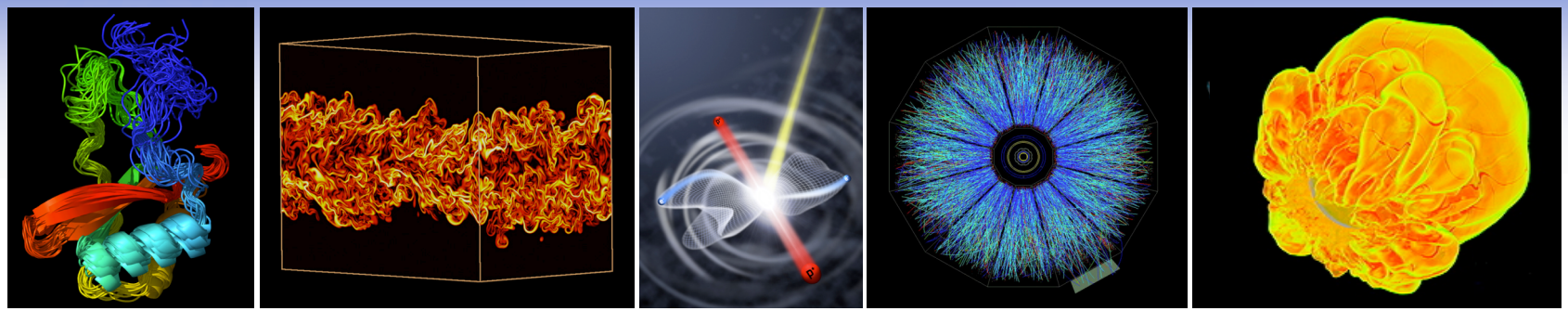


NERSC Overview

CSADS Workshop on PetaScale Applications and Performance Strategies

Katie Antypas
HPC Consultant

July 27, 2009

Office of Science
U.S. DEPARTMENT OF ENERGY

NERSC

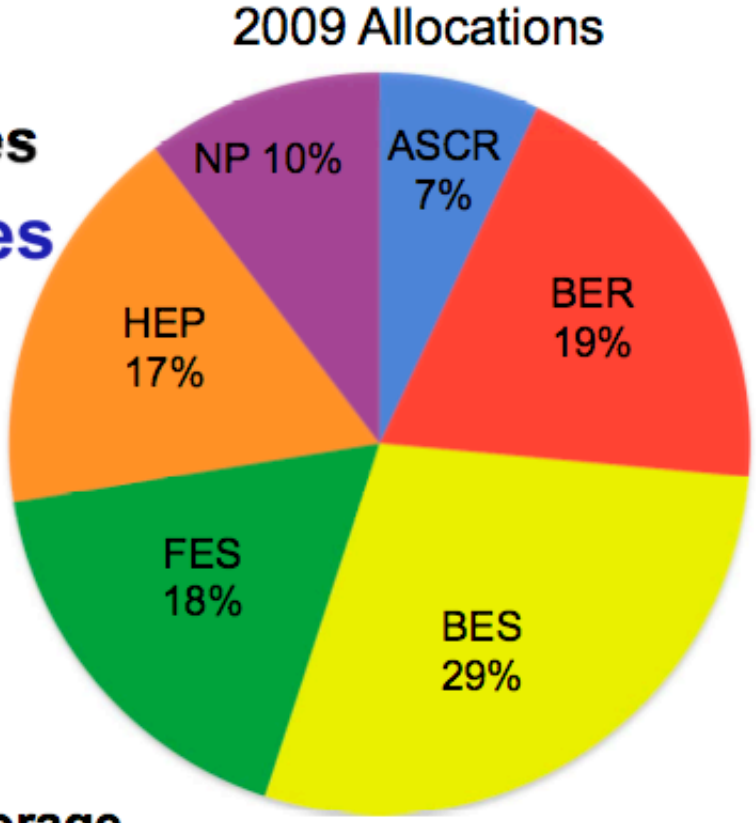
BERKELEY LAB

NERSC Mission

The mission of the National Energy Research Scientific Computing Center (NERSC) is to *accelerate the pace of scientific discovery* by providing high performance computing, information, data, and communications services for *all DOE Office of Science (SC) research.*

NERSC is the Production Facility for DOE SC

- **NERSC serves a large population**
  - Approximately 3000 users,
  - 400 projects, 500 code instances
- **Focus on "unique" resources**
  - High end computing systems
  - High end storage systems
    - Large shared file system
    - Tape archive
  - Interface to high speed networking
    - ESNEt soon to be 100 Gb/s
- **Allocate time / storage**
  - Current processor hours and tape storage



