Monitoring

HEPSYSMAN June 2013

David Crooks on behalf of Scotgrid Glasgow

Monitoring

- Motivation
- Scope
- Graphite
- Data sources
- Next steps

University of Glasgow

ScotGrid

Motivation

- Some grumbling issues with Ganglia at Glasgow (probably around multicast over network upgrades/changes in configuration)
- Upcoming cluster refresh
- Monitoring Core Ops
- Good time to take a fresh look at monitoring

GridPP UK Computing for Particle Physics

Scope

- Exploring monitoring options
- Not necessarily replacing Ganglia
- Not using active monitoring - not replacing Nagios (but allowing for interfacing)
- What can we use monitoring for?
- What data can we mix in from external sources?

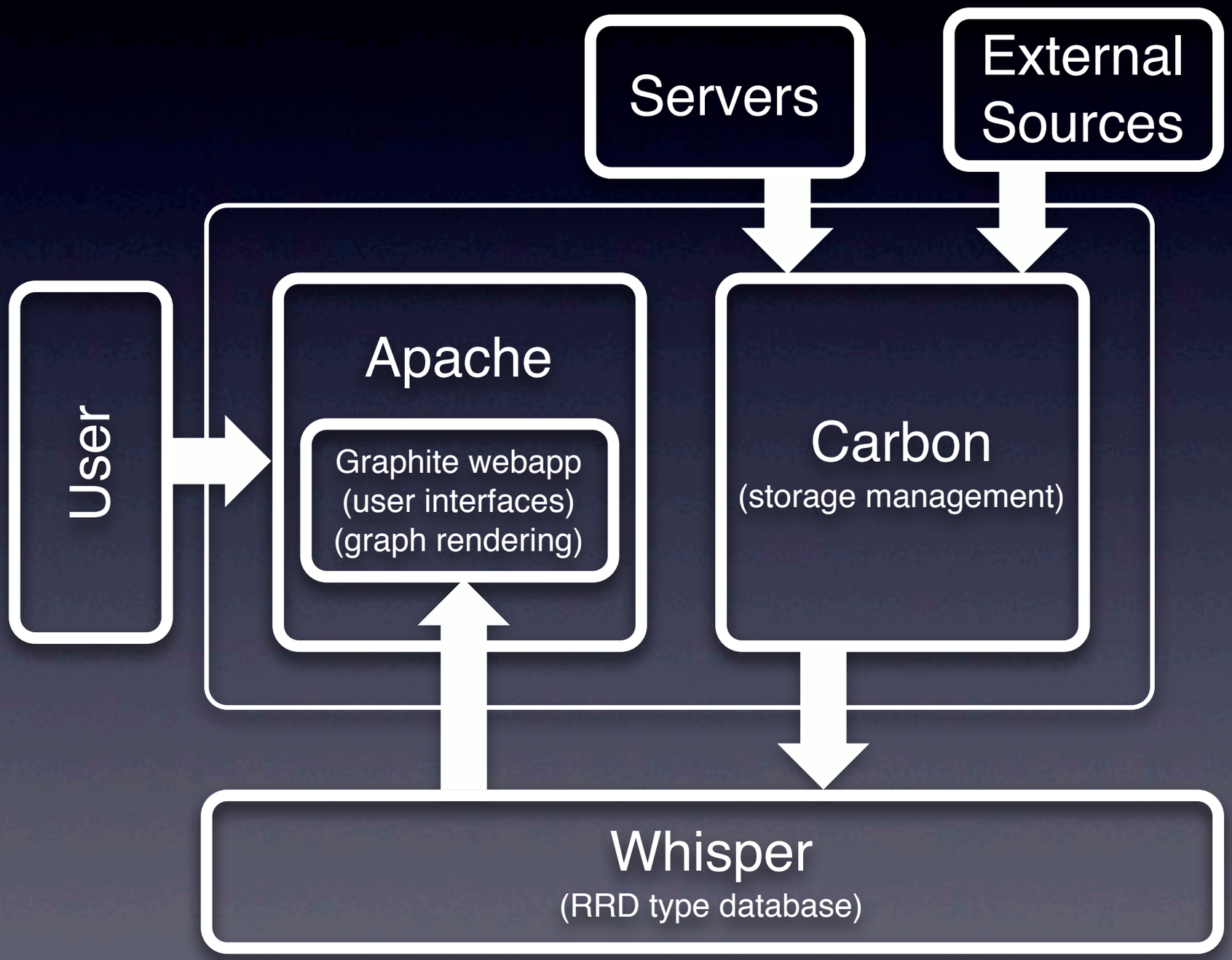
Graphite

- Explored Graphite

URL
`http://graphite.readthedocs.org`

- Popular in wider community
