

The “Cloud Area Padovana”: lessons learned after two years of a production OpenStack-based IaaS for the local INFN User Community

Paolo Andreetto

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Paolo.Andreetto@pd.infn.it

Fulvia Costa

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Fulvia.Costa@pd.infn.it

Alvise Dorigo

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Alvise.Dorigo@pd.infn.it

Federica Fanzago

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Federica.Fanzago@pd.infn.it

Matteo Segatta

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Matteo.Segatta@pd.infn.it

Sergio Traldi

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Sergio.Traldi@pd.infn.it

Marco Verlato¹

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Marco.Verlato@pd.infn.it

Fabrizio Chiarello

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Fabrizio.Chiarello@pd.infn.it

Alberto Crescente

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Alberto.Crescente@pd.infn.it

Sergio Fantinel

INFN, Lab. Naz. di Legnaro

Via Romea 4, 35020 Legnaro, Italy

E-mail: Sergio.Fantinel@lnl.infn.it

Ervin Konomi

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Ervin.Konomi@pd.infn.it

Massimo Sgaravatto

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Massimo.Sgaravatto@pd.infn.it

Nicola Tritto

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Nicola.Tritto@pd.infn.it

Lisa Zangrando

INFN, Sezione di Padova

Via Marzolo 8, 35131 Padova, Italy

E-mail: Lisa.Zangrando@pd.infn.it

¹Speaker

The Cloud Area Padovana is an OpenStack-based scientific cloud, spread across two different sites - the INFN Padova Unit and the INFN Legnaro National Labs - located 10 km away but connected with a dedicated 10 Gbps optical link. In the last two years its hardware resources have been scaled horizontally by adding new ones: currently it provides about 1100 logical cores and 50 TB of storage. Special in-house developments were also integrated in the OpenStack dashboard, such as a tool for user and project registrations with direct support for Single Sign-On via the INFN-AAI Identity Provider as a new option for the user authentication. The collaboration with the EU-funded INDIGO-DataCloud project, started one year ago, allowed to experiment the integration of Docker-based containers and the fair-share scheduling: a new resource allocation mechanism analogous to the ones available in the batch system schedulers for maximizing the usage of shared resources among concurrent users and projects. Both solutions are expected to be available in production soon. The entire computing facility now satisfies the computational and storage demands of more than 100 users afferent to about 30 research projects.

In this paper we'll present the architecture of the Cloud infrastructure, the tools and procedures used to operate it ensuring reliability and fault-tolerance. We'll especially focus on the lessons learned in these two years, describing the challenges identified and the subsequent corrective actions applied. From the perspective of scientific applications, we'll show some concrete use cases on how this Cloud infrastructure is being used. In particular we'll focus on two big physics experiments which are intensively exploiting this computing facility: CMS and SPES. CMS deployed on the cloud a complex computational infrastructure, composed of several user interfaces for job submission in the Grid environment/local batch queues or for interactive processes; this is fully integrated with the local Tier-2 facility. To avoid a static allocation of the resources, an elastic cluster, initially based only on cernVM, has been configured: it allows to automatically create and delete virtual machines according to the user needs. SPES is using a client-server system called TraceWin to exploit INFN's virtual resources performing a very large number of simulations on about a thousand nodes elastically managed.

*International Symposium on Grids and Clouds 2017 -ISGC 2017-
5-10 March 2017
Academia Sinica, Taipei, Taiwan*

POS (ISGC2017) 021

1. Introduction

The Istituto Nazionale di Fisica Nucleare (National Institute of Nuclear Physics, INFN), is the Italian Institution founded in 1951 to promote, coordinate and finance research in nuclear physics, elementary structures and basic interactions. It conducts theoretical and experimental research in the fields of subnuclear, nuclear and astroparticle physics. All of the INFN's research activities are undertaken within a framework of international competition, in close collaboration with Italian universities. INFN facilities consist of four national laboratories, 20 divisions and 8 groups.

INFN-Padova division and INFN Legnaro National Laboratories (INFN-LNL) are two of these facilities: they are located in the North-East part of Italy, 10 km far away. The longstanding collaboration between these two INFN sites covers different areas, including computing: in particular INFN-Padova and INFN-LNL have been operating together for many years a distributed LHC Tier-2 Grid facility serving the ALICE and CMS experiments [1].

At the end of 2013, these two INFN sites decided to launch a new ICT project for the provisioning of a computational and storage cloud service, aimed to satisfy the computing needs of the relevant users, in particular for those use cases that could not be easily addressed by the existing Grid infrastructure. In order to limit the proliferation of private clusters (which have a very high cost in terms of administration, and whose resources are often poorly used) the goal of the project was to provide a pool of resources that could be easily and efficiently shared among all the relevant stakeholders.

