Cluster 2010 Presentation

Optimization Techniques at the I/O Forwarding Layer

Kazuki Ohta (presenter):
Preferred Infrastructure, Inc., University of Tokyo

Dries Kimpe, Jason Cope, Kamil Iskra, Robert Ross:
Argonne National Laboratory

Yutaka Ishikawa:
University of Tokyo

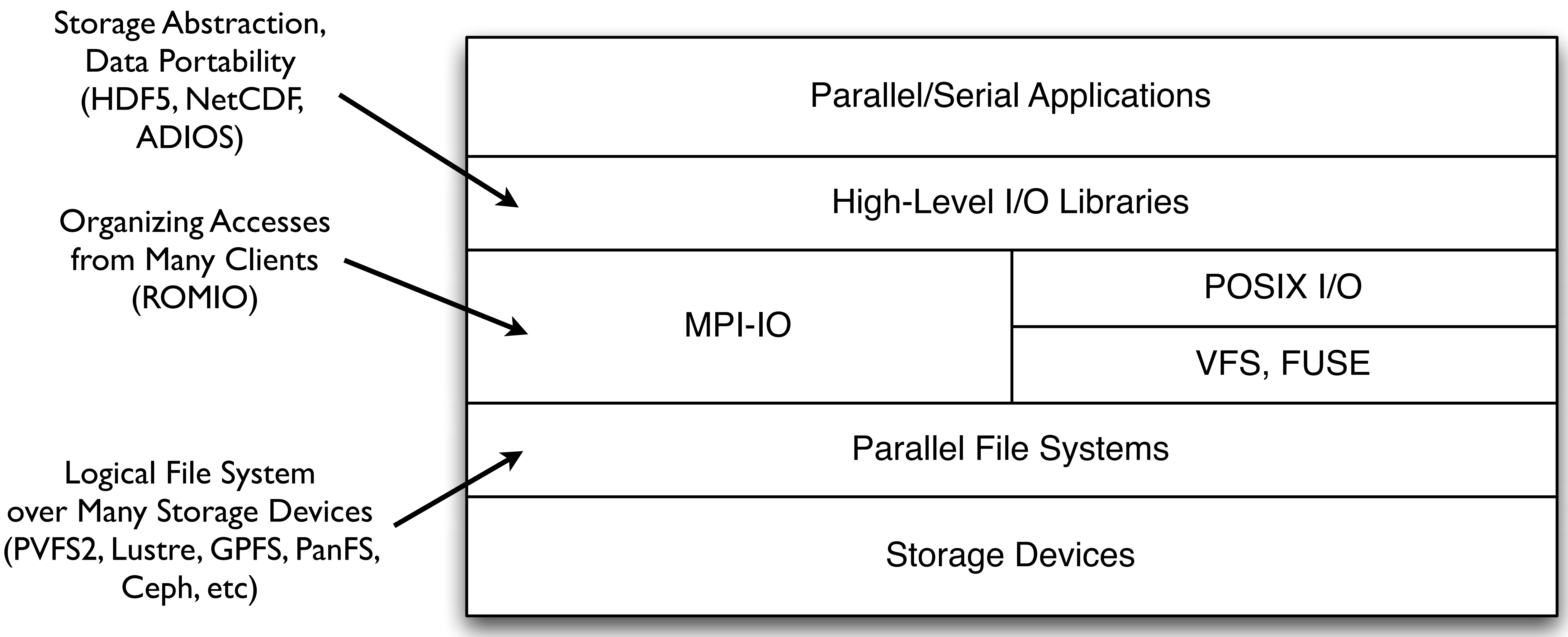
Contact: kazuki.ohta@gmail.com

Background: Compute and Storage Imbalance

- Leadership-class computational scale:
  - 100,000+ processes
  - Advanced Multi-core architectures, Compute node OSs
- Leadership-class storage scale:
  - 100+ servers
  - Commercial storage hardware, Cluster file system

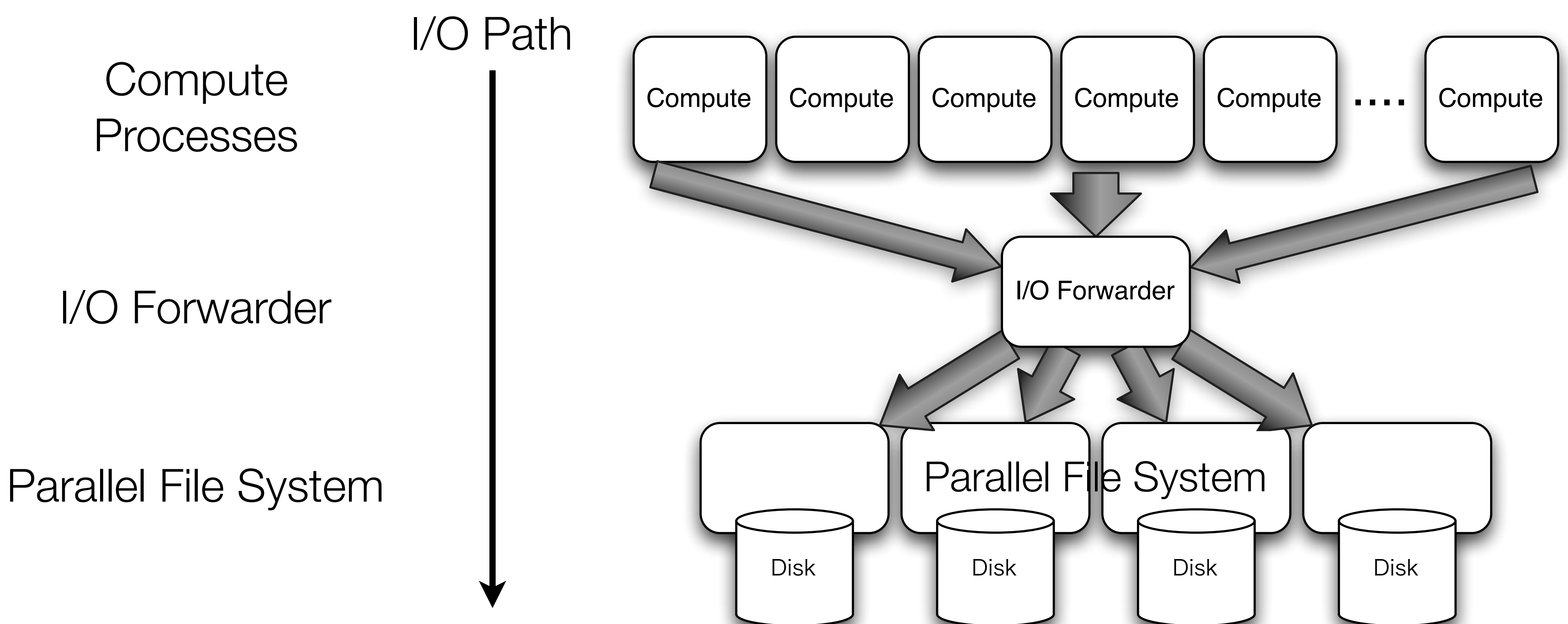
- Current leadership-class machines supply only **1GB/s of storage throughput for every 10TF of compute performance**. This gap grew factor of 10 in recent years.
- Bridging this imbalance between compute and storage is a critical problem for the large-scale computation.

