GPFS for Life Sciences at NERSC

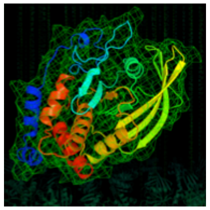
A NERSC & JGI collaborative effort

Jason Hick, Rei Lee, Ravi Cheema, and Kjiersten Fagnan

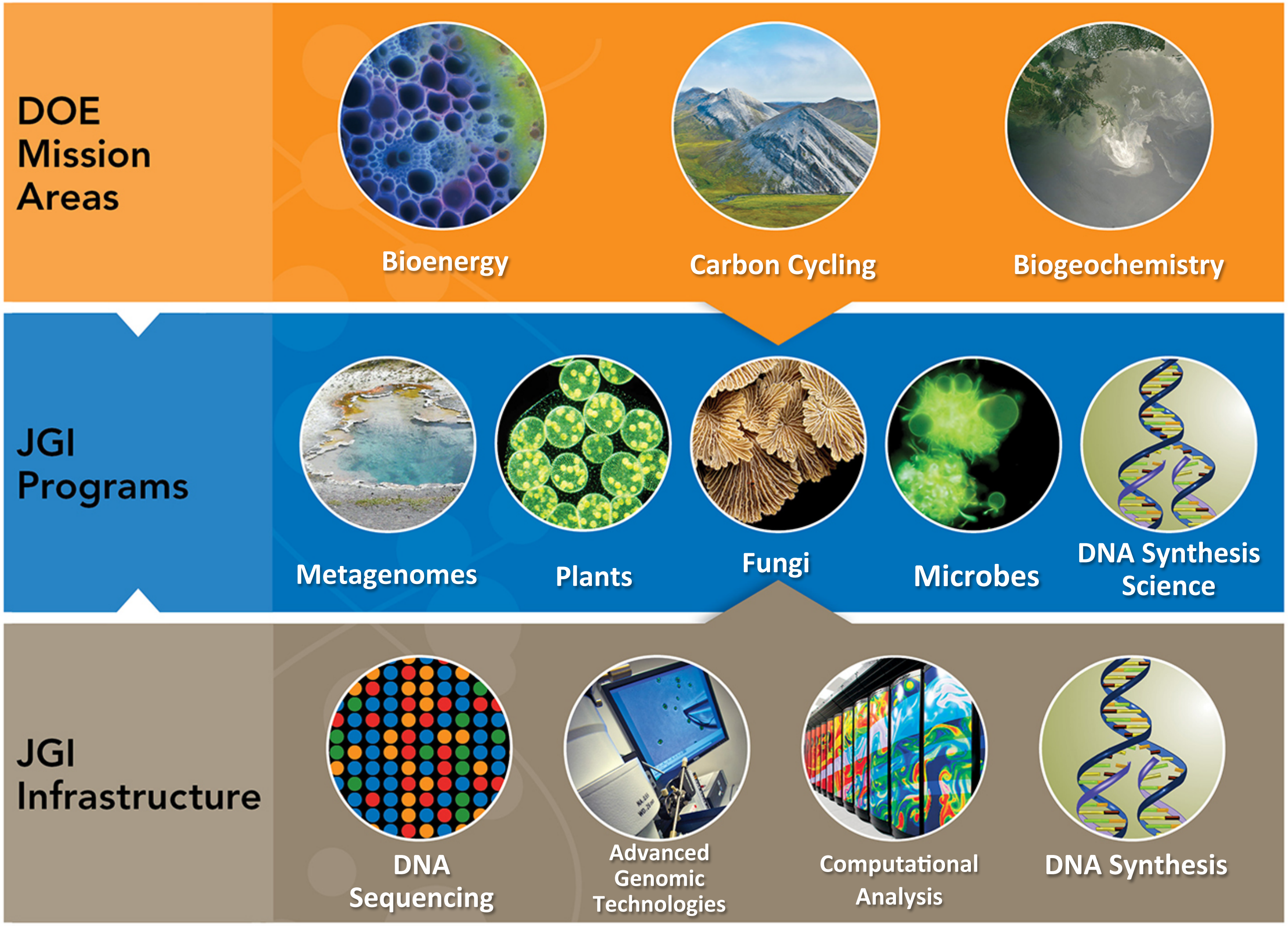
GPFS User Group meeting

May 20, 2015

Overview of Bioinformatics



A High-level Summary



DOE Mission Areas

Bioenergy

Carbon Cycling

Biogeochemistry

JGI Programs

Metagenomes

Plants

Fungi

Microbes

DNA Synthesis Science

JGI Infrastructure

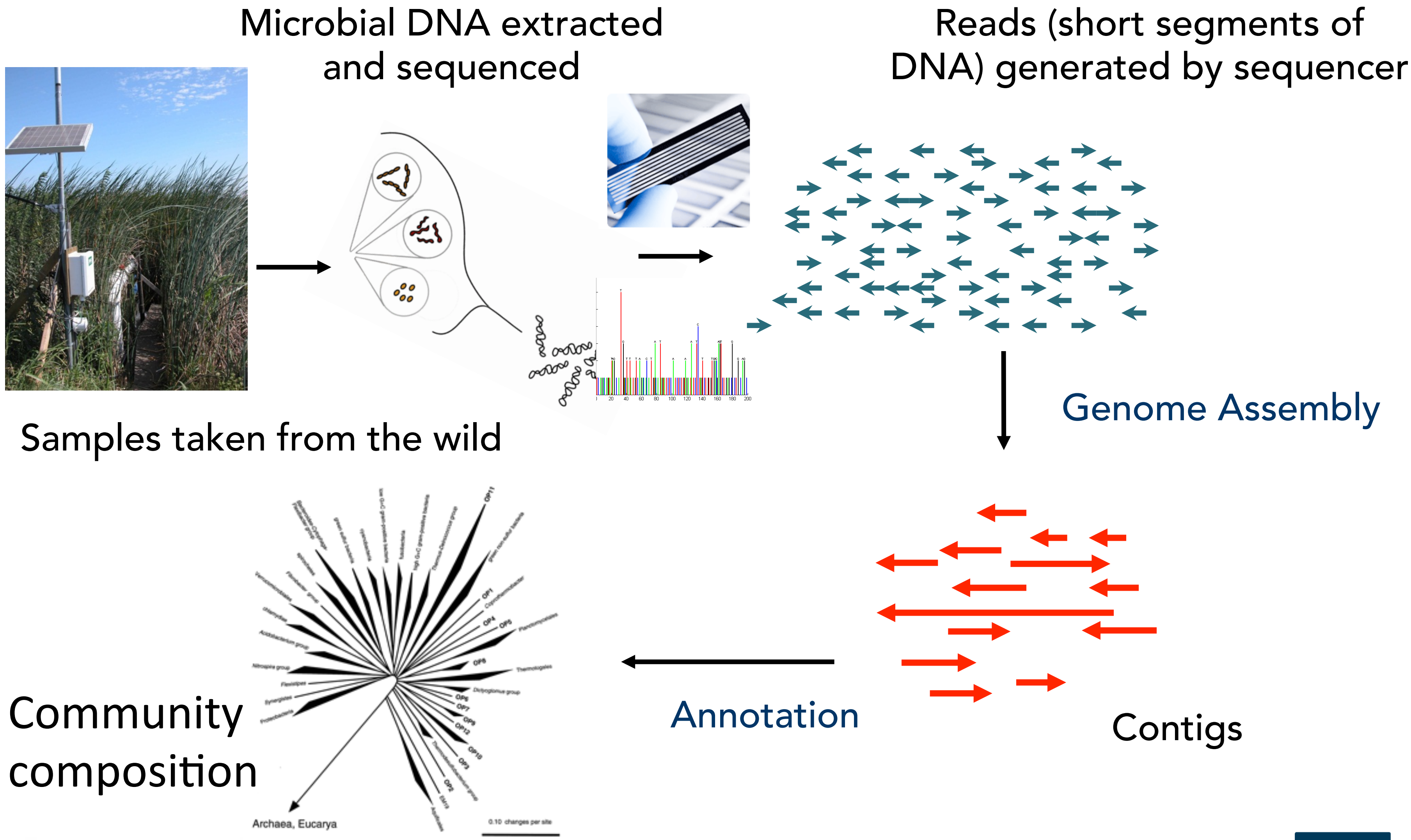
DNA Sequencing

Advanced Genomic Technologies

Computational Analysis

DNA Synthesis

Metagenome Analysis



Microbial DNA extracted and sequenced

Reads (short segments of DNA) generated by sequencer

Samples taken from the wild

Genome Assembly

Contigs

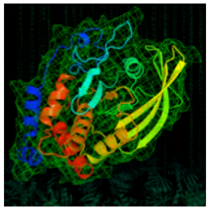
Annotation

Community composition

JGI and the NERSC partnership

NERSC

JGI

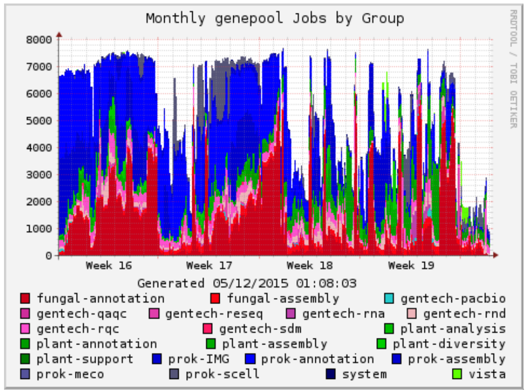


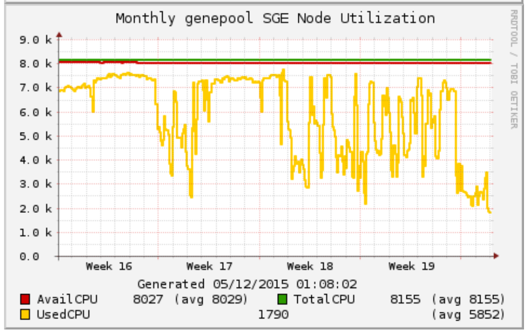
JGI and NERSC partnership

- **The U.S. DOE's Joint Genome Institute is a leading bioinformatics research facility**
