Architecture and Design of Cray DataWarp

May 2016
Benjamin Landsteiner
ben@cray.com

Dave Henseler
Doug Petesch
Cornell Wright
Nicholas J. Wright

CUG2016

Agenda

- Challenges, Observations, and Trends
- Burst buffers
- DataWarp Benefits
- Architecture
  - Hardware
  - Software
- Example job using DataWarp
- Performance
- Summary
- Q&A

Challenges, Observations, and Trends

- **Many programs do I/O in bursts**
  - Read, Compute, Write, Compute, Write, Compute, etc.
- **Want to have high bandwidth when doing I/O**
  - Compute resources largely idle during I/O
- **Disk-based Parallel FileSystem (PFS) bandwidth is expensive**
  - Capacity is cheap
  - PFS do provide lots of capacity, reliability, permanence, etc.
- **SSD bandwidth is (relatively) inexpensive**
- **Large I/O load at beginning and end of job**
- **Cray Aries network is faster, lower latency than PFS network**
  - Or at least shorter distance

Burst Buffer Concepts

- **Burst Buffer (BB) - A high-bandwidth, lower-capacity, "buffer" space, backed by a disk based PFS**
  - Increased BB bandwidth decreases time programs spend on I/O
- **BB can interact with PFS before, during, and after program use**
  - Stage data in to BB before computes allocated
  - Stage data back out to PFS after computes deallocated
  - Stage data in or out, using BB hardware, while program in computational phase
- **BBs offer much greater bandwidth per dollar**
  - So, do I/O to BB and write out to PFS over time