Previous Studies: Current I/O Software Stack



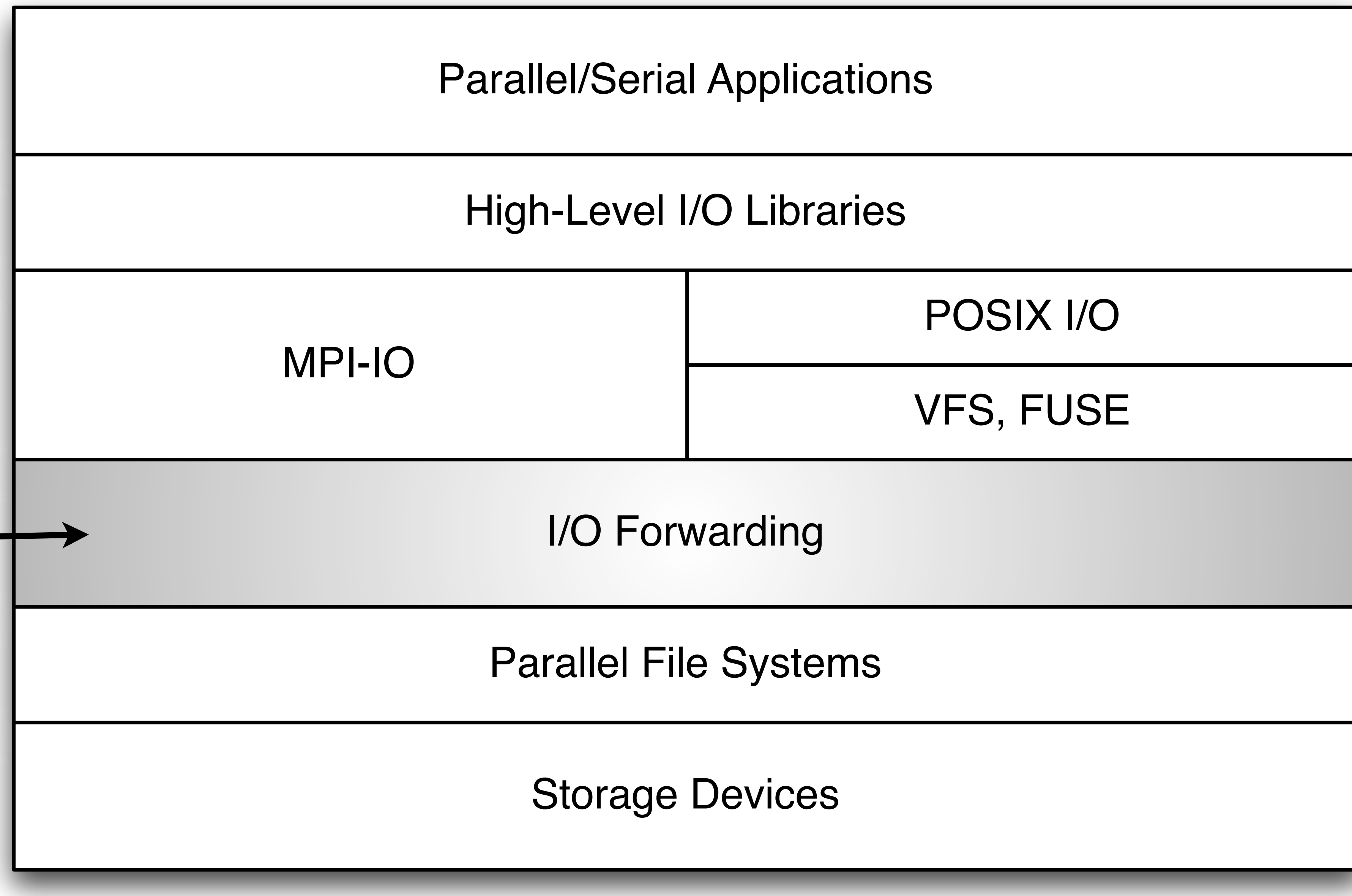
Challenge: Millions of Concurrent Clients

- 1,000,000+ concurrent clients present a challenge to current I/O stack
  - e,g. metadata performance, locking, network incast problem, etc.
- **I/O Forwarding Layer** is introduced.
  - All I/O requests are delegated to dedicated I/O forwarder process.
  - I/O forwarder reduces the number of clients seen by the file system for all applications, without collective I/O.

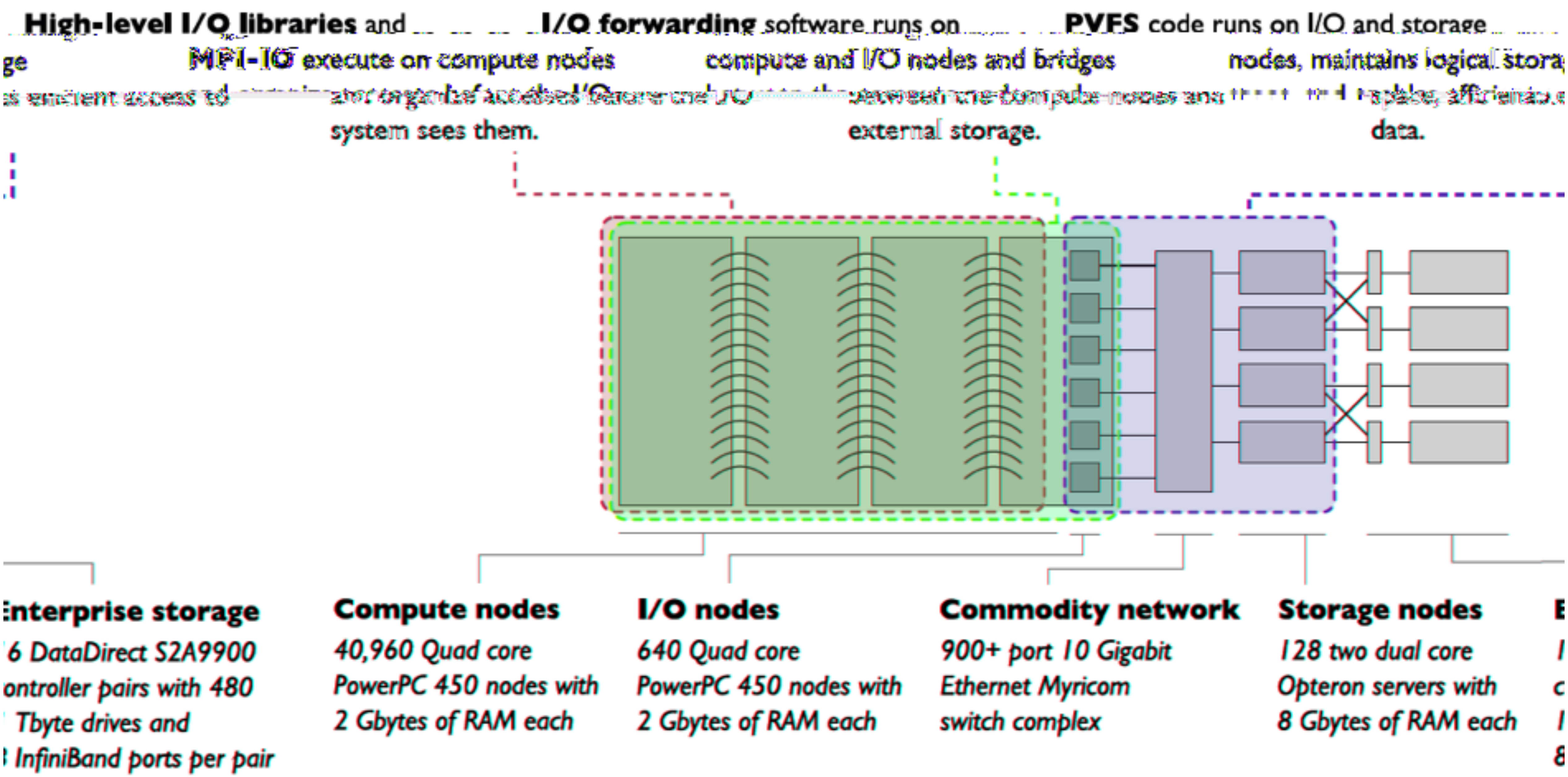


I/O Software Stack with I/O Forwarding

Bridge Between Compute
Process and Storage System
(IBM ciod, Cray DVS, IOFSL)



