

Experiences with Parallel I/O and Visualization

Aaron Jackson

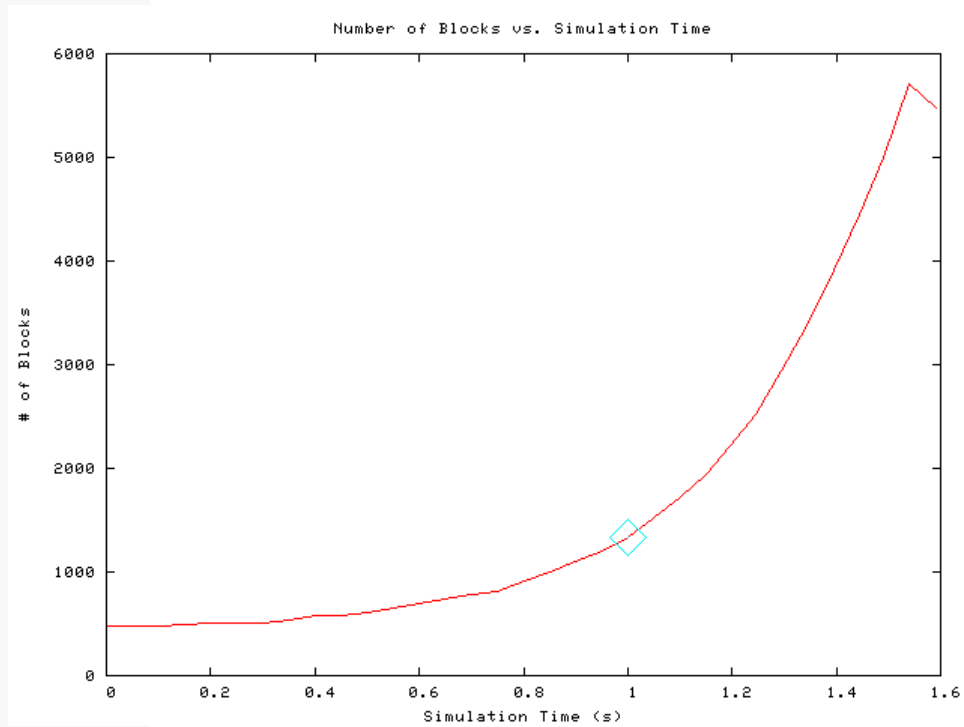
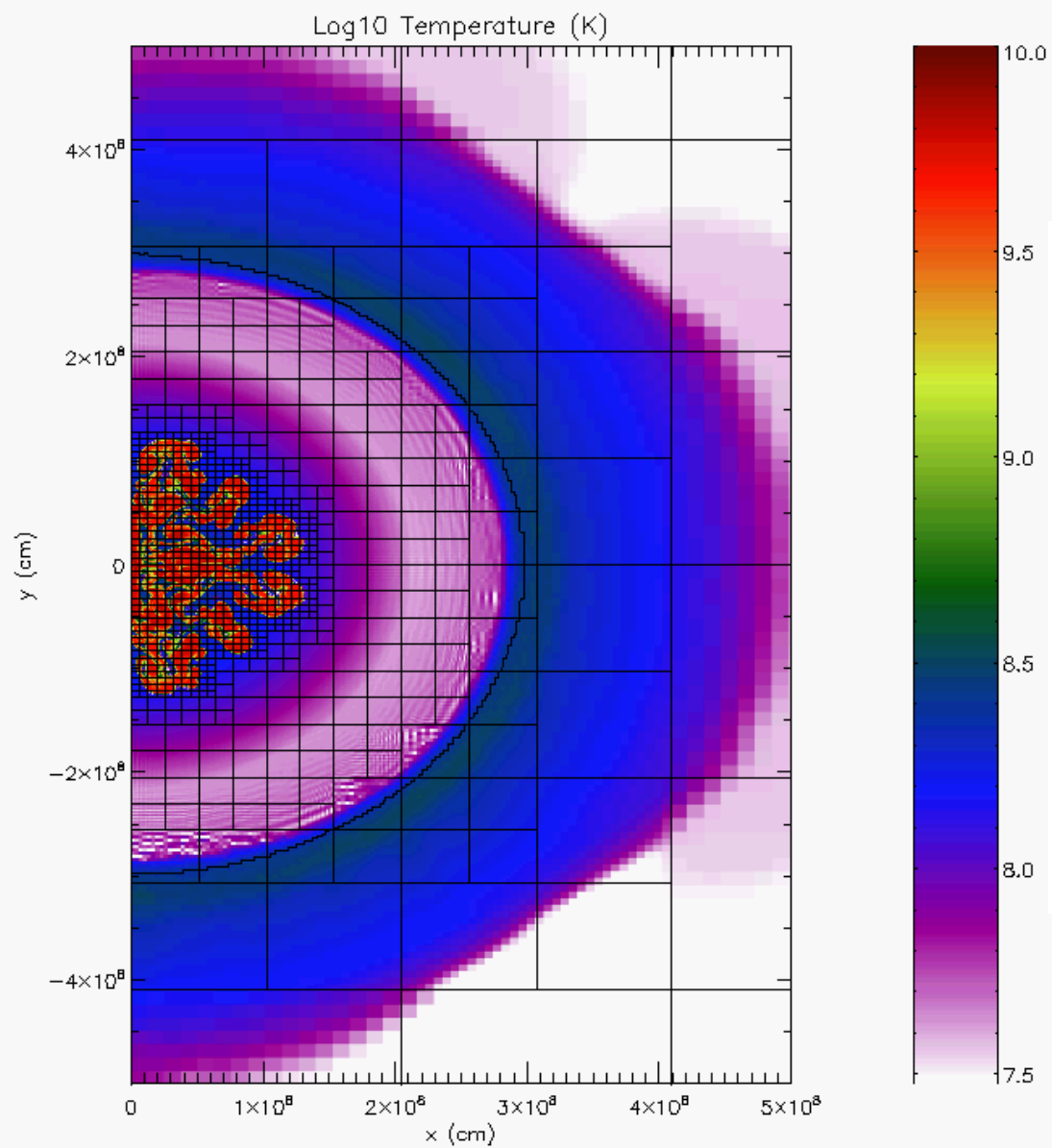
Supported in part by grant DE-FG02-07ER41516
from the US Department of Energy

