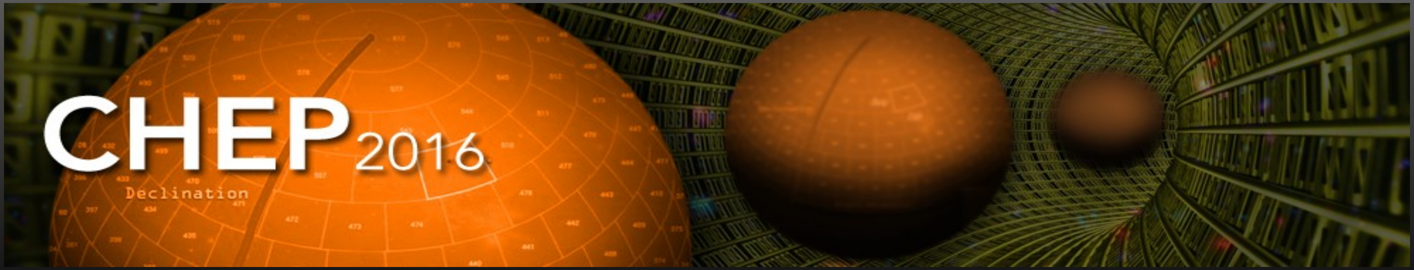
CoEPP
ARC Centre of Excellence for
Particle Physics at the Terascale

RC
CoEPP

CephFS: a new generation storage platform for Australian High Energy Physics

Goncalo Borges, Sean Crosby, Lucien Boland



CoEPP & Research Computing

- **ARC Centre of Excellence for Particle Physics at the Terascale**
  - Leads Australia in the field of HEP research
  - Joins experimental and theoretical researchers from different universities
    - Adelaide, Melbourne and Monash, Sydney
  - Provides the support for the Australian participation in major international collaborations: ATLAS at LHC / CERN and BELLE II at SuperKEKB / Japan

- **RC team is an enabler for CoEPP's scientific discovery and research.**
  - Responsible for meeting the large-scale computing requirements needed by CoEPP's international collaborations and local researchers:
    - **Operation of a Tier-2 for ATLAS**

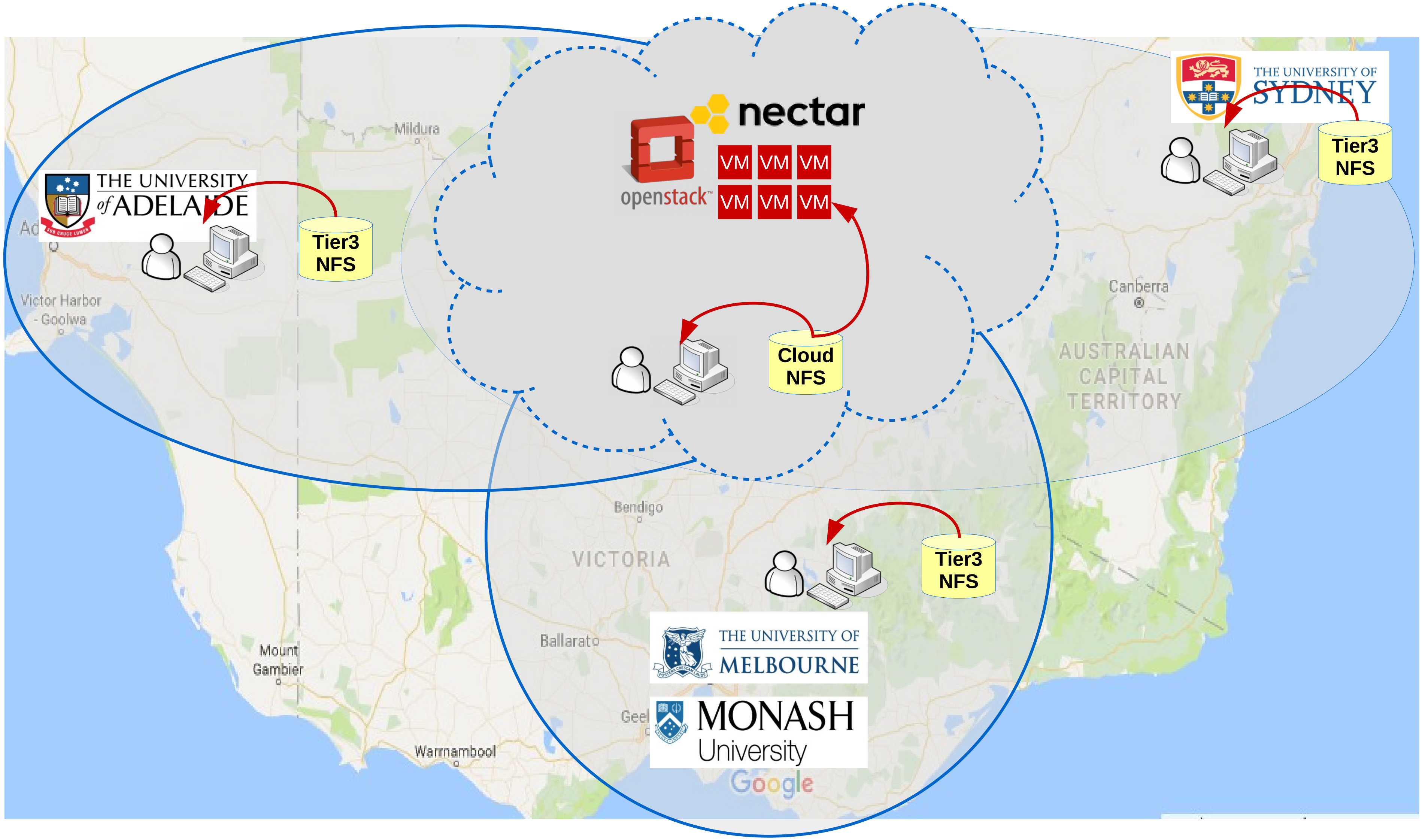
      - Most available, top 3 reliable ATLAS site since Oct / 2014

      - 11500 HS06 pledge CPUs (PBS/Torque, MAUI) + 1.2 PB of data (DPM)
    - **Support all aspects of local scientific computing activities**

      - Local NFS shares at each CoEPP pool

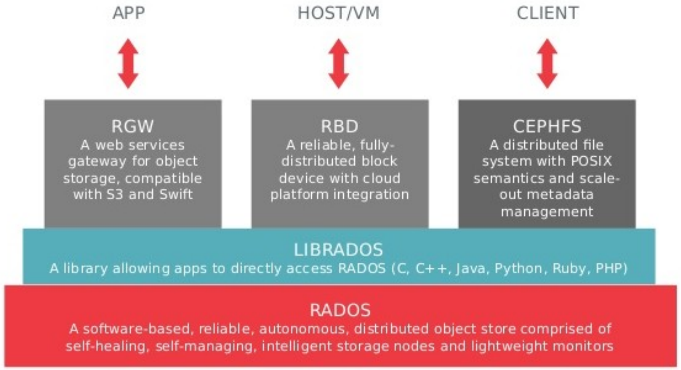
      - Elastic central computing service, capable to scale up / down resources, in Nectar OpenStack Cloud (→ 700 cores): 2014 J. Phys.: Conf. Ser. 513 032107

'Tier-3' services



Why Ceph / CephFS ?

- **Why Ceph?**
  - Open source / Community driven
    - HEP involvement already on-going
  - Designed to work with commodity hardware
    - Self recovering / healing behaviour
    - No single point of failure
  - High granularity of data operations customizations
  - Different data access methods
  - Integration with other infrastructures (libvirt, openstack, ….)


