pNFS: Parallel NFS

Bergwolf@linuxfb.org

Agenda

pNFS Technology Overview

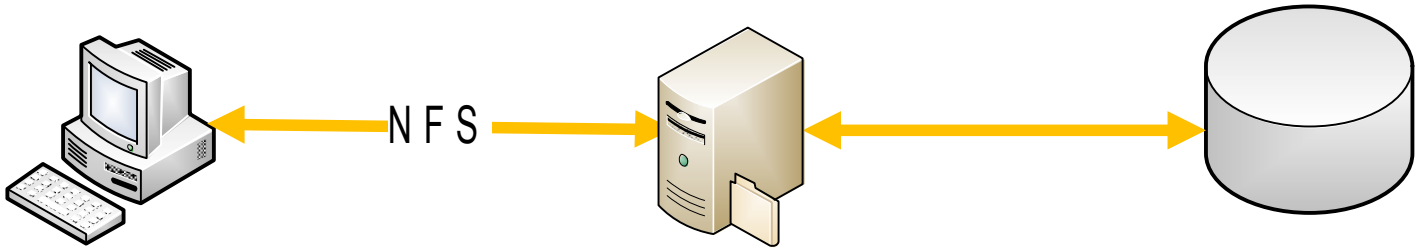
Timeline

Future Work

Discussions & Questions

Traditional NFS limitations

- Single NFS server bottleneck: limited bandwidth & CPU
- Multiple NFS servers are separate storage islands

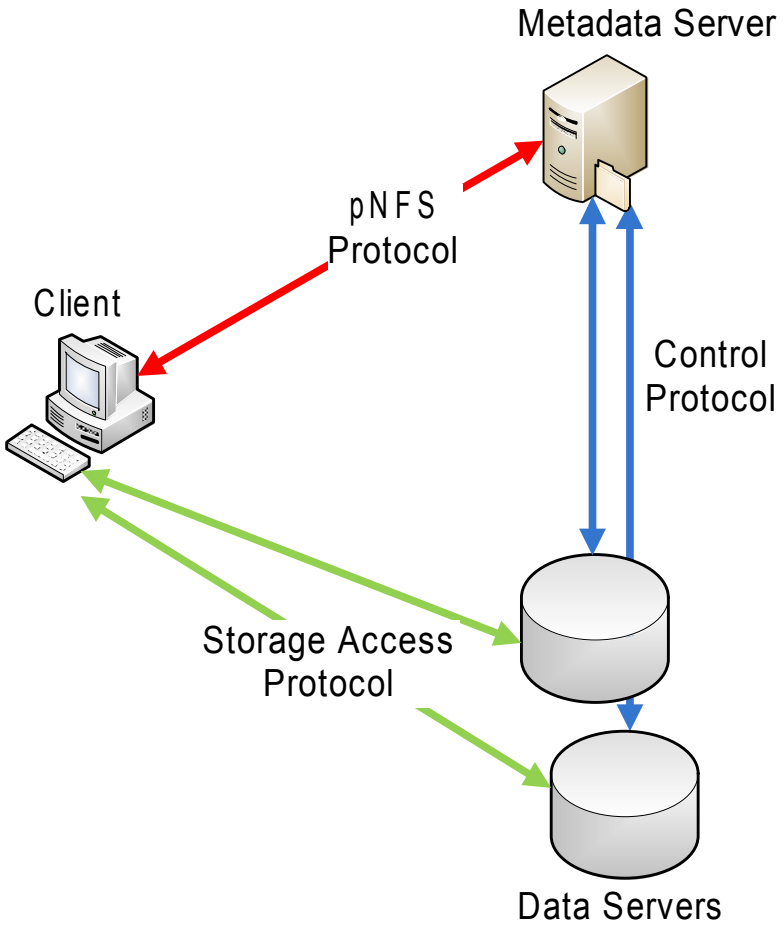


Proprietary Solutions

- SAN (IBM, EMC, Sun) and Object (IBM, HP, Panasas) file systems:
  - Clients bypasses metadata server to access data
  - Each system is doing differently implementation
- Scalable/cluster NFS servers:
  - Facilitate scalable aggregate load
  - Do not provide scalable/parallel bandwidth to a single file

What Is pNFS?

