GPFS Experiences from the Argonne Leadership Computing Facility (ALCF)

William (Bill) E. Allcock

ALCF Director of Operations

Argonne National Laboratory

- Argonne National Laboratory is located on 1,500 acres, 25 miles southwest of downtown Chicago.

A Quick Summary of Argonne

- 3450 full-time employees, 1250 of which are scientists and engineers, 290 post-docs
- Currently 8 major initiatives:
  - Five of them are applied areas, solving significant national issues
    - Biological and Environmental Systems: climate change and its impacts, computational models of biological processes and their economic, social, and health impact.
    - Alternative Energy and Efficiency: Solar, alternate vehicle fuels, advanced engines
    - Nuclear Energy: advanced fuel-cycle technologies that maximize the efficient use of nuclear fuel, reduce proliferation concerns and make waste disposal safer and more economical.
    - National Security: threat decision science, sensors and materials, infrastructure assurance, emergency response, cyber security and the nonproliferation and forensics of radiological and nuclear materials and equipment
    - Energy Storage: high performance batteries for all-electric vehicles, storage for the national electric grid and manufacturing processes for these materials-intensive devices
  - Three of them aim to develop enabling next generation tools:
    - Hard X-ray Sciences: leading-edge high-energy X-ray techniques to study materials and chemical processes under real conditions in real time
    - Materials and Molecular Design and Discovery: making revolutionary advancements in the science of materials discovery and synthesis — predicting, understanding and controlling where and how to place individual atoms and molecules to achieve desired material properties
    - Leadership Computing: High-performance computing is becoming increasingly important as more scientists and engineers use modeling and simulation to study complex chemical processes, exotic new materials, advanced energy networks, natural ecosystems and sophisticated energy technologies

Production Systems: ALCF-2

Mira – *BG/Q system*
- 49,152 nodes / 786,432 cores
- 786 TB of memory
- Peak flop rate: 10 PF
- Linpack flop rate: 8.1 PF

Vesta - *BG/Q system*
- 2,048 nodes / 32,768 cores
- 32 TB of memory
- Peak flop rate: 419 TF

Cetus - *BG/Q system*
- 1,024 nodes / 16,384 cores
- 16 TB of memory
- Peak flop rate: 209 TF

Tukey – *NVIDIA system*
- 100 nodes / 1600 x86 cores
- 200 M2070 GPUs
- 6 TB x86 memory / 1.1TB GPU memory
- Peak flop rate: 220 TF

Storage - Scratch: 28.8 PB raw capacity, 240 GB/s bw (GPFS); Home: 1.8 PB raw capacity

Production Systems: ALCF-1

Intrepid – *BG/P system*

- 40,960 nodes / 163,840 cores
- 80 TB of memory
- Peak flop rate: 557 TF

Challenger – *BG/P system*

- 1,024 nodes / 4,096 cores
- 2 TB of memory
- Peak flop rate: 13.9 TF

Surveyor – *BG/P system*

- 1,024 nodes / 4,096 cores
- 2 TB of memory
- Peak flop rate: 13.9 TF

Eureka – *NVIDIA S-4 cluster*

- 100 nodes / 800 2.0 GHz Xeon cores
- 3.2 TB of memory
- 200 NVIDIA FZ5600 GPUs
- Peak flop rate: 100 TF