ASCR's Computing Facilities

NERSC

LBNL

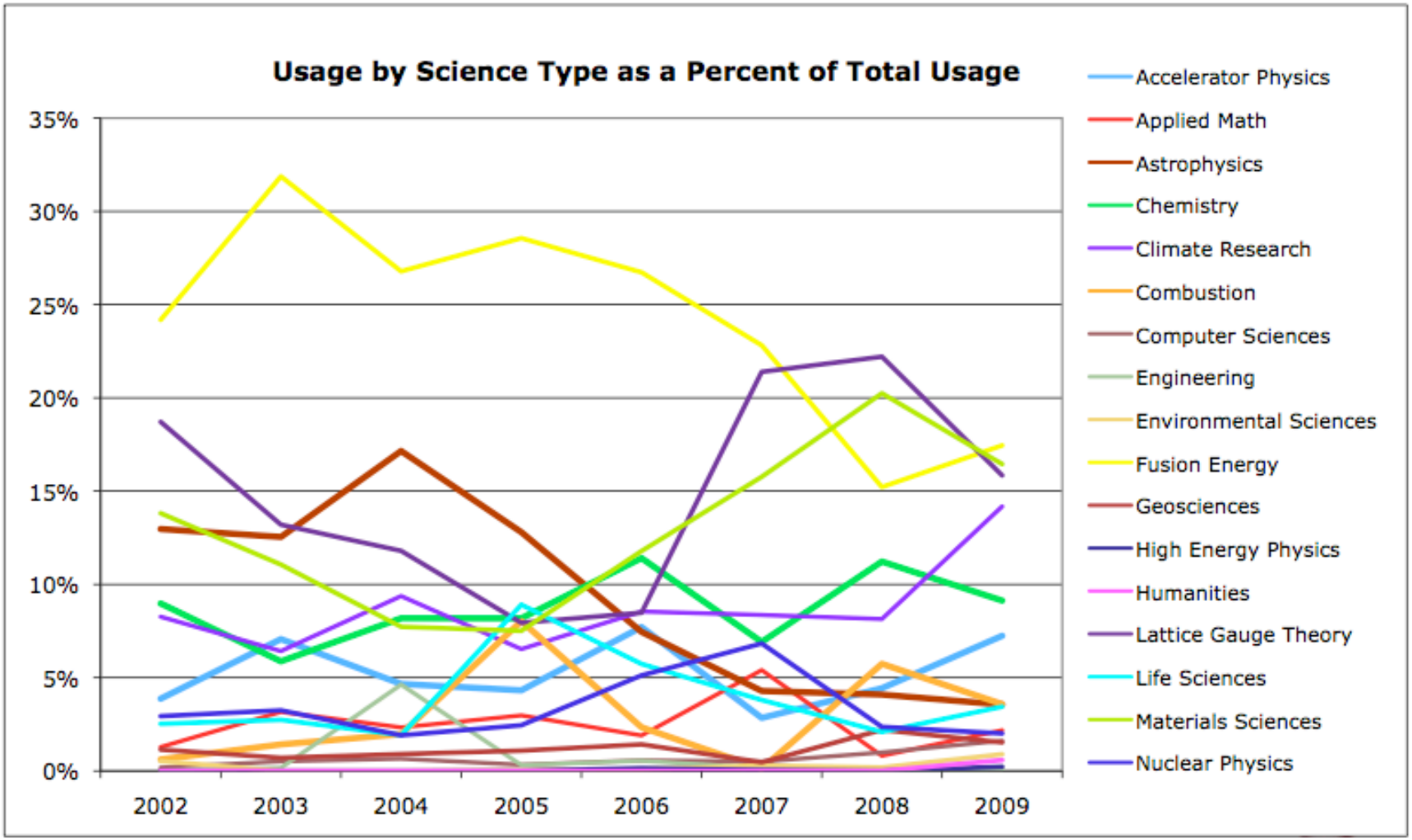
- Hundreds of projects
- 2010 allocations:
  - 70-80% SC offices control; ERCAP process
  - 10-20% ASCR (new ALCC program)
  - 10% NERSC reserve
- Science covers all of DOE/SC science

LCFs

ORNL and ANL

- Tens of projects
- 2010 allocations:
  - 70-80% ANL/ORNL managed; INCITE process
  - 10-20% ACSR (new ALCC program)
  - 10% LCF reserve
- Science areas limited to those at largest scale; not limited to DOE/SC

What's Changed in DOE Priorities for NERSC?



Getting an HPC allocation

- Not as hard as you might think
  - If you have an abstract of your research goals applying will take you 30 min or so
- A small allocation is stepping stone toward a large allocation when you need it. It helps you build a computing relationship with DOE and project reviewers.
- NERSC
  - https://nim.nersc.gov/newpi.php
- ANL
  - https://accounts.alcf.anl.gov/accounts/projects/intrepid.htm
- ORNL
  - http://www.nccs.gov/user-support/access/project-request

Account Support and HPC Consulting

- **Account support**
  - Passwords (NERSC does not use OTP keys)
  - New accounts
  - Modify accounts (add user to project)
- **HPC Consulting**
  - 9 Consultants to serve NERSC users
  - Aim to provide fast helpful advice from simple to complex
    - I can't submit my job
    - What library should I use?
    - My code is performing slowly
    - My code compiled on my department cluster but now …
  - Please contact the consultants!
  - We are paid to help make you more productive
  - We have often seen your problem many times before with other users

Consulting Goals

- **Usability and Productivity**
  - Users need more than just computing
    - Storage HPSS
    - Data analysis
    - Fast network access
    - Reasonable turn around time
    - Easy access to help
- **Get you up and running quickly**
  - Extensive tutorials
  - Web examples www.nersc.gov
  - Help from consultants

NERSC Training Accounts

- Training accounts available for workshop
- Access to NERSC Machines
  - "ssh train15@franklin.nersc.gov"
- Just need to sign form and I will give you password
- Queue with boosted priority already set up
  - Up to 24k cores
  - 6 hour wall clock limit
  - 20 concurrent jobs for the group
- Come talk to me at the break

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Systems

Office of Science U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

NERSC 2009 Configuration

Large-Scale Computing System

Franklin (NERSC-5): Cray XT4
- 9,740 nodes; 38,128 cores
- 38 Tflop/s sustained SSP
- 355 Tflops/s peak
- 8 GB of memory per quad core node

Clusters

Bassi (NCSb)
- IBM Power5 (888 cores)

Jacquard (NCSa)
- LNXI Opteron (712 cores)

PDSF (HEP/NP)
- Linux cluster (~1K cores)

