

Ganglia How-to for ITS

Emma Roos
emma@fukt.bth.se

Table of Contents

Table of Contents	2
1 Brief introduction.....	3
2 Requirements.....	3
3 Downloading	3
4 Installing with RPMs.....	4
<i>4.1 Introduction.....</i>	<i>4</i>
<i>4.2 Installing the monitoring core.....</i>	<i>4</i>
4.2.1 Gmond.....	4
4.2.2 Gmetad.....	5
4.2.3 Web frontend.....	5
<i>4.3 Installing the execution environment.....</i>	<i>5</i>
5 Installing from source	6
<i>5.1 Introduction.....</i>	<i>6</i>
<i>5.2 Compiling the monitoring core.....</i>	<i>6</i>
<i>5.3 Installing the web frontend.....</i>	<i>7</i>
<i>5.4 Compiling the execution environment.....</i>	<i>8</i>
6 Administrating Ganglia	9
<i>6.1 Configuration files.....</i>	<i>9</i>
<i>6.2 Command line tools</i>	<i>9</i>
<i>6.3 The web frontend</i>	<i>10</i>
7 Running programs with gexec.....	12
8 Links to more information.....	13

1 Brief introduction

This how-to describes how to install, administrate and use the clustering toolkit Ganglia. For an introduction about this clustering system see the report. The instructions for the Ganglia monitoring tools in this how-to (Ganglia Monitor Daemon (gmond), Ganglia Meta Daemon (gmetad) and Ganglia Metad Web Interface) is for version 2.5.0 and above, while for the execution environment (Authentication Daemon (authd) and Ganglia Execution system (gexec)) there hasn't been any new releases lately and therefore the instructions are for the latest version that existed by the time this how-to was written.

2 Requirements

The monitoring core can run on several sorts of architectures: Linux (Intel x86, Athlon, SPARC, Alpha, PowerPC and some others), Solaris, FreeBSD, AIX, IRIX, Tru64, HPUNIX, MacOS X and Windows (cygwin beta). To install with RPMs, a RedHat system running on an Intel x86 or Athlon machine is needed though. If the monitoring core should be installed on any other kind of system then it needs to be compiled from source.

The monitoring core needs multi cast enabled on the machines it is going to be installed on. You can see if multi cast is enabled by checking if the file `/proc/net/igmp` exists. Multi cast is usually enabled in the standard kernel for Linux but if its not then you have to recompile the kernel with it enabled.

The web frontend needs to be installed on a machine that runs a web server. This machine also needs RRD Tool (<http://www.rrdtool.com/>) installed and the web frontend RPM currently only works on Intel x86 machines.

The execution environment currently only seem to work on Intel x86 and Athlon machines. The packages for xinetd and openSSL also need to be installed on the system.

3 Downloading

The different parts of Ganglia can be downloaded from:
<http://ganglia.sourceforge.net/downloads.php>.

If you are going to install via RPMs then there are two RPMs for the monitoring core, one with gmond and another with gmetad (Intel 386), that should be downloaded. There is one RPM for the web interface (Ganglia Gmetad Web Frontend (for no specific architecture)). When it comes to the execution environment you need the Authd and Gexec RPMs (either for Intel 386 or Athlon).

If you are going to compile Ganglia then you should choose the tar.gz files instead. The difference in files when it comes to the tar.gz's is that the monitoring core only has one tar.gz that includes both gmond and gmetad. The execution environment needs LibE and therefore this tar.gz should be downloaded too plus that, if you want gexec to talk to the monitoring

core, you also need the Monitoring Core Library (libganglia), this library only exists as an RPM (for Intel x86 and Athlon).

4 Installing with RPMs

4.1 Introduction

The files mentioned in the installation instructions are the versions that were listed on the Ganglia download web page at the time this how-to was written. This section only describes the installation on a RedHat system with RPMs. How to compile Ganglia is described in the next chapter in this how-to.

