An overview of the DII-HEP OpenStack based CMS Data Analysis

P. Eerola[1], M. Komu[2], M. J. Kortelainen[3], O. Kraemer[3], T. Lindén[3], L. Osmani[4], S. Tarkoma[4], S. Toor[3]. J. White[3]

[1]Department of Physics, University of Helsinki

[2]Ericsson Research

[3]Helsinki Institute of Physics

[4]Computer Science Department, University of Helsinki

4.9.2014

ACAT 2014, 1st to 5th of September 2014, Prague, Czech Republic

Contents

DII-HEP project

- Collaboration between
  - Helsinki Institute of Physics (**HIP**)
  - Computer Science Department, University of Helsinki
- Funded by the Academy of Finland 2012-2014
- Goals of the project
  - Explore the latest software stacks for distributed computing infrastructures
  - Construct a secure and scalable setup for scientific applications
  - Use the CMS analysis and production framework as a test case

Initial setup

HIP runs a *distributed* CMS Tier-2 site T2_FI_HIP based on the Advanced Resource Connector (ARC) middleware

- Jade 768 cores main ARC CE, shared by CMS and ALICE
    - Madhatter dCache Storage Element connected to Jade
- Alcyone 892 cores ARC CE, shared resource, 10 km from Jade
- Korundi 400 cores ARC CE, shared resource, 10 km from Jade

Initial setup

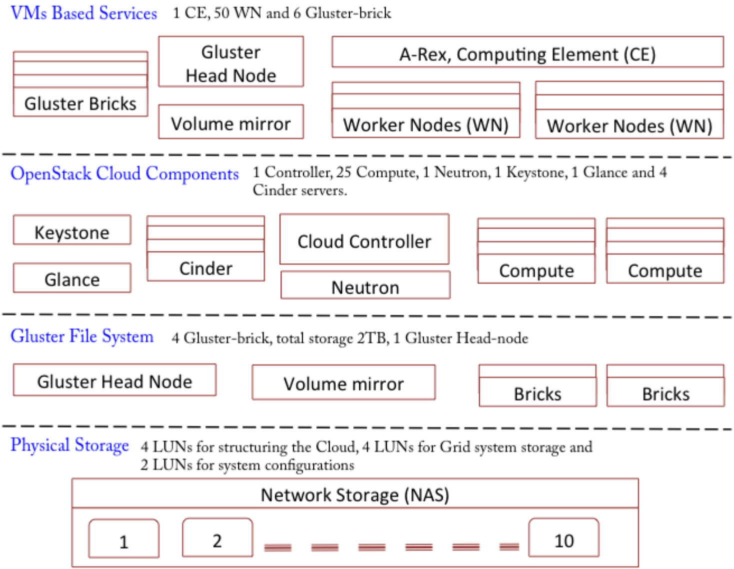
- Start from the existing production environment and integrate cloud resources into it
- Use the existing grid (ARC) interface to allow all applications to run unchanged
- The setup is **cloud based**, **grid enabled** and using **Gluster FS**
- Cloud resource
  - OpenStack based
  - Gluster File System
  - Components for system security
  - Runs on a small part of the Ukko cluster (1920 cores) also 10 km from Jade
- ARC is used to control the execution of jobs
  - Virtual ARC CE nodeslab-0002
  - Condor is used as the local batch system
  - CERN VM File System (CVMFS) is used for application software
- Monitoring with Graphite and CMS SAM and Nagios jobs

Initial setup

OpenStack

- Open Source infrastructure for public and private clouds
- Components
  - Nova (Compute)
  - Quantum / Neutron (Network)
  - Swift (Object based storage)
  - Cinder (Block storage)
  - Keystone (Identity management)
  - Glance (Image management)
- Started with **Folsom** release and the current setup is running on **Havana**, testing done with **IceHouse**
- GlusterFS used for the instances and data of the jobs
- OpenStack deployed on Ubuntu 12.04 LTS
- VMs based on Scientific Linux CERN 6.4

Initial setup



Initial setup

- So far the system has run a maximum of 200 concurrent jobs.
- The average CPU efficiency of all CMS jobs over the previous year is 83 % from the Swedish Grid Accounting System (SGAS).

