Current Status of the Ceph Based Storage Systems at the RACF

Alexandr Zaytsev
Christopher Hollowell
Hironori Ito
Tejas Rao
Tony Wong

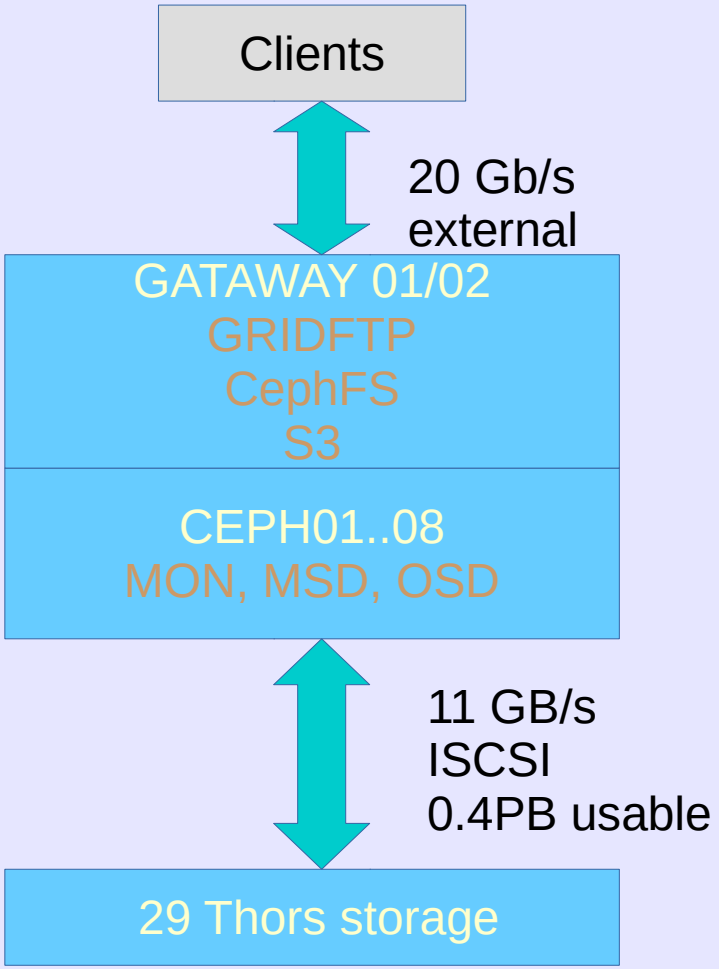
Brookhaven National Laboratory

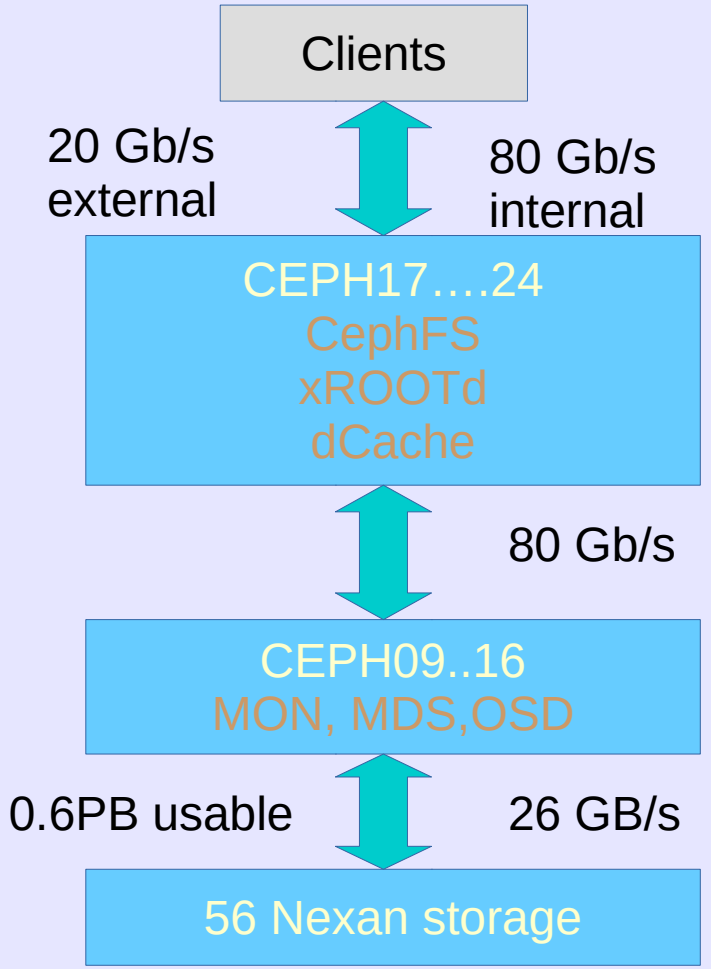
CHEP 2015
Okinawa, Japan

Motivation for Using Ceph

- Reliable storage
  - Data reliabilities
    - Multiple data reliability mechanism: Replications or erasure
  - Service reliabilities
    - No single point of failures
- Direct use of object storage layer
  - Do we really need file system as a simple or part of a catalog?
    - Our communities generally have external catalogs located elsewhere to keep track of files at particular location.
  - Performance gain of not using file system.
    - Flat system.
      - Not worrying about the limitation of inodes
  - Can we use object storage efficiently without the file system.
    - Sometimes, our thinking and/or coding assume the existence of the file system even though we don't use it explicitly.
      - Eg, find files older than 1month → mtime -mtime +30 → scan all objects in buckets to check the metadata
- Reuse of retired storage hardware
  - Extend the life of out-of-service, non-reliable storage
- Distributed file system

Current Hardware Setup





Available Interfaces

- RadosGW/S3
  - Object storage
  - Http
  - User (access key id) and password (secrete access key)
  - APIs available in many languages
  - No segmented read/write
- Block device (RBD)
  - Mounted in a host.
  - Use it like any other locally mounted storage.
    - Use it as a part of any of your favorite storage
      - XRootD data server
      - dCache storage pool
- CephFS
  - Provide the distributed file system.
  - Use it like any other distributed file system.
    - GridFTP server
    - XRootD
    - Etc...

