Imperial College London

HLT as a cloud status

David Colling

Imperial College London

People contributing ...

- Adam Huffman (Imperial College, Grid/Cloud devops),
- Alison McCrea (CERN, CMS operations),
- Andrew Lahiff (RAL and CMS Operations),
- Mattia Cinquilli (CERN, IT Support for Distributed Computing),
- Stephen Gowdy (CERN),
- Jose Antonio, Coarasa (CERN, CMS Online devops),
- Anthony Tiradani (Fermilab, CMS glideinWMS),
- Wojciech Ozga (CERN and AGH University of Science And Technology in Krakow, CMS Online devops)

- (and anybody I have missed)

Imperial College London

So why use the HLT?

Clearly this is taking effort from (parts of) several people so why bother?

The simple answer is that it is a big resource that we cannot afford not to use:

Node type	Number	cores/node	HS06/core	Total HS06	Disk/node (GB)
c1950	720	8	9.1	52416	72
c6100	288	12	17.3	59788.8	225
c6220	256	16	24.1	98713.6	451

- Total ~200K HS06 (of which ~150K HS06 is easily available - more than T0 and comparable with the total T1 cpu request)
- (Essentially) no storage available
- 2 Network paths available to CERN – 1 Gb/s Control Network, 2x10Gb/s data network
- All nodes have 2GB/core

OpenStack (Essex) installed in 2012 and initial tests with protein folding were very promising so we decided to go ahead with trying to use it for real CMS work.

Imperial College London

Using the HLT

The plan is to have the HLT available as a resource for (nearly) all of LS1, but then to use it as opportunistically after LS1 (in machine breaks etc, even for interventions that last more a few hours). However, when it is need as an HLT there must be no interference from this parasitic use. It is hoped that a cloud infrastructure will help to enable this.

The HLT was a single use cluster which meant that it didn't need the monitoring infrastructure that you would expect/need for a multipurpose

Only CMS data going from the detector to CERN IT went over the data and all other data went over the control network.

We decided to focus on reprocessing (to start with at least) and to reprocess the 2011 data.

Imperial College London

Initial Configuration

- CMSSW served over CvmFS
- Data read from and written to EOS over xrootd
- All data read and written over 1Gb/s link
- Single frontier server installed (on cms-srv-c2c01-14)
- Submission via glideinWMS
- Images are SL5 (built with BoxGrinder)

