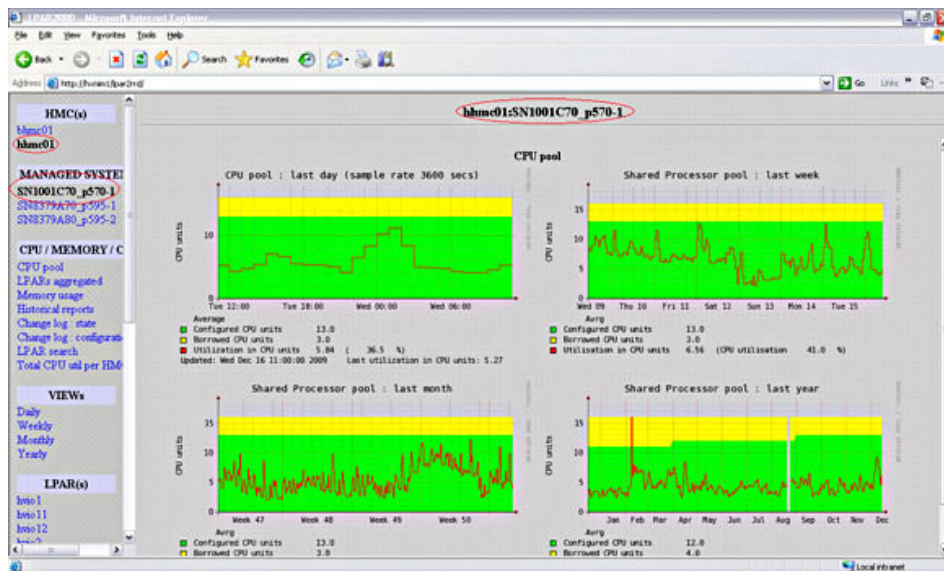
The lpar2rrd main page provides a link to both HMCs (Figure3).

**Figure 3. – The lpar2rrd main page**



After selecting an HMC (in this case hhmc01), I am presented with some processor pool data for one of my managed systems, 570-1 (Figure 4). This page displays the shared processor pool usage for 570-1 over the last day, week, month and year.

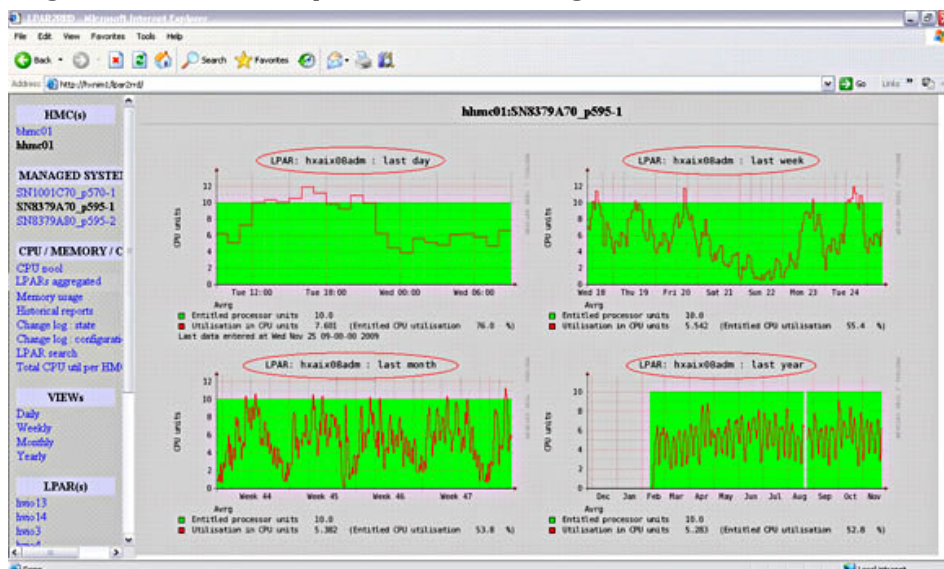
**Figure 4. Shared processor pool usage for 570-1**



On the left-hand side, under **MANAGED SYSTEMS**, there is a list of each of the Power systems connected to this HMC. If I want to look at a different frame, I simply click on the managed system name. Or if I want to look at different managed system, on another HMC, I simply click on the HMC name, under **HMC(s)** and select the desired managed system name.

To view usage data for an individual LPAR, I can select the LPAR name under the **LPAR(s)** list on the left-hand side. Shared CPU usage data is shown for the LPAR over the last day, week and year. Figure 5 is an example of this.