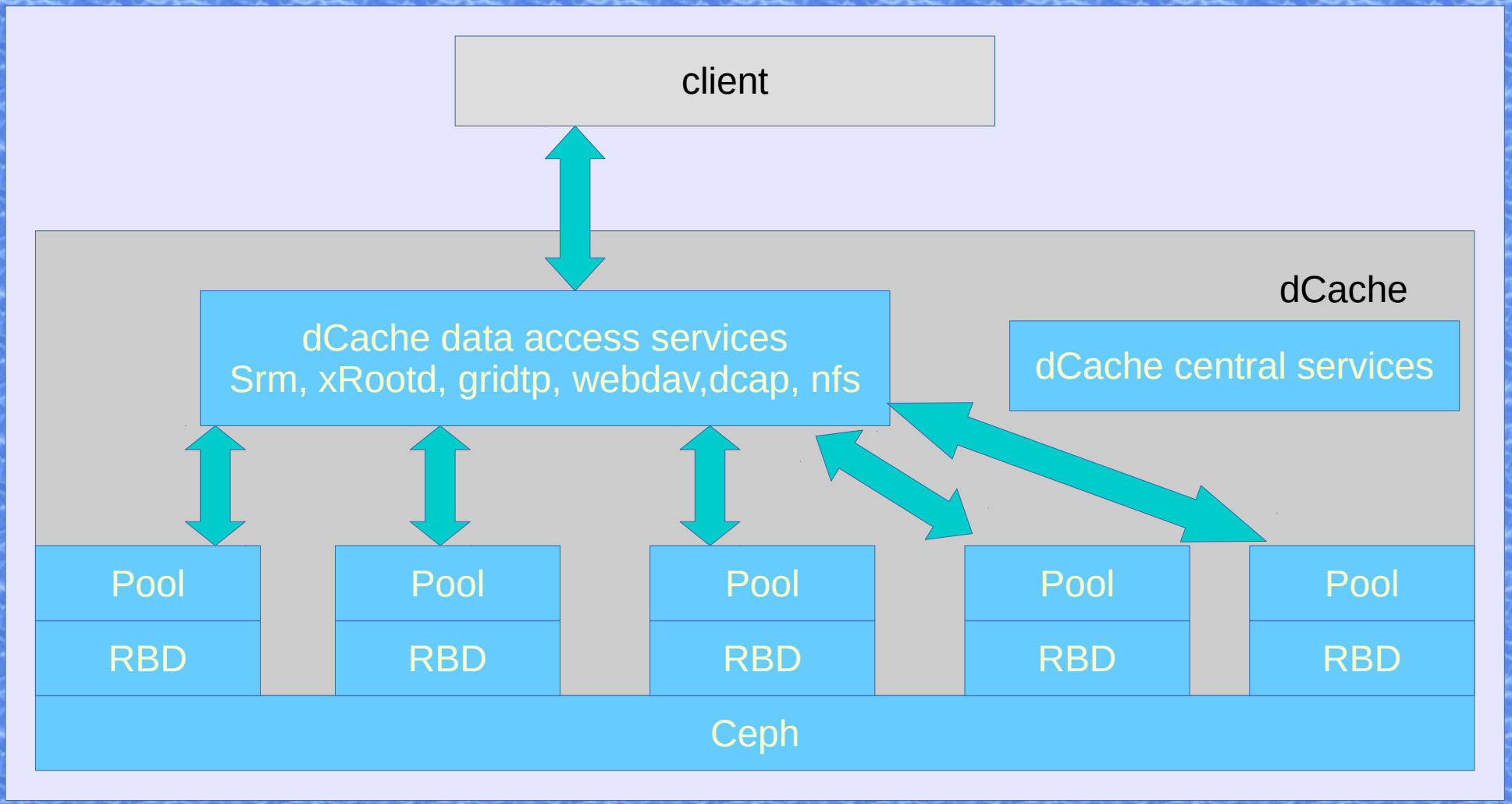
Use Case 1: RadosGW/S3

- Object storage
- Large number of small data
  - Logs, data per events, etc...
- High rate of writes
  - No File system
    - Files should be cataloged externally
      - Searching a object in S3 can be slow process.
- Being tested by ATLAS

Use Case 2: Block Device (RBD)

- Mounted storage
  - Single mount can write.
  - Can mount and read on multiple hosts.
  - The data are distributed.
  - The write will always hit/write OSD on the mounted host.
- Example
  - dCache Storage Pools
    - A pool is stateful.
      - It is associated with particular host only.
    - It has own replications but still stateful.
    - It can't increase the availabilities as it stands.

