Background

eScience Project – Sptember, 2008

- R&D for portable, better programming environment for T2K clusters
  - Xruntime – System Software
    - File Staging System
      - On-Demand File Staging System

        Catwalk [Cluster2009]
      - Catwalk-ROMIO

        File staging system having MPI-IO API

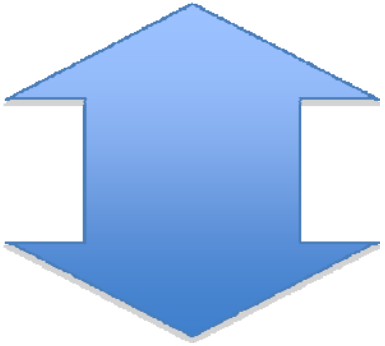
Parallel File IO Using Ring Communication Topology

- From Viewpoint of File Staging -

Atushi HORI, Kazuki OHTA, Yutaka ISHIKAWA
University of Tokyo

Problem of Parallel File IO

- Nature of Parallel IO
  - Parallel IO is inherently non-contiguous.
  - The larger parallelism, the finer IO.



Big Gap !

- Nature of Disk IO
  - Sequential (contiguous) IO is the best way.
  - The larger requests, the better performance.

Techniques so far Proposed

- Two-Phase IO [del Rosario 93]

  1st phase: Merge and re-order IO requests

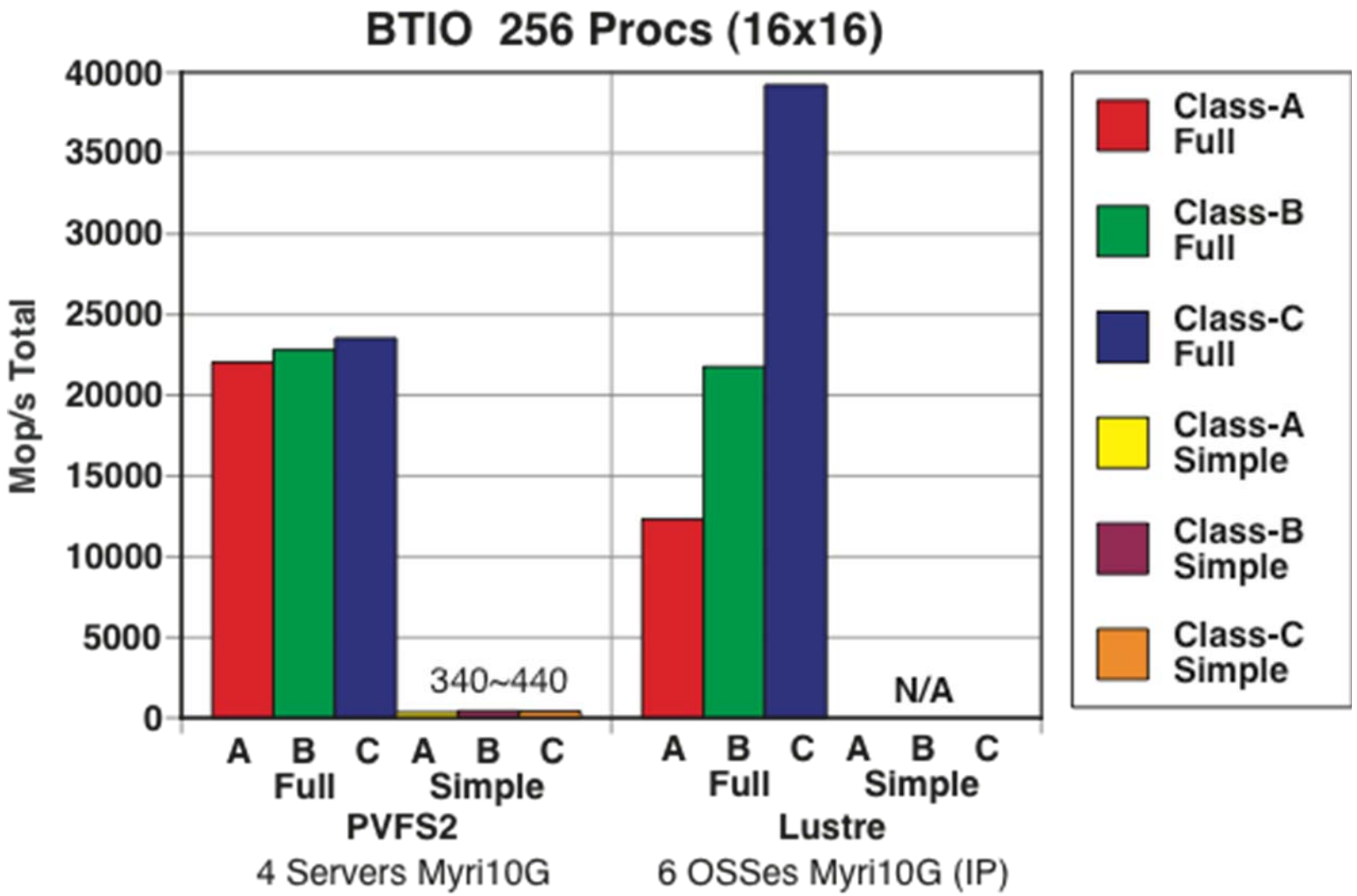
  2nd phase: Dispatch the merged, re-ordered IO requests to disks.