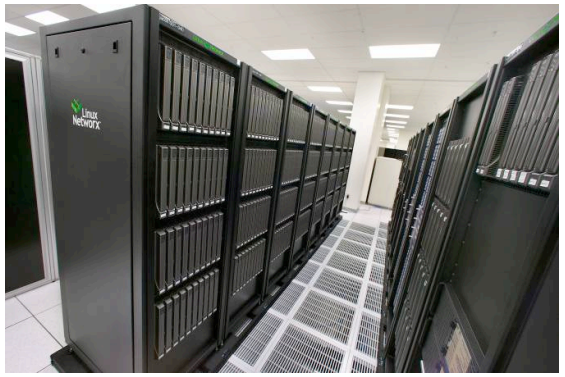
NERSC Global Filesystem (NGF)
IBM's GPFS
440 TB; 5.5 GB/s

HPSS Archival Storage
- 59 PB capacity
- 11 Tape libraries
- 140 TB disk cache

Analytics / Visualization
- Davinci (SGI Altix)

NERSC6 and SU2009

- NERSC6
  - Announcing with 1-2 weeks
  - Procured to increase computing capacity of the center by 3-5x.
  - Stay tuned!
- SU2009 (Scalable Unit Cluster)
  - NERSC users have needs for a pure Linux cluster
    - Many promising projects start from departmental clusters and need to scale up
    - Easy porting for new users
    - Best system for complex workflow models
    - Some applications best suited for Linux Cluster



NERSC Global Filesystem (NGF)

- Seamless data access from NERSC's computational and analysis resources
- Single unified namespace makes it easier for users to manage their data across multiple system
- Goals: Functionality, Reliability, Performance
- Uses IBM's GPFS filesystem

Large Storage Environment (HPSS)

- Access HPSS from any NERSC resource with:
  - HIS
  - HTAR
  - ftp/pftp
- Outside NERSC must do a few extra steps
- 61+ million files
- 44 PB capacity

Franklin Programming Environment

- Compilers (Fortran, C, C++)
  - PGI
  - PathScale
  - GNU
- Parallel Programming Models: Cray MPICH2 MPI, Cray SHMEM, Open MP
- AMD Core Math Library (ACML): BLAS, LAPACK, FFT, Random number generators, GNU Fortran libraries
- LibSci scientific library: ScaLAPACK, BLACS, SuperLU
- Profiling tools CrayPat, Apprentice2, IPM, TAU
- Performance API (Papi)
- Modules

Extensive 3rd Party Software

- Check to make sure your application isn't already installed
- Use modules command to see software availability on all NERSC machines ("module avail")
- Math - acml, aztec, dfftpack, fftw, gsl, LibSci, parmetis, parpack, petsc, pspline, superlu, sprng
- I/O - hdf5, nco, netcdf, pnetcdf
- Chemistry/Mat Sci - amber, namd, nwchem, abinit, cpmd, lammps, quantum expresso, siesta, vasp
- Visualization - idl, gnuplot, visit, ncar
- Debuggers - Allinea's DDT

Franklin Overview

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Full Linux OS

CNL (no logins)

Login Node

Login Node

Login Node

etc....

Compute Node

Compute Node

Compute Node

Compute Node

Compute Node

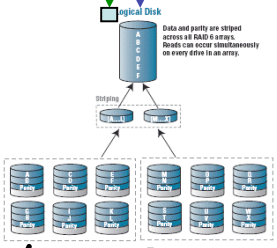
Compute Node

Compute Node

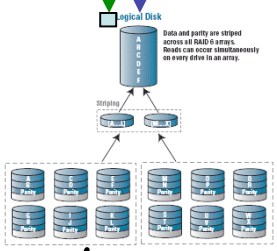
Etc ...

No local disk

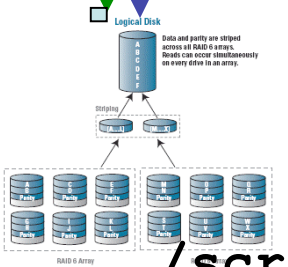
HPSS



/project



/home



/scratch

Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

What kind of OS?

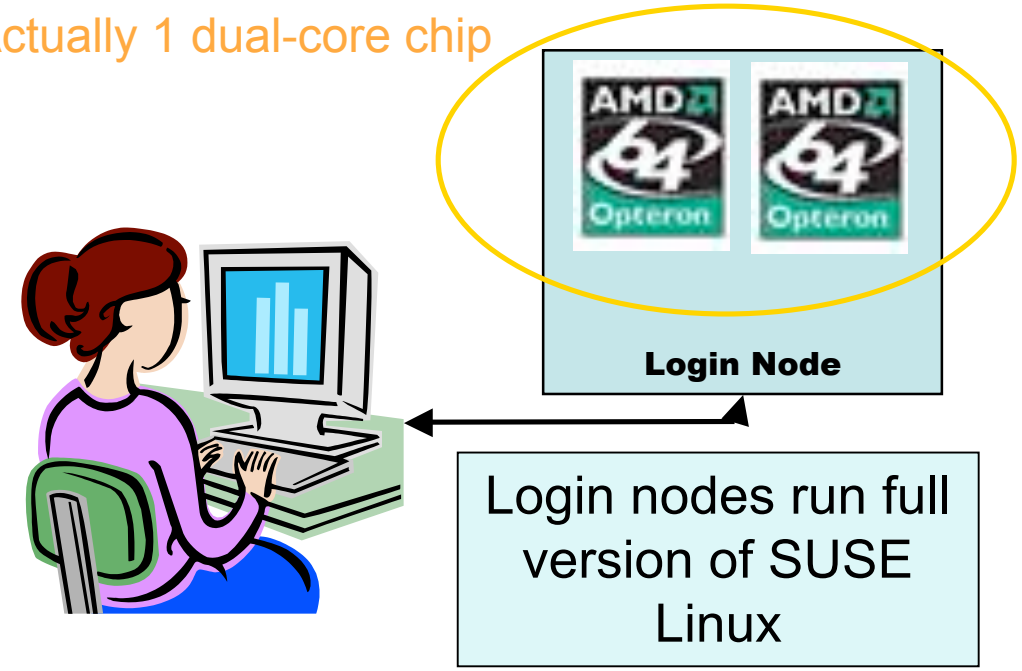
- Consider what kind of OS you are using
  - Limited OS
    - Depends on system but limited OS calls
    - Features which could be limited on compute nodes
      - Shared libraries (support coming soon)
      - Scripting languages, python, perl
      - Process control (fork, exec)
      - Can't ssh from compute node to compute node
      - Can't call system() from Fortran parallel job
      - No Java on the compute nodes
      - No X-Windows support on compute nodes