APP
HOST/VM
CLIENT
RGW
A web services gateway for object storage, compatible with S3 and Swift
RBD
A reliable, fully-distributed block device with cloud platform integration
CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)
RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

- **Why CephFS ?**
  - The POSIX-like requirement
  - The capability to make the filesystem available in different geographical locations
  - The metadata in RADOS

- **Jewel 10.2.X (X = 0,1,2,3)**
  - CephFS **stable** release (note that stable ≠ production)

    http://docs.ceph.com/docs/master/cephfs/best-practices/

CephFS ('Pre-Production' tests)

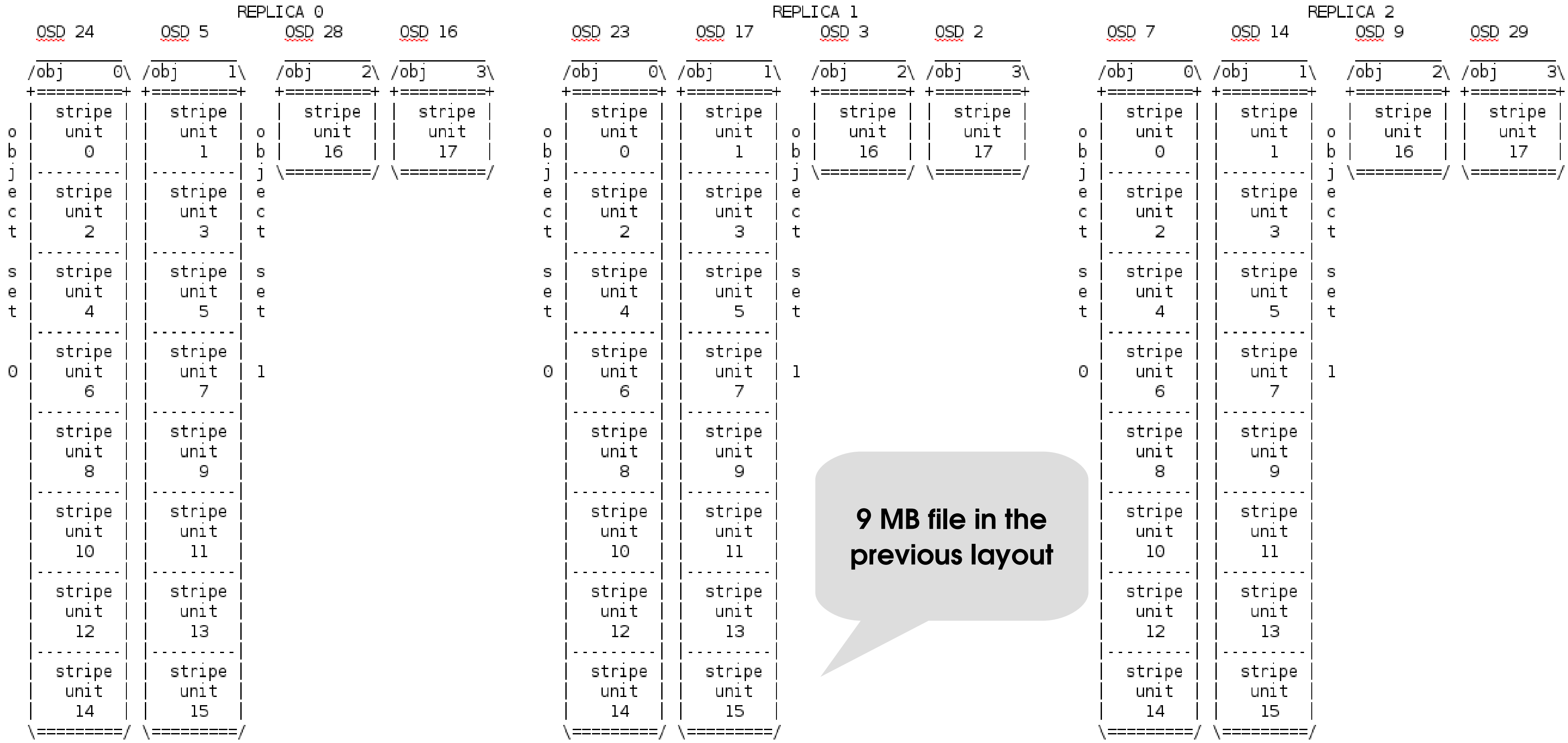
- **'Pre-production setup'**
  - 4 Dell R620 x 8 (3TB) OSDs (9.2.0)
  - Jornals in a separate OSD partition
  - A single host mount cephfs (kernel)
  - Single MDS server (32 GB RAM)

- **Single FIO bechmark**
  - Sequential write and read, random write and read.
  - Files of 8 GB; ioengine=libaio; iodepth=64; direct=1.
  - Client cache purged before each test
  - More interested in understand CephFS layouts and File stripping

- **CephFS layouts and file stripping**
  - Client writes the stripe units to their corresponding objects in parallel
  - Since objects get mapped to different placement groups and further mapped to different OSDs, each write occurs in parallel at the maximum write speed
  - Layouts are defined as extended attributes at a directory level

CephFS ('Pre-Production' tests)

```
# getfattr -n ceph.dir.layout /cephfs/objectsize4M_stripeunit512K_stripecount2
ceph.dir.layout="stripe_unit=524288 stripe_count=2 object_size=4194304 pool=cephfs_dt"
```

