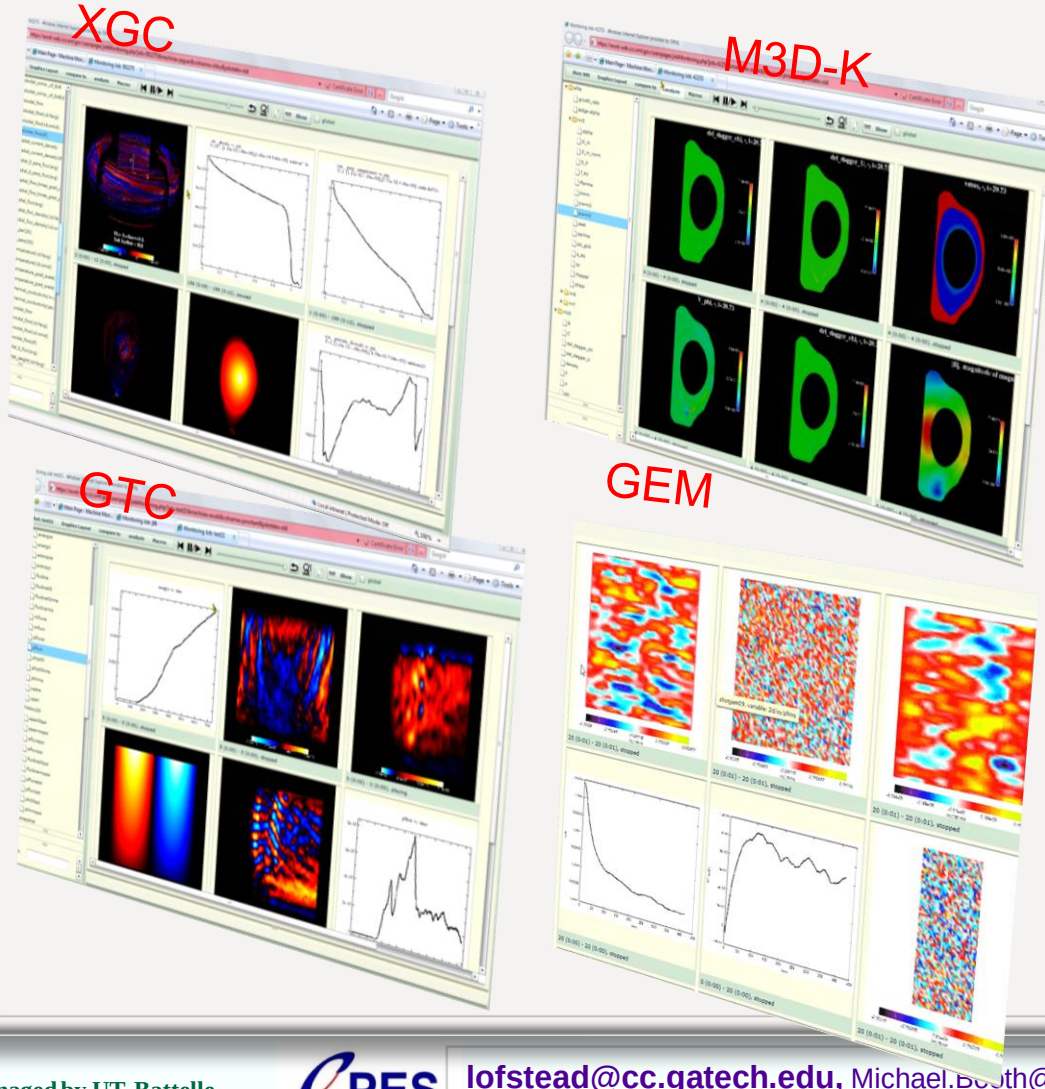
Petascale IO Using The Adaptable IO System (ADIOS)



CUG 2009

5/6/2009

Scott A. Klasky

klasky@ornl.gov

Jay Lofstead

Hasan Abbasi

Michael Booth

Fang Zheng

Karsten Schwan

Matthew Wolf

Overview

- New challenges from Jaguar upgrade
- Solution architecture
- Scheduled IO
- Aggregation IO
- Conclusion

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu,
fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

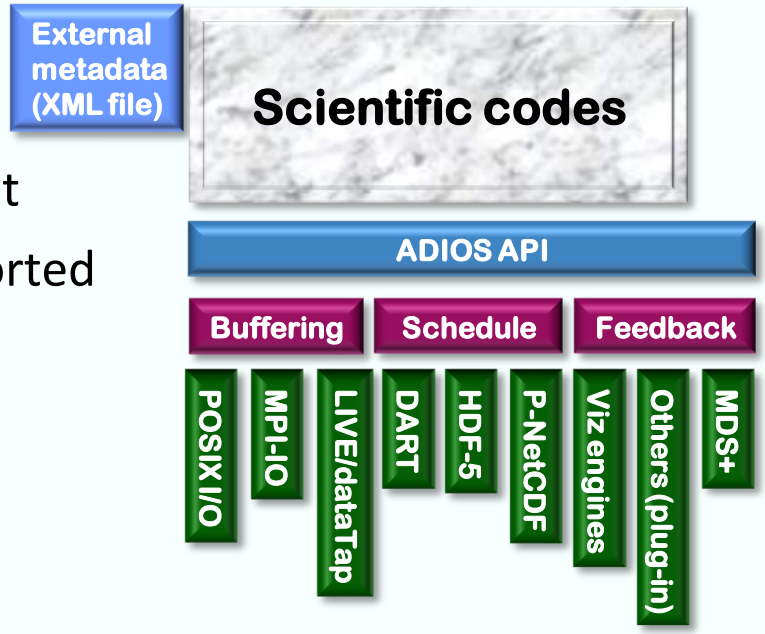
New Challenges

- Jaguar (petascale partition) is BIG
  - 149,736 cores (8 cores per node)
  - 672 Lustre storage targets (10 PB storage)
  - 2 GB per core (300+ TB RAM)
- Issues
  - Compute-to-storage ratio large
  - Lustre limitations being exposed
    - 160 OSTs per file maximum
  - Rotational media sharing performance penalties