Memory Considerations

- Each Franklin compute node has 8GB of memory.
- Running 4 cores per node 7.38 GB of user addressable memory
  - CNL kernel, uses ~300 MB of memory.
  - Lustre uses about 17 MB of memory
  - MPI buffer size is about ~100 MB.
- Quad core MPI jobs have ~1.83 GB/task.
- Change MPI buffer sizes by setting certain MPICH environment variables.
- Hints for adjusting MPICH variables on website

Running a Job on Franklin

Actually 1 dual-core chip



Login nodes run full version of SUSE Linux

On a Franklin login node:

1. Log in from your desktop using SSH
2. Compile your code or load a software module
3. Write a job script
4. Submit your script to the batch system
5. Monitor your job's progress
6. Archive your output
7. Analyze your results

`www.nersc.gov/nusers/status/queues/franklin/`

NERSC Analytics server (DaVinci)

Batch Queues

- At NERSC users submit jobs to a queue and wait in line to run
- Queue policies are set to:
  - Be fair
  - Accommodate needs
    - Users
    - DOE strategic
  - Encourage high parallel concurrency
  - Maximize scientific productivity
- Special requests always given consideration
  - Reservations
  - Emergencies

Batch Queues

- debug: short, small test runs
- interactive: implicit in `qsub -I`
- regular: production runs
  - Jobs > 512 nodes given 50% discount
- premium: I need it now, 2X charge
  - Fast turn around on Franklin, not usually needed
- low: I can wait a while: 50% discount
- special: unusual jobs by prior arrangement

IO on Franklin

- /home and /scratch(s) file use Lustre File System
/project uses GPFS file system
Failures happen - recommend checkpointing code

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Scratch Disk Space

- Disk space is expensive and therefore limited and shared among users
- Every center must manage disk space in some way (purging, begging, quotas)
- Understand the disk usage policy at your center
- Be a courteous disk space user. We want you to run very large jobs, but then we want you to back up your files (quickly)