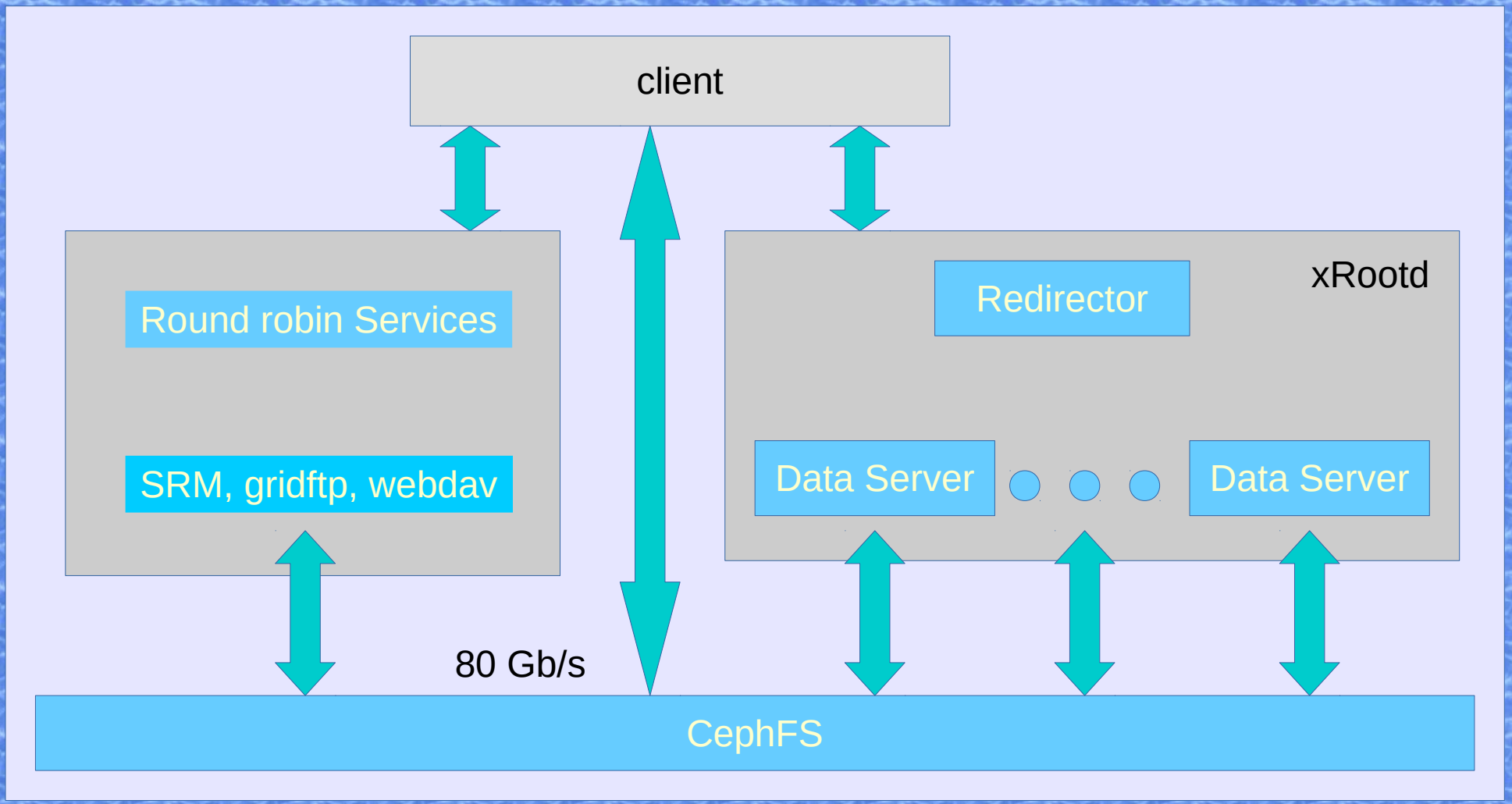
Ceph Block Device



Use Case 3: CephFS

- Features
  - Distributed file system
  - Linux Kernel support
  - POSIX comliant
  - Writes can be efficiently distributed among multiple OSDs
  - Striping, object size and destination pool control via extended file attributes mechanism
- Example
  - GridFTP server
  - WebDAV via Apache + grid-site