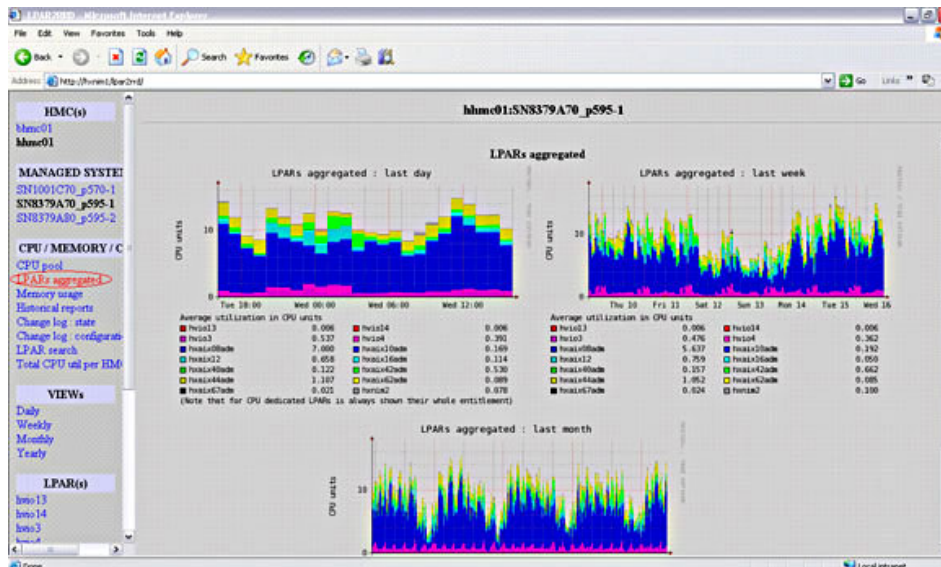
**Figure 5. Shared processor usage for an LPAR on 595-1**





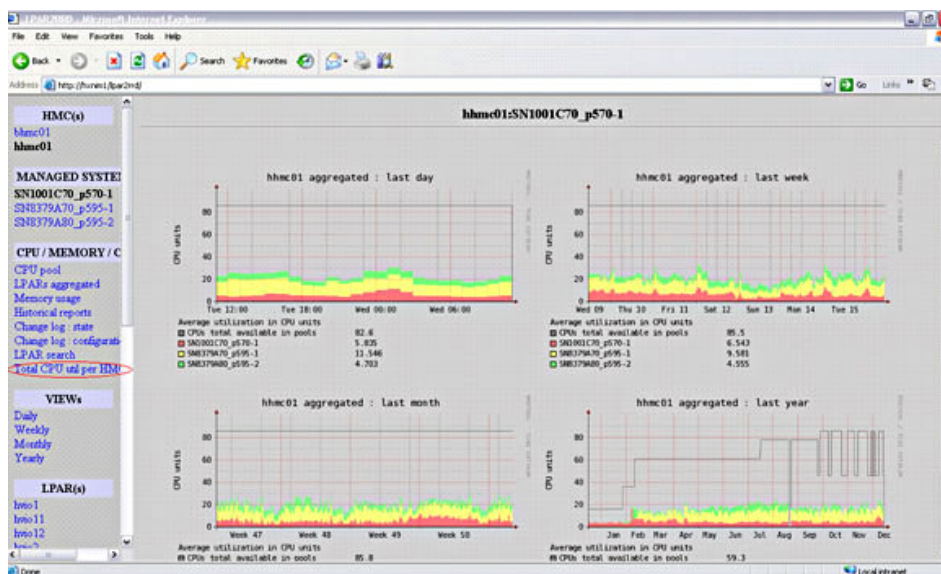
Another great feature of lpar2rrd is the aggregated LPAR view. This gives you a view of all the LPARs on a frame and how much processor each is consuming (see Figure 6 below).

**Figure 6. Aggregated LPAR view for all LPARs on 595-1**



The **Total CPU util per HMC** view is also interesting. You can quickly observe which of the managed systems connected to an HMC is using the most processor. It may help you determine where you could move an LPAR to balance your workload. For an example, you can refer to Figure 7.