- Lots of add-ons
- Multiple dashboards
- Good visualisations
- Very straightforward to effectively get arbitrary data in and out

Graphite



Modified from http://graphite.wikidot.com

Graphite

- Installation

URL
`http://graphite.readthedocs.org/en/0.9.10/install.html#id2`

  - Originally installed from `epel-testing` but now in `epel` proper

Bash
```
yum install graphite-web python-carbon python-whisper
python /usr/lib/python2.6/site-packages/graphite/manage.py syncdb
yum install -y liberation*
fc-cache
chkconfig carbon-cache on
chkconfig httpd on
```
for font installation only

Messaging

- Graphite message format

```
<metric> <value> <timestamp>

ukhep.hepsysman.users 40 1369827513
```

- Messaging Carbon

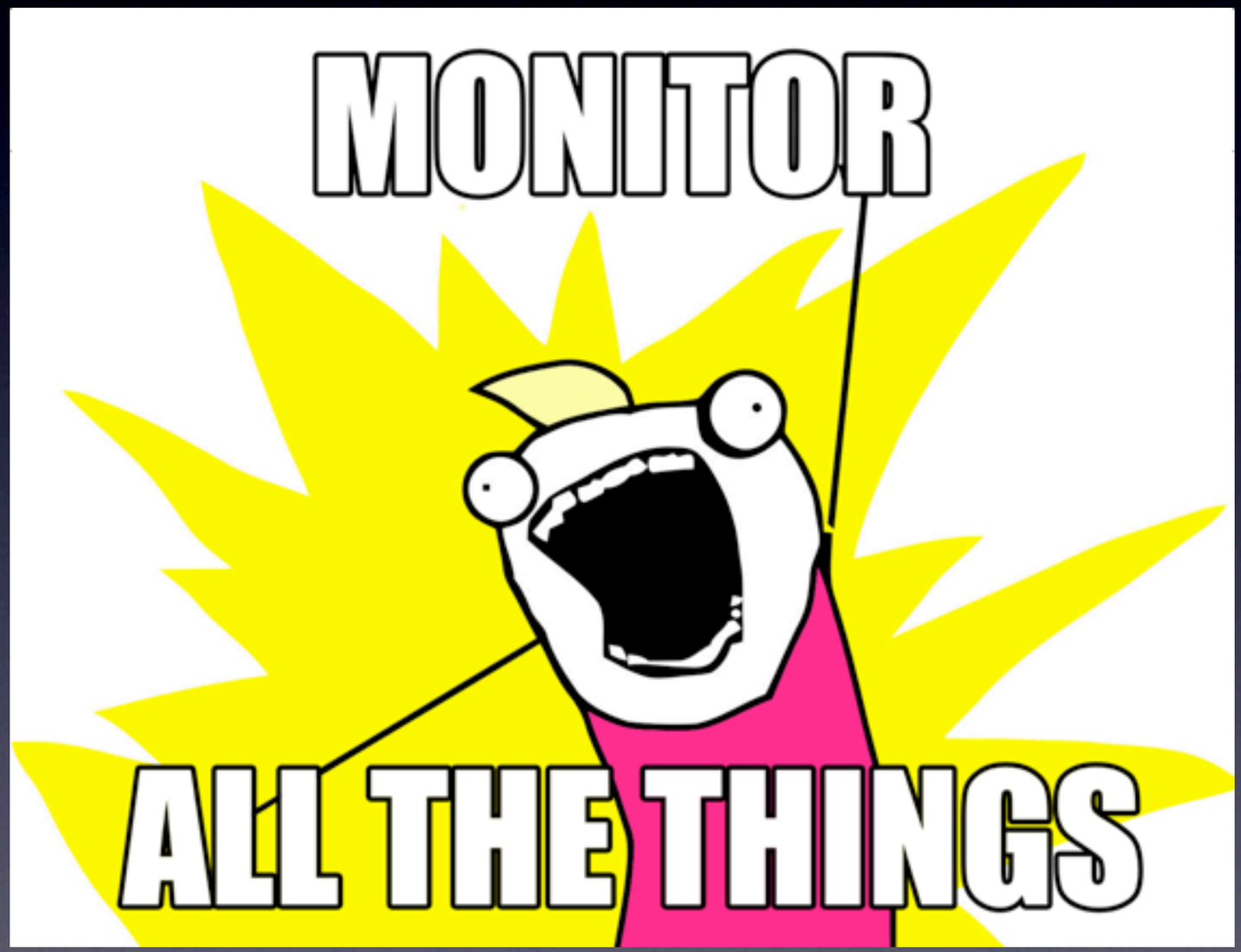
Bash

```
echo “ukhep.hepsysman.users 40 1369827513” | nc <carbon-server> <port>
```

