Parallel I/O Experiences on an SGI 750 Cluster

Troy Baer
Science and Technology Support Group
High Performance Computing
Ohio Supercomputer Center
Columbus OH, USA
troy@osc.edu

OSC
Innovations in computing, networking, and education

Overview

- Motivation
- Systems Used
  - SGI 750 Itanium Cluster
  - SGI Origin 2000 Mass Storage Server
  - Parallel File System Cluster
  - Network Infrastructure
- Benchmarks
  - `bonnie`
  - ROMIO `perf`
  - NAS `btio`
  - ASCI Flash I/O
  - 2D Laplace Solver
- Conclusions and Future Work

OSC
Innovations in computing, networking, and education

Motivation

- Linux cluster systems are generally pretty effective at getting computations done in parallel.
  - Interconnects (Gigabit Ethernet, Myrinet, Quadrics) are getting reasonably fast
  - Cluster software (eg. MPICH, PBS/Maui) is maturing
- Storage is another matter entirely.
  - Potentially lots of unshared local disks
  - NFS performance limited by
    - protocol
    - capabilities of NFS server
    - quality of NFS client implementation
  - SAN storage doesn't scale to large numbers of nodes affordably (yet? ever?)
  - Relatively limited choices for parallel file systems
    - PVFS
    - GPFS (IBM customers only?)

Motivation (con't)

- Wanted to look at clustered Itanium performance WRT a variety of file systems:
  - ext2 on local disk
  - NFS
  - PVFS

Systems Involved

- SGI 750 Itanium Cluster (`ia64.osc.edu`)
- SGI Origin 2000 Mass Storage Server (`mss.osc.edu`)
- Parallel File System Cluster

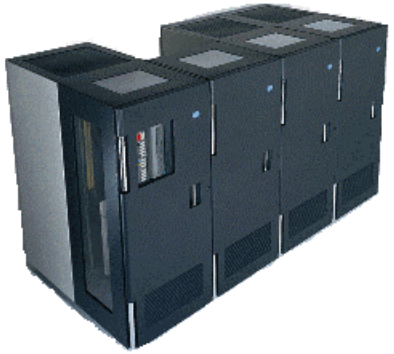
SGI 750 Itanium Cluster

- 1 Front end node
  - 2 Itanium 733MHz processors
  - 4 GB RAM
  - 350 GB of SCSI disks
  - NFS server for compute nodes' root FS
- 72 Compute nodes, each with
  - 2 Itanium 733MHz processors
  - 4 GB RAM
  - 18 GB local SCSI disk (~5 GB in `/tmp`)
- Networks
  - Myrinet 2000 for MPI
  - Gigabit Ethernet for parallel I/O and interconnect research
  - 100Mbit Ethernet for NFS and administration

OSC
Innovations in computing, networking, and education

Mass Storage Server

- SGI Origin 2000
  - 8 processors
  - 4 GB RAM
  - Multiple Gigabit Ethernet and HiPPI network interfaces
- 1 TB of Fibre Channel RAID arrays



- IBM 3494 tape robot
  - 6 cabinets, ~6000 tapes
  - 4 tape drives
- DMF/TMF for hierarchical stroage management
- NFS server for users' home directories to all OSC HPC systems
  - Cray SV1ex (16 processors)
  - SGI Origin 2000 (32 processors)
  - Itanium cluster (144[+2] processors)
  - Athlon cluster (128[+4] processors)
  - 4 Sun 6800s (72 processors)

