High Performance NFS

Roger Haskin
Senior Manager, File Systems
IBM Almaden Research Center

NFS4 and Parallel NFS Work at IBM Research

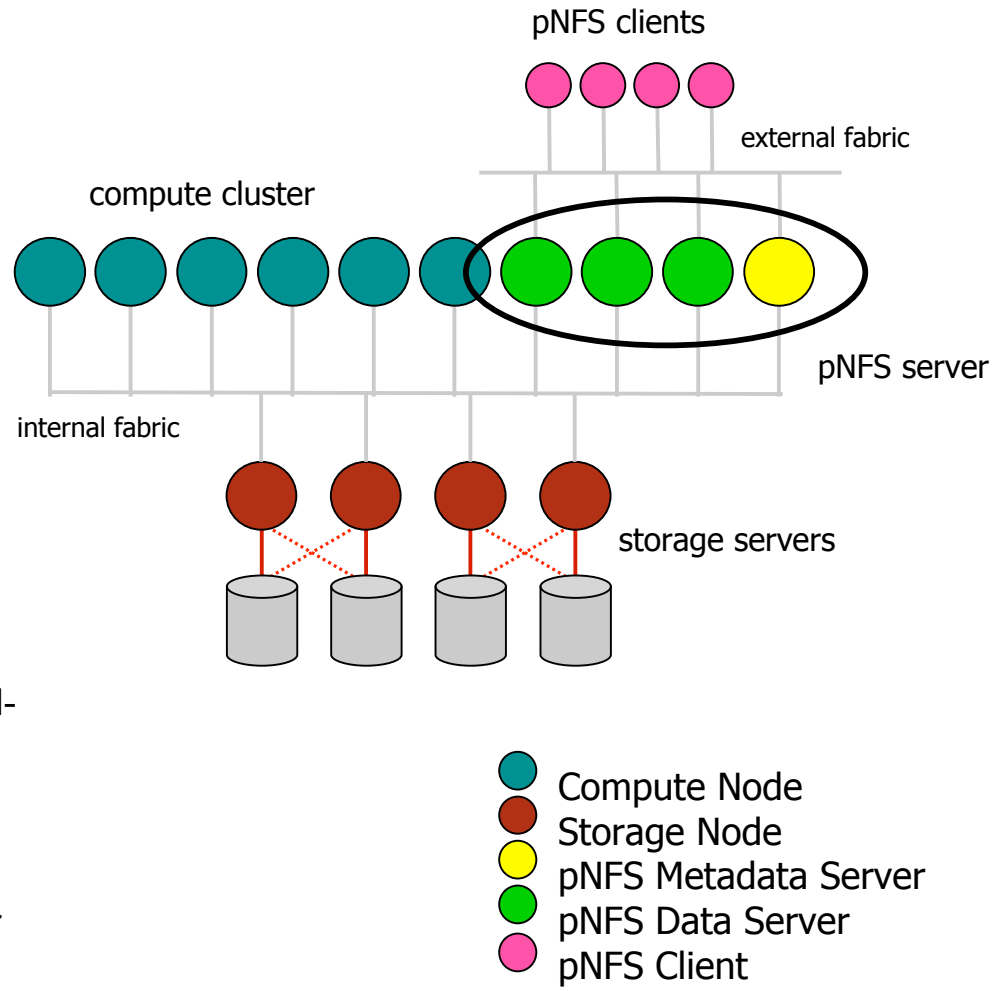
- IBM Almaden Research has a long history of work on high-performance storage projects
  - The GPFS file system originated at Almaden
- We are participating in the development of NFSv4 features including redirection
- With CITI at University of Michigan, Network Appliance, Panasas and others, we are participating in the development of parallel NFSv4 (pNFS) for Linux

High Performance NFS

- In the context of this conference, “high performance computing” implies parallel computing…
- … ergo, though technologies like RDMA are important for individual clients and servers, true “high performance” implies parallel NFS
- Work is progressing on parallel NFSv4 (pNFS), results to date are encouraging
- pNFS in NFSv4.1, now Internet Draft but basically done
  - File variant only
  - Object and block variants in separate Internet draft
- Who is doing what?
  - Sun: Solaris client and server implementation
  - Network Appliance: server implementation for NetApp filers
  - Linux client common code: CITI, NetApp, IBM (Almaden), Panasas
  - Linux client/file driver: CITI, NetApp, Almaden
  - Linux server/file: CITI, Almaden (on GPFS and PVFS file systems)
  - Linux server/object: Panasas
  - Linux server/block: CITI, with support from EMC
- “Bake-a-thon” in September at CITI
  - File variant demoed with Linux & Sun clients, Sun, NetApp, and Linux server, latter running on both GPFS and PVFS

