Table of Contents

Summary of pre-GDB meeting on Cloud Issues, July 9, 2013 (CERN)

Agenda

https://indico.cern.ch/conferenceDisplay.py?confId=253858

Experiment Reports about Cloud Usage

LHCb - F. Stagni

Virtualization: the mean to privatize the last part of the infrastructure, the OS!

- Does not mean that virtualization is a requirement every where

Many different resources today: all have in common not to use any internal queuing

- LCG sites: direct submission to CREAM CE
- 1 large pure DIRAC site : direct submission to Torque
    - Similar to a LCG site without the grid layer
    - (improperly?) Called a DIRAC CE but no DIRAC SW installed at site: rather a DIRAC plugin allowing direct submission to Torque
- HLT farm: no virtualization, PVSS driven, pure pull model (no pilot submission involved)
    - 17% of the LHCb CPU since the end of data taking
- Clouds, VAC, virtualized infrastructure

CVMS is a LHCb requirement at each site and helps to use different kind of resources by keeping the same way to access the SW

- No attempt to install CVMFS on the fly (in user space): considered a requirement that a site must fullfil
- DIRAC set up from CVMFS
- CVMFS must be preinstalled by the site

One pilot to fly everywhere

- Currently in development, soon available

Main cloud infrastructures used by LHCb

- IBEX@CERN: 100 instances
- OpenNebula@PIC: 100 instances
- Also tests with BOINC: http://lhcbathome.cern.ch/Beauty
    - Non trivial configuration but out of the box integration with DIRAC

Attempt backfilling ("job masonry") of any slot based on the resources left: done by flexible MC jobs (number of events adjusted to the resources available)

- Will use the proposed mechanism to get the information about resources left in the slot
- Also interested to run multicore jobs if possible: able to use Gaudi-MP for everykind of job

LHCb VMs based on CERNVM + amiconfig for contextualization

- Currently based on SLC5, soon SLC6
- LHCb doesn't accept that images are modified by sites

LHCb is not planning to use HLT farm for offline work during inter-fill gaps after LS1 because of the deferred trigger strategy where the data acquired is not necessarily processed immediately to take advantage of the 24h power available in HLT farm.

ATLAS HLT Farm (Sim@P1) - A. Di Girolamo

Several people/teams involved in the project: base infrastructure, network, cloud (OpenStak) experts, grid infrastructure, TDAQ

Usage currently ramping up.

Based on OpenStack Folsom release: keystone, Glance, Nova, Horizon

- VLAN isolation: 1 Gb/s for each VM
- VM Image based on CernVM 2.6.0
- Using the network infrastructure of HLT farm but bypassing date-core/SFOs for CASTOR access (direct access)
- VM monitoring with Ganglia

Was able to run 12K jobs in // over a maximum of 16.5 K jobs: limitation due to Condor master, work in progress to address it

- One Condor master cannot handle more than 10K running jobs
- Startup hicups due to too many jobs starting up at the same time
- Some job slots have been running HC jobs for 20 days

Open questions

- Streamline VM lifecycle with different clouds
- Graceful termination of VMs: ATLAS interested

CMS HLT Farm - D. Cooling

Also involving a lot of people...

200 KHS06 available

- 2x10 Gb/s data network to CERN.
- 1 Gb/s control network: the one used initially for cloud to avoid network reconfiguration

Preliminary tests with OpenSTack Essex in 2012 successful

- Ensure no interference with normal HLT operations when they are in progress
- Cloud technology seen a way to sandbox reprocessing activities

CVMFS deployment and move to xrootd access were preliminary steps to cloud deployment on HLT

Many initial issues but now running at steady 6000 jobs with 100% sucessful jobs most of the time

- Mainly reconstruction jobs
- Several problems track down to network saturation issues: most of them solved, still a problem that occured a couple of times and has not been full understood yet (the 7am syndrom!)
    - Job retried and successful then
    - a 11 GB/s limitation to EOS (despite use of 10 Gb) under investigation: considering a network upgrade
- Every hypervisor running a Squid server to avoid overloading the Frontier servers

Similar infrastructure already used to submit analysis jobs to clouds in Italy and UK.

