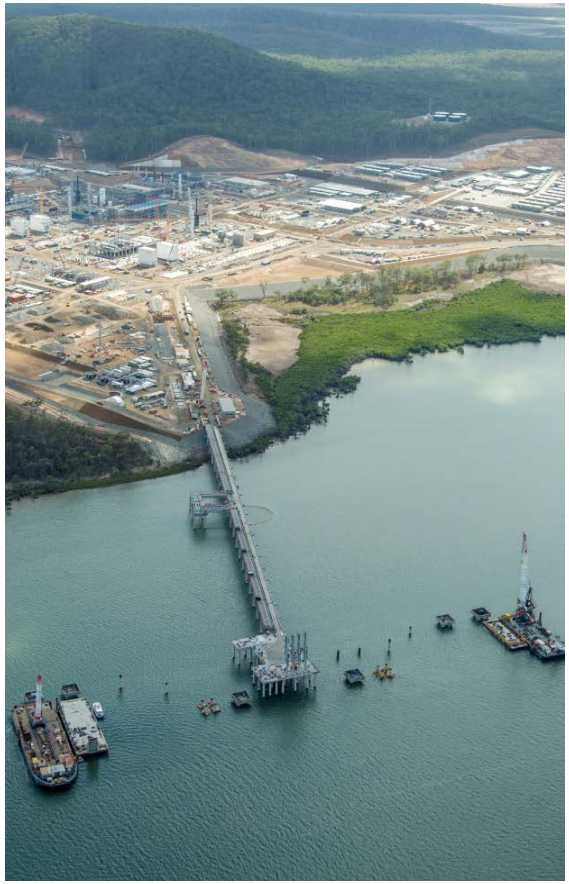
Mission-Critical Lustre at Santos

Adam Fox, Lustre User Group 2016

About Santos

One of the leading oil and gas producers in APAC

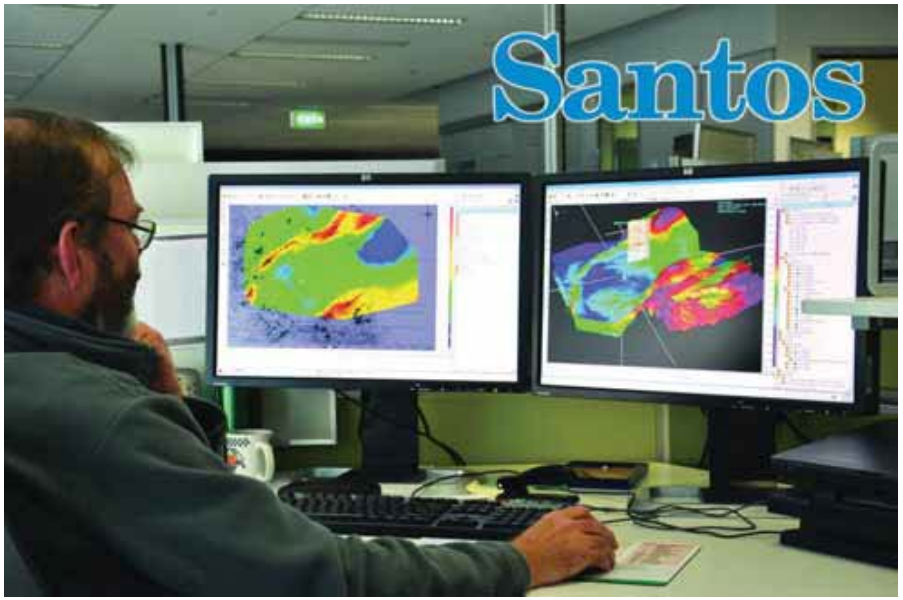


- Founded in 1954
  - South Australia Northern Territory Oil Search
- Cooper Basin
- Largest employer in South Australia
- Unix team – support Geoscience operations
  - Seismic data processing
  - Trying to find new gas and oil pockets
- Challenging low oil price environment

Why Lustre?