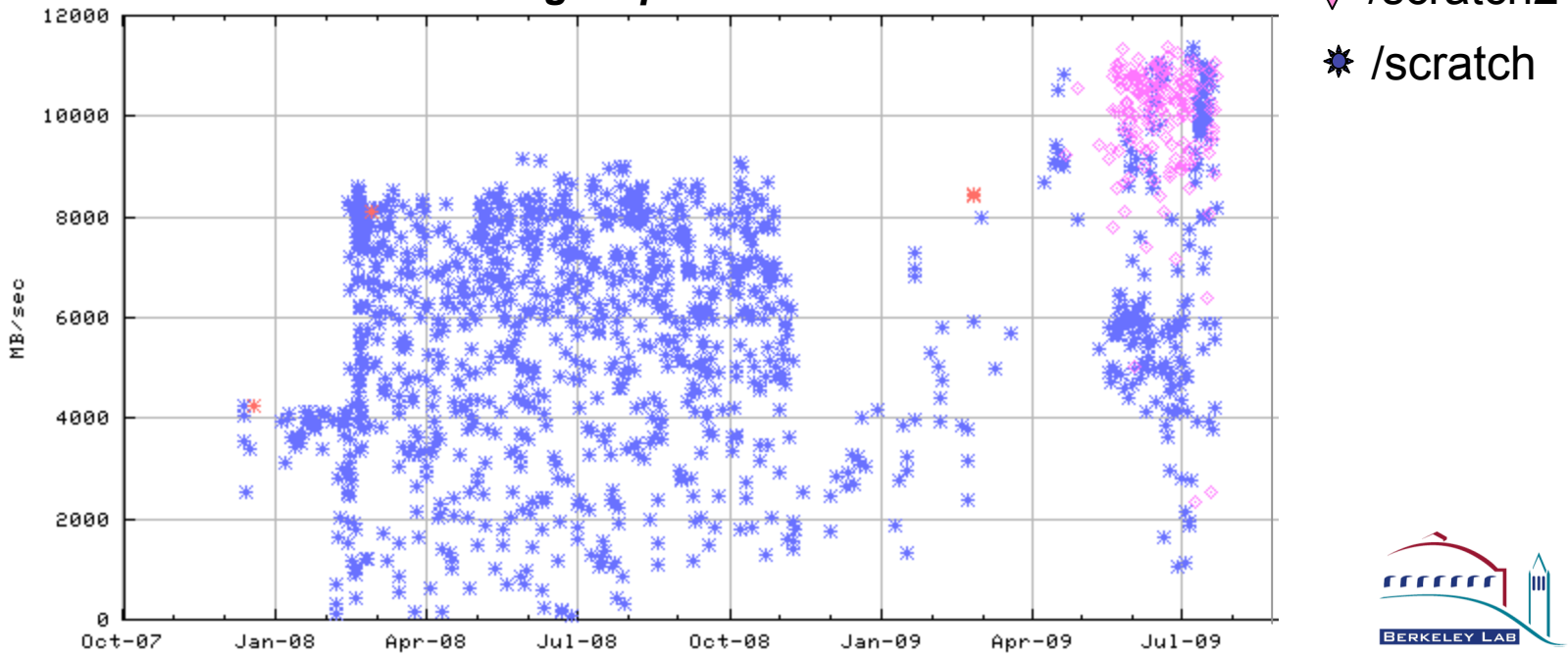
Disk Quotas

- Franklin has multiple file systems
  - /home (Default 15GB)
    - Backed up
    - Permanent
  - /scratch and /scratch2 (Default 500 GB)
    - Purged of files older than 12 weeks
    - Not backed up
    - Not permanent
  - /project (NERSC Global Filesystem)
    - Accessible from all NERSC machines
    - Currently need to request access
- Users can not submit jobs when over quota
- Projects needing larger disk quotas just need to ask

Franklin IO Upgrade

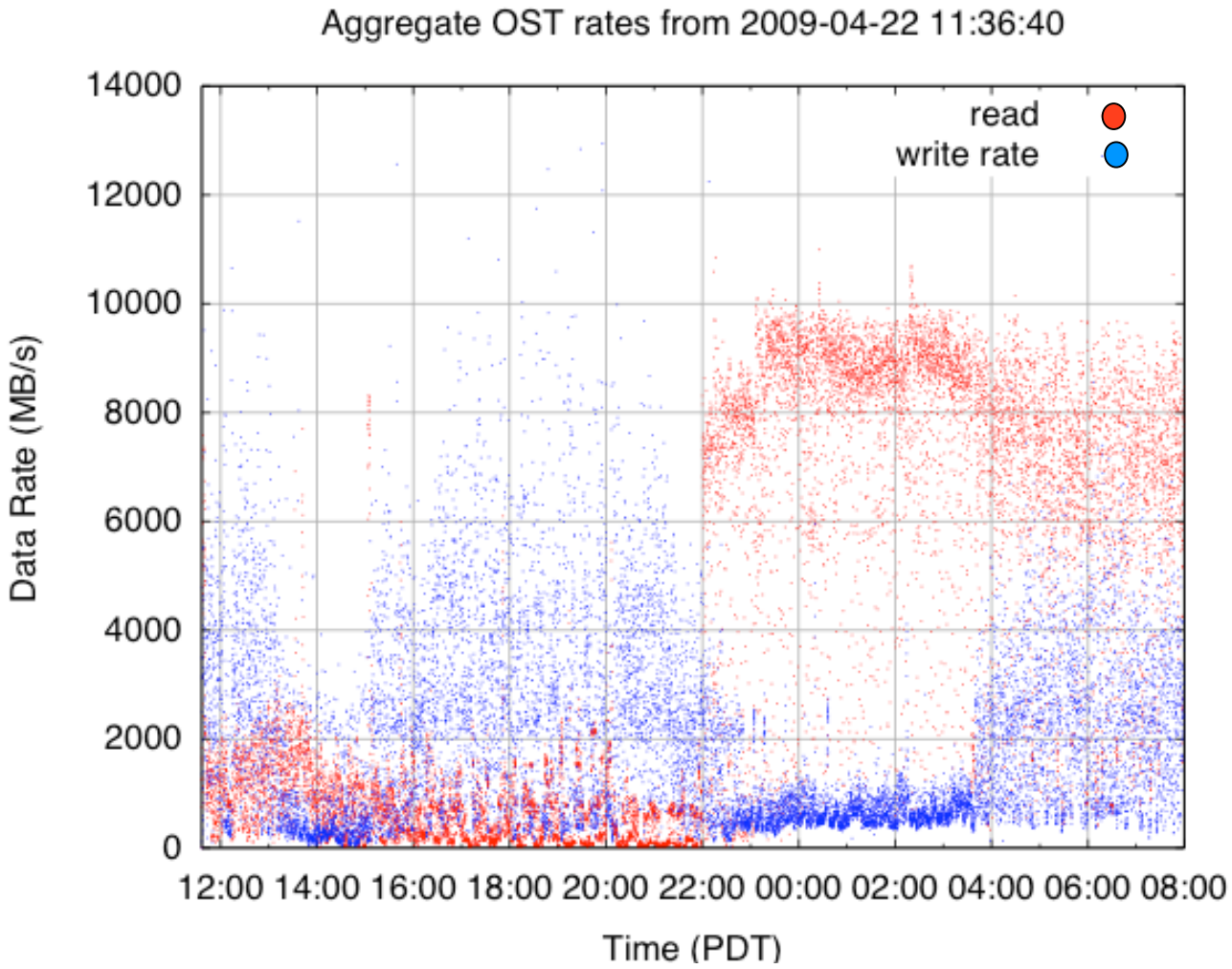
- I/O upgrade in March tripled Franklin's I/O bandwidth from 11GB/sec to ~32 GB/sec
  - Doubled # of I/O nodes
  - Distributed them more evenly within machine
- Created 2 /scratch file systems
- Increased /scratch storage by 30%