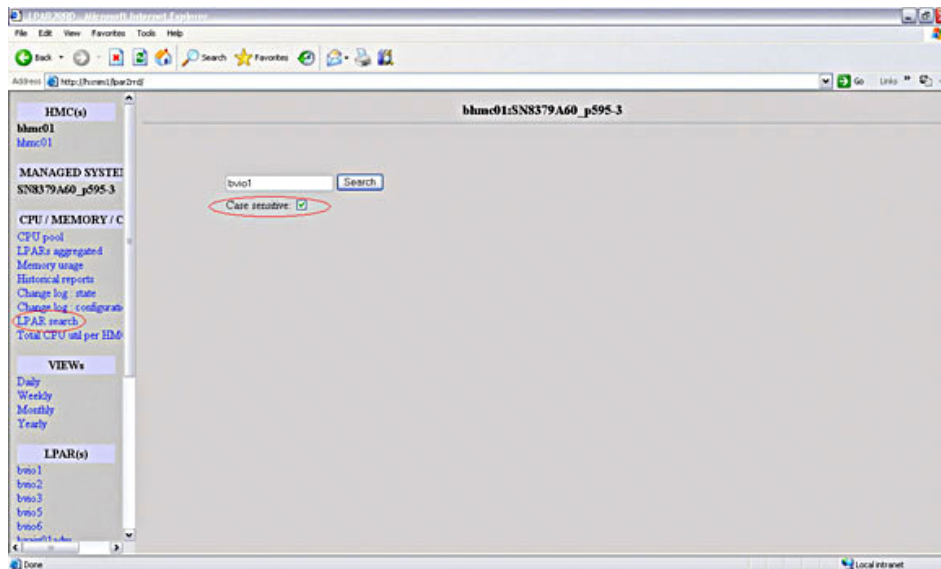
**Figure 7. Total CPU utilization per HMC view**



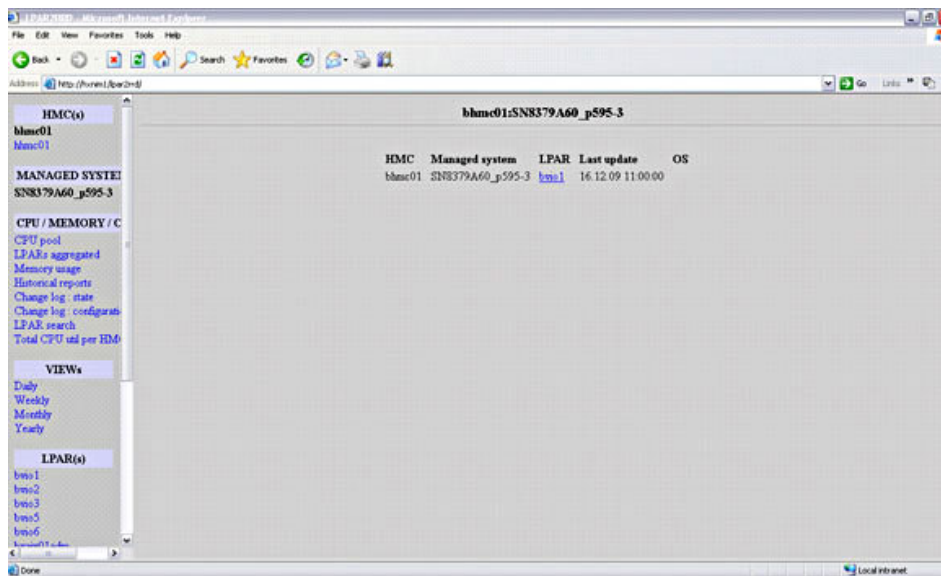
A search function is also available if you are looking for an LPAR on a particular managed system. Under **LPAR Search**, simply enter the name of the LPAR, click search and you will be presented with a link to that systems data. Note: I have found that I had to tick **Case sensitive** when entering

data into the search box, otherwise nothing would return from the search. For a screen shot example, you can refer to Figures 8 and 9, below.