  - Good relations with other established bioinformatics facilities, such as WUSTL to receive guidance
- **NERSC accelerates scientific discovery by providing high performance computing and storage as a U.S. DOE user facility**
- **Partnered in 2010 to consolidate computing and storage infrastructure at NERSC facility**
- **At the time of the partnership, JGI had:**
  - Numerous group owned clusters
  - Around 21 NFS-based file systems serving predominantly as archival storage
  - Two Isilon file systems handling sequencing storage, cluster output, and desktop storage

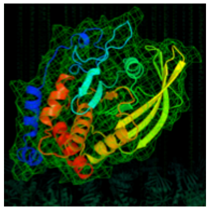
Initial assessment

- **Utilization on group clusters was sporadic and increases in sequencing were difficult to translate to computing needs**
  - Heterogeneous jobs and predominantly high throughput computing
  - Needed a centralized cluster to provide a scalable solution for making use of additional sequencing
  - Consolidated onto new fair-share cluster called Genepool
- **Pre-existing ethernet interconnect presented challenges**
  - Serial workloads preferred
  - Under-provisioned, causing high contention to storage
- **File systems were preventing them from scaling up, 2PB with 1B files**
  - Regular hangs and timeouts
  - Bandwidth was too low
  - Metadata performance was low and accounting didn't complete
  - Backups not completing
  - No storage system administrator to help resolve these issues





Initial GPFS deployment



Match different workloads to different storage systems

- **Retired Netapps filers by migrating data to HPSS archival storage**
  - 21 Netapp filers, 7 years or older
  - Users resistant at first, but were surprised at performance!
  - Developed their own data management interface (JAMO) that moves data automatically between file system and archive
- **Introduced new GPFS scratch file system to Genepool cluster**
  - Alleviated load on existing Isilon file system allowing us to decide how to use it moving forward
  - Implemented fileset quotas to help JGI balance and manage their storage allocations
  - Implemented new purge policy in combination with archival storage for establishing new user-based data management

