NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

Installing and Configuring Cacti for Network Traffic Graphing

Conway, C.

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:
https://doi.org/10.4224/8895659
Student Report; no. SR-2006-08, 2006

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=83a07761-a283-4983-869d-902dbdf2812b
https://publications-cnrc.canada.ca/fra/voir/objet/?id=83a07761-a283-4983-869d-902dbdf2812b

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Canada

DOCUMENTATION PAGE

REPORT NUMBER	NRC REPORT NUMBER	DATE
SR-2006-08		April 2006

REPORT SECURITY CLASSIFICATION	DISTRIBUTION
Unclassified	Unlimited

TITLE

INSTALLING AND CONFIGURING CACTI FOR NETWORK TRAFFIC GRAPHING

AUTHOR(S)

Conway, C.

CORPORATE AUTHOR(S)/PERFORMING AGENCY(S)

Institute for Ocean Technology
Memorial University of Newfoundland

PUBLICATION

SPONSORING AGENCY(S)

IMD PROJECT NUMBER	NRC FILE NUMBER

KEY WORDS	PAGES	FIGS.	TABLES
Networking, Cacti, Network Traffic Graphing	22 + app.	0	0

SUMMARY

This report presents an installation and configuration guide to setting up a Cacti network traffic graphing solution with particular reference to how to set up Cacti for use at the Institute for Ocean Technology. This is the final work report for Memorial University Electrical-Computer Work Term Two student Chris Conway. This represents the outcome of a term project that can be used as a guide for further system development and deployment.

ADDRESS National Research Council
Institute for Ocean Technology
Arctic Avenue, P. O. Box 12093
St. John's, NL A1B 3T5
Tel.: (709) 772-5185, Fax: (709) 772-2462

National Research Council Canada
Conseil national de recherches Canada

Institute for Ocean Technology
Institut des technologies océaniques

INSTALLING AND CONFIGURING CACTI FOR NETWORK TRAFFIC GRAPHING

SR-2006-08

Chris D. Conway

April 2006

TABLE OF CONTENTS

APPENDICES

1.0 INTRODUCTION

Cacti has been called "The Complete RRDTool-based Graphing Solution" for its diversity and ease of use for graphing and monitoring for network traffic. It is built on the RRDTool, which is a program that holds, manages, and graphs data from round robin databases. The idea behind a round robin database is that once data gets too old (past two years) it is written over. The reason for this is two fold, one is that there is very little use to keep accurate network statistics from two years ago and the other is because this curtails the size of the database, so it won't grow past its preset size.

Cacti's interface and its use of the RRDTool is done through PHP. PHP is a robust programming language that allows for a very easy to use web interface and very quick on-the-fly calls to the RRDTool to create graphs. All of Cacti's graph settings, layout options and customizable user settings are stored in a mySQL database. A mySQL database keeps data organized in a way that is very quick and easy to access, very protected, and easy to back up.

Cacti, to the regular user, looks just like a log-in-to-use web page and has a drop down menu of devices (all of the communication closets, the firewall, the passport server) each one with nodes of each data port on that device. The graphs can be set up to show anywhere from a years worth of data to a half hour interval. From these graphs a system administrator can see what areas need higher bandwidth service, upgrades, or see if something is malfunctioning in the network.

Cacti, to the system administrator, has many of the same polished options that the user enjoys. A management console allows for easy movement of graphs from tree to tree and easy creation of new graphs. The console is really editing the mySQL database, which stores all of the "what goes where" information. It requires no programming knowledge to update and add new devices, and is as easy as using any web-mail service to operate and move graphs.

This is how cacti does what it does:

Every five minutes a PHP file (poller.php) is run by the computer in the form of a Linux 'cron' job. This sends a message to the devices that the administrator has set up in the console. The devices respond back with their statistics for that time period.

This data is taken by the PHP code and saved to the round robin databases. The mySQL keeps track of what devices store data in what database, so the PHP can do this very quickly and automatically.

That's it! This runs every five minutes and keeps the RRD files updated.

When a user goes to the site the PHP (after asking the mySQL database where everything is) will call the RRDTool and ask for a graph to be generated on demand from the stored data. Within a second that graph will be generated and put up on the screen. That's all there is.

The most time consuming part of setting a Cacti system is building the interface, setting up trees and then putting graphs on each leaf. There are some tools that allow for bulk input into the mySQL tables. These are pieces of PHP code that, instead of writing to mySQL directly, act as a admin would, using the Cacti PHP code. This way Cacti's event handling is still used so there aren't any errors, but the admin doesn't have to go though hours of creating trees from the web interface. I modified the PHP code that was designed to do this to make it much easier for me to add the number of devices and ports I wanted. This PHP, unlike Cacti's PHP, doesn't run in a web browser, but rather the command line.

After all the initial setup is done and the documentation is written someone without much prior computer knowledge can maintain the databases through the web interface. The other advantage of this is that an administrator can access the system remotely though the Internet and add new graphs as if they where on the computer while still having the round robin archives safely protected.

There are other programs that run a similar system to Cacti, the most notable being MRTG, however, MRTG is not quite as easy to handle, are based on an older programming language (perl) and, most importantly, doesn't look as nice. The reason that appearances do matter here is because when effort is put into making graphs that are information rich and meaningful it is a distraction to see an unpleasing, difficult to navigate web interface.

Cacti is perhaps the best open source network graphing solution currently being developed. Cacti is still growing and although this version is stable and does everything that is required of it future versions of the software may have advantages that make managing and monitoring network traffic even easier and more useful.

1.1 Acknowledgements

I would like to thank the following people who have helped in the creation of this manual.

Gilbert Wong and Doug Walsh – For help getting all of the systems that I have installed cacti working and helping me learn how to use Linux.

