



University  
of Glasgow

Crooks, David, Mitchell, Mark, Purdie, Stuart, Roy, Gareth, Skipsey, Samuel, and Britton, David (2014) *Monitoring in a grid cluster*. In: International Conference on Computing in High Energy and Nuclear Physics (CHEP 2013), 14-18 Oct 2013, Amsterdam, The Netherlands.

Copyright © 2013 The Authors

<http://eprints.gla.ac.uk/95113/>

Deposited on: 17 July 2014

Enlighten – Research publications by members of the University of Glasgow  
<http://eprints.gla.ac.uk>

# Monitoring in a grid cluster

David Crooks<sup>1</sup>, Mark Mitchell<sup>1</sup>, Stuart Purdie<sup>2</sup>, Gareth Roy<sup>1</sup>,  
Samuel Cadellin Skipsey<sup>1</sup>, David Britton<sup>1</sup>

<sup>1</sup>School of Physics and Astronomy, University of Glasgow, G12 8QQ

<sup>2</sup>University of St Andrews

E-mail: david.crooks@glasgow.ac.uk

**Abstract.** The monitoring of a grid cluster (or of any piece of reasonably scaled IT infrastructure) is a key element in the robust and consistent running of that site. There are several factors which are important to the selection of a useful monitoring framework, which include ease of use, reliability, data input and output. It is critical that data can be drawn from different instrumentation packages and collected in the framework to allow for a uniform view of the running of a site. It is also very useful to allow different views and transformations of this data to allow its manipulation for different purposes, perhaps unknown at the initial time of installation. In this context, we present the findings of an investigation of the Graphite monitoring framework and its use at the ScotGrid Glasgow site. In particular, we examine the messaging system used by the framework and means to extract data from different tools, including the existing framework Ganglia which is in use at many sites, in addition to adapting and parsing data streams from external monitoring frameworks and websites.

## 1. Introduction

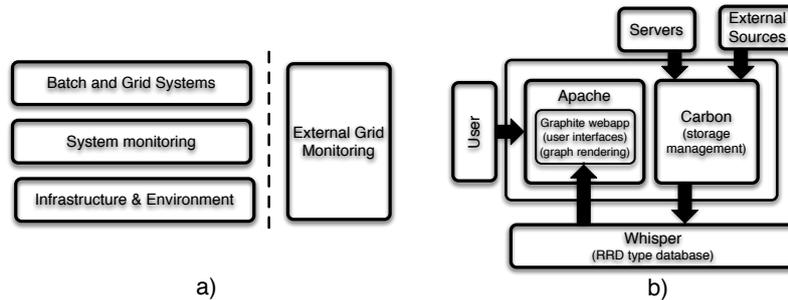
We consider a project to refresh the local monitoring for the ScotGrid Glasgow site, building upon the existing solution using Ganglia [1] and Nagios [2]. One area which is particularly important in the monitoring of any site, but in particular in our case as a grid site, is the combination of metrics from a very wide range of sources. It is this collection, parsing and analysis that forms the basis of the work presented here. In addition to particular packages that were used, we present our methodology and workflow as being applicable to other monitoring packages in the future. Although both Ganglia and Nagios have a wide range of functionality, we primarily use Ganglia for passive testing (i.e. recording of metrics) and Nagios for active testing (i.e. probing the status of a test). In this paper we will consider the passive monitoring aspect of Ganglia. Figure 1a shows a basic overview of the structure of the monitoring of a grid site, comprising infrastructure, local systems, batch and grid systems, and external monitoring.

Many of these systems, in particular the external grid monitoring, can report in different ways and different formats. This has driven the motivation for this work.

## 2. Motivation

The heterogeneous nature of the software in a grid cluster environment suggested that it was important to have a flexible monitoring solution that could accept data from various sources in a lightweight fashion. It also indicated that it was essential to have a single point of collection of the data which could then be reused, with the effective aim of decoupling the data from its visualisation.