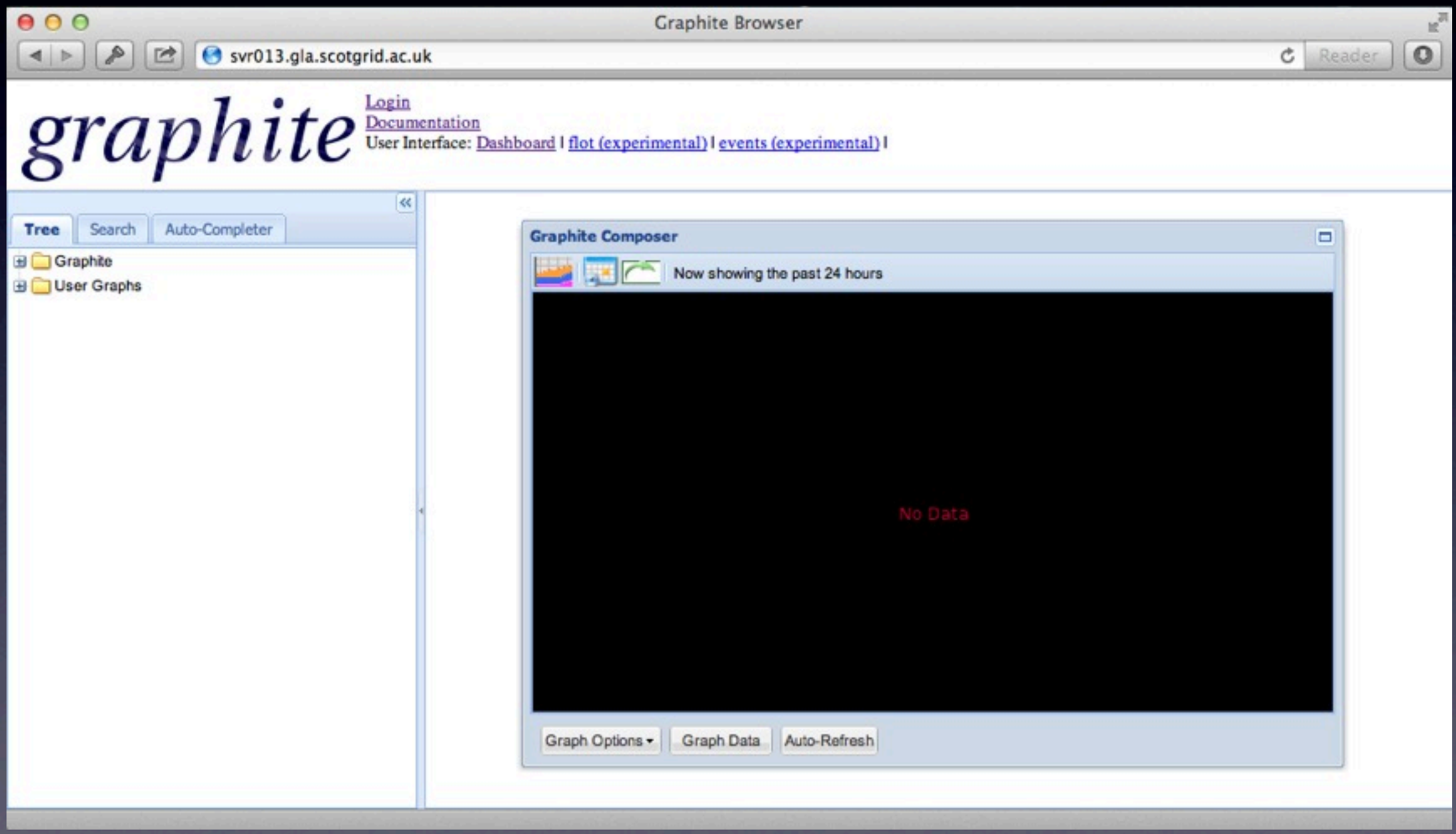
Python

```
sock.connect( (<carbon-server>, <port>) )
...
lines.append("ukhep.hepsysman.users %s %d" % (users,now))
...
sock.sendall(message)
```