OSC
Innovations in computing, networking, and education

Parallel File System Cluster

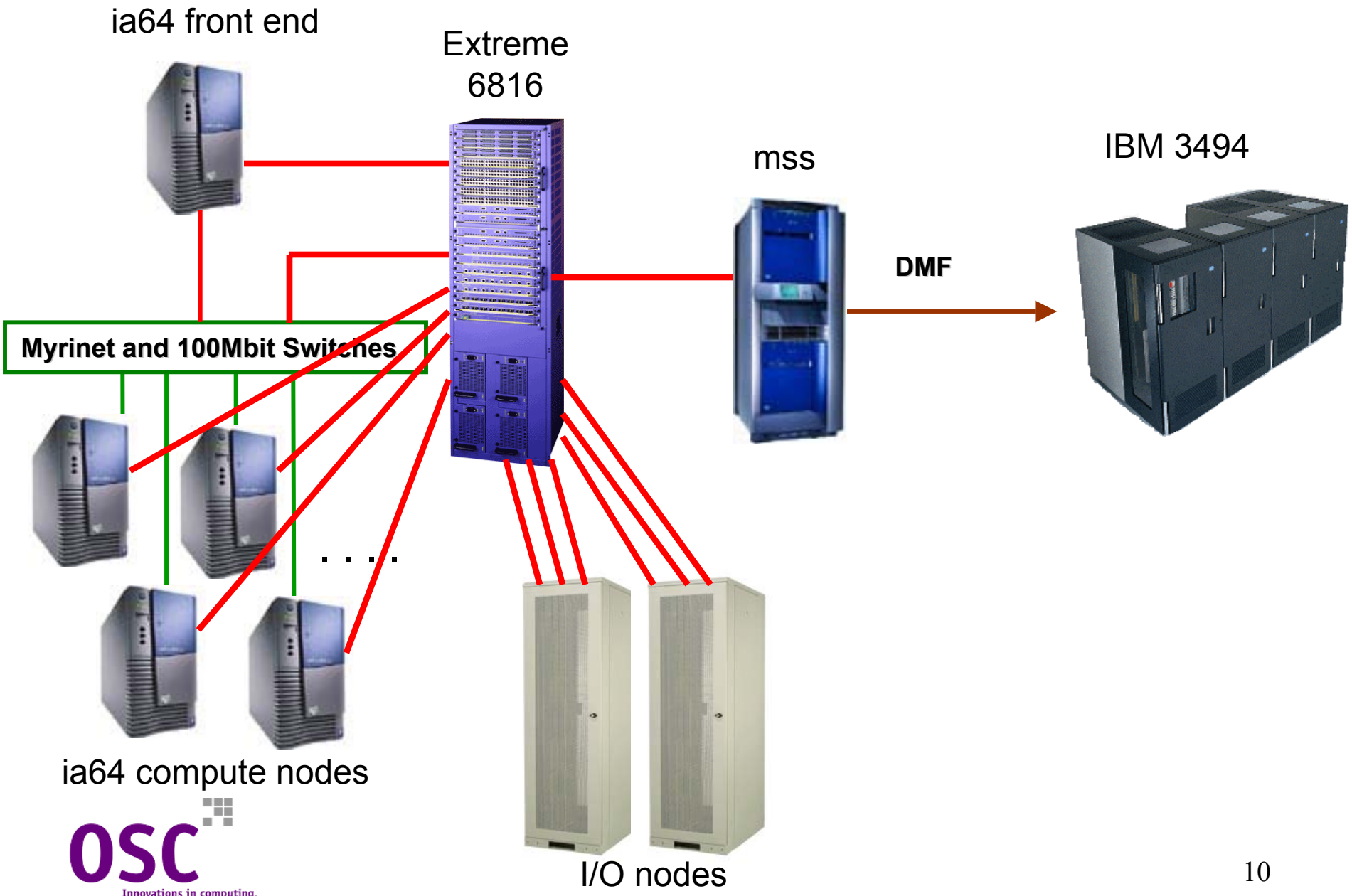
- 16 I/O nodes, each with
  - 2 Pentium III 933MHz CPUs
  - 1 GB RAM
  - 3ware 7810 ATA RAID controller
  - 8 80-GB ATA disks in RAID-5 (~520 GB usable)
  - Gigabit Ethernet and 100Mbit Ethernet
- PVFS software from Clemson Univ. and Argonne National Lab
  - Equivalent of RAID-0 across I/O nodes
  - 7.83 TB usable space, 2 GB/s max. I/O bandwidth
  - Large block size -- 64kB default, configurable at file creation time
  - Two interfaces for users
    - Linux kernel driver and user-space daemon for POSIX-style semantics (`ls`, `cp`, etc.)
    - ROMIO driver for high-performance MPI-IO

Network Infrastructure

- Extreme Networks Black Diamond 6816 Gigabit Ethernet switch
  - 10 12-port Gigabit Ethernet line cards
  - 4 8-port Gigabit Ethernet line cards
  - 2 48-port 100Mbit Ethernet cards

OSC
Innovations in computing, networking, and education

Network Topology



OSC
Innovations in computing, networking, and education

Benchmarks

- `bonnie`
- ROMIO `perf`
- NAS `btio`
- ASCI Flash I/O
- 2D Laplace Solver

OSC
Innovations in computing, networking, and education

bonnie

- Widely used benchmark for performance of UNIX-style file systems
- Has some drawbacks on large, modern systems
  - Uses a C `int` for file offsets -- max file size of 2 GB on systems where `int` is 32 bits (including Linux/ia64)
  - Uses an 8 kB buffer for doing block reads and writes -- massive overhead for file systems with block sizes larger than 8 kB such as PVFS
- OSC-developed variant called `bigbonnie`
  - Uses a C `long long` for file offsets
  - Large File Summit `#defines` to deal with > 2 GB files on 32-bit platforms
  - 64 kB buffer for block reads and writes
- Test against three file systems:
  - `/tmp` (local SCSI disk)
  - `/home` (NFS over 100Mbit Ethernet to Gigabit Ethernet on mss)
  - `/pvfs` (PVFS over Gigabit Ethernet to I/O nodes)

bonnie Results

- Stock bonnie:

```
              ----------Sequential Output-------- ---Sequential Input--- ---Random---
               -Per Char- ---Block---  -Rewrite-- -Per Char- ---Block--- ----Seeks---
Filesys    MB K/sec %CPU  K/sec %CPU  K/sec %CPU K/sec %CPU  K/sec %CPU  /sec    %CP
/tmp     2000  6099 99.2 262477 99.9 327502 99.9  3091 99.9 552000 99.9 90181.5 193.
/home    2000   147  3.7   1562  2.3    859  1.7  1909 62.7 542824 99.9   805.5   3.
/pvfs    2000  3909 63.5    199  1.1    201  2.4  2816 92.9   6614 36.3   227.2  12.
```

