LLNL Production Plans and Best Practices

Lustre User Group, 2014. Miami, FL

April 8, 2014

Marc Stearman

Parallel File Systems Operations Lead

Lawrence Livermore National Laboratory

Overview

- Philosophy
  - To mount, or not to mount?
  - Alas poor drive, I knew you well…

- How many of these file systems do we have anyways?
  - You're running how many versions?

- Where is my file system and what is it doing?
  - You have how many files?
  - Let's go Splunking!

- Could you move my data over there?
  - No, I mean over there…
  - Just a bit to the left…
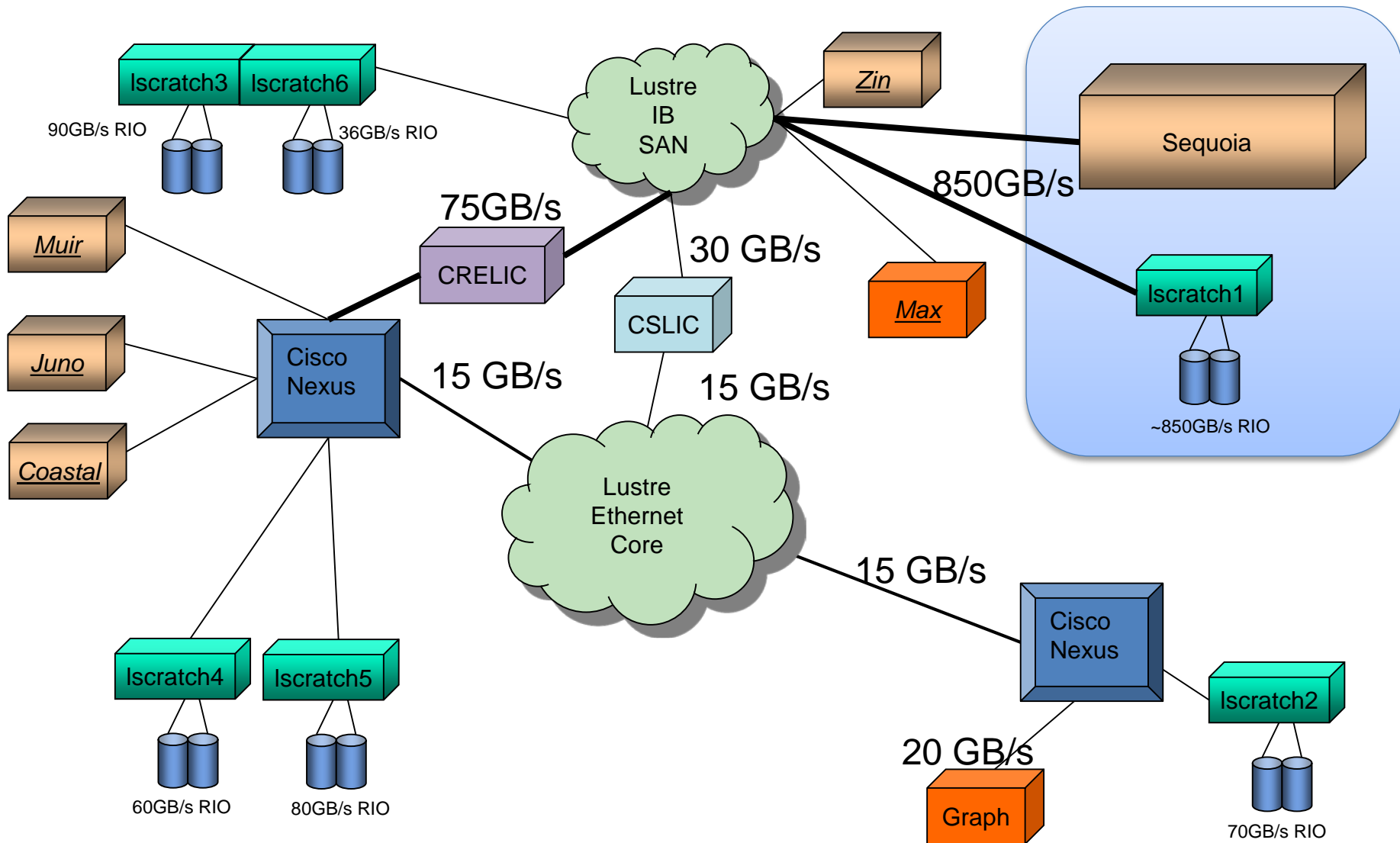 - Maybe a bit more to the right?
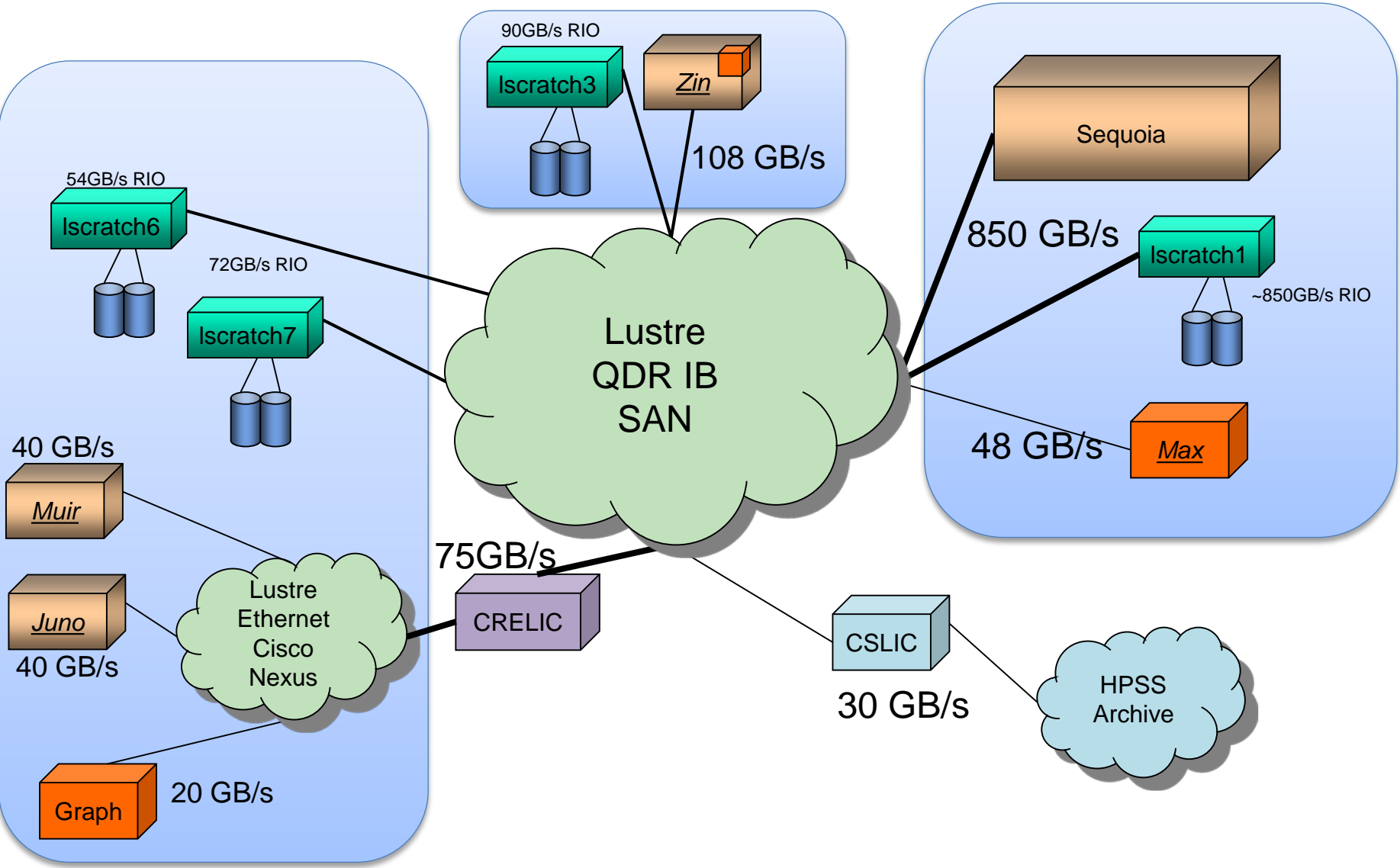
- Now what?

