Monitoring the delivery of virtualized resources to the LHC experiments

C Cordeiro[1], A Di Girolamo[1], D Giordano[1], L Field[1], D Spiga[1] and L Villazon[1,2]

[1]CERN, European Organization for Nuclear Research, Switzerland
[2]Universidad de Oviedo, Spain

E-mail: cristovao.cordeiro@cern.ch

Abstract. The adoption of cloud technologies by the LHC experiments places the fabric management burden of monitoring virtualized resources upon the VO. In addition to monitoring the status of the virtual machines and triaging the results, it must be understood if the resources actually provided match with any agreements relating to the supply. Monitoring the instantiated virtual machines is therefore a fundamental activity and hence this paper describes how the Ganglia monitoring system can be used for the cloud computing resources of the LHC experiments. Expanding upon this, it is then shown how the integral of the time-series monitoring data obtained can be re-purposed to provide a consumer-side accounting record, which can then be compared with the concrete agreements that exist between the supplier of the resources and the consumer. From this alone, it is not clear though how the performance of the resources differ both within and between providers. Hence, the case is made for a benchmarking metric to normalize the data along with some results from a preliminary investigation on obtaining such a metric.

1. Introduction

The adoption of cloud technology, specifically Infrastructure as a Service (IaaS), and the ability to dynamically create machines on demand, is being investigated by WLCG [1] as an alternative approach for delivering the required computing capacity. The approach used is very similar to the existing pilot job paradigm [2]. Just as how the pilot job arrived on the execution node of batch system is not important, how a resource is instantiated is also not important. Once instantiated, the resource carries out the same function as the pilot job and hence the pilot job can be reused in this context.

To use IaaS resources, a virtual machine (VM) is instantiated using the IaaS interface. While being instantiated, the VM is contextualized to the required environment and on starting, the pilot job that is issued to the machine is run and contacts a central task queue to retrieve the payload for a real job.

However, it is not only sufficient to instantiate VMs across many IaaS providers but in addition the full VM lifecycle needs to be managed. An important aspect for this is to monitor the instantiated VM to detect issues and act upon then. Due to the number of VMs that will be monitored, scalability is a primary concern. In addition, the monitoring data can be re-purposed to perform additional functions such as accounting, data analytics, event processing, and dynamic provisioning.

For example, when dealing with commercial IaaS providers, an invoice will be sent and as such it is important to know which VMs were run, where, and what resources were consumed. In this context, monitoring data can be reprocessed to provide accounting records for cross-checking the itemized bill. However in order to compare resource providers, a normalization factor is required to take into consideration the potential differences with the computing power delivered.

This paper describes how to monitor virtualized cloud resources from IaaS providers and how the data gathered can be re-purposed for accounting and other applications. It is outlined as follows: Section 2 describes how an established open-source tool is used to monitoring the resources; the re-purposing of the monitoring data to provide consumer-side accounting and data analytics is detailed in Section 3 along with preliminary investigations into the normalization by means of benchmarking IaaS resources. Finally some conclusions are provided in Section 4.

2. Monitoring IaaS resources with Ganglia

Ganglia [3] is a scalable monitoring system for distributed computing systems such as clusters and Grids. It is an established open-source tool that has already proven itself within the environment of the WLCG [4]. It collects monitoring data from machines (both real and virtual), performs a time-based aggregation, stores the result in a Round-Robin Database (RRD) [5] and provides a Web-based presentation layer.

VMs are monitored in the same way as real machines, every VM in a cloud infrastructure is provisioned with Ganglia's monitoring daemon (*gmond*), which uses unicast rather than the multicast default to send its monitoring data to a collector (*gmond*) hosted on the same machine as Ganglia's meta daemon (*gmetad*) and Ganglia Web. This hierarchical deployment of Ganglia is shown in figure 1.

The *gmond* configuration used has the advantage of eliminating internal network jitter that can occur when using multicast and in addition removes the need to provide a dedicated *gmond* collector(s) within the cloud infrastructure. However an intermediate *gmond* collector may be deployed in some cases which may even be within the cloud infrastructure in cases where the VM nodes are behind a NAT router.