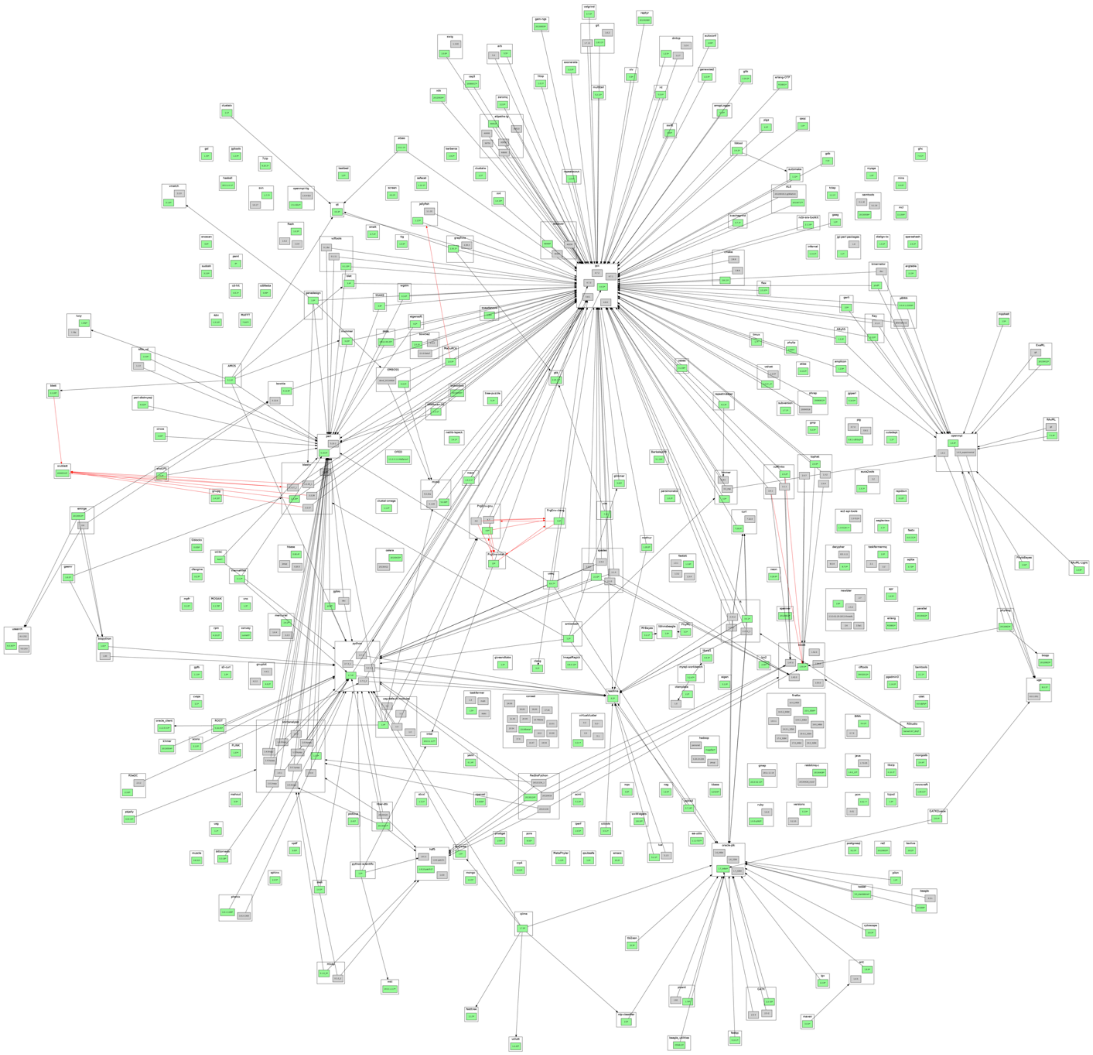
What a diverse workload

Now we're getting to the bottom of it

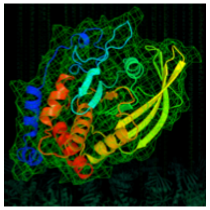
- **Lots of churn, generating O(100M) KB-sized files and then deleting them**
  - We haven't addressed this one yet
  - Especially challenging with use of snapshots and quotas
- **Users requesting 10's of GBs per second of bandwidth**
  - Encouraged use of separate file system who's sole purpose is large file I/O (10's of GBs per second)
- **Production sequencing very important to not disrupt**
  - Peak demand is about 5TB per day
  - Created another file system called "seqfs" with a very limited number of mounts to nearly exclusively handle sequence machine runs
- **Many desire read-caching for their workload (BLAST)**
  - GPFS cache getting blown by writes, algorithm not good for reads
  - Created another file system called "dna" predominantly read-only mounted

Complex software environment

- The genepool system has over 400 software packages installed
- Different users require alternative versions of the software
- The storage problem here is that all users/ jobs care how quickly their software loads!



Specific Improvements



Implement disaster protection

- **Enabled GPFS snapshots**
  - Aid in recovering data from accidental user deletes
- **Backups to HPSS**