This project led to the implementation of a Cloud facility, called Cloud Area Padovana [2], with resources spread across the two INFN sites, but integrated in a single seamless Cloud infrastructure based on the OpenStack middleware [3].

A first prototype, with restricted functionality and limited resources, was set up in the mid of 2014: this was tested and validated by a few pilot applications. More resources were then added and new services were integrated: the Cloud Area Padovana was declared production ready and made available to all users at the end of 2014.

Since then the Cloud Area Padovana infrastructure has been kept evolving, in terms of resources and functionalities. The usage of such Cloud infrastructure has been kept increasing, in terms of number of users and experiments, and in terms of actual resource usage: this increased workload also raised a few problems, not spotted before, that had to be properly handled.

This paper is organized as follows. Section 2 provides an overview of the Cloud Area Padovana, describing the offered functionalities. Section 3 presents the architecture of such cloud infrastructure, discussing also about some implementation choices, focusing in particular on the ones that had to be revised. In Section 4 some aspects regarding the operations of the Cloud are discussed. Section 5 presents the tool used to manage accounting information, while Section 6 discusses about the usage of the Cloud Area Padovana. Section 7 concludes the article.

2. Overview of the Cloud Area Padovana

Users of the Cloud Area Padovana are organized in different projects (which represents experiments or other research groups): the overall available resources are partitioned among these stakeholders.

As explained below, each project has one (or possibly more) manager (usually the team leader), who is responsible to manage the membership requests for the project he/she manages.

Users are given the possibility to self create virtual machines (using the images that best fit their needs), create block storage volumes, attach such volumes to instances, etc. Thanks to the OpenStack Heat orchestration engine, the deployment of complex applications is also supported.

Besides the support for virtual machines, the possibility to instantiate docker containers is also possible. This was implemented deploying the nova-docker service, which allows the instantiation of docker containers with the very same tools and procedures used to create virtual machines. The assessment, testing and support of such service (which is not part of the official OpenStack distribution) was done in the context of the INDIGO-DataCloud project [4].

Users access the Cloud Area Padovana functionality usually through a web-based dashboard, which was integrated with the INFN Identity and Authorization Infrastructure (INFN-AAI), so that INFN users can authenticate to the Cloud service using the same procedure used to access other INFN services. Access via command line tools is also possible: both the OpenStack and the Amazon EC2 [5] interfaces are supported.

Concerning networking, instances can alternatively be given a floating (public) IP and been accessed from the WAN, or they can be reachable from the Padova and Legnaro LANs using their private IP.

The Cloud Area Padovana also offers some higher level tools than the ones usually provided by a IaaS Cloud. For example, by integrating the *elastiq* service [6], users can instantiate on the Cloud personal or group batch clusters based on HTCondor [7]. The cluster size (in terms of number of worker nodes) is automatically scaled out depending on the job queue filling level: worker nodes are created when jobs are waiting in the queue and are eventually destroyed when they are free.

3. Architecture of the Cloud Area Padovana

Figure 1 below shows a high level view of the Cloud Area Padovana. As mentioned before this is a single OpenStack based cloud whose resources are spread between INFN-Padova and INFN-LNL. More in detail the compute nodes (where Cloud Virtual Machines are instantiated) are on both sites. At the time of writing this paper the Cloud encompasses 15 compute nodes in Padova and 13 compute nodes at INFN-LNL, providing in total about 550 physical cores. Cloud services have instead been deployed in Padova.

A 10 Gbps fiber link connects the two sites.

