IL NUOVO CIMENTO **39 C** (2016) 268
DOI 10.1393/ncc/i2016-16268-y

Colloquia: IFAE 2015

!CHAOS: A cloud of controls

S. Angius(1), C. Bisegni(1), P. Ciuffetti(1), G. Di Pirro(1), L. G. Foggetta(1), F. Galletti(1), R. Gargana(1), E. Gioscio(1), D. Maselli(1), G. Mazzitelli(1), A. Michelotti(1), R. Orrù(1), M. Pistoni(1), F. Spagnoli(1)(*), D. Spigone(1), A. Stecchi(1), T. Tonto(1), M. A. Tota(1), L. Catani(2), C. Di Giulio(2), G. Salina(2), P. Buzzi(3), B. Checcucci(3), P. Lubrano(3), M. Piccini(3), E. Fattibene(4), M. Michelotto(5), S. R. Cavallaro(6), B. F. Diana(6), F. Enrico(6) and S. Pulvirenti(6)

(1) *INFN-LNF, Laboratori Nazionali di Frascati - Frascati, Italy*
(2) *INFN-TV, Sezione di Roma Tor Vergata - Roma, Italy*
(3) *INFN-PG, Sezione di Perugia - Perugia, Italy*
(4) *INFN-CNAF, Centro Nazionale Tecnologie Informatiche - Bologna, Italy*
(5) *INFN-PD, Sezione di Padova - Padova, Italy*
(6) *INFN-LNS, Laboratori Nazionali del Sud - Catania, Italy*

received 7 January 2016

Summary. — The paper is aimed to present the !CHAOS open source project aimed to develop a prototype of a national private Cloud Computing infrastructure, devoted to accelerator control systems and large experiments of High Energy Physics (HEP). The !CHAOS project has been financed by MIUR (Italian Ministry of Research and Education) and aims to develop a new concept of control system and data acquisition framework by providing, with a high level of abstraction, all the services needed for controlling and managing a large scientific, or non-scientific, infrastructure. A beta version of the !CHAOS infrastructure will be released at the end of December 2015 and will run on private Cloud infrastructures based on OpenStack.

1. – !CHAOS: a cloud of controls project

"!CHAOS: a cloud of controls"(1) [1] is an open source project being developed by several INFN [2] partners at Italian level and supported by National Instruments and ESCO [3] companies for the implementation of different case studies. The project is

(*) Corresponding author. E-mail: `francesca.spagnoli@lnf.infn.it`
(1) Project website: `http://chaos.infn.it`

originally born within the context of High Energy Physics (HEP) and evolved from a candidate of Distributed Control Systems (DCS) & Data Acquisition (DAQ) for the SuperB HEP collider to a prototype of a national cloud infrastructure that offers monitoring and control services also to the society and industries.

To achieve these objectives, the project consists of 4 major research & development activities and a coordination, communication and documentation activity:

- Framework development: it consists of the development of C++ routines of the common architecture aimed to ensure the communication of data among the five nodes of the system: data acquisition (CU—Control Unit), presentation (UI—User Interface), proxies/indexing/storage (CDS—CHAOS Data Service), data handling (EU—Execution Unit) and system state information (MDS—Metadata Service).
- Drivers' and CU development and integration: for the development of C++ routines devoted to the drivers' implementation and CU development and their integration and deployment for the use cases; framework tests and debugs [4].
- Use cases implementation: it consists of software and hardware implementation devoted to the specific CU, EU and UI for the three main use cases: Frascati Beam Test Facility (BTF) DAQ, accelerator devices and diagnostic controls; LNS (Laboratori Nazionali del Sud) beam source control [5]; Touschek auditorium environmental control.
- IT infrastructure development and implementation: for the analysis and implementation of the cloud infrastructure and services to offer the !CHAOS framework as a service (CDS, MDS) to our HEP community and external users.

The key features and development strategies of !CHAOS are:

- scalability of performances and size,
- integration of all functionalities,
- abstraction of services, devices and data,
- easy and modular customization,
- extensive data catching for performance boost,
- use of high-performance internet software technologies.

2. – The Cloud infrastructure of !CHAOS

In the scope of this project, the application of the architectural design and the !CHAOS use cases have driven by the !CHAOS team to choose a Cloud Infrastructure as a Service (IaaS) as infrastructure to deploy backend and frontend services.

The design of the application deployment took into account the following requirements:

- high availability and reliability,
- scalability,
- disaster recovery,

– on-demand deployment following the user requests,

– automatic setting of the number of instances and the size of components according to the parameters chosen by the user,

– auto-scaling on the basis of monitoring information.