- Data Sieving [Thakur 99]
  - Access disks with larger extents, and then
  - Accessed data are sieved

Drawbacks

- Only effective on collective IO
- Need of read-modify-write, …

BTIO Examples (MPI-IO)



BTIO - MPIIO

- Two Data Access Patterns
  - Simple MPI_File_write_at()

    Easy to program, poor performance

Which way to go ?

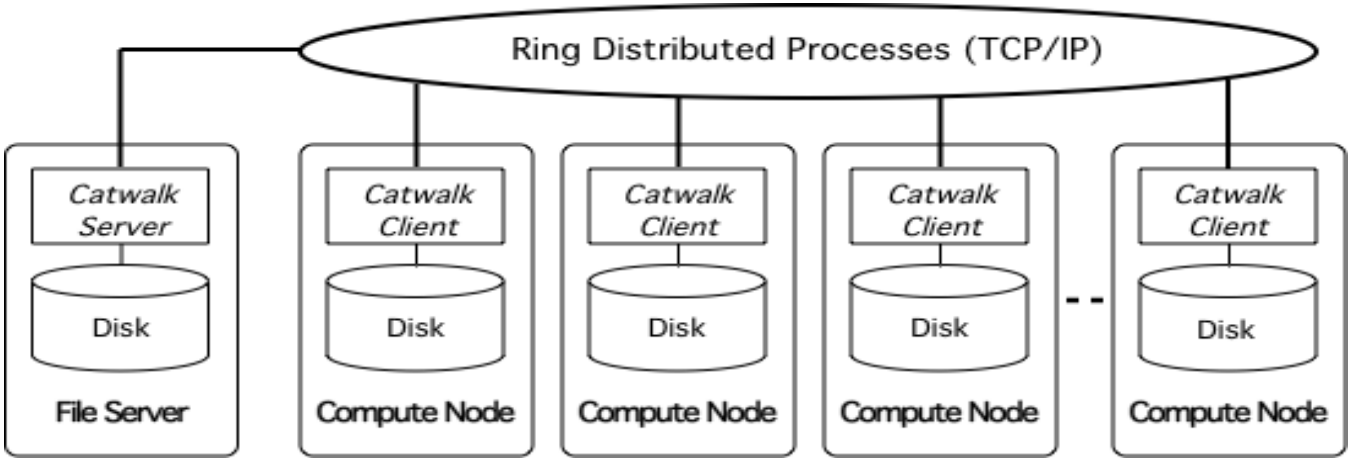
  - Full MPI_File_write_at_all()

    Tough to program, good performance

    (2-phase IO, Data sieving)

Proposed Technique

- **Ring Communication Topology [Cluster09]**



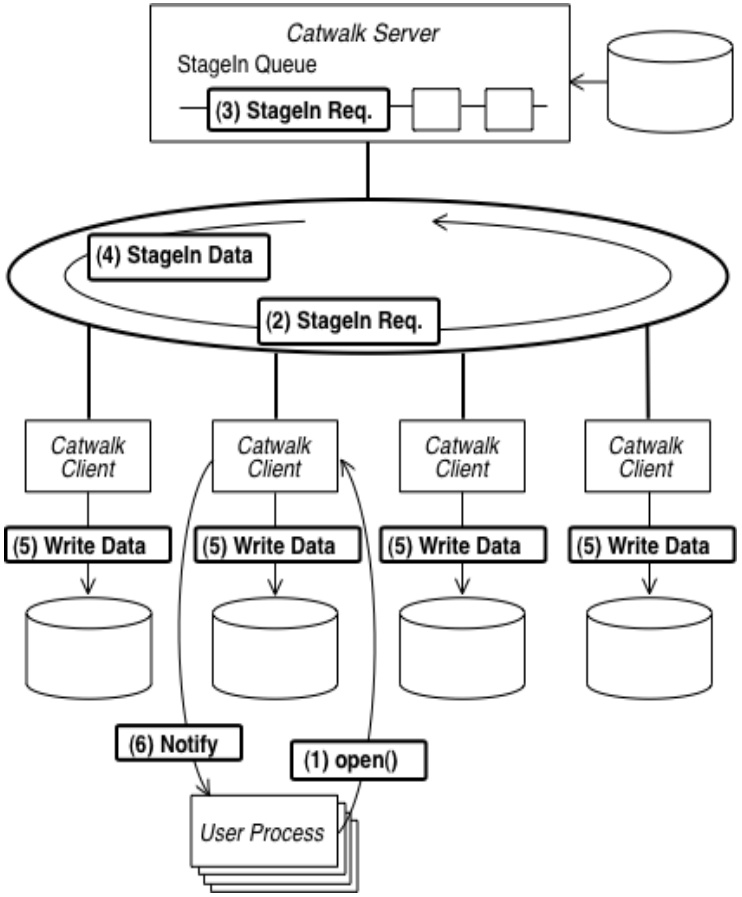
- **Read – Copy entire file on the server to each compute node.**
- **Write – IO requests are sorted and merged along with the ring.**