Imperial College London

Initial results

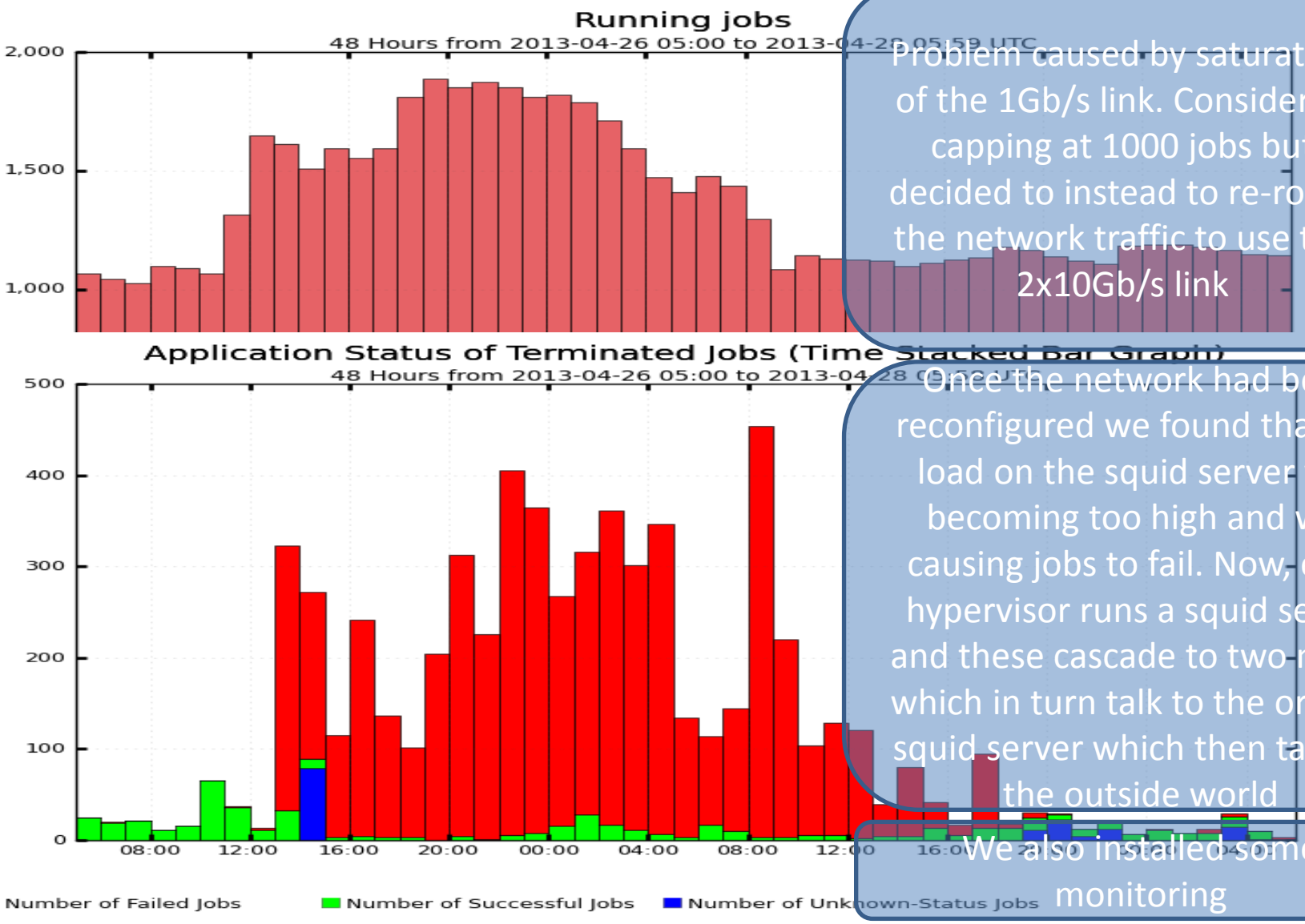
Found many, often minor but annoying, problems.

These include:

- Permissions problems with xrootd and EOS
- VMs dying because access to CvmFS was not available fast enough
- OpenStack EC2 not Amazon EC2 causing many minor problems all of which required modifications to the glideinWMS.
- Behaviour in clouds is different from behaviour in Grids so glideinWMS needed to learn how to handle the situations differently
- OpenStack controller can be “rather fragile” when asked to do things at scale so glideWMS learnt to treat it gently.
- glideinWMS loosing track of jobs (often through fragility of OpenStack) and jobs ending up in “shutoff” state
- ...

Gradually been working our way through these with the set up becoming functional and then more robust as we go...

