Building a Parallel Cloud Storage System using OpenStack's Swift Object Store and Transformative Parallel I/O

or

Parallel Cloud Storage as an Alternative Archive Solution

Kaleb Lora

Andrew "AJ" Burns

Martel Shorter

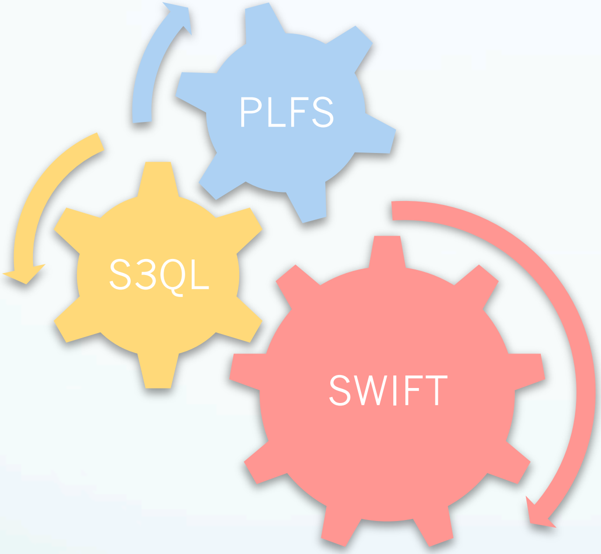
Esteban Martinez

LA-UR-12-23631

Overview

- Our project consists of bleeding-edge research into replacing the traditional storage archives with a parallel, cloud-based storage solution.
- Used OpenStack's Swift Object Store cloud software.
- Benchmarked Swift for write speed and scalability.
- Our project is unique:
  - Swift is typically used for reads
  - We are mostly concerned with write speeds