Example I/O System: Blue Gene/P Architecture



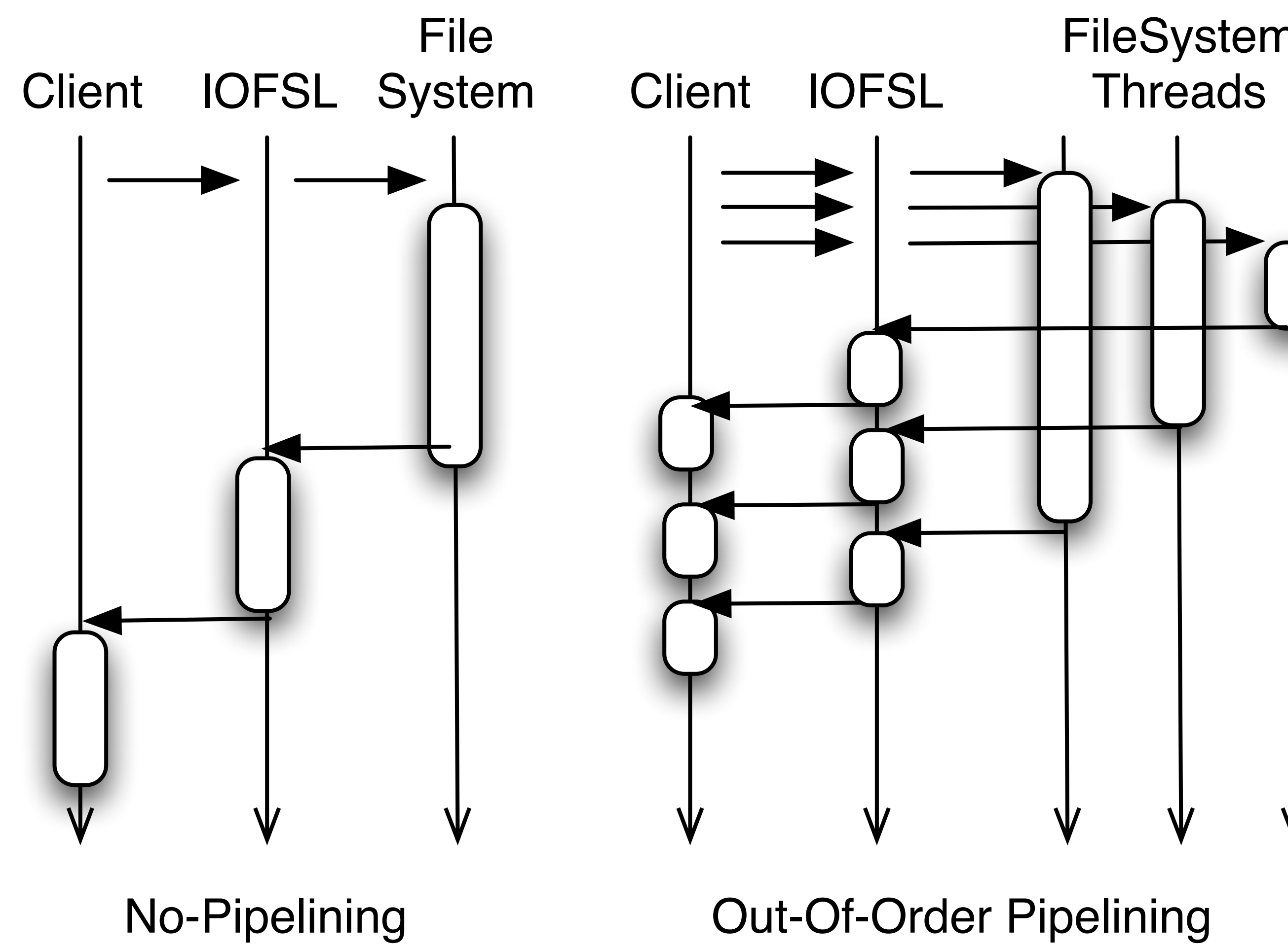
I/O Forwarding Challenges

- Large Requests
  - Latency of the forwarding
  - Memory limit of the I/O
  - Variety of backend file system node performance
- Small Requests
  - Current I/O forwarding mechanism reduces the number of clients, but does not reduces the number of requests.
  - Request processing overheads at the file systems

- We proposed two optimization techniques for the I/O forwarding layer.
  - **Out-Of-Order I/O Pipelining**, for large requests.
  - **I/O Request Scheduler**, for small requests.

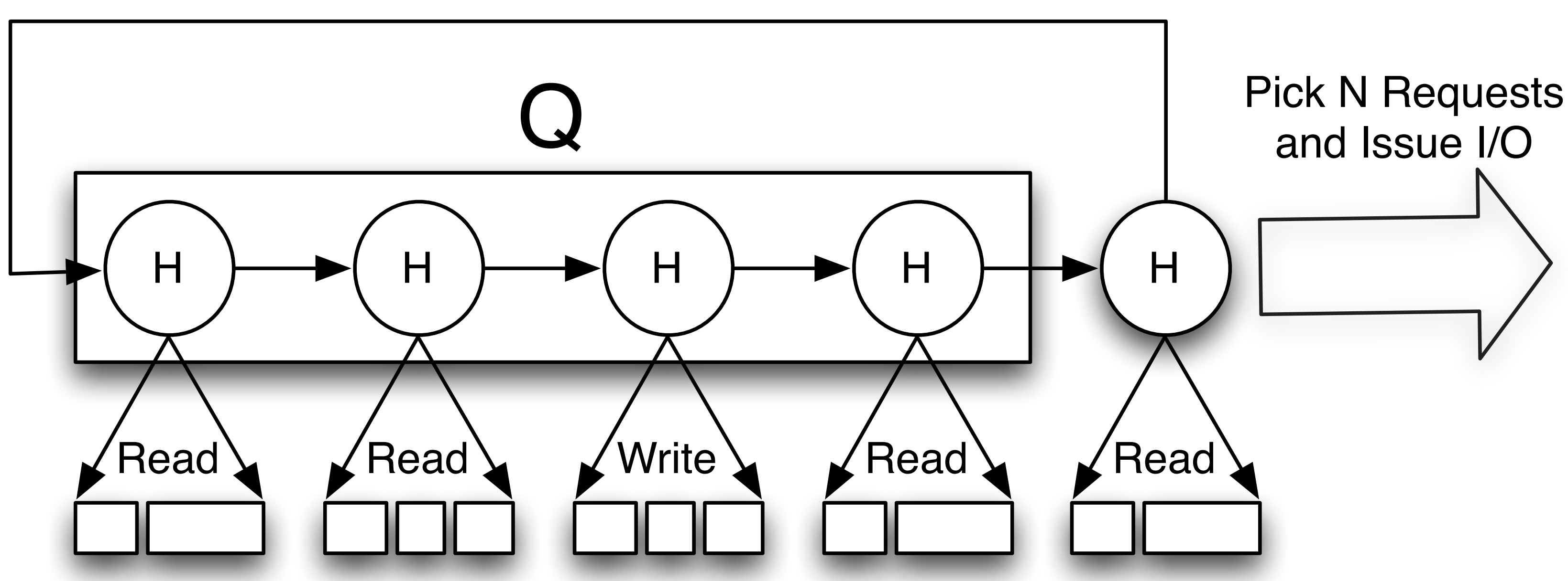
Out-Of-Order I/O Pipelining

- Split large I/O requests into small fixed-size chunks
- These chunks are forwarded in an out-of-order way.
- Good points
    - Reduce forwarding latency, by overlapping the I/O requests and the network transfer.
    - I/O sizes are not limited by the memory size at the forwarding node.
    - Little effect by the slowest file system node.



I/O Request Scheduler

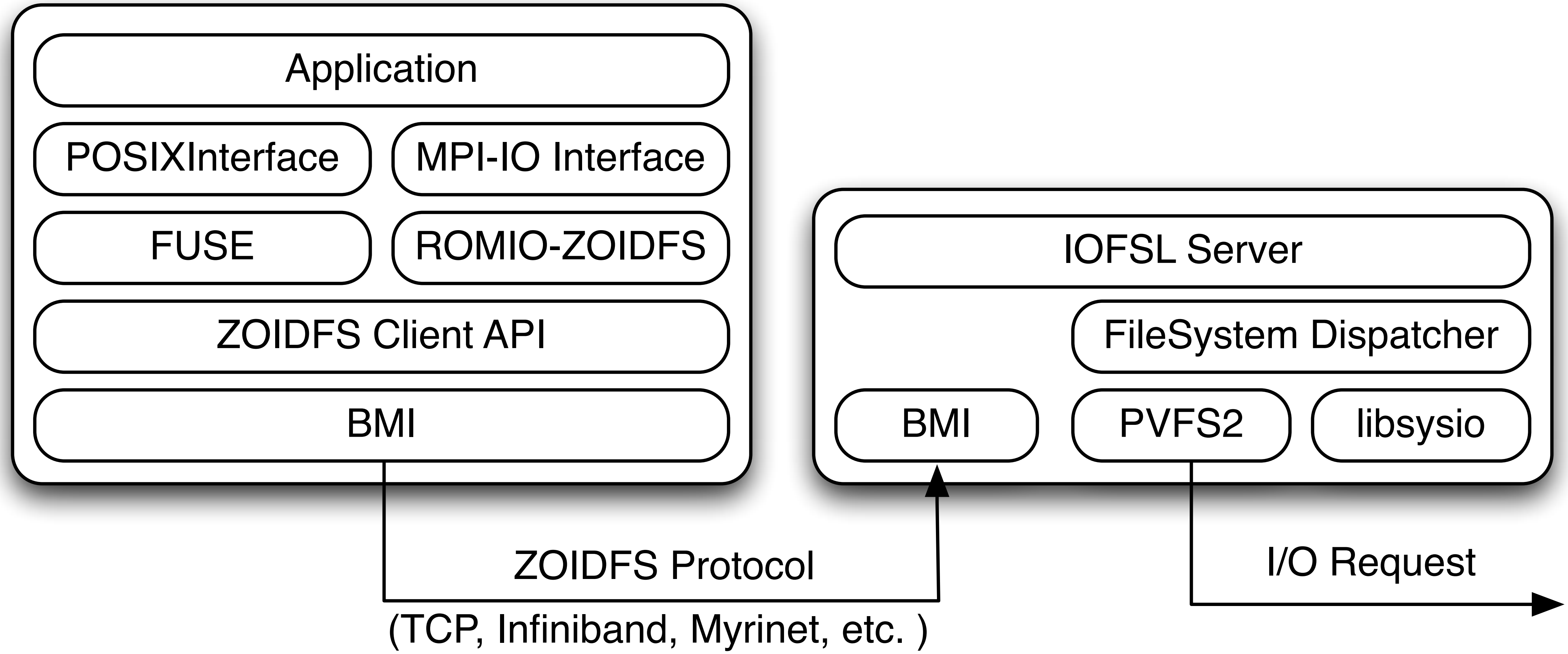
- Scheduling and Merging the small requests at the forwarder
  - Reduce number of seeks
  - Reduce number of requests, the file systems actually sees
- Scheduling overhead must be minimum
  - <u>Handle-Based Round-Robin algorithm</u> for the fairness between files
  - Ranges are managed by Interval Tree
    - The contiguous requests are merged



I/O Forwarding and Scalability Layer (IOFSL)