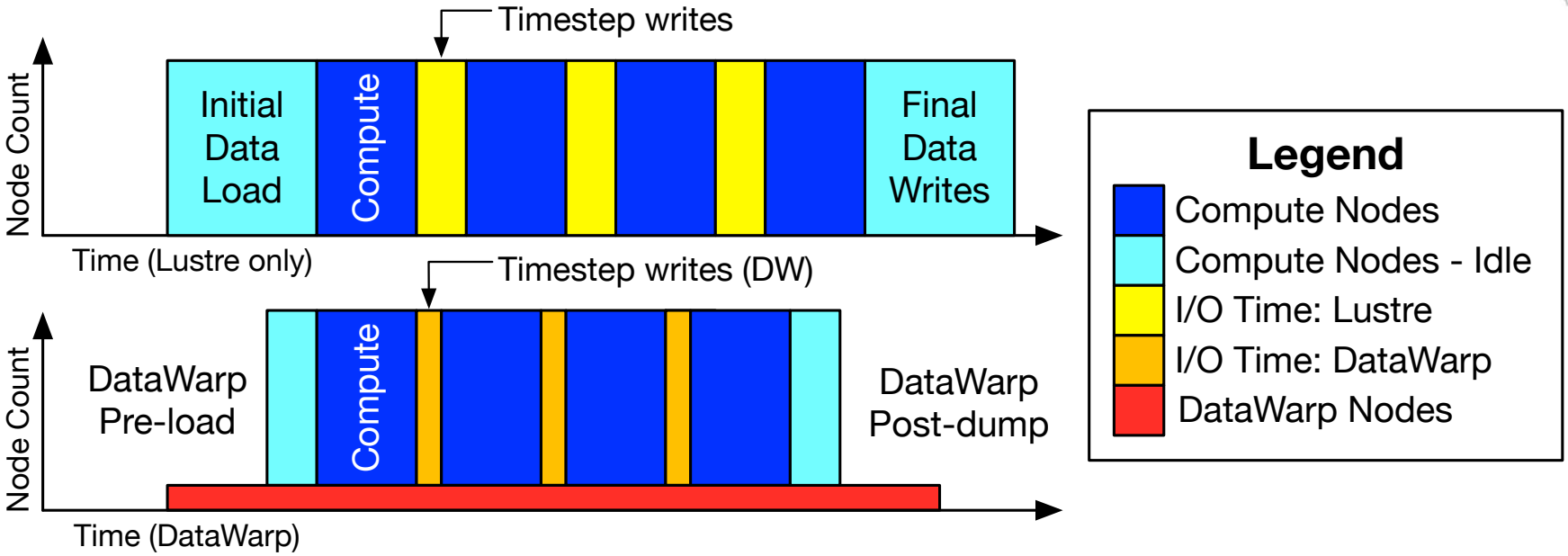
DataWarp

- **DataWarp is a performance accelerator**
- **More than just burst buffers!**
- **Hardware**
  - Nodes of flash storage, CPU, and network
  - Uses high-endurance SSDs optimized for write-intensive workloads
  - Provides high bandwidth directly to the Aries network
- **Software**
  - Software-defined storage that virtualizes the pool of flash memory
  - Dynamically provisions storage to applications
  - Workload manager integration

DataWarp Benefits

- **Reduced total cost of ownership**
  - Reduces the spend on the HDD-based PFS
  - Eliminates islands of node-installed flash
- **Increased productivity**
  - Reduced wallclock time accelerates scientific discovery
  - Scalable administration via automation and policy setting
- **Improved Quality of Service**
  - Eliminates PFS bottlenecks
- **Increased utilization of SSD**
  - Single pool of SSD storage
  - Shared by any/all compute nodes
- **Simplified Management**
  - Simple controls via job script directives
  - Automated provisioning
  - Policy management via WLM

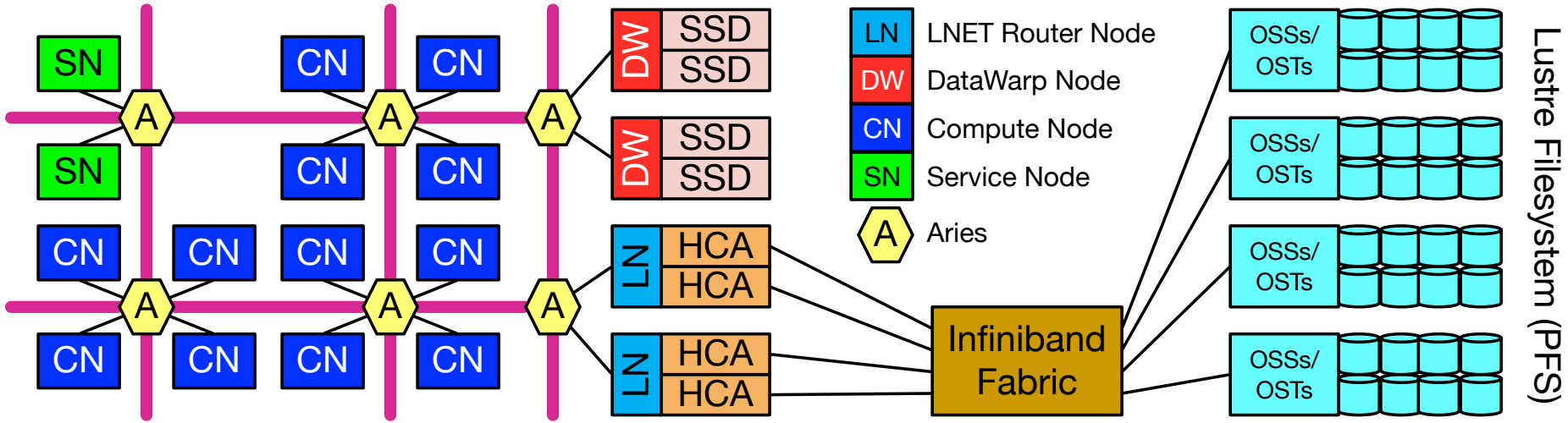
DataWarp - Reduce Compute Residence Time