**Figure 8. LPAR search box**

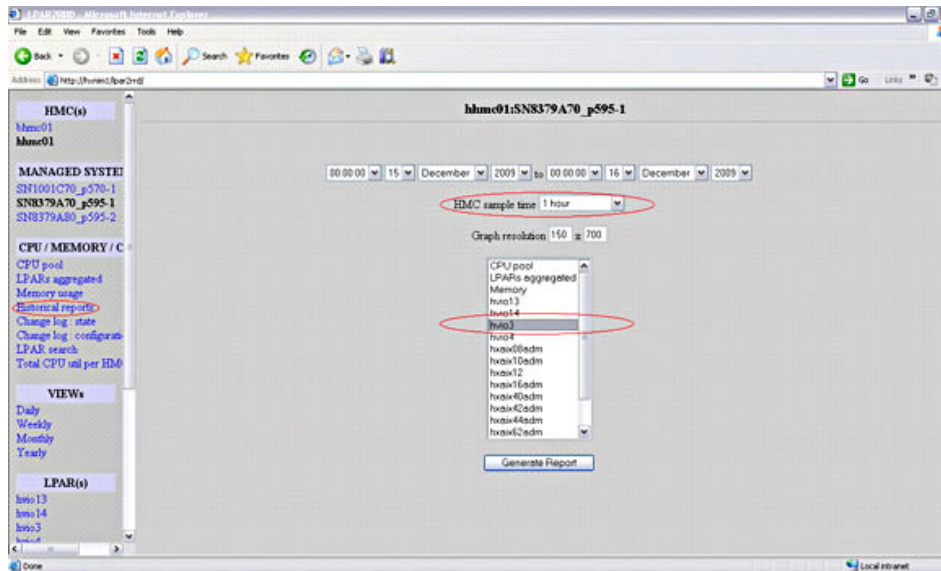


**Figure 9. LPAR search results**



It is also possible to produce historical reports based on your own date and time criteria. Do not forget to select the correct **HMC sample rate**, otherwise you will not see any matching data! For example, in [Figure 10](#), I selected hvio3. The result, in [Figure 11](#), shows me the shared processor usage for hvio3 from midnight 15<sup>th</sup> Dec 2009 to midnight 16<sup>th</sup> Dec 2009. Note: there is also a handy **Export to CSV** link on the right of the page.

**Figure 10. Generating a historical report for an LPAR on 595-1**



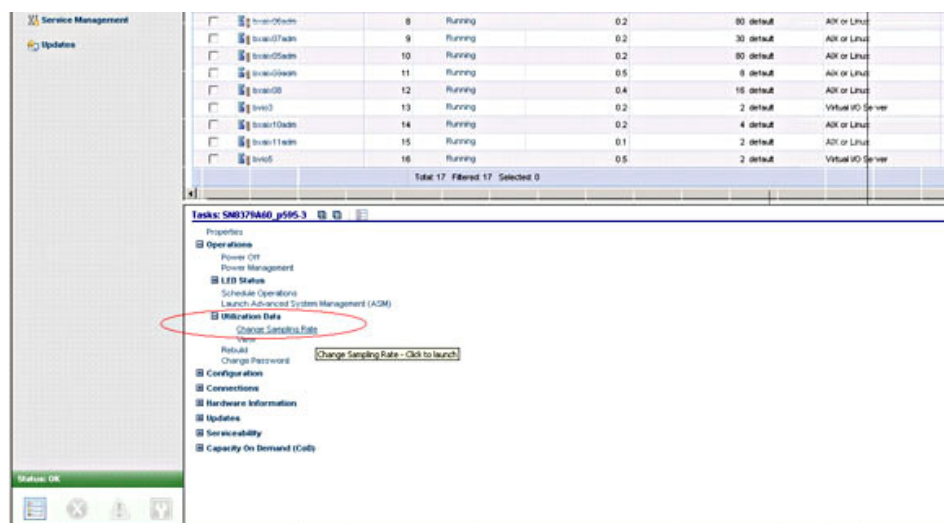
**Figure 11. Historical CPU usage graph for an LPAR on 595-1**



To collect utilization data on the HMC, data collection must be enabled for each managed system (Figure 12). This is performed via the HMC and is covered in the lpar2rrd setup guide.