  - XRootD data sever
    - Data server is stateless
      - It relies on the underline file system
      - It has no file system of its own
    - Multiple data servers can use the single distributed, mounted file system
      - Highly resilient, distributed storage with a posix file system

CephFS with Stateless Frontend services



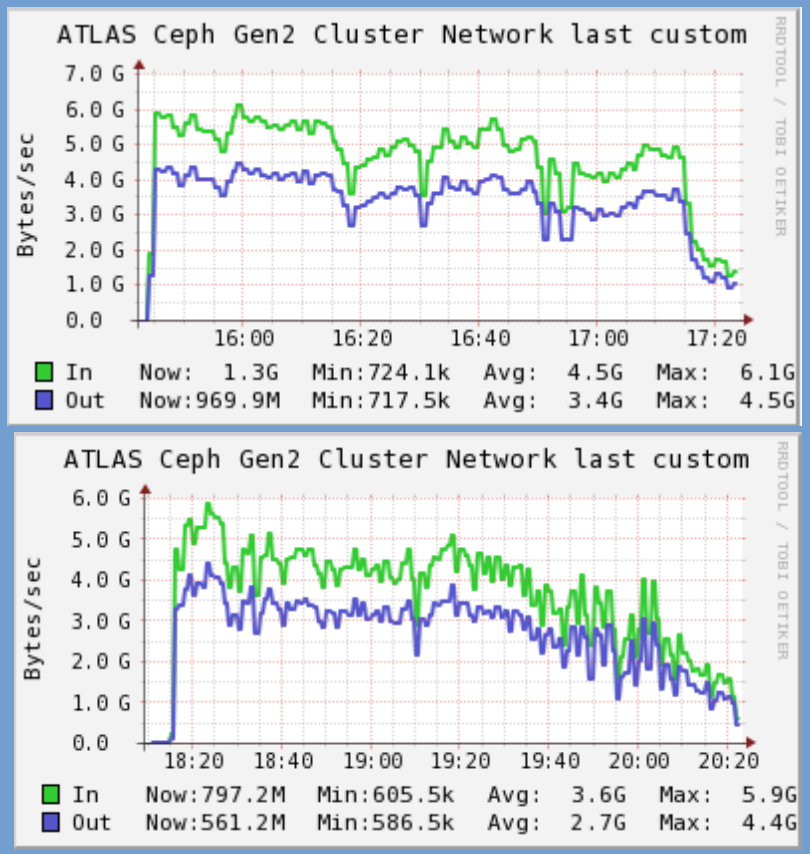
Performance Tests (Writes)

