LA-UR-14-26016

Los Alamos NATIONAL LABORATORY EST. 1943

PRObE Parallel Reconfigurable Observational Environment

New Mexico CONSORTIUM

NSF

ISTI INFORMATION SCIENCE & TECHNOLOGY INSTITUTE

The Effects of SSD Caching on the I/O Performance of Unified Storage Systems

Team Chartreuse

Heidi Sandoval
California State University

Matthew Dwyer
Lynchburg College

Anthony Pearson
St. Cloud State University

CSU BAKERSFIELD

LYNCHBURG COLLEGE EST. 1903

ST. CLOUD STATE UNIVERSITY

Outline

- Unified Storage
- SSD/Flash Caching
- Testbed
- Obstacles
- Conclusions
- Future Work

Introduction

- The Lab is currently utilizing tape drives as its primary backup storage method
  - Slow and expensive
  - Will this continue to scale?
- This experiment tests a Unified Storage System with a layer SSD/Flash caching
  - Faster I/O performance
  - Enhanced fault tolerance

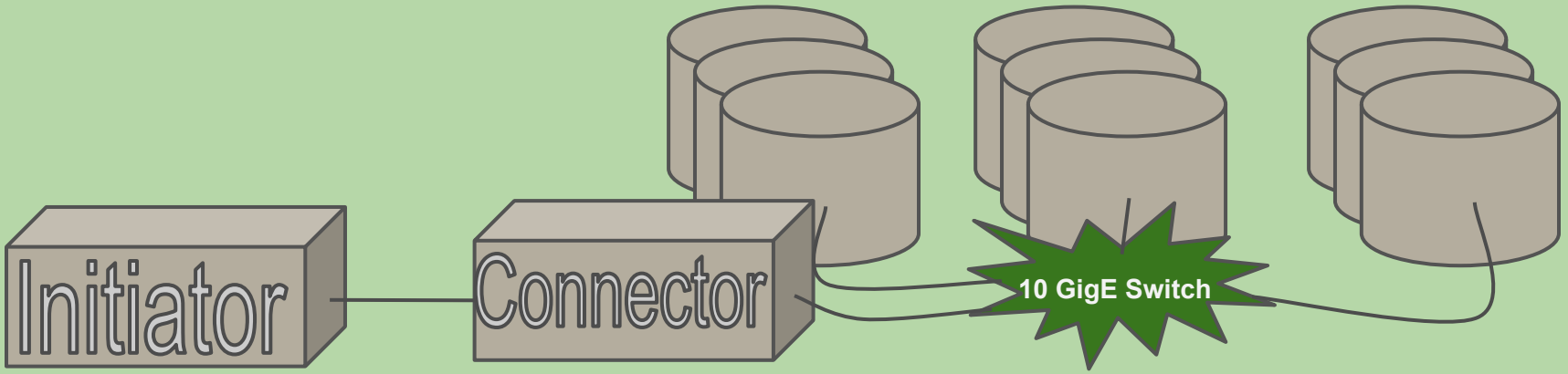
Goals

1. Implement a Unified Storage System
2. Test the impact of SSD/Flash caching on the I/O performance of the Unified Storage System

LA-UR-14-26016

Unified Storage

- Combination of two different storage systems that creates a single integrated storage structure using:
  - Storage Area Network (SAN)
  - Cloud object storage



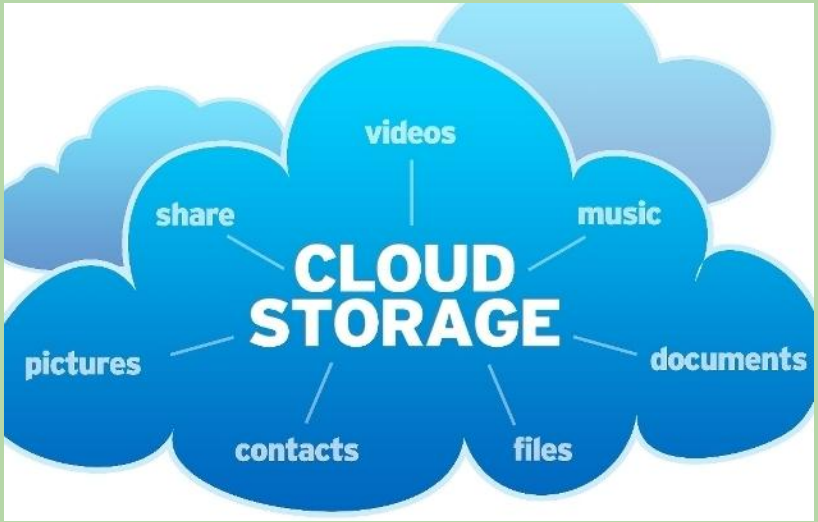
Storage Area Network (SAN)