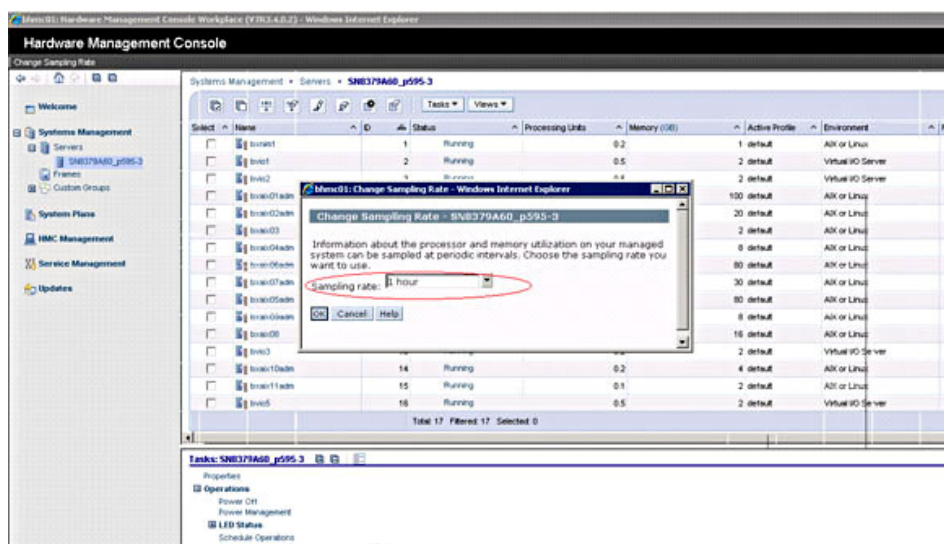


**Figure 12. Enable the capture of utilization data via the HMC**



You can also customize the sampling rate for which data is collected (Figure 13). Prior to release 2.01, the default setting was one hour. This can now be changed to less than that, depending on what you need (possible sample rates are 30s, 60s, 300s, 1800s and 3600s).

**Figure 13. Changing the sample rate via the HMC**



## The lpar2rrd environment and data

To give you an idea of what a typical lpar2rrd installation and its data files may look like, I have provided some output below. After I had followed the installation and configuration steps outlined on the lpar2rrd Web site, there was very little left to do. I have rarely needed to manage the underlying lpar2rrd environment, but it is always helpful to know a little about what's "under the hood."

The environment resided on my NIM server (hxnim1) in a directory under the lpar2rrd users home directory (e.g. /home/lpar2rrd/lpar2rrd).

```
lpar2rrd@hxnim1 /home/lpar2rrd $ ls -ltr
total 400
drwxr-xr-x  4 lpar2rrd staff   4096 Feb 11 2007 lpar2rrd
```

The lpar2rrd user on hxnim1 was able to ssh directly to both of my HMCs without a password.

```
lpar2rrd@hxnim1 /home/lpar2rrd/.ssh $ ssh hhmc01 date
Wed Nov 25 09:36:32 EST 2009

lpar2rrd@hxnim1 /home/lpar2rrd/.ssh $ ssh bhmc01 date
Wed Nov 25 10:08:40 EST 2009
```

I created a new cron job for the `load.sh` script. It would run every hour, to correspond with my one hour sample rate on the HMCs.

```
lpar2rrd@hxnim1 /home/lpar2rrd/lpar2rrd $ crontab -l
05 * * * * /home/lpar2rrd/lpar2rrd/load.sh > /home/lpar2rrd/lpar2rrd/rrdload.err 2>&1
```

I customized the lpar2rrd configuration file (as part of the installation instructions) to fit my environment. I have highlighted the variables that I changed below.

```
# Directory where the tool is installed
INPUTDIR=/home/lpar2rrd/lpar2rrd

# WWW directory where the tool places graphs, directory must exist before the first run,
# make sure that rights are correct
#WEBDIR=/home/apache/html/lpar2rrd
WEBDIR=/opt/freeware/apache/share/htdocs/lpar2rrd

# user for download data from HMC, it must exist on HMC and must have allowed access
# via ssh-keys
HMC_USER=lpar2rrd

# HMC hostname (you can specify list of hostnames separated by a space)
HMC_HOSTAME="hhmc01 bhmc01"
```

When I ran the `load.sh` script for the first time, data was collected from both HMCs for all the managed systems.

```
Working for managed name : SN1001C70_p570-1Load data for hhmc01Load hourly stats
fetching hhmc01:SN1001C70_p570-1 lpar data
fetching hhmc01:SN1001C70_p570-1 pool data
fetching hhmc01:SN1001C70_p570-1 mem data
updating rrd db : hhmc01:data : /home/lpar2rrd/lpar2rrd/data/
SN1001C70_p570-1/hhmc01/in-h
updating rrd db : hhmc01:mem : /home/lpar2rrd/lpar2rrd/data/
SN1001C70_p570-1/hhmc01/mem.in-h
updating rrd db : hhmc01:pool : /home/lpar2rrd/lpar2rrd/data/
SN1001C70_p570-1/hhmc01/pool.in-h
Drawing charts for : hvio1
Create graphs for hhmc01:SN1001C70_p570-1:hvio1:d
...
creating html pages for LPAR: hxaix66
creating html pages for LPAR: hxaix68adm
creating html pages for LPAR: hxaix69
creating html pages for LPAR: hxaix70
creating html pages for LPAR: hxaix71adm
creating html pages for LPAR: hxaix97
creating html pages for LPAR: hxnim3
```