Tobias Oetiker – For Creating the RRDTool and making all of this possible.

All of the Developers and Maintainers of Cacti and the Cacti Forums – For Creating Cacti and Helping me solve the problems that I ran into.

2.0 INSTALLING AND SETTING UP CACTI

2.1 Install Required Programs

There are four important tools that are required to install Cacti. They are PHP, the Apache Web Server, the RRDTool and mySQL. These are all free, open source and easy to find. It is assumed that a "modern" web browser is installed on the Cacti system, for example: Mozilla Firefox.

On a Debian Linux based system (such as Ubuntu) the programs can be retrieved from the terminal by using apt-get. The commands are as follows (root privileges required):

```
apt-get install apache
apt-get install php4-mySQL
apt-get install php4-cli
apt-get install mySQL-server
apt-get install rrdtool
apt-get install snmp
```

Note: Apache2 will work, but apache is used in this manual
Note: The IOT Cacti backup files are mySQL 5.x
Note: RRDTool 1.2.x is recommended

2.2 Install Cacti

2.2.1 Installing via apt-get

There are two ways to install cacti. The first is to use the apt-get command to get it. In Ubuntu if it's not found by apt-get try editing the sources.list file in /etc/apt/ to include the "backports."

```
apt-get install cacti
```

This setup may prompt for a mySQL password for the administrative user, if one is not configured then use the default or a blank, this will be configured manually later in this manual to ensure that the passwords are correct.

apt-get installs cacti to /usr/share/cacti, which is not a favorable place for a webpage that will be hosted. Instead the /var/www folder that apache uses as a default is nicer location. Move the folder from /usr/share/cacti to /var/www/cacti. This is the location I will assume Cacti is in for the rest of this manual.

The location change requires one other change, the cron file. Edit /etc/cron.d/cacti with the line:

```
(versions >= 0.8.6g):

MAILTO=root
*/5 * * * * www-data php /var/www/cacti/poller.php > /dev/null 2>&1

(versions < 0.8.6g):

MAILTO=root
*/5 * * * * www-data php /var/www/cacti/site/poller.php > /dev/null 2>&1
```

The apt-get version of cacti is usually not the current one. At the time of writing it is 0.8.6f, which has a slightly different file structure then the version that the tarball will produce. The instructions for updating can be found in section 2.2.

2.2.2 Installing via tarball

The newest version should be available at http://www.cacti.net. Just extract the tarball:

```
tar xzvf cacti-version.tar.gz
```

This may prompt for information regarding the mySQL database. This information will be edited after manually to ensure that it is correct, so the default values are fine.

Move the extracted files to /var/www/cacti/.

Create (or edit) /etc/cron.d/cacti with this line:

```
MAILTO=root
*/5 * * * * www-data php /var/www/cacti/poller.php > /dev/null 2>&1
```

Where www-data is the name of the web group.

2.3 Setting Up The mySQL Database

To create a mySQL database for Cacti there are a few commands that require some knowledge of mySQL. On an already existing mySQL install there may be a password on the root account, which is needed. On a fresh install there is no password and the following steps will create the Cacti database.

1) Create a mySQL database called "cacti":

```
mysqladmin -uroot create cacti
```

Some versions of the installer create this database, so this step may result in the mysqladmin giving the error message that the database already exists.

2) Enter mySQL manager:

```
MySQL -uroot
```

3) At the mySQL prompt type (to give cacti all permissions on it's db):

```
grant all on cacti.* to cacti@localhost identified by "cacti_password";
```

"cacti_password" can be replaced by any password. No users ever use this password in the web-based interface; it is only for internal php-mysql queries.

4) Exit mySQL and Import the default cacti database, if there is already a cacti database from another machine to be imported change this line to fit:

```
mySQL cacti < /var/www/cacti/cacti.sql
```

After this is complete the mySQL database is setup. Now edit /var/www/cacti/include/config.php (older versions need to make this edit after the "require Debian.php" line.):

```
$ database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti_password";
$database_port = "3306";
```

It is also almost always necessary to run:

```
dpkg-reconfigure -plow php4-mysql
```

The –plow sets the priority (-p) of questions to "low".

Please note that the mySQL root password should be setup at this point to protect files if it has not already been.

2.4 Starting Cacti

Set up apache to (by editing /etc/apache/httpd.conf) to have cacti as an alias. To do this insert the lines (for versions older then 0.8.6g the additional folder, /site is needed:

```
#Alias for cacti.
Alias /cacti /var/www/cacti
<Directory /var/www/cacti/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

in the alias section of that file. Also make sure that the lines of the file that need to be uncommented to allow for php4 to run are uncommented. Then restart the apache server (/etc/apachectl restart).

At this point Cacti is set up and there are few other commands that require the command line (shell). Using a web browser go to the location that Apache has its web address (localhost) and access the Cacti directory. (http://localhost/cacti)

Cacti will recognize that it is a fresh install and prompt for some basic information. Most information is gathered by Cacti and merely needs a confirmation.

After this quick configuration there will be a login screen where the default account, ‘admin’, is setup with the default password, ‘admin’.

If for any reason the password does not work it can be fixed from the mySQL prompt, using the cacti database, with the command:

```
update user_auth set password = md5('admin') where username =
'admin';
```

This can also be used to reset the admin password if it is lost or compromised.

2.5 Getting add_tree.php

There is a custom file for cacti that make the admin/creation of new graphs much easier. It was found on the Internet and customized (as it was open source) to work in an easier fashion. This file is called add_tree.php. This should be found in an IOT local drive (pccommon N:\software\cacti) in an add_tree_backup folder. The add_tree.php should be copied into the cacti directory (/var/www/cacti/) and

the file (api_import_tree.php) in the 'lib' folder of that directory copied to the lib directory of cacti. Notes on how to use that php file can be found later in this document in section 6.0.

2.6 PHP Modification

For a slightly easer way of using Cacti I have changed the file /var/www/cacti/include/top_graph_header.php.

The code to edit is found in a <td align="right"> block, slightly below:

```
<a href="logout.php">Logout</a>
```

There will be a line that has only:

```
<?php } ?>
```

This is line 132 which will be changed to:

```
<?php }else{ ?><a href = "index.php">You are not logged in: Log
In</a><?php } ?>
```

That is the line that needs to be replaced. This is so that when a user is using Cacti and the cookie expires they understand that they are now in guest mode. Before the user would have to know to go back to index.php (the home directory) as there was no link.

2.7 Permissions

It is necessary to change the permissions of the cacti files. It is ideal to have root as the owner and www-data as the group. The best permission set I have found is listed in command form:

```
chown –R root.www-data /var/www/cacti/
chmod –R 774 /var/www/cacti/
```

The permissions are setup so that www-data can write to the /rra files and execute any scripts while only giving others read access for the web pages.

3.0 BACKING UP, UPDATING AND PATCHING CACTI

3.1 Backing Up

3.1.1 Introduction

Cacti keeps all of its data structure information in a mySQL database (db) that tells the PHP where to find round robin databases (RRD's) and how to arrange the trees and data presentation. The mySQL db does NOT contain any information on statistics or any graph information (other then titles). All of this is stored in the rrd files found in cacti's rra folder. If a system move or a complete backup is required then it would be advisable to copy all of the rrd files to the new location. When Cacti is back up and running it will recognize the rrd files and continue to write to them. If the rrd backup is omitted then all of the organization and user info from the db will still be in cacti, but there will be no statistics until the first run of poller.php (automatic after five minutes if cacti is setup properly -- cron job).

3.1.2 SQL dump

To make an mySQL file of the whole cacti directory (must have permissions, e.g. root):

```
mysqldump -B cacti > /path/cacti_backup.sql
```

To put the sql file back in the db use the command:

```
mysql (dbname) < /path/cacti_backup.sql
```

Note: the dbname has to be one with write permissions to and there cannot be a Cacti database already in the mySQL database (see Appendix A for mySQL commands to remove old Cacti databases). It's usually best to just make a junk Database and then delete it after. The Cacti db will have nothing to do with that db when unpacked it is just a landing point for the data. Ensure that the permissions on the new cacti databases are correct (the grant all command).

3.2 Updating

3.2.1 Back up

Instructions for backup of the mySQL database are outlined above.

3.2.2 Finding and extracting

Find a new version of Cacti by either "apt-get-ing" as described in the installation documentation of this manual or go to Cacti's web page (http://www.cacti.net) and download the newest version. The version available on the webpage is usually more current then the one available though apt-get. Instructions for extracting from the tarball can be found in section 2.2.2.

3.2.3 Fail safe backup

It is safer to save the whole old Cacti directory until the new version of Cacti is completely up and running. Backup the old cacti directory (make a new dir, cacti_old):

```
mv /var/www/cacti /path/to/cacti_old
```

3.2.4 Copying the new version

Copy the extracted files to the cacti (now empty) directory:

```
mv /path/to/cacti-version /var/www/cacti
```

3.2.5 Editing the include file

This is the same as the initial installation.

Edit /cacti/include/config.php so that all of the entries match the db entries:

```
$ database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti_password";
$database_port = "3306";
```

3.2.6 Copying round robin archives

Copy the old round robin archives to the new rra folder:

```
cp /path/to/cacti_old/rra/* /var/www/cacti/rra/
```

3.2.7 Permissions

All folders should be owned by root and have group permissions for www-data to read and write. www-data is the name that the cron job uses to access files.

It is ideal to have root as the owner and www-data as the group. The best permission set I have found is listed in command form:

```
chown –R root.www-data /var/www/cacti/
chmod –R 774 /var/www/cacti/
```

The permissions are setup so that www-data can write to the /rra files and execute any scripts while only giving others read access for the web pages.

3.2.8 Copying plug-ins

Any extra scripts or plug-ins that are installed must be copied to the new directory.

See Sections 2.5 and 2.6 for further information about any Cacti customizations that this manual recommends.

3.2.9 Additional note

Any changes done to the path or directory must be changed in Apache by editing /etc/apache/httpd.conf. And then restarting Apache:

```
/etc/apachectl restart
```

This should be it for the upgrade. Jumping from 0.8.6f to 0.8.6h saw the /site/ folders removal. This is worth noticing because it requires a change to apache's alias configurations.

The first time the webpage, http://localhost/cacti, is reloaded notice the install and there will be a few steps to follow. As noted above, some version changes have path changes so any options that give warnings or errors may be caused by that.

Also, note that if the directory changed the path that the cron job has to run every five minuets to keep cacti refreshing and polling must be changed as well. /etc/cron.d/cacti is where it can be found. Change the path in the file to the new path.

3.3 Patching

Patching Cacti is a simple process, although it doesn't make much difference to the functionality or daily use of Cacti. . I recommend doing it from the cacti folder. Patches must be found on the cacti web page (http://www.cacti.net). Once the patch has been found replace version with the current version and patch.patch with the appropriate file name. This can be done from the command line with the command:

```
wget http://www.cacti.net/downloads/patches/version/patch.patch
```

Followed by:

```
patch -p1 -N < patch.patch
```

Patches may not be needed and newer versions may not even have any (yet).

For the current version (0.8.6h) here are the current patches installed:

fix_search_session_clear_issue.patch
fix_sql_syntax_related_to_default_rra_id.patch
nth_percentile_empty_return_set_issue.patch
mysql_5x_strict.patch

Here are the command line prompts to download them and patch Cacti:

```
wget http://www.cacti.net/downloads/patches/0.8.6h/fix_search_session_clear_issue.patch
wget http://www.cacti.net/downloads/patches/0.8.6h/fix_sql_syntax_related_to_default_rra_id.patch
wget http://www.cacti.net/downloads/patches/0.8.6h/nth_percentile_empty_return_set_issue.patch
wget http://www.cacti.net/downloads/patches/0.8.6h/mysql_5x_strict.patch

patch -p1 -N < fix_search_session_clear_issue.patch
patch -p1 -N < fix_sql_syntax_related_to_default_rra_id.patch
patch -p1 -N < nth_percentile_empty_return_set_issue.patch
patch -p1 -N < mysql_5x_strict.patch
```

4.0 USING CACTI

4.1 Reminder of Official Manual

While this manual is designed as a stand-alone manual for Cacti the official Cacti manual is still a very good source for information, especially in the area of using Cacti. This manual has been built upon the Cacti manual's sometimes-vague notes on Cacti. The Cacti forums on the Cacti web page are also a valuable source of information and are accessible to everyone.

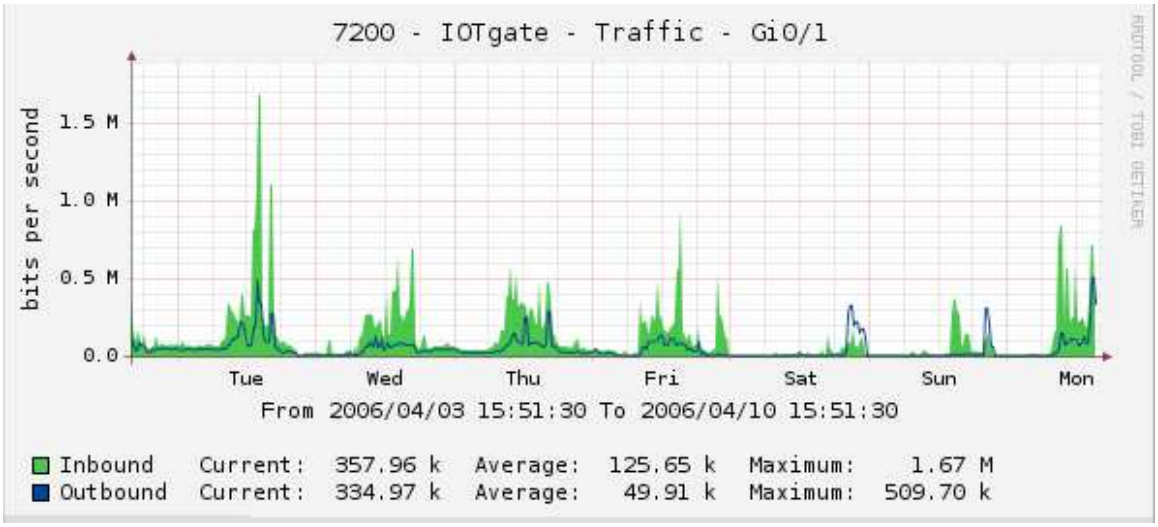

Figure 4.1 – Sample Cacti Graph (Total Incoming to IOT)

4.2 Overview Of Using Cacti

Viewing Cacti is broken down into three parts. All three have a command that allows the scope of the graphs to be shown from anywhere from the last half hour to two years. Tree view is the standard view. This shows drop down tabs of devices and different graphs on those trees. This mode is by far the easiest way to display graphs, however, it takes a little more time to set up. The next view is list view. This mode has drop down boxes to choose which device to use and then a linked list of all the graphs that are available. This mode can be cumbersome with many graphs on a device. Preview mode is much like list mode with the exception that there are small preview graphs instead of listed names. There is also a 'filter' on list and preview modes that allow for fast finding of graphs.

Clicking on a graph in any mode will show the different time scales of the graph. Clicking on the magnifying glass icon next to the graph will enter a zoom mode where sections of the graph can be highlighted and magnified. This is useful when there is a strange spike or unexpected drop off that needs to be further examined.

Managing Cacti is, overall very simple to use and was designed to be just that. It has a layout that is designed to be as easy and functional as possible and to be graphically pleasing while doing so. The main menu contains a list of options that are described in this section in more detail. For the most part however the natural way that Cacti presents its use it easy to follow.

4.2.1 Creating devices

To create a device click on the 'devices' link and then, on the following page, click on the new button in the upper right corner. Name the device and input the

IP address. If there is a SNMP domain then that needs to be inputted as well. If all of this information is correct the graph should create. If not then it will redirect back to the same page and give an error under then name of the device.

Setting the 'Associated Data Queries' to SNMP - Interface Statistics will have Cacti do a SNMP walk when the create graphs page is opened and show what is graph-able.

4.2.2 Creating graphs

To create a new graph click on the ‘add new graph’ link and choose a device. The device will then present a list of scripted options of graphs that can be created and a SNMP walk of the port traffic to that device. Simply check off the graphs that are desired. This will create data sources and graphs for the selected. The Data Source is where information on where the data is being gathered from and where the data is being stored. A graph is just a mapping to a data source.

This can be done manually, without the ‘add new graph’ button by clicking on the ‘data sources’ link and entering new data source and choosing the one that is required from the list. After creating that data source, click on the ‘graphs’ link and, then clicking add and then choosing the in and out fields of the data source that was just created from the presented drop down boxes.

