Improving Parallel I/O for Scientific Applications on Jaguar

Weikuan Yu, Jeffrey Vetter

November 28, 2007

Future Technologies Group ornl

Managed by UT-Battelle
for the Department of Energy

OAK RIDGE National Laboratory

Outline

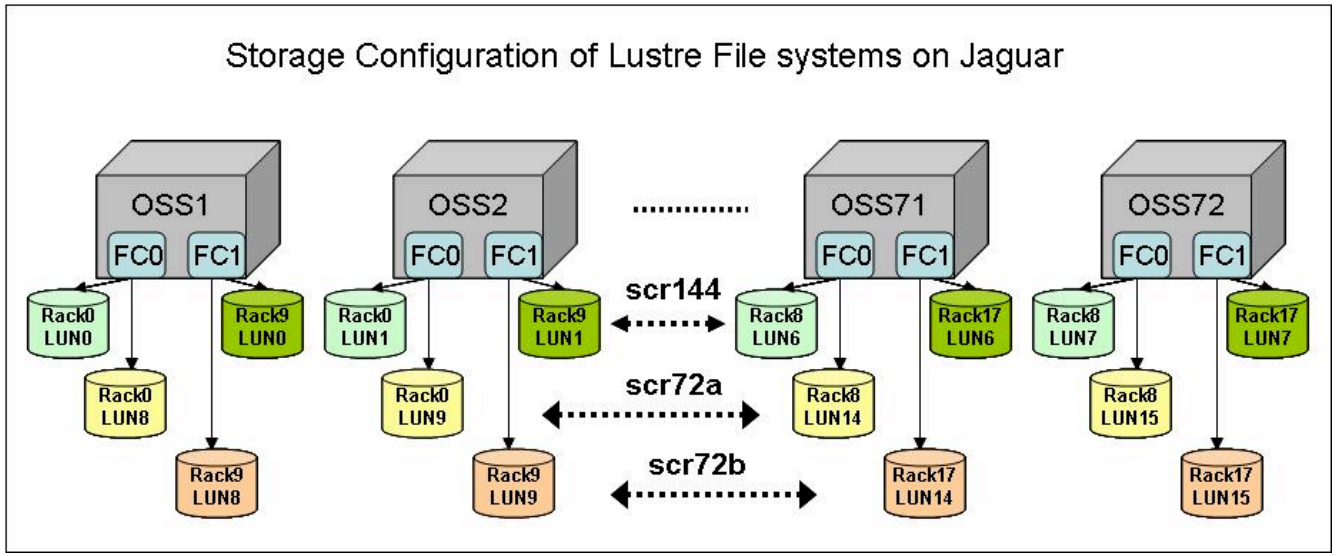
- Overview of Jaguar and its I/O Subsystem
- Characterization, profiling and tuning
  - Parallel I/O
  - Storage Orchestration
  - Server-based client I/O statistics in Lustre file system
- Optimize parallel I/O over Jaguar
  - OPAL: Opportunistic and Adaptive MPI-IO library over Lustre
  - Arbitrary striping pattern per MPI File; stripe aligned I/O accesses
  - Direct I/O and Lockless I/O
  - Hierarchical Striping
  - Partitioned Collective I/O
- Conclusion & Future Work

Parallel I/O in Scientific Applications on Jaguar

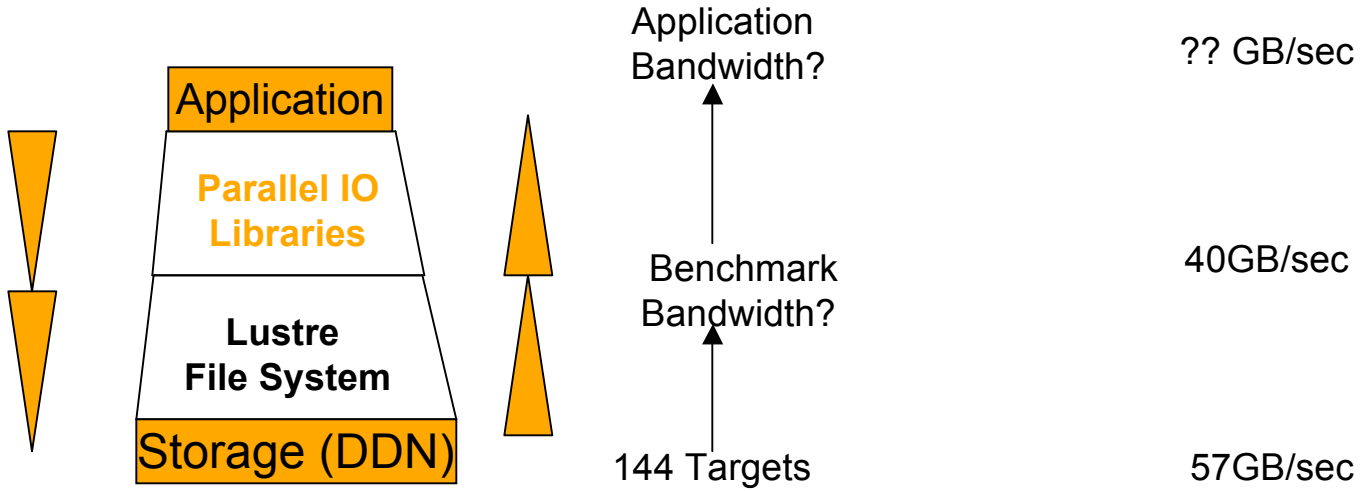
- Earlier analysis of scientific codes running at ORNL/LCF
  - Seldom make use of MPI-IO
  - Little use of high-level I/O middleware: NetCDF or HDF5 …
  - Large variance in the performance of parallel I/O using different software stacks
- Current scenarios faced for scientific codes
  - Application-specific I/O tuning and optimization, such as decomposition, aggregation, and buffering
  - Direct use of POSIX I/O interface, hoping the best from file systems
- Our Effort:
  - Obtain a good understanding of the parallel I/O behavior on Jaguar
  - Promote the use of I/O library, e.g. MPI-IO
  - Provide an open-source and feature-rich MPI-IO implementation
    - Avoid redundant efforts from different applications

ORNL Jaguar I/O Subsystem

- Storage (DDN 9550)
  - 18 racks, each of 36 TB; Fibre Channel (FC) links.
  - Each LUN (2TB) spans two tiers of disks
- Three Lustre file systems
  - Each with its own MDS;
  - Two with 72 OSTs, each of 150TB; One with 144 OSTs, 300 TB
  - 72 service nodes for OSSs, each supporting 4 OSTs



