A Parallel I/O Mechanism for Distributed Systems

Troy Baer and Pete Wyckoff
Ohio Supercomputer Center
1224 Kinnear Road
Columbus, OH 43212
{troy,pw}@osc.edu

OSC
Innovations in computing, networking, and education

Overview

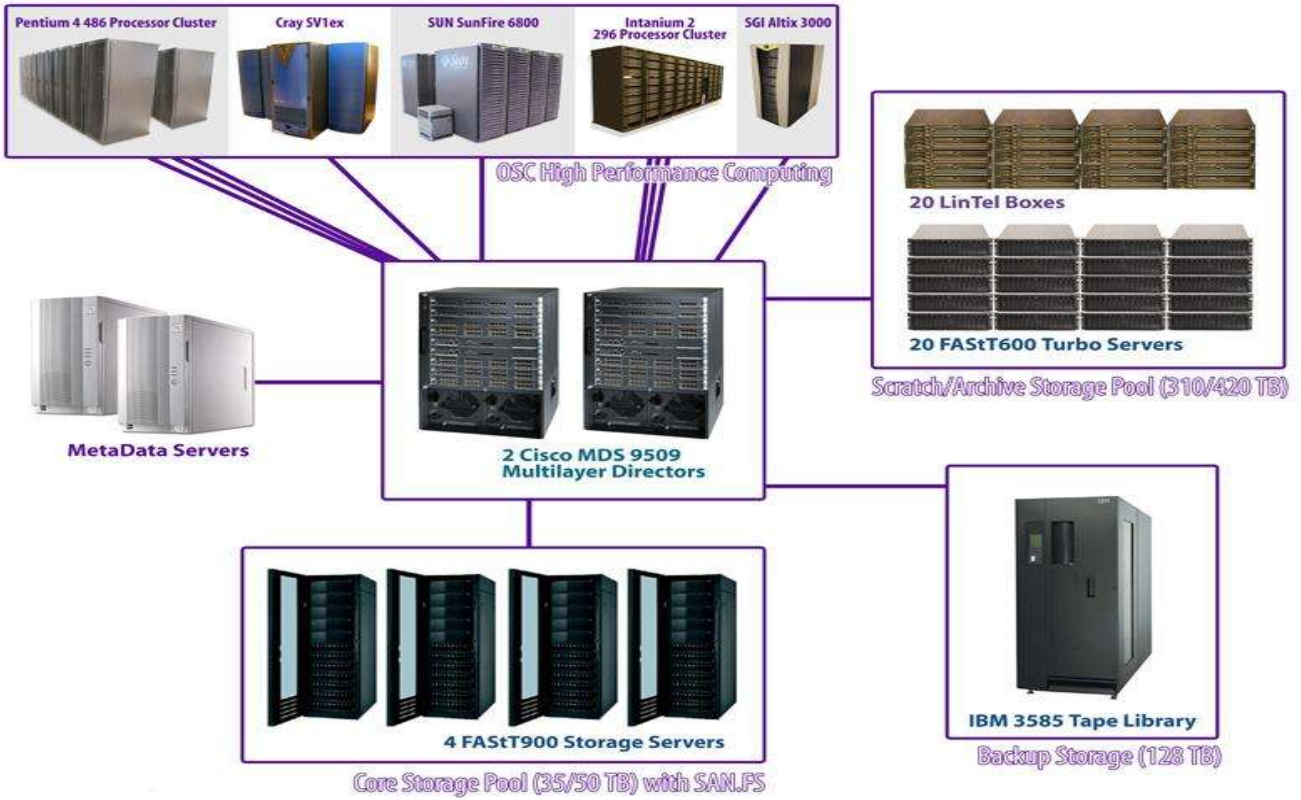
- Motivation
- Goals
- Design Issues
- Implementation Issues and Limitations
- Performance Results
- Conclusions

Motivation

- OSC has a diverse collection of computing and storage resources around the state of Ohio:
  - Central facility in Columbus with 2 large production clusters (300-500 CPUs each), 2 smaller research clusters, several SMPs, a vector mainframe, and ~450 TB of storage.
  - New satellite facility in Springfield with 2 smaller research clusters, a vector mainframe, and ~20 TB of storage.
  - ~15 smaller clusters at colleges and universities throughout Ohio, granted through OSC's Cluster Ohio program.
- Non-uniform userid management:
  - OSC central and Springfield facilities share a userid DB.
  - Each Cluster Ohio site has locally administered accounts.
  - Globus certificates used for cross-site authentication.