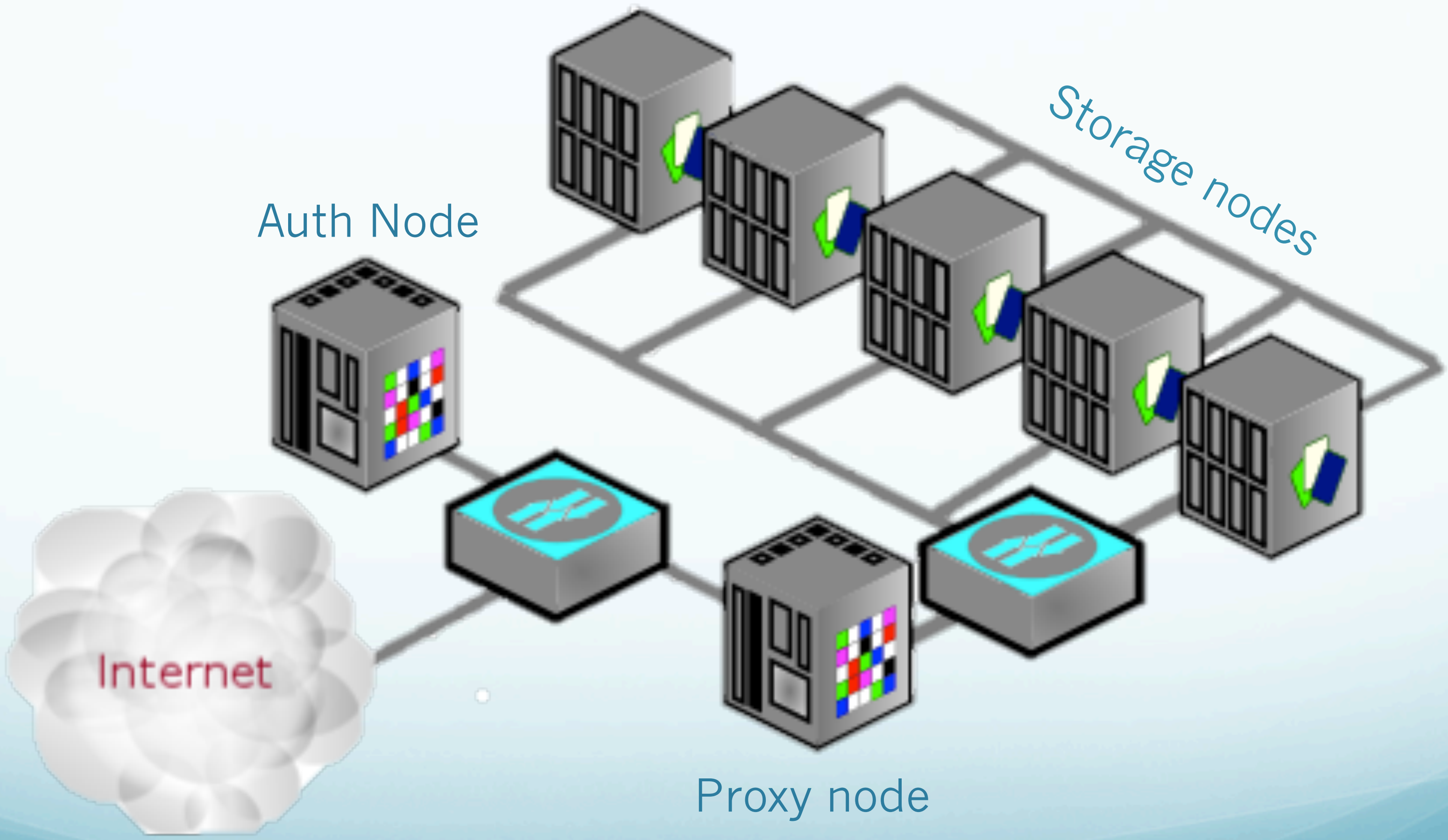
Tools/Software



- Swift
- FUSE
- S3QL
- PLFS

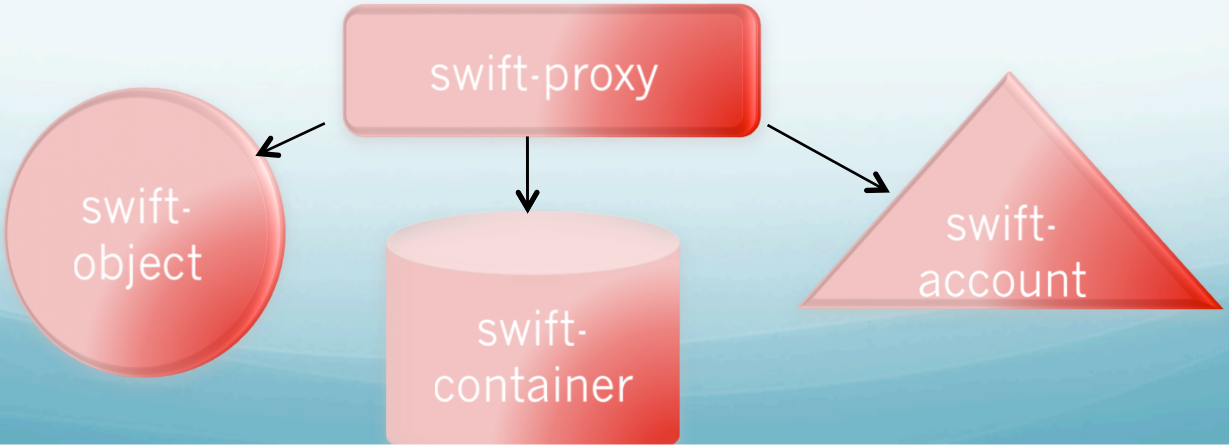
openstack™

Typical Swift Setup



Swift Component Servers

- **Swift-proxy**—Serves as the proxy server to the actual storage node. Ties all components together.
- **Swift-object**—Read, write, delete blobs of data (objects).
- **Swift-container**—Lists and specifies which objects belong to which containers.
- **Swift-account**—Lists the containers of Swift.



S3QL

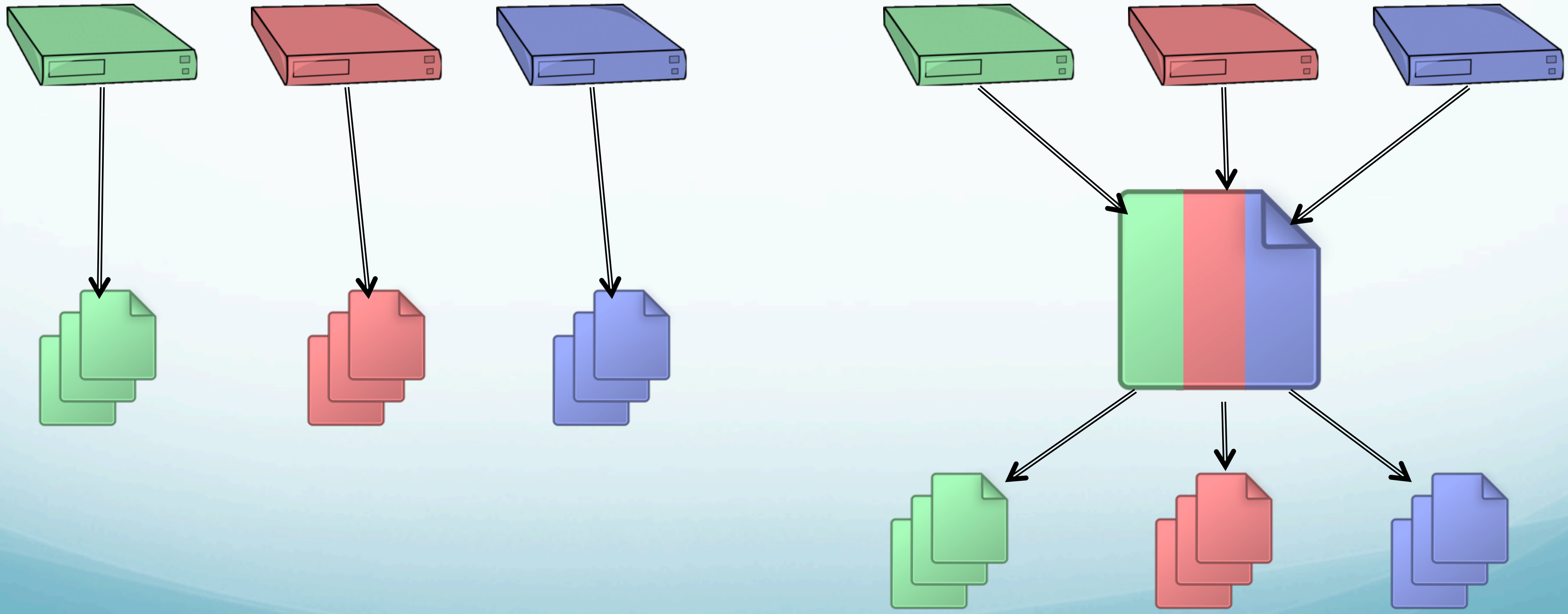
- Full-featured Unix filesystem.
  - E.g.: `/mnt/s3ql_filesystem/`
- Stores data online using backends:
  - Google Storage
  - Amazon S3(Simple Storage Service)
  - OpenStack
- Favors simplicity.
- Dynamic capacity.