Parallel I/O over Jaguar

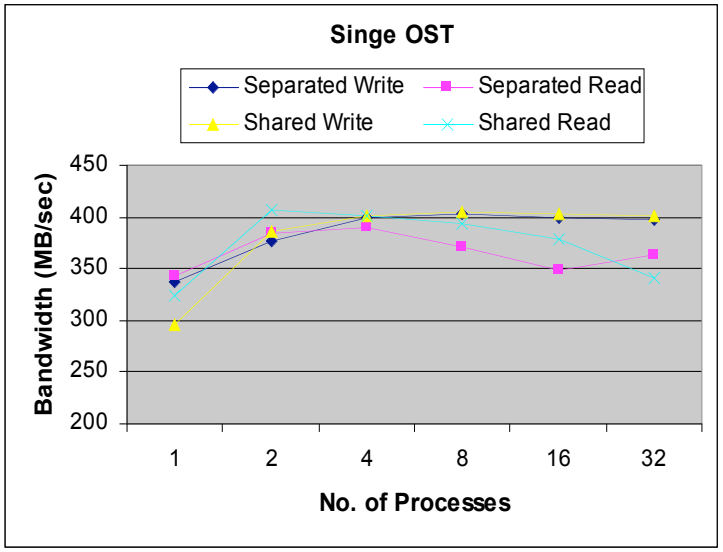


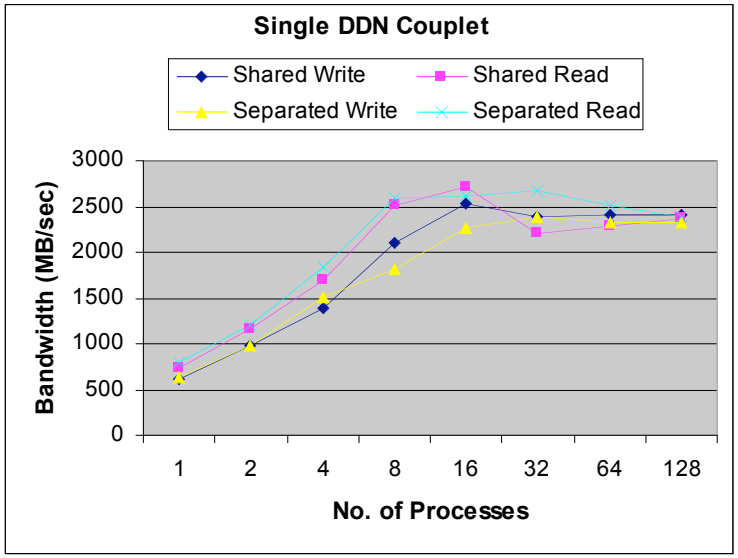
- Storage to Application, efficiency degradation
- End-to-End IO performance
  - Dependent on mid-layer components:
    - Lustre file system
    - Parallel IO libraries

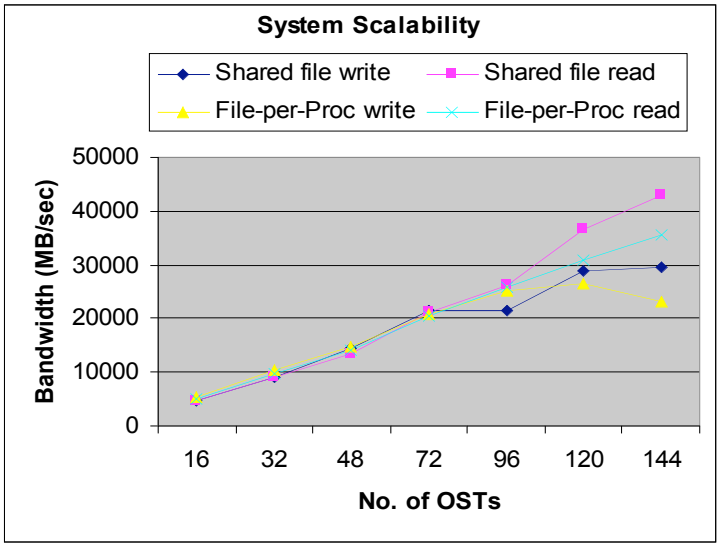
Characterization Parallel I/O

- Independent and Contiguous Read/Write
  - Share file
  - Separated files
- Non-contiguous I/O
  - Data-sieving
  - Collective I/O
- Parallel Open/Create
- Orchestrated I/O Accesses over Jaguar

