Ganglia & C3

Thomas Naughton
Oak Ridge National Laboratory

Ganglia for monitoring

- gmond – A multithreaded daemon which acts as a server for monitoring a host.
- A number of clients are available.
- gstat and ganglia for basic queries.
- gmetric for setting up custom monitoring scripts.
- Php/rrd web client. (graphical)

gmond, a few specifics

- Each gmond stores all of the information for the entire cluster locally in memory.
- Opens up port 8649 and a telnet to this port will result in a dump of all the information stored in memory. (XML formatted)
- Additionally, when a change occurs in the host that is being monitored, the gmond multicasts this information to the other gmonds.

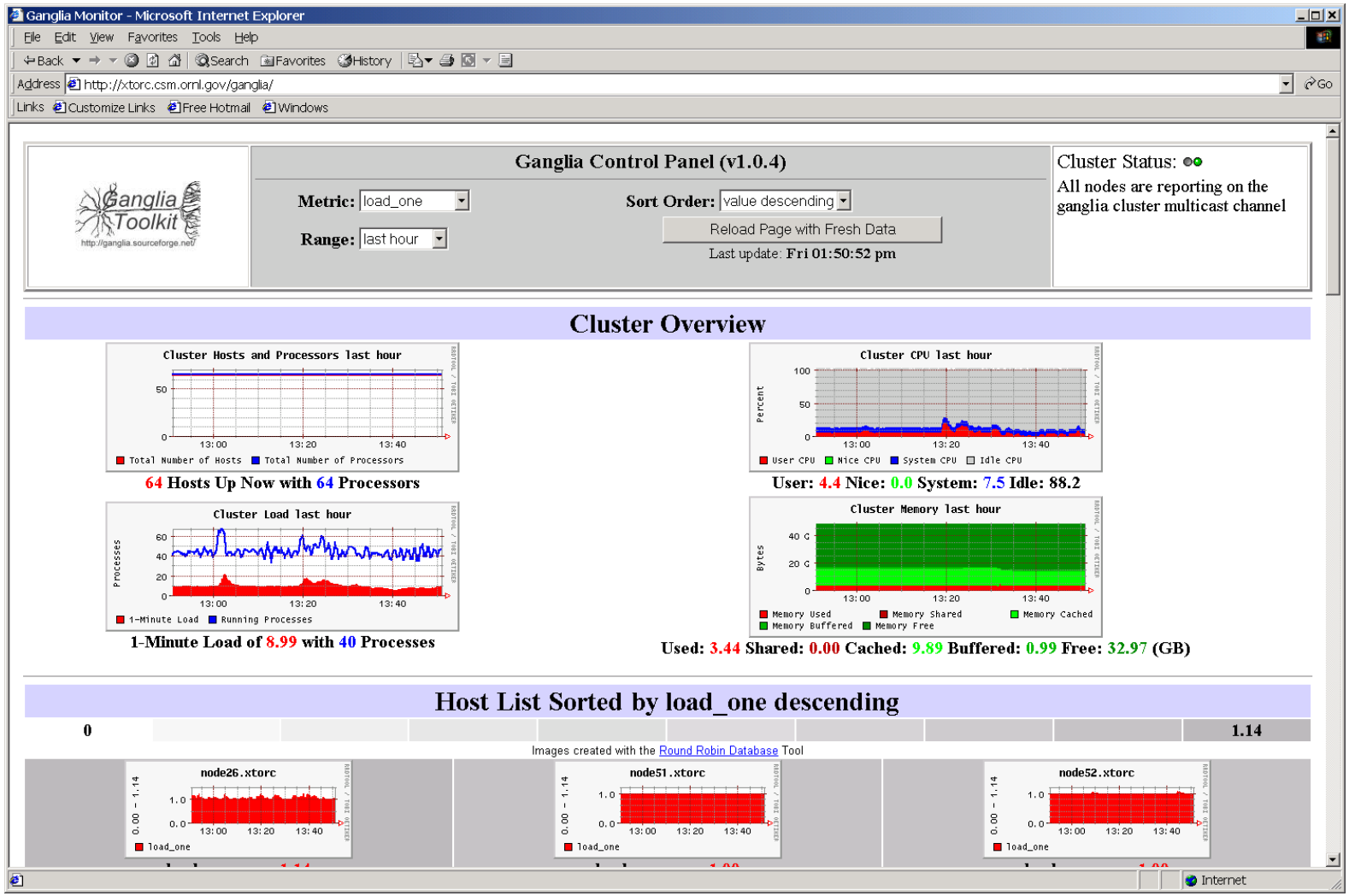
gstat

- The simplest commandline client available.
- gstat, shows all nodes with basic load info.
- gstat --help, shows general options
- gstat --dead, shows dead nodes
- gstat -m, lists the nodes from least to most loaded.