Parallelization via N-N and N-1-N

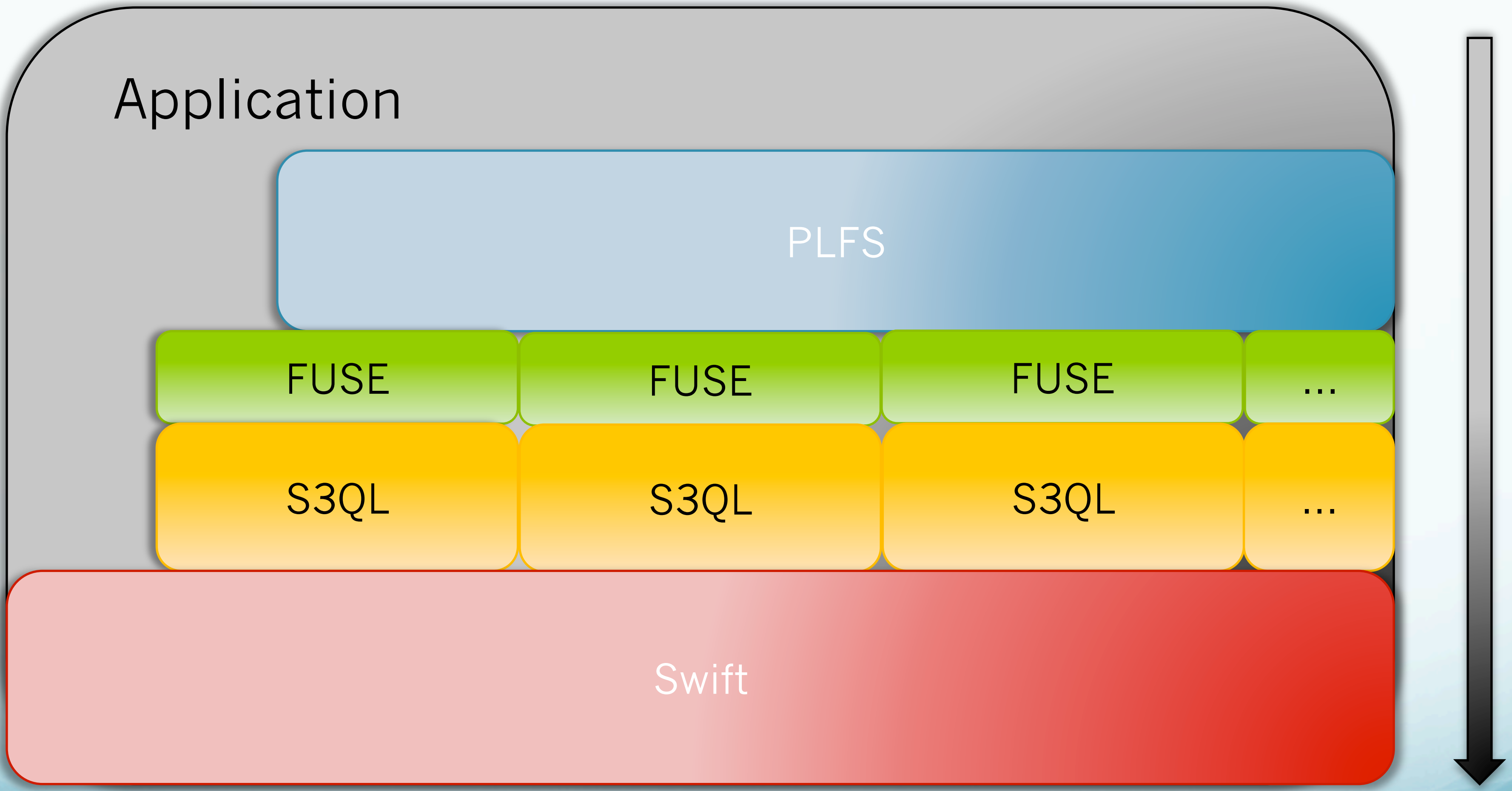
- PLFS is LANL’s own approach to parallelized data storage.
- Appears as an N-1 write(left), but actually is an N-1-N write(right).