Initial results – basic limitations



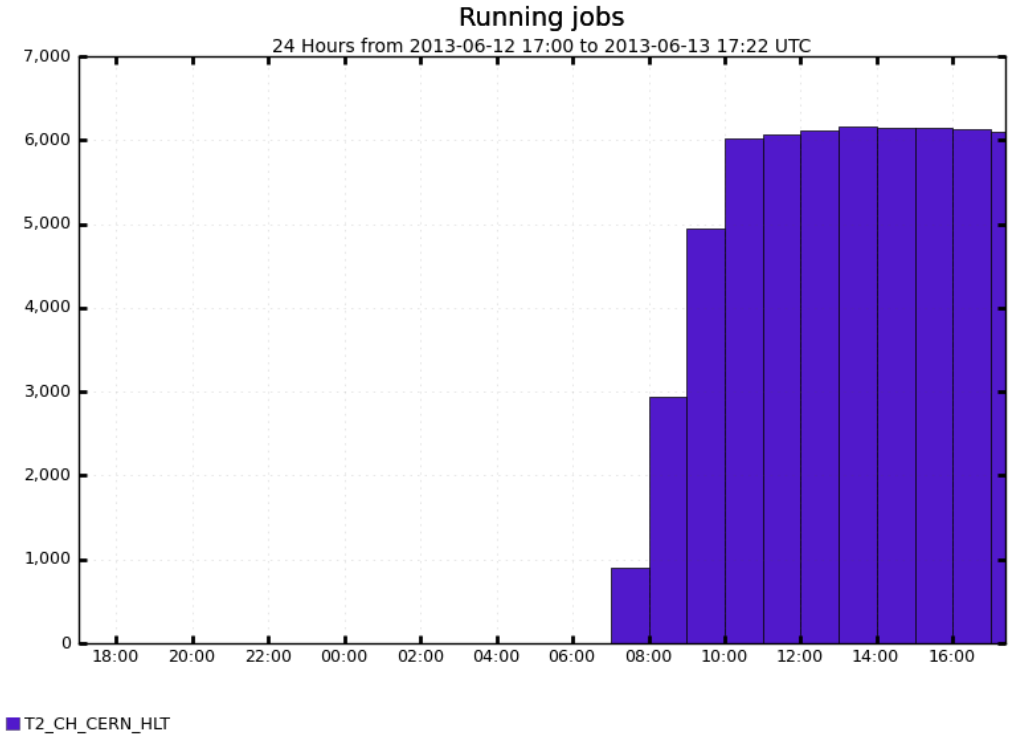
Problem caused by saturation of the 1Gb/s link. Considered capping at 1000 jobs but decided to instead to re-route the network traffic to use the 2x10Gb/s link

Once the network had been reconfigured we found that the load on the squid server was becoming too high and was causing jobs to fail. Now, each hypervisor runs a squid server and these cascade to two-nodes which in turn talk to the original squid server which then talks to the outside world