MPI-IO Impl. is named "Catwalk-ROMIO".

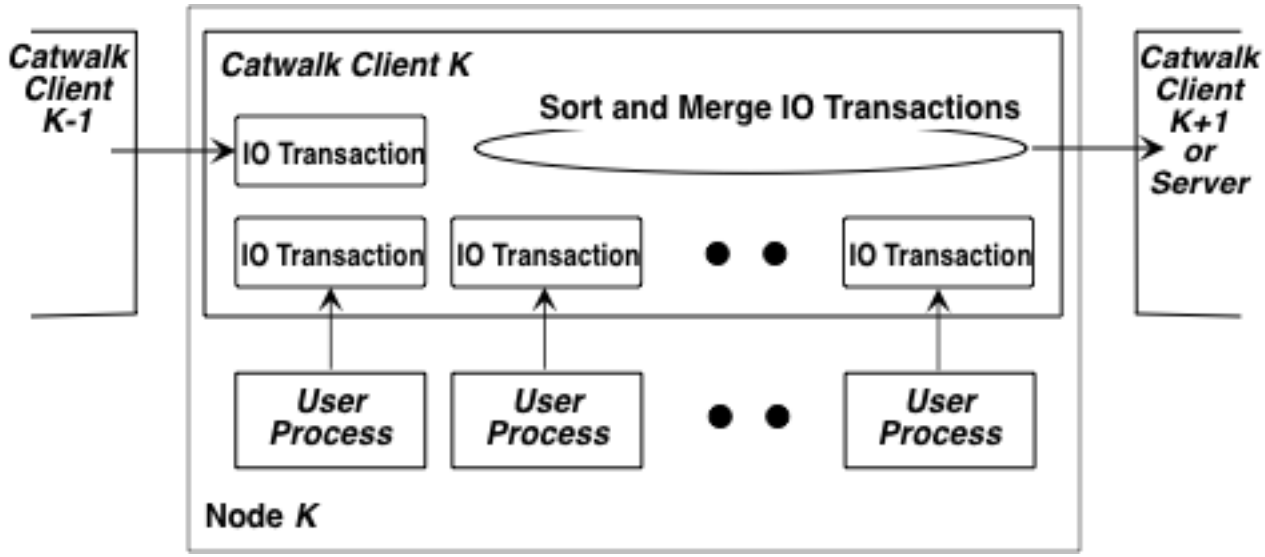
Catwalk-ROMIO - Read

- Upon users requests, Catwalk copies *entire* file on server node to every local disk of client node.
- The time needed for the *pipelined* copying depends on file size, *independent* from the number of nodes.

[Cluster09]