  - Custom software we call PIBS to perform incremental backups on PB-sized file systems

Optimizing GPFS for Ethernet cluster

- **Adjust TCP kernel setting**
  - Need Smaller initial send buffers
    - net.ipv4.tcp_wmem
    - net.ipv4.tcp_rmem
  - Prevent head-of-line blocking (saw congestion like symptoms without congestion traffic, result of flow control)

GPFS and unsupported kernels

- **They preferred Debian and initially used Debian 5 with GPFS 3.3**
  - This was a bad idea
  - Symptom was high degree/volume of expels
    - Memory errors causing GPFS asserts
- **Switched to Debian 6 with GPFS 3.4**
  - All memory errors ceased
  - Drastically reduced the number of expels

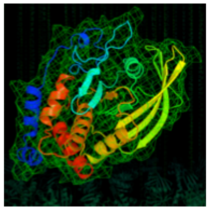
GPFS caching

- **Life sciences prefer user space allocations**
  - We disabled swap, which was key to preventing out-of-memory problems on their compute cluster
  - Experimented increasing pagepool but didn't help the broader life sciences workload
  - Moved to CNFS approach for better read caching
    - Works for broader set of workload
    - However unknown as to whether this scales for either whole file system, so limiting this to specific subdirectory/fileset of GPFS file system
      - We have different CNFS servers to isolate NFS from native GPFS
- **We would be interested in new options for tuning/using memory in GPFS**