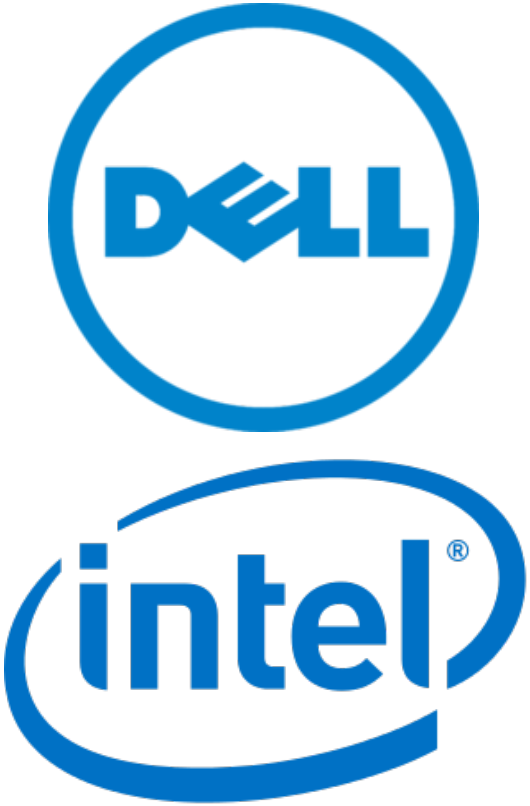
Initial Requirements for Lustre

- Geoscience storage nearing capacity and hardware end-of-life
- High overhead to maintain
  - SAN storage allocated in 16TB LUNs, shared via NFS
  - Over 9000 automount entries in LDAP to map storage structure to user-friendly filesystem layout
- NFS service availability
- SAN performance under high load
- Wanted a storage solution that could scale for performance and capacity, using commodity components
- Geoscientists connect to HPC via TurboVNC for full 3D interactivity – Red Hat Innovation winner 2011



intel DELL Santos We have the energy.

Why Lustre?



Consulting Process

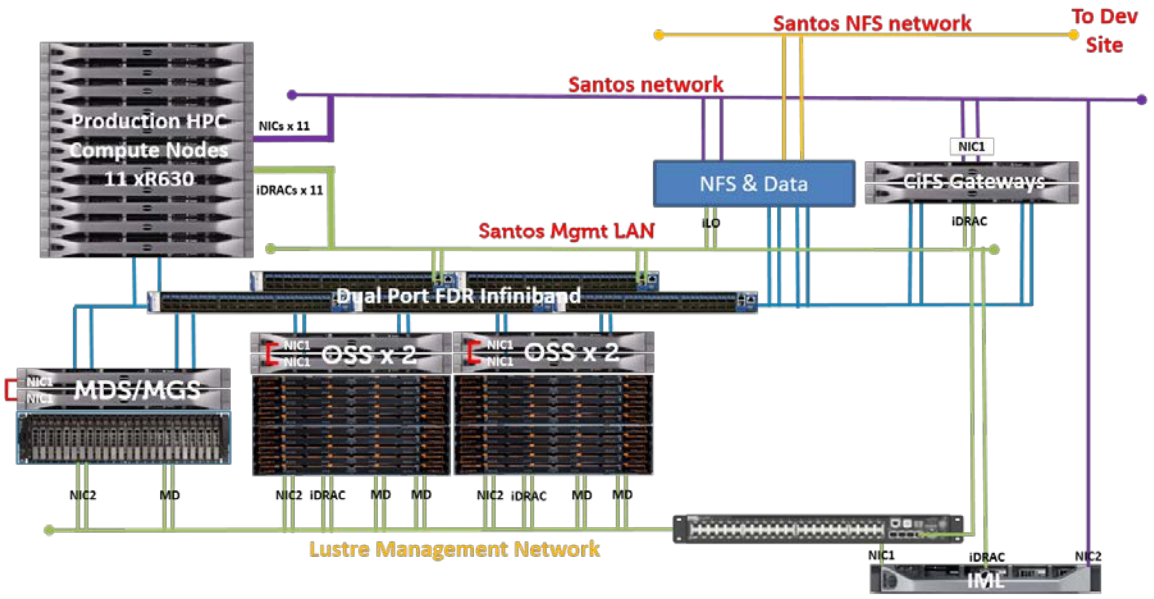
- Thorough tender process for choosing a vendor and partner to implement Lustre
- Initial Consulting Contract – As-Is, To-Be to review environment
- Partnership of Datacom / Dell / Intel was successful
  - **Pilot:** 700TB Lustre filesystem
  - **Production:** Grow to 1.3PB
  - **DR:** 530TB Lustre filesystem for replicated production data
  - Intel® Enterprise Edition for Lustre* (IEEL) 2.2.0.0
  - Intel® Manager for Lustre (IML)
  - Mellanox InfiniBand switches and network cards
  - Clustered Samba servers for CIFS gateway
  - Utilise existing NFS servers

Santos
We have the energy.