Outline

- Discuss computational research needs
- Parallel Hierarchical Data Format (PHDF5)
- Building / Installing HDF5 on BlueGene
- VisIt, a parallel visualization tool
- Building / Installing VisIt at BNL
- Using VisIt to visualize HDF5 data from your desktop (all in parallel).

Research Needs

- FLASH, a parallel AMR code developed at U. of Chicago
- Highly parallel simulation codes require parallel I/O to be efficient
- Adaptive Mesh Refinement (AMR)
- Job size increases with simulation time
- Need to be able to stop and restart simulation on a different number of nodes



time = 1.00024 s

number of blocks = 1338

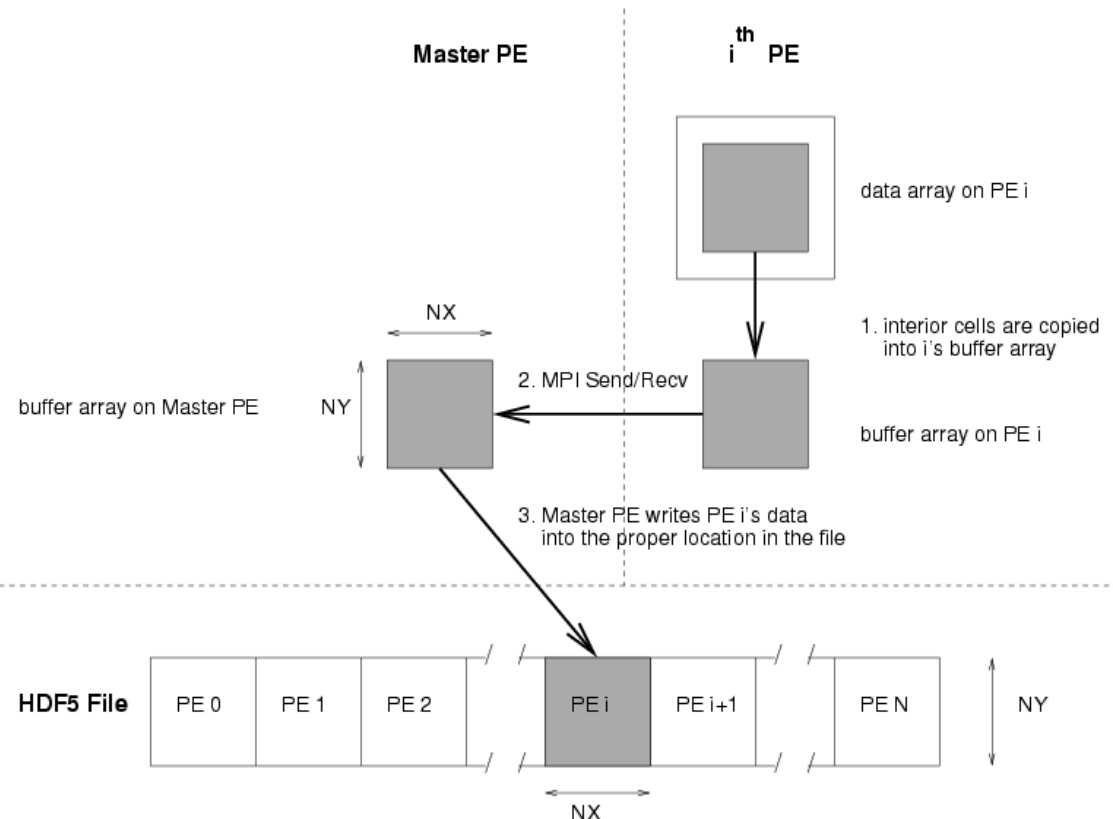
AMR levels = 11

Parallel vs. Serial I/O

- **Serial**

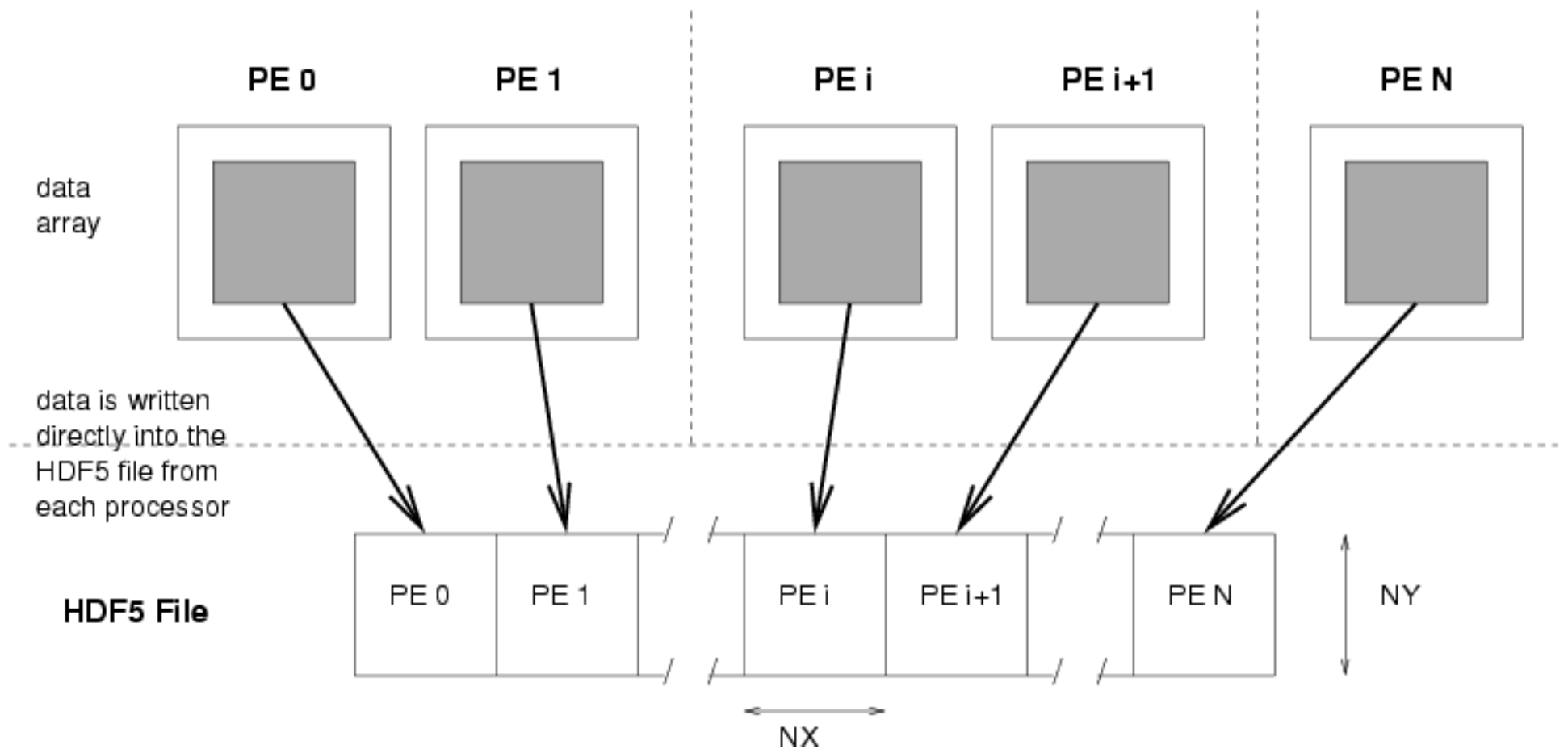
... do some calculation

```
if (rank.ne.MASTER) then
  MPI_SEND(data, ndata, &
    MPI_REAL, MASTER, itag, &
    icomm, ierr)
else
  do i=1, size
    MPI_RECV(data, ndata, &
      MPI_REAL, i, itag, &
      icomm, istat, ierr)
    write(6, *) (data(j), &
      j=1, ndata)
  enddo
endif
```



