T2_US_Nebraska
Site Report

HEPiX Fall 2016
Garhan Attebury
Holland Computing Center

• Provides research computing for all University of Nebraska entities

• ~28k cores in four clusters (and an openstack ‘cloud’)

• **USCMS Tier2 site**
  T2_US_Nebraska
  6,944 cores (6,304 2+GB mem slots)
  4.6PB HDFS storage
  + 1,500 slots and ~2PB storage ‘soon’
  100Gb ESnet/LHCONE/Internet2
  IPv6 wherever possible

• CMS Tier3 for UNL, KU, KSU

• Heavy involvement with the OSG
In the beginning …
SRM + LVS balanced GridFTP

- SRM 1 → WAN / Local Transfers
- SRM 2
- SRM 3
- TFC magic

- LVS GridFTP
  - single virtual IP address
  - WRR hashing across real servers

- mkdirs / rmdirs via fuse HDFS

- GridFTP-HDFS
  - GridFTP 1
  - GridFTP 2
  - GridFTP 3
  - ...
SRM + LVS balanced GridFTP

- 3x servers that always say ‘go talk to this thing…’?
- Java issues (OOM, cert problems, deadlocks, ‘bugs’)
- CMS doesn’t actually need SRM specific features
- Globus GridFTP supports necessary metadata ops
- CMS software happy with gsiftp in place of srmv2

- Why do we have SRM again ???
SRM + LVS balanced GridFTP

* Had to tweak site-local-config.xml to use gfal2
* Tweaked storage.xml to contain LVS virtual address
* SAM tests used outdated lcg-utils and gfal1 (patch developed)
* Bugs bugs bugs: race conditions, umask issues, etc

* All fixed up and happy (so far) !
LVS GridFTP Only

- Single SE endpoint
- 3x fewer hosts
- Easy maintenance
- Easy scaling
- IPv4 and IPv6

WAN / Local Transfers

LVS GridFTP

- single virtual IP address
- WLC hashing across real servers
- metadata ops via gridftp-hdfs

GridFTP-HDFS

GridFTP 1 → GridFTP 2 → GridFTP 3 → ... → GridFTP 12
The recipe @ Nebraska

- Outside world only connects to VRRP addresses
- Initial response (control) direct routed
- All data path bypasses directors
- Both IPv4 and IPv6 VRRP addresses
- arptables to solve ARP problem
- Naive rc.local scripts to add secondary IP addresses
- Puppet configs described here
The recipe @ Nebraska

• **Documentation:**
  
  
  Red Hat Load Balancer Administration (EL7 version)
  
  [https://github.com/gattebury/gridftp-with-lvs](https://github.com/gattebury/gridftp-with-lvs)

• **Hardware side:**

  2x LVS ‘directors’ (cheap Pentium D class calculators)
  
  12x 10Gb GridFTP-HDFS ‘realservers’
  
  Flat Layer 2 network between directors/realservers

• **Configuration:**

  Keepalived via Puppet: [github.com/arioch/puppet-keepalived](https://github.com/arioch/puppet-keepalived)

  Direct routing with WLC (Weighted Least Connection)
Updating base OS (never soon enough)

“Enterprise” means the package/kernel you have is always one notch behind than the one you want

• RHEL5 —> RHEL6 transition
  • Ran SL5 chroot environment on SL6 worker nodes (thanks to built in Condor support)
  • Lacking tooling for maintaining/updating chroots
  • Required random bind mounts
    autofs + cvmfs + bind = lots of (╯°□°）╯︵ ┻━┻
Updating base OS (round two)

- RHEL6 —> RHEL7 transition
  - Docker is trendy, lets be trendy \o/
  - Built in support for containers… both EL7 and Condor
  - Lots (no really, A LOT) of the industry doing it already
  - … sounds like a win?
Docker + EL7 @ Nebraska

• All CentOS 7.2 - CE, SE, and workers

• Workers are fed 8-core + 20GB condor jobs, one container per job — Nebraska is multicore only

• Dockerfile at: https://github.com/unlhcc/docker-osg-wn-el6
  Docker hub at: https://hub.docker.com/r/unlhcc/osg-wn-el6/

• Writeup and presentation from Derek Weitzel:

• Performance penalty < 5% of native (SL6 container on CentOS 7 host)

• Flexibility of container placement (even some in Anvil)

• Happy with it for multiple months, no complaints
SDNification (?) of transfers

- Visibility into transfers (who? what?)
- Accounting of transfers
- Steering of transfers (automagically)
GridFTP + XIO Callout + Openflow

XIO callout gives: **SRC_IP + DST_IP + DST_PORT** of data path

Source port unknown, but can assume anything from **SRC_IP** to a given **DST_PORT** is part of the calling transfer

<table>
<thead>
<tr>
<th>Control Path</th>
<th>Data Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctrl port = 38150</td>
<td>data port = 35130</td>
</tr>
<tr>
<td>ctrl port = 2811</td>
<td>data port = 50104</td>
</tr>
</tbody>
</table>

Client: 129.93.239.161

Server: 129.93.239.187

**connstr = 129.93.239.161:35130, download**
SDNification: Future work

• Write a similar callout for xrootd-hdfs

• Add a UID to a transfer to group streams

• Other CC*DNI related work
  • Security — bypassing IDS, SFC, etc …

• More hardware testing (table limits, ONOS quirks, hardware quirks)

• All T2 transfers behind this, steered around Bro IDS, in production