- bigbonnie:

```
              -------Sequential Output------- ---Sequential Input-- --Random--
              -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
Filesys    MB  MB/s %CPU  MB/s %CPU  MB/s %CPU  MB/s %CPU  MB/s %CPU  /sec %CPU
/tmp     4096   5.9 99.6  24.6 10.1  17.2  7.1   3.9 96.6  34.2  9.7 452.9  6.8
/home    8192   0.1  3.2   1.5  2.0   0.8  1.4   1.7 42.3   3.0  1.8  74.6  3.5
/pvfs    8192   3.8 63.3  11.0  7.3  16.0 25.7   3.4 91.2  32.8 25.6 583.3 30.1
```

OSC
Innovations in computing, networking, and education

Giving Up on NFS and Local Disk

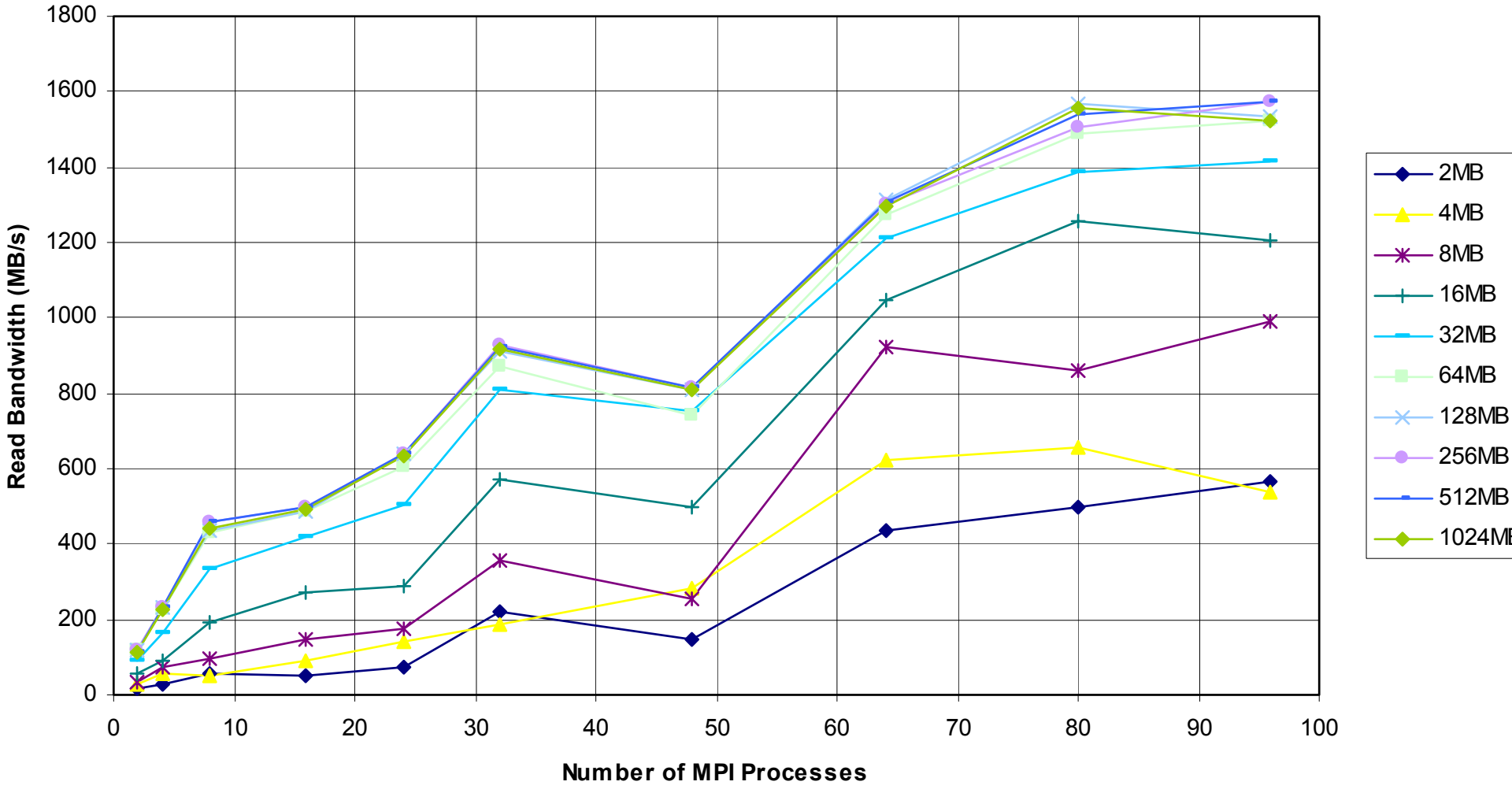
- The compute nodes don't have enough local space to be interesting.
  - Fast but relatively small -- only ~5 GB in `/tmp`, not much bigger than memory
  - Not shared between nodes
  - PVFS is about as fast as local disk for large read/write buffer sizes, and is shared
- NFS over 100Mbit Ethernet to `mss` has mediocre performance.
  - ~1.5 MB/s for writes, ~3 MB/s for reads
  - Possibly skirting the edges of IRIX NFS server scalability (currently ~160 clients)?
  - NFS over Gigabit Ethernet using jumbo frames could improve this, but would disrupt some of the other research efforts on the Gigabit network such as EMP (Ethernet Message Passing)