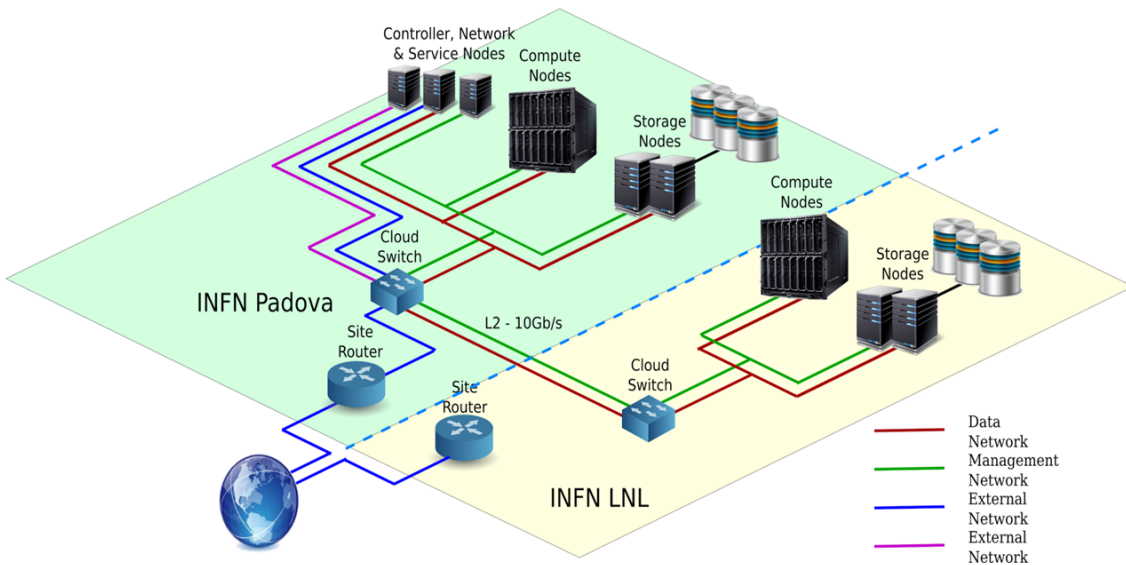


Figure 1: Cloud Area Padovana layout

Figure 2 shows how the OpenStack services were deployed.

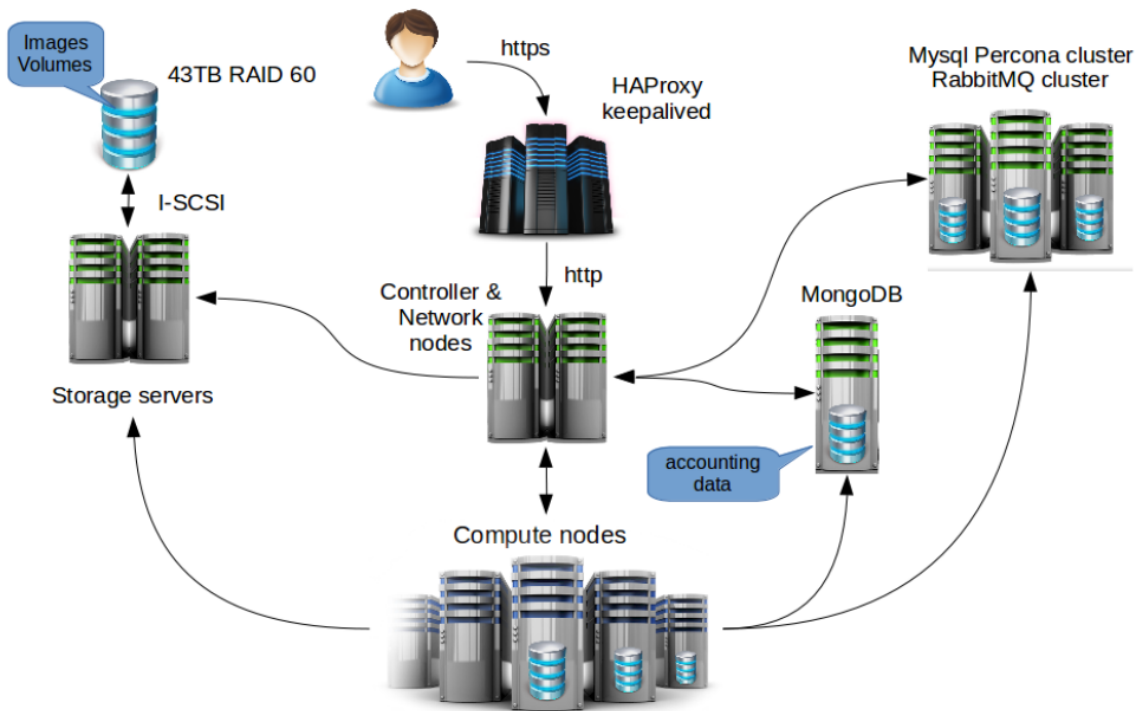


Figure 2: Architecture of the Cloud Area Padovana

The OpenStack services are configured in high availability mode on two nodes, which act as both controller and network nodes. The active/active model (services run at the same time on both nodes) was chosen to implement the high availability, relying on the HAProxy [8] daemon

(for the balancing of the incoming connections among the physical nodes), and Keepalived [9] (for the Virtual IP movement between the HAProxy servers).

HAProxy is also used to expose SSL interfaces for the OpenStack endpoints. Requests to the OpenStack services are encrypted and targeted to HTTPS URIs; HAProxy is then responsible for extracting the plain payload and forwarding it to the OpenStack APIs on the controller nodes which actually accept a plain HTTP connection.

The HAProxy - Keepalived cluster is composed by three instances, implemented as virtual machines of a Proxmox cluster.

In the first prototype the chosen configuration consisted of cloud controllers and network nodes deployed on different hosts, in high availability: two nodes for the controllers, other two hosts for the network nodes. The same hosts used as network nodes were used also as storage servers.