The data for each system is stored in the lpar2rrd data directory. Remarkably, it requires very little disk space, depending on the sample rate, the number of managed systems and LPARs. The `du` output below is from an lpar2rrd installation that has been running for several years. Only a little over 160MB of data exists for around 100 LPARs with a sample rate of one hour.

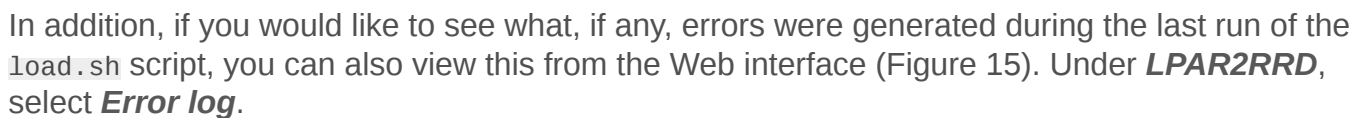
```
lpar2rrd@hxnim1 /home/lpar2rrd/lpar2rrd/data $ ls -ltr
total 0
drwxr-xr-x  3 lpar2rrd staff      256 May 16 2008  SN1001C70_p570-1
drwxr-xr-x  3 lpar2rrd staff      256 Jan 20 2009  SN8379A60_p595-3
drwxr-xr-x  3 lpar2rrd staff      256 Jan 27 2009  SN8379A80_p595-2
drwxr-xr-x  3 lpar2rrd staff      256 Feb 11 2009  SN8379A70_p595-1
lpar2rrd@hxnim1 /home/lpar2rrd $ du -sm .
162.29  .
```

Inside the data directory for a system, you will find RRD data files for each LPAR, as expected (no surprises there).

```
lpar2rrd@hxnim1 /home/lpar2rrd/lpar2rrd/data/SN1001C70_p570-1 $ ls -ltr
total 8
drwxr-xr-x  2 lpar2rrd staff      4096 Sep  01 00:05  hhmc01
total 56320
-rw-r--r--  1 lpar2rrd staff      434776 May 24 2008  hxaix21.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Sep 24 2008  hxaix05.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Sep 24 2008  hxaix03.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 24 2008  hxaix05.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 24 2008  hxaix03.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  pool.rrd
-rw-r--r--  1 lpar2rrd staff     109120 Nov 25 00:05  mem.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxnim1.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix99.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix60.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix53.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix50.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix46.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix32.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix31.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix30.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix29.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix28.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix27.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix26.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix25.rrd
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 00:05  hxaix20.rrd
...
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hxaix07.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hxaix06.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hxaix04.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hxaix02.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hxaix01.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hvio2.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hvio12.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hvio11.rrh
-rw-r--r--  1 lpar2rrd staff      434776 Nov 25 09:05  hvio1.rrh
-rw-r--r--  1 lpar2rrd staff      59960 Nov 25 09:05  in-d
-rw-r--r--  1 lpar2rrd staff      1560 Nov 25 09:05  pool.in-d
-rw-r--r--  1 lpar2rrd staff      624 Nov 25 09:05  mem.in-d
```

The good news is, once it is configured, you don't really need to worry about any of this again! In addition, if you want a quick way to view your current lpar2rrd configuration file, you can do this via

### Figure 14. lpar2rrd configuration file



### Figure 15. lpar2rrd error log



## Monitoring shared processor pools with lpar2rrd

IBM Power based systems. I have not come across any other tools (except maybe **Ganglia**, another excellent *free* tool) that can do what **lpar2rrd** does so well. I have also not come across an implementation of any products (that cost *real* money) that can do any better. The Web front-end may not look pretty but the simplicity and functionality of this tool allow us to overlook any cosmetic shortcomings. I highly recommend lpar2rrd.



## Related topics

- LPAR2RRD - LPAR CPU statistics and documentation tool for [IBM Power Systems](#).
- [lpar2rrd Demo Site](#)

© Copyright IBM Corporation 2010

([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

[Trademarks](#)

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))