Machine view for nodeslab-0002.nlab.tb.hiit.fi

Start month 2013-08 End month 2014-08 Submit

Top 10 projects for the selected date range

	Walltime days	Efficiency	Number of jobs
cms	9163	83	147290

Initial setup

- The CPU efficiency loss due to virtualization was measured to be 4 % by running the HEPSPEC 2006 benchmark on bare metal and on VMs.
- This setup provides efficient resource management, but semi-statically as the resources are manually added or removed.
- This semi-static setup has been described in a CHEP 2013 paper [1].

Elastic setup status

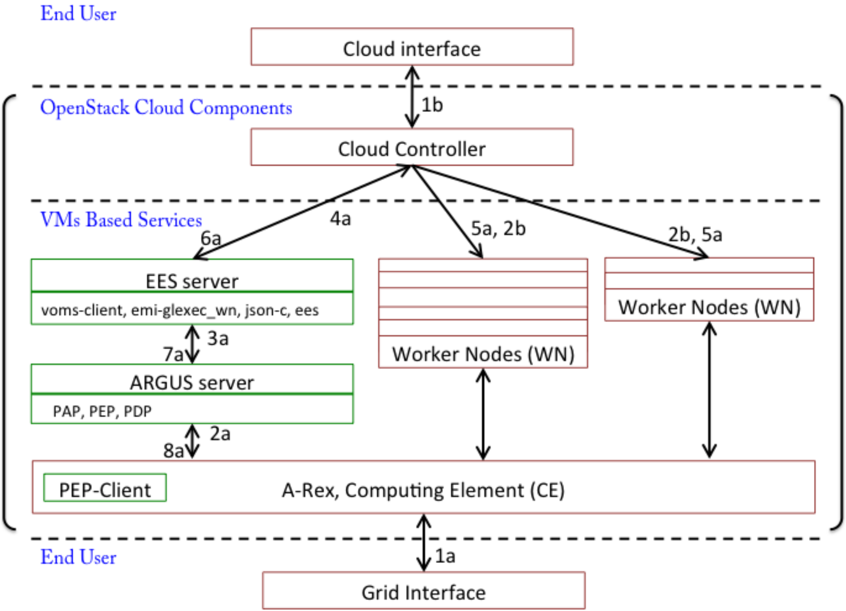
- The Argus Execution Environment Service (EES) is part of the EMI release
- EES is a service that may co-exist with Argus to start VMs or send jobs to a cloud
  - EES has been tested with Argus (PEPd)
  - EES has a plugin written for OpenNebula
  - Prototype EES OpenStack plugin has been developed.
- Elastic VM-provisioning can be implemented with the new OpenStack plugin

Elastic setup status

The EES OpenStack plugin

- Controls the start/stop of VMs in the OpenStack cluster
- Communicates with the OpenStack controller using the JSON API
- Written in C
  - Standard JSON-C library used to parse the OpenStack JSON responses
  - C Curl library used to send/receive JSON to OpenStack
  - XACML (eXtensible Access Control Markup Language) obligations are handled by the EMI saml2-xacml2 library
- For details on the EES OpenStack plugin see this article [2].

Elastic setup status



Host Identity Protocol

Host Identity Protocol (HIP), http://infrahip.hiit.fi/

- Designed for mobile networks and standardized in RFC 4423, 5201–5207, 6092
- Provides persistent cryptographic identifiers
- Supports both IPv4 and IPv6 addressing
- The namespace is secure in the sense that HIP identifiers cannot be spoofed
- The Host Identifiers (HI) are not routable, so they are translated into routable addresses (locators) between network and transport layer
- The mapping from HI to network layer locators is dynamic, which enables end-host mobility and multihoming
- This can be useful for VM live migration and site renumbering
- The HIP connections are typically protected with IPSec
- Application software can be used unchanged with the HIP protocol
- HIP can in a way be used as a VPN, but without a gateway