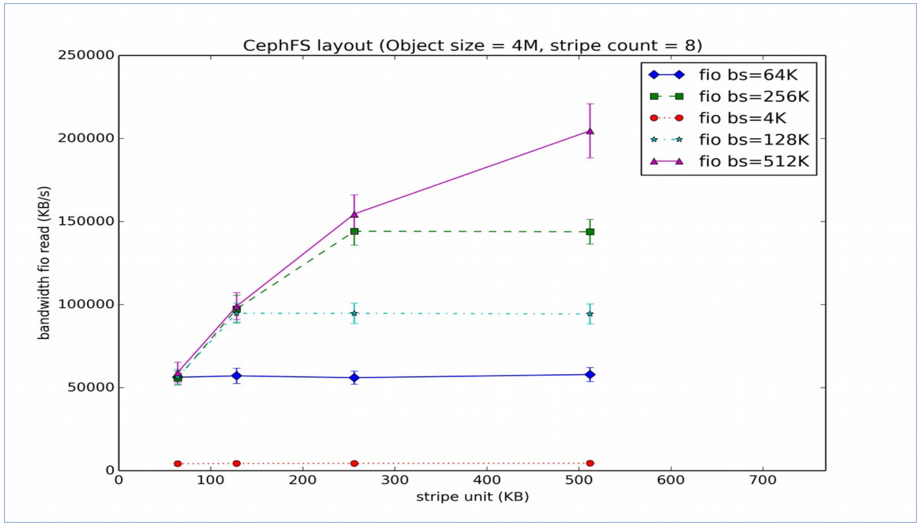


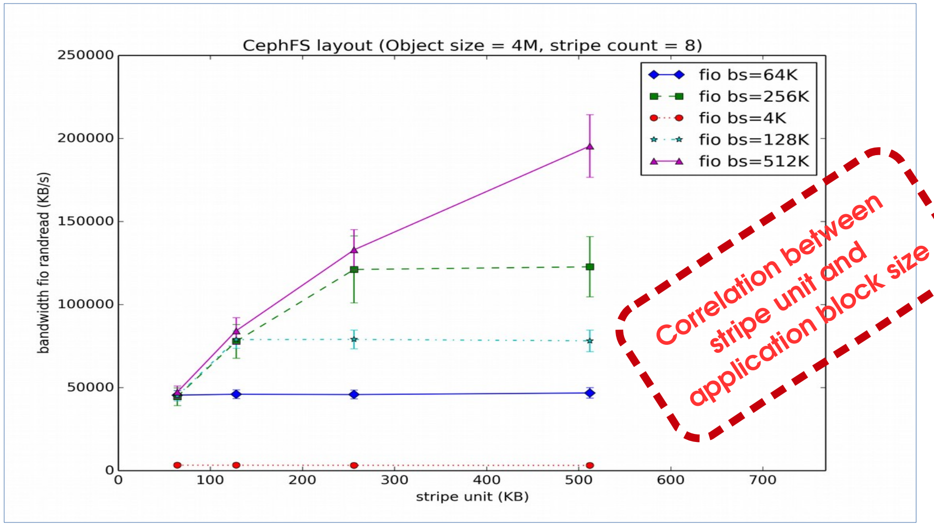
```
                       REPLICA 0                                             REPLICA 1                                             REPLICA 2
  OSD 24      OSD 5         OSD 28      OSD 16          OSD 23      OSD 17        OSD 3       OSD 2           OSD 7       OSD 14        OSD 9       OSD 29
   _________   _________     _________   _________       _________   _________     _________   _________       _________   _________     _________   _________
  /obj     0\ /obj     1\   /obj     2\ /obj     3\     /obj     0\ /obj     1\   /obj     2\ /obj     3\     /obj     0\ /obj     1\   /obj     2\ /obj     3\
  +=========+ +=========+   +=========+ +=========+     +=========+ +=========+   +=========+ +=========+     +=========+ +=========+   +=========+ +=========+
  | stripe  | | stripe  |   | stripe  | | stripe  |     | stripe  | | stripe  |   | stripe  | | stripe  |     | stripe  | | stripe  |   | stripe  | | stripe  |
o |  unit   | |  unit   | o |  unit   | |  unit   |   o |  unit   | |  unit   | o |  unit   | |  unit   |   o |  unit   | |  unit   | o |  unit   | |  unit   |
b |    0    | |    1    | b |   16    | |   17    |   b |    0    | |    1    | b |   16    | |   17    |   b |    0    | |    1    | b |   16    | |   17    |
j |---------| |---------| j \=========/ \=========/   j |---------| |---------| j \=========/ \=========/   j |---------| |---------| j \=========/ \=========/
e | stripe  | | stripe  | e                           e | stripe  | | stripe  | e                           e | stripe  | | stripe  | e
c |  unit   | |  unit   | c                           c |  unit   | |  unit   | c                           c |  unit   | |  unit   | c
t |    2    | |    3    | t                           t |    2    | |    3    | t                           t |    2    | |    3    | t
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
s | stripe  | | stripe  | s                           s | stripe  | | stripe  | s                           s | stripe  | | stripe  | s
e |  unit   | |  unit   | e                           e |  unit   | |  unit   | e                           e |  unit   | |  unit   | e
t |    4    | |    5    | t                           t |    4    | |    5    | t                           t |    4    | |    5    | t
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
  | stripe  | | stripe  |                               | stripe  | | stripe  |                               | stripe  | | stripe  |
0 |  unit   | |  unit   | 1                           0 |  unit   | |  unit   | 1                           0 |  unit   | |  unit   | 1
  |    6    | |    7    |                               |    6    | |    7    |                               |    6    | |    7    |
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
  | stripe  | | stripe  |                               | stripe  | | stripe  |                               | stripe  | | stripe  |
  |  unit   | |  unit   |                               |  unit   | |  unit   |                               |  unit   | |  unit   |
  |    8    | |    9    |                               |    8    | |    9    |                               |    8    | |    9    |
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
  | stripe  | | stripe  |                               | stripe  | | stripe  |                               | stripe  | | stripe  |
  |  unit   | |  unit   |                               |  unit   | |  unit   |                               |  unit   | |  unit   |
  |   10    | |   11    |                               |   10    | |   11    |                               |   10    | |   11    |
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
  | stripe  | | stripe  |                               | stripe  | | stripe  |                               | stripe  | | stripe  |
  |  unit   | |  unit   |                               |  unit   | |  unit   |                               |  unit   | |  unit   |
  |   12    | |   13    |                               |   12    | |   13    |                               |   12    | |   13    |
  |---------| |---------|                               |---------| |---------|                               |---------| |---------|
  | stripe  | | stripe  |                               | stripe  | | stripe  |                               | stripe  | | stripe  |
  |  unit   | |  unit   |                               |  unit   | |  unit   |                               |  unit   | |  unit   |
  |   14    | |   15    |                               |   14    | |   15    |                               |   14    | |   15    |
  \=========/ \=========/                               \=========/ \=========/                               \=========/ \=========/
```

9 MB file in the previous layout

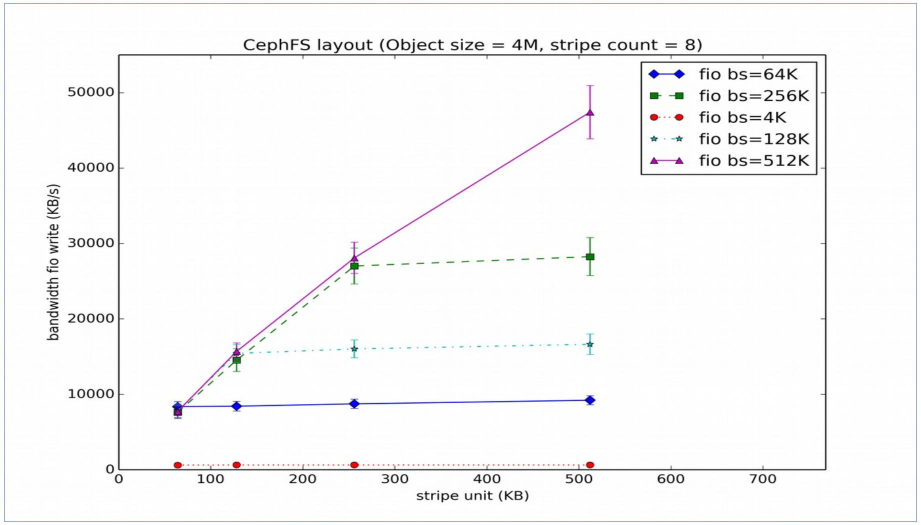
CephFS ('Pre-Production' tests)

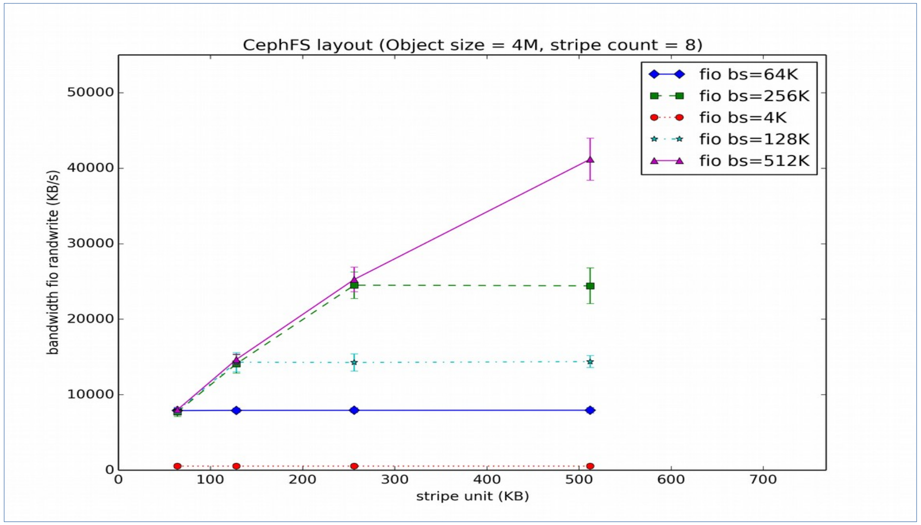
- Single FIO bechmark (stripe unit effect)