Implementation

Pilot

- MDS Server
  - 12x 800GB SSD, RAID 10
- 2x OSS Server pairs
  - 24x OSTs
  - 10x 4TB SAS HDD, RAID 6
- 700TB total usable space
- FDR 56Gb InfiniBand
  - Fat tree, non blocking
- Get some scratch space available for Geoscientists quickly
- Get a feel for Lustre in Santos environment



Santos
We have the energy.

Implementation

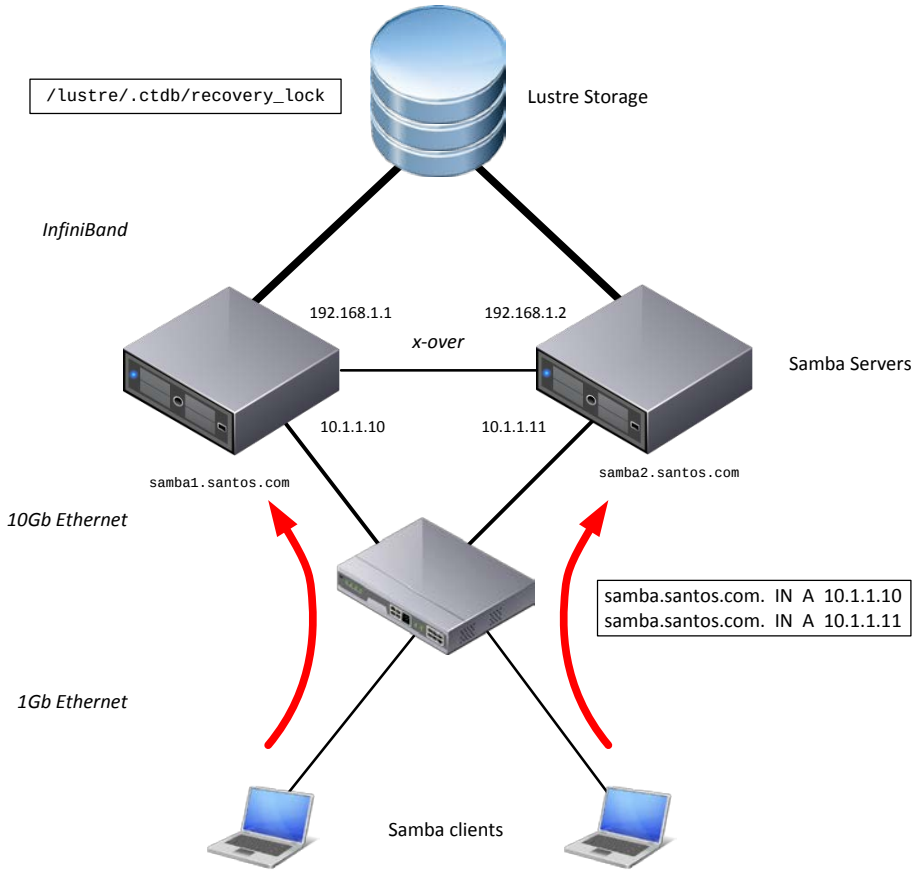
Pilot

- Implementation of pilot Lustre storage went really smooth
- User acceptance testing
  - Performance
  - Availability
- Plan was to re-use existing NFS servers as Lustre clients
  - InfiniBand cards installed, Lustre client software installed
  - Lustre was slower on existing NFS servers
    - Older CPU generation
  - Purchased 2x Dell PowerEdge R630's for NFS and use CTDB for clustering NFS
- NFS stale filehandles
  - Worked with Intel Support to resolve – upgrade to IEEL 2.4

Implementation

Samba & NFS CTDB

- Dell / Intel reference architecture
- Previous bad experiences with Linux HA solutions
  - Heartbeat, pacemaker
- CTDB handles failover of IP between nodes
  - Round-robin DNS
- So far so good, no clustering fails
- Samba 4 vs Samba 3
  - No CTDB support in Samba 4 on RHEL 6
- CTDB also used for NFS serviing



Santos
We have the energy.