The latest versions of the monitoring core's RPMs doesn't seem to have gexec support enabled so if you want to run gexec on the machine you are currently installing the monitoring core on you have to compile it manually with gexec enabled as described in the next chapter. This problem might be fixed in the future.

4.2 Installing the monitoring core

4.2.1 Gmond

First install gmond on all the machines that you want to monitor with Ganglia with the rpm command (If you are updating the package change the i option to the U option.):

```
rpm -ivh ganglia-monitor-core-gmond-2.5.1-1.i386.rpm
```

The configuration for gmond exists in a file called */etc/gmond.conf*. The default configuration works on most clusters but sometimes its necessary to change the settings in this file. The different settings are described in the file just above the specific setting. To use a setting just remove the comment tag (#) in front of it and change it to the value you want. For example if you want to change the name of the cluster remove the # from:

```
# name "My Cluster"
```

And change the *My Cluster* to what you want the cluster to be called. When this is done the line should look like this:

```
name "The name I've chosen".
```

If you like to test a configuration file you've created you can use the gmond command's --conf option. For example if you've created a file named *gmond-test.conf* that lies in the directory */etc* and wants to start the gmond daemon with these settings, then first shut down the currently running gmond daemon with */etc/init.d/gmond stop* and then start it with the new configuration with the command:

```
gmond --conf=/etc/gmond-test.conf
```

4.2.2 Gmetad

After that gmond is installed it's time to install gmetad on the machine/machines that you want to use to monitor the system (there among the machine that you want to run the web interface on) with the rpm command (If you are updating the package change the i option to the U option.):

```
rpm -ivh ganglia-monitor-core-gmetad-2.5.1-1.i386.rpm
```

The configuration of gmetad is in the file */etc/gmetad.conf* and just like */etc/gmond.conf* a description of the different settings exists above the said setting. Before gmetad can run properly you will need to define the "data sources" in the */etc/gmetad.conf* file to tell gmetad which gmond daemons to monitor. The format of the data_source lines should look like this:

```
data_source "my cluster" localhost host1 host2:port
```

There "my cluster" is the name of the cluster defined in */etc/gmond.conf* and the options after that are the different hosts running gmond that you want to use to check the cluster. Gmetad will check the first host in the list and if that host doesn't respond it will try the second host listed etc. If you don't define a port then gmetad will try the default port defined by the configuration files.

After that the data_source setting has been defined start gmetad with */etc/init.d/gmetad start*.

4.2.3 Web frontend

Install the web frontend RPM on the machine that you want it to run on, don't forget to have a web server on the machine, with the rpm command:

```
rpm -ivh ganglia-webfrontend-2.5.1-1.noarch.rpm
```

Test your installation by visiting the URL: *http://webserver/ganglia-webfrontend/index.php* in a web browser where *webserver* is the host name or IP of the machine that you installed the web frontend on.

4.3 Installing the execution environment

Before authd and gexec are installed on the nodes a pair of SSL keys needs to be created. This is done with the commands:

```
openssl genrsa -out auth_priv.pem  
chmod 600 auth_priv.pem  
openssl rsa -in auth_priv.pem -pubout auth_pub.pem
```

After you have done this copy auth_pub.pem to the */etc* directory on all the nodes that will receive gexec requests (most likely all the nodes on the cluster). This can be done with the command scp:

```
scp auth_pub.pem node1:/etc/auth_pub.pem
scp auth_pub.pem node2:/etc/auth_pub.pem
```

Then copy `auth_priv.pem` to all the nodes that are going to launch gexec requests (the machines in the cluster that users can log on to and run programs on). This can also be done with the `scp` command:

```
scp auth_priv.pem node1:/etc/auth_pub.pem
```

The machines that you will launch gexec requests on will need to have both `auth_priv.pem` and `auth_pub.pem`.