- All further tests, all of which use MPI-IO, are done only to `/pvfs`.

ROMIO `perf`

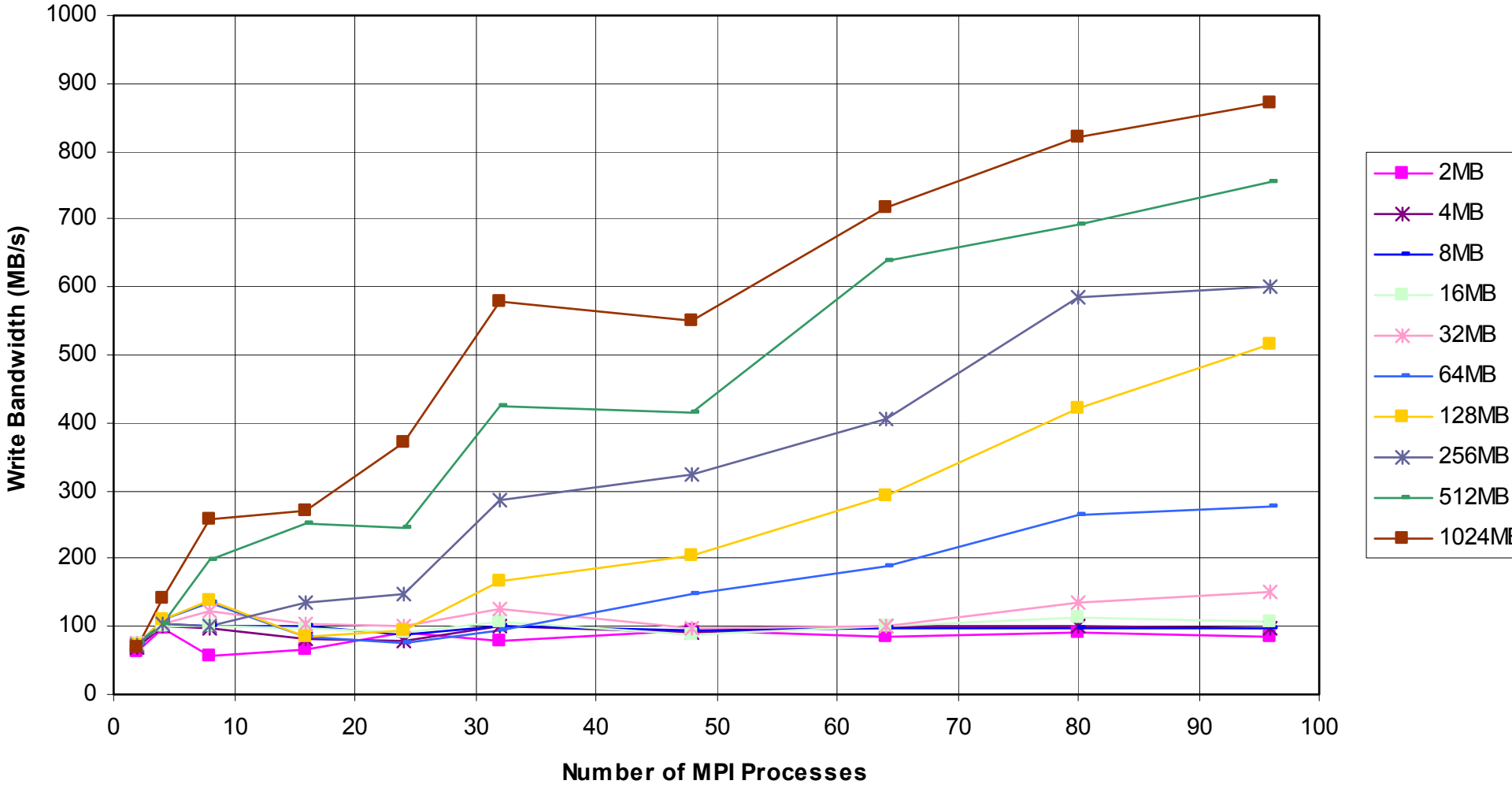
- ROMIO is the reference implementation of MPI-IO from Argonne National Lab
- It includes an example called `perf` which measures MPI-IO performance in four simple cases:
  - `MPI_File_write()`
  - `MPI_File_read()`
  - `MPI_File_write()` followed by `MPI_File_sync()`
  - `MPI_File_sync()` followed by `MPI_File_read()`
- This gives an approximate upper bound on the performance that can be sustained to the file system (in this case PVFS) with an MPI-IO application.
- Data array size is 4 MB per process by default; OSC's tests varied this from 2 MB to 1 GB.