Motivation (con't.)

- Interest in distributed access to storage resources:
  - Data-intensive computing on remote data.
  - Loosely-coupled parallel simulations with remote data and/or output.



Motivation (con't.)

- For parallel simulations, the most widely used interface for doing I/O on shared files is the MPI-I/O part of the MPI-2 standard:
  - Blocking or non-blocking
  - Individual and collective operations
  - Basis for higher-level parallel I/O APIs
    - HDF5
    - Parallel NetCDF
- ROMIO, the reference implementation of MPI-I/O, has a modular driver interface called ADIO:
  - Numerous file system drivers:
    - UNIX FS
    - NFS
    - PVFS
    - Others
  - However, there is currently no grid-enabled file system driver.

Motivation (con't.)

- As of mid-2003 (when this project started), there was no widely available implementation of MPI-I/O for a Globus-based grid I/O protocol:
  - RIO ("Remote I/O") is described in papers from the 1996-1997 time frame, but is not included in the current ROMIO source.
  - Private communications with the ROMIO developers indicated that they had no plans for a Globus-based protocol driver at that time.
- The GridFTP protocol is a logical choice as an application-level I/O API:
  - Widely used as part of Globus software stack.
  - Has most of the functionality needed to implement MPI-I/O.

Goals

- Implement a ROMIO ADIO driver for the GridFTP protocol:
  - URI-based namespace.
  - Let Globus layers underneath GridFTP handle authentication and authorization.
- Support as much of MPI-I/O as possible, as simply as possible:
  - Blocking operations by default.
  - Use ADIO layer functionality as much as possible.
  - No shared file pointers or atomic operations.
  - Characterize and work around limitations of GridFTP protocol where possible.
- Measure performance using MPI-I/O applications:
  - ROMIO `perf`.
  - ASCI Flash using Parallel NetCDF.

Design Issues

- File namespace is URI-based:
  - `ftp://host/path/to/file`
  - `gsiftp://host/path/to/file`
- Like FTP, GridFTP has separate control and data channels which can be operating simultaneously.
- The GridFTP API is asychronous with callbacks by design; a blocking read or write operation works as follows:
  - Initiate I/O transaction on control channel
  - Initiate one or more data transfers on data channel
  - Acquire lock
  - Wait for control transaction to complete (which implies completion of data transfers)
  - Release lock

Design Issues (con't.)

- Semantic mismatch for asynchronous calls:
  - While the GridFTP API is asynchronous, it only allows one control channel transaction at a time on a given file handle.
  - MPI-I/O semantics allow for multiple non-blocking operations on the same file, so the non-blocking operations must either silently block or be handled with a FIFO operation queue.
  - Currently, non-blocking operations block silently.
- The GridFTP data channel read operation has no offset argument, making reads on non-contiguous regions more complex.
  - Entire region read into memory.
  - Relavant sections copied into output buffer.
  - Obviously does not scale for very large, sparse regions.
- GridFTP has no concept of access modes at the API level, so these must be enforced at the ADIO level.
- GridFTP has no equivalent to `creat()`, `fseek()`,or `fsync()`.

Implementation Issues

- GridFTP does not like to operate through NAT firewalls:
  - GridFTP clients go into passive mode during writes.
  - Observed symptom is that reads work, but writes fail.
  - Significant problem for any site that puts compute systems on non-routable or otherwise private networks.
  - Workaround is to map a range of TCP ports to each compute node at the firewall, which does not scale well beyond a few tens of nodes.
- Limitations to MPI-I/O functionality:
  - No atomic operations (i.e. `MPI_File_set_atomic()`).
  - No shared file pointers.
  - `MPI_File_sync()` is effective a no-op.