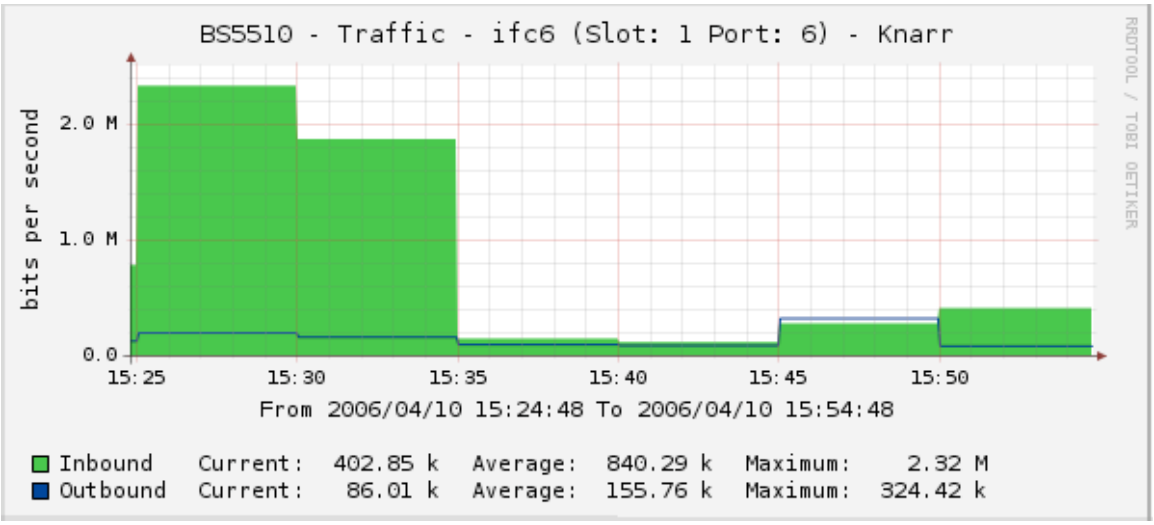

Figure 4.2 – Last Half Hour Scope for Knarr (Windows Server)

4.2.3 Creating trees

Trees can be created and managed via the trees link. To add a new tree click the add link in the upper right corner. To manage the nodes on a tree click on the tree that needs managing. Here more nodes and sub-nodes can be added.

To create a large number of trees, nodes and sub nodes it is recommended to use add_tree.php. Information about how to use add_tree.php can be found in Part 6.

4.2.4 Data sources

Data Sources, as mentioned above, hold how to get the data and where to store the data. Mapping a graph to a data source is what creates graphs. A graph can be deleted without deleting the data source. Data sources are automatically created when using the 'add new graph' functionality.

Each data source can only be mapped to one round robin archive (rra) however there can be more then one graph made for each data source.

Data input methods, the next section, deals with some more advanced uses of data sources.

4.2.5 Creating templates

Building effective templates can be trickier then the rest of Cacti's web interface. In templates (graph) data can be mapped to several types of graph objects. Area's are traditional bar-graph-like graph, line 1-3 are lines of varying thickness and GPRINT is a way of making text on the bottom of the graph. The legend function will create three GPRINT items, a max, current, and min, which is very useful for data-rich graphs. The duplicate template feature is very useful for trying new templates.

I have created several graph templates (for 4, 8, and 25 port graphs). If anything is unclear about how to build a template then look at the default Interface - Traffic (bits/sec) template. This can be used as a good template for making further templates due to its very basic settings.

More information for the creation of templates is given in the next section, Data Input Methods when both Data Templates and Graph Templates are required. For the most part there is little reason to create a graph template without an new Data Input Method.

4.2.6 User management

User Management is a very nice and easy to use feature in Cacti. Clicking the 'Users' link allows for the creation of accounts and the management of users. Creating a user is easy, just clicking on the add button in the top right corner will bring up a new user screen. There are many, many different levels of permissions that a user can have and I will not go in to all of them because most of them are self evident, such as 'user must change password on first login'.

The more useful feature of User Management is the Graph Permissions tab. This tab brings up the Graph Permissions page that limits the users access to certain graphs. The default policy drop-down can be set to allow or denied and then exceptions of certain graphs, devices or tress can be made. This can allow users to only see graphs that are relevant to them. Disallowing tress can be somewhat useless, as going into list mode or preview mode allows the user to still see the graphs. I suggest Devices as the limiting tool for non-privileged users.

4.3 Data Input Methods

4.3.1 Introduction

Cacti can use scripting languages to collect data. It can interpret most languages installed on the system and commonly uses scripts written in python, perl, and PHP. They are only limited to what the script can do with data collection. There are several steps involved in making scripts work with Cacti that are outlined here.

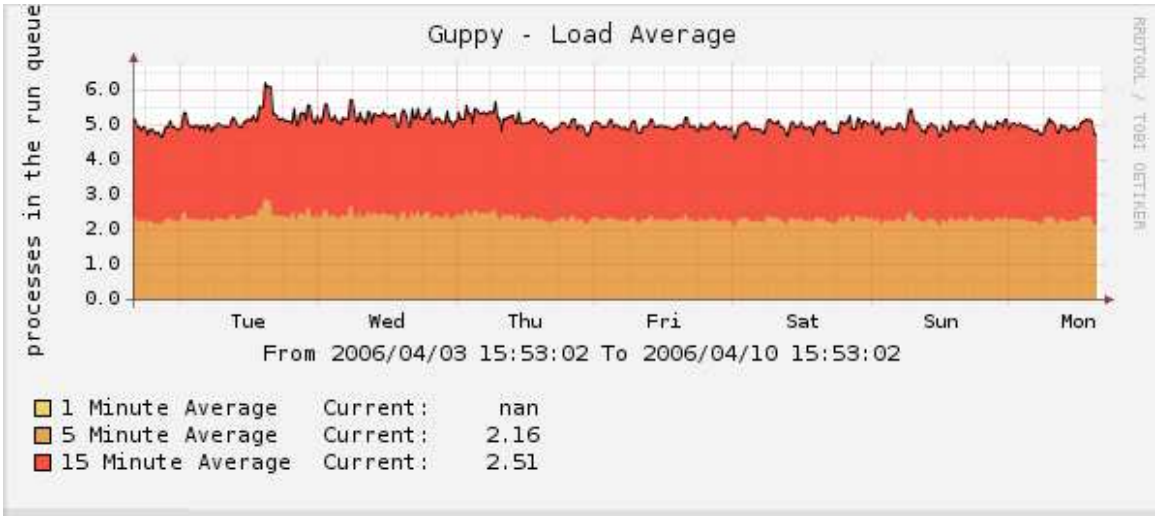

Figure 4.3 – Graph Made From uptime Script on Guppy (Cacti's Server)

4.3.2 Create new method

Go to the Data Input Methods screen and click the new button. Enter a name and change the drop down box to "Script/Command". In the input string box type:

```
python <path_cacti>/ups/gettemp2.py -a <upsno>
```

Where python can be any language that cacti understands (installed on the system) and <path_cacti> is a command for cacti to use the path to cacti (where the script is). The -a was defined in the script to prepare for an input and save it. This can very on the scripts usage and should be checked before trying to make the script work in Cacti (make sure it works on the command line before Cacti).

The <upsno> is a variable that will be defined later; any name can be used for this variable.

Note that the IOT-Python scripts are set up so that, in the command line, the command:

```
python /path/to/gettemp2.py -a 1
```

will get the temperature information of UPS 1. This is modified somewhat from the original, which used the -n command.

4.3.3 Create input field

Add an Input field. Make sure that the drop down box is selected on the newly created variable. Everything else can be left blank except for the last box (Special Type Code), which needs the input "hostname". Add an Output Field, no special instructions required.

4.3.4 Data templates

Next click on the "Data Templates" button in the menu and select to add a new template. Everything can be named freely, just making sure that the drop box is set to the new field that was just created. Save it and re-enter the template. Now there should be an Output Field box (select the output field from before). There also should be a "Custom Data" box; this is where the input variable from before can be defined. For the cases of the UPS scripts this should be the UPS number (1 or 2), but depending on the script it can be anything.

After saving click on the "Graph Template" link and add a new graph template. This can be created like a normal graph. After the first set up template items can be added. I suggest adding an area (graphed to the newly created template) and a Legend type object to create a simple descriptive graph.

4.3.5 Create data source

Now the data source can be created. Under the Data Source link choose the device that the script should run off of (for simplicity it is best to have it run from localhost). Click on the Add link and create the new data source from the template. This will create the rrd and start collecting data every five minutes (with the cron job).

4.3.6 Create graphs

The graph can now be created. Under graph management click add; choose the graph template and host and then the data source.

Now the script should be creating the graph!

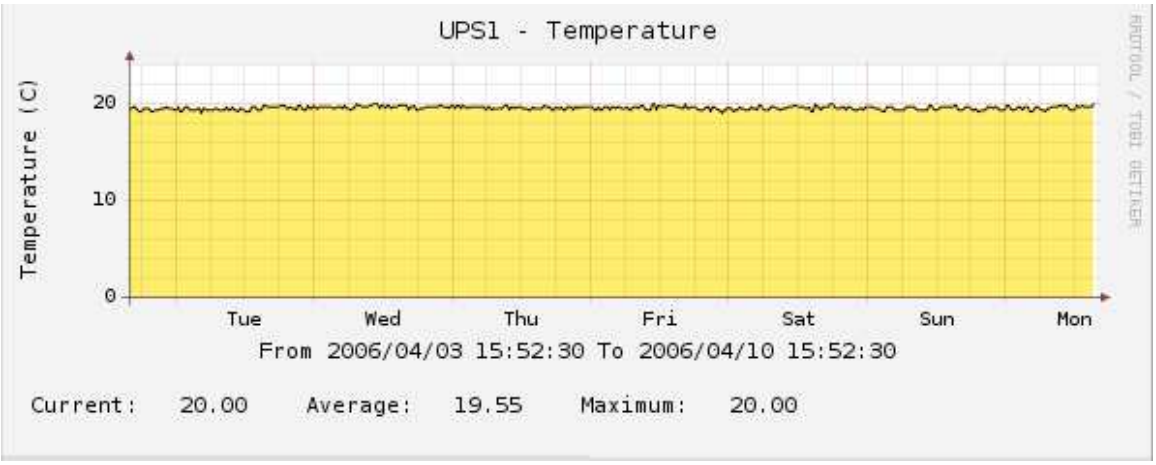

Figure 4.3 – Graph Made From upstemp Script

4.3.7 Notes

A Data Source is where the rra's are kept. A data source will use the script it was created with, even if that script is modified. If any changes to the script are made then the data source must be recreated.

The data generally needs a new line after it (default for python) or nothing after it. Sometimes it can be tricky, so make sure (using the data source and graph debug modes) that the RRDTool is receiving the data correctly. If it is not receiving the data correctly the graph debug and data source debug should look fine and the graph should display nothing (if there are number stats turned on then they should display 'nan').

4.4 Templates

This is pretty self-explanatory, however I will go over the basics.

Any template that is created in cacti can be exported as an XML or test file to be transferred to another mySQL Cacti database without having to copy the whole database. This allows for data templates that are time-consuming to create to be moved with little effort. This also allows for sharing of useful templates online. If there is a template type that is required sometimes is can be found with a web search.

4.5 The UPS Scripts

The UPS scripts are custom python scripts that read log files in from /var/log/ that contain the output strings from UPS one and two. The first time setup of the scripts was one of the most time consuming parts of creating the cacti database. The difficult parts where mostly that the way of creating new data sources was

not clearly documented. The process of creating the data sources is outlined above for any additional additions.

The UPS scripts are found in /var/www/cacti/UPS (and, for back up purposes, pccommon N:\software\cacti\ups), not the cacti scripts directory. In the folder there are scripts labeled “getsomething.py” and “getsomething2.py”. The original script from MRTG is the one without the two and the one with the two is the modified version which cacti uses.

It is very important that www-data has +x permissions on these files, as the cron has to run them with the rest of the queries at runtime. Running poller.php from command line (see section 4.1 for information about this) can show any errors associated from these scripts not working. The most common error is having the scripts not find the log files. To fix this, make sure that the log files are being delivered and that they are readable by www-data.

4.6 Disabling Cacti

For maintenance it may be required to temporally stop Cacti from using resources. Because poller.php is the direct cause of all other Cacti system calls it simply needs to be disabled. To do this edit /etc/cron.d/cacti and comment out all of the lines. This will, after a few minuets, stop cacti from doing any polling and leave holes in the graphs.

4.7 Speeding Up Cacti

Cacti can be taxing (causing uptime to rise from near zero to an average of three on the Pentium threes that I have built on) on a host system and speeding up cacti can be useful. For the most part the most important thing is the processor speed. The most taxing things are the cron job’s SNMP get commands that need to run for every graph. This means that when these are running the web server may slow down and other processes on the computer may suffer. RAM use is not that high, but upgrading may produce marginal speed increases.

5.0 COMMON PROBLEMS

5.1 Error Checklist

This is a quick list of things to check if a graph is broken. Further descriptions and solutions to the problems in this list are found in this section.

-The cron job is not pointing at the correct location of poller.php
-php is not configured to run from command line
-php is not configured with the adodb libraries
-cacti mySQL database not created or not imported with the default database
-Permissions on rra files (www-data need +w)
-Permissions on files in the cacti directory (www-data should be the group and allowed to read and execute all files)

-The username/passwords in config.php are wrong
-Apache web-server not configured to run php
-Scripts missing or www-data permissions are wrong
-Wrong version of RRDTool selected
-Host blocks certain IP's (Accelar, in the case of IOT)
-Firewall doesn't allow SNMP connections

5.2 mySQL Connect Error

It is recommended for whole next section to be read before proceeding with a solution.

After installing and setting up cacti (or moving a cacti mySQL db) if things aren't graphing with 10 minutes it may be useful to manually run poller.php (it runs automatically with cacti but if there is a setup error it will not indicate what's wrong) to refresh all the graphs and create all the needed rrd files in the /var/www/cacti/rra folder.

To do this, type in terminal:

```
php /var/www/cacti/poller.php
```

If this results in:

```
Fatal error: Call to undefined function mySQL_connect() in /usr/share/adodb/drivers/adodb-mysql.inc.php on line 337
```

Or some similar error PHP must be configured to run with the adodb mySQL commands. To do that type:

```
dpkg-reconfigure -plow php4-mysql
```

Say YES to both the options.

This will allow Cacti to use the adodb scripts to interact with the mySQL database.

The downside to running poller.php this way is that root will be the owner of all the files and they will not have group write access (for automatic poller.php cron). Therefore the permissions must be changed for Cacti's /rra folder. The correct ownerships are:

Owner: root
Group: www-data

5.3 No Graph Production

If, upon connection, http://localhost/cacti displays the error:

> Warning: mysql_connect() [function.mysql-connect]: Access denied for user: 'cacti@localhost'
> (Using password: YES) in /usr/share/adodb/drivers/adodb-mysql.inc.php on line 338

A password miss-match causes this error. The password for the mySQL database does not match the password given in config.php. config.php is easy to edit and can show the passwords in plain text. After checking that password then change the mySQL password to match it. Commands for changing mySQL passwords are given in Appendix 1.

5.4 Possible Permission Problems

If Cacti is not producing graphs (even after running poller.php) check the permissions on /var/www/cacti/rra.

They are required to be (drwxrwx---):

File Owner: root
File Group: www-data

This applies for both the folders and the files in the folders. This permission can be applied to the entire Cacti folder and all files associated with Cacti.

If the permissions are not set up right the graphs will not generate. When looking at the graphs in ‘debug mode’. The debugger will say that RRDTool doesn’t have permissions.

Any data accessed by scripts (such as log files and incoming data streams) must also have read permissions for www-data. For example, the /var/log/local* logs that are sent to the cacti machine (guppy) may need to have read permissions for www-data. As the scripts are rotated every week they need to be configured to rotate into the www-data group. This caused some problems for the setup at IOT, as the logs constantly rotated into the adm group. Manually editing the log-rotate file allowed for this to be avoided.

5.5 Not a Number (nan) Errors

If all of the graph stats (listed in the graph image, below the graph) are showing “nan” then there is a problem with the RRDTool. Make sure that the right version was selected in the Cacti setup.

Cacti web interface -> Settings

This can also happen if the group permission settings are not allowing www-data to write in poller.php or the rra folder. The best way to set up the whole cacti folder is to give www-data group ownership of everything.

www-data is the id used by the cron job. The log will show this error as well and is located in: /var/log/cacti/.

5.6 IP Blocking

When setting up devices ensure that the machine running Cacti is not on an IP block list form the host that it is trying to access.

Note: Accelar is configured this way, so make sure that the machine that Cacti is running from is allowed to run SNMP on Accelar.

6.0 ADD_TREE.PHP

6.1 Introduction

add_tree.php is a bulk import tool that is very easy to use and eliminates the trouble of creating many trees and nodes for devices. For example:

A stack with 7 modules and each module has 24-25 ports.

To do that all through cacti's web interface it would (did) take a lot of time and would require each of the almost 200 ports to be manually created and then mapped with a graph.

Add_tree.php was originally open source code that would allow command line creation of trees and nodes. This was modified so that it could create many trees and map many nodes and sub-nodes in one command.

Information about how to find add_tree.php and the associated files can be found in Appendix D.

6.2 Using add_tree.php

This must be run from the command line. The command 'PHP' prior to a PHP file will run the file.

Additional instructions can be found by running add_tree.php with no arguments.

6.2.1 Creating a new tree

There is usually one tree per device, so in command line the following command will create that tree:

```
php add_tree.php --type tree --name Unit --sort-method a
```

This will then give a new 'tree-id' number, which I will call tID below.

6.2.2 Populating a tree

Each tree will have nodes that hold each graph. For this example I will create 25 nodes on the tree, one for each port on a routing device:

```
php add_tree.php --type node --node-type header --tree-id tID --parent-node 0 --name port --multi-head 25
```

This will then give a list of new 'parent-node' ID's the first of which I will call pNO below.

The multi-head method is custom to this version of add tree. It can accept any value and creates that many sub nodes on the tree. It will output the nodes with a node name and a number. For example, port 01, port 02, port 03 would result from a name of port and a multi-head of 3.

6.2.3 Mapping graphs to trees

Please Note
Sometimes, if the rrd's are set up at different times for a list of ports on a device they will not follow numerical ordering. If all the graphs where created at the same time then this method should work. If they are not in numerical ordering then the graphs and rrd’s can be deleted and recreated so they are in order or be graphed via the web interface.

The graph id can be found in the title path to the graph under the graph management tool in Cacti's web interface.

```
php add_tree.php --type node --node-type graph --tree-id tID --parent-node pNO --graph-id [first graph id] --rra-id all --multi-head 25
```

Also, note that the original add_tree.php had a lot of checking build in, this command is accidentally run twice it won't do much damage (maybe one or two double graphs, usually not though).

Appendix A

mySQL Notes

From the mySQL prompt (shell> mysql -uroot):

Show the DB list:

show databases;

Use this database:

use (dbname);

After choosing which db to use, viewing the tables in the db:

show tables;

Changes the permissions on dbname:

grant all on (dbname).* to (user)@(domain);

Change the password on dbname:

grant all on (dbname).* to (user)@(domain) identified by “pass”;

Creates database:

create database (dbname);

Removes database:

drop database if exists (dbname);

Appendix B

Other RRDTool Based Solutions

Other Solutions:

Tstat - http://tstat.tlc.polito.it/index.shtml
RRD Statistics - http://dolly.czi.cz/coyote/packages/rrd.asp
MUNIN - http://munin.projects.linpro.no/
MRTG - http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

Web Links:

Cacti - http://www.cacti.net
rrdTool - http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/
php - http://www.php.net
mySQL - http://www.mysql.com
Apache HTTP Server - http://www.httpd.apache.org

Appendix C

References

“Cacti Manual.” Cacti. Cacti: The Complete RRDTool-based Graphing Solution. 10 January 2006 <http://www.cacti.net>.

“Cacti Forums.” Cacti. Cacti Forum. 17 February 2006. <http://forums.cacti.net>.

Appendix D
add_tree.php

add_tree.php was originally found in a package called bulk import tools on the cacti web forum. It can be found via this link:
http://forums.cacti.net/download.php?id=4063

The required files from that package are add_tree.php and (from the lib directory) api_import_tree.php. These files should be found (already edited) on an IOT local drive (pccommon N:\sofrware\cacti\add_tree_backup) as well.

The modifications to make add_tree.php work like it should for this manual are all to be made to add_tree.php.

What follows are the changes made to add_tree.php by Chris Conway:

1. In the area of the code called “function usage()” around line 53 I have added a short description of my modification:

```
edit to php (Chris Conway - 2006/01/25): put in multi-head support.
this allows for creation of a series of numbered nodes: ex. name =
port, multi-head = 10 gives: port 01, port 02... port 10
```

2. Around line 127-128 there is a large case statement. I have added one more case before the default case, it is:

```
Case “—multi-head”;
        $i++;
        $numHead = $_SERVER[“argv”][$i];
        break;
```

3. Around line 223 there is a line that states:

```
# $nodeId could be a Header Node, a Graph Node, or a Host node.
```

And on line 226 there is a line that states:

```
return 0;
```

Between these two lines there is a segment of code. This code is to be replaced with the new code that is listed below. The code is not optimized or anything of that nature, but it gets the job done and it improves functionality.

The following is the code:

```
#additional add_tree code of bulk tree creation.
        $orgiName = $name;
        if($numHead > 0)
        {
          if($nodeType == 'graph')
          {
                $parentNode--;
                $graphId--;
          }

          for($i = 0; $i < $numHead; $i++)
          {
              if($numHead > 0)
              {
                    if($nodeType == 'header')
                    {
                          $temNo = $i + 1;
                          $temHold = $temNo;
                          if ($temNo < 10)
                           {
                             $temHold = "0".$temNo;
                           }
                           $name = $orgiName." ".$temHold;
                    }
                    elseif($nodeType == 'graph')
                    {
                          $parentNode++;
                          $graphId++;
                    }
              }
           $nodeId = api_tree_item_save(0, $treeId, $itemType, $parentNode,
$name, $graphId, $rra_id, $hostId, $hostGroupStyle, 1, false);
           printf("Added Node node-id: (%d)\n", $nodeId);
           }
        }
        elseif($numHead == 0);
        {
           $nodeId = api_tree_item_save(0, $treeId, $itemType, $parentNode,
$name, $graphId, $rra_id, $hostId, $hostGroupStyle, 1, false);
        printf("Added Node node-id: (%d)\n", $nodeId);
        }
```

That should be all of the code modification that is required to make add_tree.php work as instructed above. Once again, it is much easer to take the files that have already been edited, but if they no longer exist, then downloading bulk_import and editing with my PHP code will produce the same results.