(Additional) Motivation



Graphite Webapp

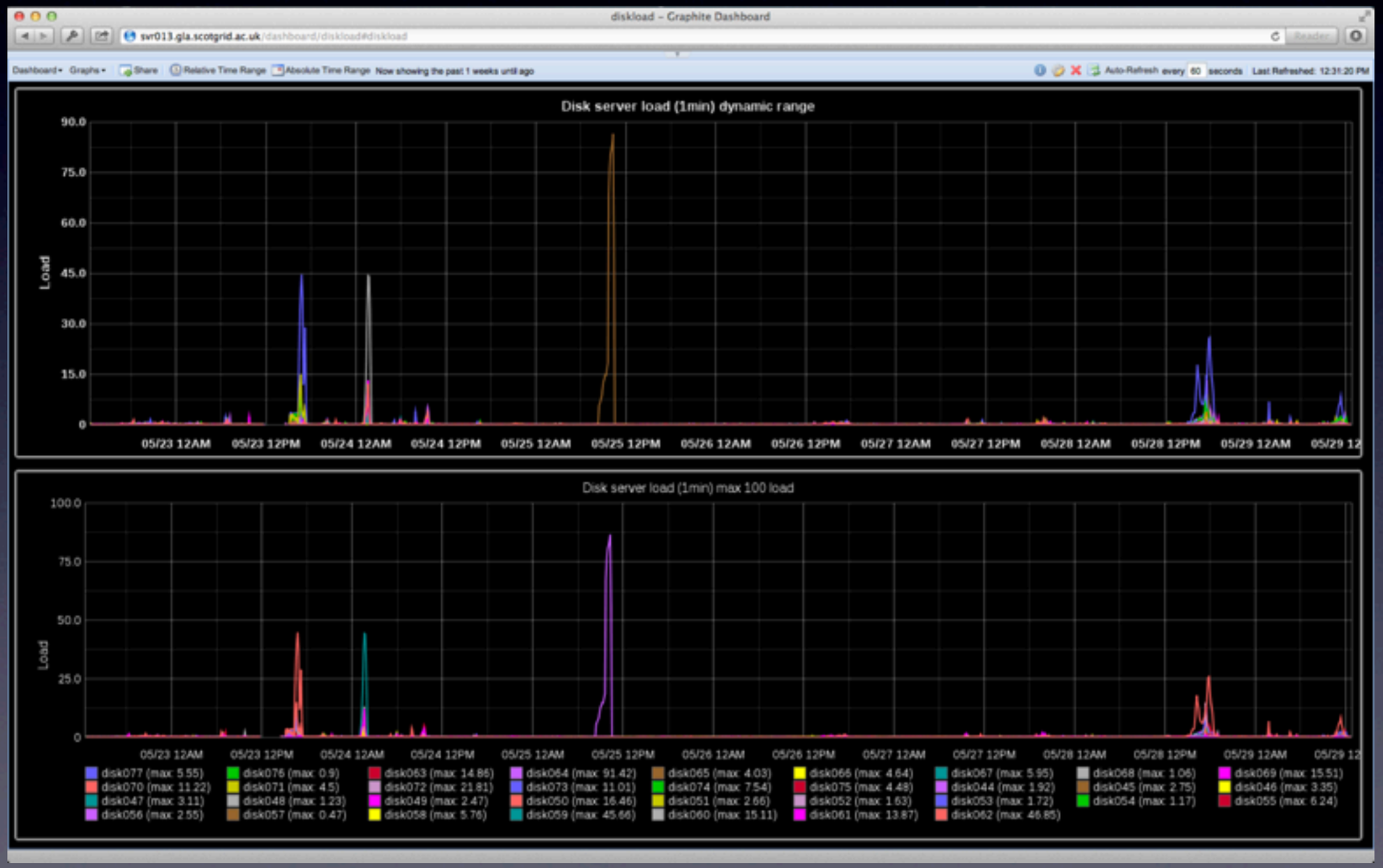


Dashboard

Data sources (internal)

- Currently using timed cron jobs
- System monitoring
  - New collectors (disk load *via* python) - not all Ganglia metrics mirrored
  - Collecting from Ganglia (currently batch data collected using brute force `netcat` method + `qstat|grep` + messaging)
- Local environment monitoring
  - Temperature (existing bash scripts + messaging)

Diskload


diskload – Graphite Dashboard
svr013.gla.scotgrid.ac.uk/dashboard/diskload#diskload
Dashboard
Graphs
Share
Relative Time Range
Absolute Time Range
Now showing the past 1 weeks until ago
Auto-Refresh every 60 seconds
Last Refreshed: 12:31:20 PM
Disk server load (1min) dynamic range
Load
90.0
75.0
60.0
45.0
30.0
15.0
0
05/23 12AM
05/23 12PM
05/24 12AM
05/24 12PM
05/25 12AM
05/25 12PM
05/26 12AM
05/26 12PM
05/27 12AM
05/27 12PM
05/28 12AM
05/28 12PM
05/29 12AM
Disk server load (1min) max 100 load
100.0
75.0
50.0
25.0
disk077 (max 5.55)
disk076 (max 0.9)
disk063 (max 14.86)
disk064 (max 91.42)
disk065 (max 4.03)
disk066 (max 4.64)
disk067 (max 5.95)
disk068 (max 1.06)
disk069 (max 15.51)
disk070 (max 11.22)
disk071 (max 4.5)
disk072 (max 21.81)
disk073 (max 11.01)
disk074 (max 7.54)
disk075 (max 4.48)
disk044 (max 1.92)
disk045 (max 2.75)
disk046 (max 3.35)
disk047 (max 3.11)
disk048 (max 1.23)
disk049 (max 2.47)
disk050 (max 16.46)
disk051 (max 2.66)
disk052 (max 1.63)
disk053 (max 1.72)
disk054 (max 1.17)
disk055 (max 6.24)
disk056 (max 2.55)
disk057 (max 0.47)
disk058 (max 5.76)
disk059 (max 45.66)
disk060 (max 15.11)
disk061 (max 13.87)
disk062 (max 46.85)