Philosophy – "love of Wisdom"

- Globally mounted file systems.
  - To mount or not to mount
  - 4 years ago…Absolutely!
  - Today, not so much

- File systems:
  - Open Compute – 20PB in 5 File Systems
  - Secure Compute – 22PB in 3 File System plus the 55PB Sequoia File System
  - We use ZFS compression with avg 1.5x ratio
  - File systems range from 200M – 800M files each

- A single client or IB leaf switch can cause cascading problems out into the network
  — This brings the MDS to its knees, often needing a reboot
  — Clients on other clusters are impacted affecting users work

- We are now experimenting by creating smaller islands, and isolating the fault domains

Network Complexity

Network Simplified

90GB/s RIO

lscratch3

Zin

108 GB/s

54GB/s RIO

lscratch6

72GB/s RIO

lscratch7

40 GB/s

Muir

Juno

40 GB/s

Lustre
Ethernet
Cisco
Nexus

Graph

20 GB/s

Lustre
QDR IB
SAN

75GB/s

CRELIC

Sequoia

850 GB/s

lscratch1

~850GB/s RIO

48 GB/s

Max

CSLIC

30 GB/s

HPSS
Archive

Philosophy – "The Mind-Body Problem"

- Really, the "Not Enough Bodies Problem"

- 25% Reduction in SysAdmin and Operations staff over past year

- Aging hardware requires lots of attention, failing 20-30 drives per week

- Varied set of hardware, ldiskfs and ZFS based file systems, some originally started at Lustre 1.6

- Devised a plan to move to single HW vendor all running ZFS within a 6 month time frame
  - Reduces hardware support and repair demands
  - Eliminating ldiskfs reduces scope of testing and developer time

File Systems and Versions

- Last year at LUG, LLNL had file systems made of:
  - DDN 9500 with 48-bay enclosures
  - DDN 9900 with 60-bay enclosures
  - NetApp E5400 60-bay RAID enclosures

- LLNL had the following combinations of Lustre software in production:
  - 2.4 with ZFS
  - 2.4 with ldiskfs
  - 2.1 with ldiskfs and 2.1 clients
  - 2.1 with ldiskfs and 1.8 clients

- Focus on Stability by reducing the variations in hardware and software versions; converge on one platform, with Lustre 2.4 with ZFS backend

- Less variation allows developers to focus on stability and performance, and allows system architects to evaluate new hardware configurations for future deployment

Managing Your Data

- Using Robinhood to scan and track changelogs
  - Haven't had to purge yet
  - Find it more accurate and detailed than quotas
  - Discovered a number of issues (Bugs) when first implementing.
  - Did not work on 1.8 and 2.1 FS not formatted with dirdata
    — Fid2Path errored with ENODATA
  - Major problem scanning and keeping up with the changelogs due to single client MDS performance
    — Schema changes cause a rescan of the file system
    — Heavily loaded file systems fall behind…8 day old data is not useful
  - Looking at ways to improve using a partial scan feature

Monitoring Your Systems

Metrics are only as good as the data you gather

- **Enclosure Monitoring without RAID controllers**
  - Our MDS nodes are pure ZFS with JBOD
  - Using sas2irsu from LSI to monitor drive status