Implementation

Puppet

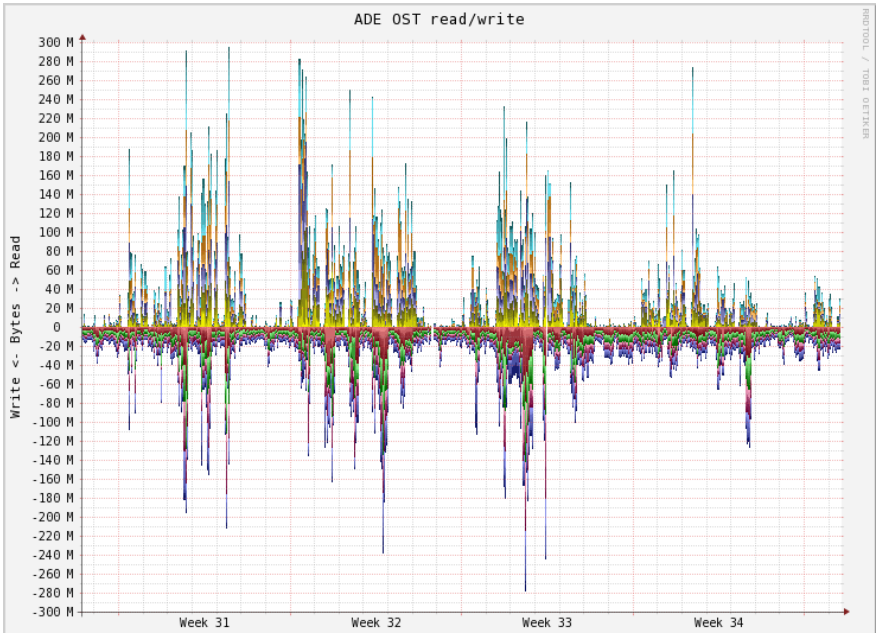
```
class lustre::client (
  $lnet_networks = "o2ib0(ibbond)",
  $infiniband = true,
  $mount_fs = true,
  $mount_device,
  $mount_options,
) {

  if $infiniband {
    include lustre::ofed
  }

  package { 'lustre-client':  ensure => installed, }

  file { '/etc/modprobe.d/lustre.conf':
    ensure  => present,
    owner   => 'root',
    group   => 'root',
    mode    => '0644',
    content => "options lnet networks=\"${lnet_networks}\"\n",
  }

  File['/etc/modprobe.d/lustre.conf'] -> Package['lustre-client']
```

- Puppet for configuration management
- IML still does Lustre server configuration
- Mainly used for Lustre client configuration
  - LNET module options
  - Enable InfiniBand
  - Mount options

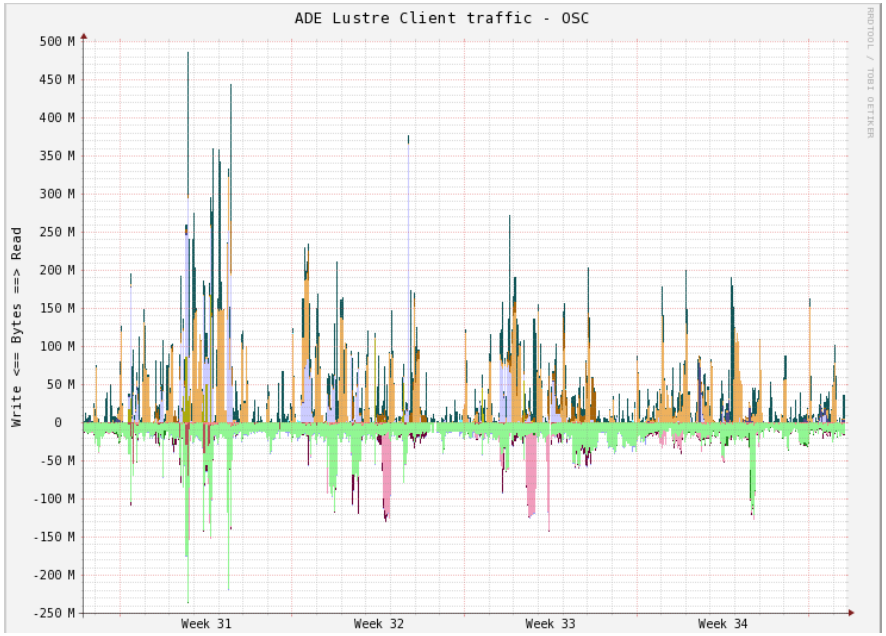
Santos
We have the energy.

