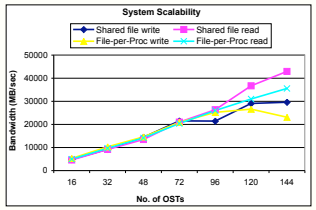
Parallel I/O on the Cray XT

Weikuan Yu and Jeffrey S. Vetter

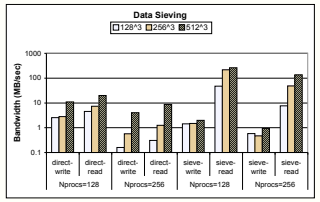
Characterization and Tuning

(Yu, Vetter and Oral. IPDPS'08)

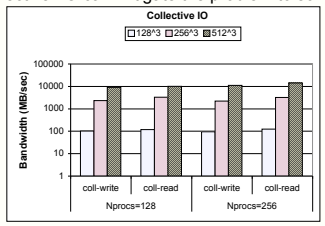
- Sequential, large block I/O scales well on Cray XT



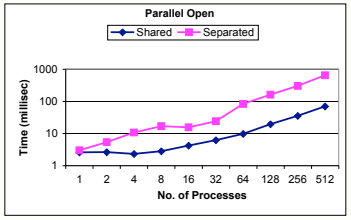
- Small, overlapping I/O leads to poor performance



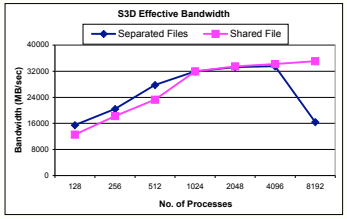
- Collective I/O can mitigate the problem to some extent



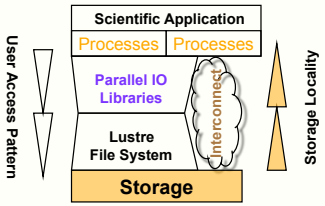
- File creation does not scale due to the metadata problem



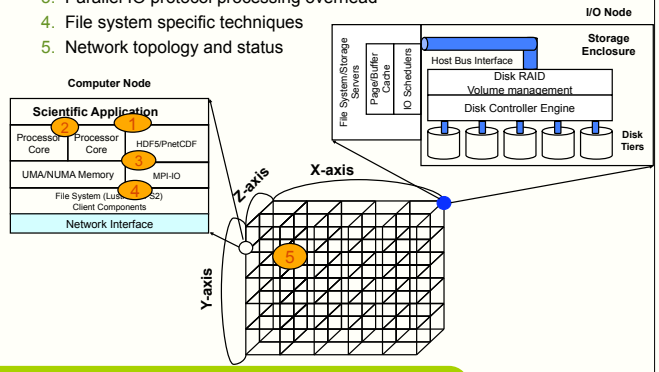
- Using a shared file when possible sustains I/O performance, e.g. for a combustion application S3D



Parallel I/O and Opportunities



- Opportunities:
  1. Application hints and data manipulation
  2. Processor/Memory Architecture
  3. Parallel IO protocol processing overhead
  4. File system specific techniques
  5. Network topology and status

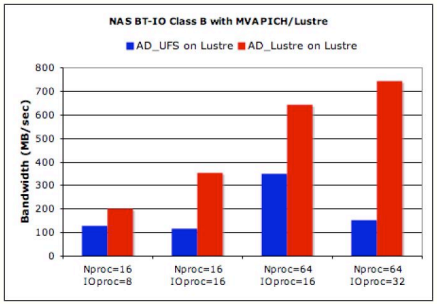


- Parallel I/O Involves Complex components
- Two main flows of information
  - Losing out in user access pattern going down
  - Losing out in storage locality info going up
- End-to-end I/O Performance
  - Depends on application knowledge
  - Depends on knowledge of parallel I/O libraries
  - Needs good understanding of file systems

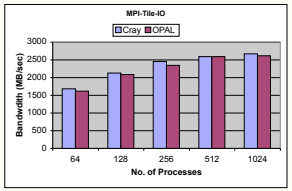
OPAL: Open Source MPI-IO Driver for Lustre

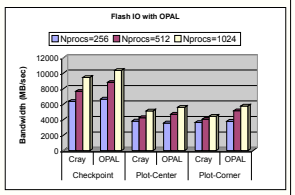
(Yu, Vetter and Canon. SNAPI'07)

- Parallel I/O over Cray XT
  - A proprietary implementation from Cray
- OPAL: An open source implementation of MPI-IO
  - Overcome the restriction of a proprietary MPI-IO stack
  - Improved data-sieving implementation
  - Arbitrary striping specification over Cray XT
  - Lustre stripe-aligned file domain partitioning
  - Release via MVAPICH-1.0 and MPICH2-1.0.7

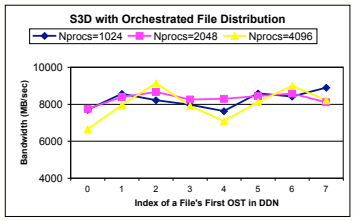


- Performance validated against Cray stack
  - Comparable performance on independent I/O
  - Comparable performance on collective I/O pattern





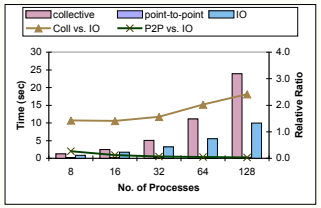
- Accessibility to profile internal MPI-IO implementation
- Benefits in controlling applications' file distribution, as shown for a combustion application S3D



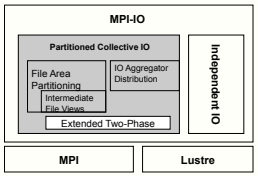
Partitioned Collective I/O

(Yu and Vetter. ICPP'08)

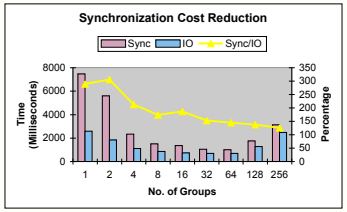
- Global Synchronization forces a **collective wall** on I/O



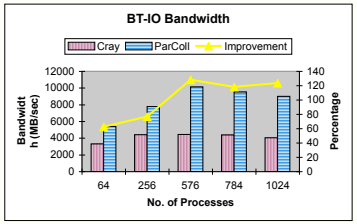
- ParColl: a scheme for I/O aggregation with balanced group size and synchronization cost



- Significantly reduces synchronization overhead at 1024 processes, 8 per partitioned group.



- Improves the performance of NAS I/O benchmark, BT-IO, by 120%



- Improves the I/O performance of Flash by 38%