Now `authd` needs to be installed on all the machines that you want to run gexec on. This you do with the `rpm` command:

```
rpm -ivh authd-0.2.1-1.i386.rpm
```

Then its time to install gexec in the same way:

```
rpm -ivh gexec-0.3.4-1.i386.rpm
```

After that add gexec in `/etc/services` under `# Local services`, the line should look like this:

```
gexec      2875/tcp      # Caltech gexec
```

5 Installing from source

5.1 Introduction

The files mentioned in the installation instructions are the versions that were listed on the Ganglia download web page at the time this how-to was written.

5.2 Compiling the monitoring core

First unpack the tar.gz with the source that you've downloaded:

```
tar -xvzf ganglia-monitor-core-2.5.1.tar.gz
```

Then go into the directory that got created:

```
cd ganglia-monitor-core-2.5.1
```

If you are going to compile `gmetad` then you have to make sure that the configure script will find RRD Tool's library file `librrd.a` and header file `rrd.h` first. For the script to find these files they have to be found in `/usr/local/lib` (for `librrd.a`) and `/usr/local/include` (for `rrd.h`), this can be done by doing a soft-link to these files in these directories. For example if RRD Tool is installed in `/usr/local/rrdtool` you should do something like this:

```
ln -s /usr/local/rrdtool/include/rrd.h /usr/local/include/  
ln -s /usr/local/rrdtool/lib/librrd.a /usr/local/lib/
```

Now you should run the configuration script. If you want to install gmetad then add the `--with-gmetad` flag as an option and if you want gexec to run on the machine (and have the monitoring core aware of it) you also have to add the `--enable-gexec` flag. The command line should look something like this:

```
./configure --with-gmetad --enable-gexec
```

Now run the command `make` and the programs will compile. Then run the command `make install` and after that copy the configuration file to the directory `/etc/`:

```
cp ./gmond/gmond.conf /etc/
```

Edit the file if necessary and then it's time to do so that gmond is started at reboot:

```
cp ./gmond/gmond.init /etc/init.d/gmond  
chkconfig --add gmond  
/etc/init.d/gmond start
```

If you are installing on the machine that you are planning to run the web frontend on then you also have to install gmetad. First copy the configuration file to `/etc/`:

```
cp ./gmetad/gmetad.conf /etc/
```

Edit the different settings in the file if necessary and add the `data_source` line in the same way as described in chapter 4.2.2. Then make so that gmetad can run and is started at reboot:

```
mkdir -p /var/lib/ganglia/rrds  
chown -R nobody /var/lib/ganglia/rrds  
cp ./gmetad/gmetad.init /etc/init.d/gmetad  
chkconfig --add gmetad  
/etc/init.d/gmetad start
```

5.3 Installing the web frontend

See that gmetad, a web server and RRD Tool exists on the machine that you want the web frontend on and if they aren't there install them. Also make sure that the web server knows how to handle PHP (for Apache the `mod_php` module needs to be enabled).

Copy the tar.gz with the web frontend files that was downloaded into your website tree (usually `/var/www/html/`) and then unpack the file there:

```
cp ganglia-webfrontend-2.5.1.tar.gz /var/www/html/  
cd /var/www/html/  
tar -xvzf ganglia-webfrontend-2.5.1.tar.gz
```

Test your installation by visiting *http://webserver/ganglia-webfrontend/* in a web browser (*webserver* is the hostname/IP of your web server).

5.4 Compiling the execution environment

You first need to install libE. Unpack the tar.gz with the libE source you've downloaded:

```
tar -xvzf libe-0.2.2.tar.gz
```

Then go into the library that got created:

```
cd libe-0.2.2
```

Now run the configure script:

```
./configure
```

Then run the command *make* and after that install libE with *make install*.

When libE had been installed it's time to compile authd. First the SSL keys should be created and put in */etc/* as described in chapter 4.3. After this it's time to compile authd. First unpack the tar.gz with the authd source that you've downloaded:

```
tar -xvzf authd-0.2.1.tar.gz
```

After that go into the directory created (authd-0.2.1) and run the configure script:

```
./configure
```

Then run the command *make* and after that install authd with *make install*.