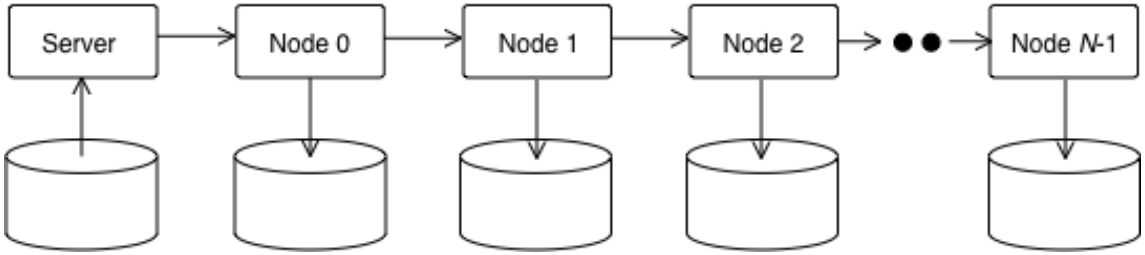
Catwalk-ROMIO – Write



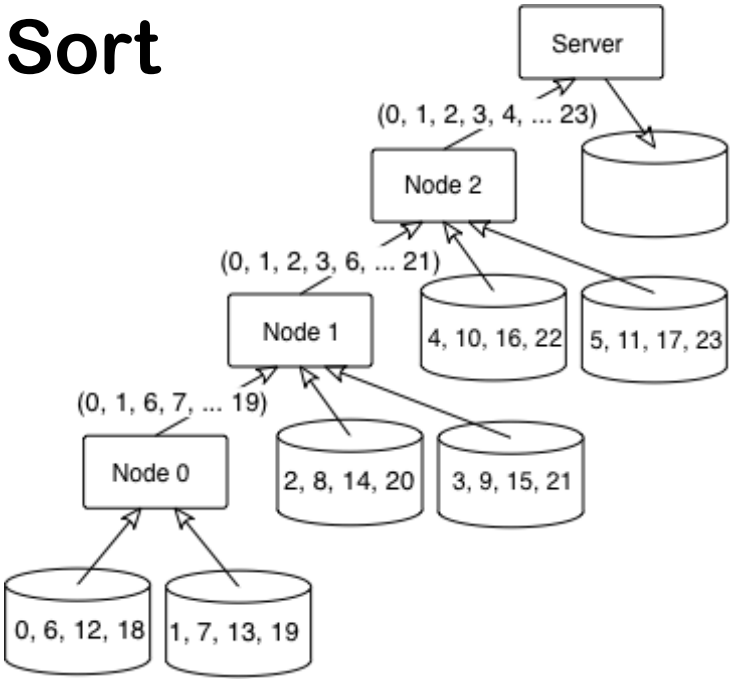
- IO requests are sorted by offset and merged if possible, so that
  - Lots of fine grain, back-and-forth IO requests can get
  - Smaller number of coarser, contiguous IO requests.

Catwalk-ROMIO Illustrated

- **Read - Pipeline**



- **Write - Merge Sort**



Theory

Assuming Network is much faster than Disk

Tr, Tw: Time to read and write
B: Disk bandwidth
S: Size of file
N: Number of nodes

- **Catwalk-ROMIO Read**

  $Tr = S / B +$

  $S / N / B$

  Time to stage-in dominates

  Time to read local disk. When N is 16, only 6 %

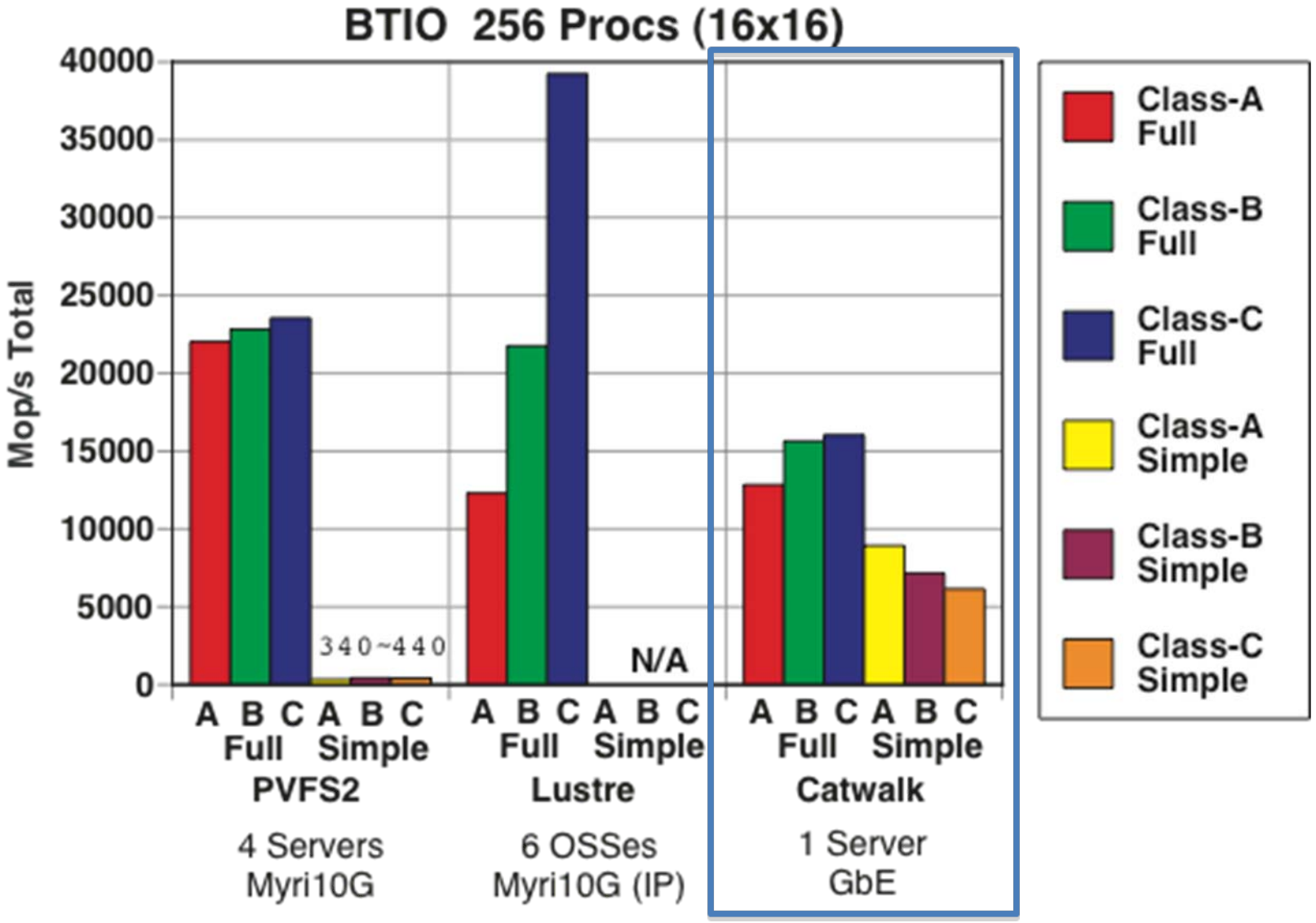
- **Catwalk-ROMIO Write** (No local disk is involved)