ROMIO `perf` -- Read Performance



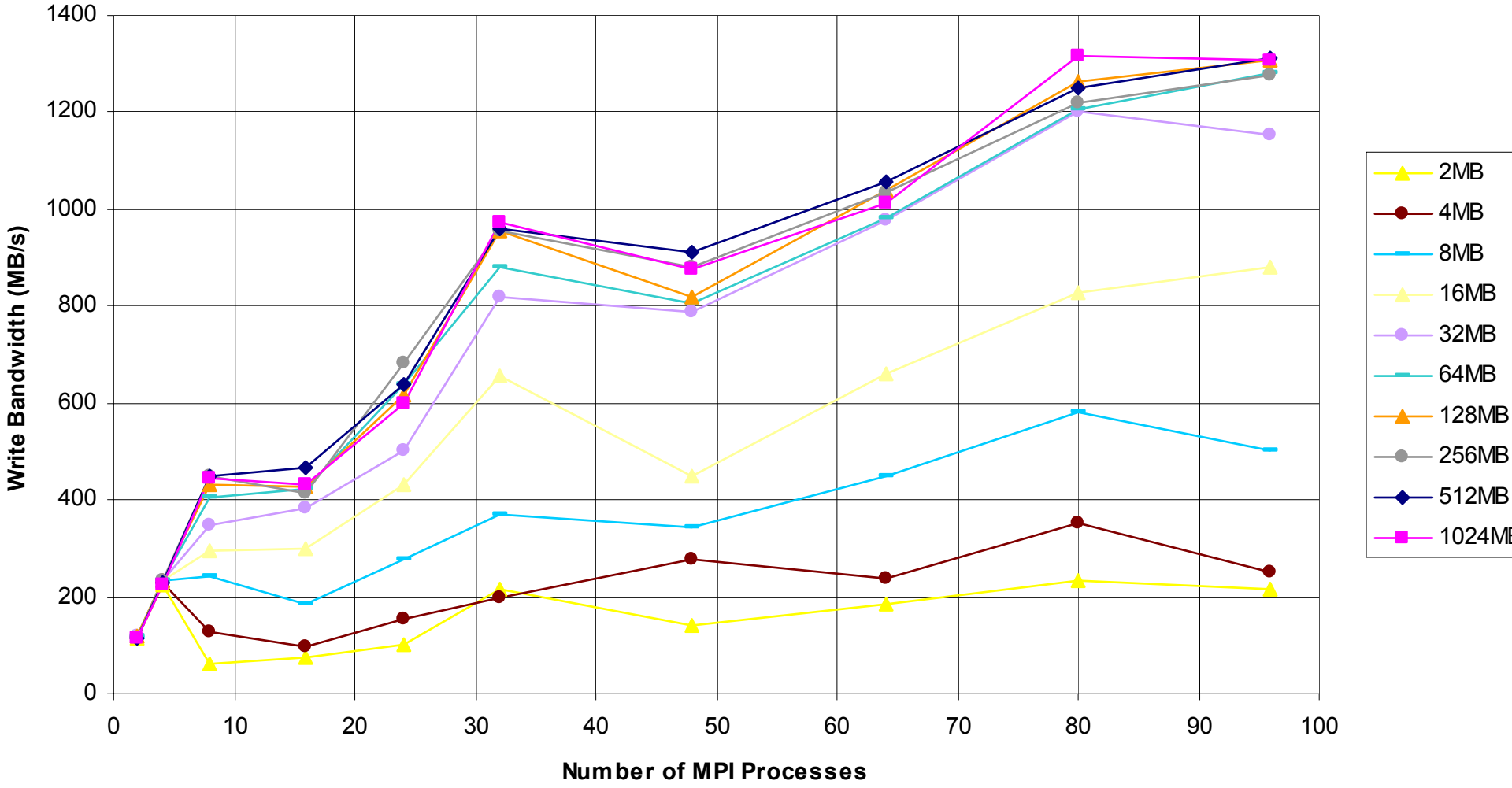
OSC
Innovations in computing, networking, and education