- Block level protocols
- Communicates over Fibre Channel
  - Hardware used:
    - QLogic Corp. ISP2532 8Gb Fibre Channel
- Uses the Encapsulated SCSI protocol
  - Allows nodes to discover SCSI target devices
  - Software used:
    - targetCLI
- Enforces POSIX style file environment

Cloud Object Storage

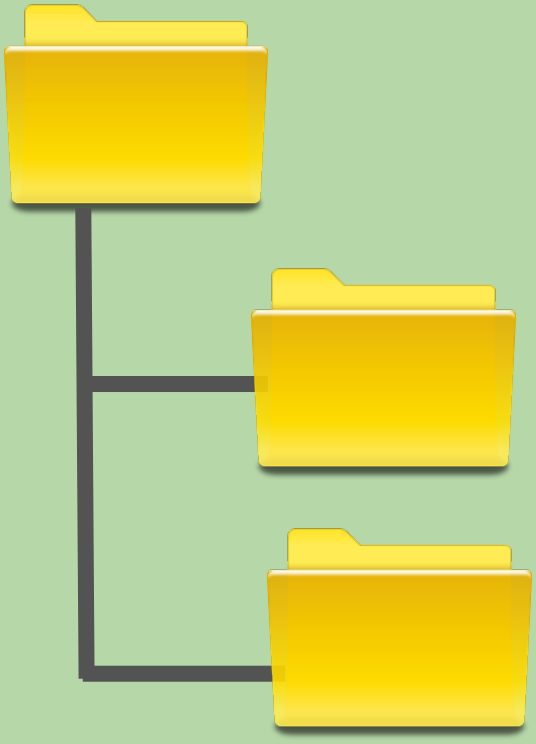
- Communicates over TCP/IP
  - Connected in a ring topology
  - Hardware used:
    - 10GigE Ethernet Switch
    - Mellanox MT26448 10GigE
    - Myricom Myri-10G
- Servers within the cloud are solely used for storing and retrieving files

Unified Storage Representation



Bridging

SAN Storage



Benefits

- Using a Unified Storage System enables us to reap the following benefits:
  - Reduced Hardware Requirements
  - Uses a POSIX interface to perform I/O operations on remote block devices
  - Fast ethernet connection (10 GigE) among the storage nodes for communication
  - Fibre channel is a reliable method for transferring data
    - Often used in secure corporations
  - Implements object storage, which allows for the usage of erasure coding

SSD/Flash Caching

- A method used to speed the I/O processes of local and remote block devices by caching data to faster SSD/Flash devices
- Methods Used:
  - dm-cache
    - write-back enabled
  - bcache
    - write-back enabled
- Hardware Used:
  - Samsung Evo 1TB SSD
  - OCZ PCI-E Flash 960GB

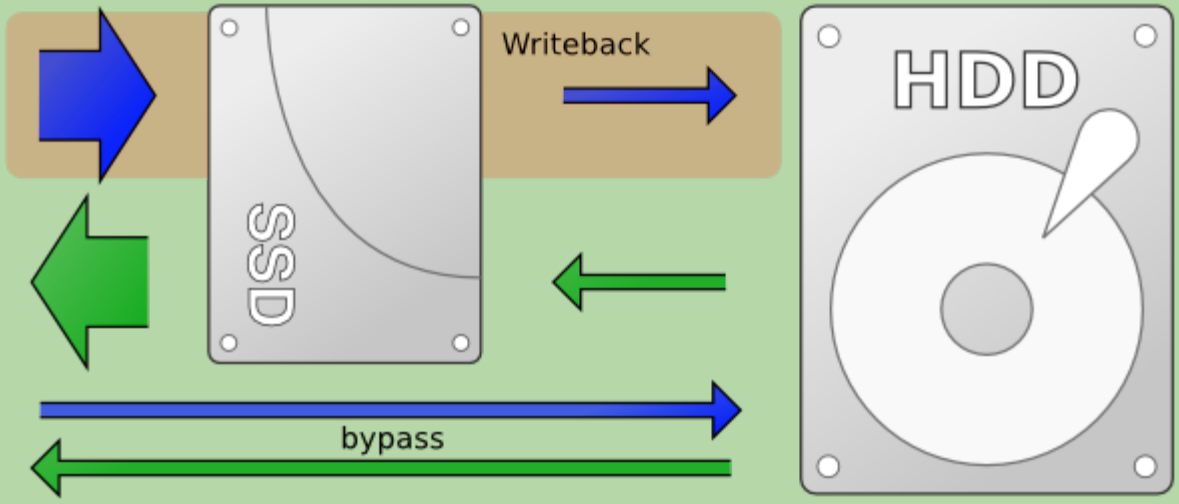
dm-cache

- A device-mapper target that allows the creation of hybrid volumes--block device and SSD combination
- Does not cache data that involves sequential reads and writes (better suited for block devices)
- Requires three physical storage devices:
  - Origin Device: provides slow primary storage (usually a local or remote block device)
  - Cache Device: provides a fast cache (usually a SSD)
  - Metadata Device: records blocks placement and their dirty flags, as well as other internal data

bcache

- Converts random writes into sequential writes
  - First, writes data to the SSD
  - Then, buffers data from the SSD to the HDD in order
- Must be configured to obtain higher performance
  - Parameters such as “sequential_cutoff” must be disabled