  $Tw = S / B$

  Time to write on server node

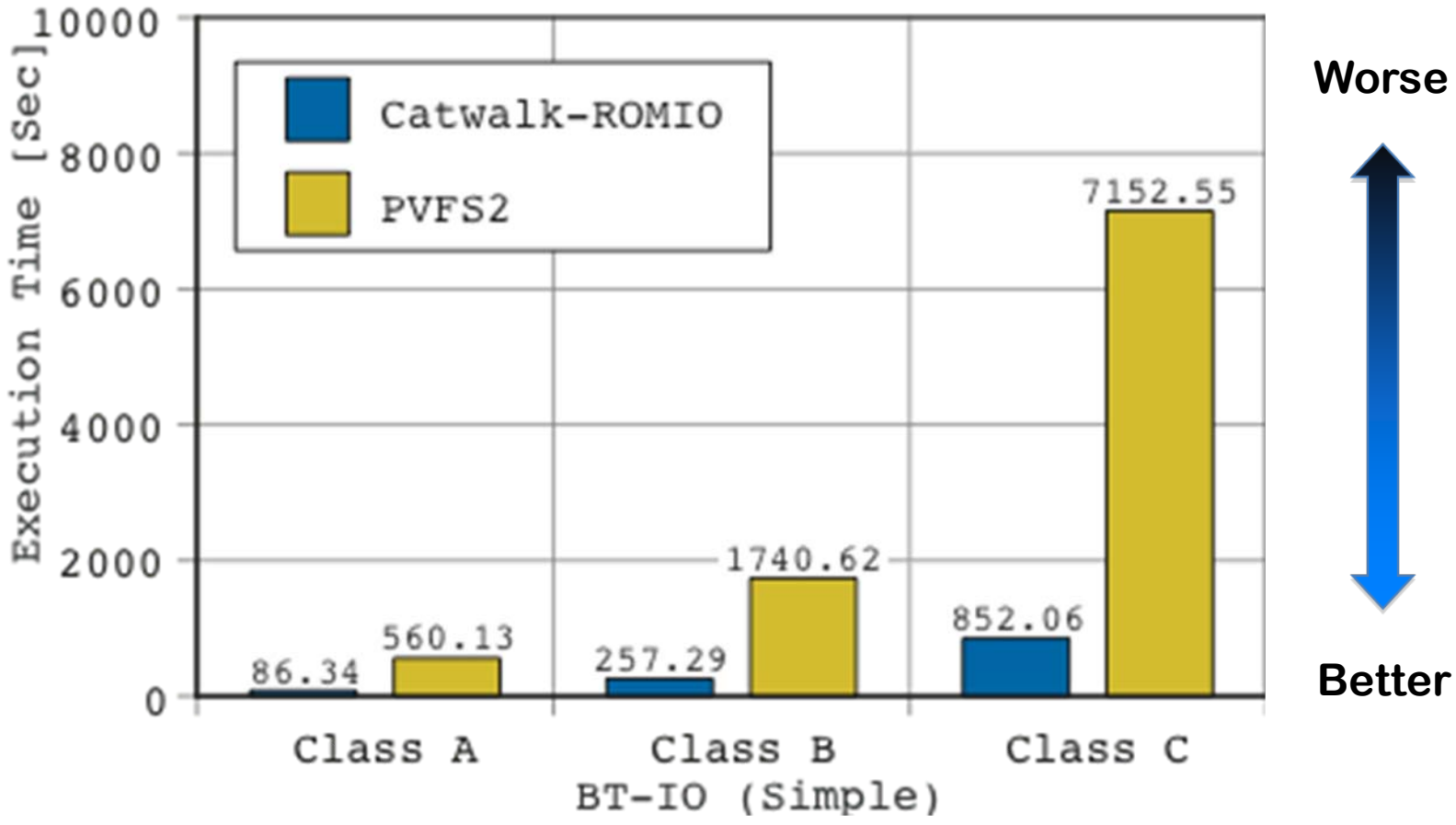
Note: Disk bandwidth (B) may degrade very much, if IO pattern is not COARSE and CONTIGUOUS.