Parallel I/O

- Sits on MPI/IO library



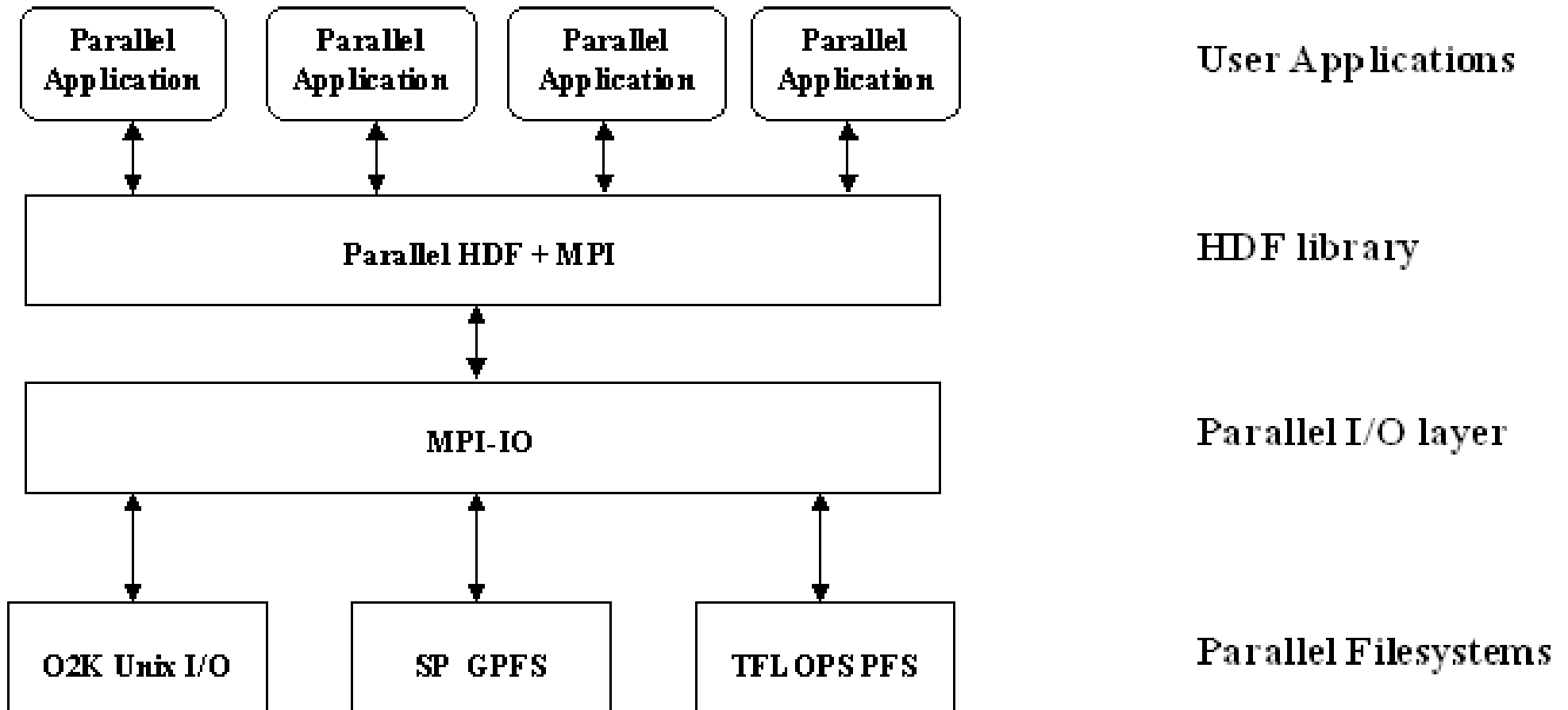
Parallel I/O

- What do we want from parallel I/O?
 - Take advantage of BlueGene's parallel hardware
 - Portability, we want to read and write data on someone else's hardware (parallel or serial)
 - Data to be independent of the number of processors
 - Restart Capability
 - Simple to implement

