Site report UiO

NDGF AHM 2020-2
Darren, Vincent, Maiken

Overview

- Compute
- Storage

Compute

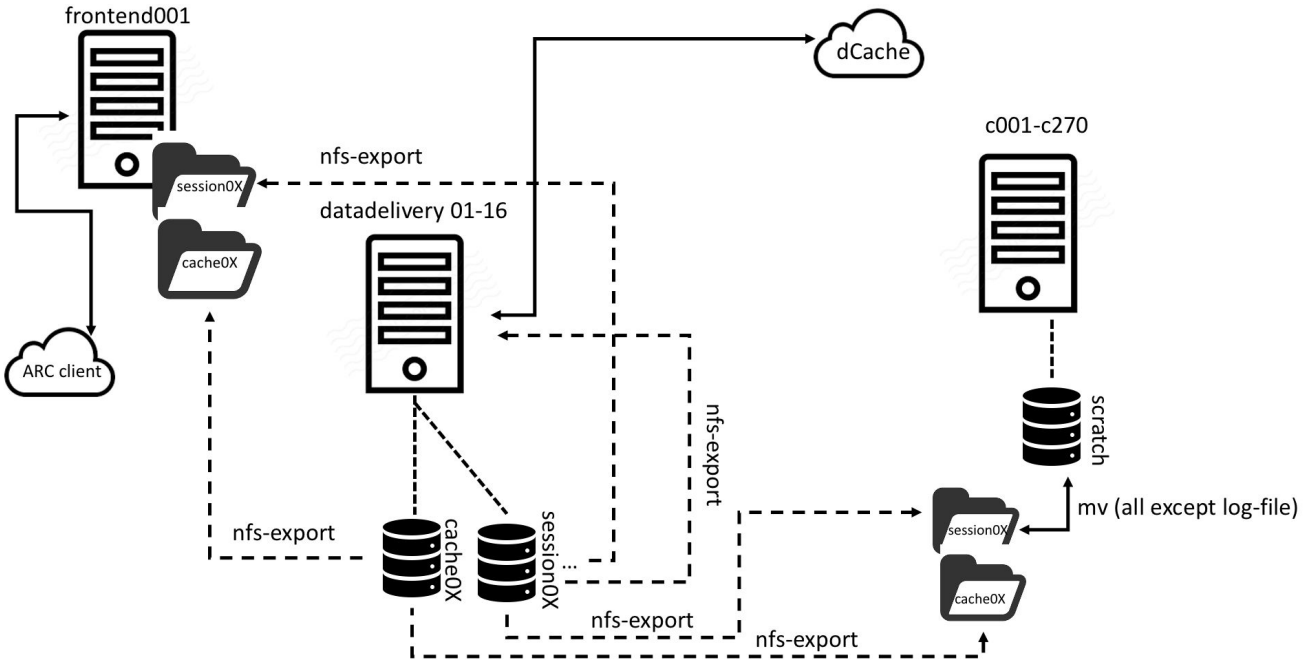
Current (and final) setup

270 compute nodes x 8 cores
AMD Rome Epyc
14.98 HEPSPEC/cpu
SLURM
25 GigE link between nodes

16 remote datadelivery servers
10TB cache on each server
100G session on each server

- sessiondir, nfs exported to compute, frontend, other datadelivery servers
- cachdir, nfs exported to frontend, compute

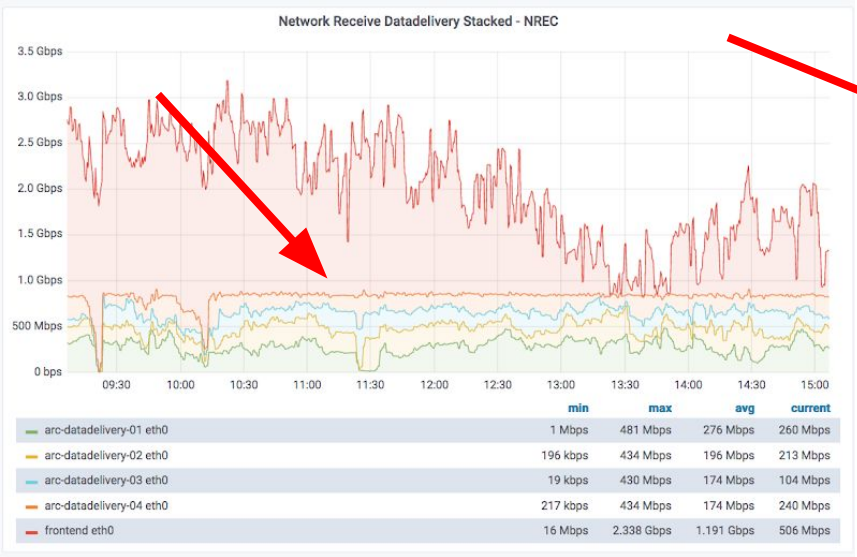
frontend does neither input datastaging nor output datastaging
- to offload ARC-CE as much as possible