BTIO with Catwalk-ROMIO



BTIO including Reading Time

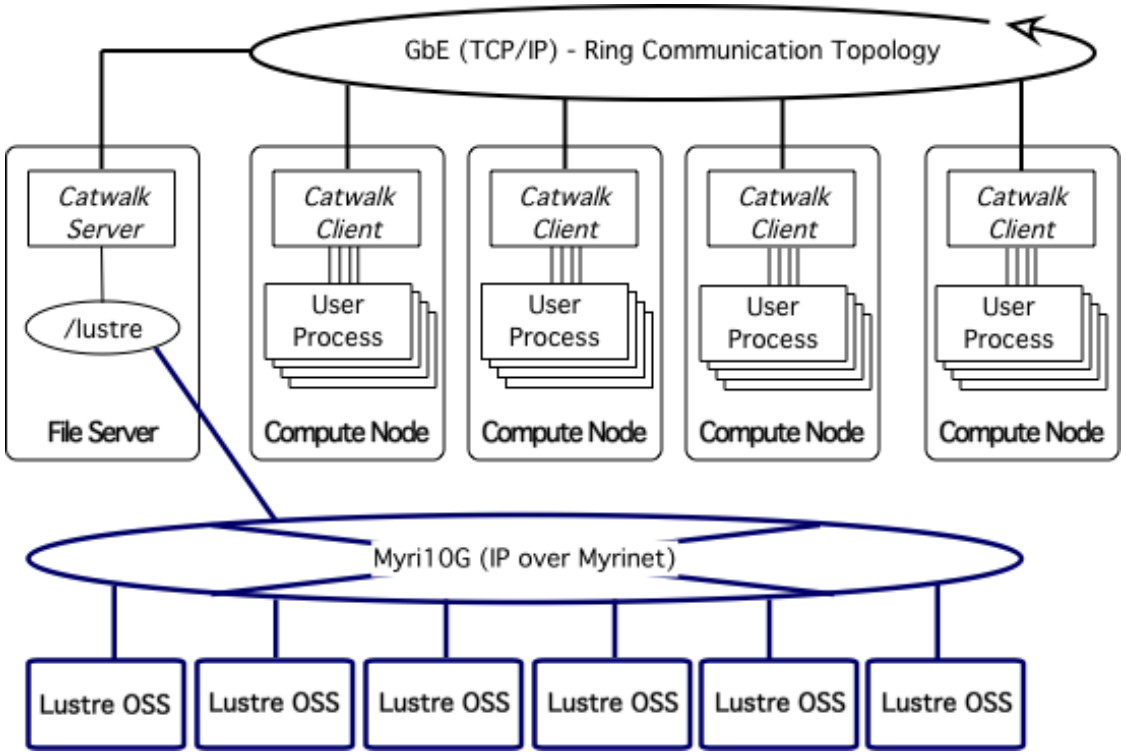
% time mpirun bt.[ABC].256.mpi_io_simple