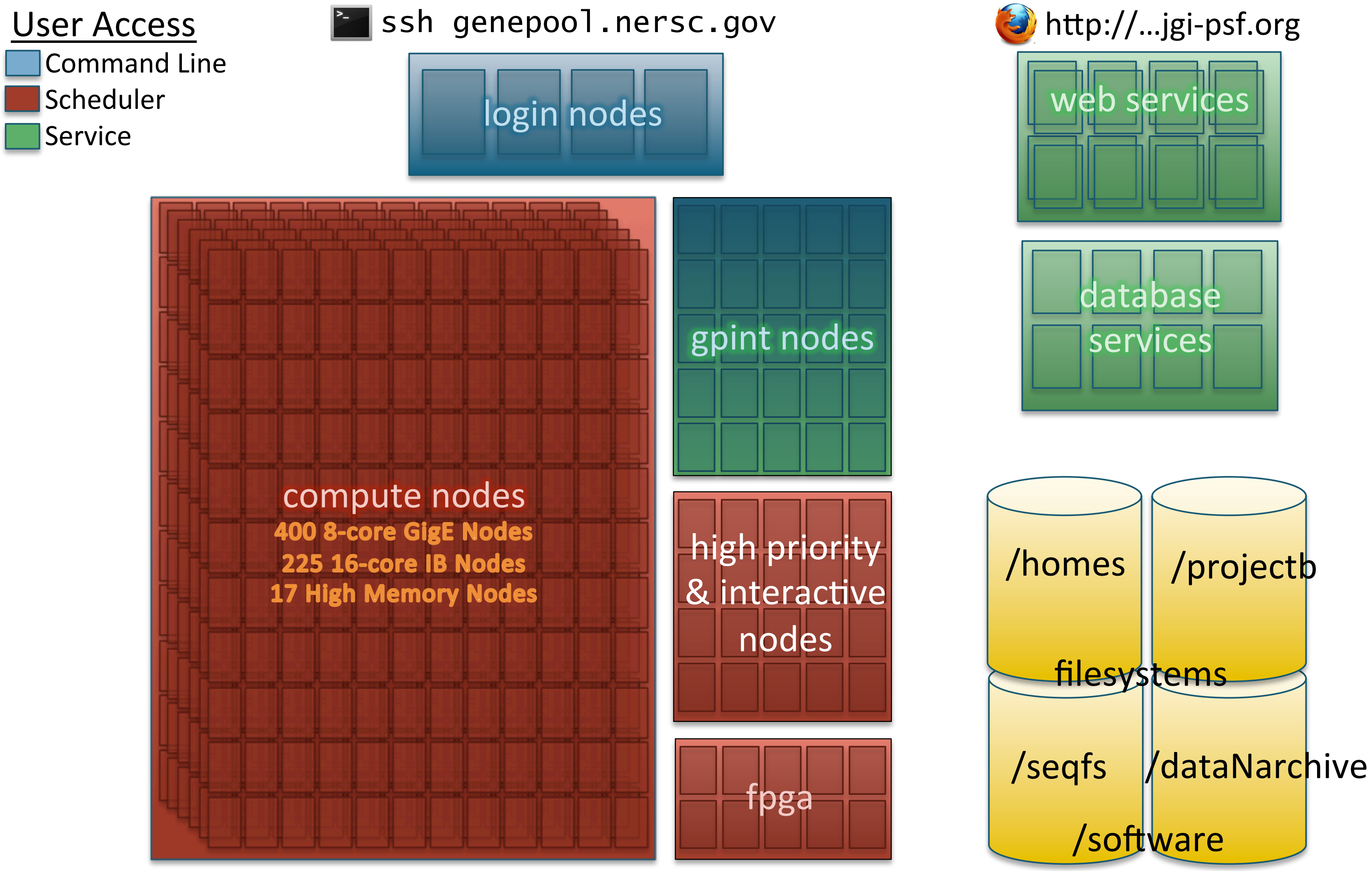
Moving from Ethernet to IB

- **Genepool has nodes on either ethernet or IB**
  - IB expels much less frequent, performance more consistent, but challenges are routing/topology
    - To isolate/scale GPFS separately from compute IB, deploying custom gateway servers routing between compute IB and storage IB
    - Deploying custom gateway servers to route ethernet over storage IB
  - Ethernet flow control/fair share and normal architecture (L1/L2) do not enable GPFS to perform adequately for JGI workloads
    - Detuning stabilizes GPFS for availability (i.e. eliminates expels) but our performance was less than 1GB/sec per compute node, with higher variability in performance