The *gmetad* component has several data sources, polling each collecting *gmond* periodically and storing the monitoring data in a local RRD. With an increasing number of machines, the *gmetad* has to write more often into the RRD causing more time to be spent in I/O operations, which also potentially decreases the lifetime of the physical hard disk drive. For this reason, RRD files are placed into a *tmpfs* file system which grows up to half the server's memory. To avoid data loss, scheduled tasks are used to backup the volatile RAMDisk-based RRD to the local hard disk drive, keeping up to 5 days of history before rotating. Other scheduled tasks are used to observe the *tmpfs* filesystem usage, the *gmetad* service status and to clean physical RRD files referring to

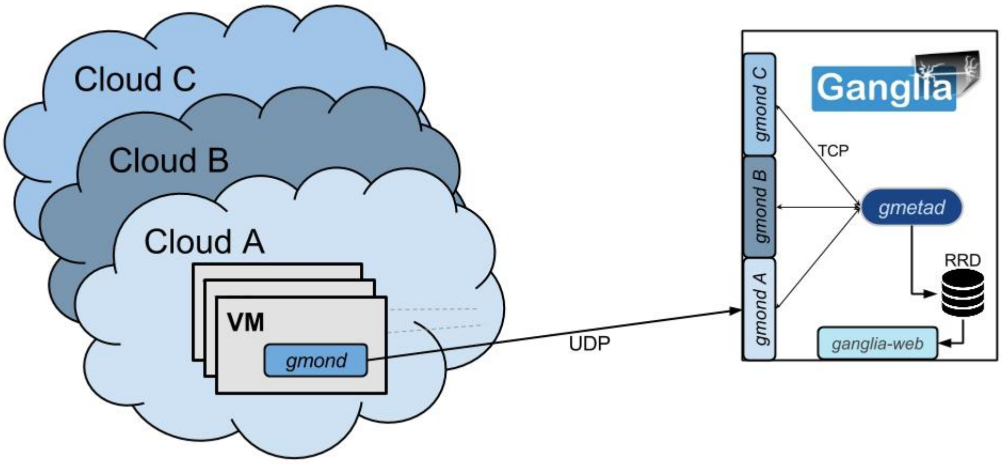


Figure 1: Ganglia deployment architecture.

hosts that have been in the down state for longer than a set period.

Ganglia Web frontend accesses the RRD in order to generate the plots shown in Ganglia's Web interface as seen in figure 2. This interface has been slightly modified to host a JSON file that contains configuration information for each cloud infrastructure, allowing the *gmond* to be automatically configured during the VM's contextualization.

The Ganglia deployment presented is currently being used to monitor the ATLAS [6], CMS [7] and LHCb [8] cloud infrastructures, including commercial providers. A total of 6 different Ganglia instances are used to monitoring more than 20 clusters as there are different cloud activities within each experiment. The Ganglia instances are located at CERN and deployed using Puppet[1], which allows dynamic configuration and simplifies the longterm maintenance.

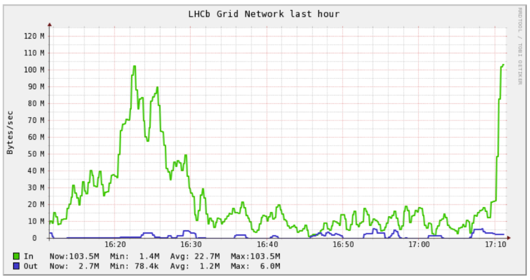
The monitoring data provides a detailed overview of the resource usage within each computing infrastructure to a high level of granularity and also some useful summaries.

3. Re-purposing monitoring data

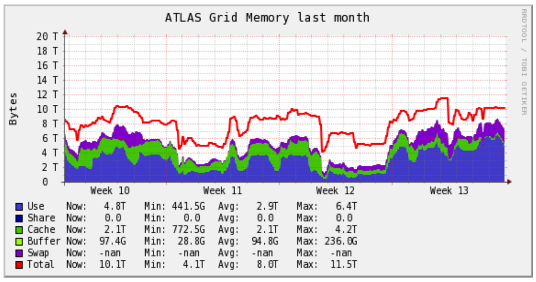
In distributed computing environments where workflows can be complex, additional information on utilization can be valuable. With the current monitoring use case, the data provided is purely informative as no action is automatically triggered. The monitoring data can be re-purposed for other applications, including accounting, benchmarking and data analytics. In this section, different mechanisms and tools that can be used to extract additional information for other applications from the monitoring data are presented.

To extract the data Ganglia's Web API can be used or it can be obtained directly by listening to the specific TCP ports of the *gmonds*. Another option is to periodically extracted the data directly from the RRD used by Ganglia's *gmetad*. A custom application was created and deployed using Puppet on Ganglia's *gmetad* server to

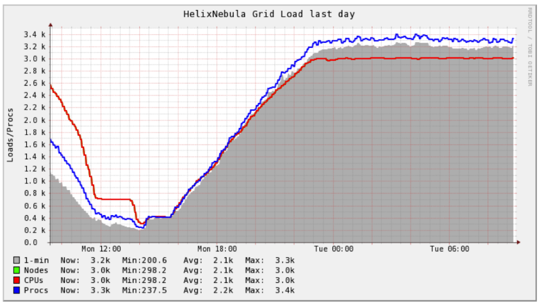
[1] https://puppetlabs.com/



(a) Overall LHCb network monitoring of cloud resources for one hour.