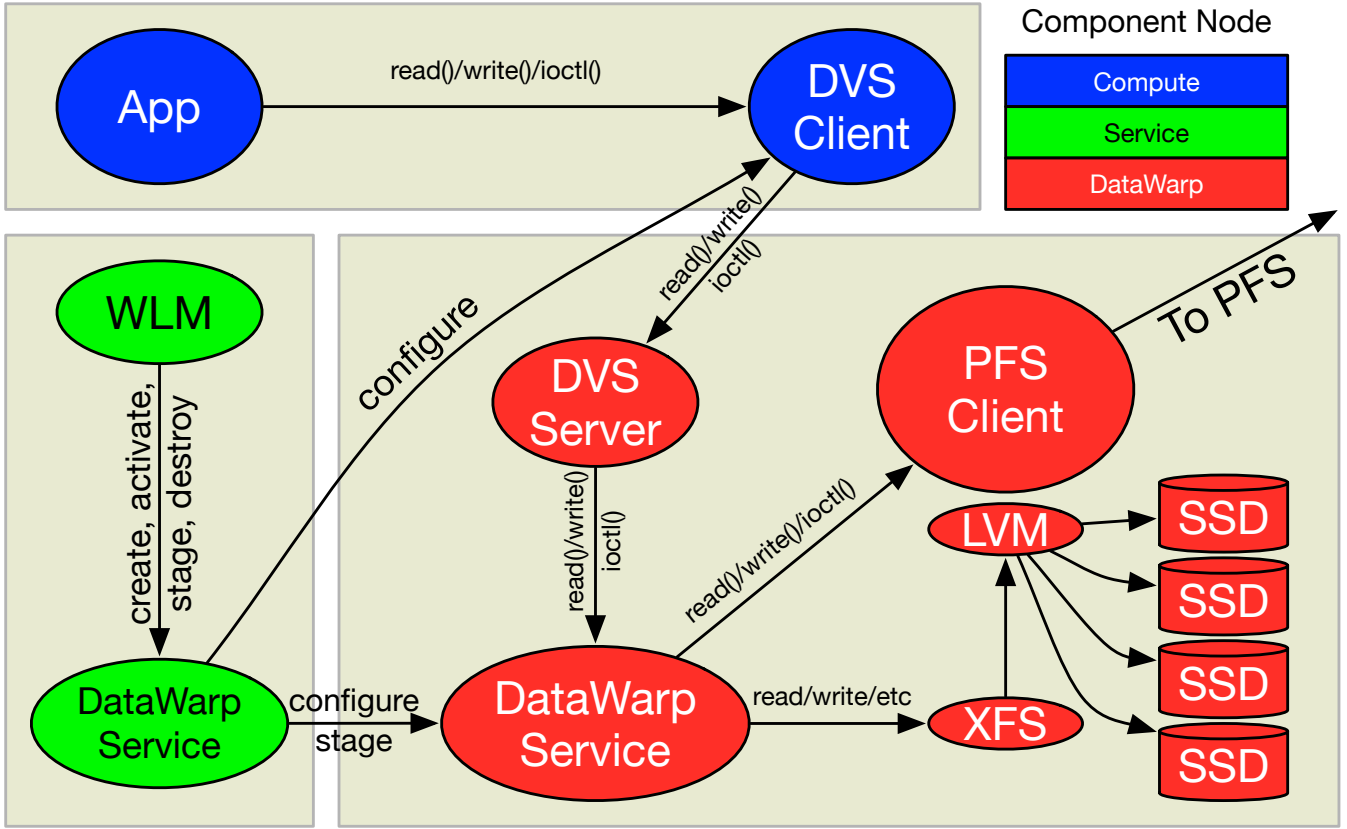
Cray DataWarp Architecture & Design

- High bandwidth SSD devices in service nodes directly attached to Cray Aries network
- Software to weave together all the nodes to create a pool of available space
- Allocation of portions of the space to different users by request, on a job-by-job basis or on a persistent basis
- Provision of a POSIX-compatible filesystem view to users
- Support for asynchronous requests
- Direct stage-in and stage-out from the service nodes to the backing PFS
- Implemented in phases (phase 1 released)