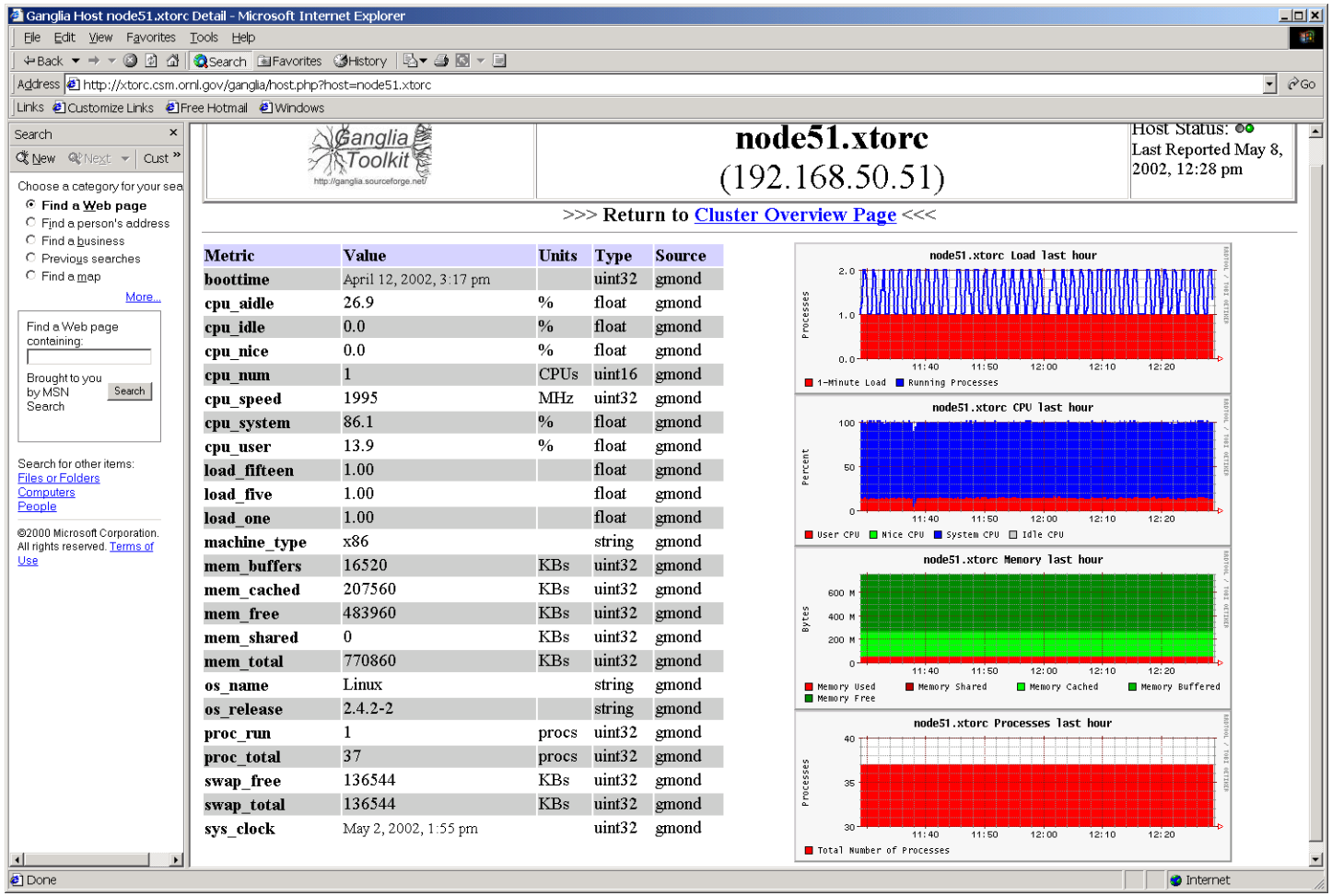
ganglia is for specific metrics

- A python script, added for convenience.
- It is both an executable and a class (library).
- gmond monitors 15 metrics by default.
- ganglia --help to see the metrics.
- To run: ganglia metric metric …
- Example: ganglia cpu_nice

The web client: php/rrd



Displaying all the metrics.



gmond across multiple clusters

- gmond --trusted_host xxx.xxx.xxx.xxx
- Allows setting up a unicast connection to another gmond across the Internet.
- Must do this on each gmond, so that the communication is 2-way.

Ganglia Summary

- gmond is scaleable because of its use of multicast.
- gmond is usefull, as it allows realtime information gathering of which hosts are alive, before running a job.
- Available at: ganglia.sourceforge.net
- Now an included package in OSCAR.

Ganglia / C3 Example

Description of *sync_users*

The *sync_users* script that ships with OSCAR is a very simple example usage of *cpush* to distribute the files:

/etc/[passwd,group,shadow,gshadow]

to the nodes manually or via a cron entry.

Statement of Problem

The default *sync_users* script is very simple and a very annoying characteristic is that it stalls when any of the nodes are down. (Stalls until the SSH timeout for that node.)

All available nodes roll by perfectly but the script pushes 2-4 files and the stall happens at the end of each *cpush* (file). Therefore if the timeout is 2 minutes, it could hang for ~8 minutes if no CTRL+C is applied.

Re-Statement of Problem

Need some way to dynamically determine the "down nodes" and skip them when running *sync_users*.

Also, need to display the list of missed nodes.

Enter Ganglia…

Same day the *sync_user* discussion took place Ganglia was demo'd by a group member.