When authd has been installed it's time for gexec. First libganglia needs to be installed, this library only exists as a RPM package so install the RPM packages you've downloaded:

```
rpm -ivh ganglia-monitor-core-lib-2.5.1-1.i386.rpm
```

Now unpack the gexec source tar.gz:

```
tar -xvzf gexec-0.3.4.tar.gz
```

Now go into the directory created by the command:

```
cd gexec-0.3.4
```

Before you run the configure command and if you are going to run the monitoring core on the machine (and want gexec to interact with it) you first have to copy the directory *lib/ganglia/* from the monitoring core source directory into the directory created by the gexec tar.gz. For example if both the monitoring core (version 2.5.1) and gexec (0.3.4) tar.gz-files were

unpacked in the same directory (for example `/var/tmp`) you should copy the directory like this:

```
cp -R /var/tmp/ganglia-monitor-core-2.5.1/lib/ganglia /var/tmp/gexec-0.3.4/
```

This will create a directory called `ganglia` in the `gexec-0.3.4` directory with the same contents as the `ganglia-monitor-core-2.5.1/lib/ganglia` directory.

Now its time to run the configure script (with the `--enable-ganglia` flag if you are going to run the monitoring core on the same machine):

```
./configure --enable-ganglia
```

After this run the command `make` to compile `gexec` and the install it with `make install`. Last you should edit `/etc/services` and add under `# Local services` the line:

```
gexec      2875/tcp      # Caltech gexec
```

6 Administrating Ganglia

6.1 Configuration files

As described under the installation instructions the configuration for `gmond` and `gmetad` exists in the files `/etc/gmond.conf` and `/etc/gmetad.conf`. In these files the different settings are also described quite well.

When it comes to the web frontend most of its configuration options exists in the file `conf.php` that lies in the `ganglia-webfrontend` directory that lies under the web server's main directory for web pages (commonly `/var/www/html/`). There you can for example change the location of `gmetad` and `RRD Tool` if they are installed elsewhere, you can also change how the different graphs should be displayed and some other settings. All these settings are described with comments in the file.

6.2 Command line tools

The monitoring core has two command line tools: The Ganglia Metric client and The Ganglia Status Client.

The **Ganglia Metric Client** (`gmetric`) is used to get different metric values from the different machines running `gmond` (that broadcasts on the same multi cast channel). The program formats and sends a special multi cast message to all machines with `gmond` that are listening.

All the metrics have: a name (option: `-n` or `--name`), value (option: `-v` or `--value`), type (option: `-t` or `--type`) and optionally units (option: `-u` or `--units`).

The name is what the metric should be called, value is the value of the metric generated by a program that exist on the machine, type defines what kind of data type the metric is in (string, 8, 16 and 32 bit integer (int8 etc), 8, 16 and 32 bit unsigned integer (uint8 etc), float and

double) and units print what the unit of the metric should be called. For example if there is a program called `cputemp` that shows the temperature of the machine's CPU in Celsius, then to send this metric to all the listening gmonds type:

```
gmetric --n temperature --v 'cputemp' -t int16 -u Celsius
```

To continuously check this metric you have to add it to the systems cron table (Read the man pages for crontabs with *man crontab* and *man 5 crontab* for more information how to do this).

There are also other options to the `gmetric` command. What these are and what they do can be seen by typing *gmetric --help*.

With the **Ganglia Status Client** (`gstat`) you can get a status report for the cluster. If you only type `gstat` you will get a small status report on the cluster looking something like this:

CLUSTER INFORMATION

```
    Name: Exjobbskluster
    Hosts: 2
Gexec Hosts: 0
Dead Hosts: 0
    Localtime: Wed Dec 4 14:02:12 2002
```