Implementation

Collectd Monitoring and Graphing



Stacked graph of per-OST read/write traffic

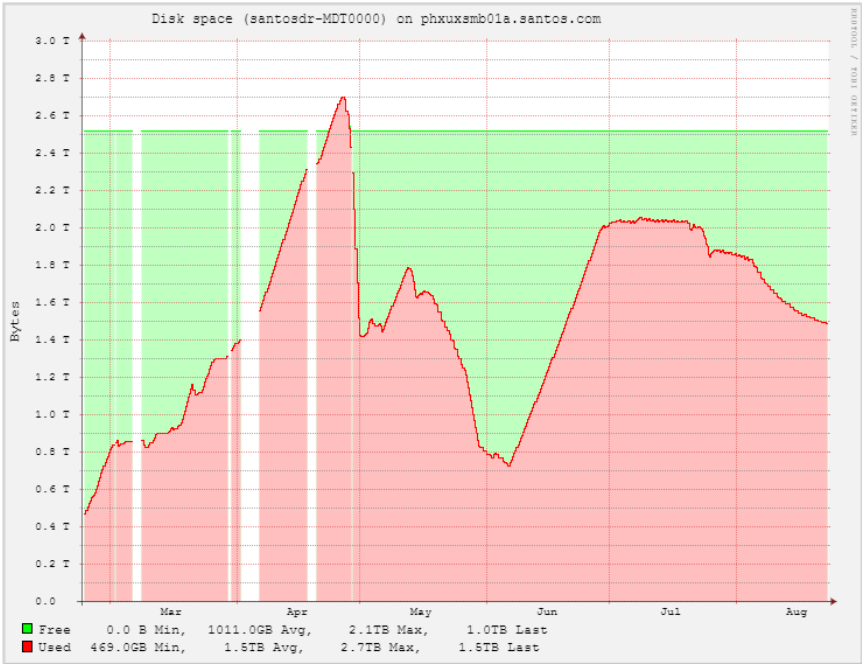


Stacked graph of per Lustre client OSC read/write

- Handy for troubleshooting what Lustre client is causing most IO

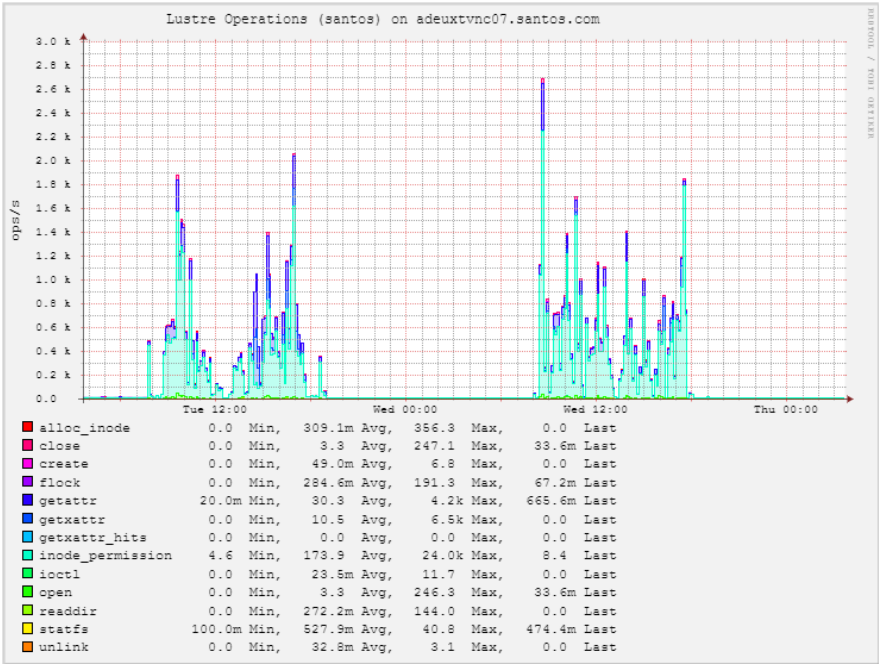
Implementation

Collectd Monitoring and Graphing



Disk usage per OST/MDT device

- MDT filled up – went into reserved space