- **LLNL is using Splunk for monitoring and correlation of events**
  - "My job is slow!", "Why did my job die?"
    - Splunk aggregates logs from client and server clusters, so events can be correlated with simple queries and narrow time frames. No need to use "grep" on 6 different clusters.
  - "What happened to the center over the weekend?"
    - Data ingested can be customized to display reports, and graph values. LLNL is ingesting meminfo and arcstats to graph memory and ARC usage to track down OOMs and work on ZFS performance tuning
  - "Things seemed slow last Tuesday…Can you look?"
    - Run IOR regularly from various clusters and send the data into Splunk, so I can graph the results and look for problems
  - Splunk can also send alerts based on saved queries, which we use to track down problems on the IB fabrics.
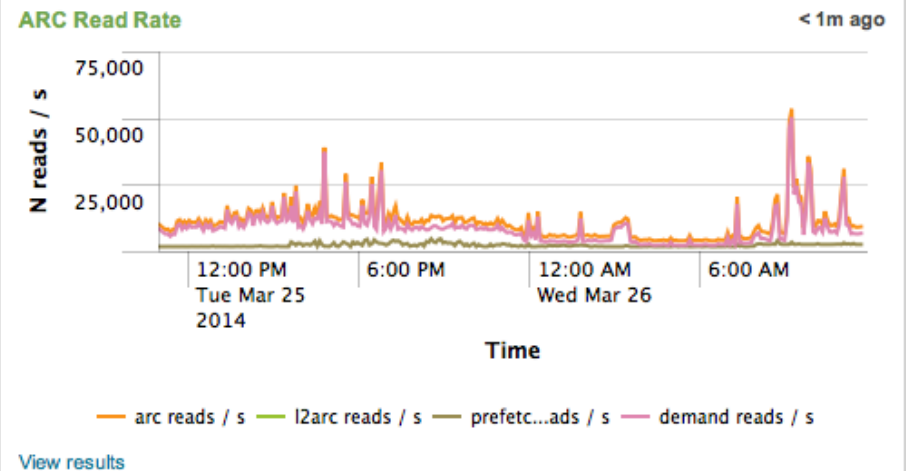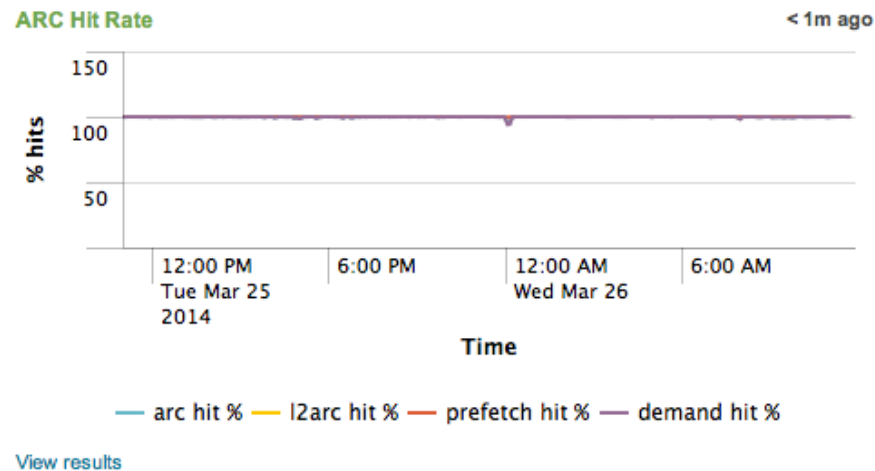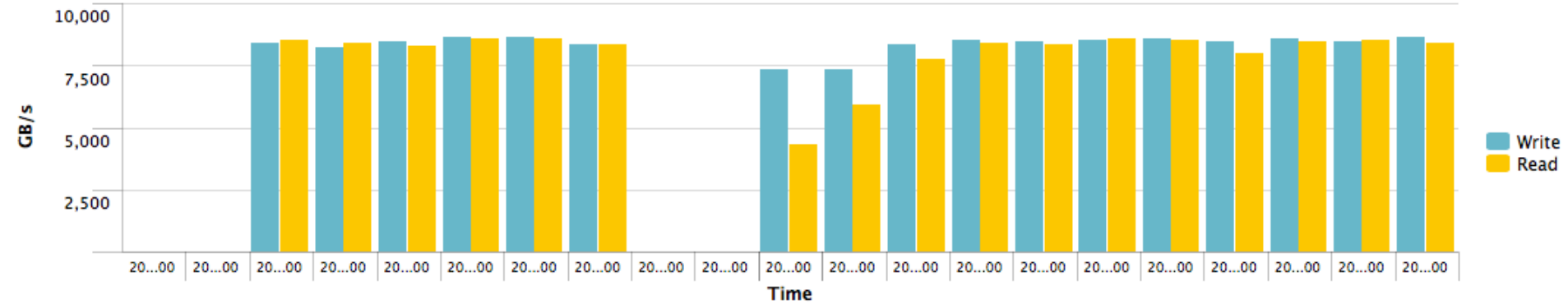
Splunk ARC Stats Graphs