Catwalk-ROMIO Front-End

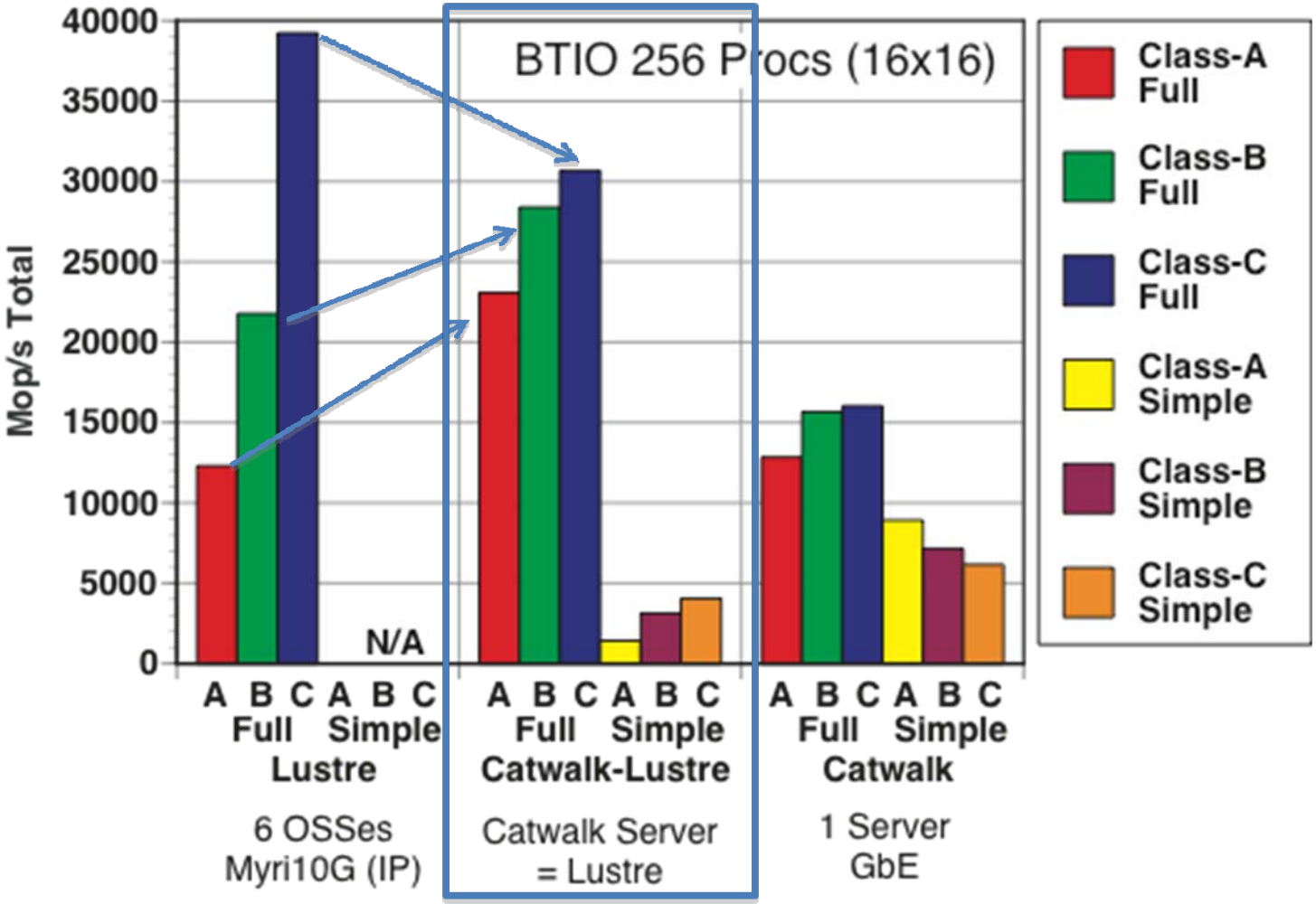
- Assumed that Catwalk server accesses local disk.
- What happens if Catwalk server access a parallel file system ?

  Catwalk-ROMIO as a FRONT-END



Catwalk-ROMIO and Lustre

Catwalk-ROMIO can be a front-end of a parallel file system to improve IO access pattern.



Summary

- Catwalk-ROMIO Read
  - Copying entire file in pipeline
  - The largest extent of *data sieving*
- Catwalk-ROMIO Write
  - Similar to *two-phase IO*, but ring topology
  - No explicit IO phsases, but pipelined
- Catwalk-ROMIO
  - with one server can perform better than PVFS2 (4 servers), if IO pattern is fine, complex, and not collective.
  - can be a front-end to improve access patterns.

Problem of Parallel File IO

- Nature of Parallel IO
  - Parallel IO is inherently non-contiguous.
  - The larger parallelism, the finer IO.

Catwalk-ROMIO can narrow this gap !!

- Nature of Disk IO
  - Sequential (contiguous) IO is the best way.
  - The larger requests, the better performance.

AND

- **Catwalk-ROMIO requires no dedicated hardware (network, disks, servers).**

Catwalk can be a cost-effective solution !

Q&A

You can download Catwalk from

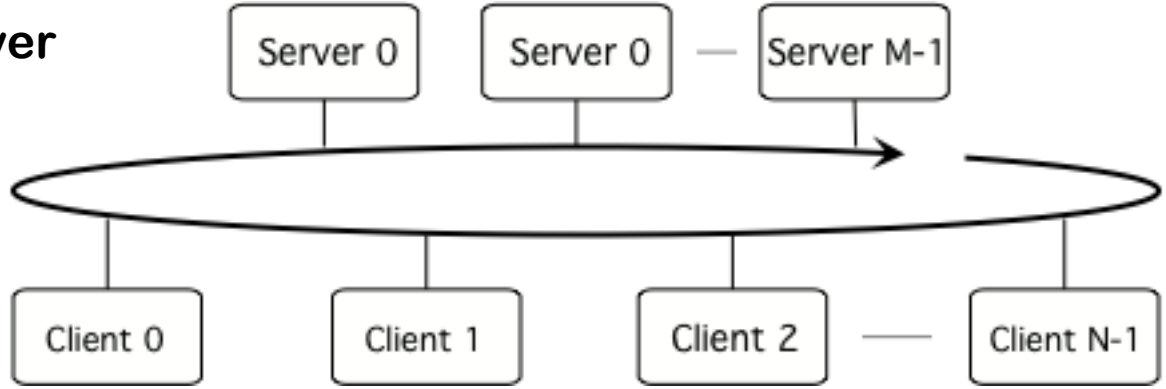
http://www.pccluster.org

as a part of SCore package.