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

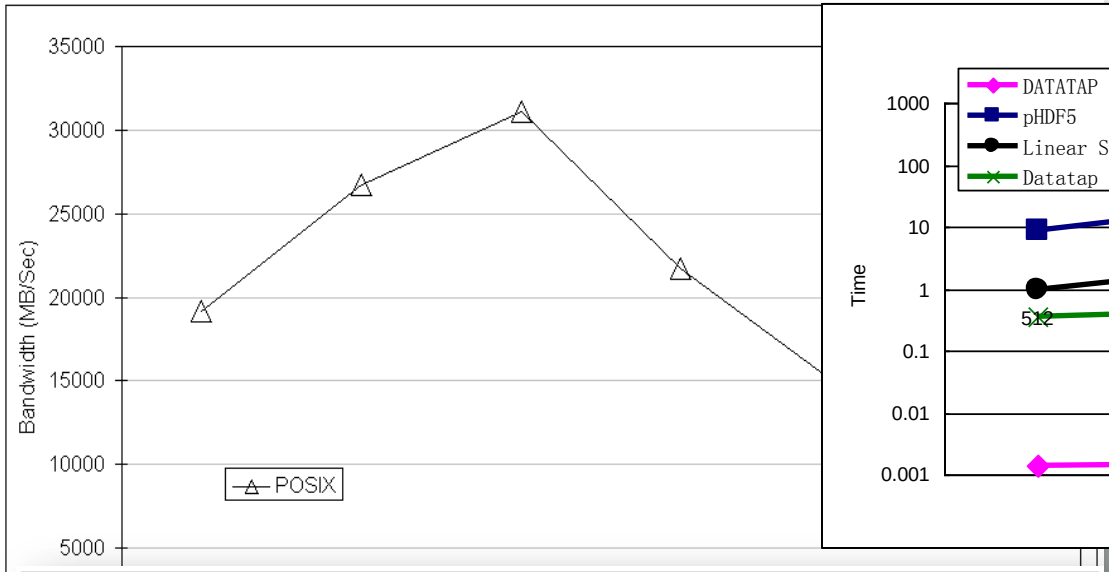
Solution architecture: Adaptable I/O System

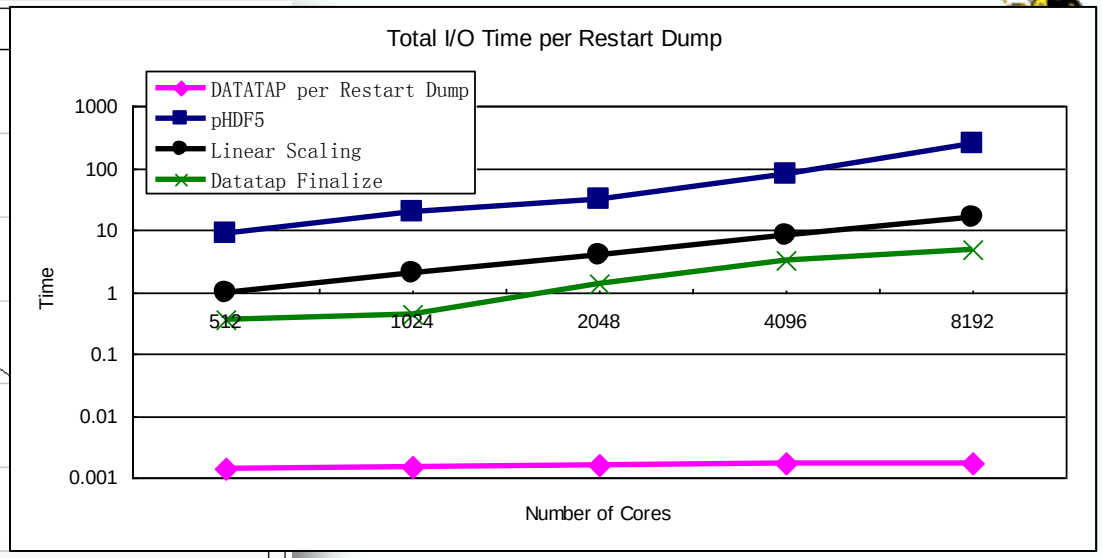
- Overview
  - Allows plug-ins for different I/O implementations
  - Abstracts the API from the method used for I/O
- Simple API, almost as easy as F90 write statement
- Synchronous and asynchronous transports supported with no code changes
- Componentization
  - No need to worry about I/O implementation
  - Components for I/O transport methods, buffering, scheduling, and eventually feedback mechanisms
- Change I/O method by changing XML file only.
- ADIOS buffers data.
- ADIOS allows multiple transport methods per group
- ADIOS contains a new file format (BP) for "optimal" performance
- Make custom transport for new machine!