N-N

N-1-N



How the Four Applications Interact



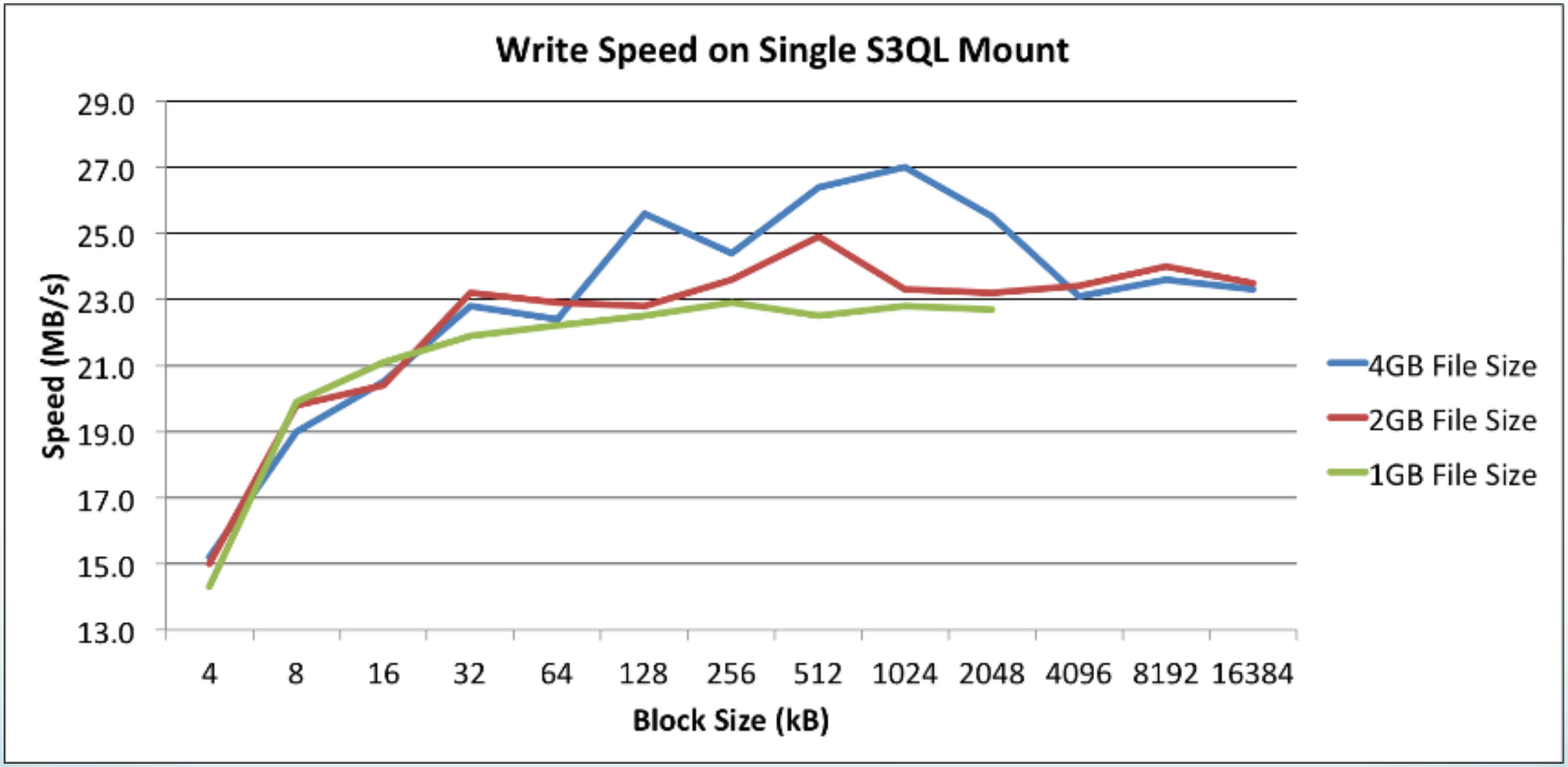
Baseline Performance Testing

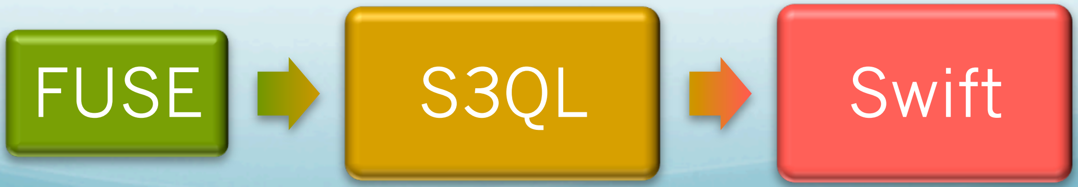
Single Node Tests

Baseline Test Setup

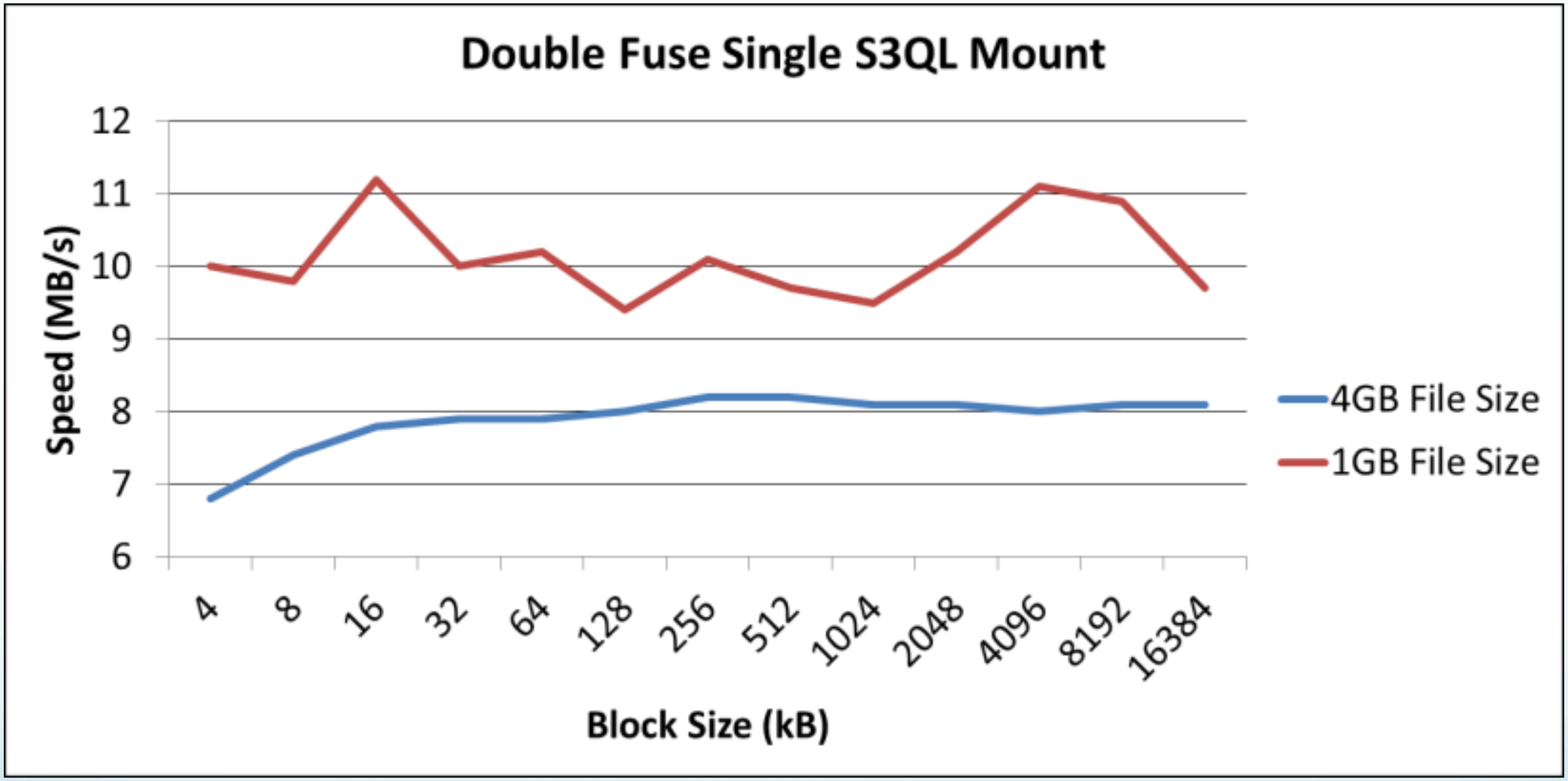
- Wrote a script to write various block and file sizes
- Wrote 1GB, 2GB, and 4GB files
- Tested multiple configurations
  - single write to a single file system
  - single write to single PLFS mounted file system
  - 3 separate writes to 3 file systems simultaneously
- Graphed the results to watch trends