ROMIO `perf` -- Write Performance w/ Sync



OSC
Innovations in computing, networking, and education

ROMIO `perf` -- Write Performance w/o Sync

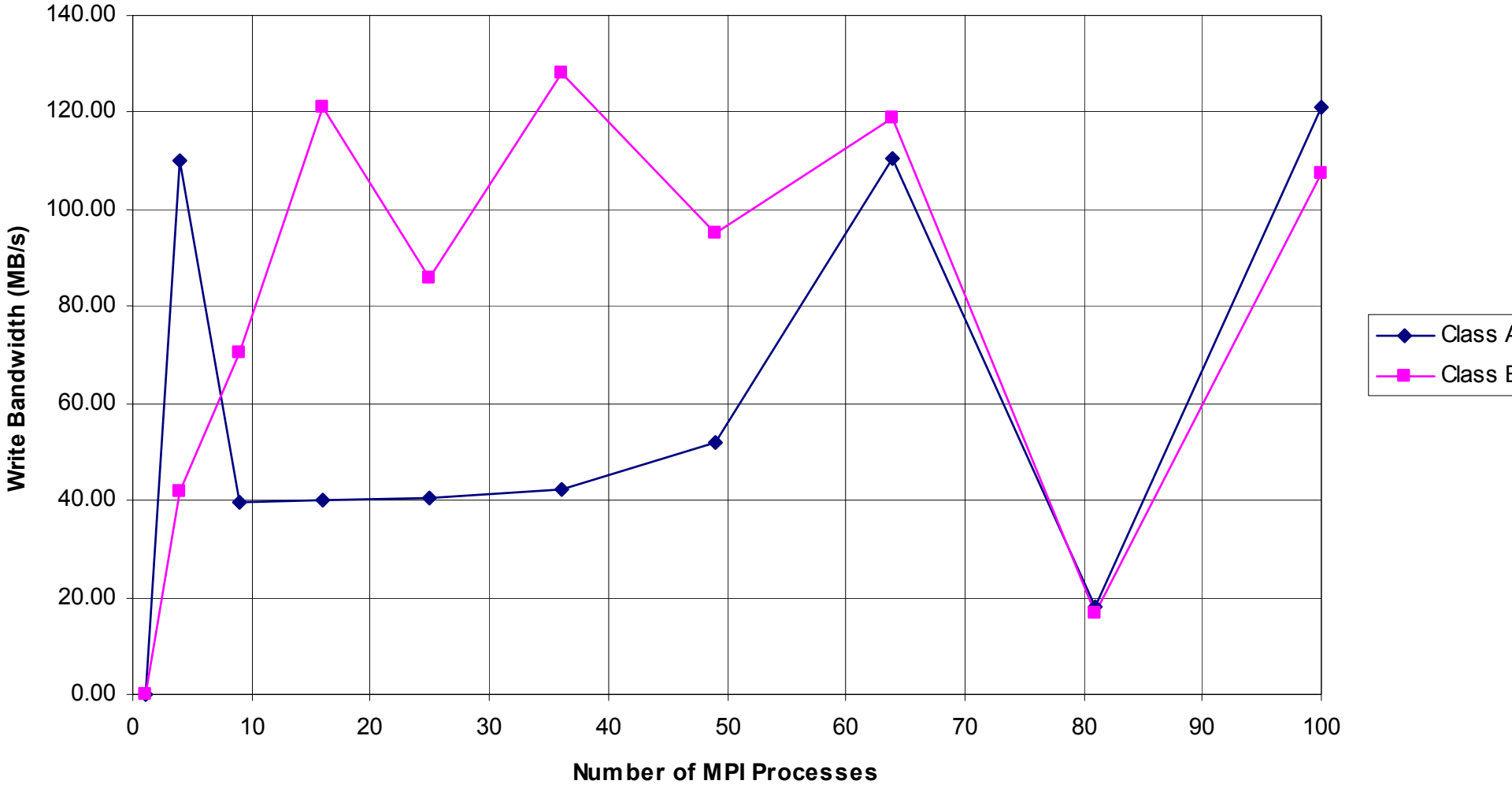


OSC
Innovations in computing, networking, and education

NAS `btio`

- `bt` is one of the original NAS Parallel Benchmarks
  - `bt` == "block tridiagonal"
  - Simulates behavior of many CFD codes based on implicit numerical schemes
- `btio` adds parallel I/O using MPI-IO
  - Periodic checkpointing of solution values
  - Two versions
    - "simple" -- Several calls to `MPI_File_write_at()` per checkpoint, no use of MPI derived data types
    - "full" -- One call to `MPI_File_write_at()` per checkpoint, with an MPI derived data type used to organize the data
  - Three configurations
    - Class A -- 62^3 volume
    - Class B -- 102^3 volume
    - Class C -- 162^3 volume, would not run due to a bug in the version of ROMIO included in the current MPICH/ch_gm from Myricom