- pNFS protocol
  - NFSv4.1
- Storage access protocol
  - File (NFSv4.1)
  - Block (FC, iSCSI, FCoE)
  - Object (OSD2)
- Control protocol
  - Implementation decision

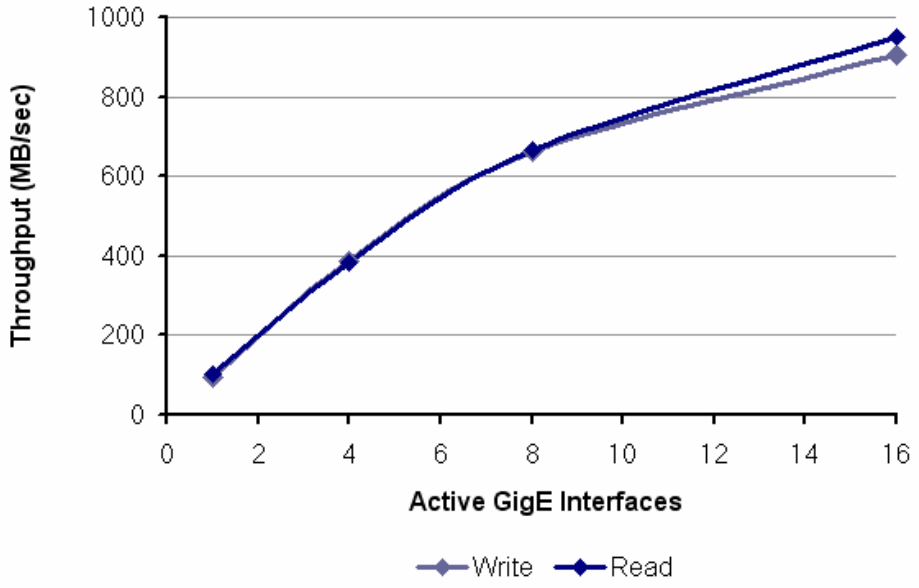


pNFS Advantage

- Distributed data access across the storage cluster
- Reduce load on NFS server
- I/O Acceleration

pNFS Performance

- Linear I/O increase
- Client performance limitation is on backend
- Numbers taken with EMC MPFS, from which pNFS block layout derives



pNFS Related Standards

- RFC 5661: NFSv4.1, including pNFS file layout
- RFC 5662: NFSv4.1 XDR representation
- RFC 5663: pNFS block/volume layout
- RFC 5664: pNFS object operations

Other NFSv4.1 Key Features

- EOS (Exactly Once Semantics)
- File and directory delegation
- No more “silly rename”
- Data retention
- Session trunking and client ID trunking