Independent & Contiguous I/O

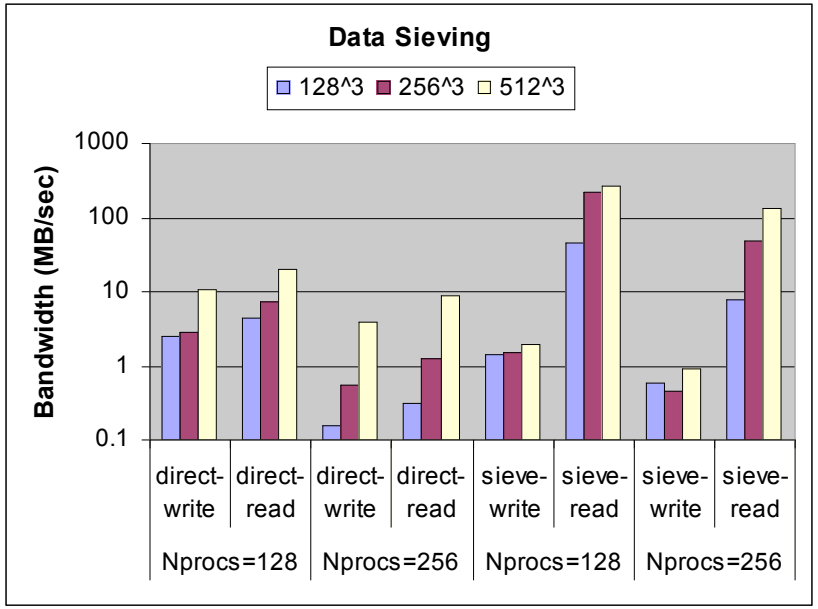


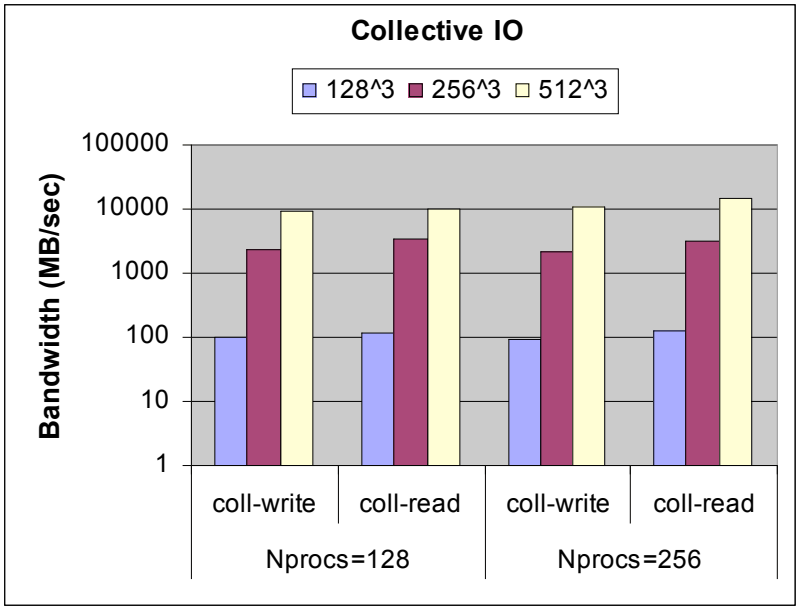




- Independent & contiguous I/O scales well unless too many clients are contending for the same servers (over-provisioned)
- Write scales better with increasing number of processes
- Read scales better with very large striping width

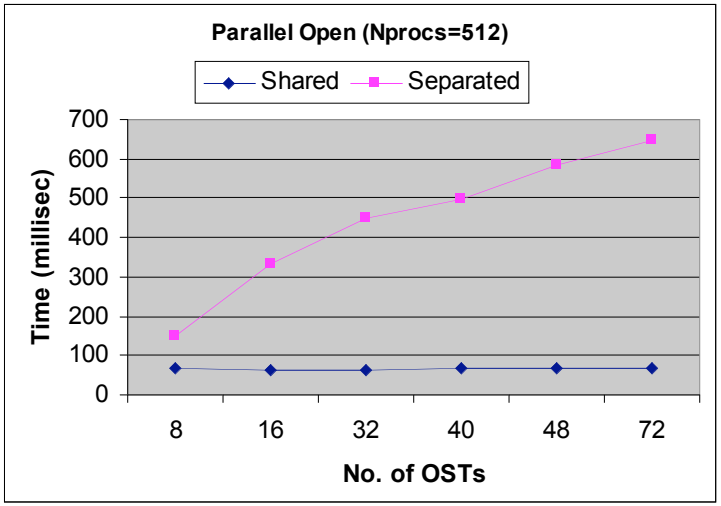
Small and Noncontiguous I/O

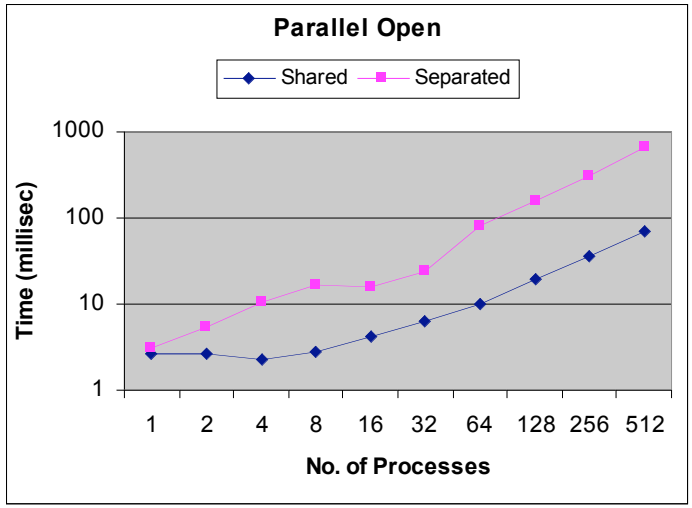


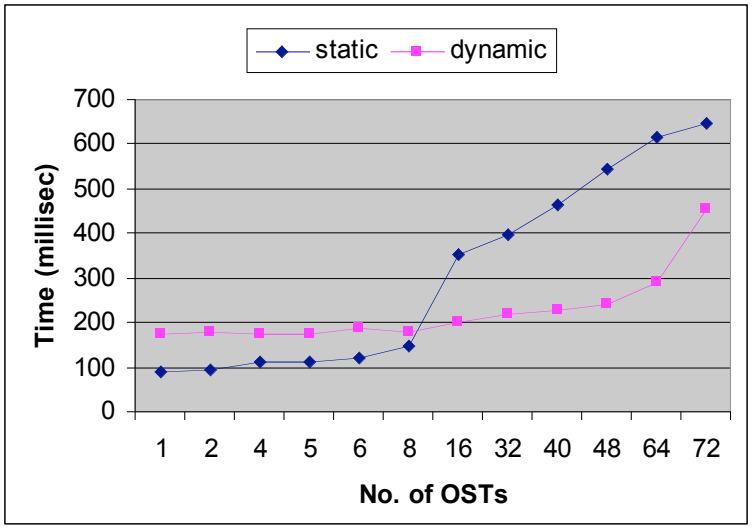


- Small and Noncontiguous I/O results in lower performance
- Strong scaling makes the problem even worse
- Data sieving from individual processes helps only read, but not write due to increased lock connections
- Collective I/O is able to help on this by aggregating I/O and eliminating lock contentions