NAS `btio` "full" Performance

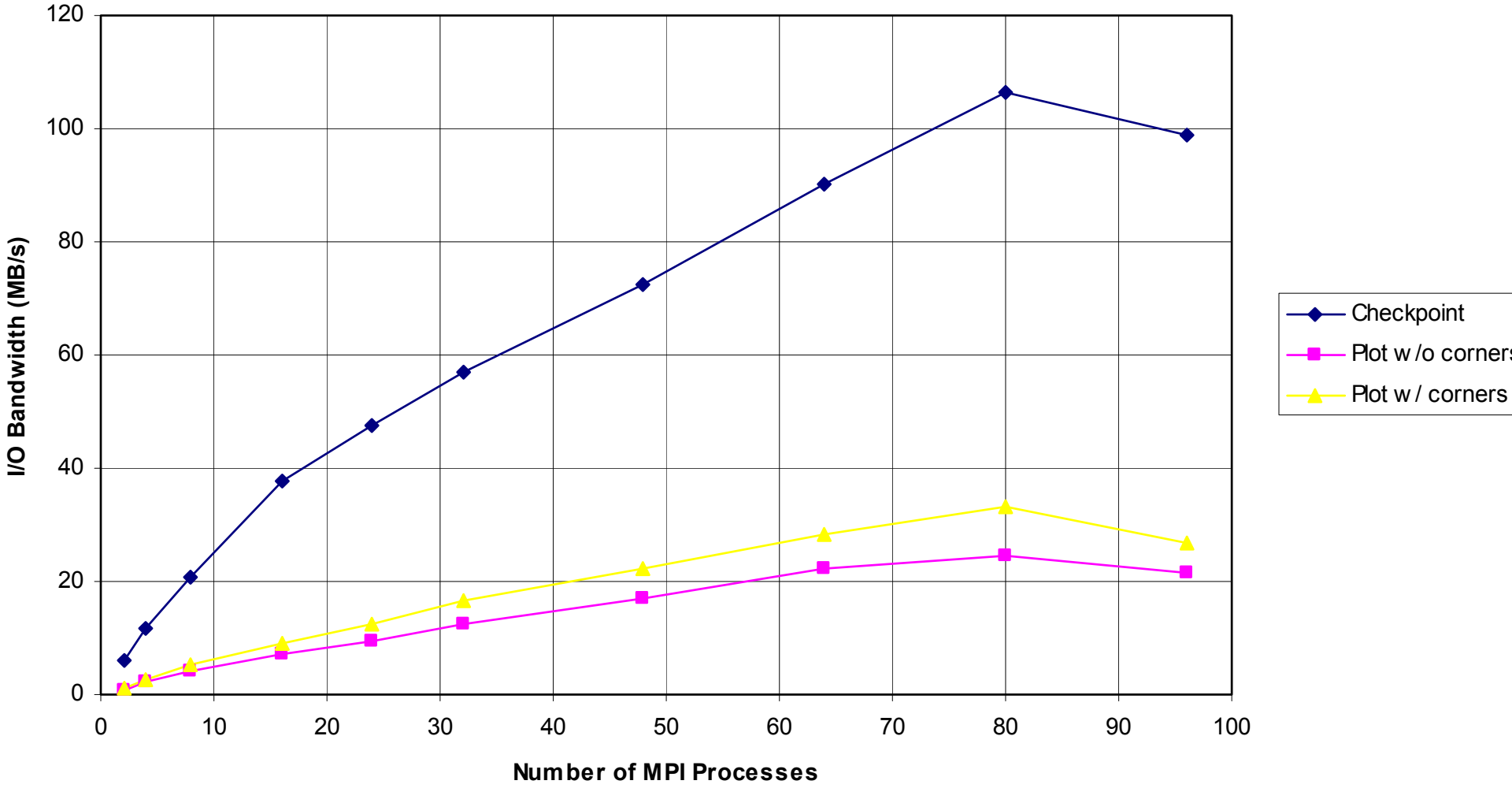


OSC
Innovations in computing, networking, and education

ASCI Flash I/O

- ASCI Flash is an application used to simulate astrophysical thermonuclear flashes.
  - Developed at University of Chicago
  - Used on several on the USDOE ASCI systems
- Uses the parallel interface to HDF5 over MPI-IO.
  - Checkpoint files
  - Cell- and corner-based plot files
  - Significant amount of overhead involved in data type processing at HDF5 level
- I/O portion of the code has been separated out into the Flash I/O benchmark.

