Scalable Metadata over NFS

Presented at
Ceph Days – Boston
On 6/10/2014

Casey Bodley casey@cohortfs.com

Why NFS?

- IETF standard protocol
- Well established, many implementations
- Client is available on all major platforms
- Extensible protocol can adapt to future needs

Data Scalability

Capacity requirements are growing rapidly

Many techniques for scaling data:

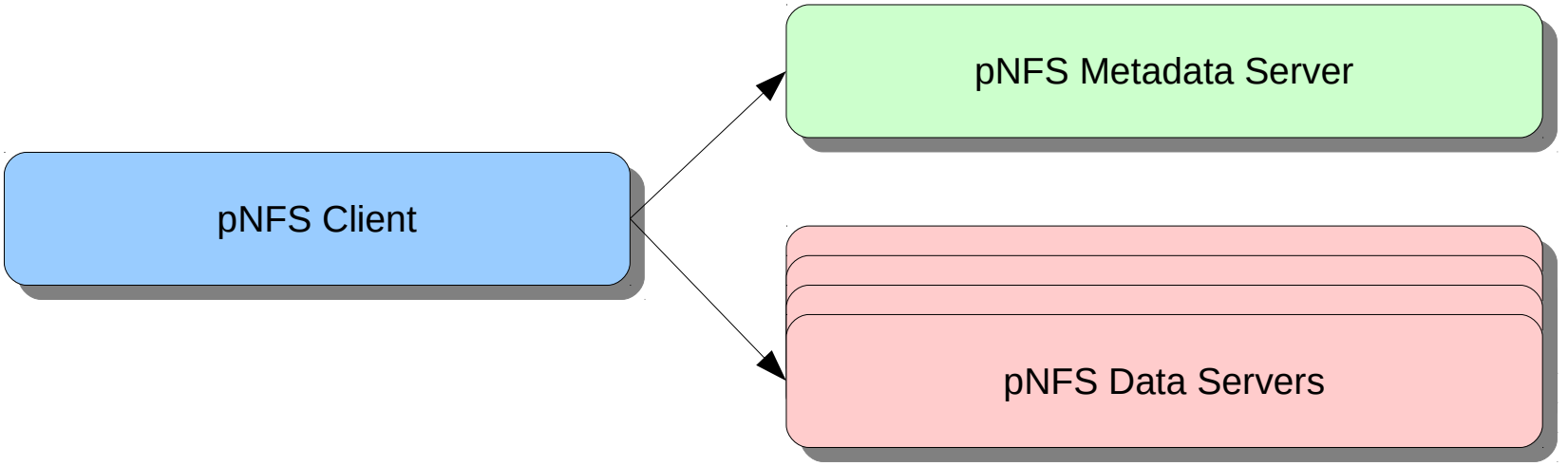
- Raid, object striping
- CRUSH

How can NFS take advantage?

Parallel NFS (pNFS) in NFSv4.1

- pNFS clients speak directly to storage devices
- Support for different storage backends:
  - Block/Volume
  - OSD (T10)
  - File (uses NFS protocol)