Parallel Open/Create



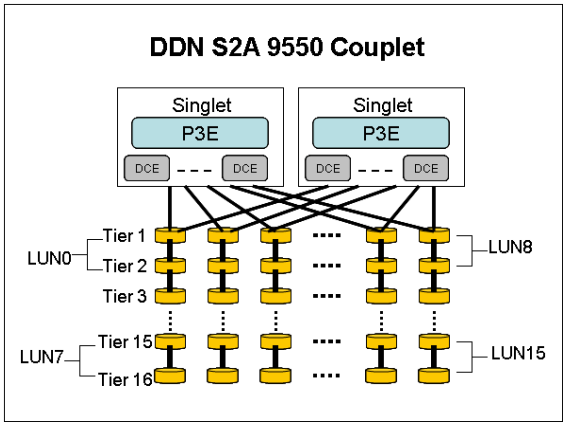




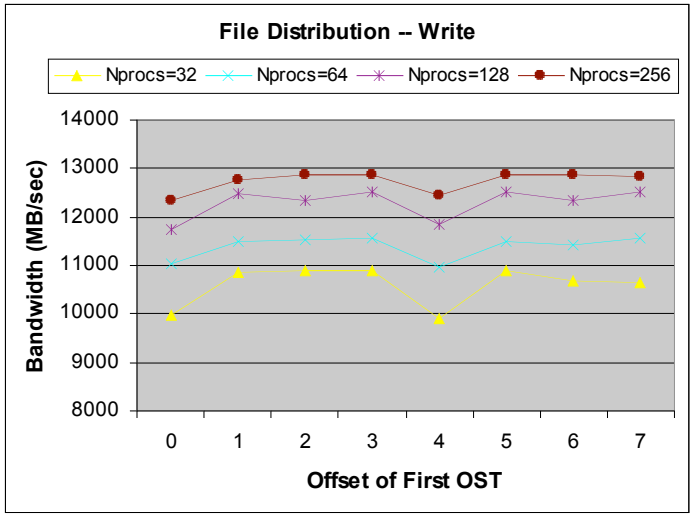
- Use a shared MPI file for scalable parallel Open/Create
- Select the first OST dynamically for scalable parallel creation

Orchestrated File Distribution

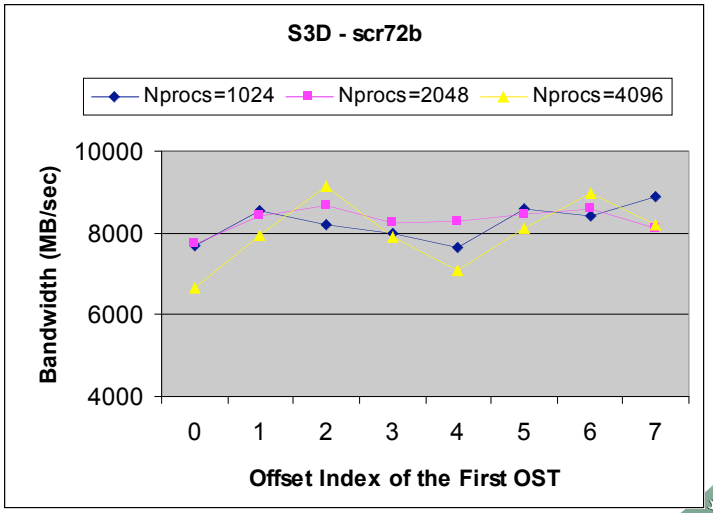
LUN/OST Organization of DDN Storage Couplet



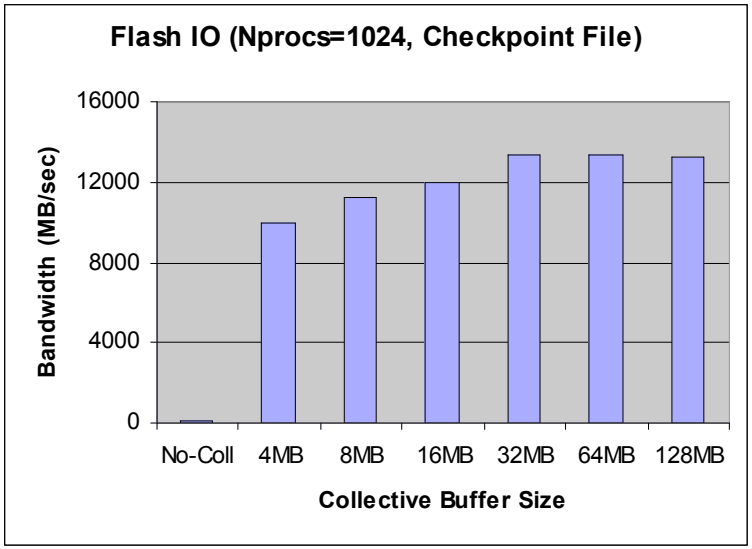
Impact of File Distribution with a Couplet