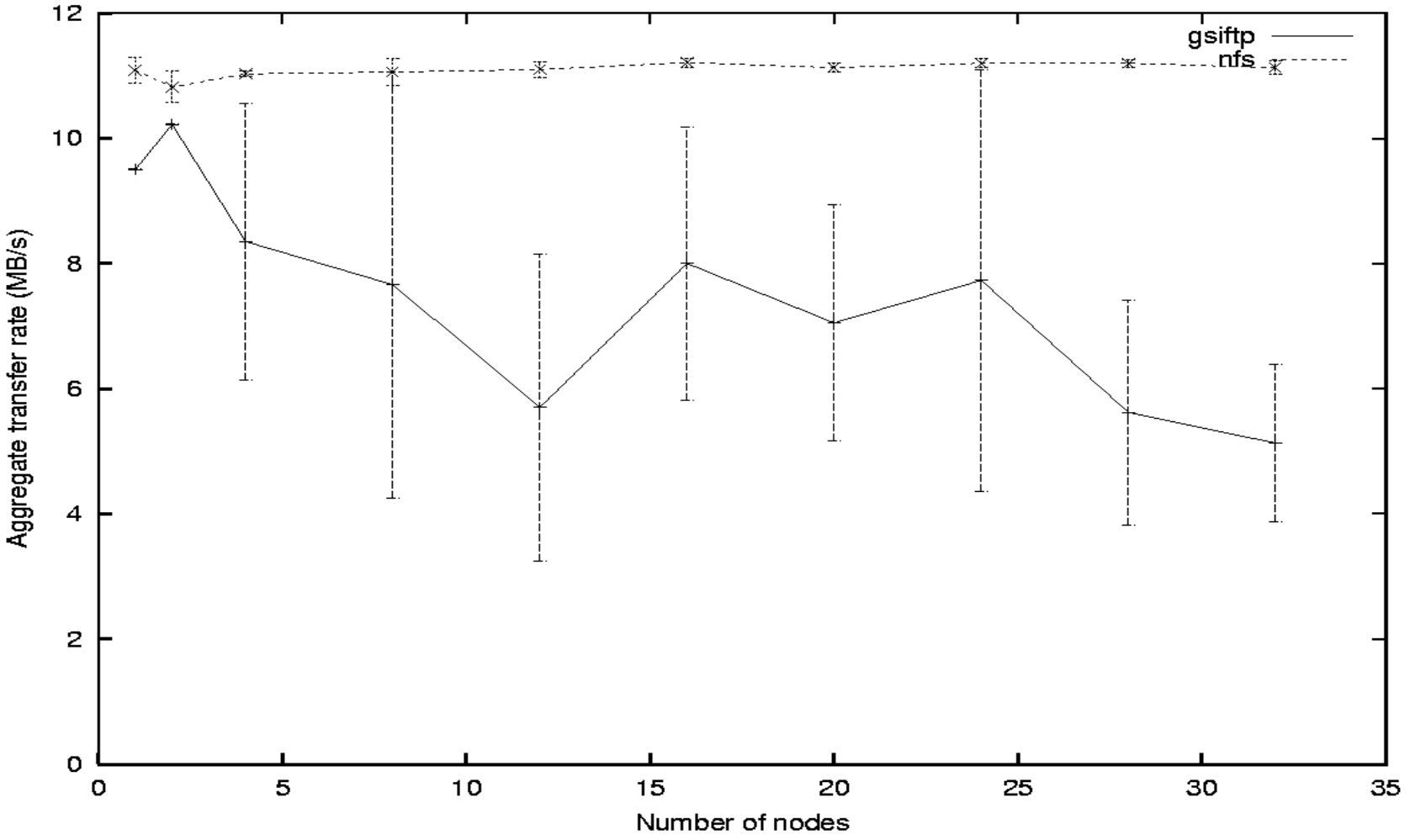
Hints for GridFTP files in ROMIO

- `(keyword,value)` pairs stores in an `MPI_Info` object
- Used to tune driver-specific parameters
- Hints implemented for GridFTP
  - `ftp_control_mode={extended,stream}`
  - `parallelism=`(integer number of threads connecting to ftp server)
  - `striped_ftp={true,false}`
  - `tcp_buffer=`(integer size in bytes)
  - `transfer_type={ascii,binary}`
- Defaults are:
  - `ftp_control_mode=extended`
  - `parallelism=1`
  - `striped_ftp=false`