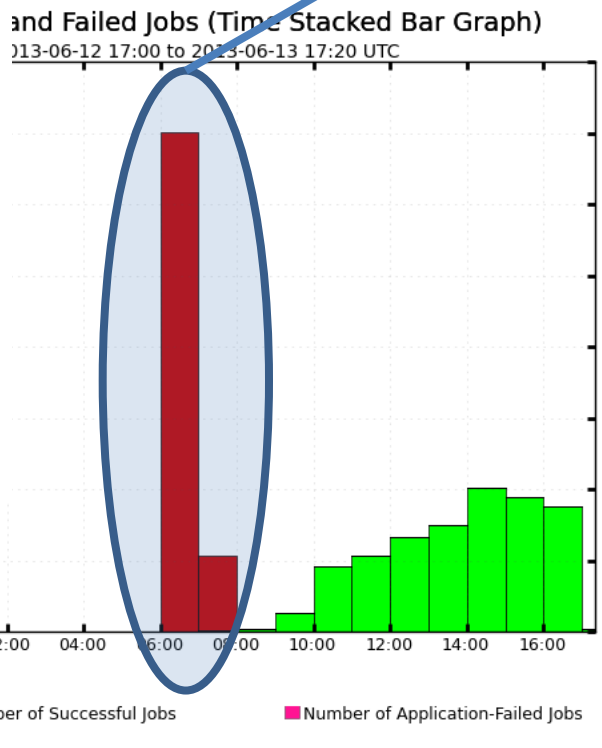
We also installed some monitoring

Current Status

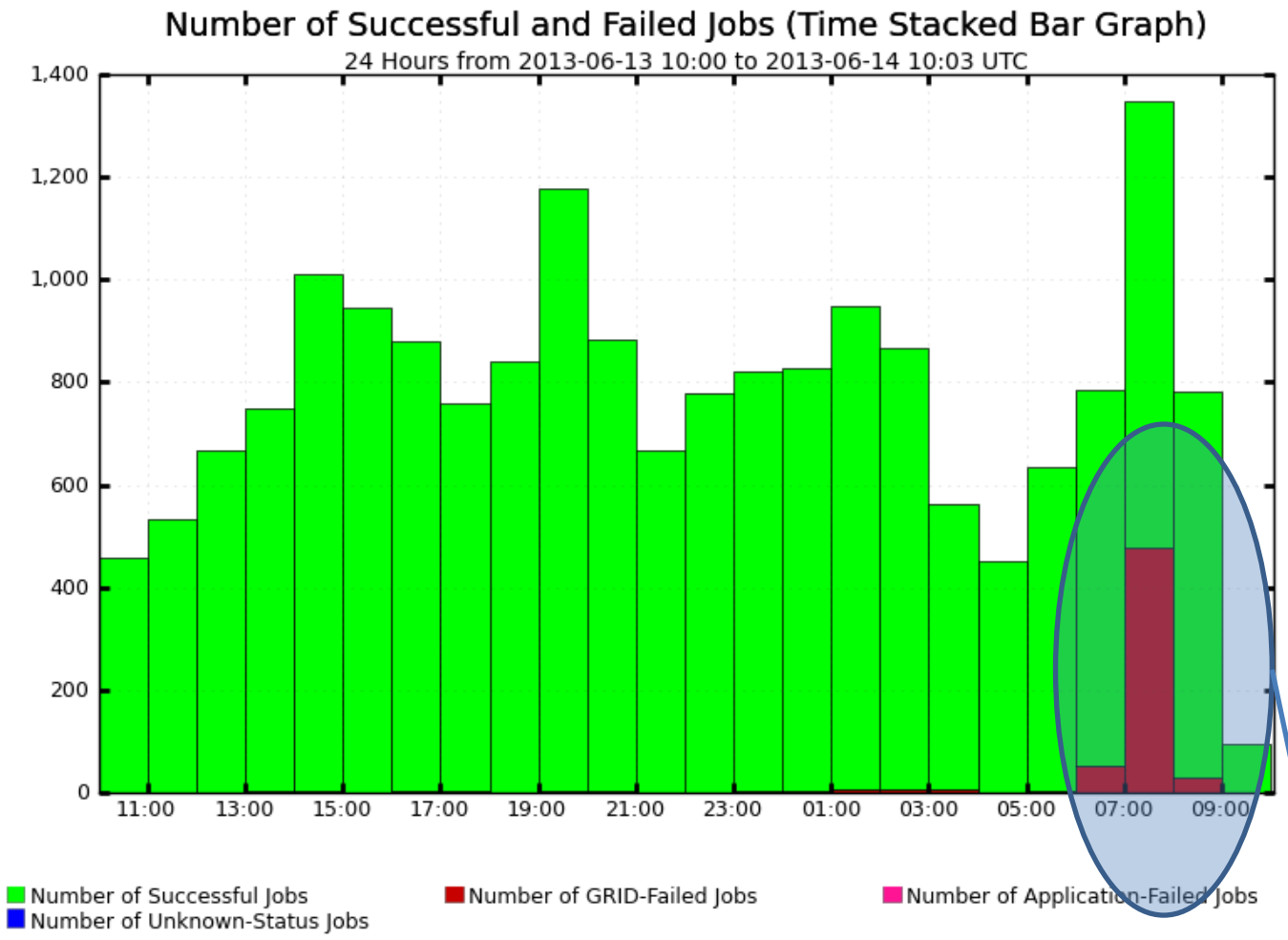
As of yesterday, we are running with a new pre-release of condor and ...