DataWarp Hardware Architecture Overview



DataWarp Software Components

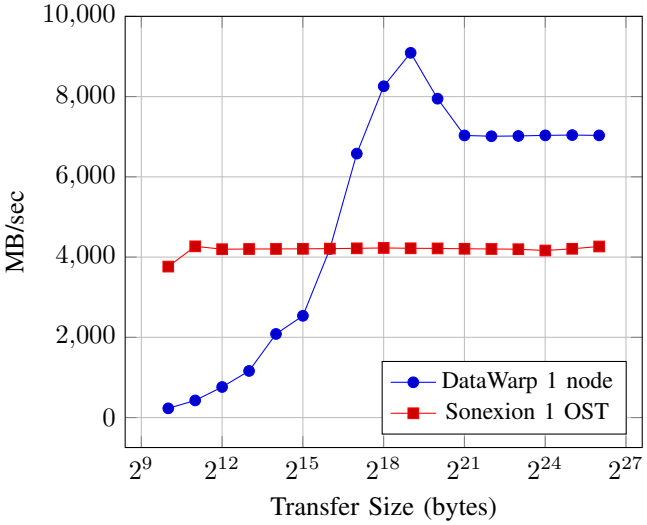


WLM Example - Job Script

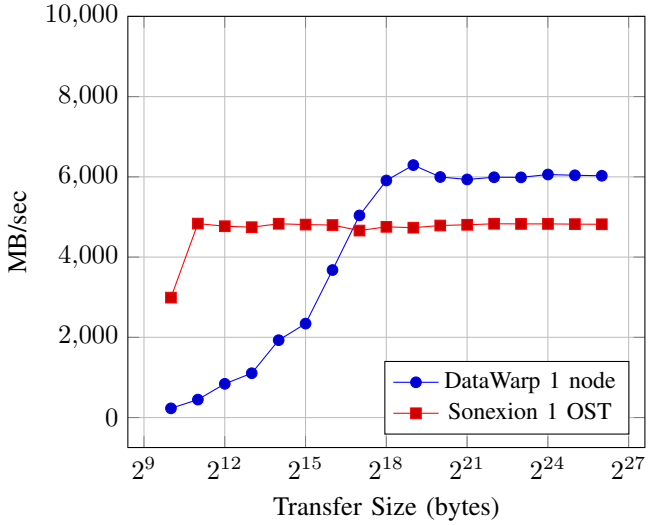
```
#!/bin/sh
#DW jobdw type=scratch access_mode=striped capacity=100TiB
#DW persistentdw name=common_dbs
#DW stage_in type=file source=/pfs/user/input \
                       destination=$DW_JOB_STRIPED/input
#DW stage_out type=directory source=$DW_JOB_STRIPED/results/ \
                            destination=/pfs/user/results/

aprun -n 5000 a.out \
    --database=$DW_PERSISTENT_STRIPED_common_dbs/abc \
    --parameter-file=$DW_JOB_STRIPED/input \
    --resultsdir=$DW_JOB_STRIPED/results
```