pNFS



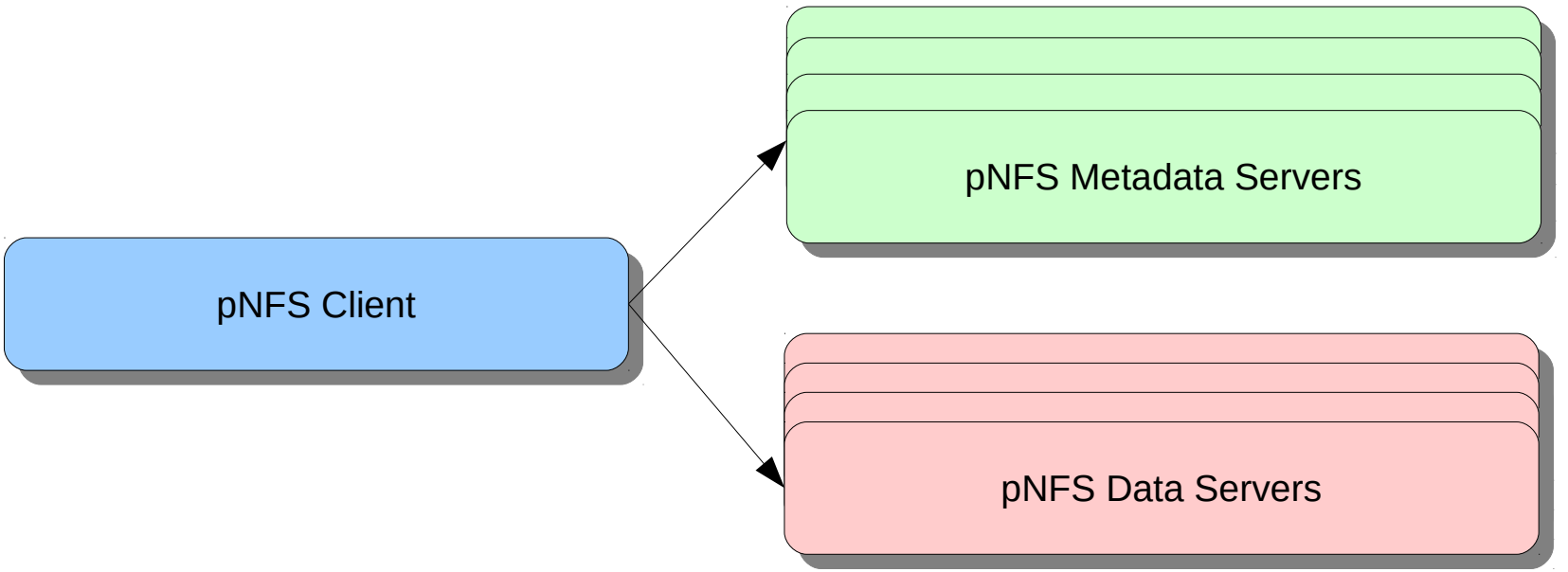
pNFS Layouts

- State that describes data placement for a file
- Acquired with LAYOUTGET
- Released with LAYOUTRETURN
- Recallable with LAYOUTRECALL callback
- Consistency via LAYOUTCOMMIT
- Optional; can always read/write via MDS

Metadata

- Still bottlenecked on one MDS
- Metadata-heavy workloads:
  - Large directories shared by many clients
  - Scientific workloads that create many files, operate in many threads
- Techniques for scaling:
  - Distributing metadata
  - Load balancing
- How can NFS take advantage?

pNFS Metastripe



CohortFS

pNFS Metastripe

- IETF Draft:
  - http://tools.ietf.org/html/draft-mbenjamin-nfsv4-pnfs-metastripe-02
- Two layout types:
  - Filehandle striping
  - Directory striping
- Client mounts initial MDS
- Layouts direct it to other MDS devices

Filehandle Striping Layout

- Single layout covers entire filesystem
- Placement 'hints' as NFS file attribute
- Talk directly to each file's auth MDS
- Not recalled on changes to file placement

Directory Striping Layout

- Describes how entries are placed in a directory
  - Currently supports hashes over simple stripes
  - Next iteration will include Ceph fragtrees
- Fine-grained, layout per directory
- Recalled on changes to placement
- CREATE (mkdir) can give hints for directory size/striping

Metastripe Operations

- Directory listing with PREADDIR
  - pNFS client reads from multiple stripes in parallel
  - Client enforces ordering on entries for telldir/seekdir
- Directory modifications
  - Bulk operations with relaxed mtime consistency
  - Client uses LAYOUTCOMMIT to restore consistency