2˙1. *Cloud IaaS*. – Within the Cloud glossary, IaaS is a self-service model for accessing, monitoring and managing remote data center infrastructures, such as compute (virtualized or bare mental), storage and networking services. Instead of having to purchase hardware outright, users can purchase IaaS based on consumption, similar to electricity or other utility billing. A consumer can unilaterally provision resources, such as computing, network or storage services, as needed automatically without requiring human interaction with each service provider. These resources can elastically be provisioned and released, in some cases automatically, to rapidly scale up or down, following the demand. In this context, the !CHAOS application deployed on top of a IaaS exploits the characteristics of on-demand service and elasticity. Beside big providers that offer Cloud services based on closed source software, some open communities have been growth in the last years with the aim to develop open-source software to implement public or private Cloud services (IaaS and more). One of the most promising Cloud software with an increasing worldwide community is OpenStack [6], an open source product that can be deployed on open source platforms. OpenStack has strong backing from the industry, with major ICT players directly supporting it; it enjoys a steady growth in terms of both functionalities and developers; it has an open and extensible architecture, mainly written in Python; it interoperates with other Cloud stacks and APIs. Moreover, there is significant experience with OpenStack deployment, configuration and extensions within the INFN. For all these reasons, OpenStack has been chosen as IaaS software on which the !CHAOS application as a Service can be built.

2˙2. *!CHAOS as a Service*. – As described by Catani *et al.* [7], the !CHAOS main services (!CHAOS Data Service—CDS—and Metadata Service—MDS) rely on common backend services, such as a file system, a database and a shared cache object. The !CHAOS team studied a solution for each of these services (both the common ones and the !CHAOS specific ones), to integrate them in a generic IaaS based on OpenStack. The final goal of this activity is to produce a Software as a Service (SaaS) implementation of !CHAOS, in order to provide users the possibility to automatically deploy a complete !CHAOS instance on a private or public Cloud environment. Automating a deployment make the !CHAOS environment creation more efficient, saves time to users and helps them to reduce the possibility of manually introducing configuration errors.

Figure 1 represents a simplified scheme of the !CHAOS deployment on OpenStack. The common backend services are on top of the OpenStack components and can be automatically provided within the Cloud environment, as Platform as a Service (PaaS) components; the !CHAOS frontend services (CDS and MDS) exploit the virtual instances of PaaS components and can be run as unique or multiple instances. The frontend services have to communicate in a bidirectional way with the remote !CHAOS clients, such as the Control Unit (CU) and the User Interface (UI). As the remote clients could not be identified by a public IP address, a VPN service is deployed within the OpenStack Cloud to allow the network traffic to and from the frontend services.

In the current version, the !CHAOS software depends on a POSIX file system, a MongoDB [8] NoSQL database in a cluster configuration and a Couchbase [9] cluster as shared

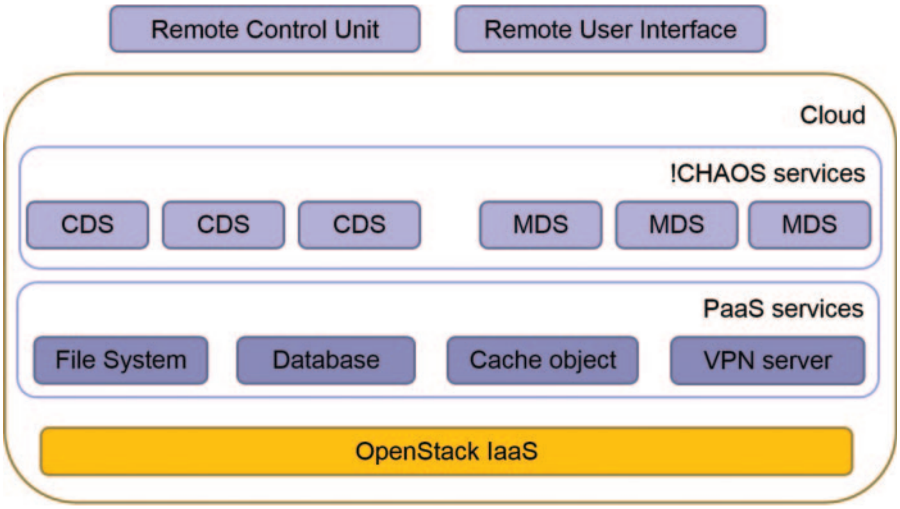

Fig. 1. – Scheme of !CHAOS deployment on OpenStack.

cache object. To supply this PaaS level the !CHAOS team exploited the capability of Heat [10], the OpenStack component that implements the function of infrastructure orchestration, *i.e.* a programmatic approach to create and deploy full stack configurations. A set of Heat templates have been created to deploy:

– A file system based on a Ceph [11] virtual infrastructure composed by 3 Monitor, 3 Object Storage Device (OSD), with a configurable backing store size, and 1 Metadata Service (MDS).