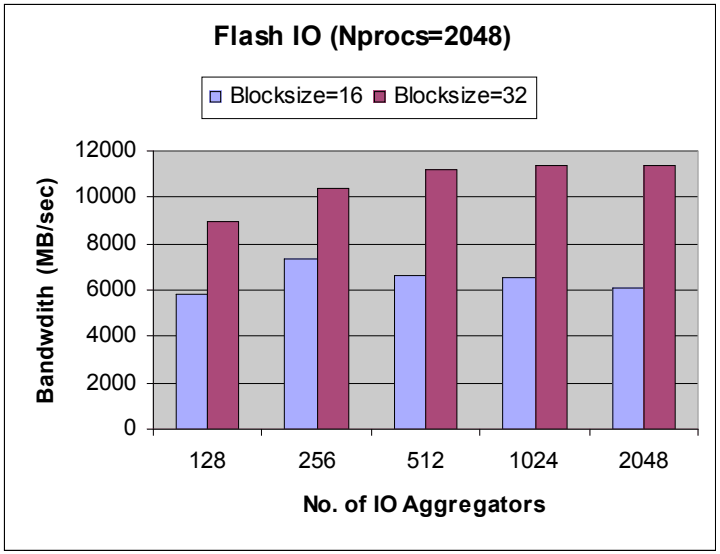


Impact of File Distribution for S3D



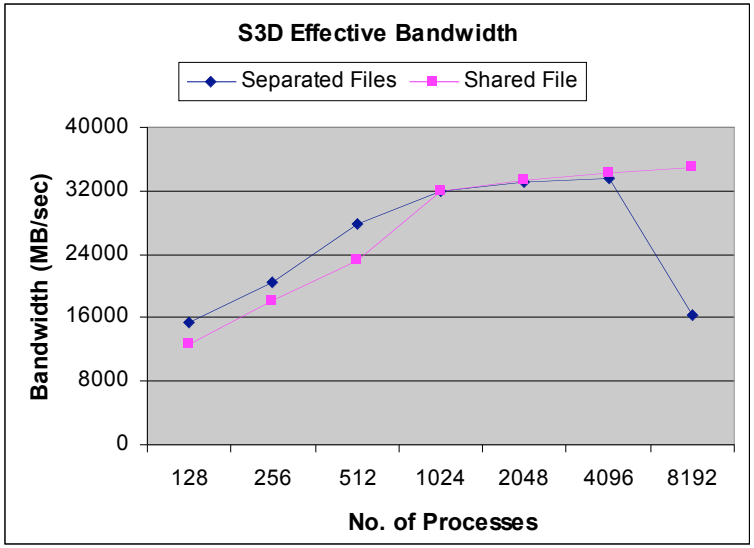
Flash IO Tuning

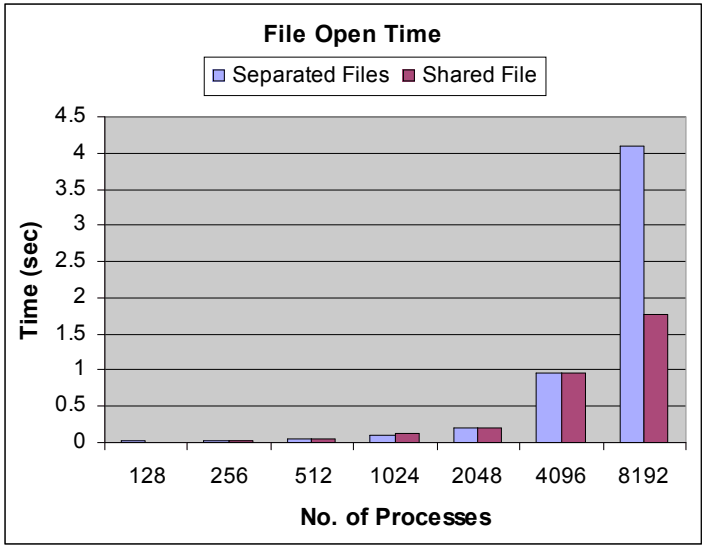




- Collective I/O significantly improves the I/O performance of Flash I/O
- For small block size, choose a smaller number of aggregators for Flash I/O

Using a Shared MPI File For S3D

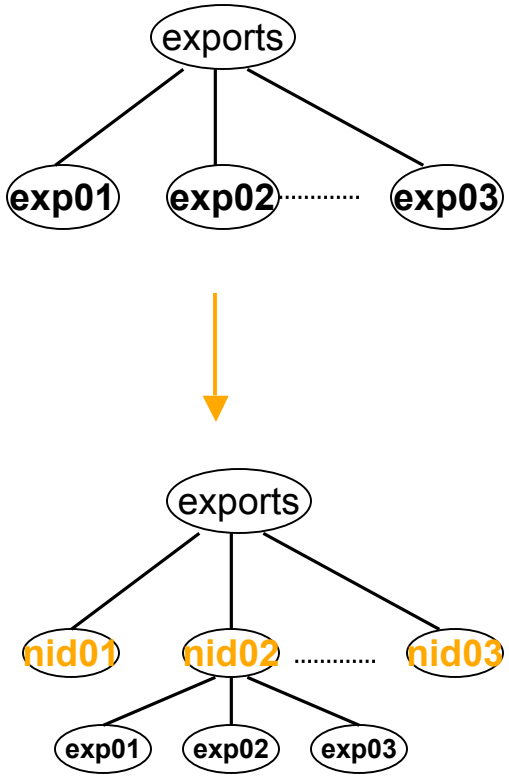




- Using separated files, S3D does not scale up to 8K processes due to file creation cost
- Using a shared MPI file for S3D can reduce file creation cost
- More convenient in managing much fewer number of files

Profiling through Server-based Client I/O Statistics in Lustre

- **Lustre Client Statistics**
  - **/proc/fs/lustre/*/*/exports**
  - **Per-exp client statistics is not sufficient**
    - **liblustre clients are transient**
  - **Provided per-nid client statistics**
  - **Integrated into Lustre source release**
- **Benefits**
  - **In-depth tracing of aggressive usage**
  - **Fine-grained I/O profiling**
  - **Knobs for future I/O service control**

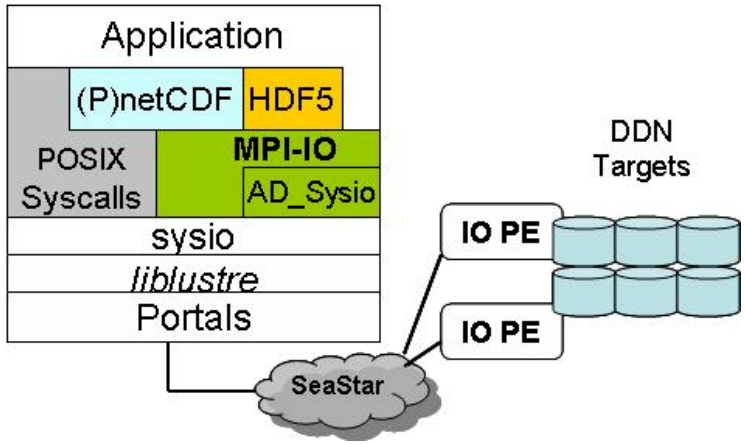


Parallel I/O Instrumentation & Optimization

- Overview of Jaguar and its I/O Subsystem
- Characterization, Profiling and Tuning
  - Parallel I/O
  - Storage Orchestration
  - Server-based client I/O statistics in Lustre file system
- Optimize Parallel I/O over Jaguar
  - OPAL: Opportunistic and Adaptive MPI-IO library over Lustre
  - Arbitrary striping pattern per MPI File; stripe aligned I/O accesses
  - Direct I/O and Lockless I/O
  - Hierarchical Striping
  - Partitioned Collective I/O
- Conclusion & Future Work

Parallel I/O Stack on Jaguar

- Parallel I/O Stack
  - MPI-IO, NetCDF, HDF5
  - Using MPI-IO as a basic building component
  - Default Implementation from Cray
    - With components named AD_Sysio over sysio & liblustre
- Lack of an open-source MPI-IO implementation
  - Proprietary code base; Prolonged phase of bug fixes
  - Stagnant code development w.r.t Lustre file system features
  - Lack of latitudes for examination of internal MPI-IO implementation
  - Lack of freedom for other developers in experimenting optimizations



OPAL

- OPportunistic and Adaptive MPI-IO Library
- An MPI-IO package optimized for Lustre
- Featuring
  - Dual-Platform (Cray XT and Linux); unified code
  - Open source, better performance
  - Improved data-sieving implementation
  - Arbitrary striping specification over Cray XT
  - Lustre stripe-aligned file domain partitioning
  - Direct I/O, Lockess I/O, and more …
- Opportunities
  - Parallel I/O profiling and tuning
  - A code base for patches and optimizations
- http://ft.ornl.gov/projects/io/#download