- IOFSL Project [Nawab 2009]
  - Open-Source I/O Forwarding Implementation
  - http://www.iofsl.org/
- Portable on most HPC environment
  - Network Independent
    - All network communication is done by BMI [Carns 2005]
      - TCP/IP, Infiniband, Myrinet, Blue Gene/P Tree, Portals, etc.
  - File System Independent
  - MPI-IO (ROMIO) / FUSE Client

IOFSL Software Stack



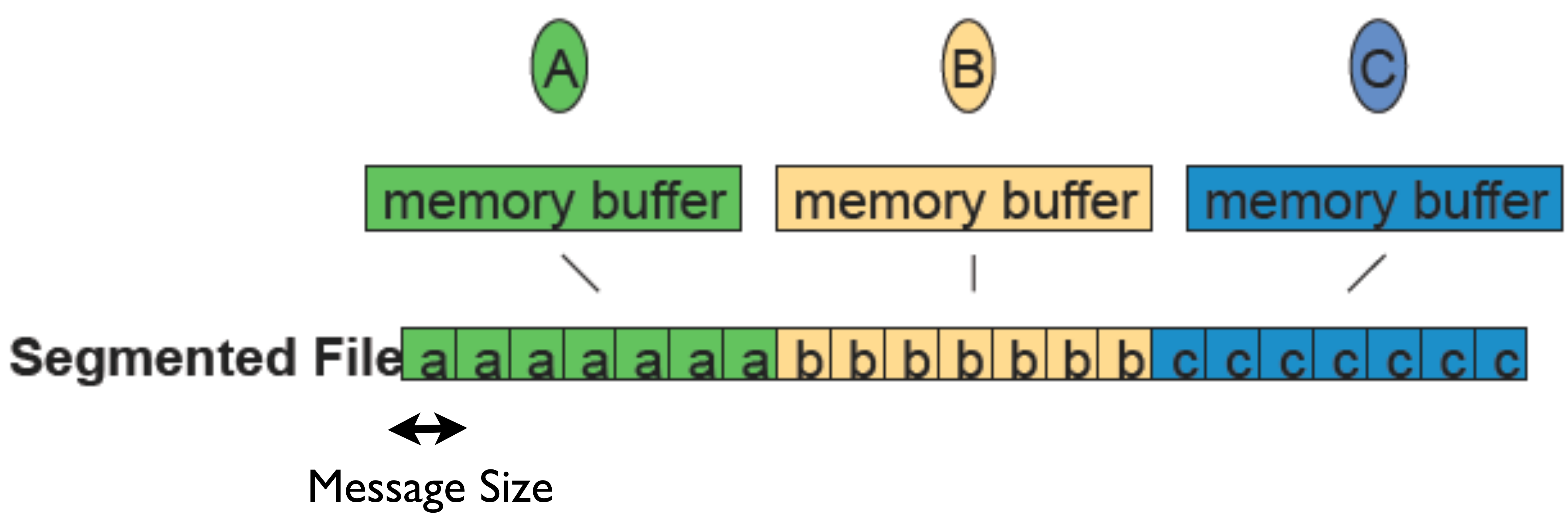
- <u>Out-Of-Order I/O Pipelining</u> and the <u>I/O request scheduler</u> have been implemented in the IOFSL, and evaluated on two environments.
  - T2K Tokyo (Linux Cluster), and ANL Surveyor (Blue Gene/P)

Evaluation on T2K: Spec

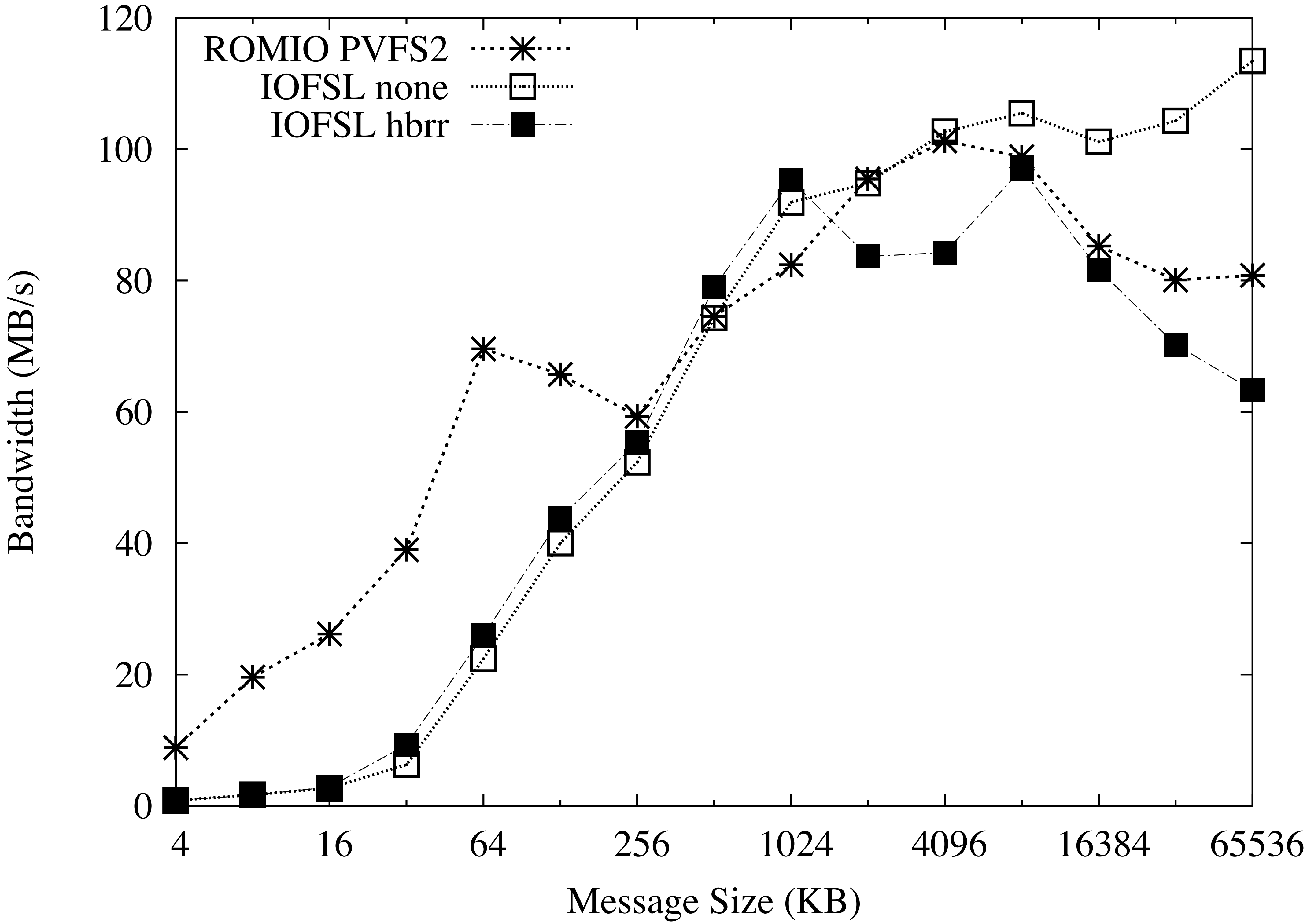
- T2K Open Super Computer, Tokyo Sites
  - http://www.open-supercomputer.org/
  - 32 node Research Cluster
  - 16 cores: 2.3 GHz Quad-Core Opteron*4
  - 32GB Memory
  - 10Gbps Myrinet Network
  - SATA Disk (Read: 49.52 MB/sec, Write 39.76 MB/sec)
- One IOFSL, Four PVFS2, 128 MPI Processes
- Software
  - MPICH2 1.1.1p1
  - PVFS2 CVS (almost 2.8.2)