I/O Performance Monitoring During Production Time using 64 proc IOR Test



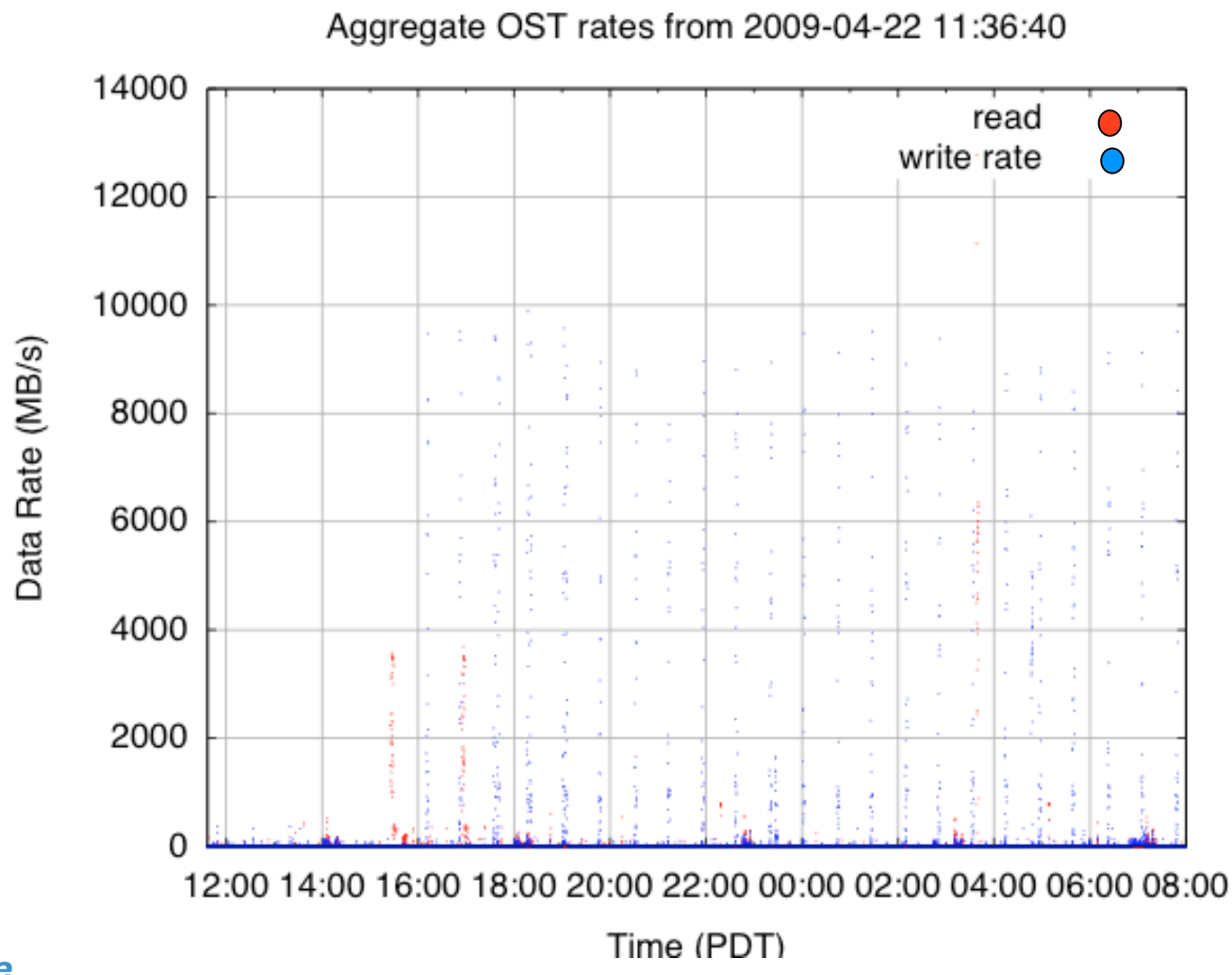
Lustre Monitoring Tool

/scratch



Lustre Monitoring Tool

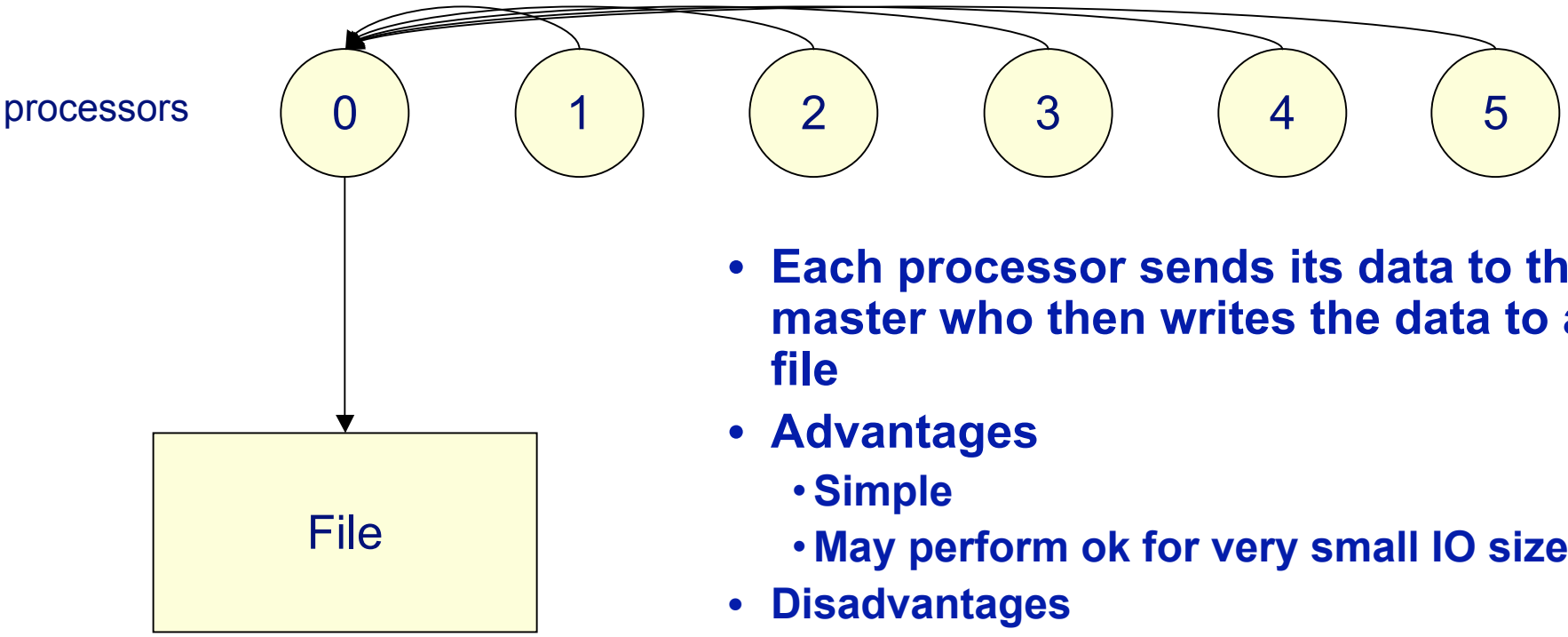
/scratch2



But what about my application?

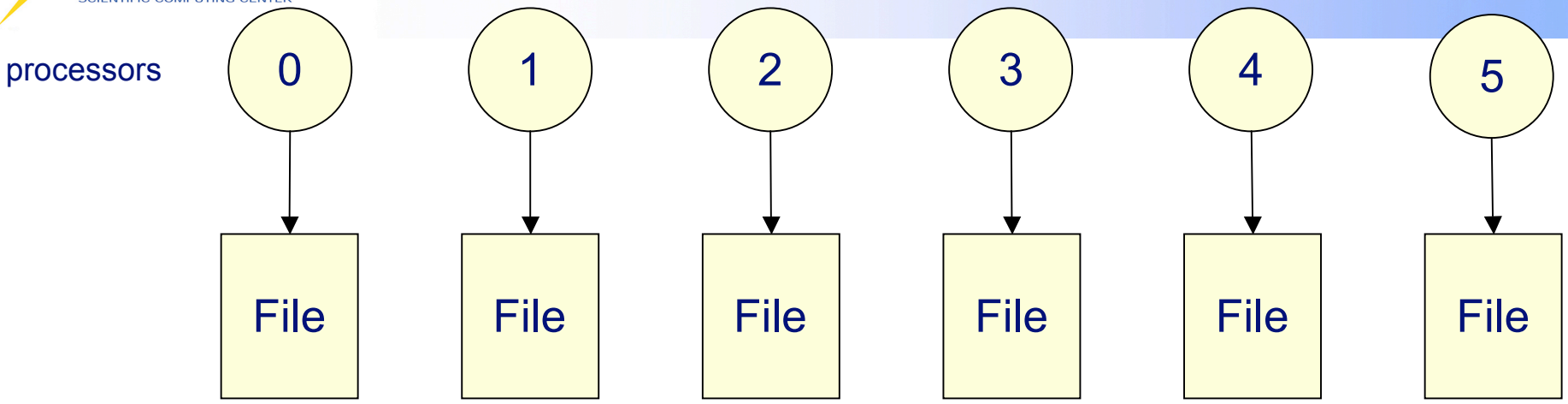
- Most I/O bandwidth out of a Franklin node
  - 1 core: 350-400 MB/sec
  - 2 cores: 700 - 800 MB/sec