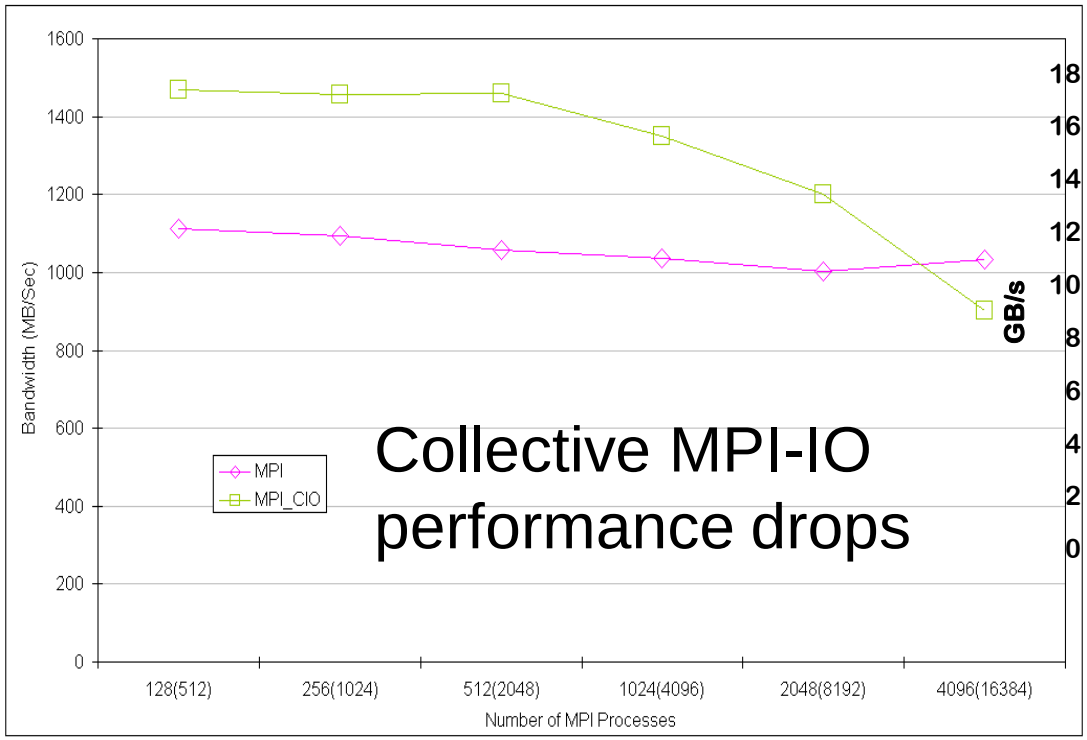


lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

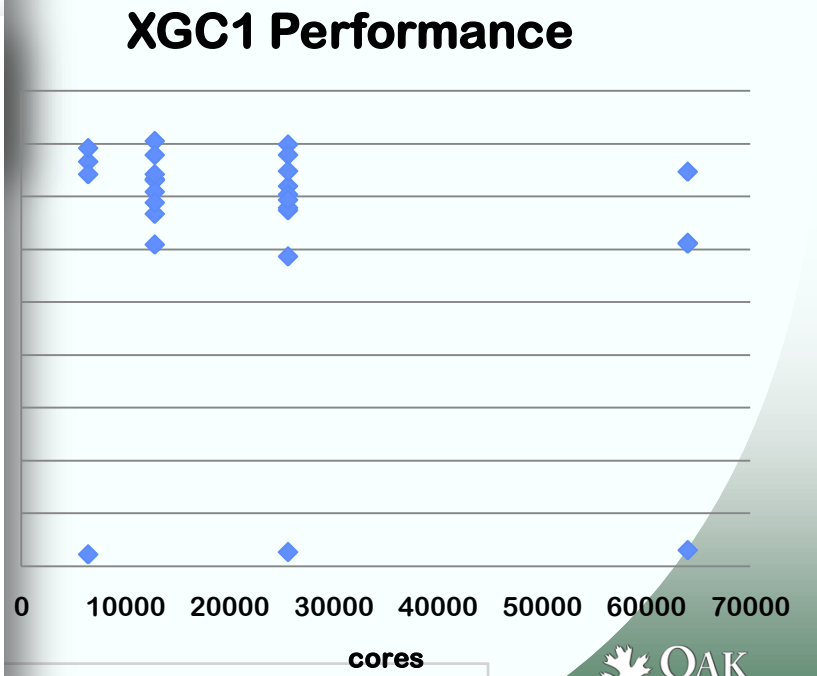
Previous results with ‘old’ methods







Collective MPI-IO performance drops



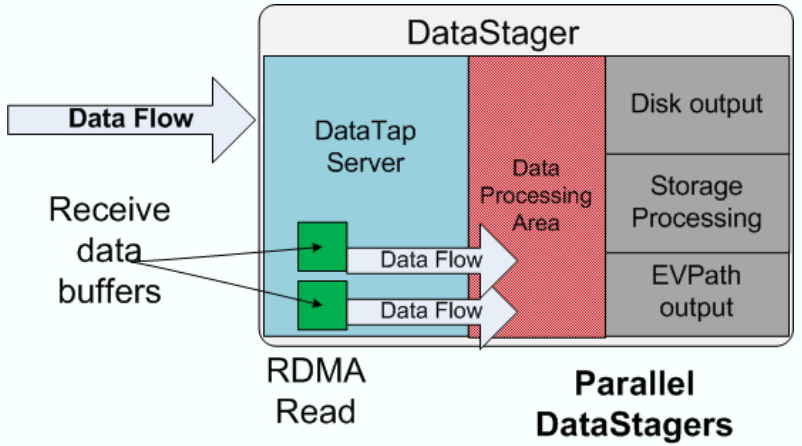
@Sun.COM, hasan@gatech.edu,
.edu, schwan@cc.gatech.edu