Metastripe Prototype

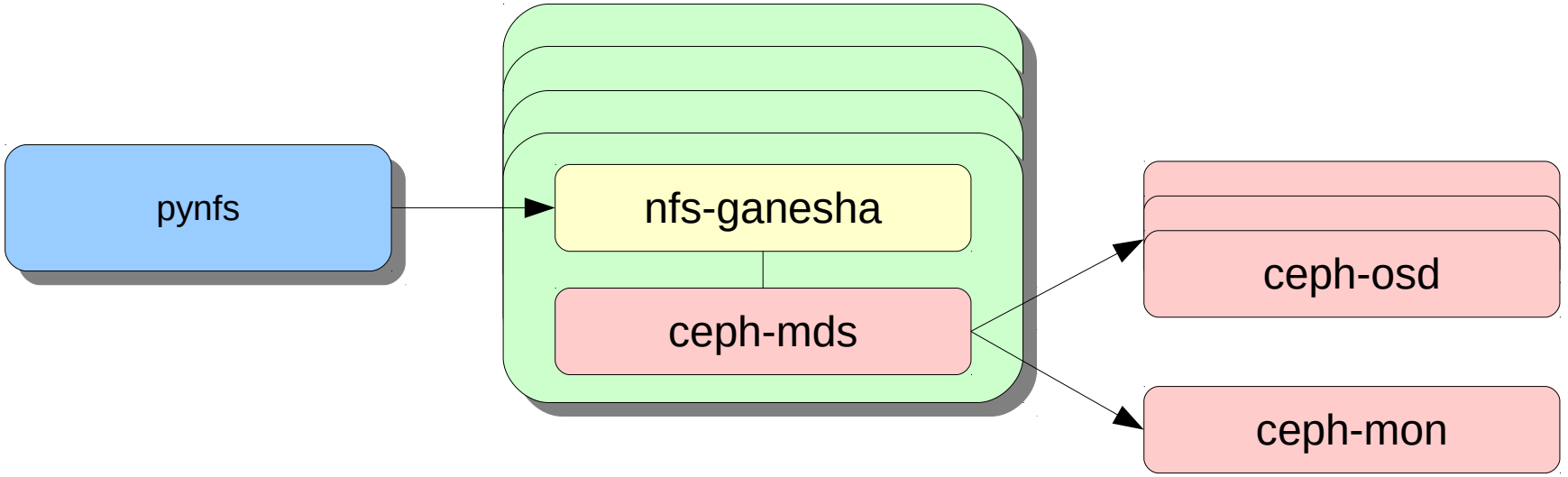
Server: nfs-ganesha

- https://github.com/nfs-ganesha
- Existing support for Ceph

Client: pynfs test suite

- git://git.linux-nfs.org/projects/bfields/pynfs.git

Metastripe Prototype



CohortFS

nfs-ganesha and libcephfs

- libcephfs: added registration for callbacks
  - MDSMap changes
  - Directory placement changes
- nfs-ganesha: serves filehandle and directory layouts based on callbacks

ceph-mds

- Replaced dir fragments with simple stripes
- Added per-stripe locks to avoid serializing on inode locks
- Asynchronous, batched parent stat updates
- Stripe locks use cap updates instead of invalidates

Parent Stats

- Parent stat updates (frag_info_t, nest_info_t) are queued in predirty_journal_parents()
- Sent in batches to their auth MDS
- Provides weak consistency for directory mtime
- Support for LAYOUTCOMMIT incomplete

Client Stripe Caching

- Added client capabilities to Stripe locks
  - Allow client to cache directory entries for each stripe, independently of inode caps
- Added cap updates for Stripe locks
  - Sent synchronously to clients instead of invalidating
  - ex. added 'foo', removed 'bar' since last update

Future Work

- LAYOUTCOMMIT and mtime consistency
- Add support for directory layouts based on Ceph fragtrees in IETF draft

Q/A

Scalable Metadata over NFS

Presented at
Ceph Days – Boston
On 6/10/2014

6/10/2014 Casey Bodley casey@cohortfs.com

Why NFS?

- IETF standard protocol
- Well established, many implementations
- Client is available on all major platforms
- Extensible protocol can adapt to future needs

Data Scalability

Capacity requirements are growing rapidly

Many techniques for scaling data:

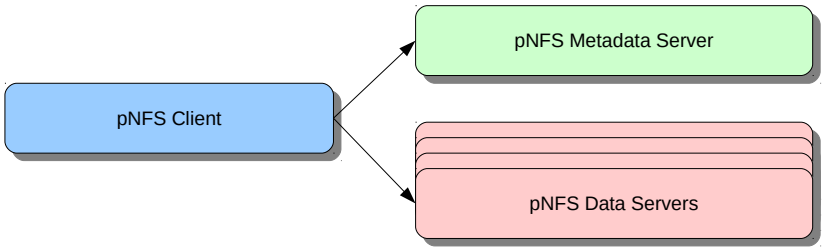
- Raid, object striping
- CRUSH

How can NFS take advantage?