SAS Based MDS
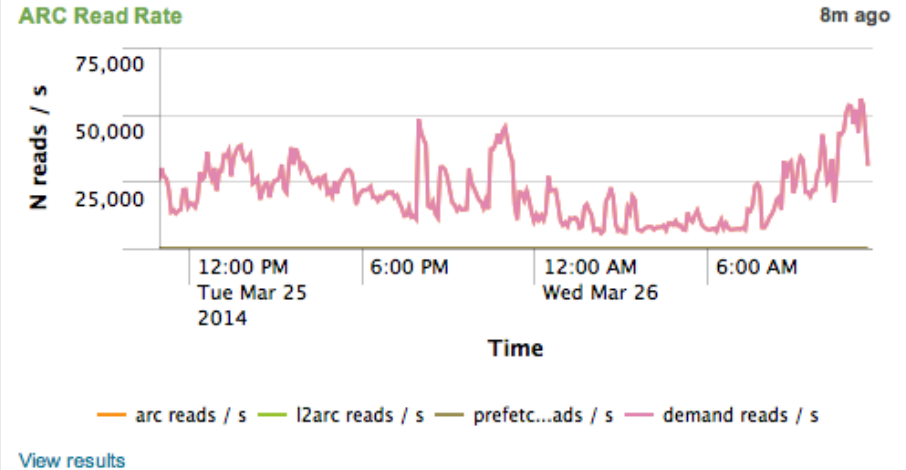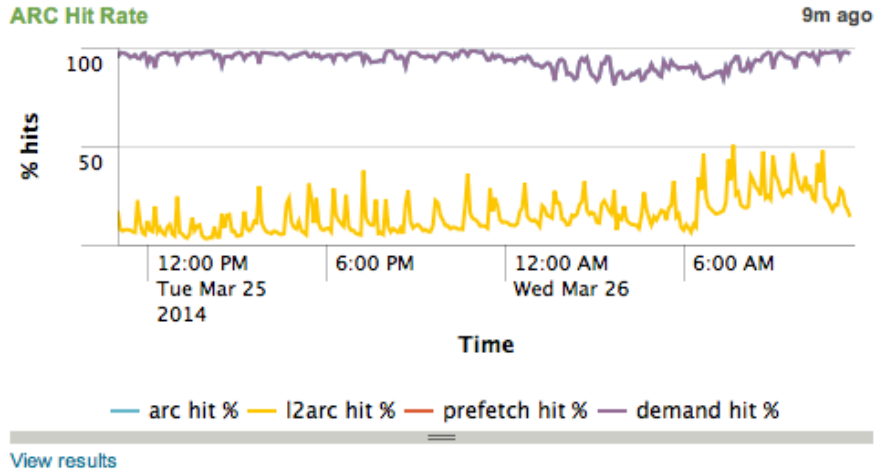
SSD Based MDS

# Splunk IOR Results Graphs
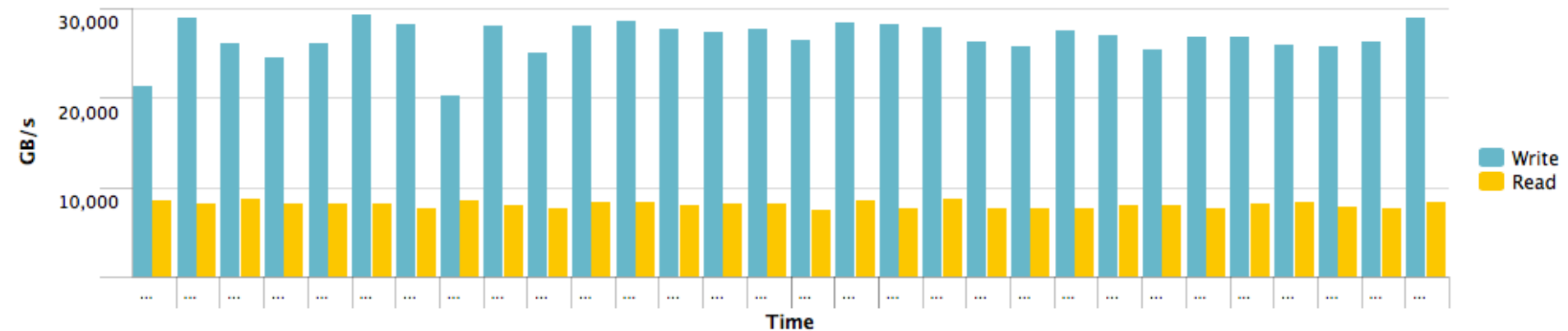
Data Movement

- Last Year, I talked about moving data from one file system to another
  - Poor single client metadata performance
  - lack of parallel tools

- Adam Moody along with others at Oak Ridge, Los Alamos, DDN, etc. collaborated on a project called FileUtils
  - http://fileutils.io
  - Uses libcircle and MPI to do a variety of parallel file operations.
  - Significantly faster than other tools
  - Almost ready for prime time at LLNL

- rsync still our recommended data movement tool
  - Main concern was dcp would overwhelm the MetaData Servers
  - We do plan to work closely with the power users and schedule movements of large data sets