  - `tcp_buffer=16384`
  - `transfer_type=binary`

Performance Results

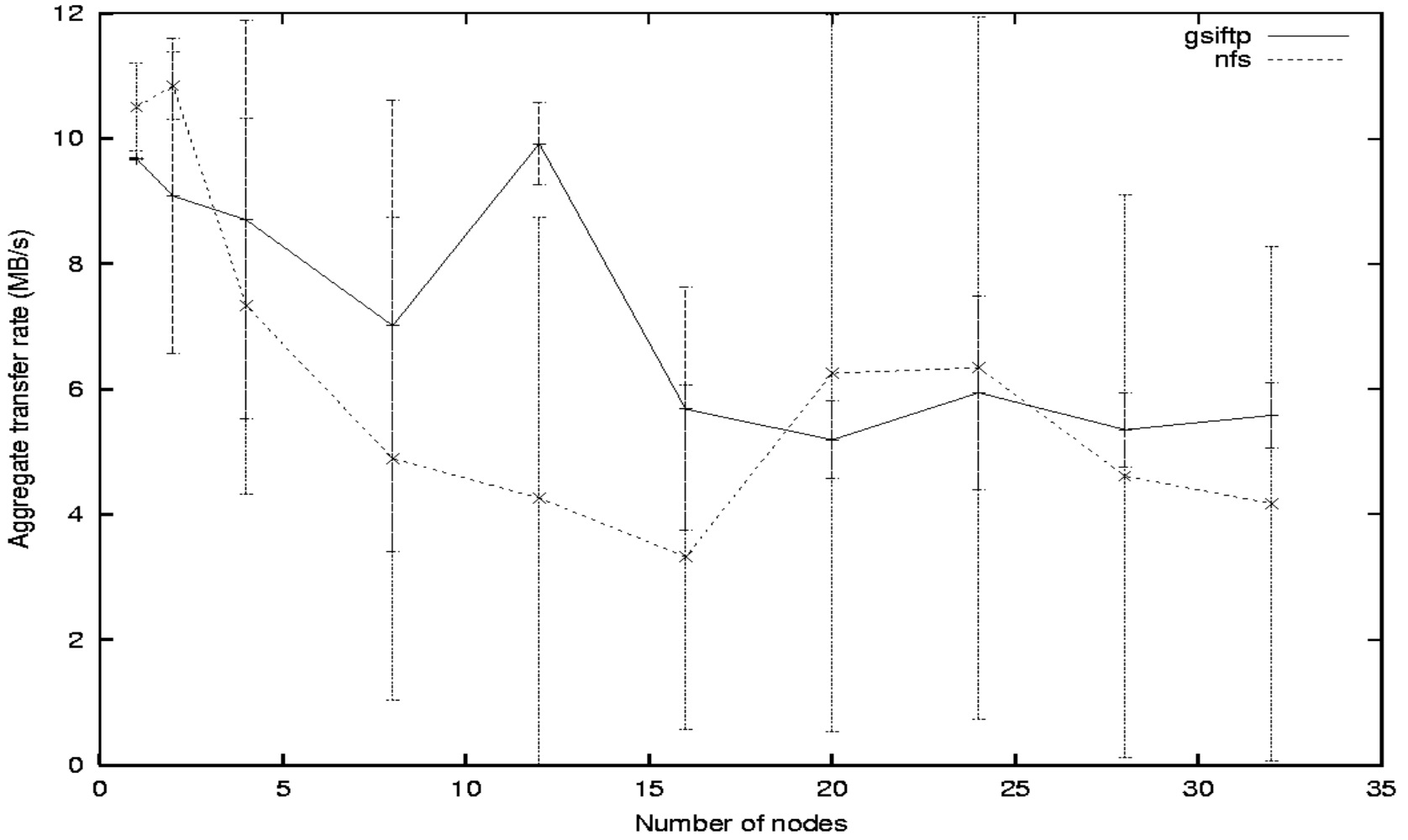
- Developed and tested on OSC's BALE research cluster:
  - 1 file server node
  - 50 compute nodes
  - Myrinet and 100Mbit/s Ethernet
  - Tests compare GridFTP vs. NFS over Ethernet
- Performance tests:
  - ROMIO `perf` example program, modified to run multiple times and compute average and standard deviation.
  - ASCI Flash I/O benchmark, using Parallel NetCDF backend from Argonne Nat'l Lab and Northwestern University.