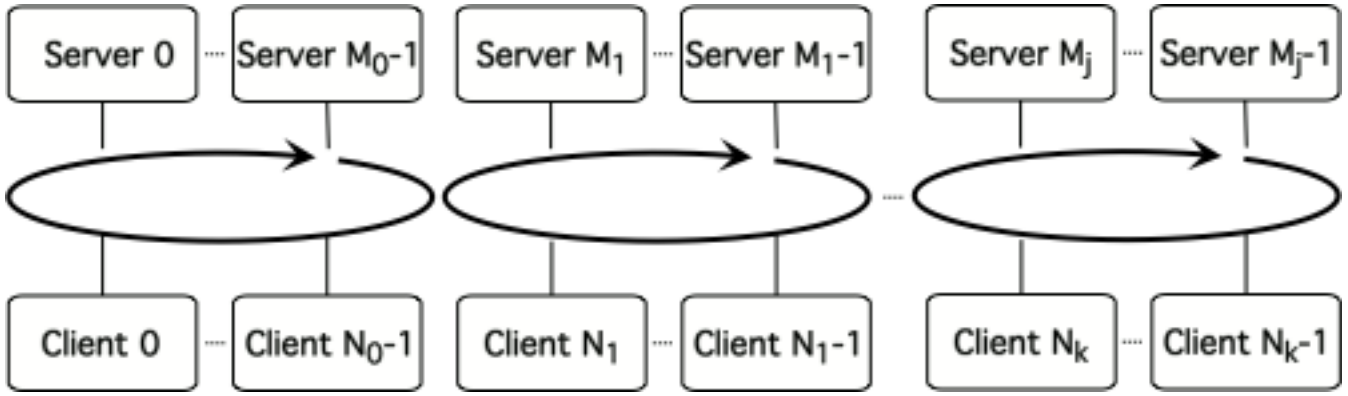
Future Work

Catwalk-ROMIO can have only one server !

Multi-Server



Multi-Server + Multi-Ring



Catwalk-ROMIO Usage Model

T2K Tokyo

T2K Tsukuba

T2K Kyoto

Data files on your private machine can be accessed by remote machines without explicit copying.

Catwalk implementation is FULLY user-level, and can be installed to any Linux system without root capability, even if it is up and running.

Catwalk-ROMIO Invocation

- **Basic**

  % catwalk –romio mpirun catwalk a.out

  Server

  Clients

- **With SSH port forwarding**

  server% catwalk –mpi –SSH comp32

  comp32% mpirun catwalk a.out

  Server

- **With Batch Scheduler**

  login% catwalk –mpi

  export CATWALK_MPIIO=*XXX:YYY*

  login% emacs submit.script

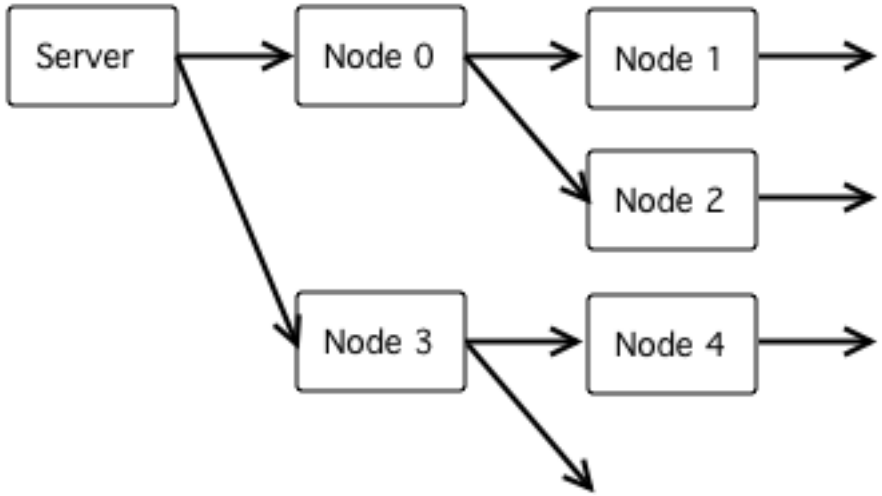
  mpirun catwalk -server *XXX:YYY* a.out ...

  login% qsub submit.script

  Server

  Clients

Pipeline vs. Tree



Star Topology

Bs = Bn/N, Bt = Bn

Ring Topology

Bs = Bn, Bt = Bn * N

PipeFill = N

K-ary Tree

Bs = Bn / K, Bt = Bn / K * N

PipeFill = $\log_k N$

Bs: Server Bandwidth

Bn: Network Link Bandwidth

Bt: Total Bandwidth

N: Number of Nodes