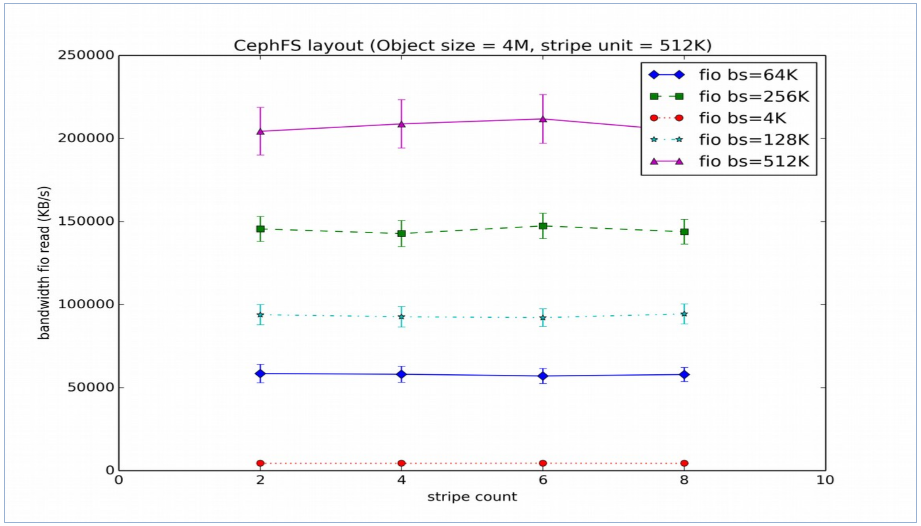
Correlation between stripe unit and application block size

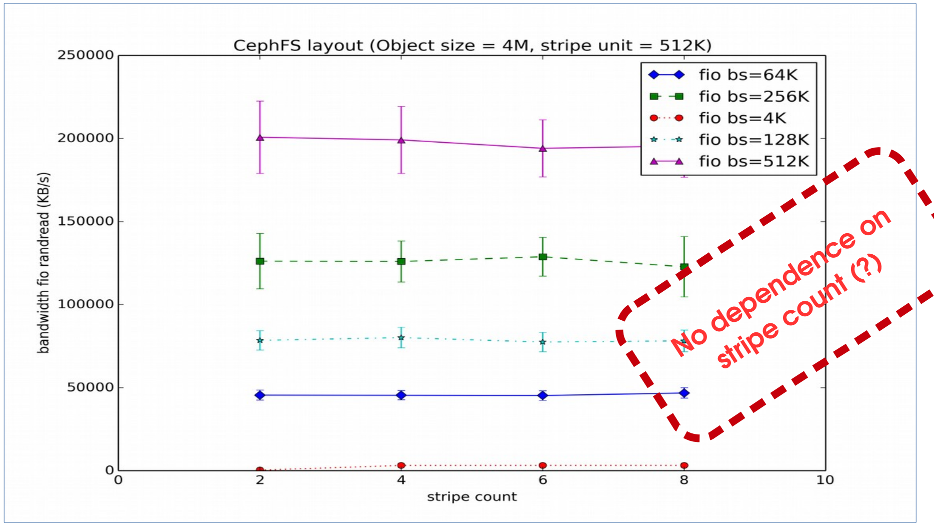




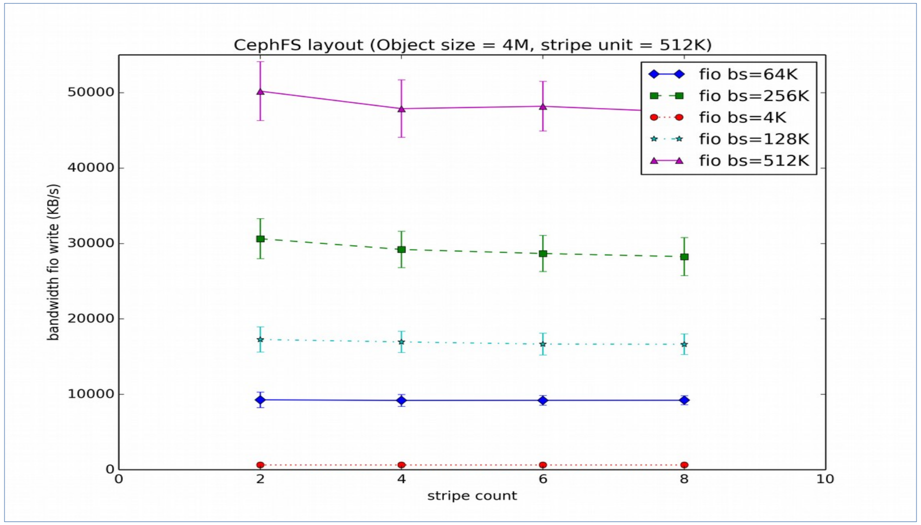
CephFS ('Pre-Production' tests)

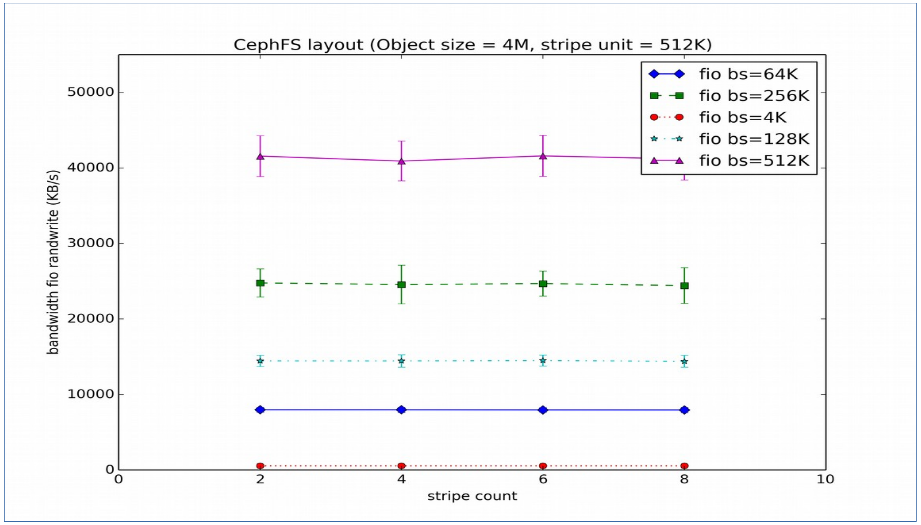
- Single FIO bechmark (stripe count effect)