pNFS Timeline

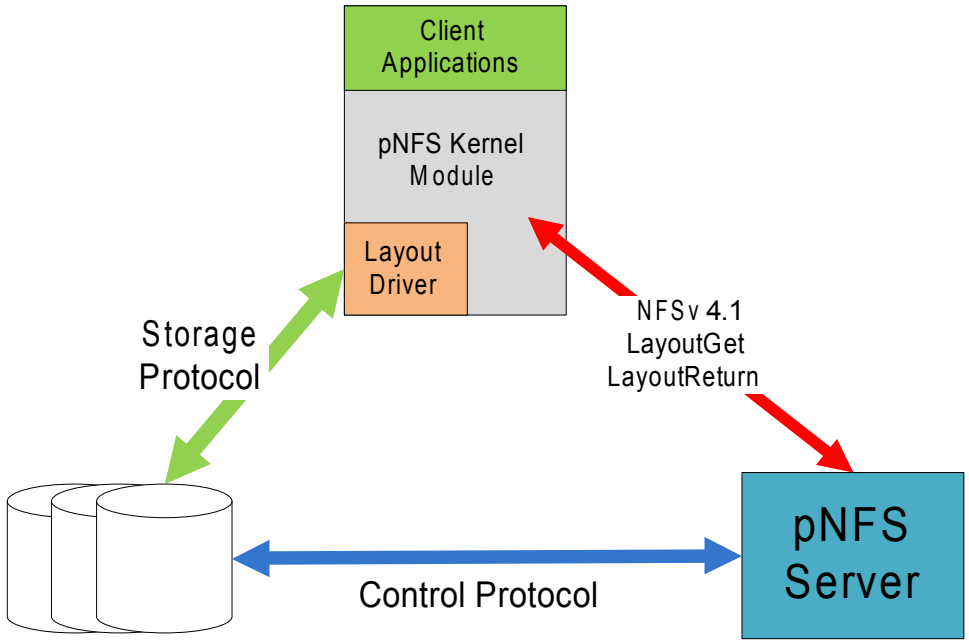
- 2004 — CMU, NetApp and Panasas drafted pNFS problem and requirement statements
- 2005 — CITI, EMC, NetApp and Panasas drafted pNFS extensions
- 2005 — NetApp and Sun demonstrated pNFS at Connectathon
- 2005 — pNFS added to NFSv4.1 draft
- 2006-2008 — specification baked
  - Bake-a-thons, Connectathon
  - 26 iterations of NFSv4.1/pNFS spec
- 2009 — RFC submitted
- 2010 — RFC published
- 2010 — Fedora 10 includes pNFS server/client gits and rpms
- 2010 — RHEL6.x is promised to have pNFS

Industry Support

- BlueArc
- CITI
- CMU
- EMC
- IBM
- LSI
- NetApp
- Ohio SuperComputer Center
- Panasas
- Seagate
- StorSpeed
- Sun Microsystems (now Oracle)

Linux pNFS Client

- Applications transparency
- Fewer support issues for storage vendors
- Generic pNFS client with layout driver plugins



pNFS Primitives

- Layout
  - Describes how a file is distributed among storage
- Segment
  - Each layout has one or more segments. A segment is a distribution pattern.
- Device
  - Describes the storage (file server, backend block device, OSD data server)
- Layout OPs
  - Layoutget, layoutreturn, layoutcommit, cb_layoutrecall

Linux pNFS Client Driver API

- struct pnfs_layoutdriver_type
  - layout driver specific functions (segment ops, layout ops)
- struct pnfs_layout_hdr
  - generic layout header
- struct pnfs_device
  - generic device
- struct pnfs_layout_segment
  - generic layout segment

Linux pNFS Merge Plan

- End of 2010
  - Linux 2.6.37 — Files pNFS client and server
- Feb, 2011
  - Linux 2.6.38 — Object pNFS client and server
- May, 2011
  - Linux 2.6.39 — Block pNFS client and server

What Is Expected In NFSv4.x

- File, Block, Object layout driver
- Server side copy
- Federated file system

Open Issue – Layout stateid, open mode relationship

Client Behavior	Server Returns
OPEN read-only	open stateid1
LAYOUTGET with IOMODE_READ	layout stateid1
CLOSE	
OPEN read-write	open stateid2
LAYOUTGET with IOMODE_READ using open stateid2	layout stateid1
LAYOUTGET with IOMODE_RW using layout stateid1	NFS4ERR_OPENMODE

Open Issues

- Is LAYOUTRETURN still necessary for forgetful mode?
- Slot id is required for every COMPOUND request, but idempotent non-modifying requests do not really need one
- Security issue for block layout
- Can File, Object and Block layouts co-exist in the same storage network?


EMC²
where information lives®