HLT farm turned out to be a very good testbed for testing cloud MW (OpenStack) at scale.

CMS Private Cloud Usage - M. Cinquilli

// testing of several cloud infrastructures: OpenStack, OpenNebula, RackSpace, StratusLab

- Using GlieinWMS to handle VM lifcycle
- Based on CernVM 2.6.1 + Ganglia + Condor + CVMFS
- CERN AI: 800 core quota, SLC6/CentOS6 images
    - ♦ Storage in EOS

Agile Infrastructure performance: higher success rate than regular T2

- CPU efficiency seems at least comparable if not better than on bare metal WN

Issues found: EC2 differences between implementation

- Would like a common interface if possible
- DeltaCloud seen as a possible alternative: exposes DeltaCloud, EC2 and DMTF CIMI
    - ♦ DeltaCloud requires a "proxy" server
    - ♦ Starting to work on integration of DeltaCloud with Condor
    - ♦ Currently in maintenance mode
- libcloud seen as another interesting, more active alternative but not yet integrated to Condor
    - ♦ libcloud support Deltacloud as a backend...

Contextualization

- Currently maintaining a golden image for every kind of images or any update of the SW
- Would like to move to dynamic contextualization to reduce the number of updates to images: initial tests done with CernVM + amiconfig
- Started to work on CloudInit use and sharing work with CernVM team

Would like to see a general policy about which tools/interfaces a grid site should provide...

ATLAS Private Cloud Usage - R. Taylor

Many clouds used, mainly based on Nimbus and OpenStack

- 780k jobs over the last 15 months
- Both Xen and KVM hypervisors

VM images can be run both on Xen and KVM (same image): plan to use the same images over all clouds (15+)

Cloud usage in Atlas based on CloudScheduler developped at UVic and NRC

- Based on Condor
- Supports a lot of different cloud backends
- Take in charge the instanciation of the VM based on Condor queue contents: dynamically manages the quantity and type of VMs
- Installable with pip

VMs and dynmaic Squids (Shoal)

- 1+ Squid VM per cloud
- Other VMs dynamically discover it/them

Infrastructure now established and in production

Discussion

Every experiment wants to minimize the number of images to maintain: critical for sustainability

- Use CernVM as the base image, use CVMFS to avoid image updates triggered by experiment SW updates.
- User and site contextualization is required for experiments: no intent to produce an image per infrastructure, need to pass a few parameters at instanciation time (like connection information for pilot factory)
    - So called SSH contextualization is not really a solution as it requires the images to contain the SSH credentials... better done with true contextualization.
- Ian Gable: site contextualization is site dependent and often requires hacks in the image because of inconsistent metadata used at each site for contextualization. Would need a defined list of metadata to use.
    - Michel: can Ian provide a summary list of his experience with "metadata inconsistency" across sites? Please post to the list.

Efficient access to data remains an open question: more standard access to the storage will allow simpler site contextualization

- More difficult for a VM to have the appropriate kernel to access a particular storage for example

Running the same image everywhere requires agreement on which metadata are published by sites

- Ian G. will make a list of the problems identified by Atlas in its experience with private clouds for further discussion on the list

VM Graceful Termination

Vac - A. McNab

Vac = Vacuum

- Infrastructure as a Client: VMs pop up without the need to be triggered by a submitted pilot job
  - Created from the Vacuum!
  - VM is contacting the central queue/pilot factory to get a payload: pure pull model
- Factory nodes (very physical machine) decide which types of VM to create: 1 or more VM per VO
  - 90% of the work done by 2 or 3 VOs: worth the effort to provide them the appropriate VMs
- No head node: each factory node communicate with others to discover what needs to be started or can be reused
- Contextualization implemented: can run the same images as clouds
- Used at 4 sites in UK: 3 are running routine LHCb production MC
- Endpoint service type for VAC registered in GOCDB
- Accouting data written in PBS and BLAHP format, allowing to reuse APEL parsers
- BDII publication not yet implemented but not really needed