Batch

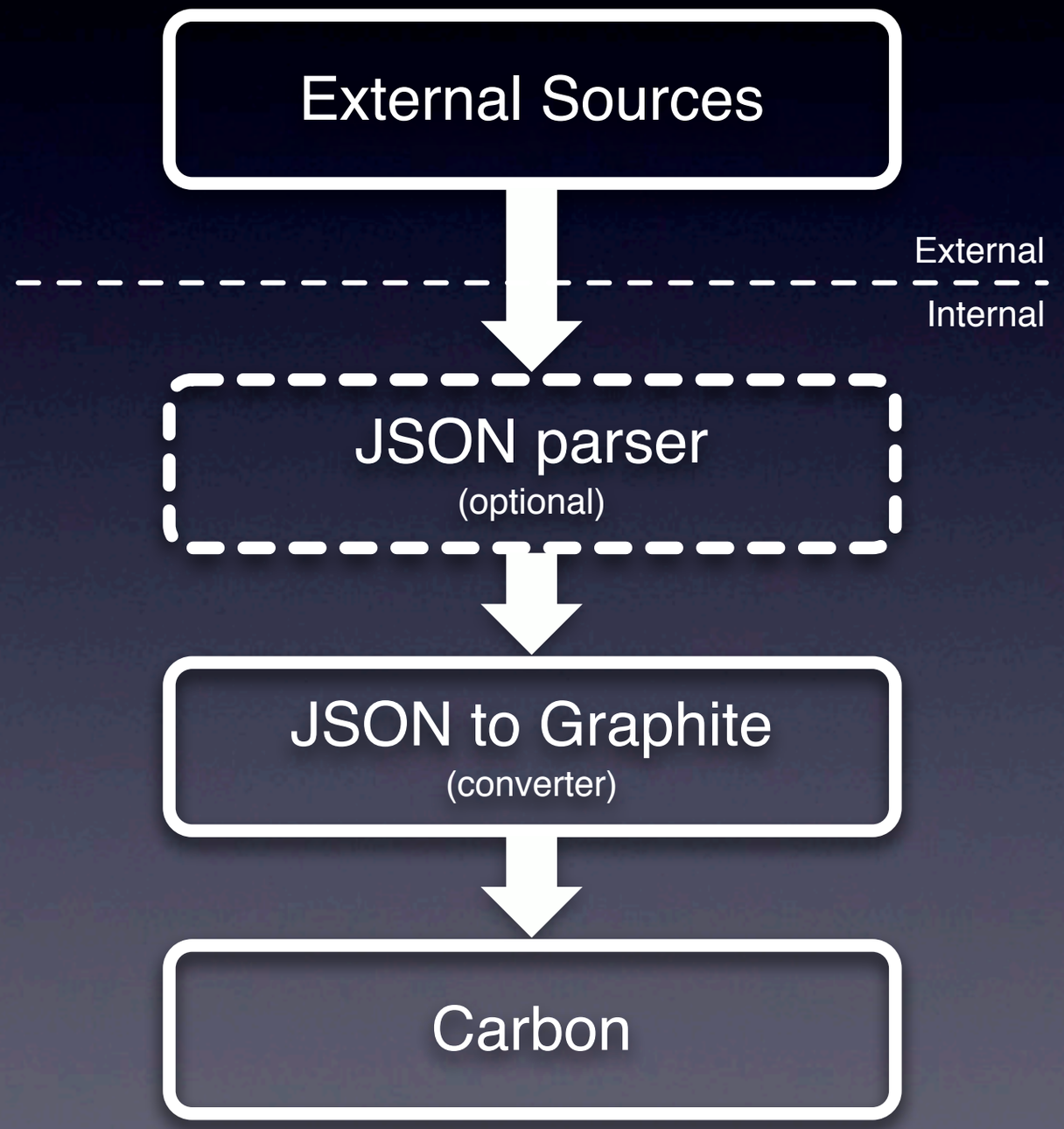

batch_detail_groups – Graphite Dashboard
svr013.gla.scotgrid.ac.uk/dashboard/batch_detail_groups#batch_detail_groups
Running jobs by Queue
All jobs by state
Job Efficiency
Queued jobs by Queue
Queued jobs by group (no ATLAS/LHCb)
Running jobs by group (no ATLAS/LHCb)