Evaluation on T2K: IOR Benchmark

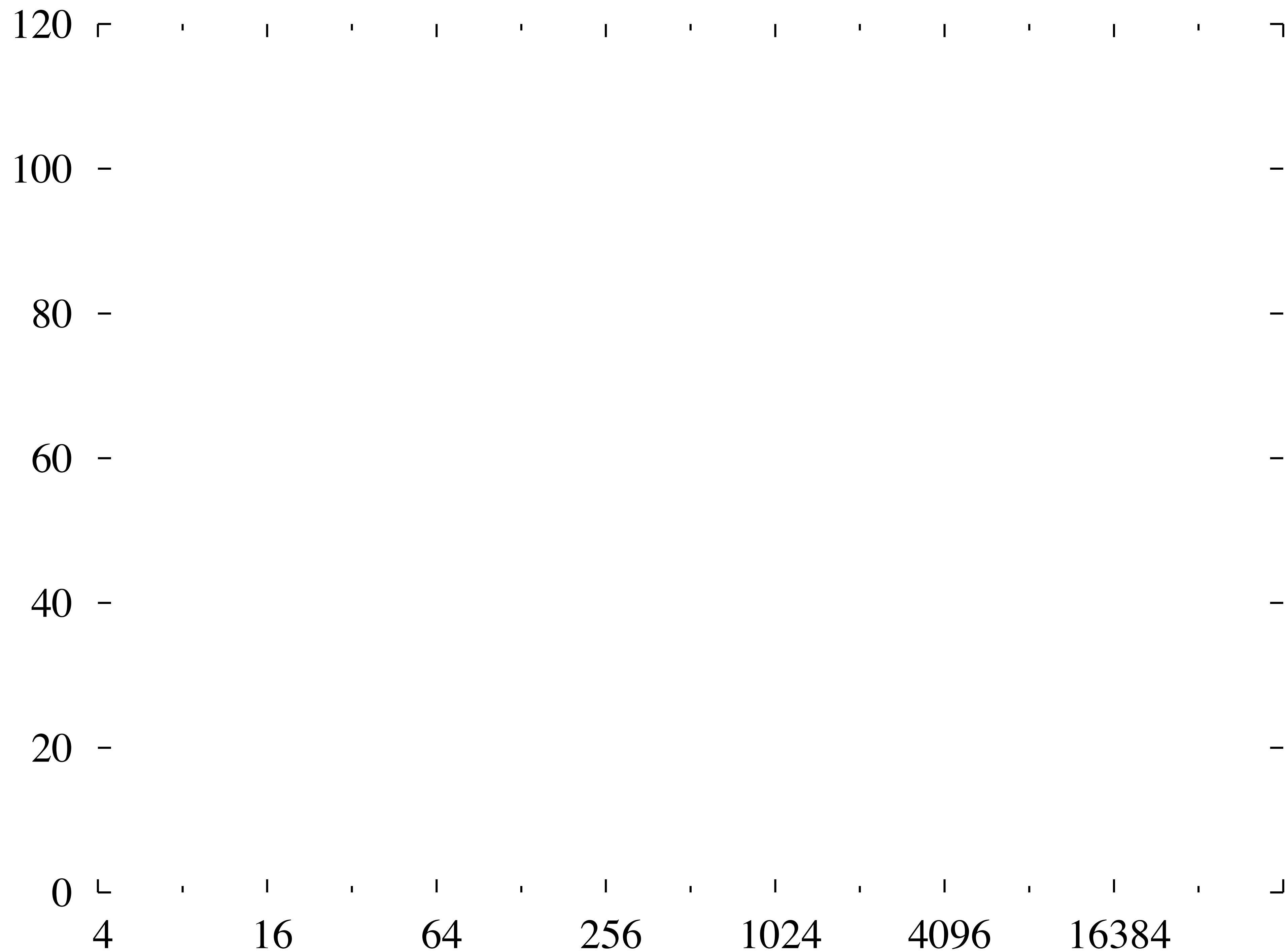
- Each process issues the same amount of I/O
- Gradually increasing the message size, and see the bandwidth change
  - Note: modified to do fsync() for MPI-IO



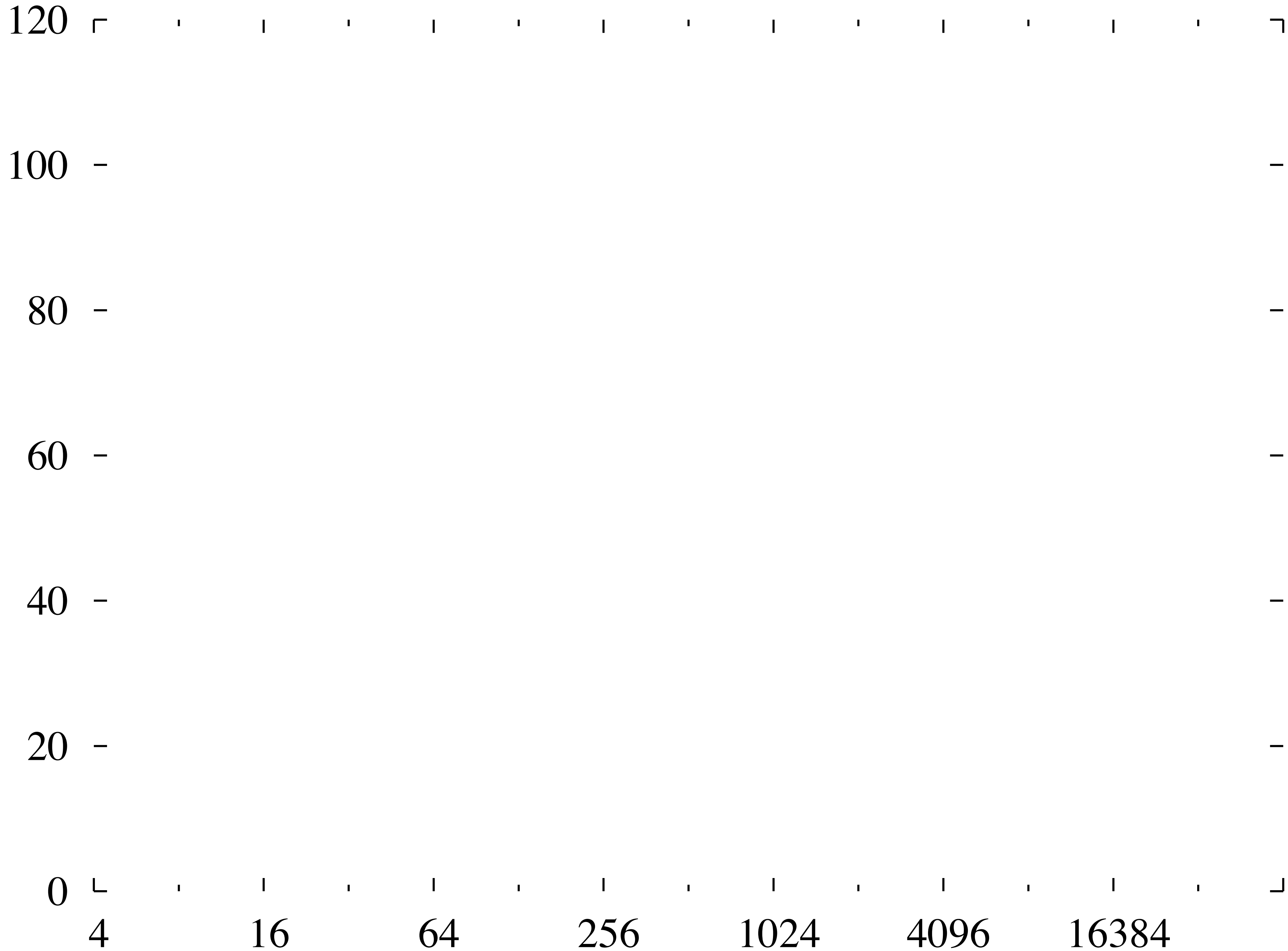
Evaluation on T2K: IOR Benchmark, 128procs