– A MongoDB cluster composed by a configurable number of shards (each of them in replica 3); 3 configuration servers; a configurable number of query routers.

– A Couchbase cluster composed by a configurable number of nodes.

– An OpenVPN server.

To deploy the frontend services, ad-hoc virtual images have been produced, based on an Ubuntu 14.04 operating system.

2˙3. *OpenStack testbed*. – The deployment of the !CHAOS services in Cloud have been tested on the OpenStack infrastructure operated at INFN-CNAF. This infrastructure is based on the Juno release [12] of the OpenStack framework; each node of the infrastructure runs a CentOS 7.1 operating system; the following nodes are deployed:

– 2 controller nodes in HA active/active.

– 2 network nodes in HA active/active.

– 8 compute nodes (a total of 128 cores and 500GB of RAM capacity.

– 3 nodes for database cluster (Percona XtraDB), messaging system (RabbitMQ) and configuration manager (Apache ZooKeeper). Each node is running 1 instance of each service.

A storage space of 16 TB is the backend for a GPFS file system used to store images, virtual machine disks and persistent volumes.

2˙4. *Public Clouds for !CHAOS*. – Regarding the !CHAOS deployment on public clouds, the team has been carried out preliminary studies to check its feasibility. Currently limits do not exist in terms of compatibility for the operation of !CHAOS backend services. As of Amazon and Google Cloud documentation and datasheets, we could use Amazon DynamoDB [13] or Google Cloud DataStore [14] for NoSQL; Amazon ElastiCache [15] or Google Memcache [16] for in-memory cache; Amazon Simple Storage Service (S3) [17] or Google Cloud Storage [18] for archival storage. Dynamic provisioned Amazon Elastic Compute Cloud [19] or Google Compute Engine [20] instances would run the !CHAOS framework applications as CDS, MDS, UIs and EUs. The achievement of such goal would allow a wider and easier deployment of !CHAOS for consumer, small business or enterprise world without having to manage the IT infrastructure and in a dynamic, cost effective, pay-per-use way.

3. – Conclusions and next steps

The beta version of !CHAOS as a Service has been provided at the end of December 2015. The !CHAOS team has been working on a more flexible and reliable version of !CHAOS as a Service [21]. A first activity has been carried out to improve the high availability of the solution: the instances of backend and frontend services have been deployed on different availability zones, that are logical groupings of computing resources, *i.e.* groups of hosts that either share common resources, such as storage and network, or have a special property, such as trusted computing hardware. The deployment of services on different regions (*i.e.* IaaS running on different data centers) has been tested in order to provide the availability of services in case of disaster issues.

Another important improvement has been the dynamic sizing of the !CHAOS platform, on the basis of such parameters chosen by the user (*i.e.* through a web dashboard). This feature has been built by exploiting the Heat APIs. From the beginning of this project, the Heat templates developed to deploy the services have been designed to be highly configurable. Moreover, an activity has been carried on to implement auto-scaling solutions in order to add (or remove) instances of !CHAOS services in case of need, using the auto-scaling feature of Heat.

REFERENCES

[1] Antonucci F. *et al.*, *!CHAOS: a cloud of controls -MIUR project proposal*, INFN 14-15/LNF; `https://www.lnf.infn.it/sis/preprint/`.
[2] `www.infn.it`.
[3] `www.esco.it`.
[4] Bisegni C. *et al.*, *First operational experience with the !CHAOS framework*, in *Proceedings of PCAPAC 2014, Karlsruhe, Germany*; `http://www.JACoW.org`.
[5] Mazzitelli G. *et al.*, *Nucl. Instrum. Methods Phys. Res. A*, **515** (2003) 524.
[6] `www.openstack.org`.
[7] Catani L. *et al.*, *Introducing a new paradigm for accelerators and large experimental apparatus control systems*, *Phys. Rev. ST Accel. Beams*, **15** (2012) 112804.
[8] `www.mongodb.org`.
[9] `www.couchbase.com`.
[10] `https://wiki.openstack.org/wiki/Heat`.

[11] www.ceph.com.
[12] https://www.openstack.org/software/juno/.
[13] http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html.
[14] https://cloud.google.com/datastore/docs/concepts/overview.
[15] http://aws.amazon.com/it/elasticache/.
[16] https://cloud.google.com/appengine/docs/python/memcache/.
[17] http://aws.amazon.com/it/s3/.
[18] https://cloud.google.com/storage/.
[19] http://aws.amazon.com/it/ec2/.
[20] https://cloud.google.com/compute/.
[21] Bisegni C. *et al.*, *!CHAOS status and evolution*, in *Proceedings of IPAC2015, Richmond, VA, USA* May 2015.