Performance: ROMIO `perf` (Read)



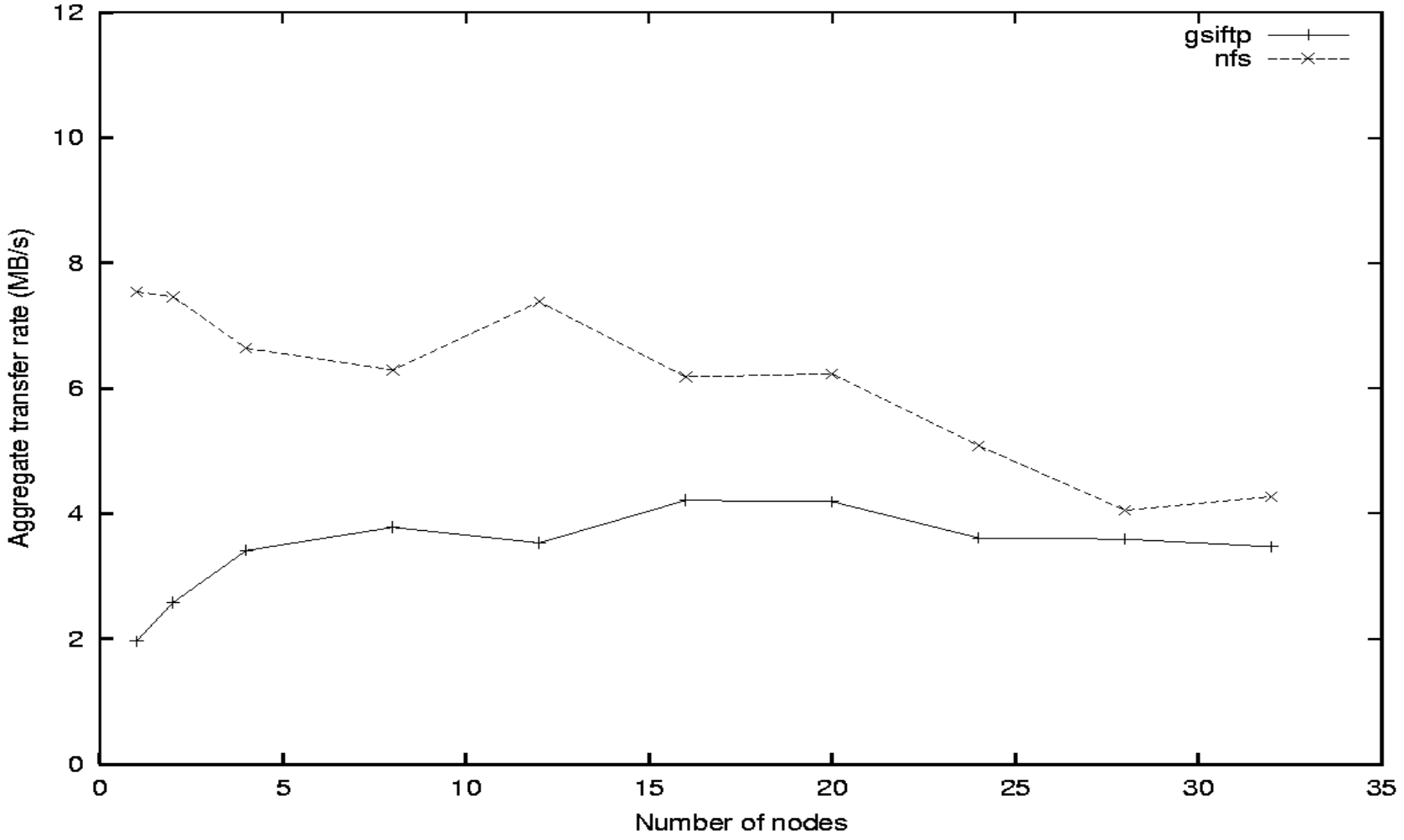
OSC
Innovations in computing, networking, and education