Resulting Architecture Today



JGI’s Compute Cluster at NERSC



User Access

🟦 Command Line

🟥 Scheduler

🟩 Service

```
ssh genepool.nersc.gov
```

http://...jgi-psf.org

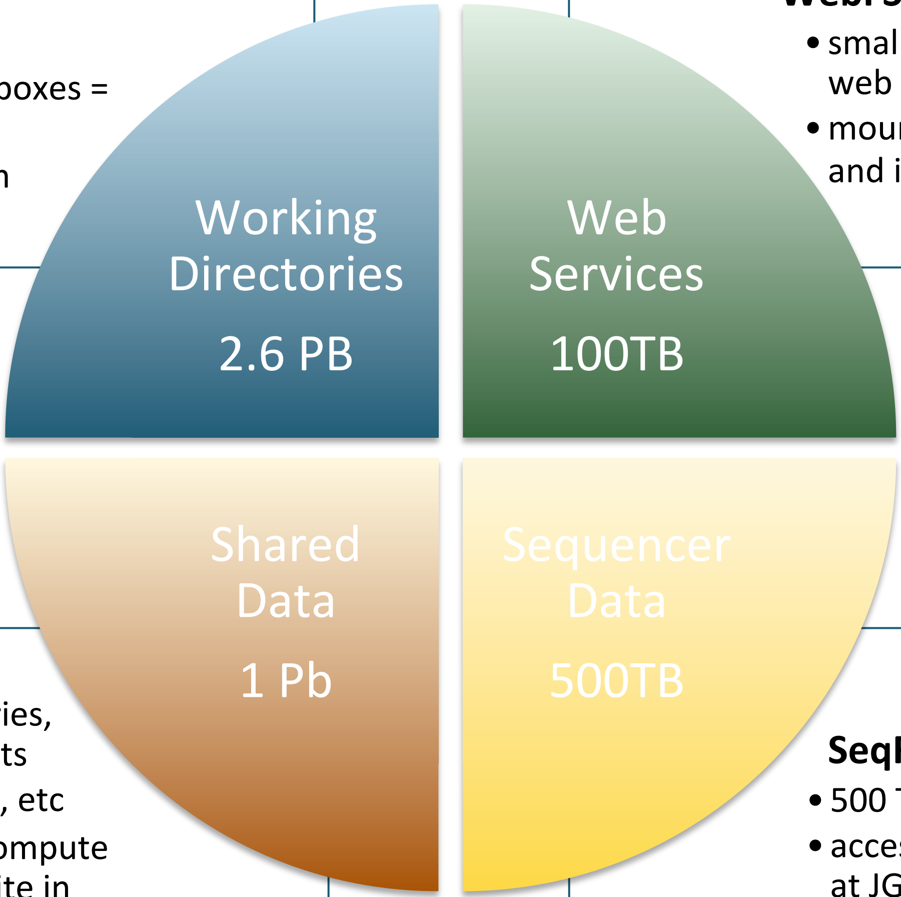
JGI Data Storage

/projectb

- 2.6PB
- SCRATCH/Sandboxes = “Wild West”
- Write here from compute jobs

WebFS

- small file system for web servers
- mounted on gpwebs and in xfer queue



Working Directories

2.6 PB

Web Services

100TB

Shared Data

1 Pb

Sequencer Data

500TB

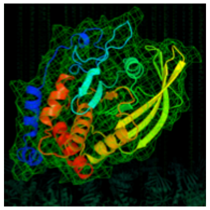
DnA

- Project directories, finished products
- NCBI databases, etc
- Read-only on compute nodes, read-write in xfer queue

SeqFS

- 500 TB File system
- accessible to sequencers at JGI

Future Directions



Desired root cause analysis of expels

- **Explored using GPFS callbacks to collect data on node expels**
  - Ultimately want to determine health of node
  - Gathered information counter on network interfaces, sent IB/ethernet pings
  - However, there still lacks a central method of detecting issues with remote clusters (issue only sent to remote cluster manager)
  - GPFS 4.1 sends notifications of congestion to owning cluster, a major improvement, but still not enough to determine node health

Other initiatives underway

- **Scheduler upgrade/enhancements**
  - Consider better features for job dependencies
  - Optimizations for high throughput workload
- **HPC Initiatives study**
  - Identify changes necessary to enable bioinformatics workload to on the largest HPC systems
- **Workflow software**
  - Help manage work external to compute system scheduler
- **Data management tools**
  - Evaluating different software for loose coupling of GPFS and HPSS systems (SRM/BeStMan, iRODS, GPFS Policy Manager, …)
- **Consider small file optimizations**
  - File System Acceleration (FSA) using DataDirectNetwork's Infinite Memory Engine (IME) in front of GPFS

Summary

- **Life sciences workloads:**
  - Predominantly high throughput computing, we consider it data intensive computing
  - Diverse in their demands on file systems
    - Segregating workloads into separate file systems was extremely helpful (latency sensitive to bandwidth demanding, optimizations for reading)
  - Benefit from using archival storage (e.g. HPSS) to improve data management
  - Required special data management software
    - They developed their own solution, called JAMO to move data between archive and file system
  - Drastic availability improvements when shifting to IB over ethernet
- **GPFS works well for the JGI**
  - Eager to explore ideas for isolating small file workloads