



STORAGE DEVELOPER CONFERENCE

Where The Storage Development Community Connects

2007



pNFS in OpenSolaris

Spencer Shepler

Siddheshwar Mahesh

Sun Microsystems

<http://opensolaris.org/os/project/nfsv41>



Agenda

- Brief recap of NFSv4.1's pNFS
- OpenSolaris
 - NFSv4.0 capabilities
 - NFSv4.1 planned capabilities
 - ZFS recap
 - NFS/RDMA
 - pNFS Design
 - Status

NFSv4.1 (pNFS)

- pNFS servers capable of striping regular files across multiple storage devices
- A pNFS server consists of:
 - A metadata server (MDS) that implements the full NFSv4.1 protocol
 - One or more storage devices
- A pNFS client is an NFSv4.1 client that is prepared to directly access storage devices
- The pNFS client finds out about storage devices from the MDS via a new LAYOUTGET operation

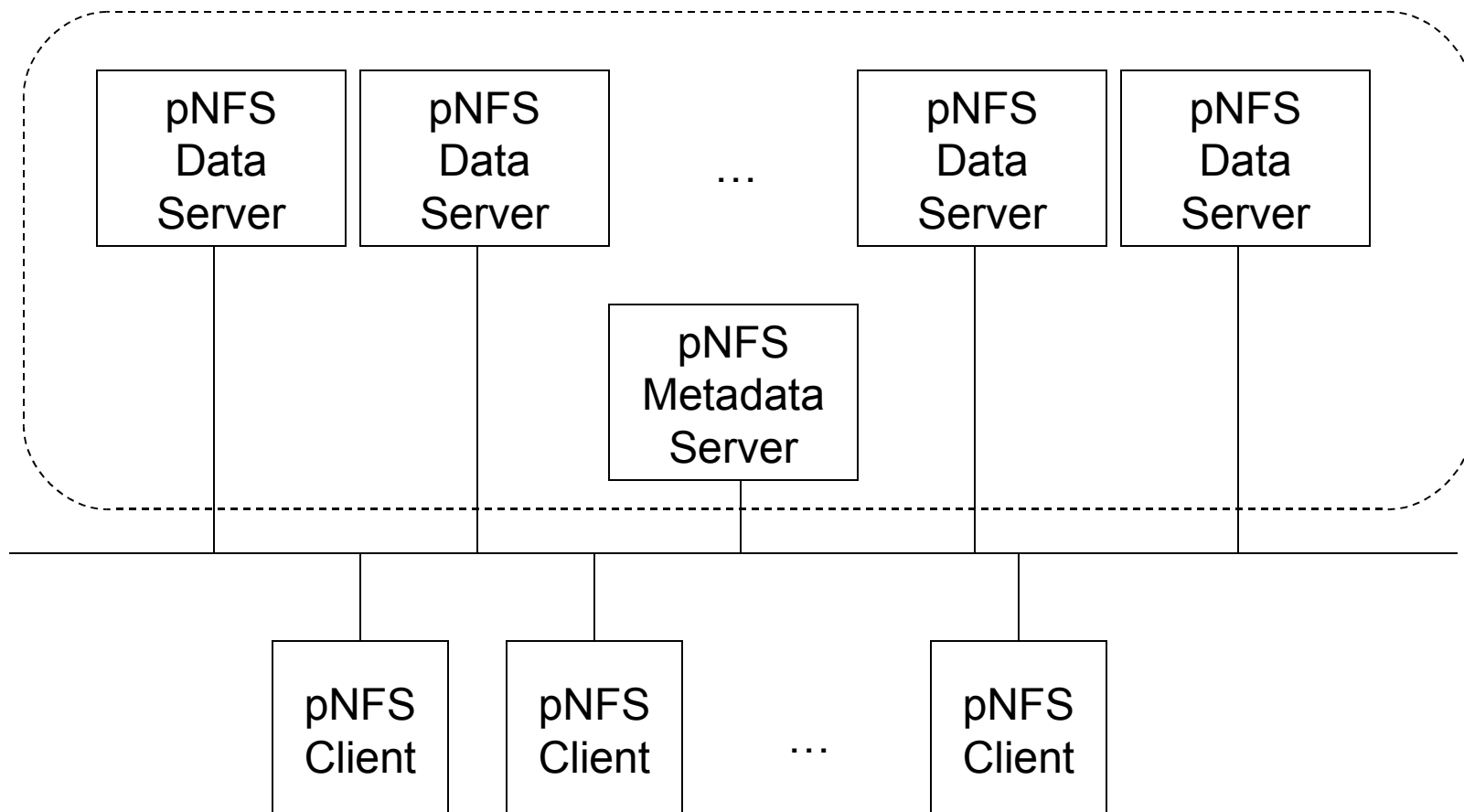
NFSv4.1 pNFS (continued)