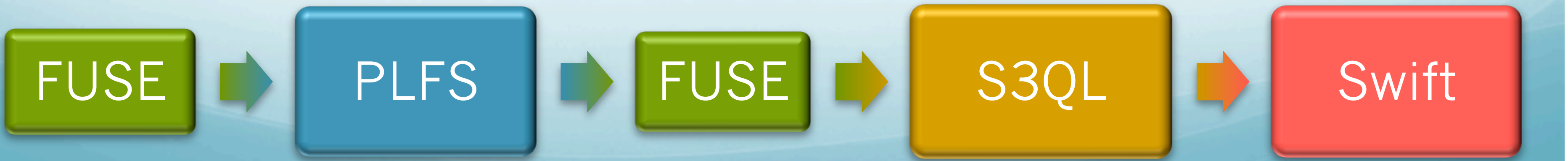
Found Ideal Block Size



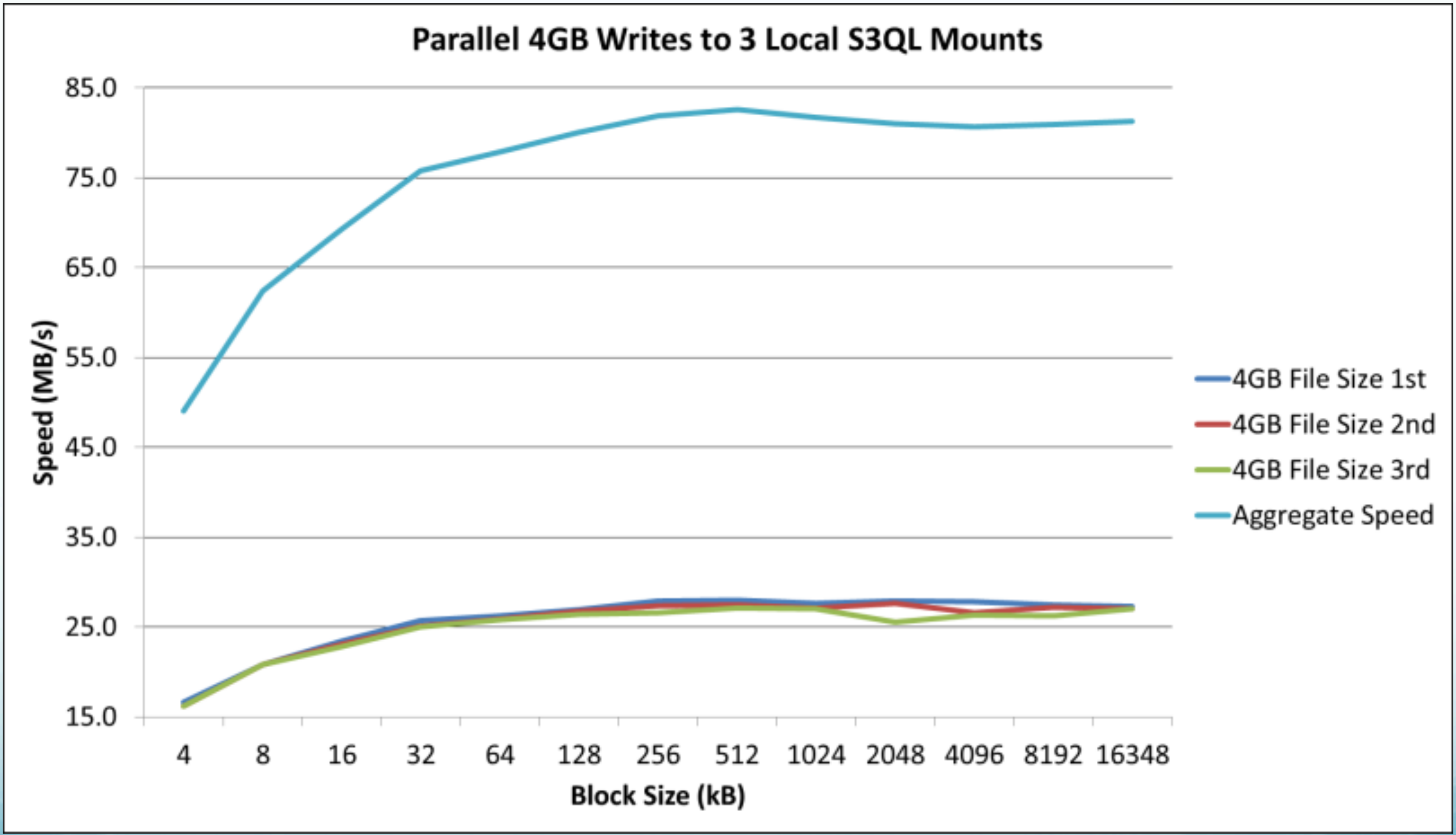


Discovered FUSE Limitations





Local Parallelization Increased Performance



Baseline Performance Testing was Successful

- We found an ideal block size.
- Single node parallelization is efficient
- FUSE is a limiter in our setup
- Single write performance was in line with normal cloud storage performance (~25-30MB/s)

Target Performance Testing

Parallelization Benchmarking and Scalability

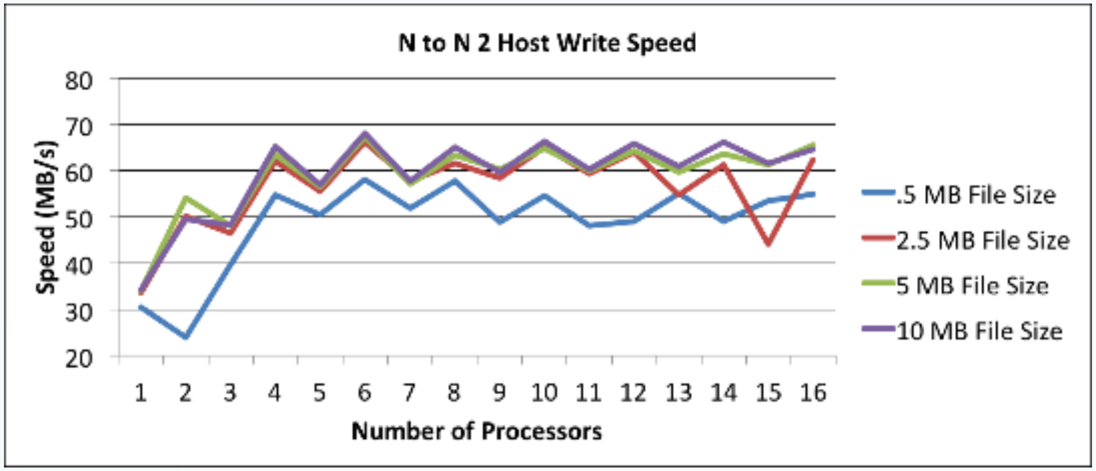
Target Performance Testing Used Multiple Nodes

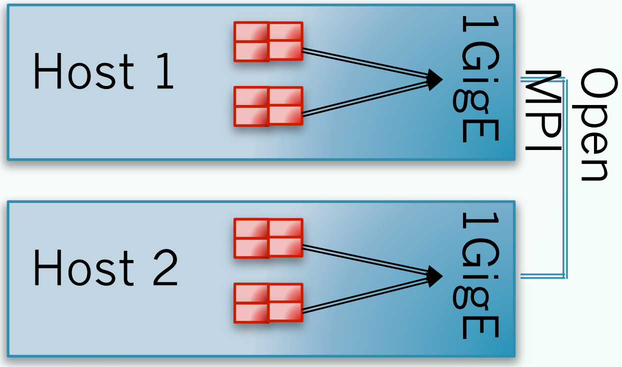
- Used Open MPI for parallelizing tests across the whole cluster.
- Tested performance scaling from 1 to 5 hosts.
- We were able to get 40 processes running at once because each host contained 8 cores.