We saw problems with such configuration (e.g. if a network node had to be rebooted, this operation had impact on the Cloud storage system) and therefore we then decided to separate storage nodes and other services. Considering also that, at least according to our experience, deploying the controller and network nodes on different hosts is not strictly needed, we decided to change the deployment model, using two nodes as combined controller-network nodes, without mixing storage and cloud services.

A Percona XtraDB cluster [10], composed of three instances, is used to host the relational databases needed by the OpenStack services. Originally this Percona cluster was hosted on three virtual machines, but, for performance reasons, it was then migrated to three physical machines. The access to this database by the OpenStack services had to be properly tuned, in particular setting different primary instances for the different databases. The three machines used for the Percona instances also host three RabbitMQ [11] instances configured to provide highly available AMQP queues (which are exploited by almost all the OpenStack services).

A single MongoDB instance is instead used for the accounting data, collected by the OpenStack ceilometer service, since high availability was not considered critical for such database.

The cloud storage is provided by an iSCSI storage server installed in Padova. The provided 45 TB of disk space is used to store Cloud images and snapshots (Glance service) and to provide block storage to the instances (Cinder service).

This storage, exposed by two storage servers, has been configured using GlusterFS [12].

In the first setup of the Cloud Area Padovana, this storage was used also to implement a shared file system among all compute nodes, used for the ephemeral storage of the Cloud virtual machines. This was done to enable the live-migration of virtual machines among different compute nodes.

Unfortunately we experienced some scalability and performance problems with such setup: some applications stressed too much the storage system, impacting the whole infrastructure.

For this reason, and also considering that the possibility of live migrating instances was considered a key functionality only for a very small subset of applications, we decided to consider a different implementation for the nova ephemeral storage: now most of the compute nodes use their local storage disks for the nova service. Only a few compute nodes use a shared

file system: these hypervisors, which support the live migration of instances, are exposed as an ad-hoc availability zone and are targeted to host critical services.

Concerning the networking, OpenStack Neutron with Open vSwitch and GRE (Generic Router Encapsulation) [13] is used. Two provider routers have been configured. The first one has the external gateway on the public network: this is used for the instances that can be given a floating IP. The second one has instead the external gateway towards the internal LAN: it allows instances to be reachable from the Padova and Legnaro LANs using their private IP.

Extensions to the OpenStack identity and dashboard services were implemented, to integrate with SAML [14] based identity providers. In particular the integration with the INFN Authentication and Authorization Infrastructure (INFN-AAI) was implemented.

The OpenStack dashboard was also enhanced by the integration of a software module to manage the registration of users and projects. Before being allowed to use the Cloud, a new user must subscribe to one or more existing projects, or alternatively require the creation of a new one. Requests for registration must traverse a registration workflow which involves different subjects, each one with different roles and capabilities. The cloud administrator has all the privileges of users and projects and he is responsible, eventually, for accepting or rejecting the request for registration. The project administrator is instead the user of the project that is responsible for validating any new request for registration with that project. Since a cloud administrator is not supposed to know if a new user is allowed to be a member of a project, this responsibility is delegated to the relevant project administrator. The complete registration workflow is shown in Figure 3 below.