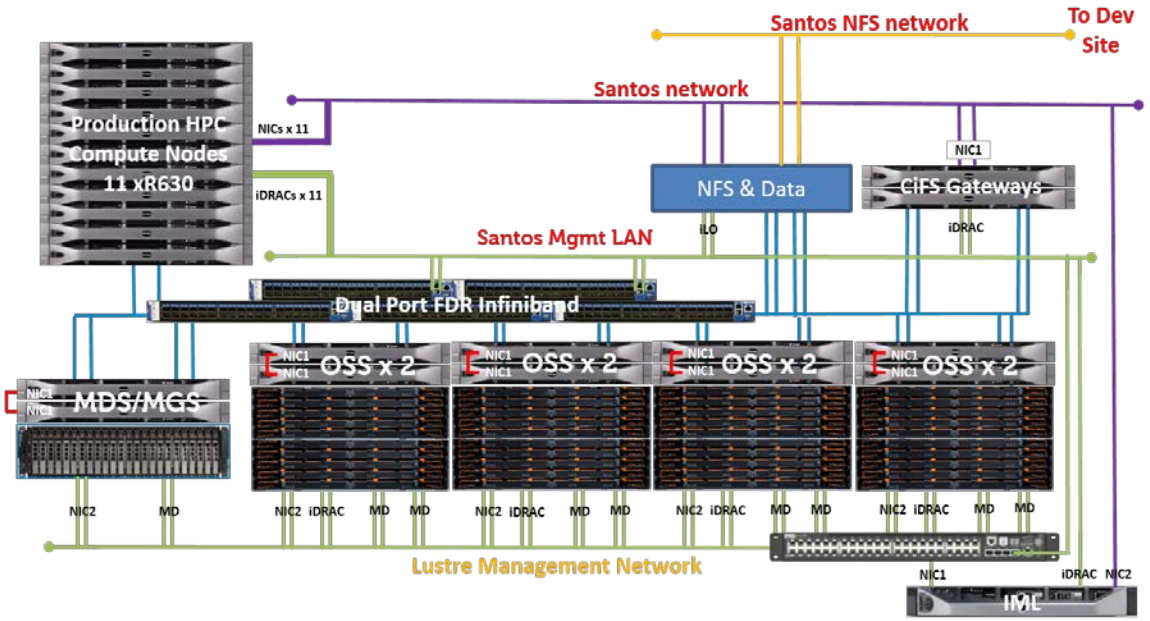
Lustre client metadata operations

- Counters from /proc/fs/lustre/llite/stats

Implementation

Production

- Install additional 2x OSS pairs
  - 24 OSTs → 48 OSTs
  - 700TB → 1.3PB
- IML made for easy expansion
- Configured Lustre servers on Ethernet network
- 40GB/s read
- 24GB/s write

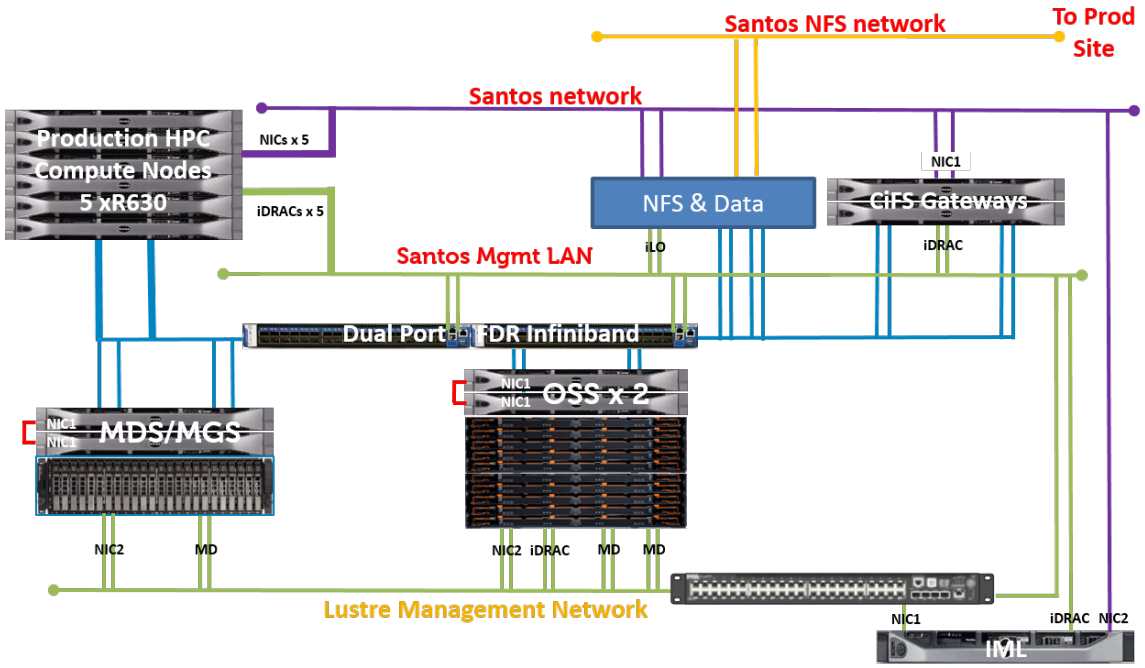


Santos
We have the energy.