Evaluation on T2K: IOR Benchmark, 128procs

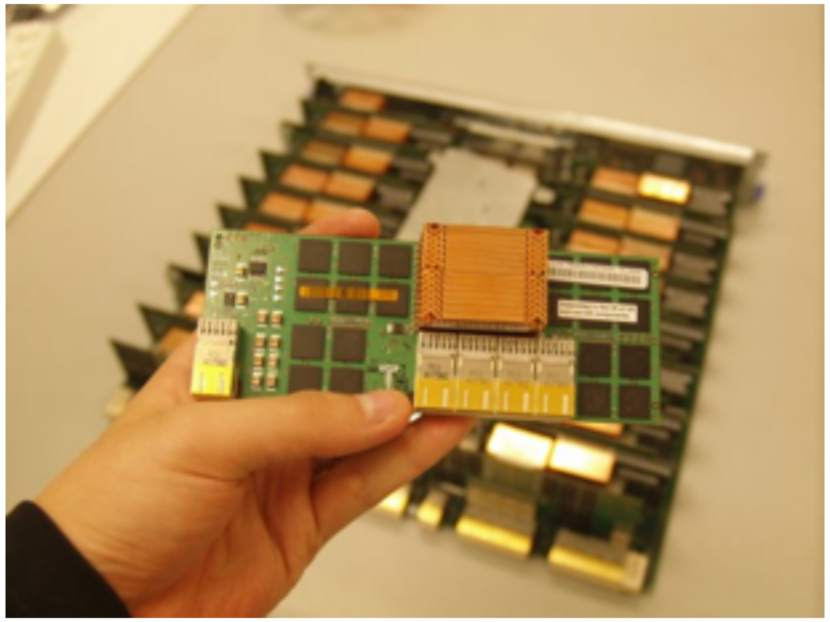


Evaluation on T2K: IOR Benchmark, 128procs

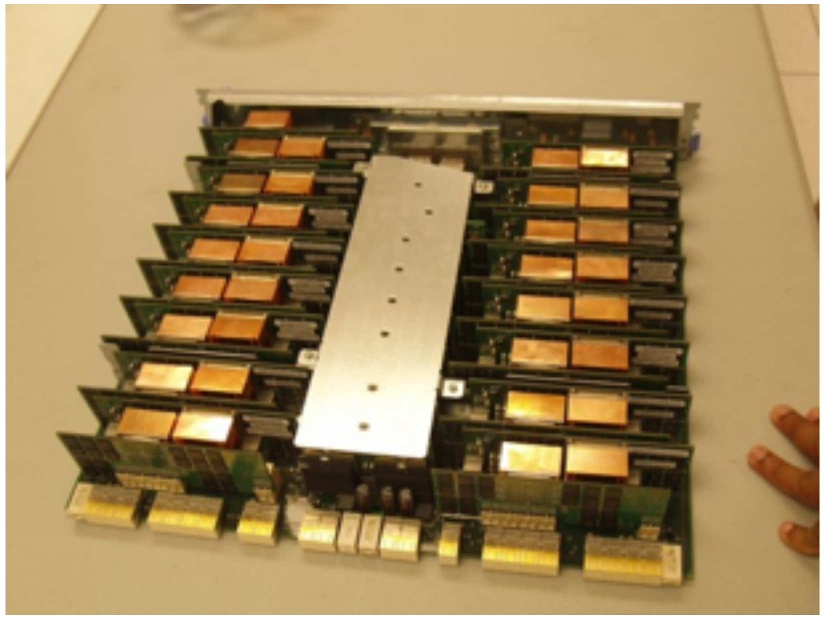


Evaluation on Blue Gene/P: Spec

- Argonne National Laboratory BG/P “Surveyor”
  - Blue Gene/P platform for research and development
  - 1024 nodes, 4096-core
  - Four PVFS2 servers
  - DataDirect Networks S2A9550 SAN
- 256 compute nodes, with 4 I/O nodes were used.