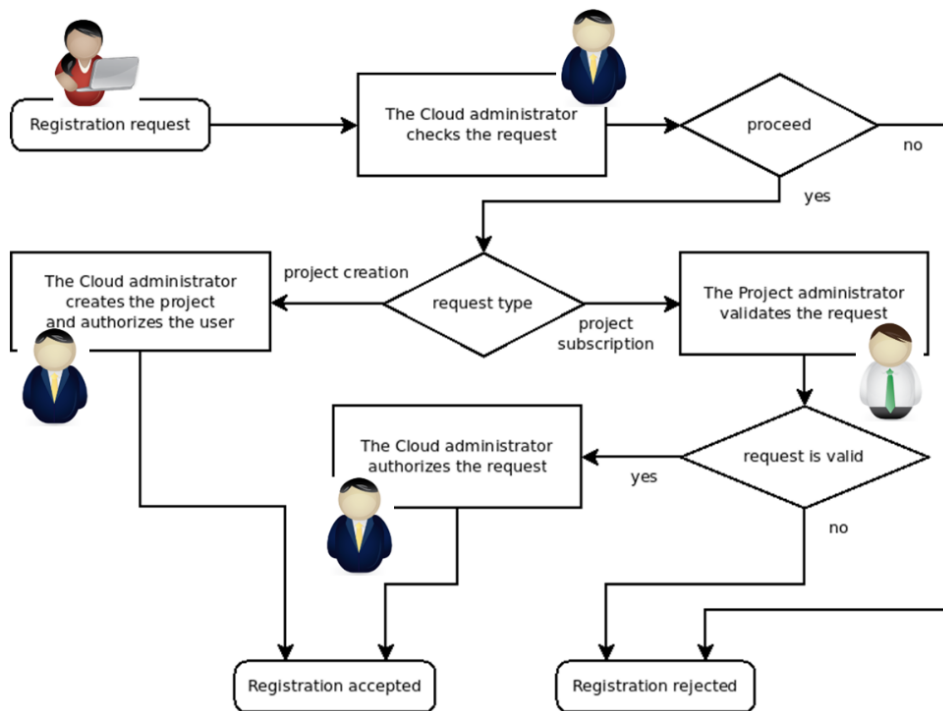


Figure 3: Registration flowchart

4. Operations

We learned, from our previous experiences in managing large computing infrastructure (in particular the WLCG Padova-Legnaro Tier-2) that any manual configuration of the resources and of services should be avoided. Manual interventions can in fact be error prone, leading to problems that can then be hard to debug. Moreover specific customizations can then be forgotten and lost when updates are done. While for the administrator of the WLCG Tier-2 site we are using xCAT [15] as configuration management system, for the Cloud Area Padovana we decided to use some different technologies and tools: we chose Puppet [16] and Foreman [17] as configuration and provisioning tools for the Cloud Area Padovana.

Foreman is responsible for provisioning hosts via PXE and acts also as Puppet Server managing all nodes configurations. Using Foreman we use to build different hosts by installing them with the desired operating systems. All hosts are associated to respective hostgroups according to their roles in the infrastructure. In particular for the different hostgroups we use different set of puppet classes, for the configuration of OpenStack and of other ancillary services.

These tools helped in reducing the cost of running the infrastructure. E.g. a compute node can be (re)installed and (re)configured in a completely automated (and reproducible) way very quickly.

Every change done on the production cloud is first tested and validated on a testbed. This is a small infrastructure which however has the very same architecture of the production facility (i.e. there are two controller-network nodes where services are deployed in high-availability, there is a Percona cluster, etc.) and which is therefore able to resemble it.

Applied updates are then automatically tested using the monitoring infrastructure discussed below, to be sure that these changes don't break the functionality and performance of the system.

This testbed is in particular used to test OpenStack updates. As balance between having latest features and fixes, and the need of limiting the manpower, we use to perform one OpenStack update per year (i.e. skipping one OpenStack release).

We are currently running the Mitaka release of OpenStack, and are starting the migration to Ocata.

Monitoring is a key aspect in the operations of the Cloud Area Padovana, to be able to early detect possible problems. In particular three tools are used:

- Ganglia [18]
- Nagios [19]
- Cacti [20]

Ganglia is used to check the system parameters (e.g. CPU load, memory usage, etc.) of the nodes composing the infrastructure.

Nagios is heavily used to check the functionality and performance of all the deployed services. Some of the used Nagios sensors have been found on the Internet, while others are in-house developments. Examples of the latter sensors are a probe that instantiates a VM on each compute node and tests its network connectivity and a probe that periodically creates a Cinder volume and tries to attach it on an existing instance.

Cacti is instead used in particular for network related information. Particular focus was also given to security auditing, to be able to manage security incidents. This task, which is particularly challenging in a Cloud environments since the relevant instances are dynamically created and destroyed, is even more complex in the Cloud Area Padovana, because of its peculiar network setup.

By using some specific tools (in particular the ulogd [21] software) and by archiving all the relevant log files, we are now able to trace any internet connection initiated by instances on the Cloud. This allows e.g. to find the person responsible of a security incident, even if in the meantime the relevant Cloud instance was deleted.

5. Management of accounting information

Ceilometer, the OpenStack component responsible to collect and manage accounting information, was deployed on the Cloud Area Padovana. We found however some problems with its usage, in particular:

- each metric managed by Ceilometer refers to a single resource. Aggregating data (e.g. per project) must be done on the fly and therefore can be extremely slow;
- it is not straightforward to get information that involve multiple metrics (this is the case e.g. of the efficiency in CPU usage);
- there are some known bugs that require a post-processing to possibly correct the collected data;
- each metric data point is accompanied by redundant metadata, posing limits to the period of data retention.

Because of these problems we implemented a tool, called CAOS [22], which manages and then presents the accounting data provided by OpenStack (in particular those collected by the Ceilometer service). CAOS enables tracking of usages at different levels: e.g. per project or per compute node, in such a way that both current and past consumption of resources can be easily determined.

The architecture of the CAOS service is represented in Figure 4.