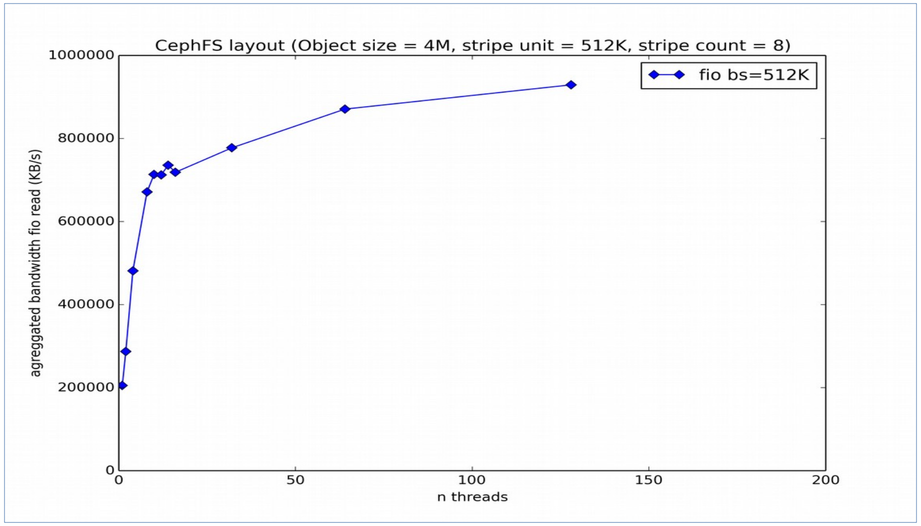
No dependence on stripe count (?)

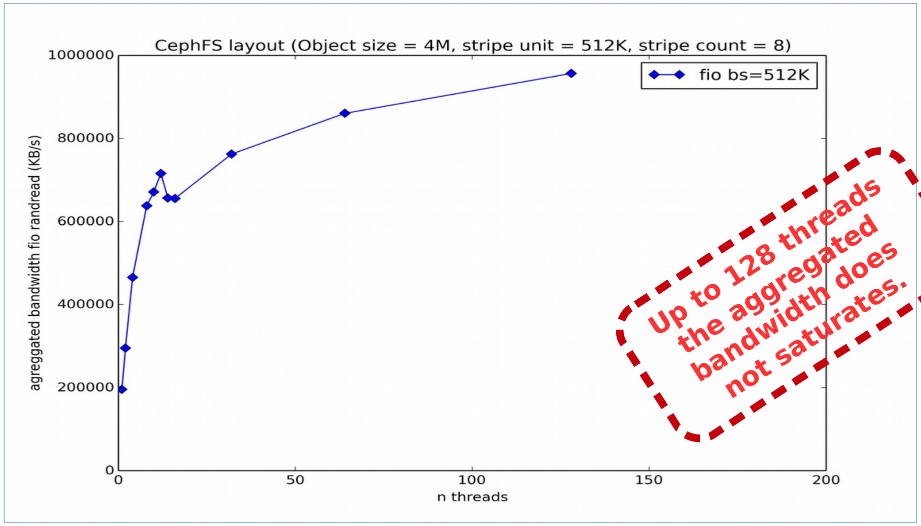




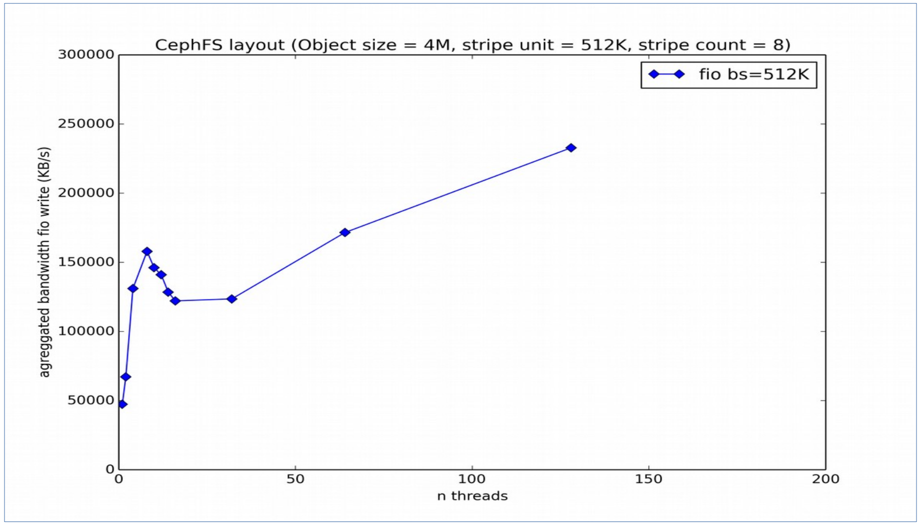
CephFS ('Pre-Production' tests)

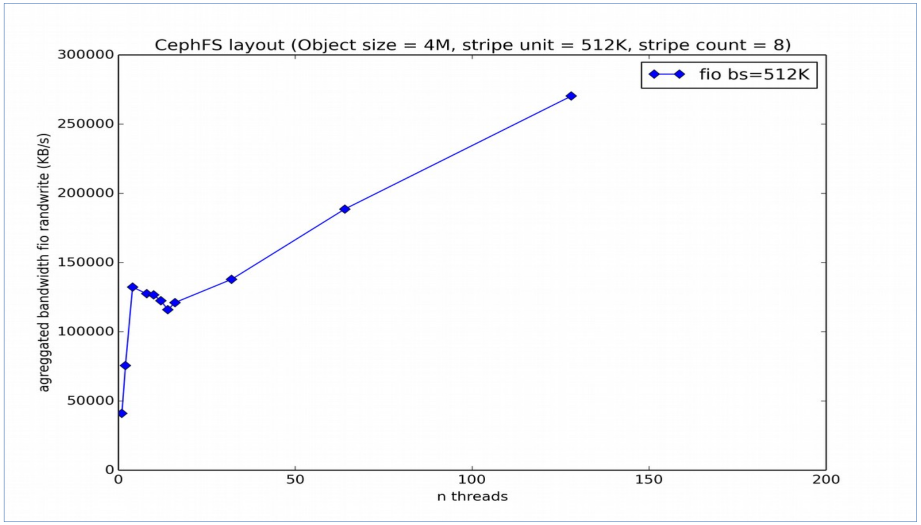
- **Multithread FIO bechmark**





Up to 128 threads the aggregated bandwidth does not saturates.