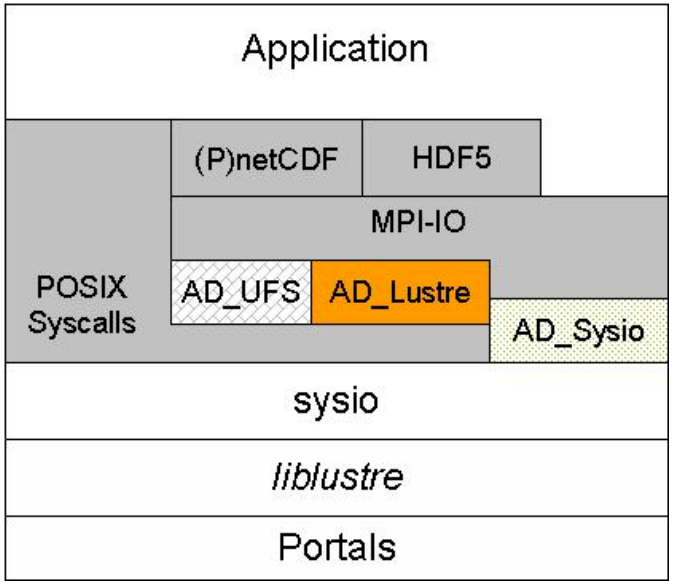
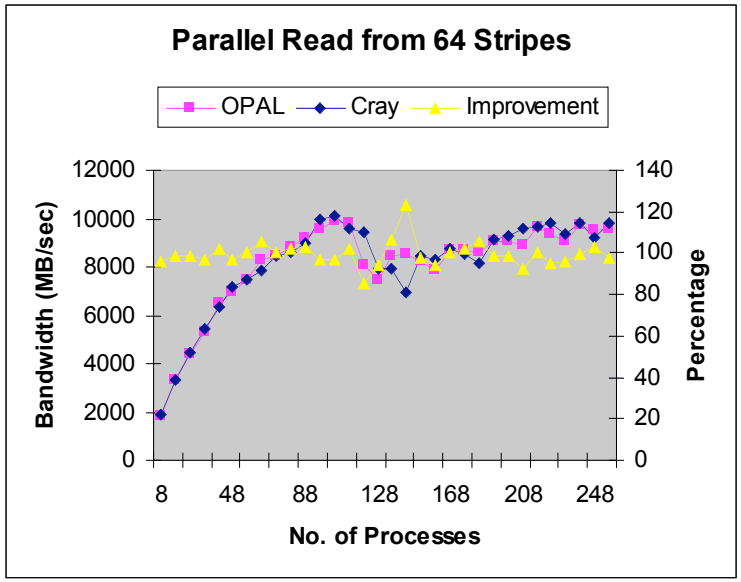
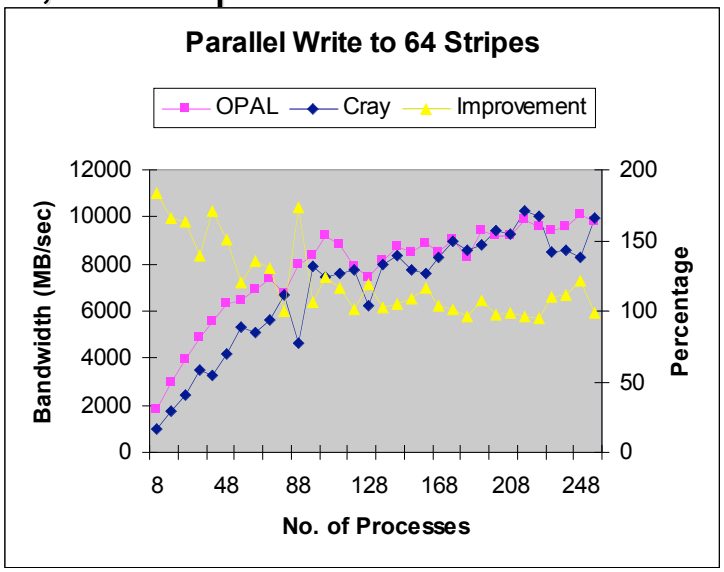


Diagram of OPAL over Jaguar

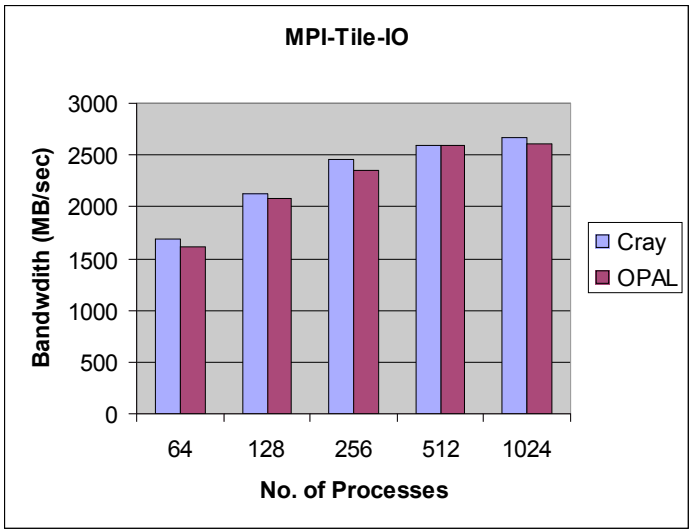
Independent Read/Write

- IOR Read/Write
  - Block size 512MB, transfer size 4MB
  - Read from or write to a shared file, 64 stripes wide