Test Details

- 74 clients with 1Gb/s
- Write files
  - ~a few GB / file
- Protocol
  - xRoot
- Program
  - xRootd
  - Xrdcp
- Data replication factor of three

Observations

- 1.5 GB/s maximum
- Degradation of performance with more writes



Shown the total network activity of entire Ceph cluster. Useful bandwidth seen by clients is 1/3 of what is shown

BROOKHAVEN NATIONAL LABORATORY

CHEP2015 OKINAWA, japan

US ATLAS

Performance Tests (Reads)

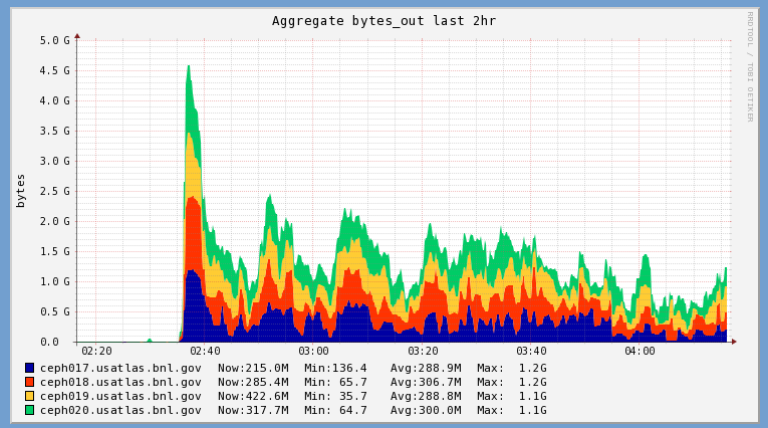
Test Details

- 74 clients with 1Gb/s
- Read
- Protocol
  - xRoot
- Program
  - Xrootd
    - 4 or 8 data servers
  - Xrdcp
- Data replication factor of three

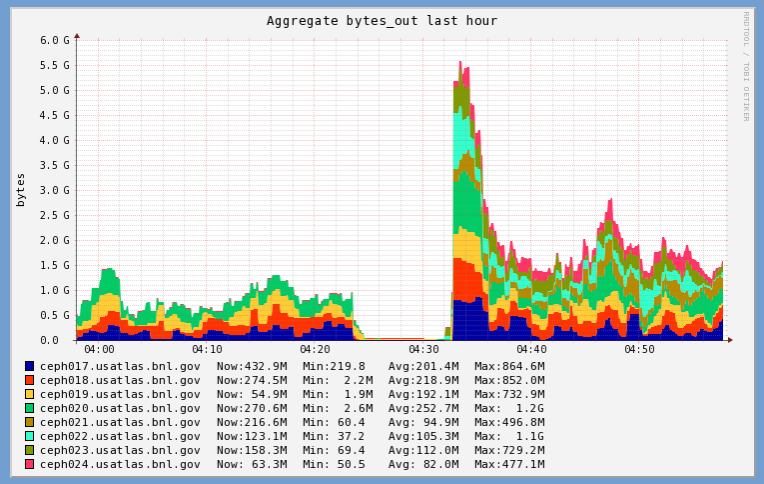
Observations

- Large spike on the first read.
- The aggregated useful bandwidth on the clients seems to be unaffected by the number of data servers involved in the test

4 data servers



8 data servers



BROOKHAVEN NATIONAL LABORATORY

CHEP2015 OKINAWA, japan

US ATLAS

Future Plans

- Further testing and tuning of systems under various front end services
  - GridFTP, WebDAV, XRootD, dCache
- Understand how to use the object storage layer with maximum efficiency
  - Identify things to avoid
- Federated storage across multiple regions

Conclusion

- Ceph can be utilized to provide various storage services that our communities commonly use:
  - GridFTP, dCache, XRootD, WebDAV
- Ceph RadosGW/S3 allows us to use the object storage layer of Ceph without the limitations and disadvantage of a file system
- Measuring performance of all three storage layers of Ceph under the common use cases provides us with information needed for using Ceph as a backend for other storage services frequently used by HEP/NP community