N to N Write Tests had Interesting Results

- Immediate performance improvement with adding nodes even with a small number of processors per node
- Also noticed spikes of increased performance at each number of processes that was a multiple of the number of hosts we were using
- Stable, didn't break the S3QL mounts to the Swift containers

2-3 Host Test Results





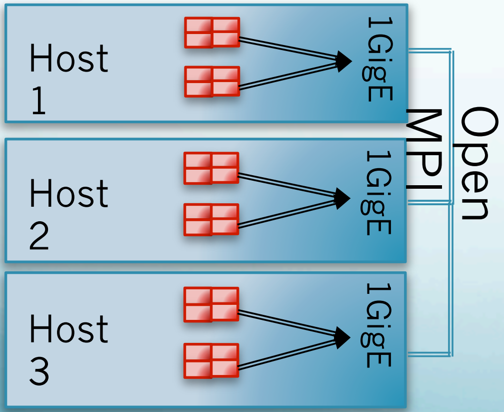
Host 1

1GigE

Host 2

1GigE

Open MPI



Host 1

1GigE

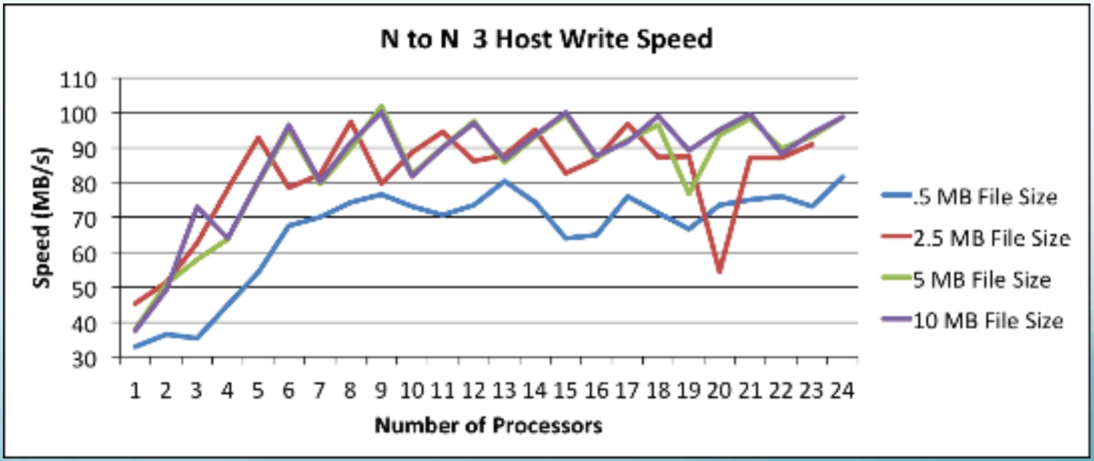
Host 2

1GigE

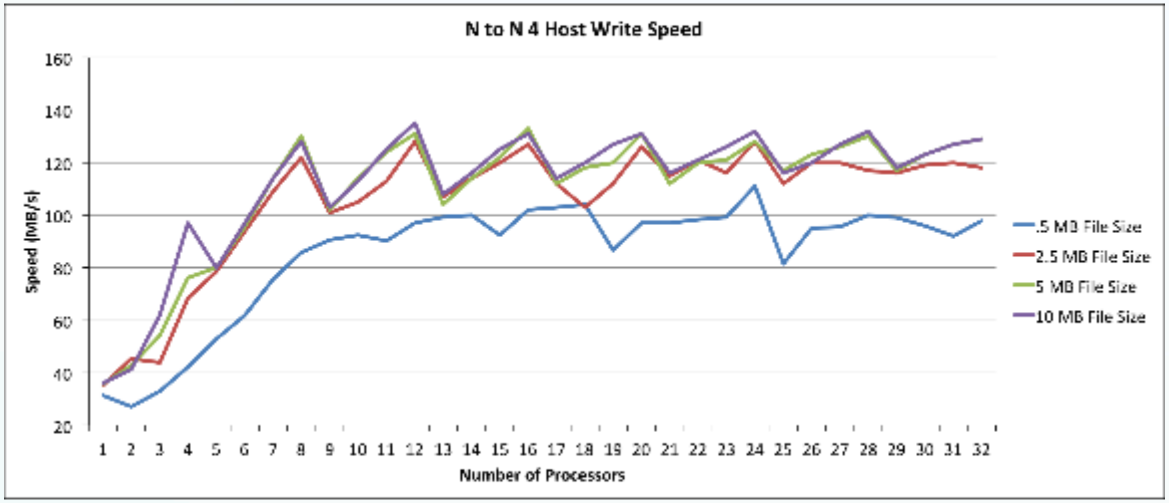
Host 3

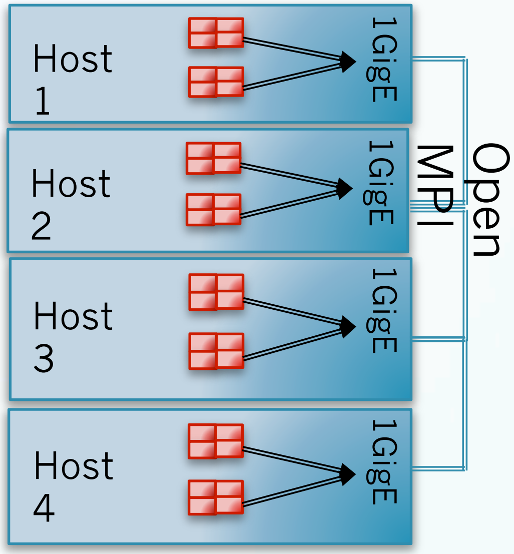
1GigE

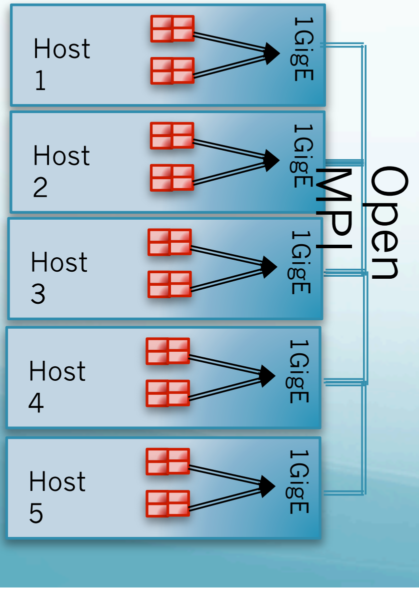
Open MPI

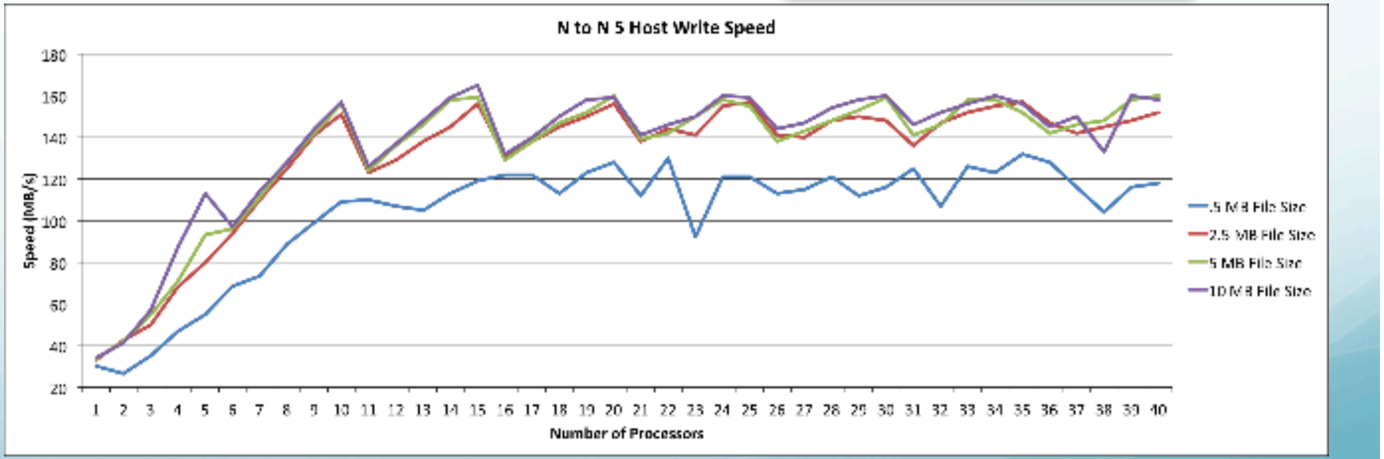


4-5 Host Test Results



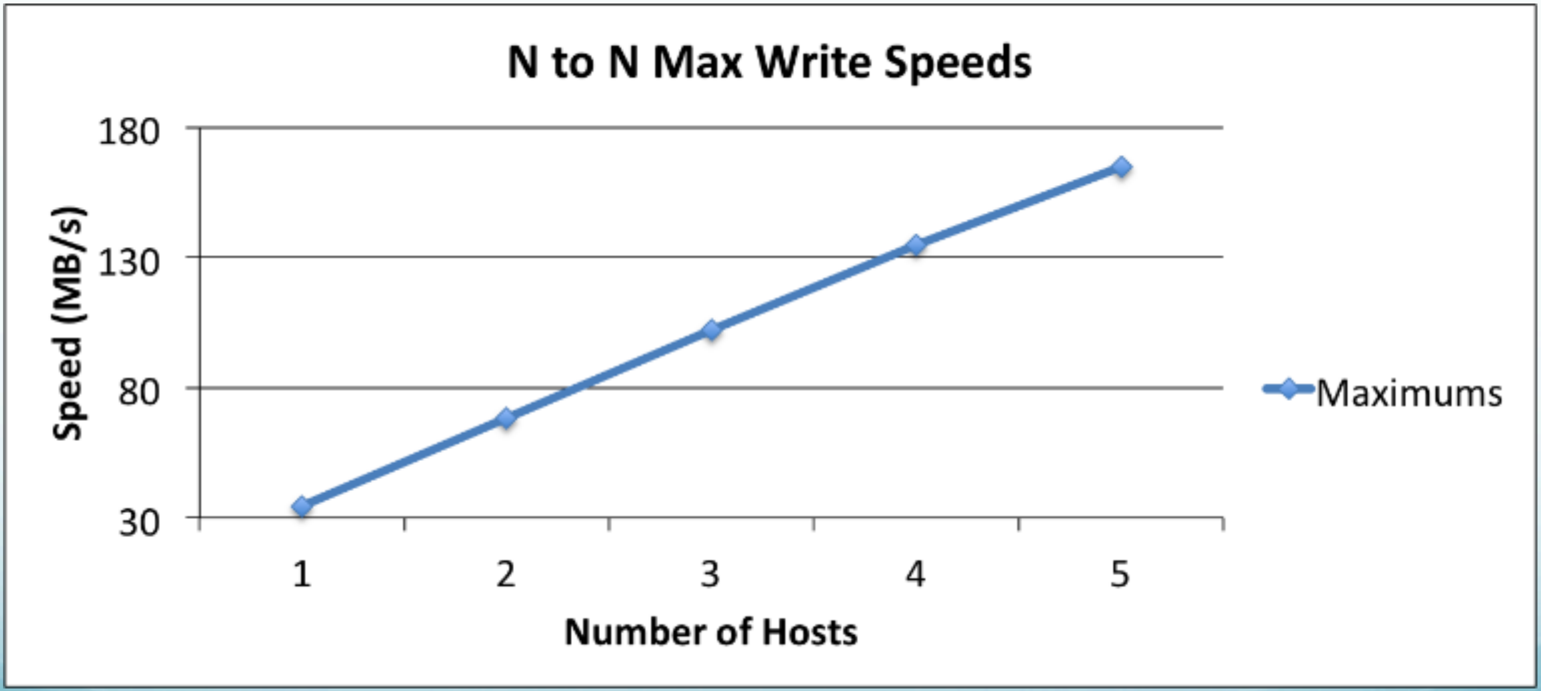






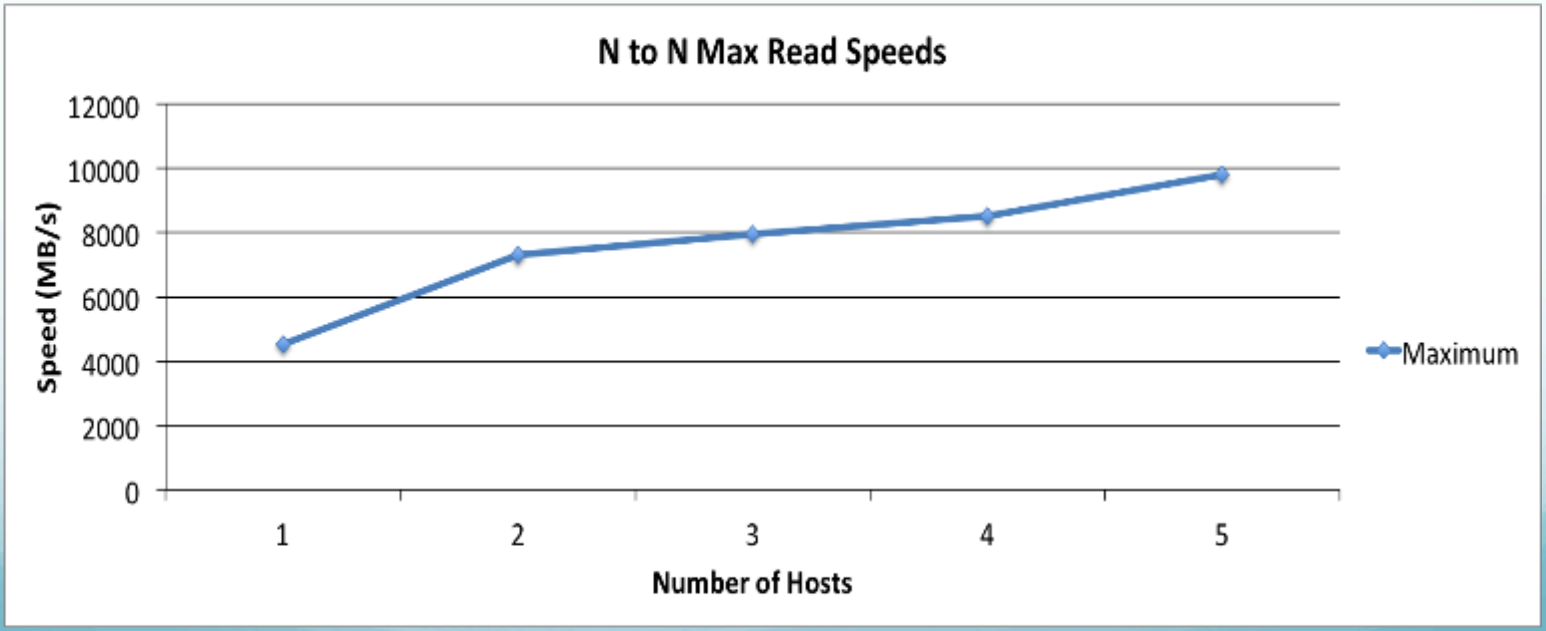
Our Tests Show Cloud Storage Scales Well