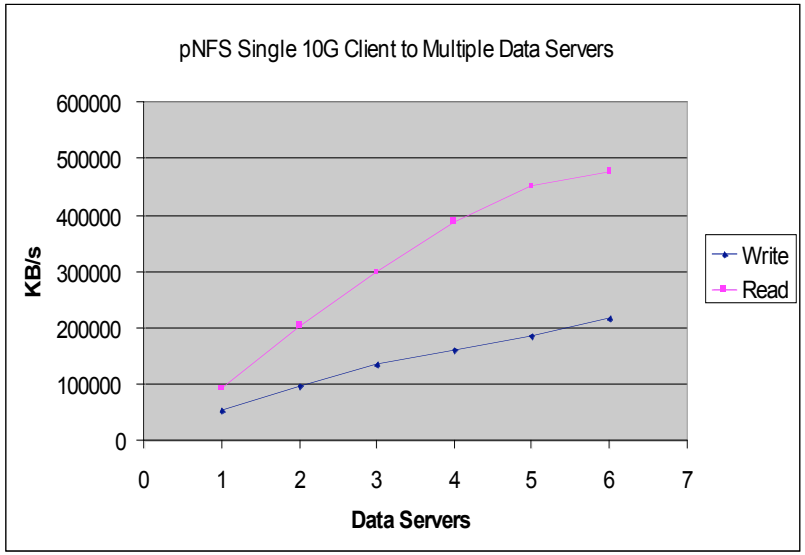
Parallel NFSv4 (pNFS) prototype on GPFS

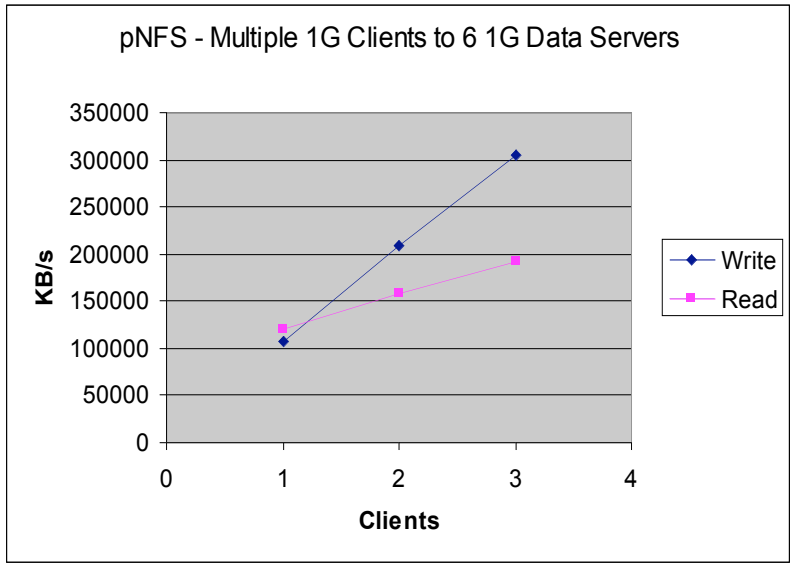
- pNFS extends NFSv4 to support parallel access
  - Clients use metadata server to create/delete/open/close
  - Clients get map of data from metadata server
  - Map directs clients to data servers for read/write operations
- Linux pNFS prototype on GPFS
  - GPFS modified to enable pNFS metadata server (i.e. to return map)
  - Other GPFS nodes are pNFS data servers
  - GPFS allows any node to be a metadata server, so clients can be load-balanced across the cluster
  - GPFS allows any data server to serve any data; metadata server chooses map to load-balance across data servers
  - pNFS server cluster can be part of a compute cluster or can be its own stand-alone pNFS server cluster
  - Similar implementations possible with other parallel file systems



pNFS performance scaling

- pNFS addresses two scenarios:
  - High-performance (e.g. 10GbE) client driving a server cluster
  - Parallel/cluster clients driving a server cluster
- Test runs on small cluster at CITI
  - Single 10GbE client accessing server cluster of varying size
    - iozone -aec -i 0 -i 1 -+n -r 256K -s 1G -f /mnt/nfs4/io1 -U /mnt/nfs4
  - Varying number of 1GbE clients accessing 1GbE server cluster
    - mpirun -np {1-3} -machinefile /tmp/gpfs.nodes /nas/test/IOR -b 1g -t 32k -g -e -F -o /mnt/ior/t1





Will pNFS conquer the world?

- Probably not soon!
- Main create/delete/open/close/read/write path working, but recovery path still under development
  - Recovery for failed data server is hard
  - Probably a year to complete this at current course and speed
- Don't try this at home, folks!
  - Tuning needed, performance anomalies on less-than-perfect hardware
- Will pNFS compete with parallel file systems?
  - Similar technology (SANergy, High Road) is not ubiquitous in HPC
  - pNFS is clearly better for some access patterns (e.g. file-per-process) than others (parallel access to shared file)
    - NFS consistency semantics
    - File-level delegations
  - Parallel file systems have more opportunity to do global optimization
    - They typically know the cluster members
    - They can see the global activities of the application