Workflows submitted with wrong requirements, were cancelled



Overnight

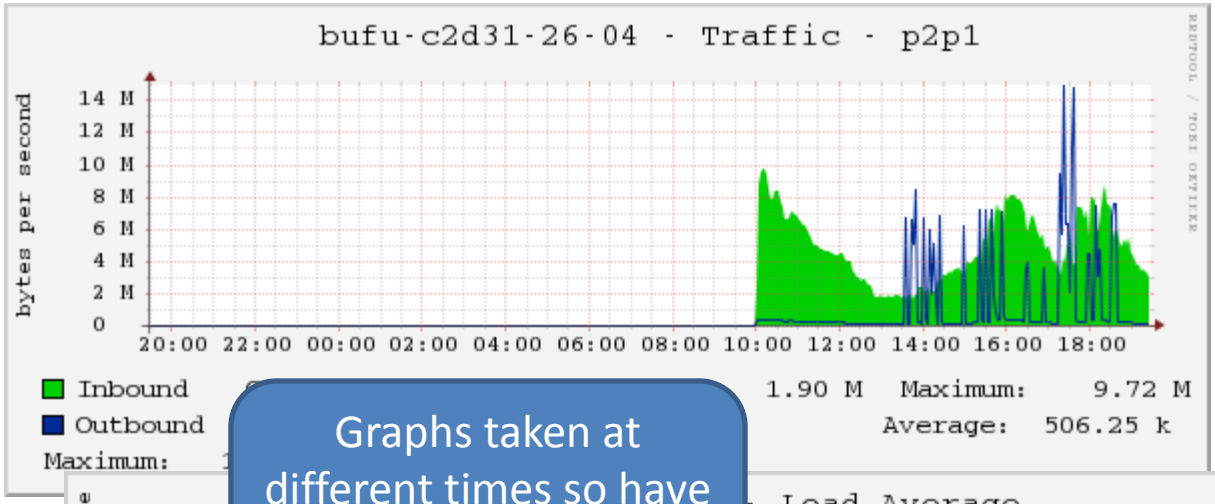


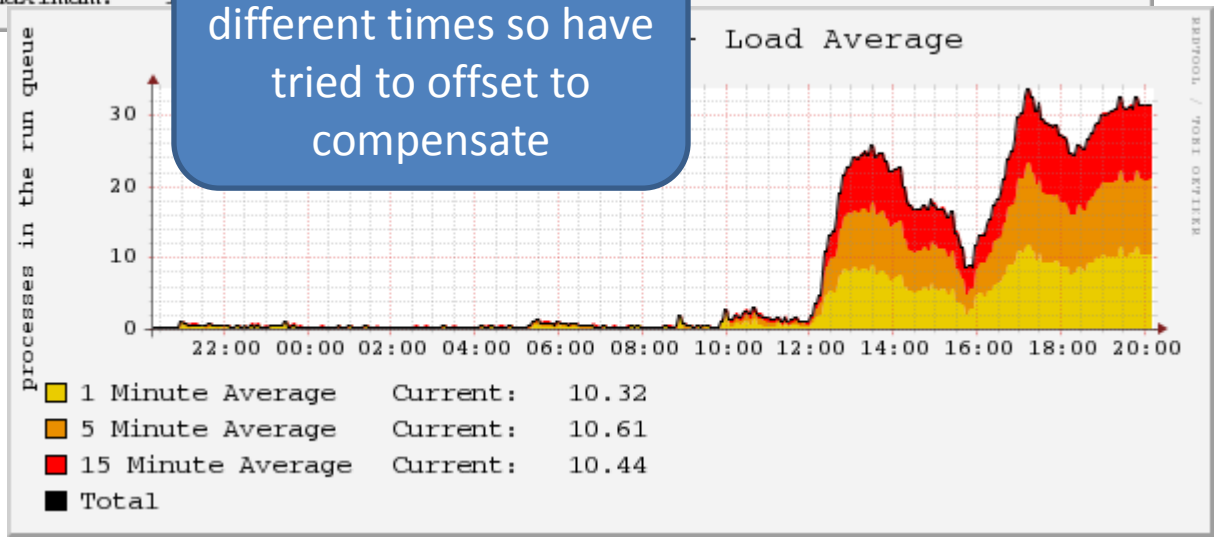
Ran a steady 6000 jobs overnight.

Some problem at ~7am (being investigated) but did recover, still 97% successful!

Current Status

Looking at the monitoring on individual machines shows interesting results





Graphs taken at different times so have tried to offset to compensate

Network activity starts about 2 hours before load average on the machine increases.

Still appear to be network limited.

20Gb/s/8=2.5GB/s
2.5/550nodes=4.5Mb/s

However not so slow that jobs die.
Will have network monitoring soon!

Imperial College London

Next...

- Establish that the setup is now robust (and continue to indentify ways of catching problems and dealing with them)

- Monitoring of the main data network (may even have happened by now) and extend the monitoring.

- Run a fewer number of jobs to see what they would look like if not network limited.

- Possibly upgrade network

- Start reprocessing

Imperial College London

After that...

- Start to look at migration strategies so that we can use it as opportunistically after LS1

- Reduce the involvement of core HLT sysadmins - this is not their (core) job. We have started to do this a little already.