  - 3 cores: ~1100 MB/sec
  - 4 cores: ~1100 MB/sec

Serial I/O



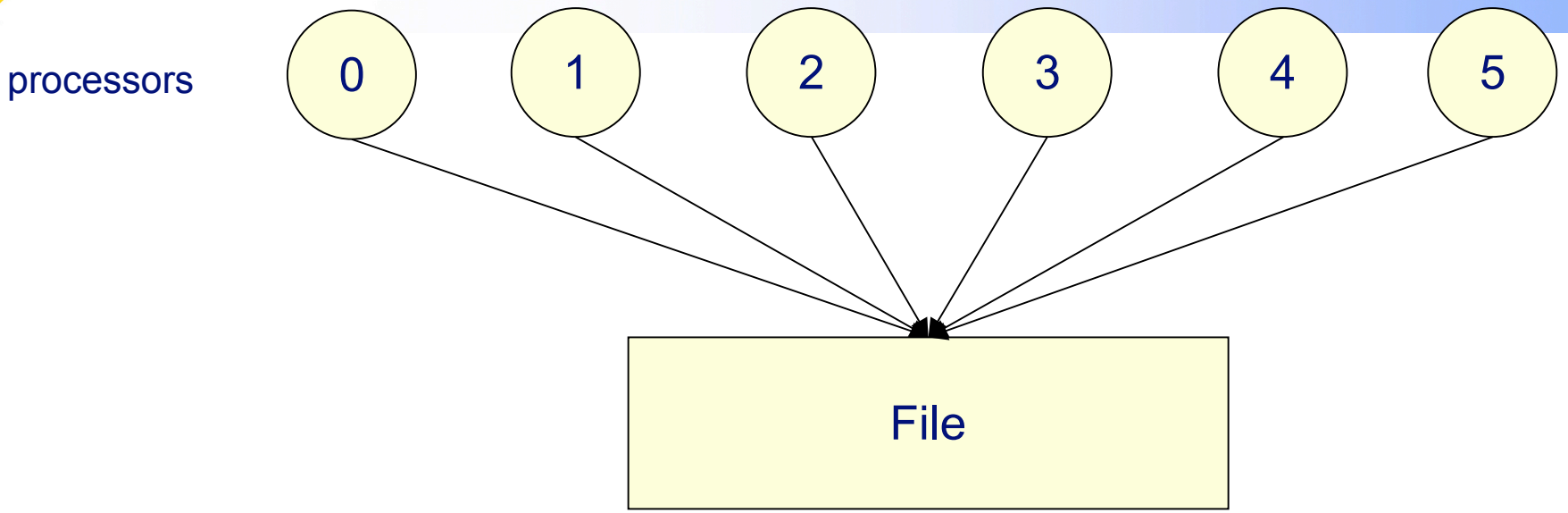
- Each processor sends its data to the master who then writes the data to a file
- Advantages
  - Simple
  - May perform ok for very small IO sizes
- Disadvantages
  - Not scalable
  - Not efficient, slow for any large number of processors or data sizes
  - May not be possible if memory constrained