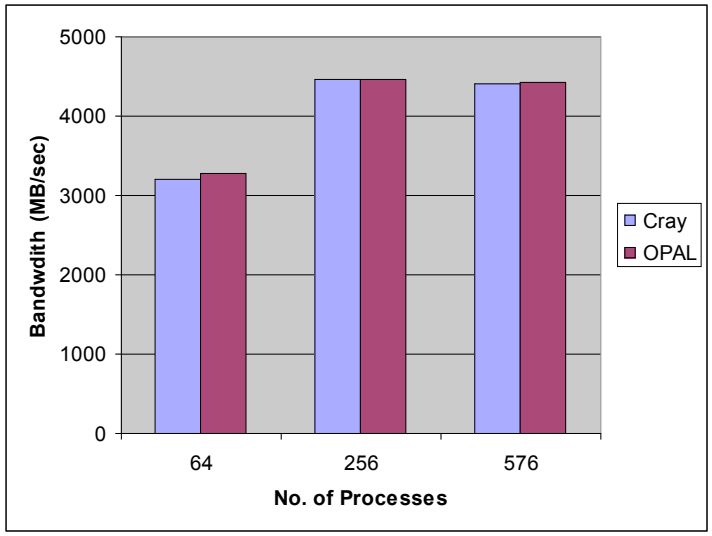




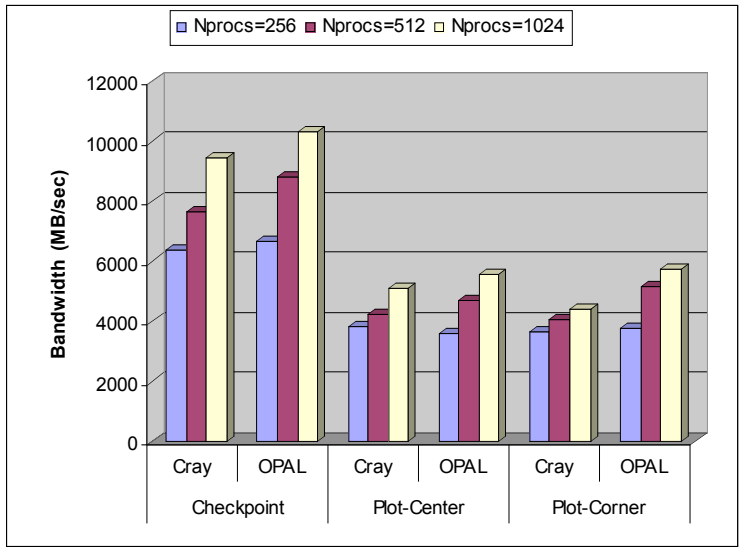
Scientific Benchmarks – Results



MPI-Tile-IO



NAS BT-IO



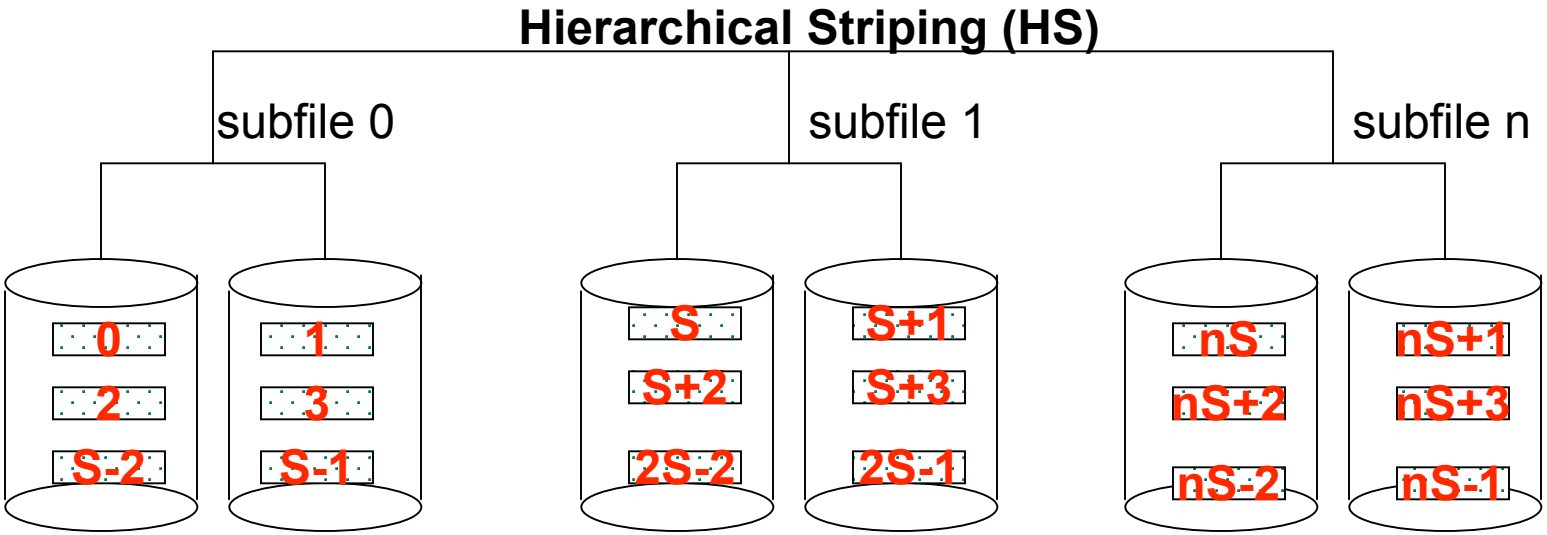
Flash I/O

Direct I/O and Lockless I/O

- Direct I/O
  - Perform I/O without file system buffering at the clients
  - Need to be page-aligned
- Lockless I/O
  - Client I/O without acquiring file system locks
  - Beneficial for small and non-contiguous I/O
  - Beneficial for clients with I/O to the same stripes
- Recently implemented in OPAL for Lustre platforms
  - Works over Linux cluster and Cray XT (CNL)
  - Performance improvement for large streaming I/O on a Linux cluster
  - Benefits over Jaguar/CNL is yet to be determined

Hierarchical Striping

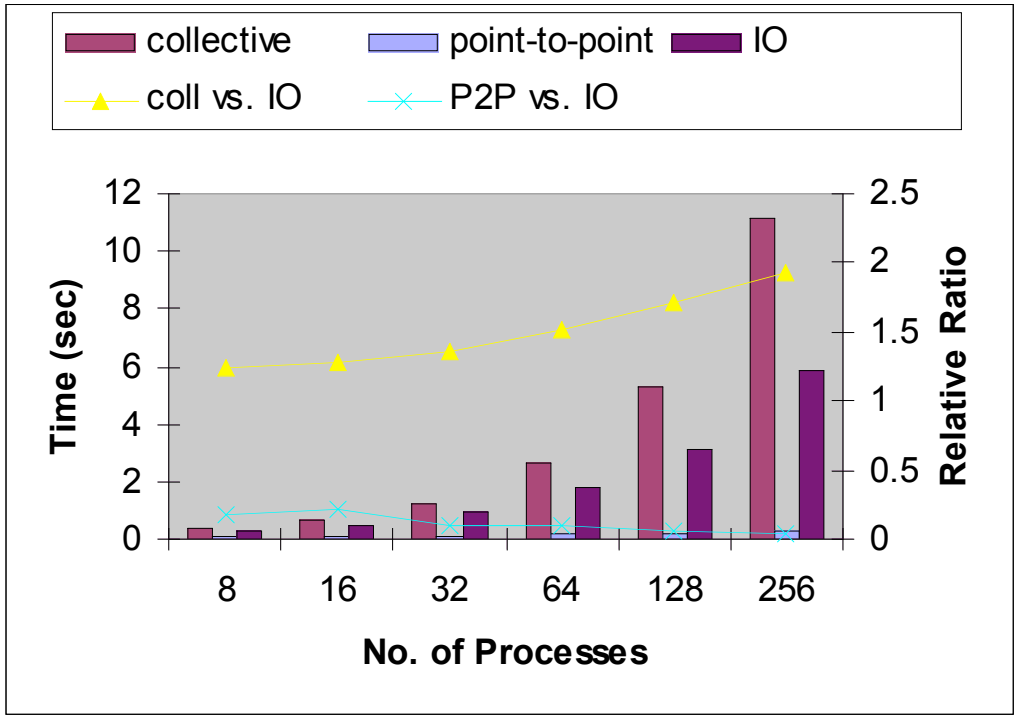
- An revised implementation of MPI-I/O with subfiles (CCGrid'2007)
  - Break Lustre 160-stripe limitation
  - Avoid over-provisioning problem
  - Mitigate the impact of striping overhead
- Data Layout: hierarchical striping
  - Create another level of striping pattern for subfiles
  - Allow maximum coverage of Lustre storage targets