- LAYOUTGET returns a layout that describes the striping pattern for a given file
- layouts are recallable which allows pNFS servers to re-stripe a file if desired or necessary
- striping patterns can indicate if a some or all of a pattern has mirrors
 - clients are not required to construct mirrors
 - Thus pNFS offers RAID 0 and RAID 1+0

pNFS Storage Device Types

- pNFS supports multiple Storage Device types (aka layout types)
- A layout can stripe a file over just one type of device
- The NFSv4 working group currently specifies three types:
 - Files [storage “device” is an NFSv4.1 server]
 - Blocks [storage device is an iSCSI or FC target]
 - Objects [storage device is an Object Storage Device (OSD)]
- Additional types require a standards-track specification
- If a client does not support a given device type it can issue I/O directly to MDS

pNFS Server



NFSv4.0 in OpenSolaris

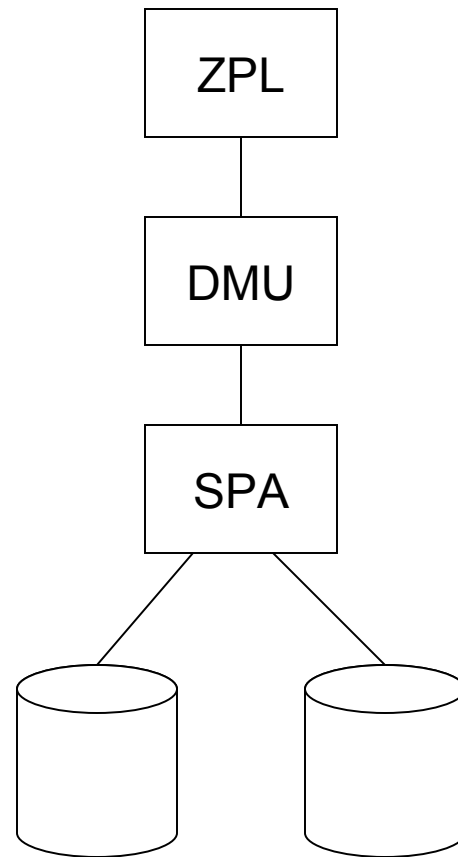
- Kerberos (auth, integrity, privacy)
- Integrated namespace between NFSv3/v4
- File delegations
- Full ACL support with ZFS
- Client “follow mounts” (available soon)

NFSv4.1 in OpenSolaris

- Sessions (integration with Kerberos)
- pNFS files-based data servers
- Meta-data and data servers use ZFS
- NFS/RDMA (Infiniband) support

ZFS

- Pooled storage
- Provable end-to-end data integrity
- Transactional design
- Simple administration