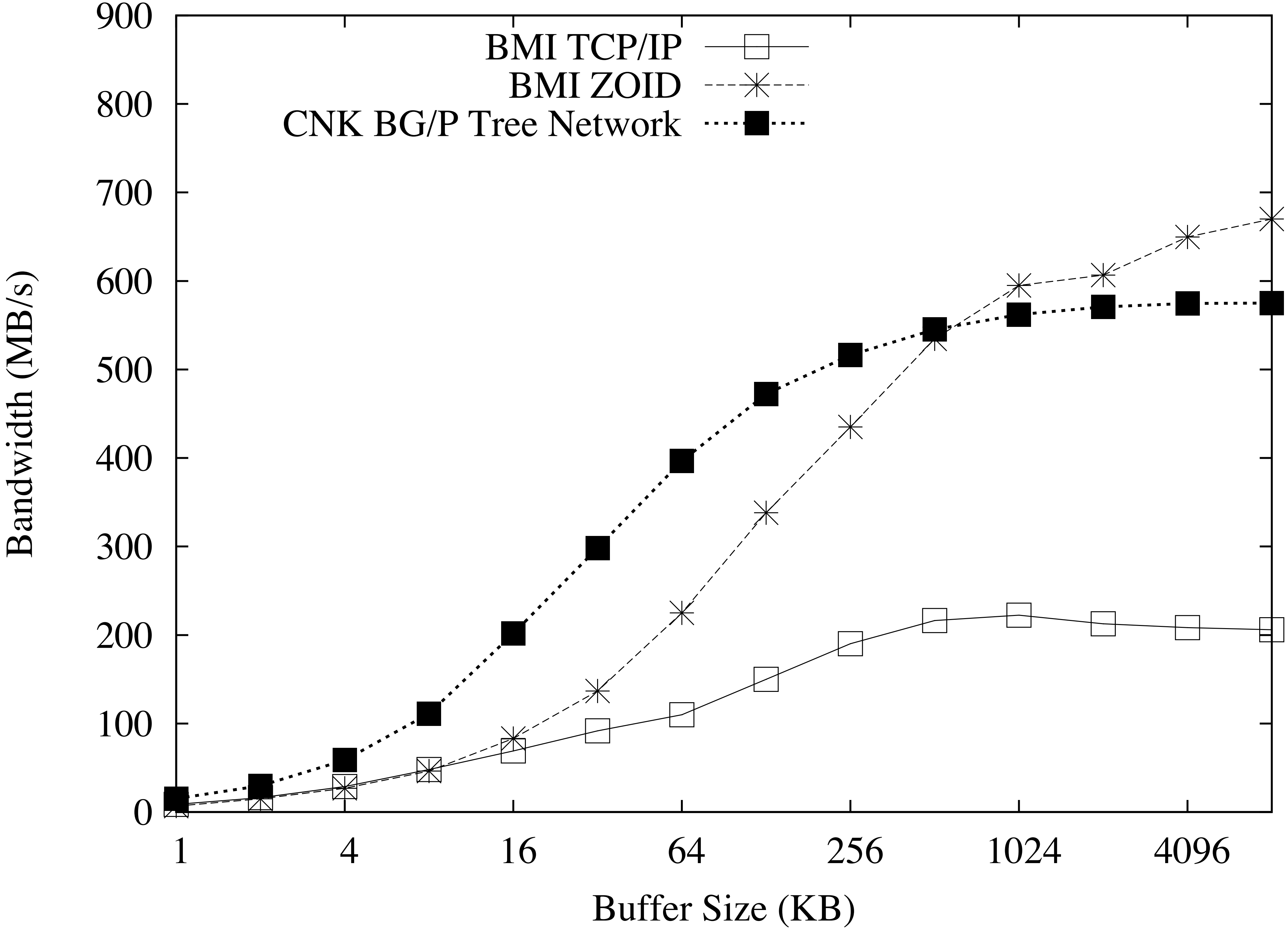
Node Card: 4 core



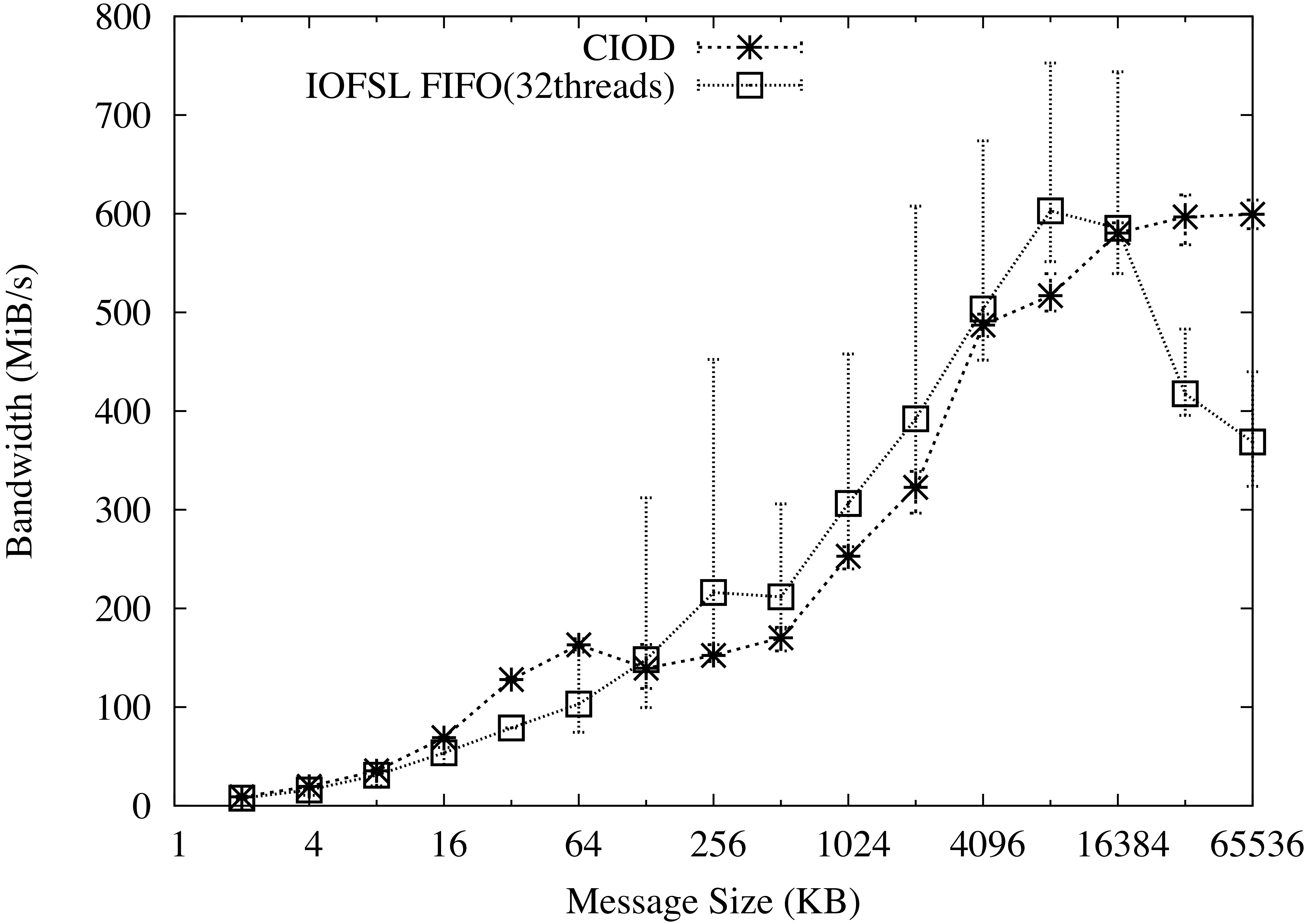
Node Board: 128 core

Rack: 4096 core

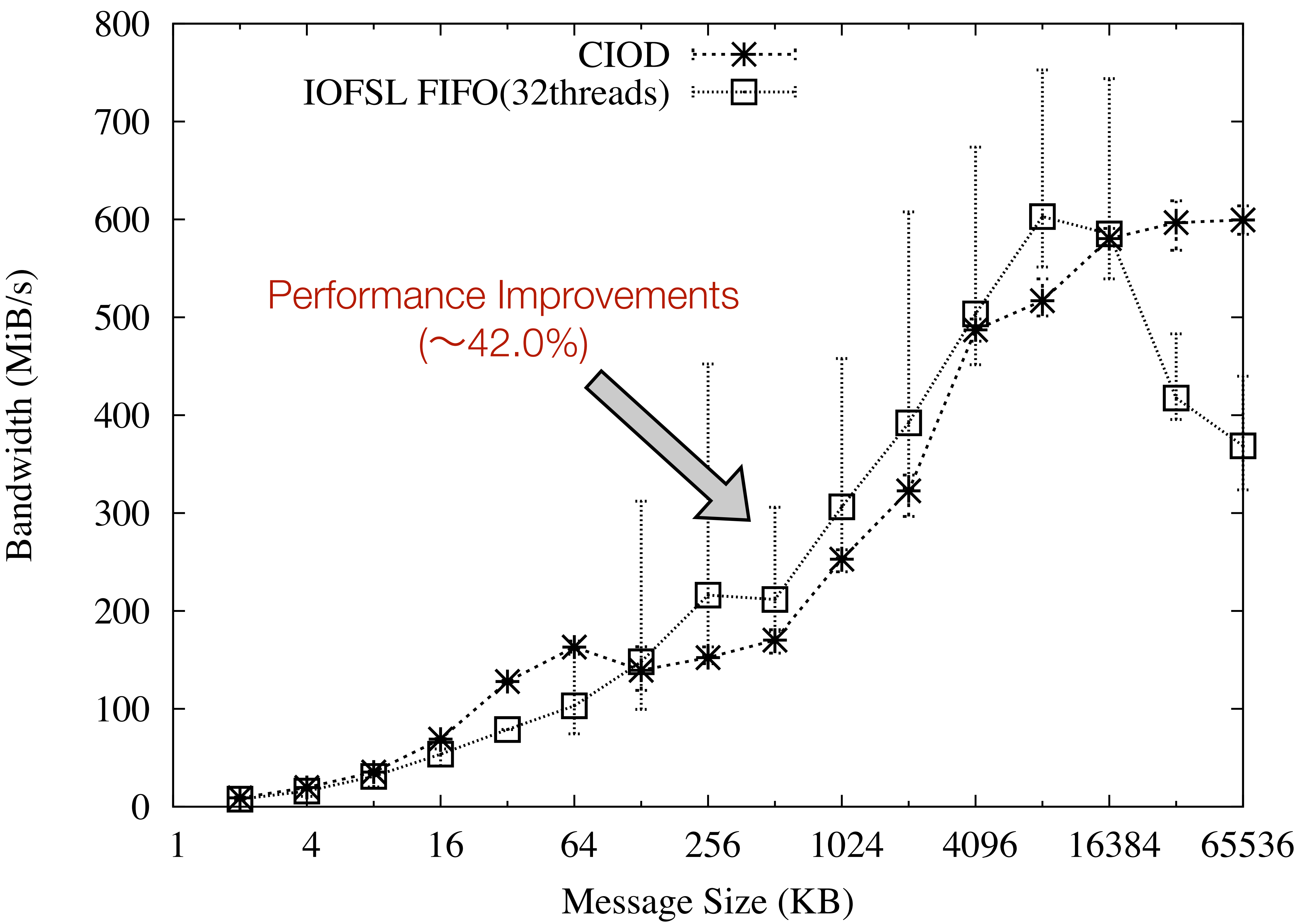
Evaluation on BG/P: BMI PingPong



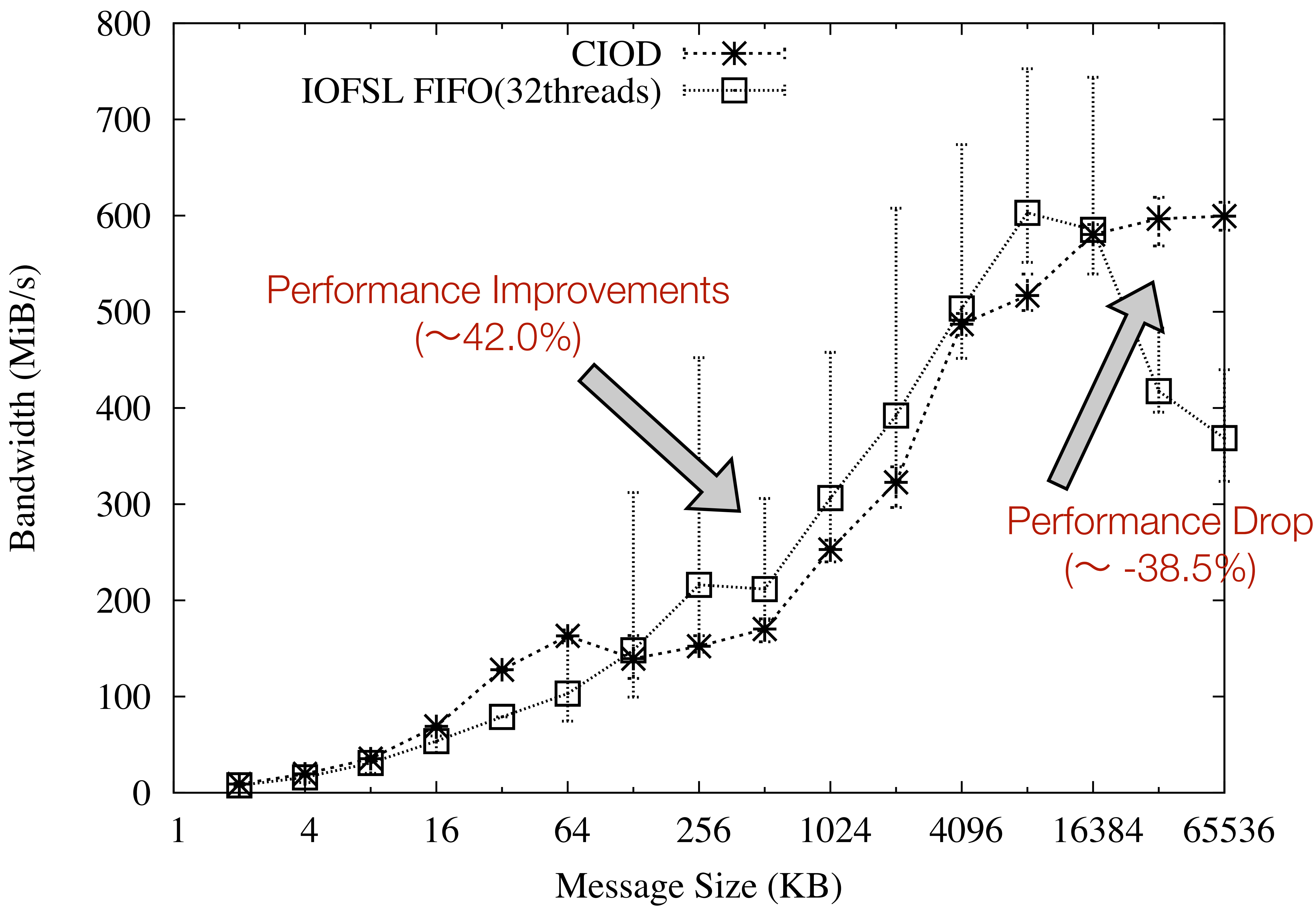
Evaluation on BG/P: IOR Benchmark, 256nodes