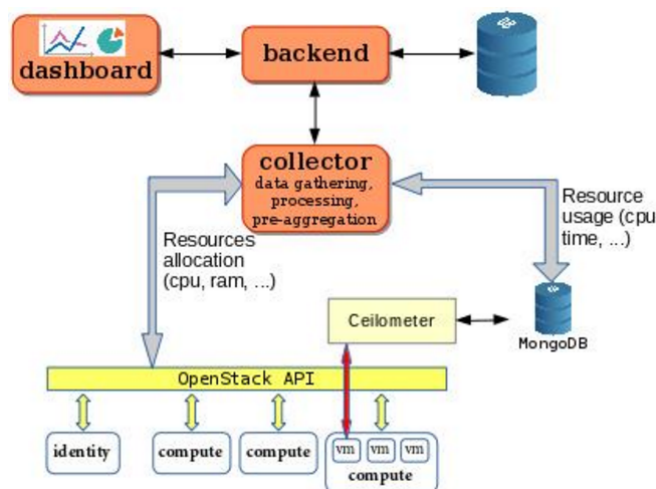


Figure 4: Architecture of the CAOS service

The collector gathers data at regular intervals from both the OpenStack API and the Ceilometer service. The former is used for tracking Identity information (e.g. projects), resources allocation and compute nodes state; the latter is used mainly for tracking real consumption, such as the used CPU time of instances. We decided to read data directly from the MongoDB instance to overcome some Ceilometer limitations and to have more flexibility in the retrieval of the information. The data is then analyzed and pre-aggregated at coarser granularity (e.g. hourly), enabling almost unlimited data retention. The collector also computes more complex metrics which involve operations across different metrics, such as the CPU efficiency.

The backend (Elixir on Erlang VM) provides a time series framework for writing and reading metrics: the data can be used to monitor the actual or very recent state of the cloud, or it can be aggregated (e.g. over all projects) or downsampled at a given resolution or time range to provide resources accounting.

The dashboard (an Angular/Javascript application), allows the Cloud administrator to easily get resource usage information (CPU time, wall clock time, efficiency) for all (or some) projects, for a given time slot period, by querying data from the backend. Such information can also be presented as graphs such as the one shown in Figure 5.