Performance - Maximum Sequential read/write



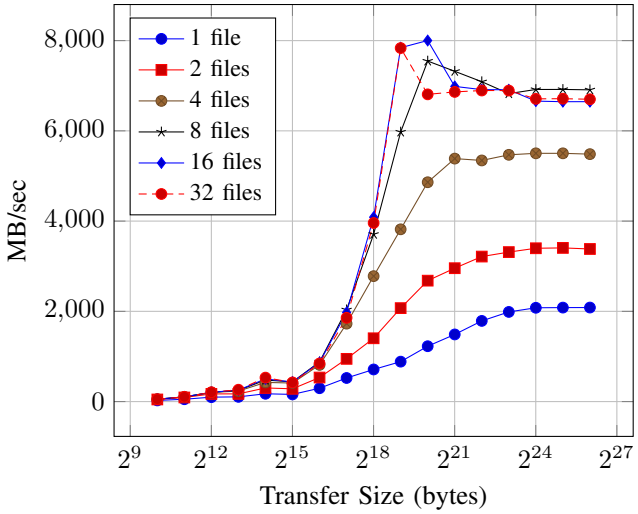
Read



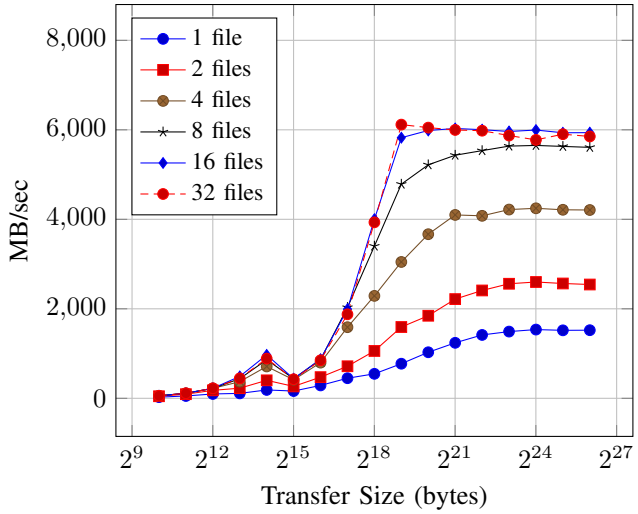
Write

- **Best performance at 2^{19} bytes (512KiB)**
- **For small transfers, Lustre exceeds DataWarp performance**
  - DVS does not have client-side caching - yet
- **DataWarp has the potential to greatly exceed lustre performance**

Performance - Saturating a node



Read

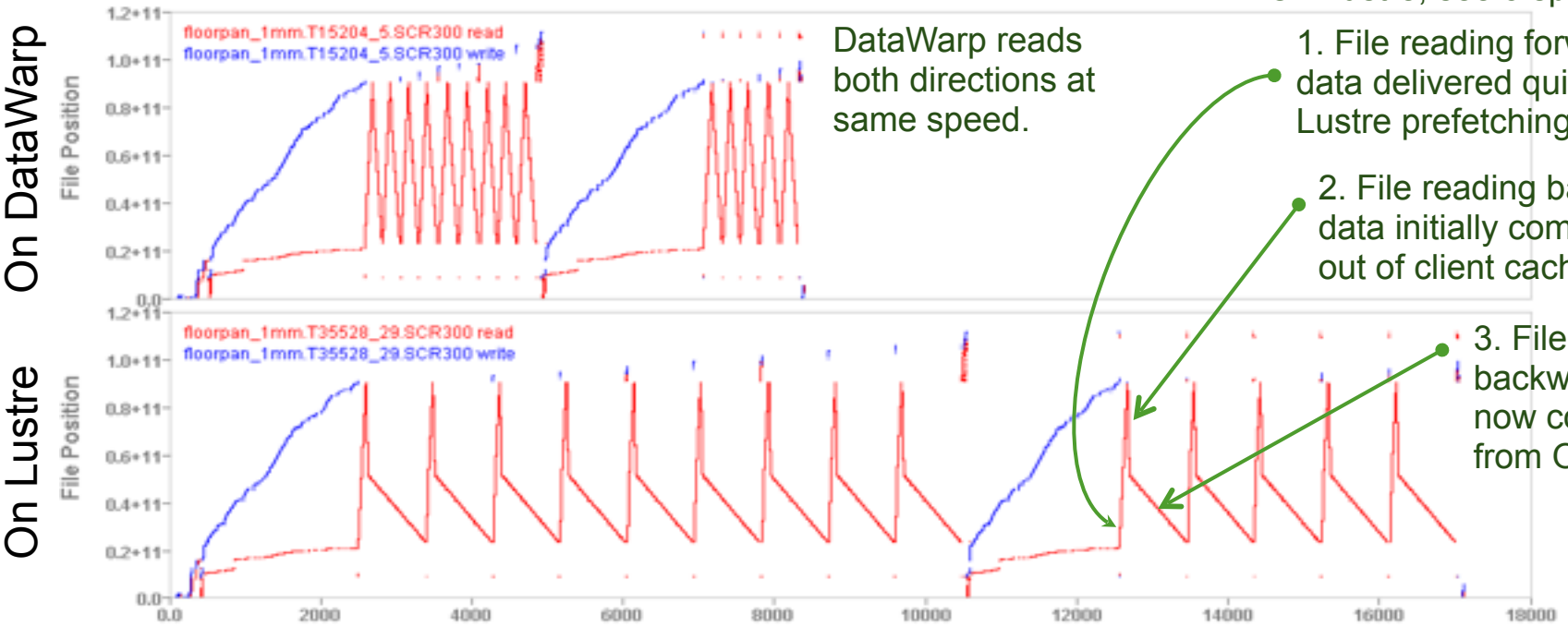


Write

- **Single compute node, single DataWarp node**
- **Read saturation at ~8 streams**
  - Bottleneck: Aries
- **Write saturation at ~16 streams**
  - Bottleneck: SSDs