Implementation

Disaster Recovery

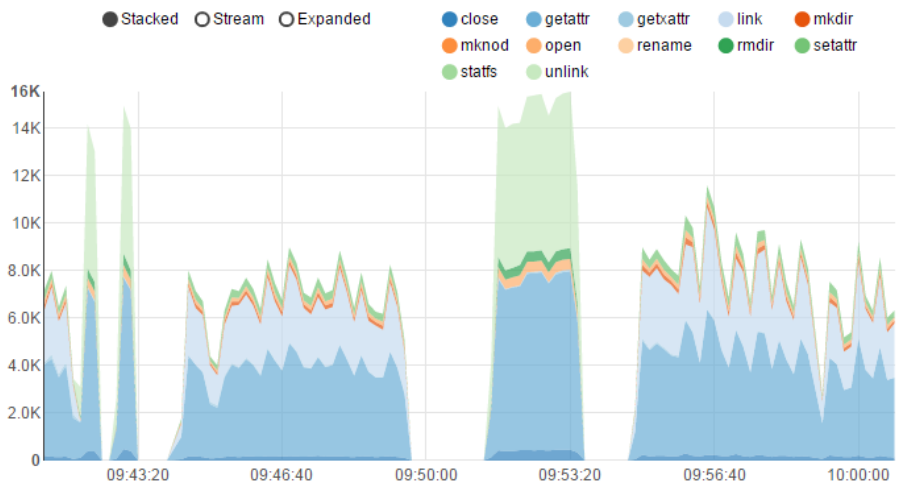
- Separate Lustre installation
- 1x OSS server pair
  - 12x OSTs
  - 10x 6TB SAS HDD RAID 6
- 520TB total usable space
  - Only production data
  - Replicated from prod site
- DR Samba and NFS servers



Santos
We have the energy.

Replication

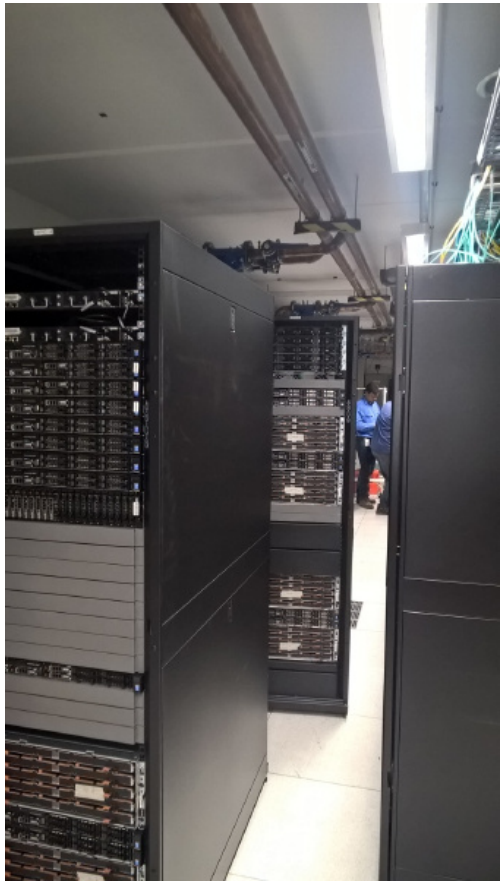
- Initial plan was to use rsnapshot / rsync for replication between sites
  - Hard link files that do not change
- Lots of unstructured data
  - Most data does not change, but many files
  - 30 million files, 380TB of data
- Rsync not point-in-time consistent
- Rsnapshot too slow to replicate
- fpart – “Sort files and pack them into partitions”
  - https://github.com/martymac/fpart
- fpsync – wrapper script for fpart and rsync
- Use fpart to walk filesystem tree, feed partitions into a set number of rsync processes
- Hitting limit of metadata updates on MDS
- Settled on 6 concurrent rsync processes
- Perl script drive fpart and rsync processes
  - Make it rsnapshot-like – snapshot retention



Data Centre Switch

- All going to plan, and then...
- Swap the data centres around!
- Server room cooling failure
- Secondary site more reliable
- Lustre first to move
  - Move before data migration
  - Risk mitigation
- Use COPE Sensitive Freight to move whole racks