(b) Overall ATLAS memory monitoring of cloud resources for one month.



(c) Overall load monitoring for the Helix Nebula cloud resources during one day.

Figure 2: Example plots taken from different Ganglia Web monitors.

periodically extract data from the RRD and inserted into an InfluxDB [9] instance running on a different machine. Simultaneously, daily CSV files are generated from the data and copied into EOS Disk Storage [10] located at CERN where they can be accessed later. The period chosen was less than 1 hour so that the resolution is not decreased due to Ganglia's aggregation feature. With this approach it is possible to extract and store 1 data point each 15 seconds for each of the 28 metrics on each host, resulting in a total of 4 838 400 data points per host per month.

3.1. Accounting

The accounting of the used resources in a given time interval is an indispensable application when using cloud infrastructures, especially from commercial providers. The main purpose for accounting is to know how many resources were used, by who, where and how efficiently. The realization of consumer-side accounting allows validation of the provider's accounting, typically an invoice when dealing with commercial providers.

By default the Ganglia monitoring system reports on monitoring metrics that are of interest for accounting including: number of hosts (VMs), number of CPU cores, memory usage, network usage, etc. As the monitoring data is built upon periodic reports from the VMs, the integral of that data will yield the accounting record for that period. The accuracy of such an approach will depend upon the period used, which in this case is 15 seconds.

The version of the Ganglia Web frontend that was used offers the ability to download the data points used to plot the graphs as a JSON file. A simple average of the values multiplied by the period will yield the amount of used resources for that period. In addition the reported load average value can be used to infer if the resources were successful used. A prototype [2] of this approach has been developed and is being used to not only cross-check the provider accounting records but more importantly, to validate the real invoices that have been received from commercial providers.

3.2. Resource profiling

With the current adoption of cloud infrastructures by the WLCG, resource profiling is important for comparison with the current Grid resources. The aim is to identify an experiment specific metric that can be adopted as computing unit for high energy physics workflows. Such metric is anticipated to play a crucial role, even in a scenario where a standard unit of computing is defined by commercial providers such as the Amazon [3] ECU. Using a monitoring-based approach to compare the cloud technology with the current Grid resources can potentially provide additional insight.

In a cloud environment the performance of virtualized resources is unknown and hence, the profiling of these resources with respect to the experiment specific workloads becomes essential. A method is required that allows for a quick and clean validation of effective computing resources and the identification of under performing resources. A practical example is shown in figure 3 where two different cloud infrastructures, providing the same amount of resources, perform differently when running the same jobs in different types of workloads. The benchmarking described aims to measure CPU performance in terms of CPU time spent per event to process several workloads. It can be seen that even if the CPU time per event depends on the specific workload, the relative performance between resources from different providers is independent from the workload. This performance ratio is first verified by running several Kit Validation (KV) [11] benchmarking datacards and then confirmed by using HammerCloud [12] for job submission, with a well defined benchmark workload.

In this case the resource profiling provided a clear picture of which commercial cloud provider would perform better under the same conditions.

3.3. Data analytics

By having high-resolution monitoring data, data analytics can perform exhaustive studies that investigate resource behaviours and the reasons behind certain events. A good example of such application was an evaluation within the context of the Helix Nebula initiative [13], where the monitoring data stored in EOS enabled an *a-posteriori* analysis of the deployed resources during one month of production activity. Approximately 30 GB of compressed CSV data for more than one month of execution on a medium/large scale (2000 - 3000 VMs) was stored.

Figure 4 shows the number of running VMs during one month of activity as reported by the Ganglia RRD monthly aggregated view (figure 4a) and by using the full