(Stripe width: 2; Stripe size: w)

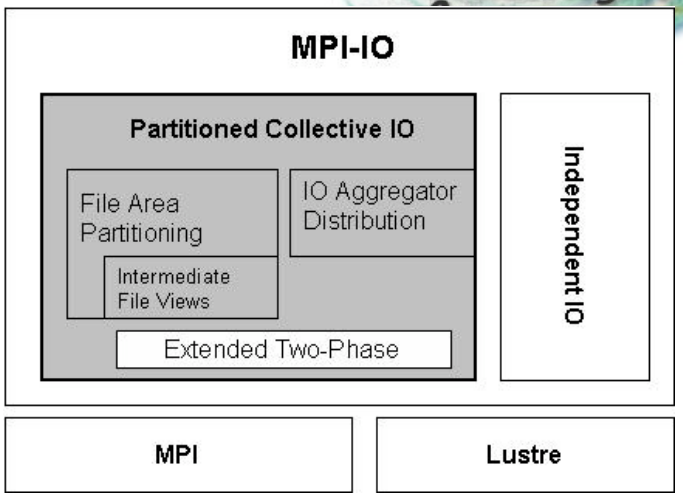
Global Synchronization in Collective I/O on Cray XT

- Collective IO has low performance on Jaguar?
  - Much time is spent on global synchronization
  - Application scientists are forced to aggregate I/O into smaller sub-communicators

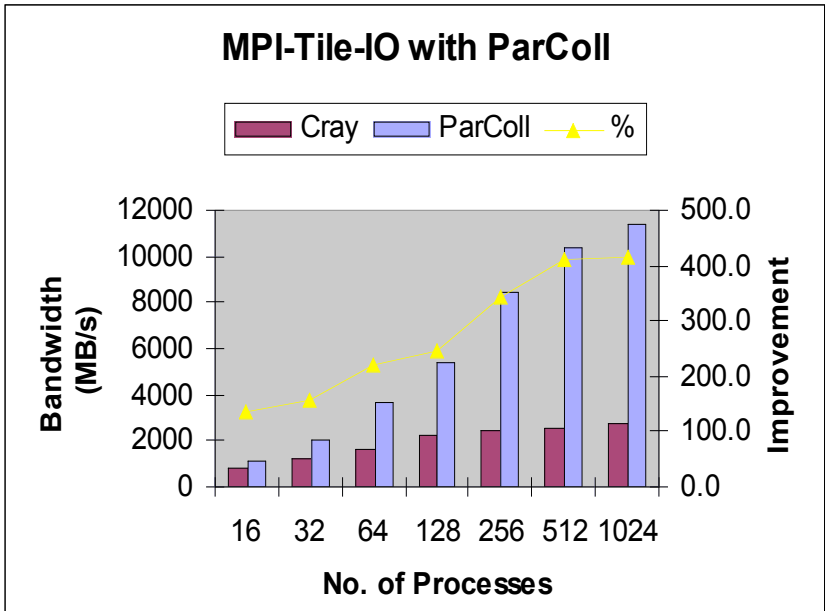


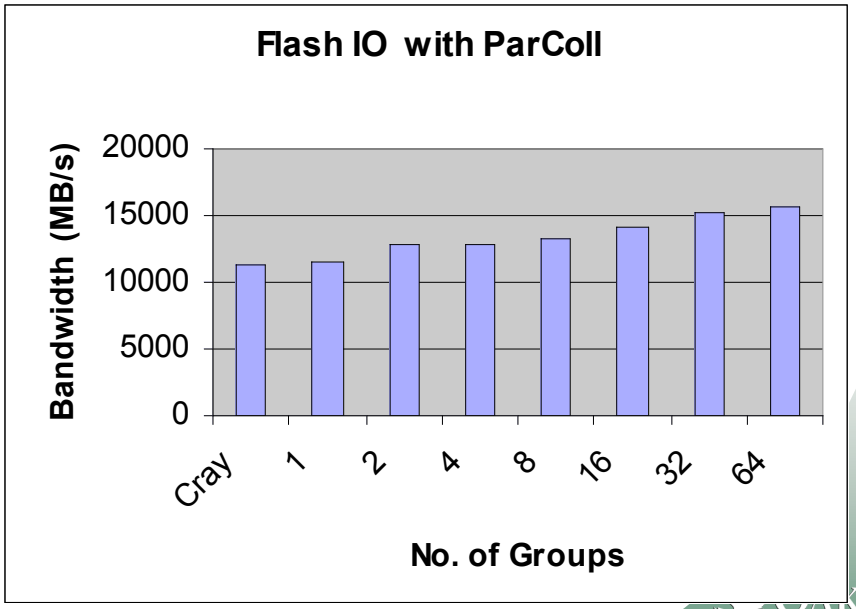
Partitioned Collective I/O (ParColl)

- Prototyped in an open-source MPI-IO library
- Divide processes into smaller groups:
  - Reduced global synchronization
  - Still benefit from IO aggregation
- Improves collective IO for different benchmarks



Benefits to MPI-Tile-IO and Flash IO





Conclusion & Future Work

- Characterized I/O Performance over Jaguar
- Tuned and optimized various I/O Patterns
- To make OPAL a public software release
- To model and predict application I/O performance
- A pointer again:
  - ☺ http://ft.ornl.gov/project/io/