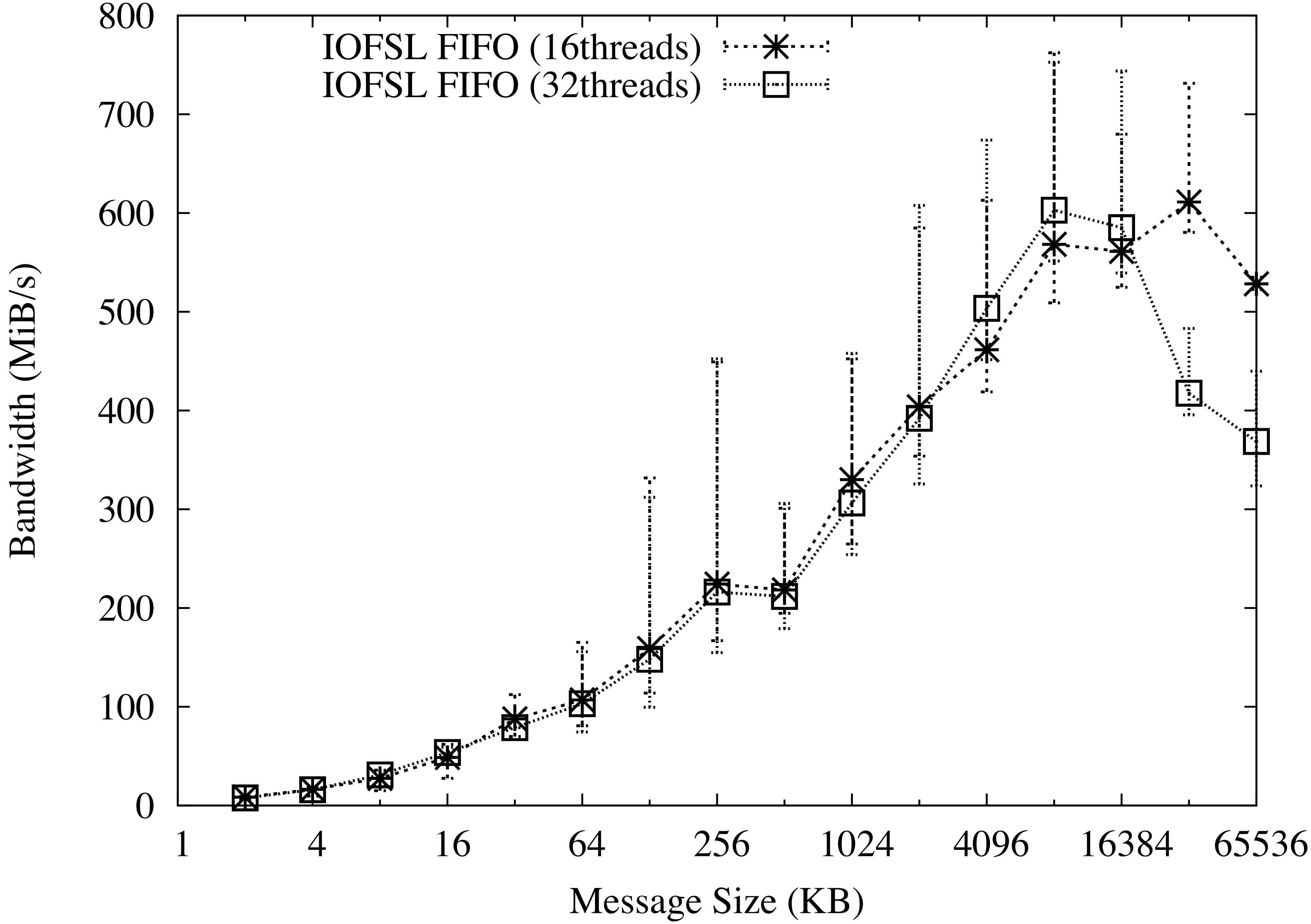
Evaluation on BG/P: IOR Benchmark, 256nodes



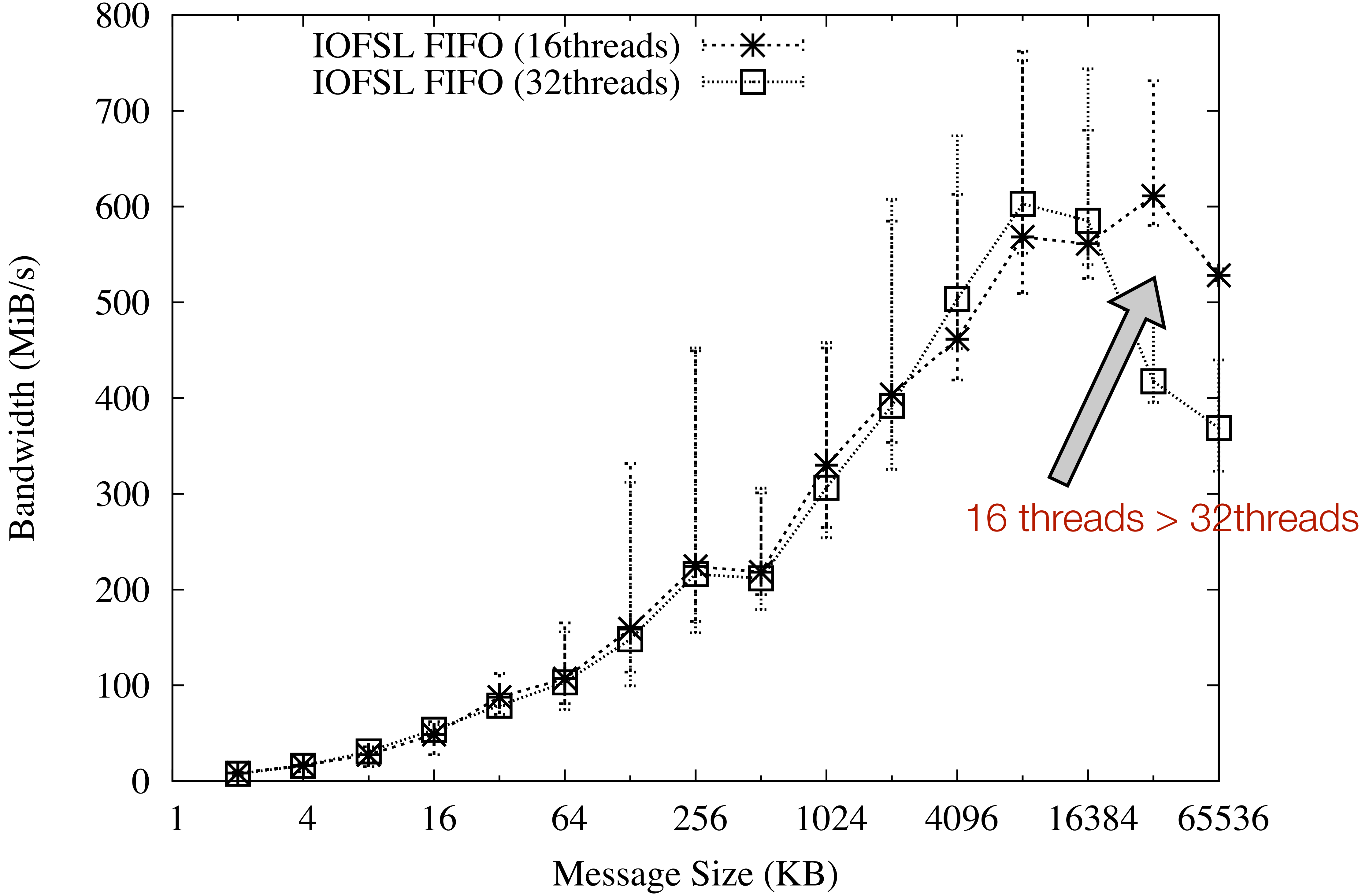
Evaluation on BG/P: IOR Benchmark, 256nodes



Evaluation on BG/P: Thread Count Effect



Evaluation on BG/P: Thread Count Effect



Related Work

- Computational Plant Project @ Sandia National Laboratory
  - First introduced I/O Forwarding Layer
- IBM Blue Gene/L, Blue Gene/P
  - All I/O requests are forwarded to I/O nodes
    - Compute OS can be stripped down to minimum functionality, and reduces the OS noise
  - ZOID: I/O Forwarding Project [Kamil 2008]
    - Only on Blue Gene
- Lustre Network Request Scheduler (NRS) [Qian 2009]
  - Request scheduler at the parallel file system nodes
  - Only simulation results

Future Work

- Event-driven server architecture
  - reduced thread contension
- Collaborative Caching at the I/O forwarding layer
  - multiple I/O forwarder works collaboratively for caching data and also metadata
- Hints from MPI-IO
  - Better cooperation with collective I/O
- Evaluation on other leadership scale machines
  - ORNL Jaguar, Cray XT4, XT5 systems

Conclusions

- Implementation and evaluation of two optimization techniques at the I/O Forwarding Layer
  - I/O pipelining that overlaps the file system requests and the network communication.
  - I/O scheduler that reduces the number of independent, non-contiguous file systems accesses.
- Demonstrating portable I/O forwarding layer, and performance comparison with existing HPC I/O software stack.
  - Two Environments
    - T2K Tokyo Linux cluster
    - ANL Blue Gene/P Surveyor
  - First I/O forwarding evaluations on linux cluster and Blue Gene/P
  - First comparison between BG/P IBM stack with OSS stack

Thanks!

Kazuki Ohta (presenter):
Preferred Infrastructure, Inc., University of Tokyo

Dries Kimpe, Jason Cope, Kamil Iskra, Robert Ross:
Argonne National Laboratory

Yutaka Ishikawa:
University of Tokyo

Contact: kazuki.ohta@gmail.com