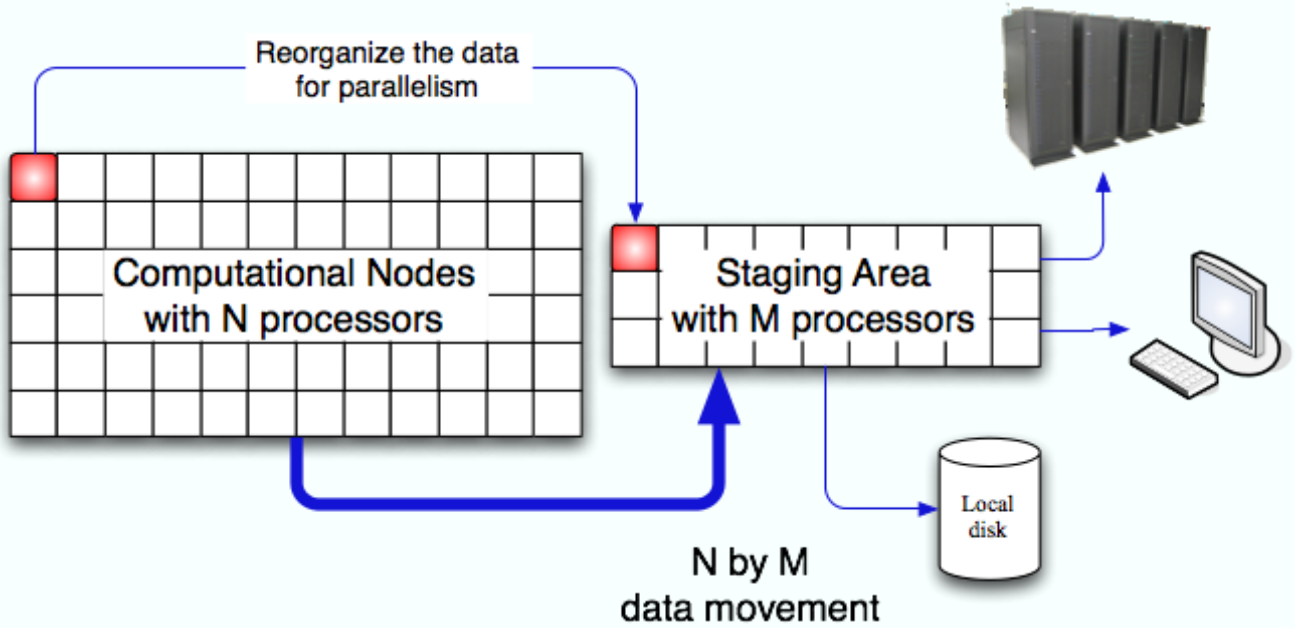
SDM CENTER

OAK RIDGE National Laboratory

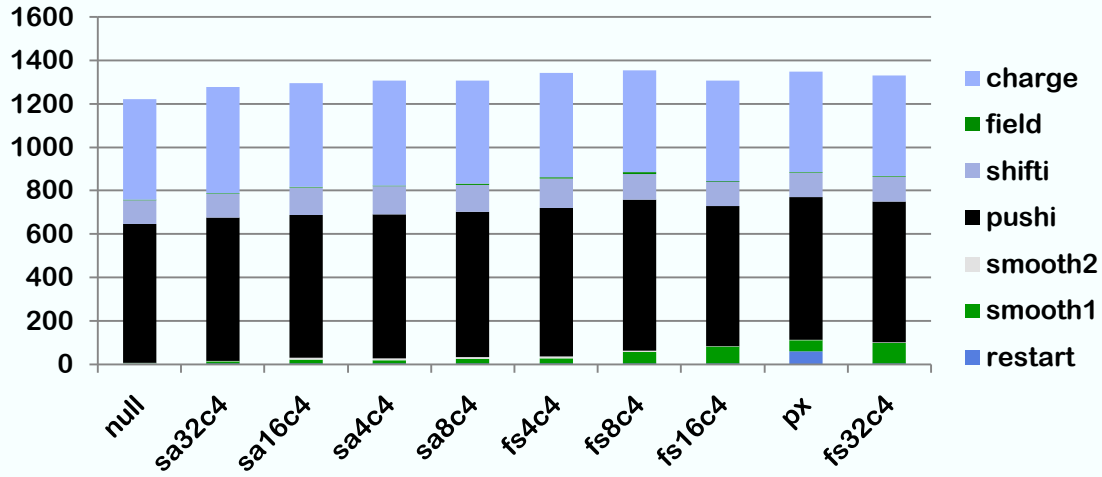
Staging Area

- Additional resources for buffering before storage
- Simple operations like aggregation
- Complex analysis and compression operations
- Domain specific services
  - Combination of extraction, processing and storage
  - Placement to optimize performance

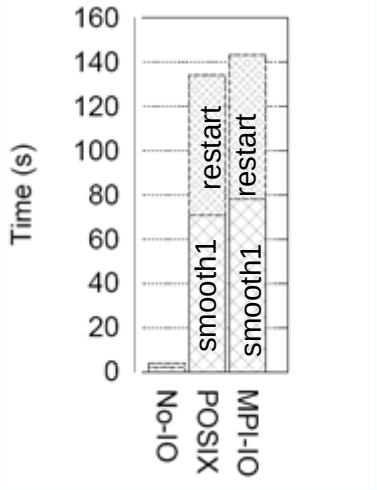




Asynchronous I/O effects code performance



- Null = no io
- px = posix
- **saXcY** = state aware with X number of staging nodes,Y level of concurrency for requests.
- **fsXcY** = continuous drain: with X number of staging nodes,Y level of concurrency for requests.