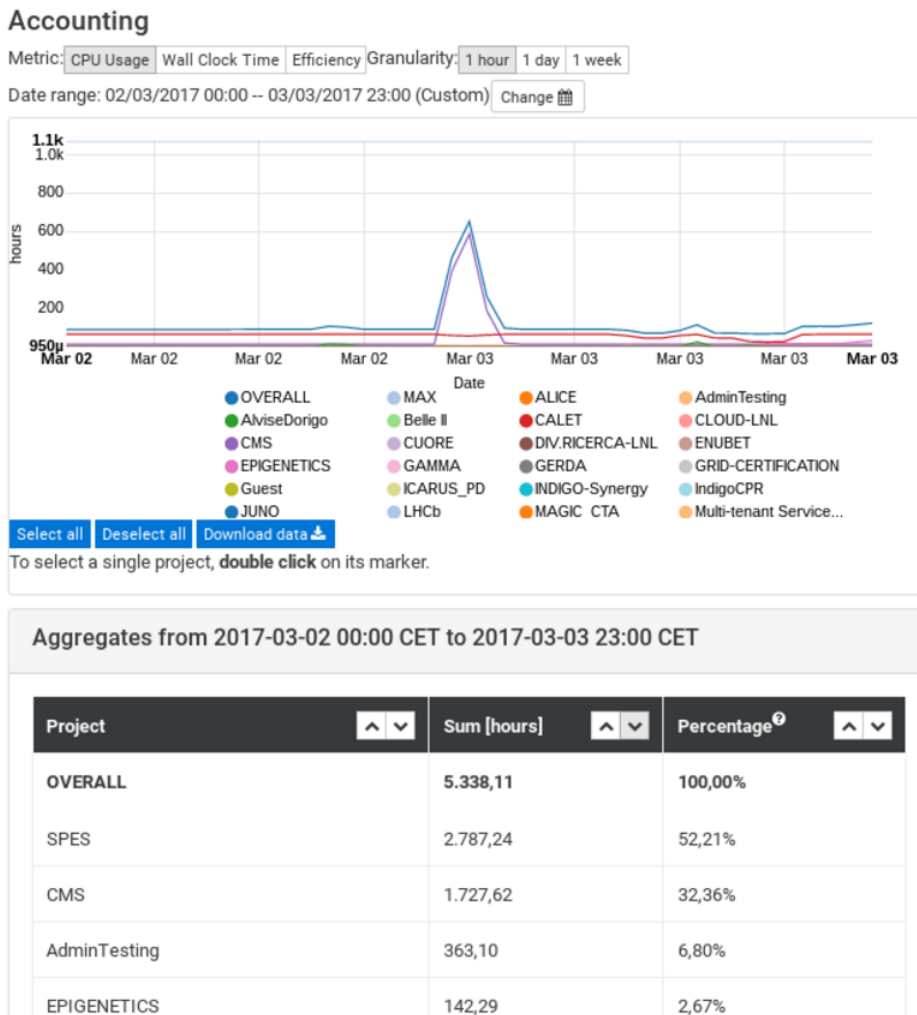


Figure 5: Example of graph generated by CAOS

POS (ISGC2017) 021

6. Usage of the Cloud Area Padovana

About 100 users, belonging to about 30 projects are currently registered in the Cloud Area Padovana. They use the Cloud in different ways: interactive access (e.g. for analysis jobs, code development and testing, etc.), batch computations, etc.

Users of the CMS and SPES experiments are at the moment the most demanding customers of the Cloud Area Padovana.

The main usage of cloud resources done by users of the CMS local group is to run interactive or batch processes when the load of work required by data analysis or production is really high or urgent to be executed without overloading the available physical resources: the virtual resources are considered as an addition of the ordinary computing power shared among users. For this reason the virtual machines have to reproduce the setup and the environment of CMS experiment as prepared in the physical facility of User Interfaces (UI) located in Padova.

Data are read/written from/to the Tier-2 dCache storage system (more than 2 PB of disk space): thanks to a special setup data stored in this dCache system can be directly accessed via the native dcap protocol. Storage is also provided by a Lustre file system (about 80 TB of disk space): this one is used for home directories and to store other general purpose data. Experiment software is instead taken from CVMFS, a http-based file system.

Currently physical UIs are used to run interactive processes, to submit processes to the Tier-2 nodes managed by the LSF batch system or to submit them to the grid distributed environment. The user's advantage to run interactive jobs on virtual UI is the possibility to instantiate private UI with an amount of CPU and RAM depending on user need. Generally the most required flavor includes 32 virtual CPUs and 64 GB of RAM.

The virtual user interfaces are intensively used but for short period, i.e 5x32 cores for 3 days running more than 3000 CPU intensive production jobs reading input data from dCache and writing outputs to Lustre; 5x32 cores for a week running analysis job (fit RooFit), CPU and RAM intensive with typical length of 12-24 hours.

Starting in March 2016 CMS also uses the Cloud resources in batch mode, using the elastiQ system mentioned above, that improves the usage of project resources. Originally the batch cluster nodes were instantiated using as image a snapshot based on cernVM (that provides HTCondor and elastiQ by default) adapted for the CMS environment. Since November 2016 the image for the master and slaves nodes is the same one used for virtual UI, plus HTCondor and elastiQ software.

After some months of sporadic usage for tests and infrastructure evaluation, the elastic cluster is now heavily used from the B physics group for the generation of pseudo experiments (Toy MonteCarlo) needed for the evaluation of statistical error in the measurement of angular parameters related to a rare decay of the B meson. Jobs run around 1-2 hours reading and writing outputs from/to Lustre. In order to avoid problems with the mount of Lustre with a high number of clients, some tuning were required. Currently CMS can run up to 125 HTCondor worker nodes, implemented as VMs using a flavor with 6 VCPUs and 12GB of RAM, that means a total number of concurrent running jobs equal to 750.

All CMS users submit jobs to the same batch cluster: the fairshare among users is guaranteed by the batch system.

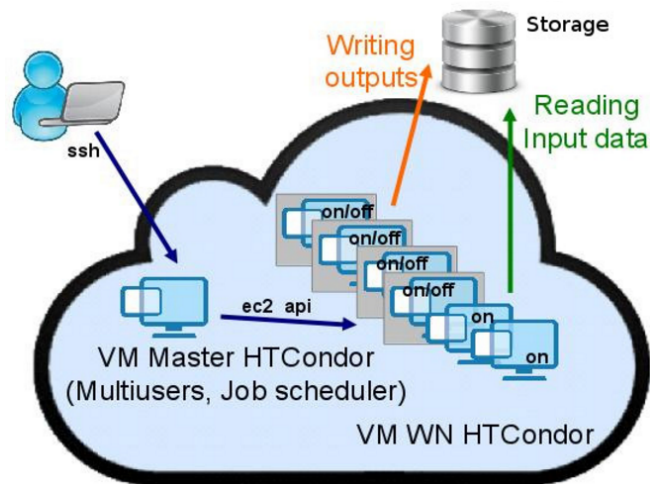


Figure 6: The CMS elastic cluster on the Cloud

The SPES experiment is instead using the Cloud to address some modeling computational requirements.