- All went really smooth
- Added delay to final cutover



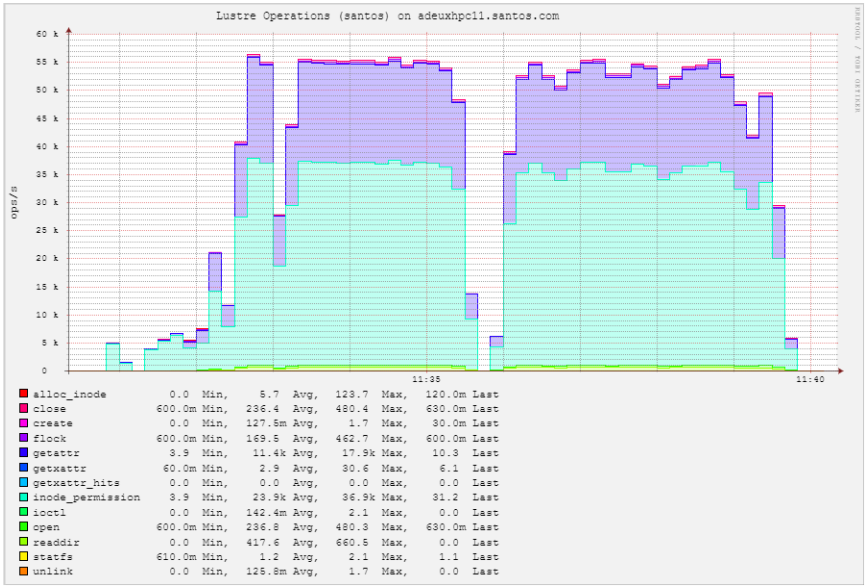
Production Cutover

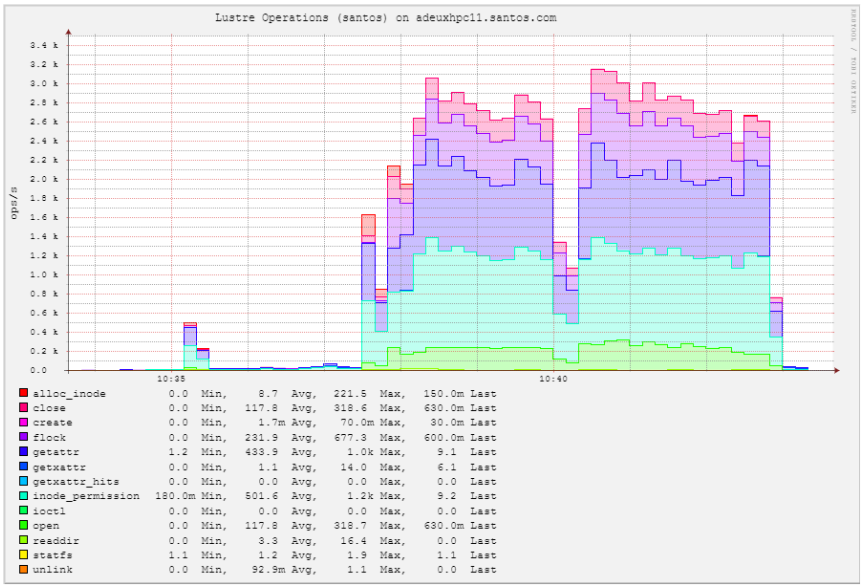
- Finally get to do final migration and cutover to Lustre
- Cutover process fairly straightforward:
  - Stop all NFS clients
  - Perform final rsync
  - Update automounts in LDAP
  - Restart autofs
- Wait for Monday morning rush!
- No major issues
  - 32-bit applications running over NFS did not like 64-bit inodes
  - Resolved by moving to direct Lustre client – handles 32-bit system calls

Production Cutover

Outstanding Issues

- Application performance with large amount of small IO or metadata operations
- E.g.: 6.6 million stat system calls (4002 files stat'ed 1661 times)
- Using NFS in front of Lustre seems to mask the effect, not ideal though





Santos
We have the energy.

Future Improvements

- Upgrade to IEEL 3
  - Parallel metadata updates
- Set LNET peer credits
  - cat /proc/sys/lnet/peers – large negative numbers
  - Need Lustre outage to set credits and peer_credits on Lustre clients
- Implement Robinhood
  - Manage unstructured data
  - Could feed into replication process – don't need to walk whole filesystem each time