- Interesting result: Synchronous I/O is also effected.
- Function for smooth1 (lots of all to 1) is greatly effected.
- Why? Lustre dirty cache

Output Method	Total Runtime (s)	Total Performance Penalty (%)
No IO	1547.34	0
Fortran Binary Write	1722.25	11.3
Unmanaged Stream	1703.60	10.1
Managed Stream	1646.09	6.4

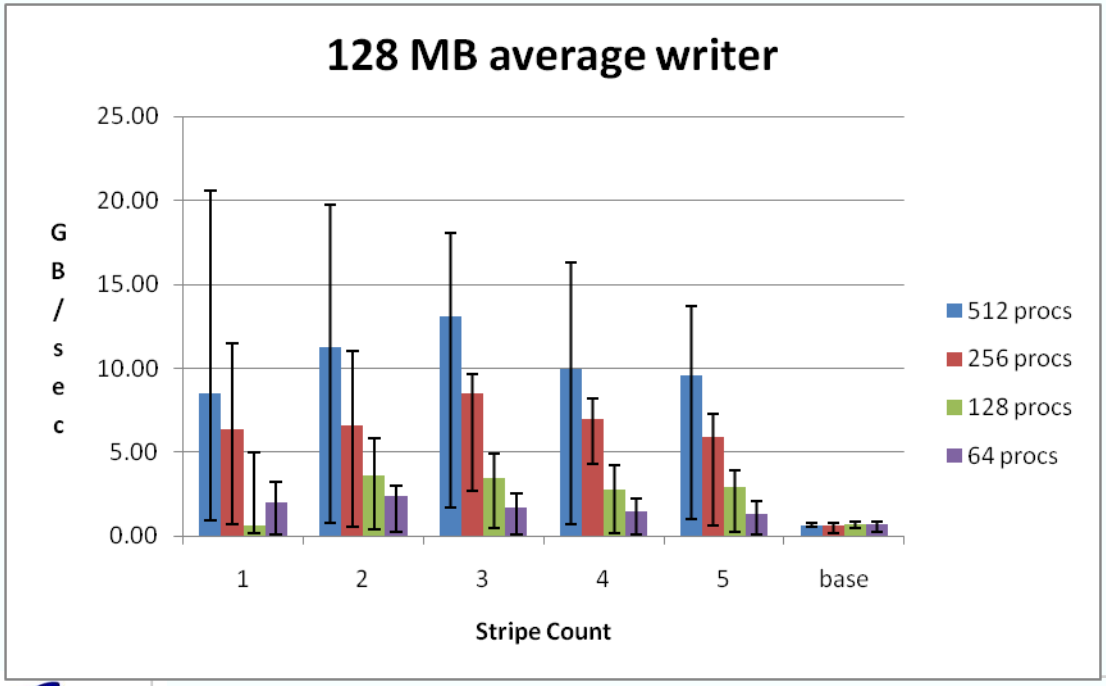
CPES Center for Plasma Edge Simulation
lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu
SDM CENTER
OAK RIDGE National Laboratory

Scheduled IO

- Goals:
  - Reduce concurrent access to storage
  - Minimize data movement
  - Increase use of storage system
- Approach:
  - Split output – use as many of the 672 storage targets as makes sense
  - Schedule metadata – partially serialize open calls to reduce metadata contention
  - Schedule IO – use ‘token passing’ approach for triggering IO for a process
- Examine both staging and direct IO

Scheduled IO (staging small)

- DataTap asynchronous transport method (< 1% node overhead)
  - Generally 512 cores or less additional
- 128 MB per process (weak scaling)



CPES Center for Plasma Edge Simulation
lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu
SDM CENTER
OAK RIDGE National Laboratory

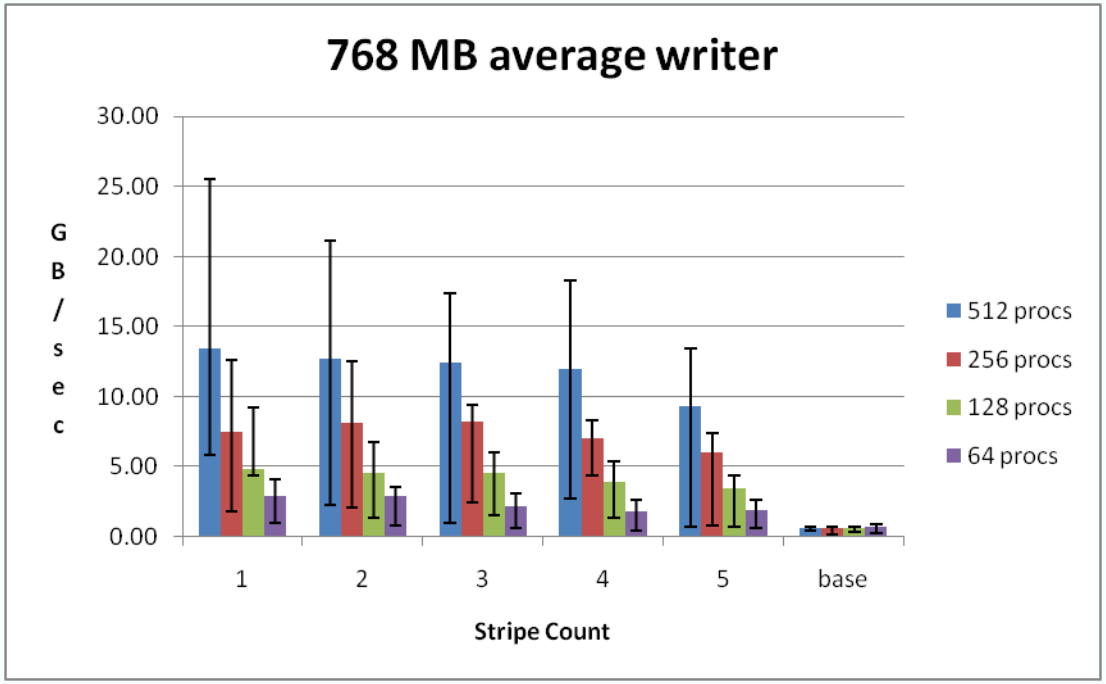
Scheduled IO (staging small)