Parallel HDF5

- Parallel Hierarchical Data Format (PHDF5)
- Takes advantage of BlueGene's parallel I/O infrastructure
- HDF4 does not have parallel I/O functionality
- Portability
 - Endianess
 - No post-calculation file manipulation
- Motivation
 - Restart on different # of processors

Parallel HDF5



Pset Ratio

- According to the NYBlue partition naming scheme pset ratio is defined below:
 - A 1:16
 - B 1:32
 - C 1:64
 - D 1:128
- Each I/O node serves a group of compute nodes given by the ratio above
- This architecture is abstracted by the MPI-IO interface

Parallel HDF5 Resources

- The HDF Group website has some tutorials:
 - The HDF Group Homepage
 - <http://hdf.ncsa.uiuc.edu/HDF5/>
 - Parallel HDF5 Tutorial
 - <http://www.hdfgroup.org/HDF5/Tutor/parallel.html>
- Mike Zingale has some good example programs written in C:
 - IO Tutorial
 - http://www.astro.sunysb.edu/mzingale/io_tutorial/

Building HDF5

- Thanks to Stratos Efsthadiadis for coordinating communications and thanks to Adam C. Lichtl for his notes / scripts on building HDF5.
 - More information at www.astro.sunysb.edu/ajackson/hdf5.html
- HDF5 (v1.8.1) has two API versions (v1.6 and v1.8) which can be set at build time with the configure flag “--with-default-api”

Building HDF5

- HDF5 on BG/L
 - Front-end vs. Cross-compile
 - Cross-compile builds HDF5 for the compute nodes
 - One of the parallel tests built in to the PHDF5 build script did not finish, but running FLASH which uses PHDF5 wrote HDF5 files without errors
 - We are in the process of testing restart capabilities

VisIt at BNL

- Cluster for visualization:
 - Hostnames: vis1-4.bluegene.bnl.gov
- No transferring of files necessary
 - Shares GPFS w/ BG/L & BG/P
 - Can visualize data while simulation is still running
- Understands HDF5 file format
 - Many conventions: Tetrad, SAMRAI, CosmosPP, Pixie, FLASH, Chombo, GTC, H5Nimrod, M3D, XDMF, CGNS, Silo

VisIt in Parallel

- VisIt has the ability to run in parallel to render large data much faster than it could serially
- VisIt works fastest in distributed mode
 - Front-End for Visualizing
 - Back-End for Rendering and Computing
- These two components connect through the network, so we have to supply a direct communication channel

Direct Access to BNL

- We can accomplish this direct channel via SSH-Tunneling on the SSH Gateway at BNL
 - Instructions for Linux at www.astro.sunysb.edu/ajackson/access_bgl.html
- You will need access to SSH gateway and have a public dsa key deployed to NY Blue
- There is no current solution for this in Windows that I am aware of (might try SSH Secure Shell)

Current Progress

- VisIt (as of Aug. 1st)
 - Build was “successful” in that there were no printed errors during the compilation process
 - VisIt front-end can connect to parallel VisIt back-end on the visualization cluster and can access datafiles
 - VisIt seems to be able to read data using the HDF5 library to get correct limits on plots
 - However, VisIt only seems to be able to graph the legend, axis, and labels, not the actual data.

Troubleshooting

- Getting the front-end and back-end of VisIt working together relies on the SSH connection
 - Currently, vis1-4 are configured to drop inactive SSH connections after ~15 minutes. VisIt does not handle dropped connections gracefully and usually ties up the node such that it does not respond to SSH connection requests

Loose Ends

- MPICH configuration
 - Get VisIt running on all 4 nodes of the Vis Cluster on all 32 processors
 - Instructions will be posted on
 - <http://www.astro.sunysb.edu/ajackson/visit.html>
- Test multi-user environment
- Resolve data rendering issue

VisIt Resources

- VisIt Homepage
 - <http://www.llnl.gov/VisIt>
- VisIt Users Forum
 - <http://VisItusers.org>
- VisIt Users Mailing List
 - visit-users@ornl.gov
- Using VisIt at BNL
 - <http://www.astro.sunysb.edu/ajackson/bluegene.html>