pNFS use of ZFS

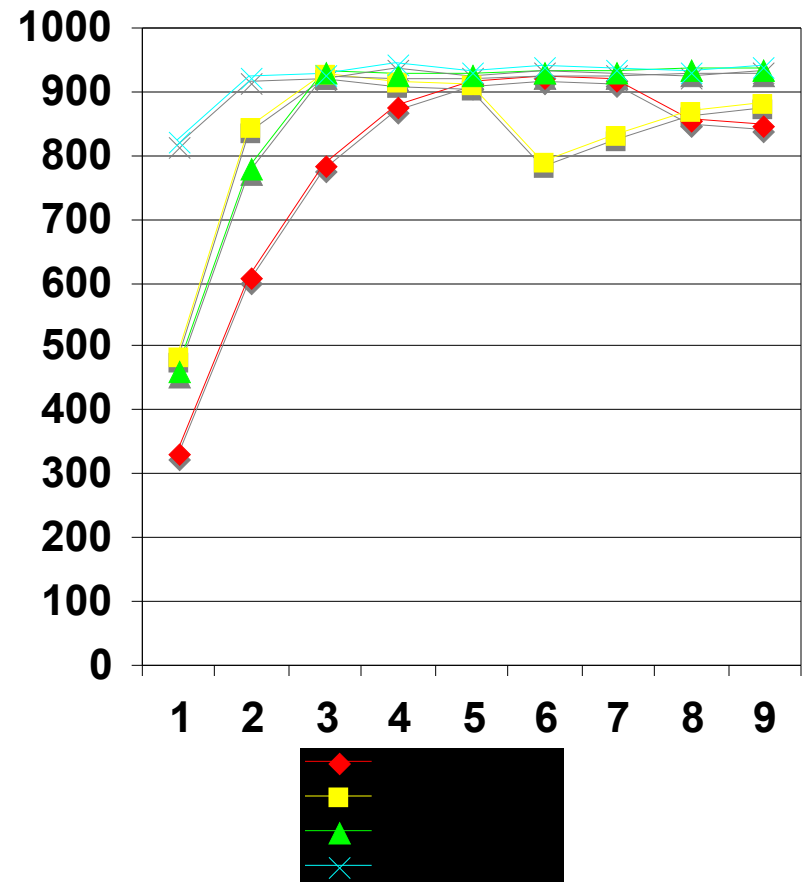
- Meta-data server
 - ZFS (ZPL) for namespace and attribute storage
 - ZFS ACLs for NFSv4 support
 - “system” attributes used for pNFS layouts
- Data server
 - ZFS (DMU) for file object storage

NFS/RDMA

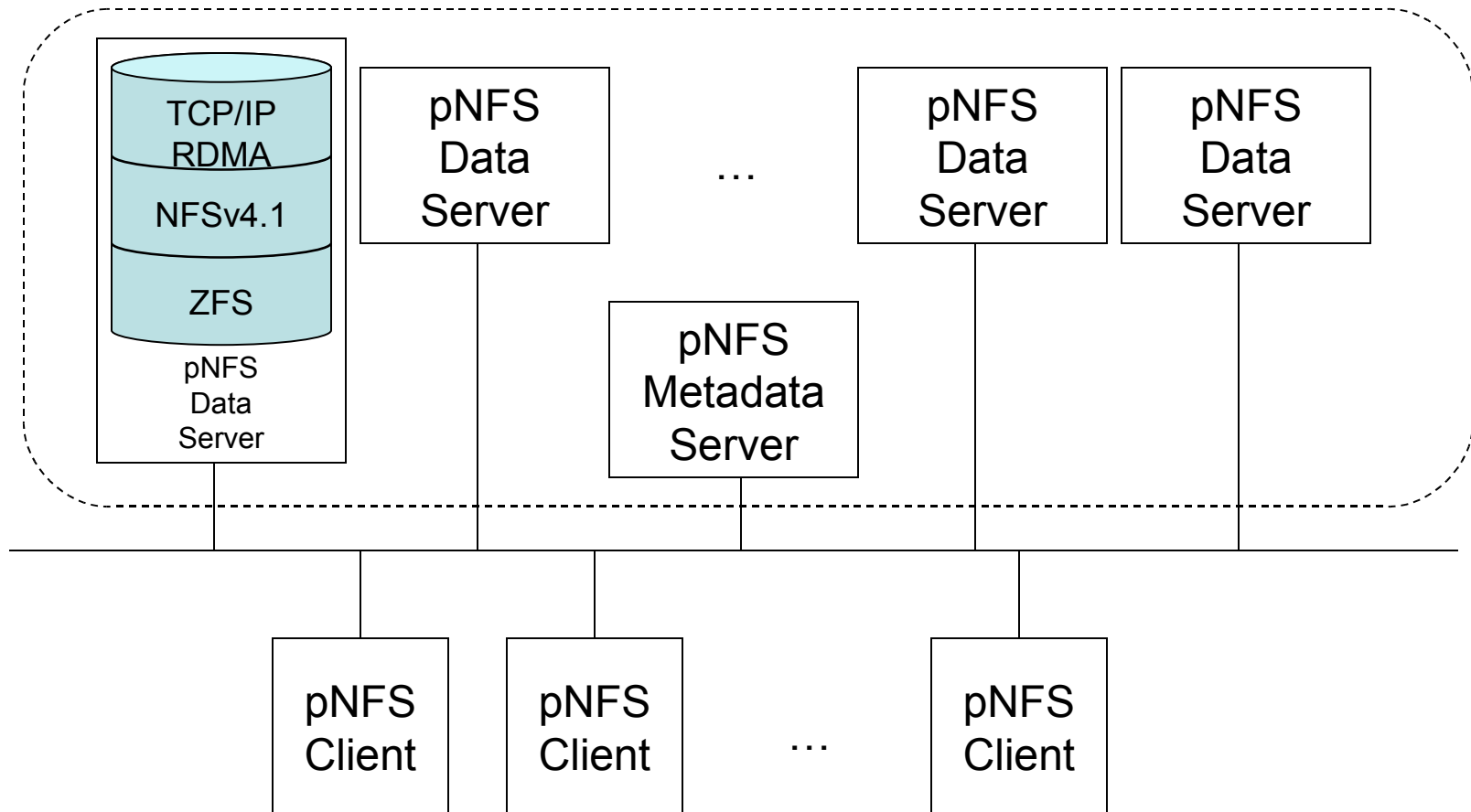
- RDMA enablement (Infiniband)
- Following latest IETF I-Ds
- Expect to deliver into OpenSolaris in 6 weeks