Temperature

Data sources (external)

- As well as local data sources, tested gathering data from external sources
- Could use netcat, etc., but cleaner method would be to use something like JSON (one possible output from Graphite)

Graphite JSON



Graphite JSON

- Modified version of `httpJsonStats`
- (single shot vs daemon)
- Reads remote JSON
- Currently take external JSON/CSV formats and use hand-written parsers to return following format:

JSON

```
{
        "users": 40
}
```

JSON-Graphite converter config

JSON

```
{
  "global": {
    "GRAPHITE_SERVER" : "127.0.0.1",
    "GRAPHITE_PORT"   : 2003,
    "INTERVAL"        : 600,
        "LOG_FILE"          : "/var/log/httpJsonStats.log",
        "ERR_LOG_FILE"      : "/var/log/httpJsonStats.log",
        "PID_FILE"          : "/var/run/httpJsonStats.pid"
},
"ukhep":
{
    "host": "<JSON host>",
    "port": "<port>",
    "groups": {
        "hepsysman": {
            "URN": "/json/users.json"
        }

    }
}
```

External examples - Panda

- Panda

URL
`http://dashb-atlas-job.cern.ch/dashboard/templates/web-job2/`

- XML target ⟶ JSON
- Parsed into separate job types for inclusion

Panda monitoring (JSON)

Accounting

- EGI Accounting data for Glasgow

URL
`http://accounting.egi.eu/egi.php?ExecutingSite=UKI-SCOTGRID-GLASGOW`

- Extended CSV target
- Locally parsed into JSON and split into separate VOs

EGI Accounting (CSV→JSON)