ASCI Flash I/O Performance



OSC
Innovations in computing, networking, and education

ASCI Flash I/O Comparison with Other Sites

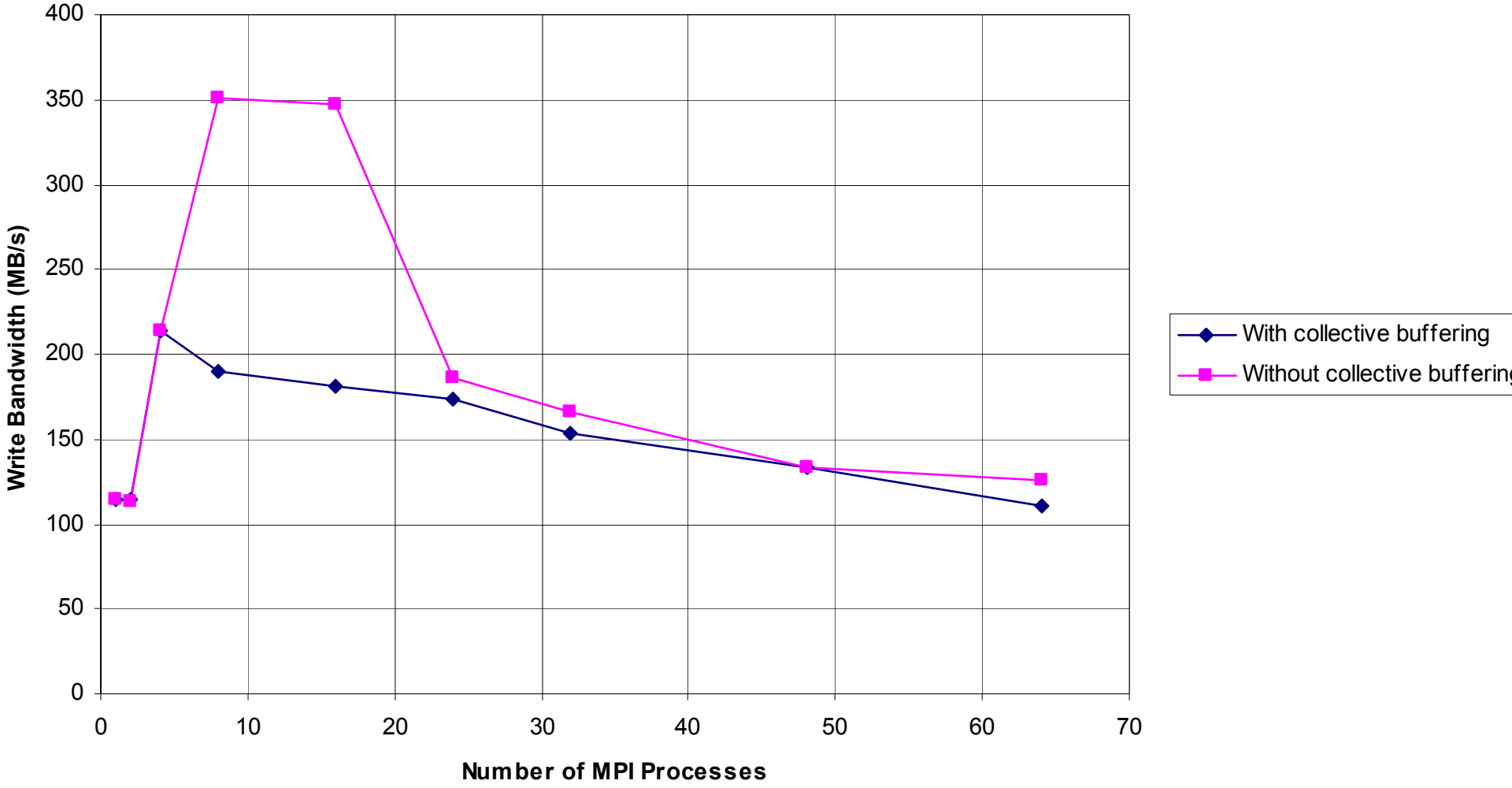
Site	System	File System	Max Checkpoint Performance
LLNL	ASCI Blue Pacific	GPFS	21.3 MB/s @ 64 procs
ANL	Chiba City	PVFS	57.4 MB/s @ 256 procs
OSC	SGI Itanium Cluster	PVFS	106.4 MB/s @ 80 procs
LLNL	Frost	GPFS	212.0 MB/s @ 768 procs

OSC
Innovations in computing, networking, and education

2D Laplace Solver

- The STS group at OSC uses a code that solves Laplace's equation in 2D to demonstrate a variety of programming techniques.
  - Vectorization
  - OpenMP
  - MPI
  - Hybrid MPI/OpenMP
- The MPI version of this code was adapted to checkpoint itself using MPI-IO
  - Sets several file hints, including FS block size
  - Uses `MPI_File_write_all()` to do writes
  - Tried both with and without collective buffer
  - Buffer size gets smaller as process count increases

2D Laplace Solver Performance



OSC
Innovations in computing, networking, and education

Conclusions

- PVFS on Itanium is a very good performer as a parallel file system for storing temporary files
  - ~1.6 GB/s read, ~1.4 GB/s write peak for parallel
  - 100-400 MB/s sustained write performance on real world parallel codes
  - About as fast as a single local disk for serial applications
- However, PVFS is not necessarily a good general purpose FS
  - Metadata operations (eg. `ls`, `df`) very slow
  - RAID-0 like behavior makes FS somewhat fragile

Future Directions

- A couple more benchmarks
  - Effective I/O bandwidth (`b_eff_io`)
  - LLNL Scalable I/O Project's `ior-mpiio`
- Connect PVFS to Athlon cluster
- PVFS drivers or proxy for non-Linux systems?
- PVFS v2 testing
  - Native GM and VIA transports as well as TCP/IP
  - Better metadata handling
  - Mirroring at FS or file level