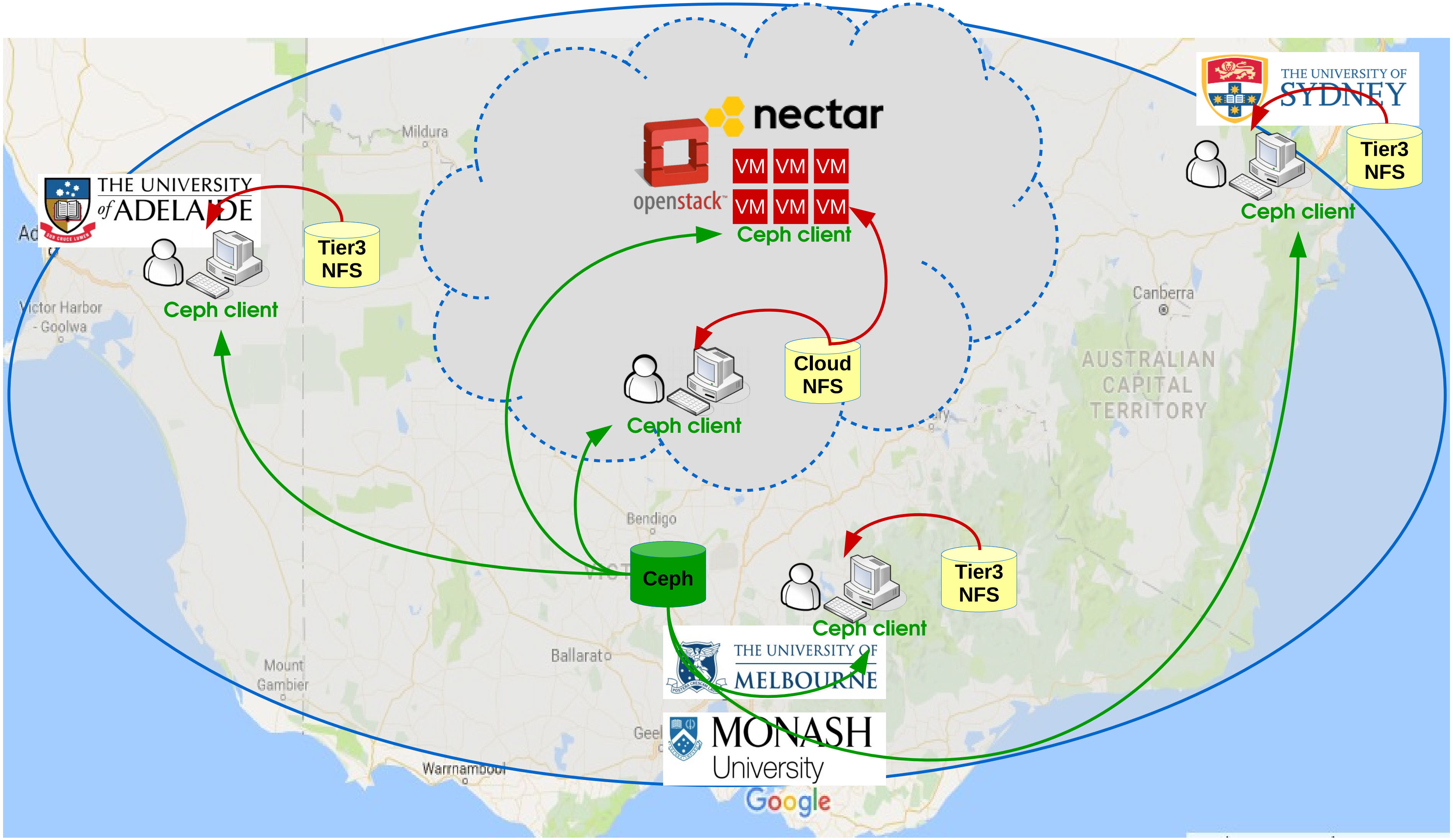




Production Setup

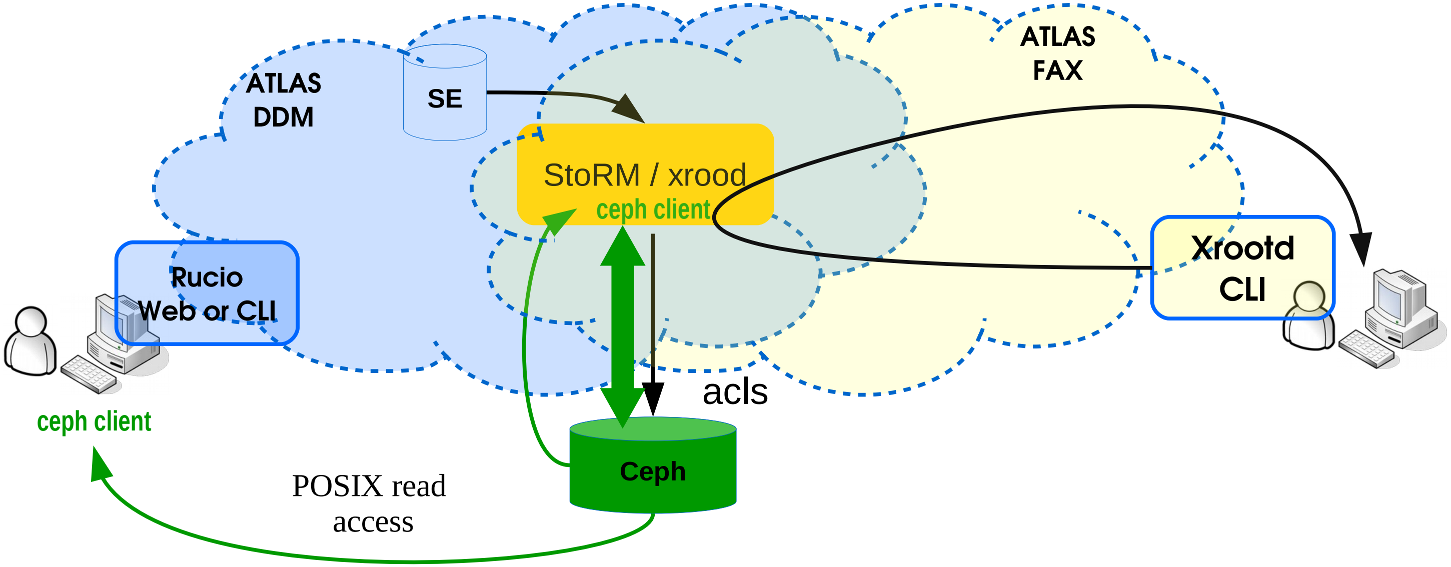
- **Ceph Object Storage Cluster**
  - 3 monitors (physical hardware)
  - 11 storage servers, 112 OSD, 305 TB of raw storage
    - Centos7, 4 GB of RAM / OSD
    - 7 Dell PowerEdge R620 storage servers
      - PERC H710 Mini internal controller, PERC H810 external controllers
      - 4 storage servers x 8 OSDs (3 TB/each) + 3 storage servers x 16 OSD (3 TB /each)
    - 4 Dell PowerEdge R710
      - PERC 6/i internal controller, IBM Server RAID M5025 external controller
      - 8 OSD (3 TB / each) per storage server
    - Internal network with MTU 9000, txqueuelen + TX/RC buffer tunning
    - Intel DC S3550 SSDs (120 GB) for OSDs journals (4 OSDS : 1 SSD)
  - CephFS
    - Dedicated pools for CephFS data and metadata; size=3, min_size=2 (3 replicas)
    - Active MDS → Dell PowerEdge R520, 32 GB RAM + 8 GB SWAP
    - Standby-Replay MDS → VM 8 GB RAM + 8 GB SWAP
    - > 200 ceph-fuse clients

Enhanced 'Tier-3' services



Tier-2 integration (StoRM / xrootd)

- **StoRM**
  - Uses POSIX filesystem as data backed
  - Allows to set group ACLs
- **Xrootd (using its posix driver)**
- **ATLAS DDM and FAX integration**
  - A secondary ATLAS LOCALGROUP DISK served by cephfs (using the kernel client)
  - Read access for local atlas users



Summary / Conclusion

- **CephFS was 'kind of' forced' on us …**
  - Due to the geographical nature of our ARC centre
  - Due to the collaborative nature of HEP research / researchers
- **CephFS is in 'Production' for more than an year**
  - Started with Infernalis and now in Jewel
  - Researchers are heavily using it. Apart from some minor issues (from a user's perspective), researchers are happy
- **Managing CephFS...**
  - Deploying and installing is quick and easy; continuous operation is difficult. Problems are, most of time, only visible 'a posteriori.
  - Software (both Ceph and CephFS) a bit buggy
  - Some 'hairy' issues we already detected from a 'site admin' perspective
    - No real showstoppers but with some complexity involved
- **The disclaimer to our users is: “Just put data there you can regenerate or retrieve from somewhere else”**

Fall HEPIX 2016 ...