Ganglia maintains information about nodes in the cluster and most relevantly it offers a nice tool, *gstat*, with options to list available nodes and their load!

Quick *sync_users2*

So, a quick *sync_users2* was whipped up using Ganglia's *gstat* in conjunction with C3's *cpush* to make a "smarter" script.

The script uses output from *cnum* and *gstat -m*

The output is massaged to build the *cpush* command-line and to clearly report missed nodes.

Usage example

Things could <u>almost</u> be done from a command-line like this:

```
root: # gstat -m > upnodes.tmp && \
> cpush -l upnodes.tmp /etc/passwd && \
> rm upnodes.tmp
#repeat for all files: passwd,group,shadow,gshadow
```

Instead…just type:

```
root: # ./sync_users2
```

Perl Script Summary

Build a hash of the default cluster in the c3.conf file (use *cnum*)

```
%c3conf = munge_c3conf("/opt/c3-3/cnum"); #name->num
```

Get list of Up/Avail nodes (via Ganglia)

```
@uplist = get_nodelist("/usr/bin/gstat -m");
```

Munge standard nodelist into C3-3 format nodeN->:N

```
@c3nodelist = c3ify_nodelist($aref_uplist, $href_c3conf);
```

Build C3-3 cmd-ln nodelist

```
$nodes = ":" . join(",", @{$c3nodelist});
```

Distribute the files with the above cmd-ln nodelist

```
`cpush  $nodes  /etc/passwd`;
```

Print missed nodes info

```
@missed = get_missednodes($aref_uplist, $href_c3conf);
print "\n  Missed nodes:\n  @missed  \n";
```

Ganglia / C3: Comments

This is just a simple application of C3 and Ganglia.

The goal was to use these two tool to create a "smarter *sync_users*"… this has been met.

Since C3 and Ganglia can be use by standard users (not just root) this method could be used by anyone for user-level scripts.