**Figure 1.** a) Key components of a grid site monitoring system. b) Graphite operating structure

We looked at the Graphite package [3], currently available in the EPEL repo [4], for this solution. Established over the past few years, it excels at the lightweight messaging we required (the structure of the Graphite platform is shown in figure 1b). The basic format of the Graphite messaging system is given below with an example message:

```
<metric> <value> <timestamp>  
cluster.node.temperature 25 1369827513
```

This message is then sent to the Carbon server at the listening port; for example

```
echo "cluster.node.temperature 25 1369827513" | nc <carbon-server> <port>
```

A metric does not need to be defined before use; the structure of the data within Graphite is determined by the choice of namespace used in the naming of the metric.

### 3. Data sources

An important feature of this work was to deliberately work with a wide range of data sources to establish what could be drawn into the monitoring. In this section we note briefly the main data sources currently in use and any special considerations in each case.

#### 3.1. Internal monitoring

The internal system monitoring at Glasgow had previously been carried out using Ganglia; as work with Graphite continued, we wanted to have a reliable source of local system monitoring data. After considering a number of options, we felt that staying with Ganglia meant that we had a reliable and well understood source of systems data; with the latest version of Ganglia, metrics can be sent directly to Graphite with a configuration setting.

In addition to Ganglia, we use a suite of hand-written script/cron jobs to harvest metrics from other areas such as environment monitoring tools. These were often pre-existing scripts which have been converted in a straightforward way to work with Graphite.

#### 3.2. External data sources

A key element in this work was the use of external monitoring, such as experiment monitoring and accounting. We used a combination of JSON sources for this information, as well as parsed CSV files in one case. The workflow for this process is :

External Sources → JSON parser (optional) → JSON-to-Graphite (converter) → Carbon

We use a package called `httpJsonStats` [5] to watch external JSON sites (refactored to run as a cron job), and an internal web server that serves parsed JSON output where necessary (this was done both for operational and testing purposes). In particular, if an external source did not have a JSON output, we used a second step to parse this into JSON in order to have a common input format. The areas we explored were:

### 3.2.1. ATLAS

We used information from both of these sources of ATLAS job monitoring data:

<http://dashb-atlas-job.cern.ch/dashboard/templates/web-job2/>  
<http://pandamon.cern.ch>

using in each case the JSON output, which we then further parsed to include data relevant to the Glasgow site. The data from the new PanDA Monitor interface was particularly useful in monitoring job activations and starts, as discussed in Section 4.

### 3.2.2. EGI Accounting

We wanted to monitor the EGI Accounting portal for the Glasgow site; in this case we used the Extended CSV file output from

<http://accounting.egi.eu/egi.php?ExecutingSite=UKI-SCOTGRID-GLASGOW>

and then parsed the CSV into JSON.

### 3.2.3. GStat

<http://gstat2.grid.sinica.edu.tw/gstat/summary/json/>

GStat provides a JSON summary of data for all sites - in this case it was relatively simple to pull out the specific Glasgow site data.

## 4. Example analysis

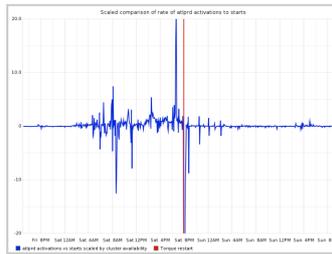
Figure 2 shows a case study of the use of Graphite to flexibly use data from different sources. The aim of this analysis was to study a possible operational situation where the Torque [6] batch server stops starting jobs and must be restarted. The graph shows, in blue, data on the difference between the rate of job activation and the rate of job starts, taken from the previously mentioned ATLAS PanDA monitor JSON interface.

The peaks in the plot flag delays in the job starts that could flag a failure of the batch system and the need for a restart. A delay in the job starts could also indicate the normal scheduling process of the cluster when it is full. As a result the data is also scaled by the available capacity of the cluster, so any peaks taking place when the cluster is full should be suppressed.