Host Identity Protocol

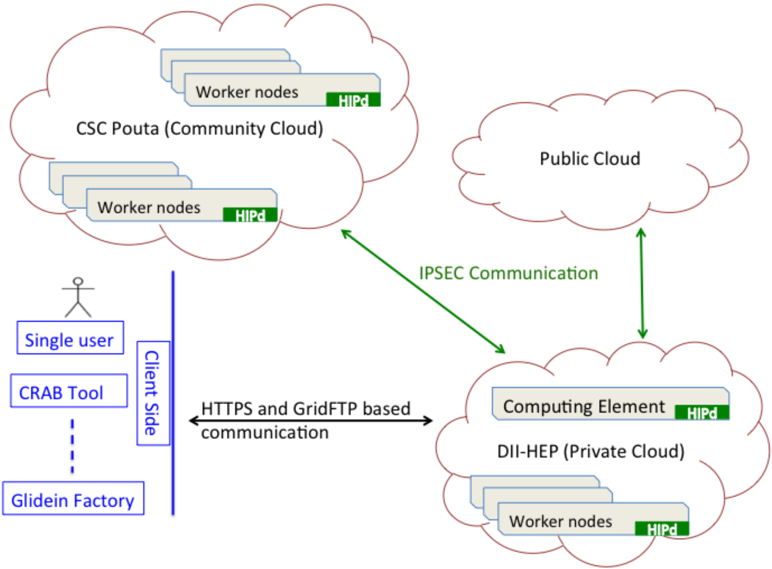
HIP protocol performance tests:

- Performance for CMS CPU intensive CRAB (CMS Remote Analysis Builder) jobs have been compared for three different cases: no specific security mechanism, HIP protocol and SSL/TLS based security.
- The jobs run for about 170 minutes and each test was running for a about a week on 200 concurrent job slots
- The summary of these tests is that the HIP protocol creates negligble CPU and network overhead on the CE
- For more details on the HIP protocol use in DII-HEP see this paper [3]

Secure Hybrid Cloud setup

- CSC, The Finnish IT Center for Science has a new data center in Kajaani, more than 500 km from Helsinki
- CSC has a HP 9216 cores (Q4 $\approx$ 17000 cores) supercluster Taito, with 16 cores and 64 GB RAM on most nodes
- The Pouta cloud is an Infrastructure as a Service running on a part of Taito
- OpenStack Grizzly is in use on Pouta
- In Helsinki there are currently 40 cores running on OpenStack
- The HIP protocol and Condor flocking are used for cloud bursting from Kumpula to Kajaani
- Tests with 44 cores running on Pouta and connected to the OpenStack in Helsinki are currently in progress

Secure Hybrid Cloud setup



Future work

- Work on making the EES plugin for OpenStack production quality is continuing
- Work on scaling up the intra-cloud setup based on HIP is continuing
- The Jade cluster is planned to be replaced by a cloud setup on Pouta
- This cloud setup could be used by other scientific applications, like in the Finnish Grid and Cloud Infrastructure consortium

Summary

- A scalable infrastructure for CMS jobs has been created which is **cloud based** and **grid enabled**
- The ARC grid interface has been used to provide the interface for CMS applications, so that all CMS jobs can run unchanged
- The cloud setup itself does not rely on grid tools
- The same setup could be used for other applications as well
- The cloud provides flexibility in managing the infrastructure
- Up to 200 simultaneous jobs have run so far
- An EES OpenStack plugin prototype has been written
- Initial results show that the HIP protocol can be very helpful in resource mobility and security
- Cloud bursting has been demonstrated using the HIP protocol and Condor flocking for 44 cores so far over a distance of more than 500 km

[1] S. Toor, L. Osmani, P. Eerola, O. Kraemer, T. Lindén, S. Tarkoma and J. White, *A scalable infrastructure for CMS data analysis based on OpenStack Cloud and Gluster file system*, J. Phys.: Conf. Ser. 513 062047.

[2] J. White, S. Toor, P. Eerola, T. Lindén, O. Kraemer, L. Osmani, S. Tarkoma, *Dynamic Provisioning of Resources in a Hybrid Infrastructure*, submitted to PoS.

[3] L. Osmani, S. Toor, M. Komu, M. J. Kortelainen, T. Lindén, J. White, R. Khan, P. Eerola, S. Tarkoma, *Secure Cloud Connectivity for Scientific Applications*, submitted to IEEE Tran. Cl. Comp.