CLUSTER HOSTS

```
Hostname                LOAD                CPU                Gexec
CPUs (Procs/Total)  [  1,  5, 15min]  [ User, Nice, System, Idle]
```

ii

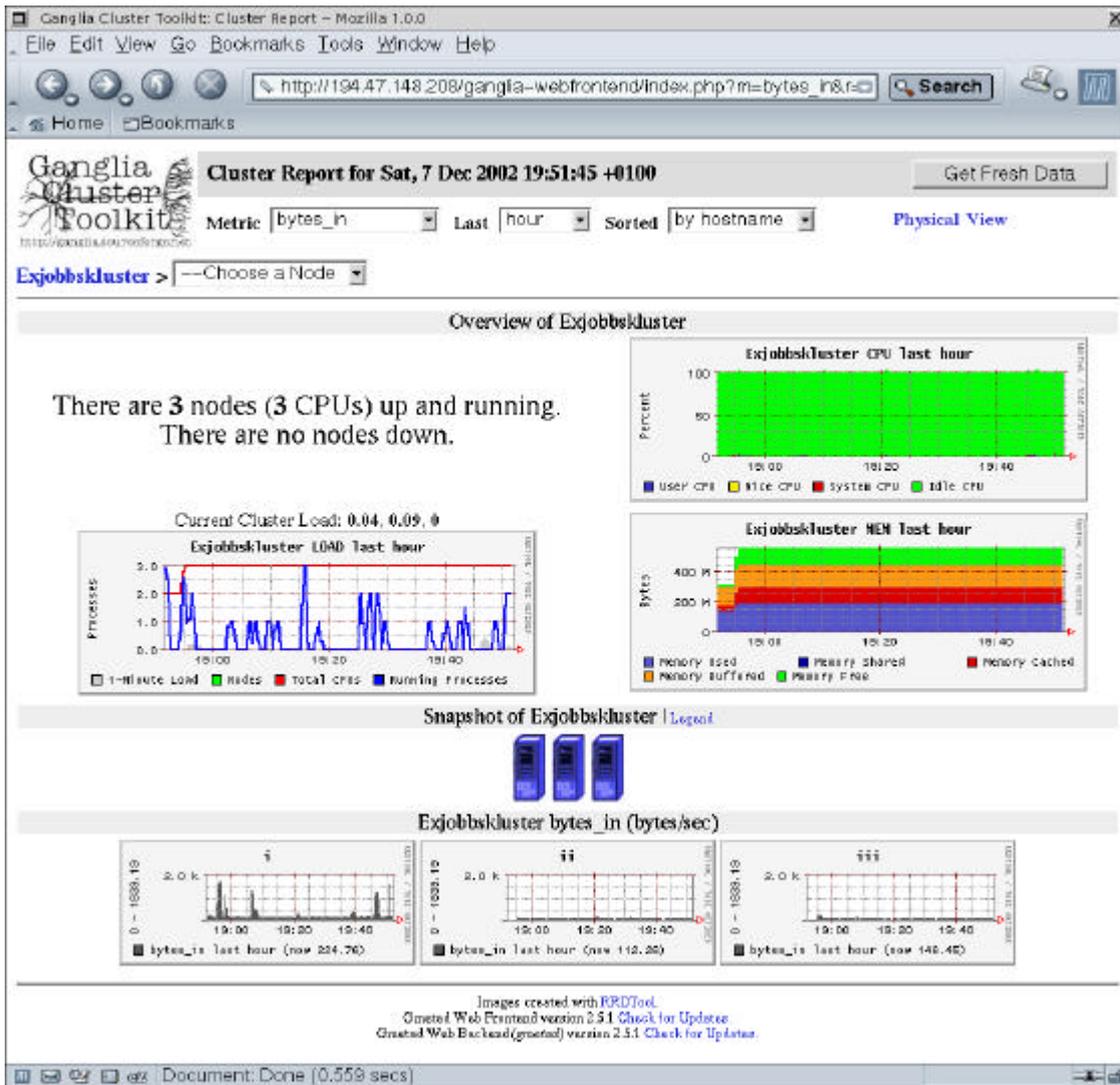
```
  1 (  0/ 50)  [ 0.00, 0.00, 0.00] [ 0.1, 0.0, 0.2, 100.0]  ON
```