- A few ‘bad egg’ processes spoil performance
- Stripe count of 1 best overall, 3 best on average
- Lots of files generated based on stripe count

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Scheduled IO (staging large)

- DataTap asynchronous transport method (< 1% node overhead)
  - Generally 512 cores or less additional
- 768 MB per process (weak scaling)



CPES Center for Plasma Edge Simulation
lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu
SDM CENTER

Scheduled IO (staging large)

- Fewer ‘bad egg’ processes
- Stripe count 1 best overall, average
- Lots of files still

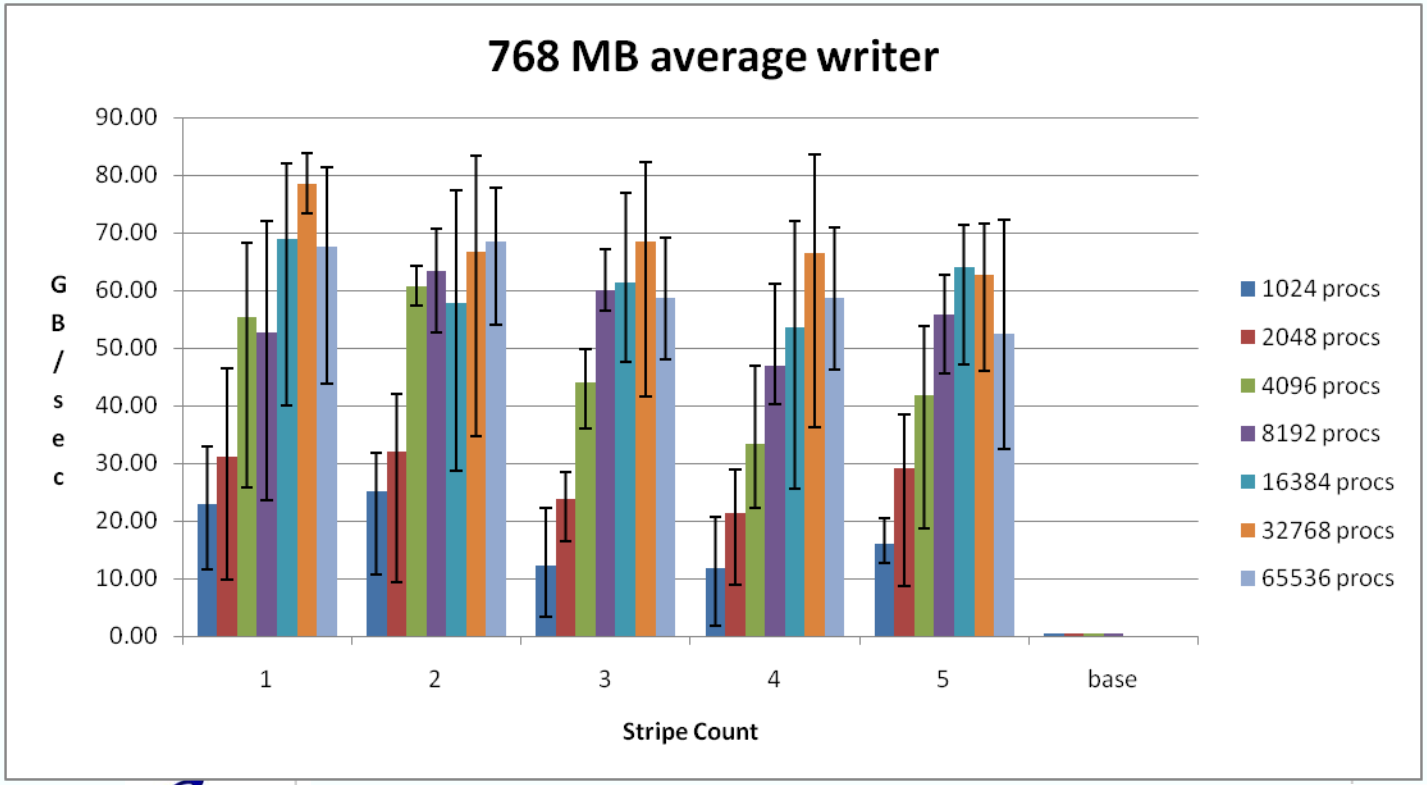
- Staging is not the best approach for fast writes!

Scheduled IO (direct IO)

- MPI-IO Independent
- Serialized MPI_Open calls
- Stripe size set to maximum written from a process
- Writer offset set to stripe boundaries

Scheduled IO (direct IO)

- No additional node overhead (weak scaling)
- Using MPI_File_write
- Performance includes open, close, index



lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Scheduled IO (direct IO)

- 32K cores (32768) yielded best overall performance
  - 24 TB total data, < 3 minutes
- 64K cores seems to be slightly slower

- Lots of writers taking turns yields excellent performance (75% of peak)!