SSD/Flash Caching Representation



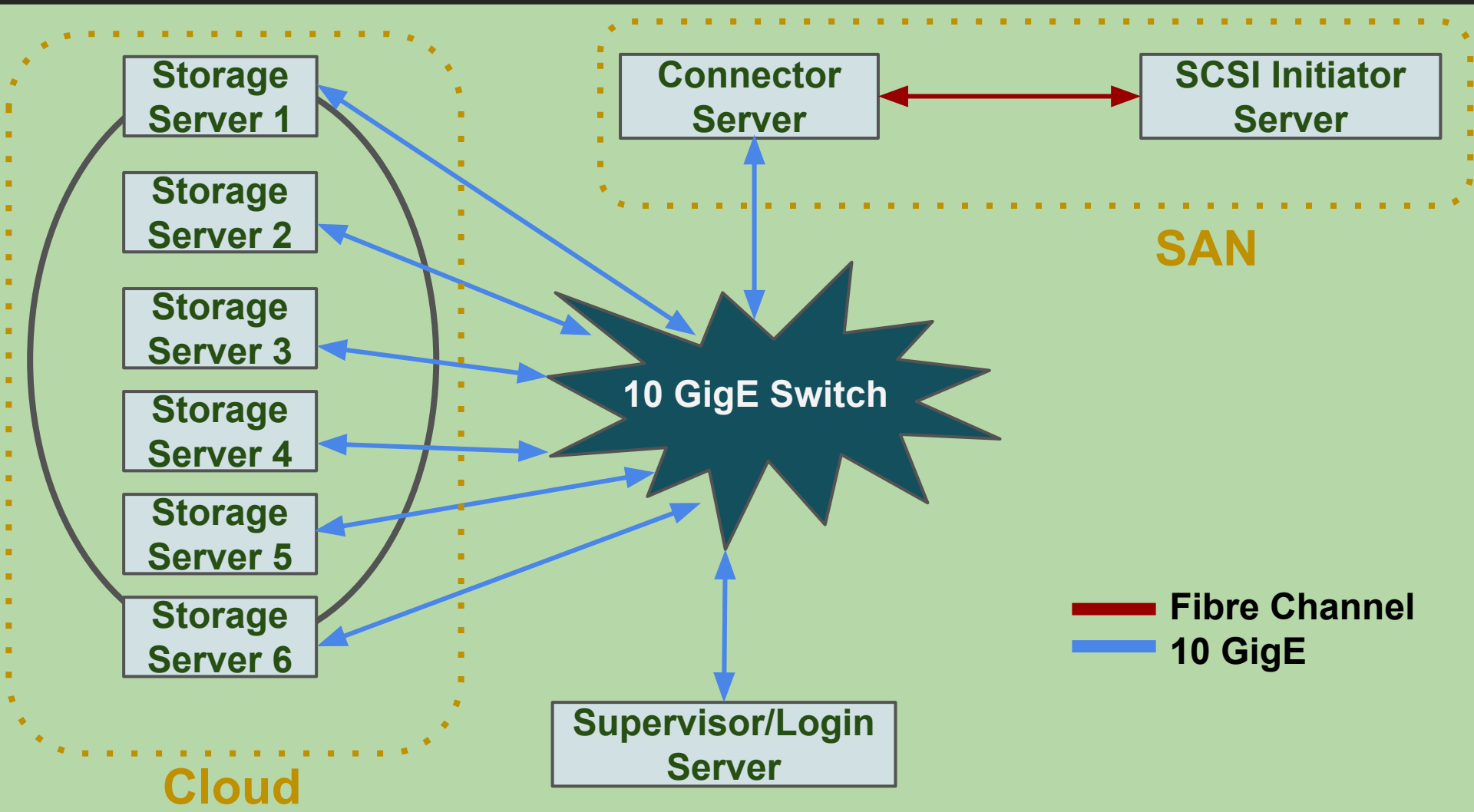
http://pommi.nethuis.nl/ssd-caching-using-linux-and-bcache/

- Demonstrates the writeback caching process
- Initially, writing is done only to the cache
- The write to the backing store is postponed until the cache blocks containing the data are about to be modified/replaced by new content