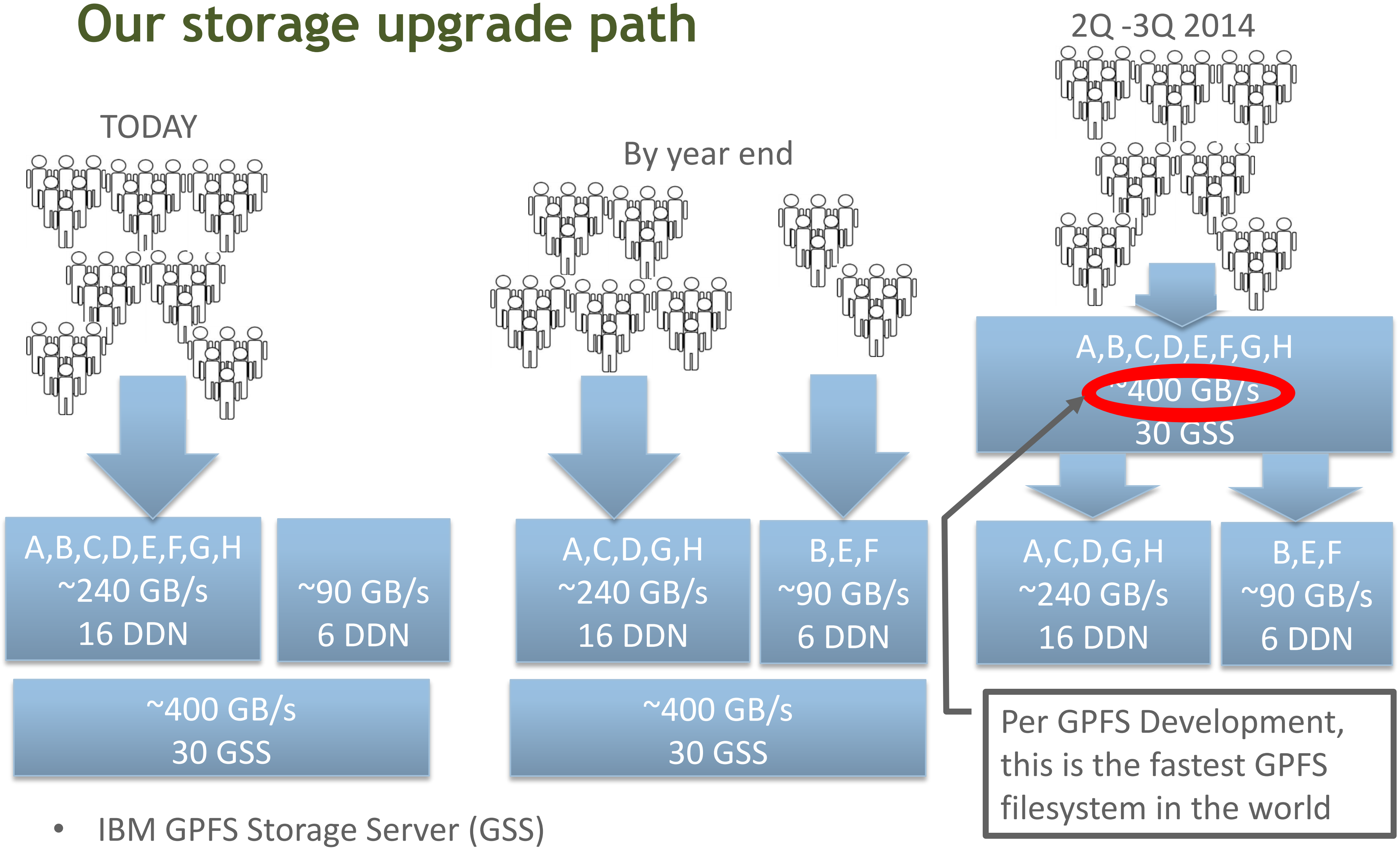
Storage

- Scratch: 6+ PB capability, 80 GB/s bandwidth (GPFS and PVFS); Tape: 15.9 PB of archival storage, 15,906 volume tape archive (HPSS)

Storage at ALCF

- Our primary file system is GPFS
  - We also run a 500TB PVFS file system on Intrepid, but are GPFS only on Mira
- We have different storage policies than most of the other DOE labs
  - We do not purge at all
  - On intrepid, it was the honor system and we nagged when things got full; It worked, but not well.
  - On Mira, we have quotas in effect, which forces the users to manage their disk space.
  - Weird side effect: We have more available disk capacity than tape capacity
- We basically had one file system for each "turn of the crank" and everything mounted that one file system
  - Very convenient for the users; Very little time spent on data management and movement.
  - However, single point of failure.
  - We ran this way for 5 years without problems, then had a hard power failure and found disk corruption. For a variety of reasons, it took nearly 3 weeks to fsck our 4.5PB file system and get it back online
  - Having a second file system (PVFS) saved our bacon, because we had a few users that could run from there.
- We had just configured Mira the same way when the corruption hit. You can imagine what happened next…

Our storage upgrade path



Per GPFS Development, this is the fastest GPFS filesystem in the world

- IBM GPFS Storage Server (GSS)
  - Uses GPFS "Native RAID" (Software RAID)
  - No controllers; x3650 servers with attached JBODs

“Burst buffer like” solution

- For those of you in the DOE HPC space, you will know what I mean; For those of you who are not, think “L1 cache for storage”.
- Before anyone yells “off with his head” I said burst buffer like, but it is not a burst buffer by the most common proposal (they don’t exist yet)
- It does, however, share some characteristics
  - Fast, but not big enough to permanently keep the data there (a cache)
  - Improves performance for most common scenarios (defensive I/O, possibly in-situ analysis)
  - Forces us to deal with how to manage moving data in and out without user intervention
- It keeps the aspects of our storage system that we like, but avoids the ones we don’t:
  - To the user, it still looks like a single namespace visible to everything
  - They will see better performance than today
  - We have three file systems so if one is down, some of our users can still run
- Makes use of GPFS Active File Management (AFM)
  - But not in the way GPFS developers planned ☺
  - They designed it primarily for WAN file systems, so performance characteristics are different, particularly the speed of migration into and out of the cache.
  - Collaborating with GPFS development team; Specific features are being added to support the “burst buffer scenario”.
  - I am not going to talk about them because they are NDA, but IBM can if they would like ☺

We would like to take this one more step…

- We already use HPSS (tape storage)
- GPFS – HPSS – Interconnect (GHI)
  - Use HPSS as a hierarchical storage back end for GPFS
  - Policy driven object store
    - Make a copy of a file on disk to tape "shortly" after creation (wait a while in case they delete)
    - If the file goes some period of time (weeks or months) without being accessed delete the copy on disk
    - It still shows up in the GPFS metadata and will be automatically be migrated back to disk if accessed.
      - We have requested a change to this functionality. We would like to be able to configure it to throw an error instead to avoid accidental major migrations.
  - Gives us a "backup" of scratch, something most HPC centers (including us) do not do
  - Ideally, users never have to touch their data; Policy will automatically take care of migrating to tape and removing data from disk that has not been used; Entry stays in GPFS metadata and can be explicitly recalled by command or file is automatically recalled from tape if it is accessed.
    - This includes long term archival after a project ends.
    - Use GPFS filesets; New feature allows filesets to have their own inode pool, so don't have to worry about inode pool getting too large (at least we hope, proof will be in the pudding)
    - When a project ends, policy will move all the data from disk, but leave their fileset in place. If they every come back and ask for a file, it is a simple copy.