- **All the technical details (issues and tuning):**

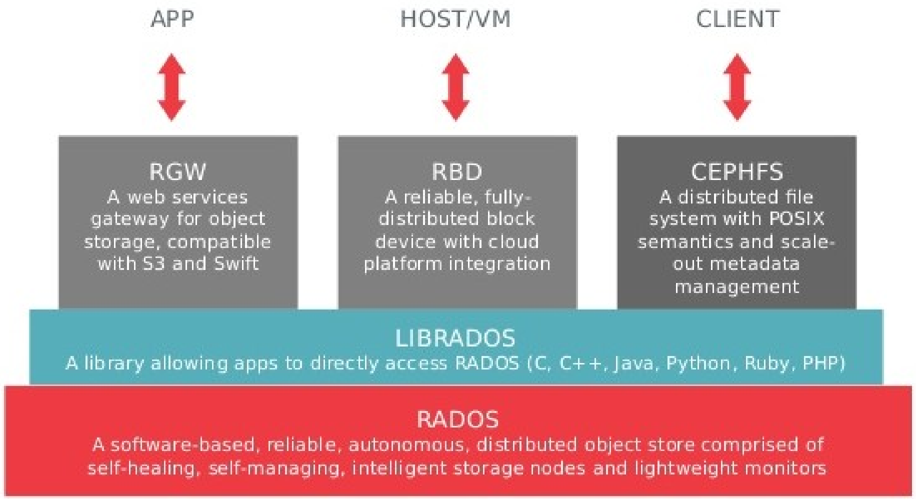
  https://indico.cern.ch/event/531810/contributions/2309925/

  - Compilation issues / restrictions in SL6
  - Ceph-fuse performance and configuration
    - client tuning
    - performance over wan
    - understanding memory usage, threads, …
  - Cephfs recovery tools
  - MDS bugs and issues

- **BACKUPS**

Ceph (a nutshell explanation)

- **Ceph is a cutting edge, open source, distributed data storage technology**
  - Based on intelligent object storage devices (OSD), a combination of CPU, network interface, local cache and underlying disk space.
  - Clients can use multiple access methods:



APP

HOST/VM

CLIENT

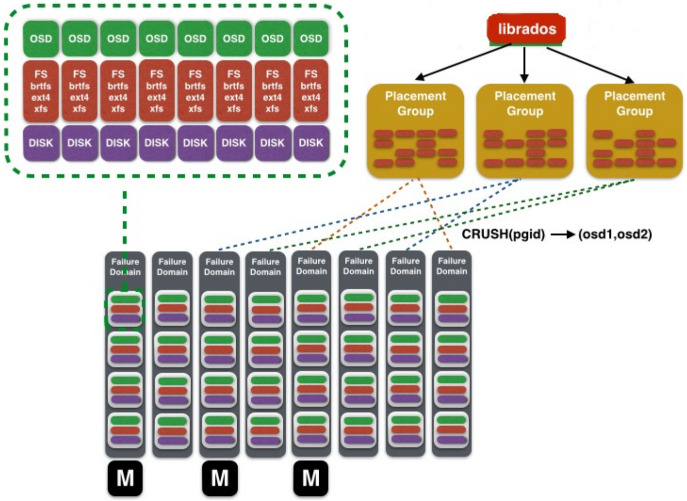
RGW
A web services gateway for object storage, compatible with S3 and Swift

RBD
A reliable, fully-distributed block device with cloud platform integration

CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management

LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)

RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors



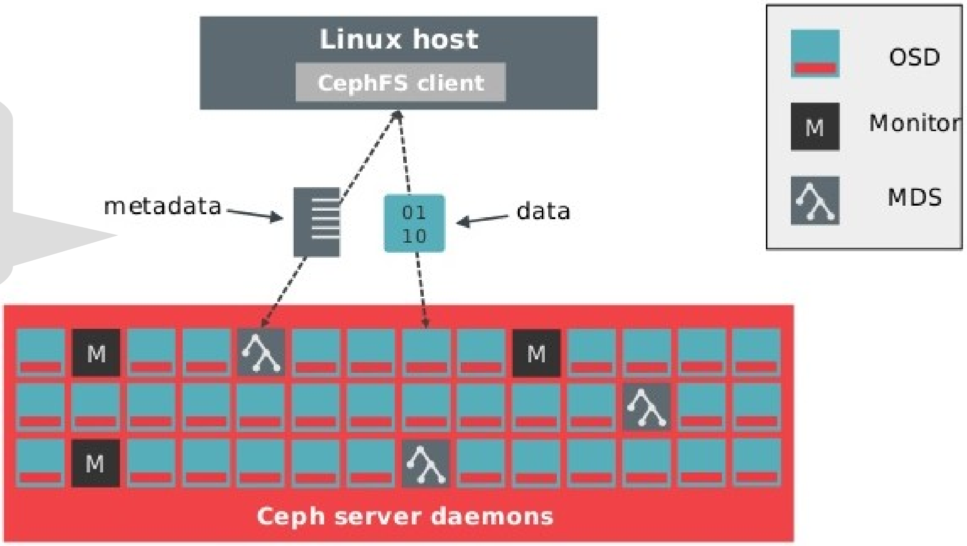
  - Data is stripped by writing byte ranges to predictable (variable size) named objects, grouped in placement groups, and delivered to OSDs…
    - According to a specific but configurable algorithm (CRUSH).
    - Many configuration rules can be set to control data access, replication, distribution and integrity, with high focus on scalability, performance and redundancy.

CephFS (a nutshell explanation)

- **POSIX compliant filesystem**
  - Drop in replacement for any other local or network filesystem
  - Scalable data and metadata: objects stored directly in RADOS
  - Cluster of metadata servers
  - Extra functionality: snapshots, recursive statistics

Zero size objects.

CephFS Metadata is stored as key-values in the osd leveldb (omap)



- **Jewel 10.2.X (X = 0,1,2,3)**
  - CephFS **stable** release (note that stable ≠ production)