Performance: ROMIO `perf` (Write)



OSC
Innovations in computing, networking, and education

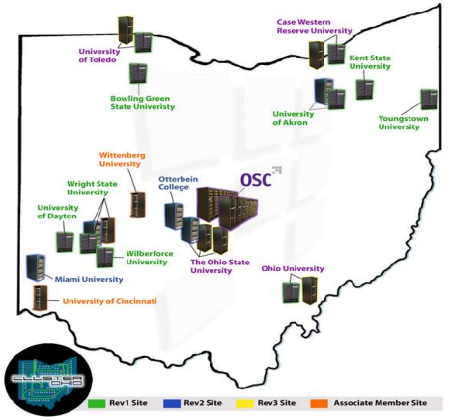
Performance: ASCI Flash using PNCDF



OSC
Innovations in computing, networking, and education

Parallel I/O in a Grid Environment

- Because of the passive writer problem mentioned earlier, MPI-I/O over GridFTP is effectively read-only for systems behind NAT firewalls. However, it still has significant uses.
- For instance, OSC has used MPI-I/O over GridFTP to distribute bioinformatics database files to Cluster Ohio sites for use by an MPICH/G2 based implementation of BLAST:
  - No file prestaging required.
  - Each MPI process uses MPI-I/O over GridFTP to retrieve the necessary files if they are not already locally accessible.



Conclusions

- GridFTP can be used to implement almost all of MPI-I/O's functionality:
  - Missing functionality is also missing on numerous other parallel file systems (eg. PFS and PVFS).
  - Allows MPI-I/O applications to be seamlessly transitioned onto the grid in many cases.
- Performance roughly comparable than NFS over same network:
  - Reads almost always slower.
  - Writes sometimes equal or faster.
  - Much more secure over wide area networks thanks to Globus PKI infrastructure.
- Patch including GridFTP driver for ROMIO has been submitted to the ROMIO developers.
- In production on OSC production cluster systems and Cluster Ohio rev. 2 & 3 sites.

Future Work

- Better support for doing strided reads:
  - Current implementation reads entire region and copies out desired pieces – clearly does not scale to sparse collections of data in very large (GB-TB range) files.
  - Add a hint to allow user to select a size or sparsity threshold, beyond which each portion of data would be read directly.
- Better non-blocking support:
  - Initiating a non-blocking operation would cause that operation to be added to an operation queue.
  - When one operation ends, it would automatically start the next operation in the queue.
  - Blocking operations could then be implemented in terms of a non-blocking operation plus a wait.
- Other MPI-I/O functionality?
  - Shared file pointers?
  - GridFTP operations unlikely to be made any more atomic than they already are.

Availability

- Patch, slides and paper available at `http://www.osc.edu/~troy/romio-gridftp/`.
- Patch has been also submitted to ROMIO developers.