Graceful termination: Vac strategy is to use machinefeatures and jobfeatures (NFS exported to VM from factory node (hypervisor))

- If VM not shutdown before the time defined, it is killed
- A shutdown command defined that allows the VM to shutdown itself (with optional arguments to use as the shutdown message)
- Detailed termination strategy as a contract
- Graceful termination is a chance given to job/VM but not required to be used
- Sites motivated by being able to shutdown their resources without impacting job efficiency for the VO
- Allow job "masonry" (backfilling): use as much of the available CPU until the advertized shutdown time by packing shorter jobs

Target shares: VAC avoids "fair shares"...

- No history recorded
- Implemented instantaneously if there is competition between VOs
- Unfair in the sense that no attempt to achieve them over a period of time
- Possibility to use them updating the target shares based on the history and pushing the updates to factory nodes: slow update, long time occonstant
  - Updates could be computed from accounting

Discussion:

- Security issue: how to do the initial authentication:
  - Need trust between ops team and the experiment team. Idea is to have a service certificate which is registered centrally.
- Traceability: same issue in other clouds
- VM lifetime used: like a batch farm slot for now
- Requires some SW to be installed on the hypervisor. How would that work for hyperV
  - Yt does not. They support XEN and KVM via libvirt

Discussion

LHCb has difficulty to use efficiently graceful termination because it is difficult to know the power of the machine

WLCG Ops Coord set up a TF to track the implementation of machine features

- Sites with virtualized environments welcome!

machinefeatures implemented at CERN on all batch WNs (virtualized or not)

Shutting down a VM requires a command hooking back into the cloud manager: VM shutdown command is not enough

- Done at CERN for OpenStack and OpenNebula

How site contextualization initializes machinefeatures is documented in a Twiki page that must be circulated on the list by Ulrich

Target/fair share in clouds

Introduction - U. Schwickerath

Batch systems fair shares are implement because resources are overbooked and overbooking is reflected in batch queues

In clouds, we still need to make an efficient of idle resources still meeting pledges

- Current solutions often lead to static partitionning
- No concept of queue in clouds: a requests that cannot be fullfilled imediatly is refused
- Some attempts to use batch systems to schedule VMs to reimplement queueing but users generally don't want queueing...

Economic model

- Old ideas based on a central broker, discussed in particular by the Torino group
- Difficult to compute the price of a resource for resources that are paid upfront

Bottom line: keep it simple, avoid "island" solutions

- Ensure that a single user cannot eat all resources when they are underused
- Allow to "penalize" big users when there is competition to give small users a chance to get access to the resources

Interesting ideas based on virtual currency and virtual bank

- Resource providers regurlarly update customer (virtual) bank accouts
- Users issues taks stored in a central queue
- Users send "requests for offer" to all providers and use the one with the lower cost: the cost must be lower than the balance of their virtual bank account
- No existing implementation yet for this model...

Discussion

Several reactions that this looks over complicated.

DIRAC trying to prototype a simple/simpler economic model.

Tony: to ensure a reasonnable job turn-around, a site has to shutdown regularly VMs (max lifetime)

- What to shutdown may be based on the accounting history
- A site that has a supported VO running under pledge may start VM for the VO to give it a chance to run a payload

Philippe: VMs over pledged may be started with a shorter lifetime

- Would also like the resources used over pledge because there are some free resources to be "cheaper" than normal resources

VAC idea of computing credits based on accounting vs. pledge is interesting

Probably difficult in the IaaS cloud world to implement fair usage without keeping a few free slots for VOs under quota

Wrap-Up

Continue working on machinefeatures and graceful termination

- More sites are welcome to participate

Continue discussion/tests on fair usage of clouds: possibility of VAC approach in a IaaS cloud?

Summary will be done at GDB: look at GDB agenda.

This topic: LCG > 20130709PreGDB
Topic revision: r2 - 2013-07-16 - MichelJouvin