    http://docs.ceph.com/docs/master/cephfs/best-practices/

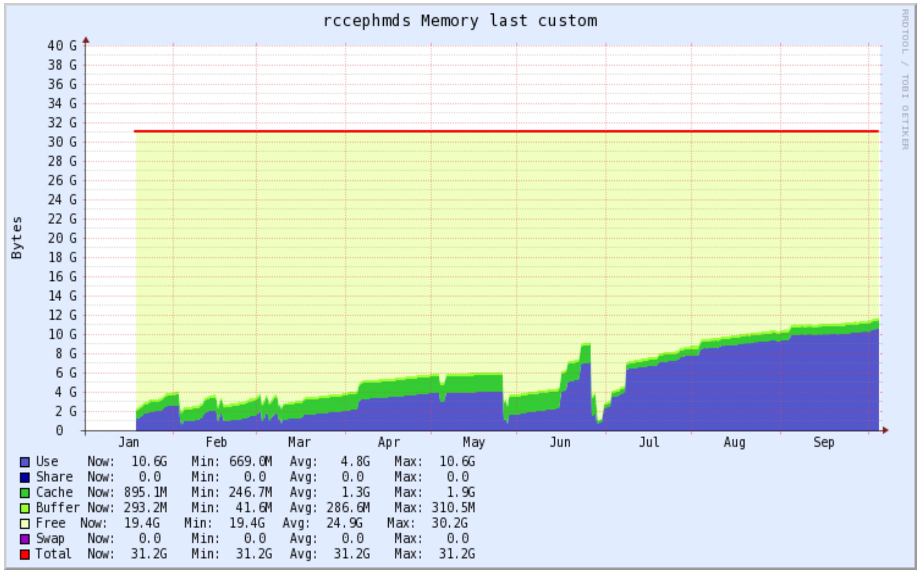
CephFS Clients

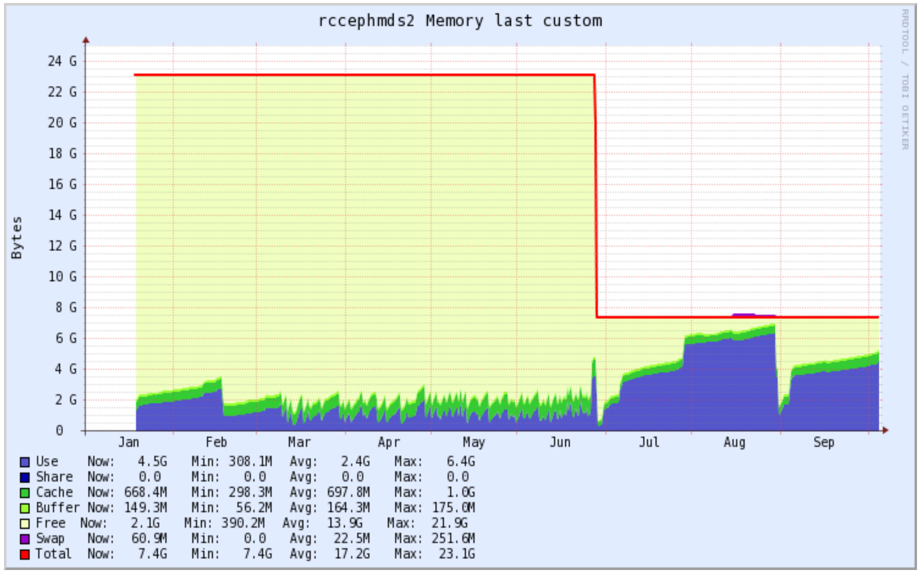
- **CephFS clients (fuse vs kernel)**
  - The kernel client provides the best performance. However, it is always outdated in terms of bug fixes and enhanced functionalities.
  - We opted for the fuse client because
    - Flexibility: Works in user space
    - Reliability: Synced with latest developments
    - Portability: Easily patched, recompiled and deployed.

- **CephFS (10.2.2) under SL6 .**
  - No support for RH6 flavours because it relies on C++11 features only available in GCC > 4.7. SL6 default GCC version is 4.4
  - Compile ceph in SL6 with GCC 4.8, Python 2.7, Fuse 2.9.2 and Boost 1.53
  - ceph-fuse Started by puppet and enabled via Environment Modules.
  - Normally running 200 ceph-fuse clients, mostly over wlan, with the potential to scale up once more VMs are started on Nectar cloud to satisfy demand.

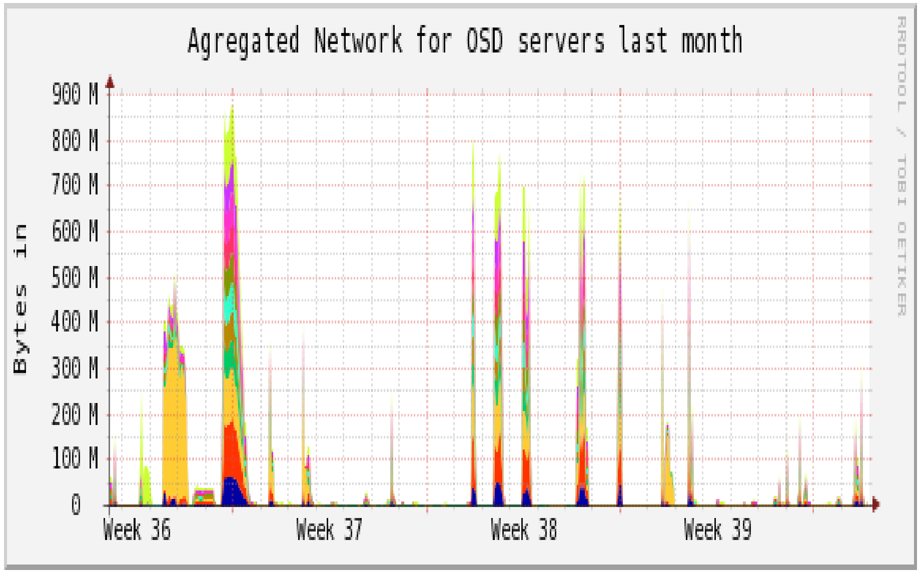
CephFS MDS

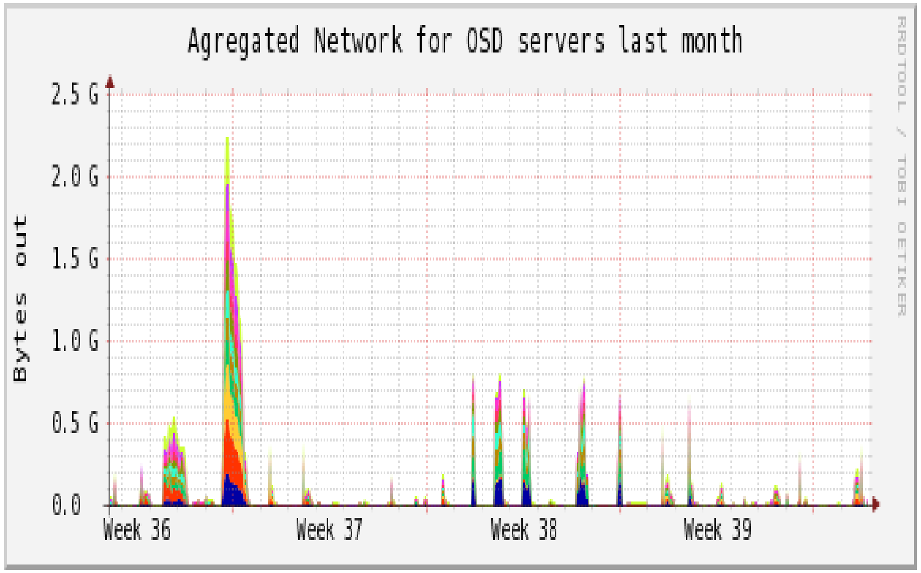
- **Stable configuration (one active MDS + standby-replay MDS)**
  - Currently single threaded, SSD pools may help performance but not critical
- **MDS is (mostly) about RAM**
  - You want to cache as many inodes as possible.
  - The default number of CInodes to cache is 100k. The size of metadata structures is:
    - CInode = 1400 bytes; CDentry = 400 bytes; CDir = 700 bytes → 2KB (?)
  - A 'back of an envelope' calculation give a way to low value 100k * 2KB ≃ 200 MB
  - You should increase 'mds cache size' if you have available RAM





Ceph OSDs


Agregated Network for OSD servers last month
Bytes in
900 M
800 M
700 M
600 M
500 M
400 M
300 M
200 M
100 M
0
Week 36
Week 37
Week 38
Week 39
RRDTOOL / TOBI OETIKER


Agregated Network for OSD servers last month
Bytes out
2.5 G
2.0 G
1.5 G
1.0 G
0.5 G
0.0
Week 36
Week 37
Week 38
Week 39
RRDTOOL / TOBI OETIKER