Nastran Example – Forward/Backward Reads

File position (left) vs Time (bottom)



DataWarp reads both directions at same speed.

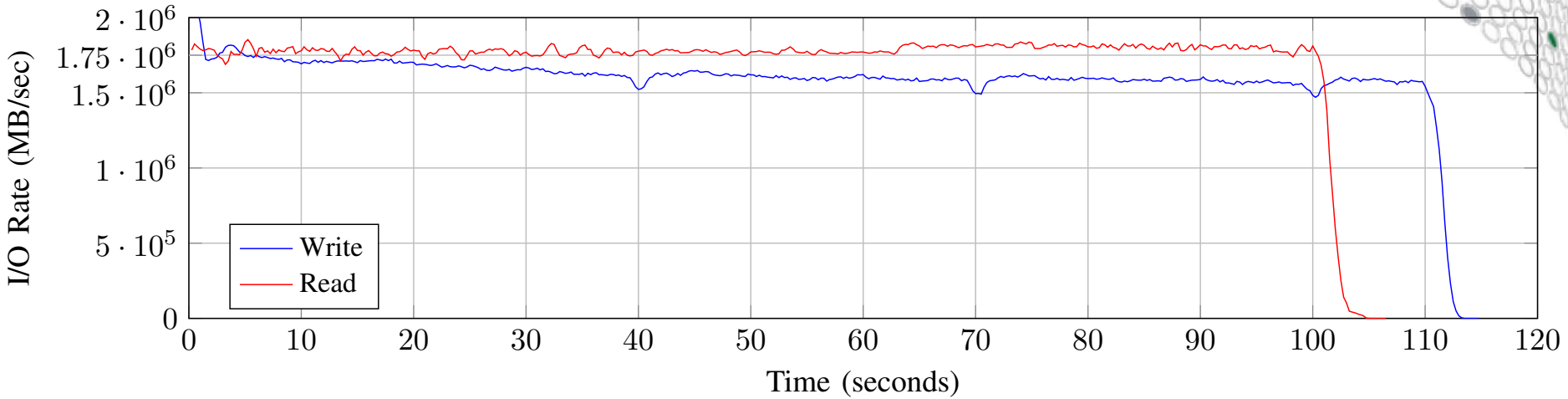
On Lustre, see 3 speeds:

1. File reading forwards, data delivered quickly using Lustre prefetching
2. File reading backwards, data initially comes quickly out of client cache
3. File still reading backwards, data now comes slowly from OSTs

Lustre job takes twice as long.

I/O activity in the SCR300 file, showing the forward and backward passes of reading the factored matrix.

IOR: 1.66TB/sec read, 1.54TB/sec write



- 264 DW nodes
- Intel P3608 SSDs
- 5280 compute nodes

- IOR POSIX FPP
- 16GiB per file
- 2 ranks per node
- 512KiB transfer size

- CLE 5.2.UP04
- type=scratch
- access_mode=stripe

Summary

- Fast SSDs accessible over Aries network allow for a big jump in I/O bandwidth
- DataWarp can be provisioned for bandwidth, PFS can be provisioned for capacity and resilience
- Workload Manager integration enables jobs to request DataWarp
- Bandwidth scales with number of DataWarp nodes

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publicly announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, REVEAL, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

Q&A

- Benjamin Landsteiner
- ben@cray.com

- Works seamlessly with the existing scientific applications
- Integrates with the existing environment
- Dynamic and flexible through virtualization
- Many practical uses for a broad range of scientific applications