Data Movement Speeds

File System to File system copy,
100,000 1MiB files lscratch4 to lscratch7
March 2014
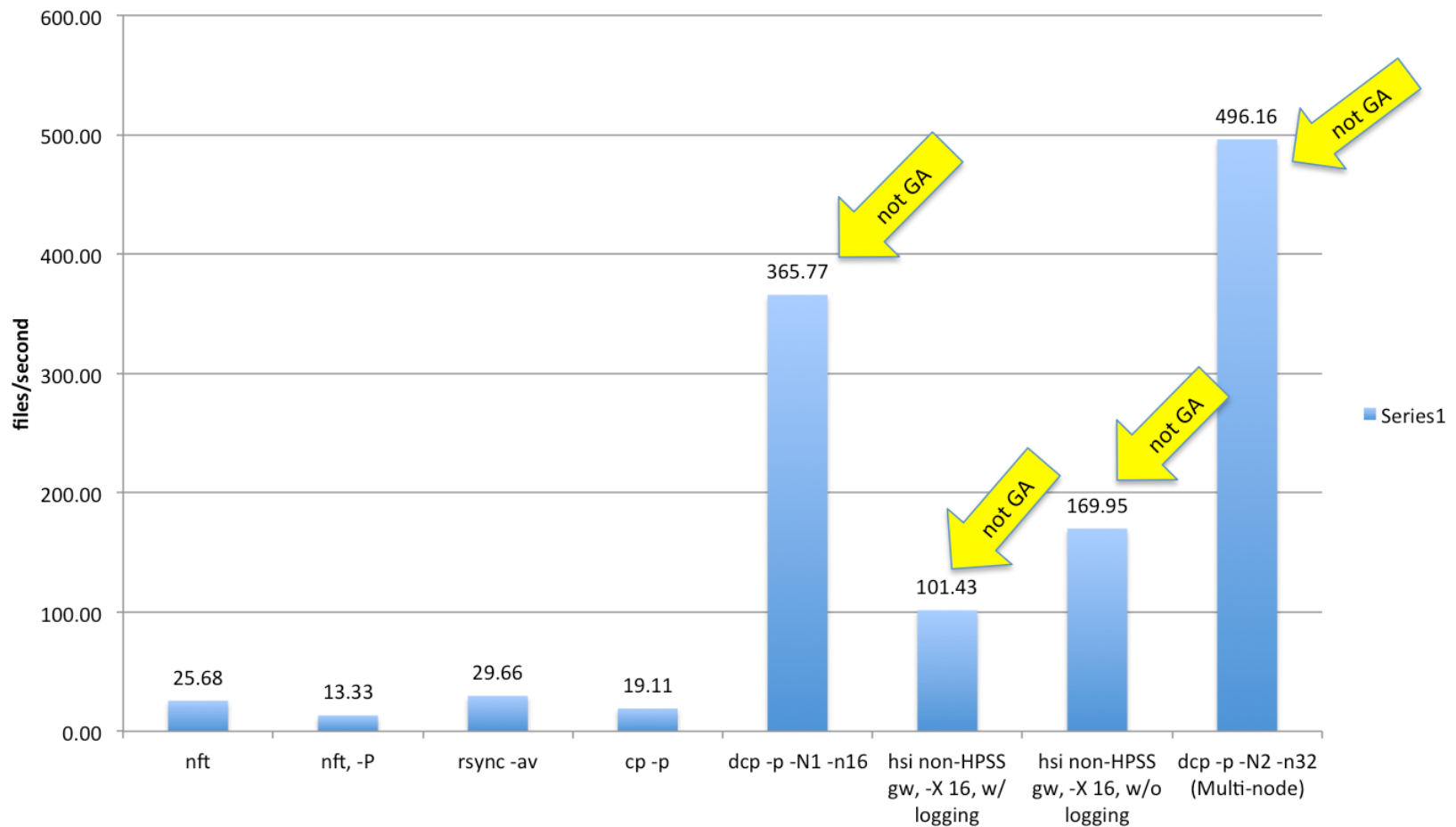
Next Steps

- Continue Experiment with "Islands of Storage"
  - Dedicated file systems vs Global file systems

- Next Generation Hardware
  - ZFS with JBODs is the goal

- ZFS Performance enhancements
  - ZIL work
  - Further ARC tuning and bug fixing

- Better Monitoring
  - RAID controllers tell us when drives fail
  - JBODS have minimal to none monitoring/reporting

- Hardening Fileutils

- DNE vs Multiple file systems

Last Slide

- This slide intentionally left last
  - Seriously, I have nothing left to say
    —I think there is a reception after this…
      – Are you really still reading this?

Marc Stearman
stearman2@llnl.gov