Parallel NFS (pNFS) in NFSv4.1

- pNFS clients speak directly to storage devices
- Support for different storage backends:
  - Block/Volume
  - OSD (T10)
  - File (uses NFS protocol)

pNFS



6/10/2014 Casey Bodley casey@cohortfs.com

CohortFS

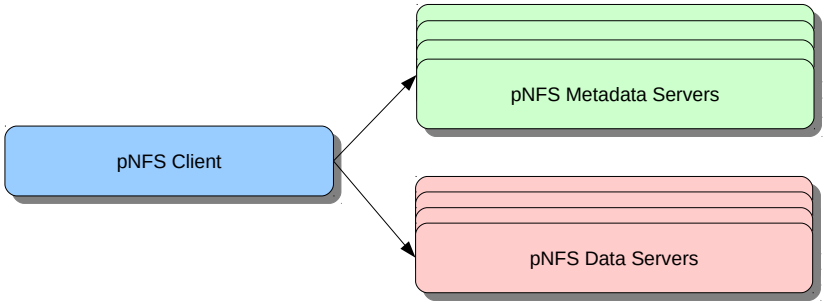
pNFS Layouts

- State that describes data placement for a file
- Acquired with LAYOUTGET
- Released with LAYOUTRETURN
- Recallable with LAYOUTRECALL callback
- Consistency via LAYOUTCOMMIT
- Optional; can always read/write via MDS

Metadata

- Still bottlenecked on one MDS
- Metadata-heavy workloads:
  - Large directories shared by many clients
  - Scientific workloads that create many files, operate in many threads
- Techniques for scaling:
  - Distributing metadata
  - Load balancing
- How can NFS take advantage?

pNFS Metastripe



6/10/2014 Casey Bodley casey@cohortfs.com

CohortFS

pNFS Metastripe

- IETF Draft:
  - http://tools.ietf.org/html/draft-mbenjamin-nfsv4-pnfs-metastripe-02
- Two layout types:
  - Filehandle striping
  - Directory striping
- Client mounts initial MDS
- Layouts direct it to other MDS devices

Filehandle Striping Layout

- Single layout covers entire filesystem
- Placement 'hints' as NFS file attribute
- Talk directly to each file's auth MDS
- Not recalled on changes to file placement

Directory Striping Layout

- Describes how entries are placed in a directory
  - Currently supports hashes over simple stripes
  - Next iteration will include Ceph fragtrees
- Fine-grained, layout per directory
- Recalled on changes to placement
- CREATE (mkdir) can give hints for directory size/striping

Metastripe Operations

- Directory listing with PREADDIR
  - pNFS client reads from multiple stripes in parallel
  - Client enforces ordering on entries for telldir/seekdir
- Directory modifications
  - Bulk operations with relaxed mtime consistency
  - Client uses LAYOUTCOMMIT to restore consistency

Metastripe Prototype

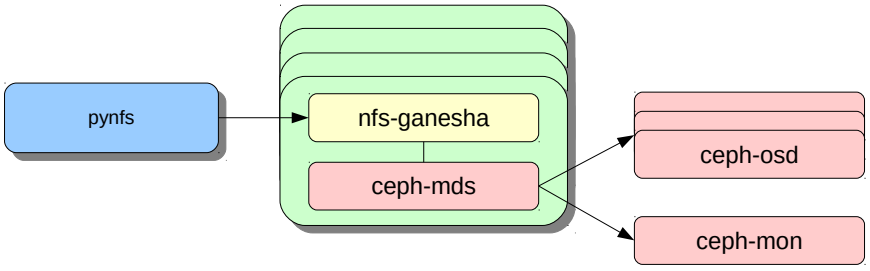
Server: nfs-ganesha

- https://github.com/nfs-ganesha
- Existing support for Ceph

Client: pynfs test suite

- git://git.linux-nfs.org/projects/bfields/pynfs.git

Metastripe Prototype



6/10/2014 Casey Bodley casey@cohortfs.com

CohortFS

nfs-ganesha and libcephfs

- libcephfs: added registration for callbacks
  - MDSMap changes
  - Directory placement changes
- nfs-ganesha: serves filehandle and directory layouts based on callbacks

Casey Bodley casey@cohortfs.com

ceph-mds

- Replaced dir fragments with simple stripes
- Added per-stripe locks to avoid serializing on inode locks
- Asynchronous, batched parent stat updates
- Stripe locks use cap updates instead of invalidates

Parent Stats

- Parent stat updates (frag_info_t, nest_info_t) are queued in predirty_journal_parents()
- Sent in batches to their auth MDS
- Provides weak consistency for directory mtime
- Support for LAYOUTCOMMIT incomplete

Client Stripe Caching

- Added client capabilities to Stripe locks
  - Allow client to cache directory entries for each stripe, independently of inode caps
- Added cap updates for Stripe locks
  - Sent synchronously to clients instead of invalidating
  - ex. added 'foo', removed 'bar' since last update

Future Work

- LAYOUTCOMMIT and mtime consistency
- Add support for directory layouts based on Ceph fragtrees in IETF draft

Q/A