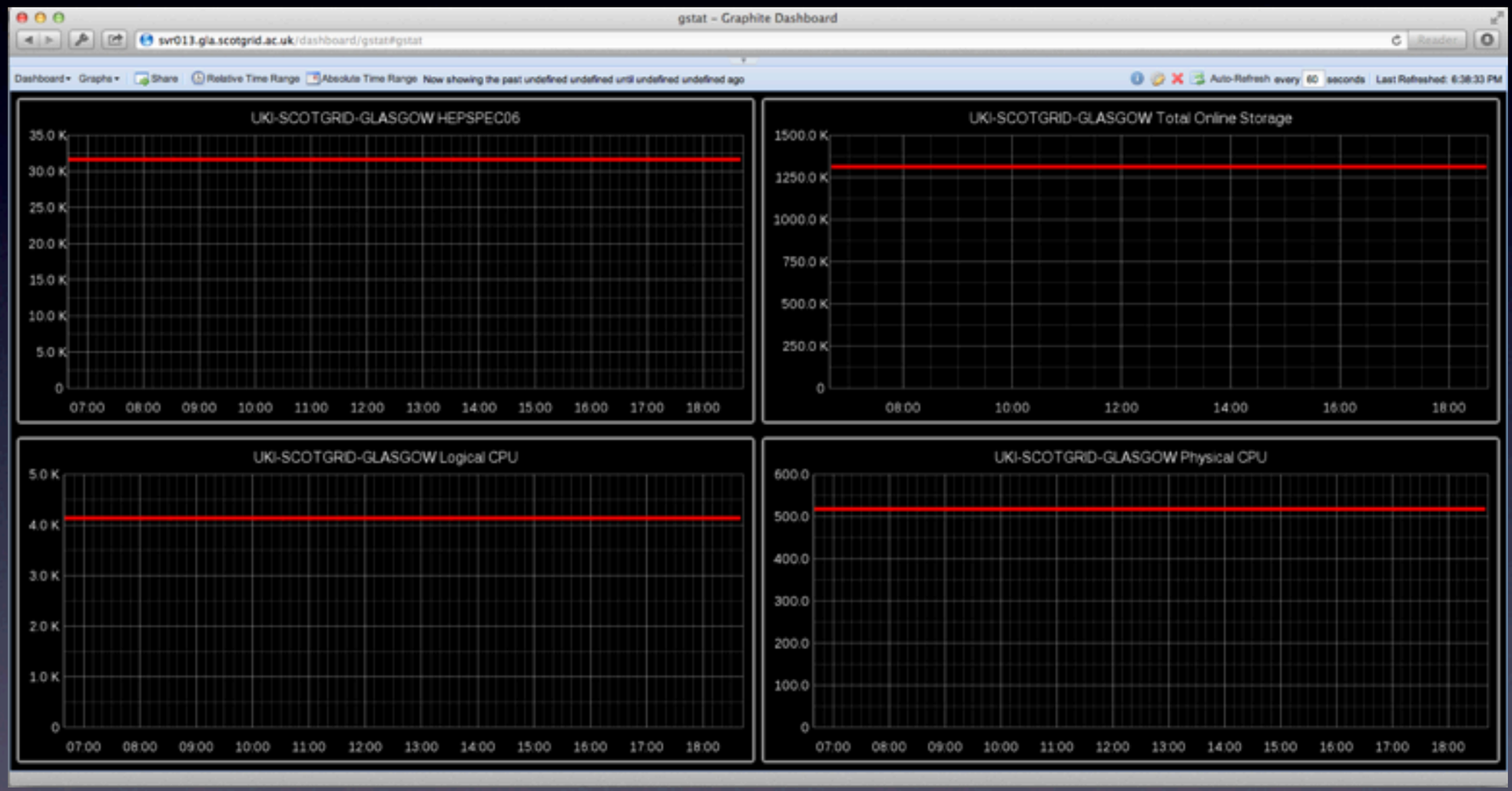
GStat

- Also used the GStat JSON target

URL
`http://gstat2.grid.sinica.edu.tw/gstat/summary/json/`

- No real additional parsing required

GStat (JSON)


gstat – Graphite Dashboard
svr013.gla.scotgrid.ac.uk/dashboard/gstat#gstat
Dashboard
Graphs
Share
Relative Time Range
Absolute Time Range
Now showing the past undefined undefined until undefined undefined ago
Auto-Refresh every 60 seconds
Last Refreshed: 6:38:33 PM
UKI-SCOTGRID-GLASGOW HEPSPEC06
35.0 K
30.0 K
25.0 K
20.0 K
15.0 K
10.0 K
5.0 K
0
07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00
UKI-SCOTGRID-GLASGOW Total Online Storage
1500.0 K
1250.0 K
1000.0 K
750.0 K
500.0 K
250.0 K
0
08:00 10:00 12:00 14:00 16:00 18:00
UKI-SCOTGRID-GLASGOW Logical CPU
5.0 K
4.0 K
3.0 K
2.0 K
1.0 K
0
07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00
UKI-SCOTGRID-GLASGOW Physical CPU
600.0
500.0
400.0
300.0
200.0
100.0
0
07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00

Building graphs

- Currently built visually using dashboard GUI
- Can do so in a scripted fashion - next thing to investigate, possibly with additional dashboard interfaces
- Nice graph interface, generates nice visualisations

Next steps

- Security (x509 https)
- New dashboards
  - Graphene
- Scripted graph dashboards
- Syslog monitoring/system monitoring
  - Logster, quickstatd, collectd
- Nagios integration (use Graphite thresholds to trigger alerts)
- Transforming metrics - data outputs

Positives & possible wider uses

- High level monitoring/visualisation
- Useful for prototyping metrics
- Might be useful for local monitoring hubs - perhaps for metrics tracked for separate sites in a Tier 2?
- Feels similar to Perfsonar - might be interesting to gauge interest in other metrics?
- Not reinventing the wheel!

Summary

- Investigated Graphite as useful replacement/ companion to Ganglia for local monitoring
- Looked at pulling in external data sources
- Explored where this might be useful on a larger scale
- Questions, thoughts, ideas gladly received!