- Performance scales linearly as you increase the number of hosts being used for MPI



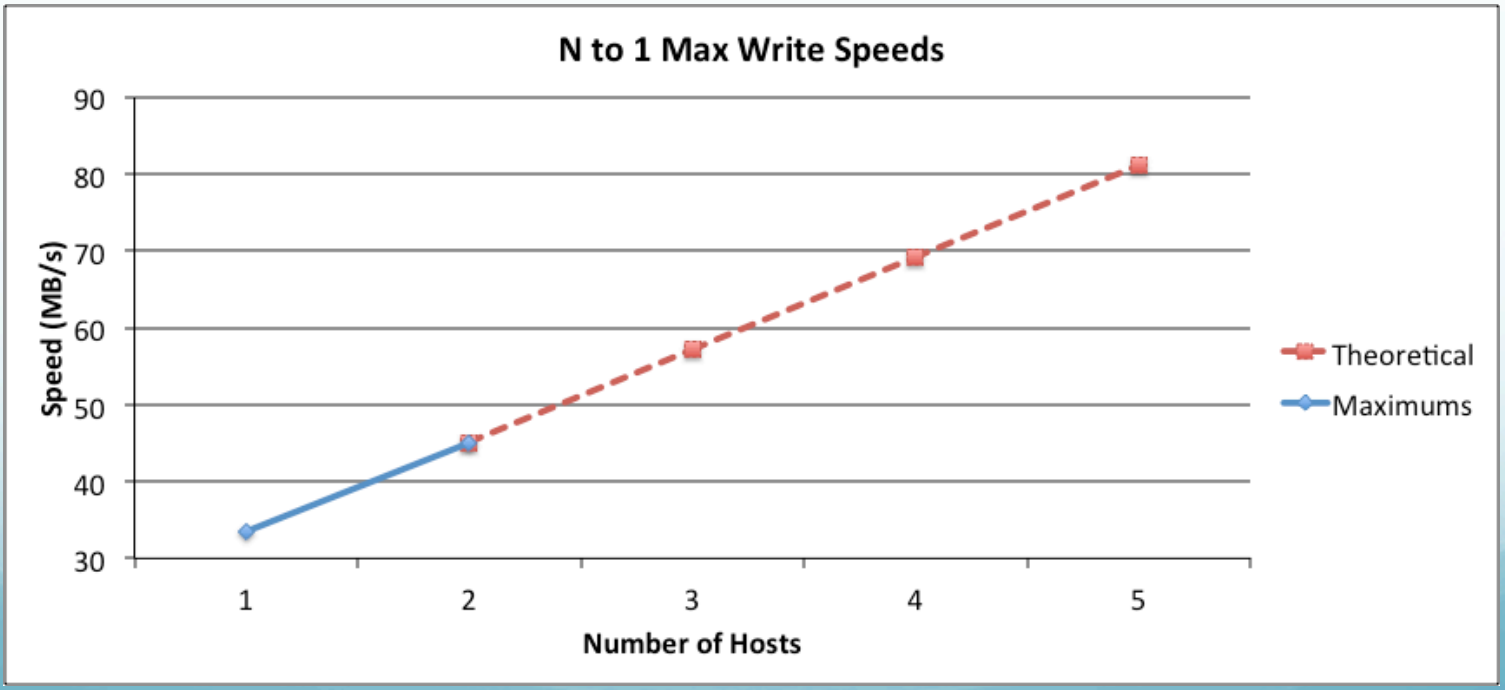
Read speeds are fast but don't tell the whole story

- Incredibly fast due to caching
- Scales very well as you increase the number of hosts being used



More work needs to be done with PLFS and S3QL

- PLFS performance results were similar to N to N performance results but added enough instability to the S3QL mounts that many failures prevented a complete set of tests



Cloud Storage is a Viable Option for Archiving

- Parallel cloud storage is possible and has good scalability in the N to N case.
  - Linear as nodes were added
- More work will need to be done to get PLFS working without breaking the S3QL mounts.

Future Work and Conclusion

Further research possibilities of cloud parallelization

Future Testing

- Test write performance impacts of increased S3QL cache sizes.
- Test CPU load impact of S3QL uncompressed vs the default LZMA compression
- Test swift tuning parameters to handle concurrent access for added stability of PLFS testing.

Other File Systems That Could Be Tested

- Test GlusterFS and Ceph as alternative cloud solutions to swift

Why is Cloud Storage a Viable Archive Solution

- Container management for larger parallel archives might ease the migration workload..
- Many tools that are written for cloud storage could be utilized for local archive.
- Current large cloud storage practices in industry could be utilized to manage a scalable archive solution.

Acknowledgements

- LANL
- Dane Gardner (New Mexico Consortium)
- H.B. Chen, Benjamin McCleland, David Sherill, Alfred Torrez, Pamela Smith, and Parks Fields (High Performance Computing Division)

Questions?