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

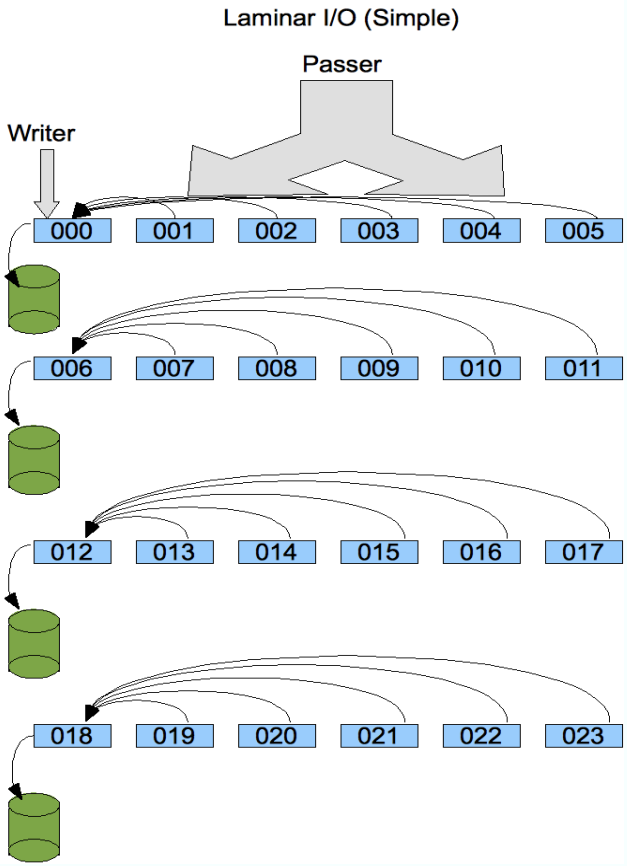
Aggregation IO

- Using the same network paths may gain advantage from fewer connection creations
- Do ‘simple’ and ‘brigade’ aggregation (explained shortly)
- Additional potential impact from lots of data movement not directly to storage

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Aggregation (simple)

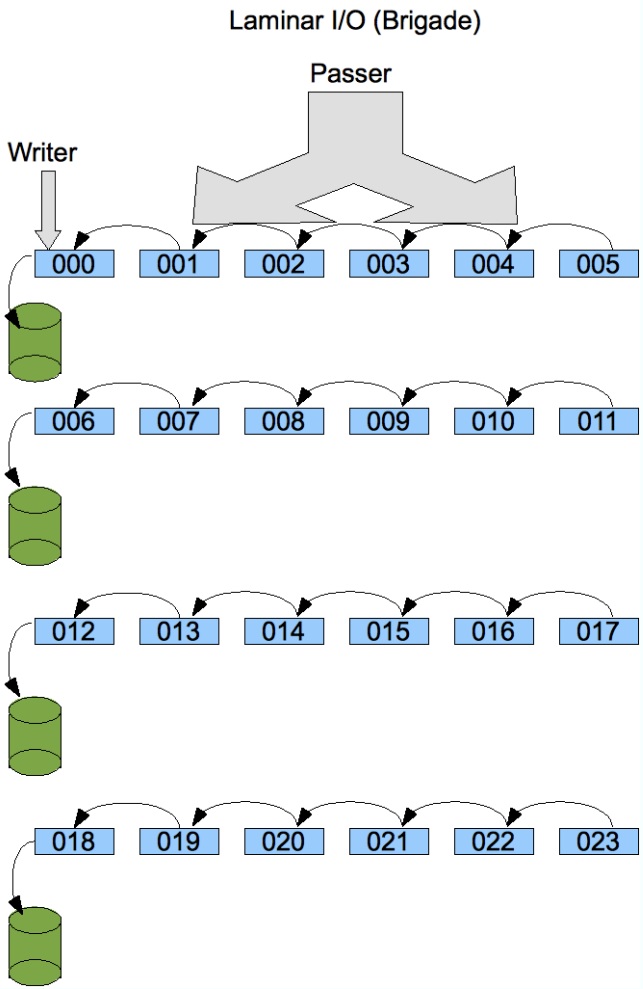
- Split procs into groups for OSTs
- Each sends data to master for that file
- Only root for file writes



CPES Center for Plasma Edge Simulation
lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu
SDM CENTER
OAK RIDGE National Laboratory

Aggregation (brigade)

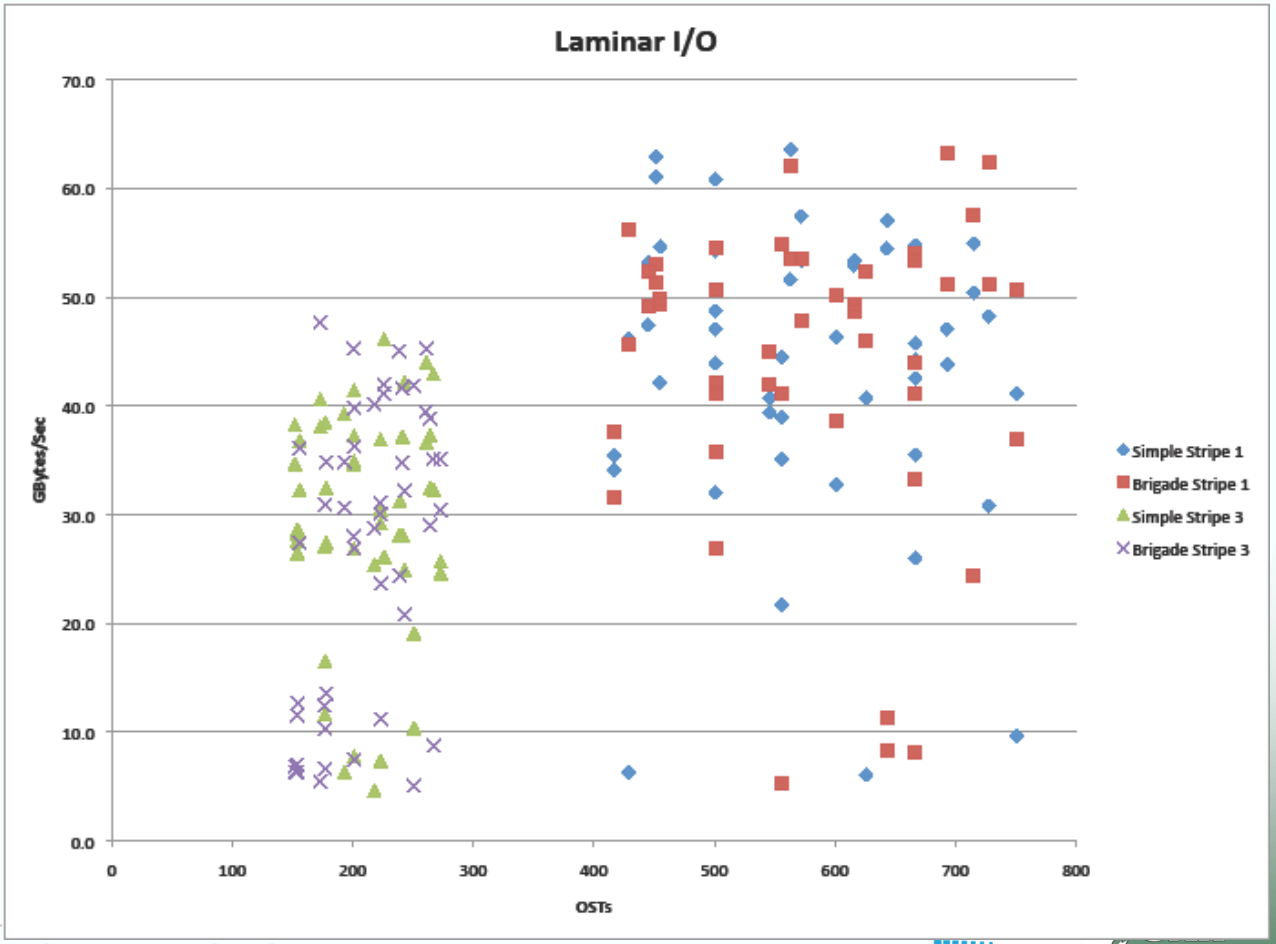
- Split procs into groups for OSTs
- Each sends to previous rank until all data makes it to root for file
- Only root for file writes



lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu,
fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Aggregation (evaluation)

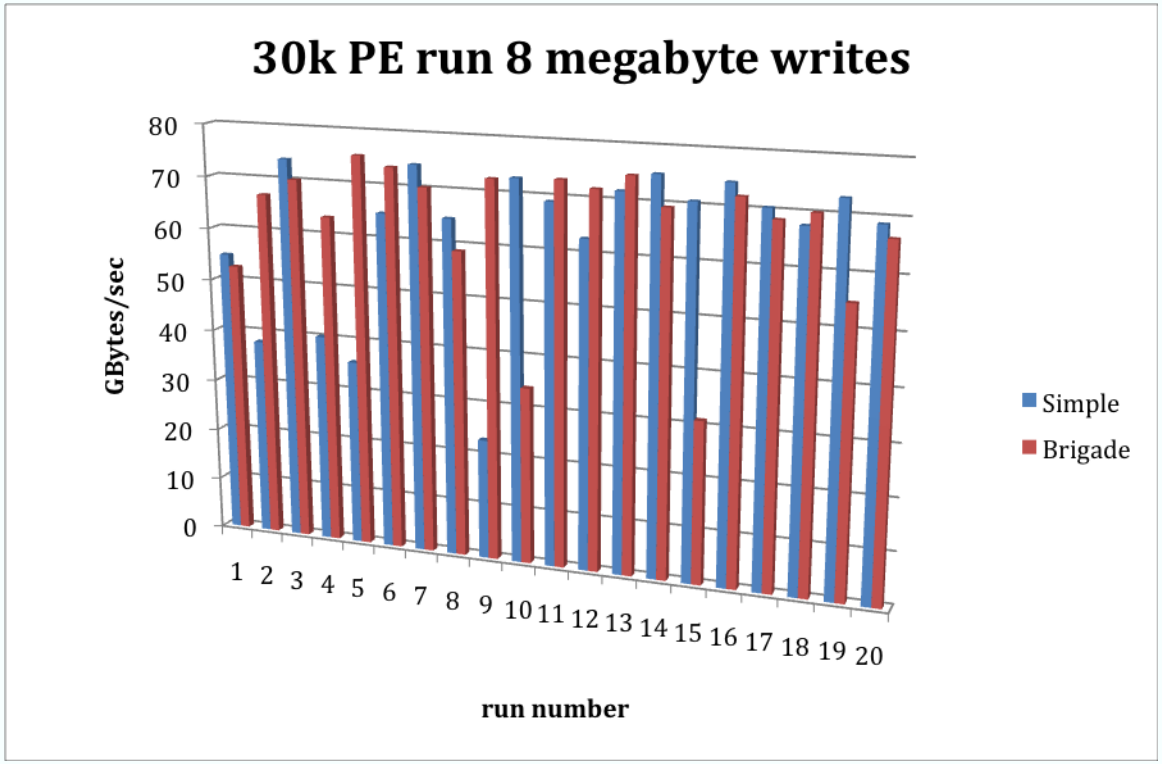
- 8 MB per process; 10,000 processes
- Stripe count of 1 better than 3
- Simple & Brigade about equivalent



lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Aggregation (evaluation)

- 8 MB per process; 30,000 processes; stripe count 1
- Brigade more consistent
  - not necessarily better



lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Conclusion

- Scheduled IO and Aggregation IO both good approaches
- Both are FAR better than default, single file IO
- Reducing OST contention yields performance gains
- Achieving large percentage of IO peak realistic (75% peak attained)

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu,
fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu

Questions?

- Funding provided by
  - ORNL
  - Sandia Labs LDRD
  - Lustre Center of Excellence
  - SciDAC GPSC
  - SciDAC CPES
  - SciDAC SDM
  - SciDAC GSEP
  - SciDAC SAPP SDM for Fusion.

lofstead@cc.gatech.edu, Michael.Booth@Sun.COM, hasan@gatech.edu, fang.zheng@gatech.edu,, mwolf@cc.gatech.edu, schwan@cc.gatech.edu