Overlaid in red is data taken from the batch server logs which shows a restart occurring. So, we have taken data from the following sources:

- PanDA monitor,
- local batch system,
- local systems logs showing process restarts,

and combined them to generate a diagnostic tool. This demonstrates the power of this approach and speaks to the flexibility of Graphite.

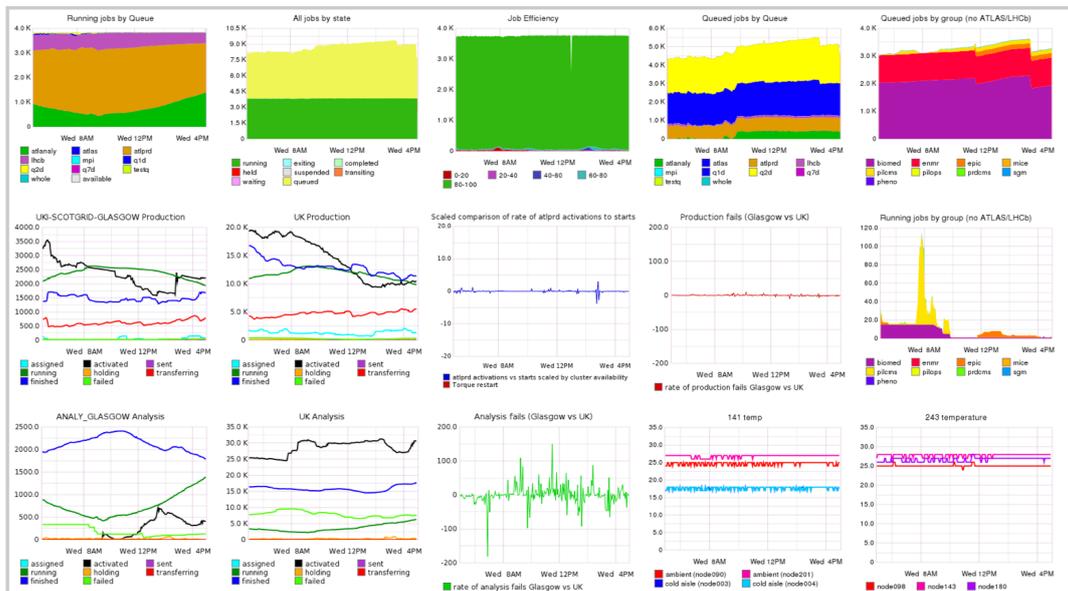


**Figure 2.** Rate of activations vs rate of running jobs, scaled to available capacity (UKI-SCOTGRID-GLASGOW/ATLAS Production)

### 5. Conclusions and Future Work

The prototype top-level monitoring page for the ScotGrid Glasgow dashboard, generated using Graphite and techniques discussed in this paper, is shown in Figure 3. It gathers into one place experiment, cluster, and environmental data. It uses raw and analysed data to present an overview of the cluster containing all the operational-critical information.

Future work consists of further exploration of the combination of metrics and the use of external sources. We will also investigate more visualisation options and the integration of Nagios data and probes to this workflow.



**Figure 3.** Prototype operational dashboard at the ScotGrid Glasgow site.

### References

- [1] *Ganglia Monitoring System* [online] <http://ganglia.sourceforge.net> [accessed 17/01/2014]
- [2] *Nagios* [online] <http://www.nagios.org> [accessed 17/01/2014]
- [3] *Graphite Documentation* [online] <http://graphite.readthedocs.org/en/latest/> [accessed 17/01/2014]
- [4] *EPEL* [online] <http://fedoraproject.org/wiki/EPEL> [accessed 17/01/2014]
- [5] *httpJsonStats* [online] <https://github.com/sverma/httpJsonStats> [accessed 17/01/2014]
- [6] *TORQUE Resource Manager* [online] <http://www.adaptivecomputing.com/products/open-source/torque/> [accessed 17/01/2014]