Parallel I/O Multi-file



- Each processor writes its own data to a separate file
- Advantages
  - Simple to program
  - Can be fast
- Disadvantages
  - Can quickly accumulate many files
  - Hard to manage
  - Requires post processing
  - Difficult for storage systems, HPSS, to handle many small files

Parallel I/O Single-file



- Each processor writes its own data to the same file using MPI-IO mapping
- Advantages
  - Single file
  - Manageable data
- Disadvantages
  - Lower performance than one file per processor at some concurrencies

Common Storage Formats

- **ASCII:**
  - Slow
  - Takes more space!
  - Inaccurate
- **Binary**
  - Non-portable (eg. byte ordering and types sizes)
  - Not future proof
  - Parallel I/O using MPI-IO
- **Self-Describing formats**
  - NetCDF/HDF4, HDF5, Parallel NetCDF
  - Example in HDF5: API implements Object DB model in portable file
  - Parallel I/O using: pHDF5/pNetCDF (hides MPI-IO)
- **Community File Formats**
  - FITS, HDF-EOS, SAF, PDB, Plot3D
  - Modern Implementations built on top of HDF, NetCDF, or other self-describing object-model API

Many NERSC users at this level. We would like to encourage users to transition to a higher IO library

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

What is Striping?

- Lustre file system on Franklin made up of an underlying set of file systems calls Object Storage Targets (OSTs), essentially a set of parallel IO servers
- File is said to be striped when read and write operations access multiple OSTs concurrently
- Striping can be a way to increase IO performance since writing or reading from multiple OSTs simultaneously increases the available IO bandwidth

What is Striping?

- File striping will most likely improve performance for applications which read or write to a single (or multiple) large shared files
- Striping will likely have little effect for the following type of IO patterns
  - Serial IO where a single processor performs all the IO
  - Multiple node perform IO, but access files at different times
  - Multiple nodes perform IO simultaneously to different files that are small (each < 100 MB)
  - One file per processor

NERSC Striping Command Shortcuts

- Unfortunately users need to know about striping in order to get decent I/O performance for I/O intensive applications
- NERSC has tried to encapsulate messy details with 3 commands
- Usage >> stripe_large mydirectory

Size of File	Single File I/O	File Per Processor I/O
<1 GB	Do Nothing Use default striping	“stripe_fpp” or use default striping
1GB - 10 GB	“stripe_small”	“stripe_fpp” or use default striping
10GB - 100 GB	“stripe_med”	“stripe_fpp” or use default striping
100GB - 1TB+	“stripe_large”	Ask consultants

Full Striping Commands

- Striping can be set at a file or directory level
- Set striping on an directory then all files created in that directory with inherit striping level of the directory
- Moving a file into a directory with a set striping will NOT change the striping of that file

lfs setstripe <directorylfile> <stripe size> <OST Offset> <stripe count>

- stripe-size -
  - Number of bytes in each stripe (multiple of 64k block)
- OST offset -
  - Always keep this -1
  - Choose starting OST in round robin
- stripe count -
  - Number of OSTs to stripe over
  - -1 stripe over all OSTs
  - 1 stripe over one OST

Recommendations

- Think about the big picture
  - Run time vs Post Processing trade off
  - Decide how much IO overhead you can afford
  - Data Analysis
    - Is there analysis you can do during your production run?
  - Portability
  - Longevity
    - H5dump/ncmpidump works on all platforms
    - Can view an old file with h5dump/ncmpidump
    - If you use your own binary format you must keep track of not only your file format version but the version of your file reader as well
  - Storability

Recommendations

- Use a standard IO format, even if you are following a one file per processor model
- One file per processor model really only makes some sense when writing out very large files at high concurrencies, for small files, overhead is low
- If you must do one file per processor IO then at least put it in a standard IO format so pieces can be put back together more easily
- Follow striping recommendations
- Consider the value of your time -- even if your advisor is not
- Ask the consultants, we are here to help!

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Portability and Flexibility

- HPC machines change quickly
  - NERSC does technology refresh every 3 years
  - Always consider challenge for users moving to machine but we need to keep up with technology and market
- Codes become more robust when run on multiple platforms with multiple compilers
- Familiarity with different programming models, performance tools and debuggers gives you an advantage
- Be prepared to be able to move from center to center - go where the cycles are

NERSC Science Over the Years

Please let us know what we can do to help!

Consult@nersc.gov