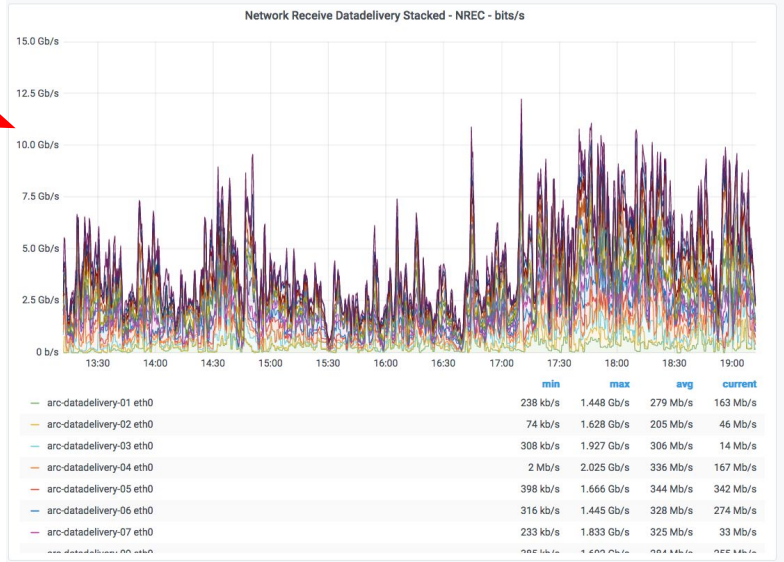


Configuration tools

- No longer use Elasticluster - too much overhead, and often things are not working as it should, waste time in support with Elasticluster - installs too much unnecessary stuff
- To create cluster: python-openstack home-grown script
- To configure: ansible
  - Extracted SLURM and other needed roles from Elasticluster ansible roles
  - Complemented with custom stuff, i.e. create needed directories, users, add users and accounts to SLURM etc etc etc.

This problem was finally solved May 2020





Problem was due to issues with uio-gw21 - constant high CPU-load, uptime of 5 years. Software update (new supervisor and updated software) in addition to (removal/adding) changes in the asynchronous routing.
The results were immediate. Immediate increase in throughput - from the cap of < 1 Gb/s to above 9 Gb/s. And the data-intensive derivations jobs immediately finished after downloading data for hours.

UIO_CLOUD now working more or less well

- Most often deliver HEPSPEC equivalent to nodes running 100% 24/7.
  - Some periods job-mix is very heavy causing idle compute-nodes - expected
  - Could again backfill with boinc - but have issues managing boinc to stay in the background
  - Am backfilling with lowpri simulation jobs - but these are not always sent out from PanDA when we need them

UIO_CLOUD - file system load

File-system is not performing optimally. SSD disks are on the way. That will probably fix the last remaining performance issues for the grid@Openstack infrastructure.

More info to this - Tor Lædre knows more details, but it is related to “slow ops” reported from the ceph-cluster.

Storage

Storage

- Transitioning away from ceph for storage.
  - Complexity was too high to troubleshoot and monitor
  - Erasure coding introduced high latencies for quorums on read
  - Suffered from other outstanding issues discussed later
- Implementing XFS on MD-RAID
  - More consistent in relation to existing practices within NDGF
  - Offers higher performance at a cost
    - Each node performs better individually as there are no network operations on reads and writes
    - When a node requires maintenance, a downtime must be scheduled.

Configuration - physical

- Currently we have a node operating as follows
  - 26 8TB SATA spinning disks
  - 2 SATA SSDs
  - Huawei storage server
    - 128GB RAM
    - LSI 3008i controller
    - LSI based SAS expander

Configuration - logical

- XFS
  - Data - MD-RAID 0 Stripe
    - MD-RAID 6 Stripe (13x8TB drives)
    - MD-RAID 6 Stripe (13x8TB drives)
  - XFS Log - MD-RAID 1 Mirror (2x800GB SSD)

Current issues

- SATA drives on SAS Expander decreases performance as writes increase.
- Average disk performance is approximately 120MB/sec per disk in write only operations from dCache
- Suspicions are that we're seeing excessive SAS expander resets
  - Cannot properly monitor as LSI does not deliver tools for ARM64 and their tools are closed source
- Writeback cache is limited to strictly what is available on disk
  - we can't configure write through cache support until we verify that it works properly
- Without LSI tools, we can't monitor, tune and optimize NCQ-2 performance.

Status

- Soft commitment to end testing this week or next
- Will reformat and rebuild all nodes at the end of testing
- dCache can be deployed very quickly on each node once it’s alive
- We will be limited only to dCache monitoring at first as UiO’s IT department rebuilds their management tools for ARM64 and CentOS.

Finally - wish list

- Will investigate in-service kernel upgrade as a possibility to minimize downtimes associated to kernel updates.