A framework, called TraceWin, is used to run SPES jobs on the Cloud. This is a client-server system where a TraceWin master service is instantiated on a Virtual Machine and then a number of TraceWin clients are elastically instantiated on the Cloud: each client receives tasks coming from the master, and execute them.

In particular the Cloud Area Padovana supported the activity of Beam Dynamics characterization of the European Spallation Source - Drift Tube Linac (ESS-DTL), that is mandatory to the SPES characterization. For this task, 100k different, 39m long, DTL configurations, based on Monte Carlo simulations (each one with 100k macroparticles), were performed.

The configurations were split into 10k groups, each one resolved by 2k parallel job running on the cloud in batch mode using the TraceWin system. Up to 500 cores on the Cloud were used simultaneously.

7. Conclusions and future work

We discussed in this paper about the Cloud Area Padovana, which has been running as a fully production service since the end of 2014. This is an OpenStack based IaaS, where some in-house developments were also integrated. Users (about 100 scientists, belonging to about 30 research groups) are using such infrastructure for different applications: interactive jobs, batch and parallel applications, development, etc.

Several activities are planned for the next future.

Concerning user authentication and authorization, a simplification of the registration workflow is being implemented. The new system will also be able to support account renewal, that will be managed per project (i.e. the expiration of an account can be different for the different projects the user belongs to). Work is also being done to integrate with external identity providers using OS-Federation.

In the underlying infrastructure, changes are foreseen in the area of Cloud storage: a new Ceph [23] based storage is being installed. It will be initially used for the Cinder block storage service but in the long term plans are to use it for all the OpenStack services.

The deployment of the Synergy [24] service, to allow an efficient fair-share based allocation of resources among multiple user communities, is also envisioned.

Last, but not least, we're planning the integration of the Cloud Area Padovana with the Cloud infrastructure owned by the University of Padova.

References

- [1] S. Badoer et al. *The Legnaro-Padova distributed Tier-2: challenges and results*, *Journal of Physics: Conference Series* **513** (2014) 032090
- [2] C. Aiftimiei et al., *Implementation and use of a highly available and innovative IaaS solution: the Cloud Area Padovana*, *Journal of Physics: Conference Series* **664** (2015) 022016
- [3] *Home page for the OpenStack project*, <http://www.openstack.org>
- [4] D. Salomoni et al., *INDIGO-Datacloud: foundations and architectural description of a Platform as a Service oriented to scientific computing*, arXiv:1603.09536v3 [cs.SE]
- [5] *Amazon EC2*, <https://aws.amazon.com/ec2/>
- [6] <https://github.com/dberzano/elastiq>
- [7] D. Thain, T. Tannenbaum, and M. Livny, *Distributed Computing in Practice: The Condor Experience, Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356, February-April, 2005.
- [8] W. Tarreau, *HAProxy-The Reliable, High-Performance TCP/HTTP Load Balancer*, <http://haproxy.1wt.eu>
- [9] A. Cassen, *Keepalived: Health checking for LVS and high availability*, (2002), <http://www.linuxvirtualserver.org>
- [10] *Percona XtraDB Cluster*, <https://www.percona.com/software/mysql-database/percona-xtradb-cluster>
- [11] <https://www.rabbitmq.com/ha.html>
- [12] *The Gluster web site*, <http://www.gluster.org/>
- [13] S. Hanks et al., *Generic routing encapsulation (GRE)*, (2000). <http://tools.ietf.org/html/rfc2784.html>
- [14] *OASIS Security Services (SAML) TC | OASIS*, <https://www.oasis-open.org/committees/security/>
- [15] *Extreme Cluster/Cloud Administration Toolkit (xCAT) home page*, <https://xcat.org/>
- [16] *Puppet Labs home page*, <https://puppetlabs.com/>
- [17] *Foreman home page*, <http://theforeman.org/>
- [18] M. Massie, B. Chun, D. Culler, *The Ganglia Distributed Monitoring System: Design, Implementation, and Experience*, *Journal of Parallel Computing*, vol. 30, no. 7, July 2004.
- [19] W. Barth, *Nagios. System and Network Monitoring*, No Starch Press, u.s (2006)

- [20] *Cacti home page*, <http://www.cacti.net/>
- [21] *The netfilter.org ulogd project home page*, <https://www.netfilter.org/projects/ulogd/>
- [22] <https://github.com/CloudPadovana/caos>
- [23] S. A. Weil et al., *Ceph: a scalable, high-performance distributed file system*, OSDI '06 Proceedings of the 7th symposium on Operating systems design and implementation, Pages 307-320
- [24] L. Zangrando et al., *Synergy, a new approach for optimizing the resource usage in OpenStack*, in Proceedings of *The International Symposium on Grids and Clouds (ISGC) 2017*, PoS (ISGC2017) 012