[2] http://cloud-acc-dev.cern.ch/accounting/
[3] http://aws.amazon.com/ec2/

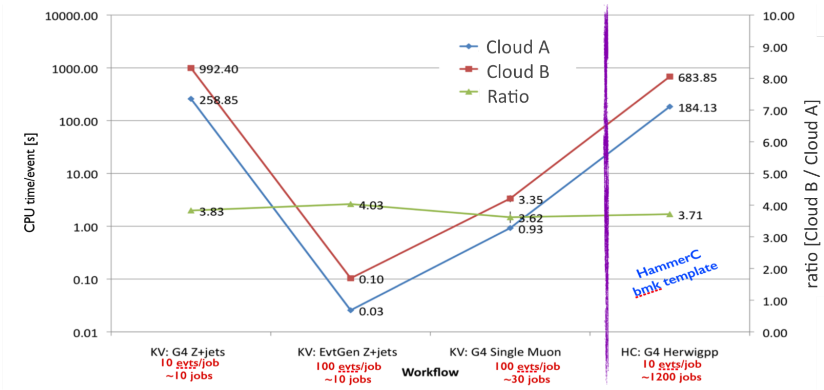
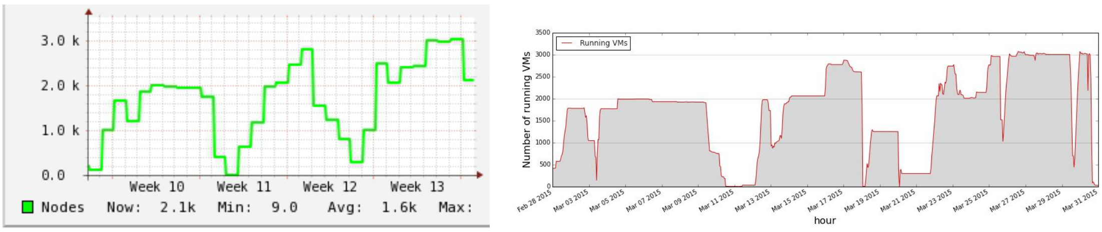


Figure 3: Example of resource profiling for two different cloud providers running the same job workloads.



(a) Number of VMs running over time, based on Ganglia's monitoring data after aggregation.

(b) Number of VMs running over time, built upon the high-resolution data points extracted from Ganglia.

Figure 4: Comparison between the usage plot given by Ganglia (a) and the high-resolution plot (b) generated from the 15 seconds resolution data points during a period of 1 month.

details of Ganglia 15 seconds resolution data (figure 4b). In figure 4a, Ganglia has already performed data aggregation, which leads to a loss of detailed information with medium/large time frames. It is clear in figure 4b that data aggregation has caused the loss of huge peaks, such as the VMs' downscale on the 26th and 30th of March 2015. The plot in figure 4b had great value at the end of the Helix Nebula production activity as it was instrumental for validating (and even preparing the provider's invoice), detecting infrastructure anomalies, and recollecting intentional and unintentional operational behaviours that occurred during the activity.

4. Conclusion

It has been shown how Ganglia can be used as a monitoring tool for cloud infrastructures. A custom application was developed to increase the value extracted from the monitoring data by re-purposing it for other applications. The data has been used as a basis for accounting applications which have played an important roles with invoice arbitration and the profiling of resources. The extraction of high-resolution monitoring data was for the first time used during an evaluation within the context of the Helix Nebula initiative and demonstrated to be a valuable source of information for the post-mortem analysis, including the generation of non-aggregated plots and the detection of short term phenomena. Usage of the monitoring data for other purposes, such as Complex Event Processing for automatically detecting undesired behaviours, can still be explored and will be studied in the near future.

References

[1] Ian Bird. Computing for the large hadron collider. *Annual Review of Nuclear and Particle Science*, 61(1):99–118, 2011.

[2] A. Tsaregorodtsev, V. Garonne, and I. Stokes-Rees. DIRAC: A scalable lightweight architecture for high throughput computing. In *Fifth IEEE/ACM International Workshop on Grid Computing*. IEEE, 2004.

[3] Matthew L Massie, Brent N Chun, and David E Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817–840, jul 2004.

[4] Ian Neilson. Site monitoring for grid services. Presented in the EGEE'07 Conference, Budapest, Hungary, 2007.

[5] Rrdtool: Round robin database tool. `http://oss.oetiker.ch/rrdtool/`, [Online; accessed 21-Oct-2015].

[6] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

[7] Giovanni Petrucciani. The CMS experiment at the CERN LHC. In *The Search for the Higgs Boson at CMS*, pages 15–58. Scuola Normale Superiore, 2013.

[8] Jr. Alves, A. Augusto et al. The LHCb Detector at the LHC. *JINST*, 3:S08005, 2008.

[9] Influxdb - an open-source, distributed, time series database with no external dependencies. `http://influxdb.com/`, [Online; accessed 15-May-2015].

[10] Andreas J Peters and Lukasz Janyst. Exabyte scale storage at CERN. *J. Phys.: Conf. Ser.*, 331(5):052015, dec 2011.

[11] Alessandro De Salvo and Franco Brasolin. Benchmarking the ATLAS software through the kit validation engine. *J. Phys.: Conf. Ser.*, 219(4):042037, apr 2010.

[12] Daniel C van der Ster, Johannes Elmsheuser, Mario Úbeda García, and Massimo Paladin. HammerCloud: A stress testing system for distributed analysis. *J. Phys.: Conf. Ser.*, 331(7):072036, dec 2011.

[13] D. Giordano et al. Accessing commercial cloud resources within the european helix nebula cloud marketplace. *to appear in J. Phys.: Conf. Ser.*, 2015.