Benefits

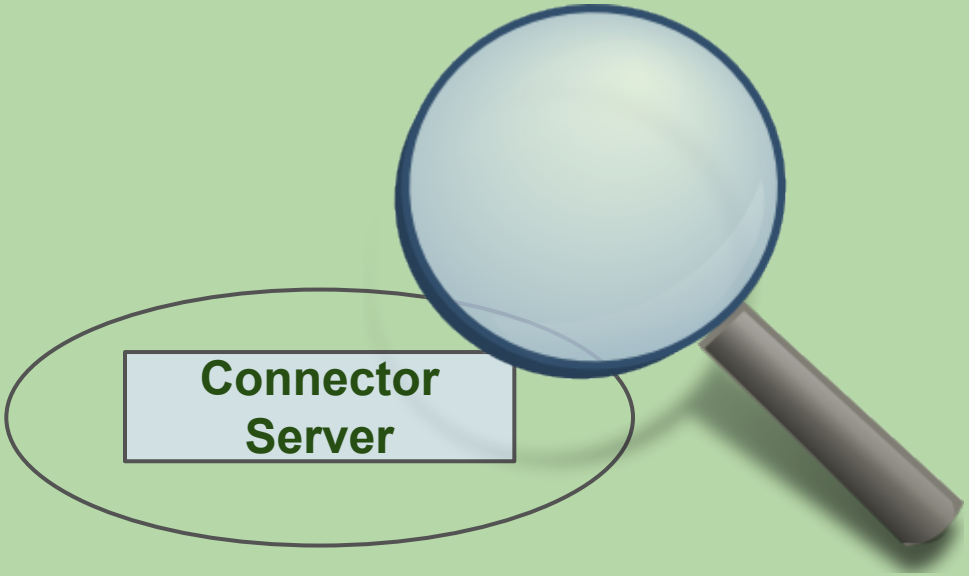
- Slower and cheaper hard drive disks could provide a large amount of storage space
- Faster flash devices could provide rapid I/O speeds
- SSD/Flash caching combines both devices, so the resultant set-up has both a large amount of storage, which operates at a fast rate

Testbed



LA-UR-14-26016

A Closer Look at the Connector



Let's take a closer look at the connector node...

LA-UR-14-26016

A Closer Look at the Connector cont.

- Acts as a bridge between both storage systems
- It uses specialized software, which allows it to read and write to the storage servers
- Even closer…
  - The connector uses a FUSE mount to access the storage servers
  - On that FUSE mount, sparse files were created, so that SCSI targets could be formed using them
  - Later, those SCSI targets were detected by the Initiator node

Benchmarking

- I/O performance tested with and without caching
- Testing methods included:
  - dd: used to run sequential writes
  - iozone: can test a variety of I/O operations both random and sequential
- File sizes ranged from 30-50GB

```
./writeTest.sh -b $blockSize -u $uniqueNum -t $testType -o
$basePath --size $size -d $directory --log $basePath/logs --
sleep 300 -y
```

Obstacles

- Linux distribution conflicts
  - Distributions that worked with the Fibre Channel didn’t work with the cloud software
- Kernel panic within the Connector Node
  - Syslogs point at the fuse mount
- Our SCSI target devices sporadically undiscoverable

Conclusions

- The SCSI fibre channel protocol operates with a limited number of Linux distributions
  - Currently, Ubuntu 14.04 is the only tested working OS
- Unified storage is under early development
- May have needed a larger ring to successfully optimize the I/O of the cloud storage system

Future Work

- Identifying the kernel panic in Connector server
  - Investigate stack trace of dereference null pointer
- Testing the effects of caching directly on the Connector
  - Could eliminate possible latency created during the data movement across the fibre channel
- Creating a RAID 0 array of the four PCI Flash devices
  - Combining all four in a RAID array (960 GB) could give a maximum I/O speed of 1800 MB/s
- Evaluate more caching methods and investigating methods to fine-tune their performance
  - Flashcache and EnhanceIO
- Unified storage setup needs further investigation

Summary

- Unified Storage
- SSD/Flash Caching
- Testbed
- Obstacles
- Conclusions
- Future Work

References

- Unified Storage
  - http://en.wikipedia.org/wiki/Converged_storage
- dmcache
  - http://blog.kylemanna.com/linux/2013/06/30/ssd-caching-using-dmcache-tutorial/
  - http://en.wikipedia.org/wiki/Bcache
- bcache
  - http://www.linux.com/learn/tutorials/754674-using-bcache-to-soup-up-your-sata-drive
  - http://bcache.evilpiepirate.org/

Acknowledgments

- Mentors
  - H.B. Chen
  - Sean Blanchard
  - Jeff Inman
- Summer Institute Instructor
  - Dane Gardner

Questions?

Heidi Sandoval
- Email: heidi.sandoval@hotmail.com

Matthew Dwyer
- Email: dwyer_m@lynchburg.students.edu

Anthony Pearson
- Email: pean0906@stcloudstate.edu