To get the syntax for the command use the help option: *gstat --help*. Some of the common options are:

- a (or --all) lists all the hosts (not just the ones running gexec)*
- d (or --dead) lists only the hosts that are dead*
- l (or --list) prints only the host list (the hosts running gexec)*

6.3 The web frontend

The web frontend is a website that show different statistics and graphs of the cluster. On the main web page (`index.php`) you get an overview of the entire cluster:



The three graphs highest up on the web page shows the load, CPU usage and memory usage on the entire cluster the latest hour. The link "Physical View" just under the "Get Fresh Data" button goes to a page that lists the different hosts with some hardware information about them.

The graphs at the bottom of the web page shows the metric chosen, in the metric drop box at the top of the page, for all the nodes in the cluster. The other two drop boxes on the same row as the metric drop box defines over what time period (last hour, day, week, month, year) the graphs should be and in what order. The fourth drop box is used to choose a node and go to a web page that shows specific metrics and settings for that node, you can also get to this page by clicking on the metric graph for the specific machine or the small computer picture above the graphs.

On the web page with information about a specific node that you get to by doing as described above shows all the different metrics that gmond is monitoring. Some of the metrics are show as text ("Time and String Metrics" and "Constant Metrics") while the rest of them are shown in several graphs on this web page. There is also a drop box that you can use to change over what time period (last hour, day, week, month, year) the different graphs should be shown.

There is also a link, Host view, that goes to another page that show the hosts hardware- and software- configuration plus some other information.

7 Running programs with gexec

To run a program on several nodes the gexec command can be used. The main syntax for the command is: *gexec -n <number of nodes> command* where: *number of nodes* is the amount of nodes that the *command* you want to run should spawn on (where 0 is all nodes). For example if you want to see the uptime of the three first nodes that gexec know about you can do this with the command:

```
gexec -n 3 uptime
```

You will then for example get the output (the number in front of the output of the command is the virtual node number (vnn)):

```
0 10:26am up 16:38, 2 users, load average: 0.01, 0.01, 0.00  
2 11:32am up 5 days, 21:52, 1 user, load average: 0.00, 0.00, 0.00  
1 10:40am up 5 days, 21:54, 0 users, load average: 0.36, 0.28, 0.26
```

For gexec to know what nodes to use its supposed to interact with gmond to get a list of hosts running gexec. Unfortunately this doesn't seem to work with the latest versions of the monitoring core but hopefully it will be fixed in the future. Because of this the hosts running gexec (or the hosts that you want at the moment to run the programs you start) needs to be defined via the environment variable GEXEC_SVRS. To set this variable use the command export (or setenv if you use tcsh):

```
export GEXEC_SVRS="node1 node2 node3"
```

If the communication between gexec and gmond had worked properly (this might work with future releases and it does work with version 2.4.* of the monitoring core) you can also manually define the gmond servers that you want gexec to ask for nodes running gexec. This environment variable is GEXEC_GMOND_SVRS. You also set this variable with the export command:

```
export GEXEC_GMOND_SVRS="node1 node2:port"
```

Gexec also has a few other options the most useful of these are:

-h (--help)

Prints the syntax of the command

-d (--detached)

Gexec will spawn the jobs and then detach. This option is good if you for example are starting up daemons or other programs that should be running in the background.

-p (--prefix-type) [none | ip | vnn | host]

This is the string that is before the output from gexec. The default is vnn (the unique identification number the node has), IP is the IP address of the node and host is the node's host name.

-P (--port)

This option is used if `gexec` on the different machines is running on another port than the one described in `/etc/services` on the machine that the command is run on.

8 Links to more information

The official Ganglia web page: <http://ganglia.sourceforge.net/>

The authd web page: <http://www.theether.org/authd/>

The gexec web page: <http://www.theether.org/gexec/>

The RRD Tool web page: <http://www.rrdtool.com/>