NFS/RDMA performance

- OpenSolaris - OpenSolaris
- Hardware
 - Sun x2200 systems
 - AMD dual dual-core 2.6Ghz
 - 4GB memory
 - Infiniband HCA with PCI-Express
 - Topspin 270 switch
- iozone
 - file size 128MB
 - record size 1MB
 - directio for mount
- X-axis represents number of threads used for benchmark



OpenSolaris pNFS Server



OpenSolaris pNFS control protocol

- Metadata and data server reboot / network partition indication
- Reporting of data server resources
- Inter data server data movement
- Metadata server proxy I/O
- Data server state invalidation

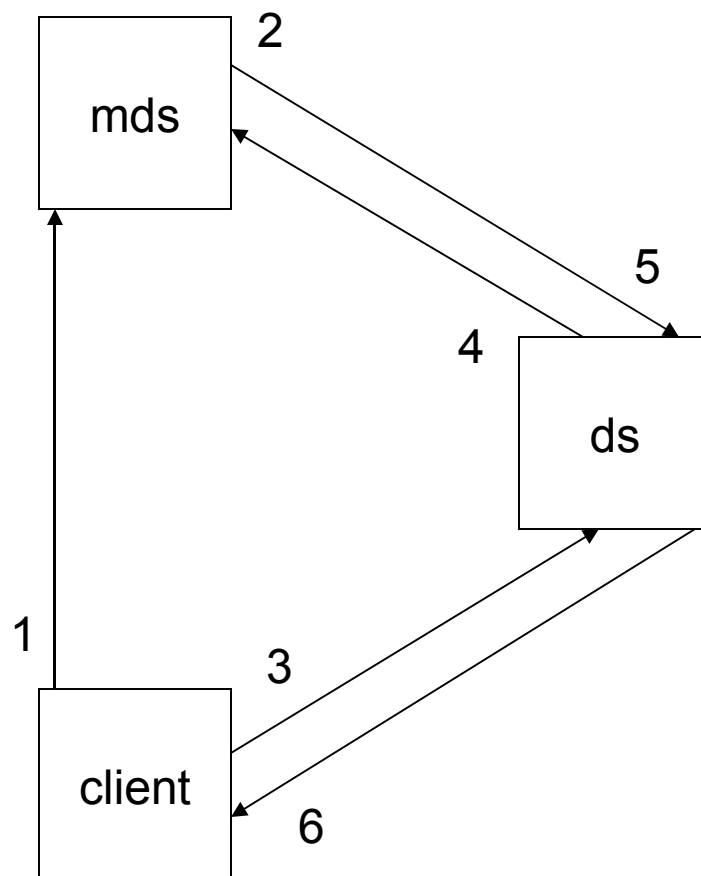
Control Protocol Operations

- DS_EXIBI
- DS_REPORTAVAIL
- DS_CHECKSTATE
- DS_INVALIDATE
- DS_READ
- DS_WRITE
- DS_COMMIT
- DS_GETATTR
- DS_SETATTR
- DS_REMOVE



open and write example

- Client opens file and asks for file layout
- Meta-data server creates layout and responds to open
- Client writes file data
- Data server checks for valid write request (first time only)
- Meta-data server validates and responds
- Data server creates file object (first time), writes data and responds to client





Administration

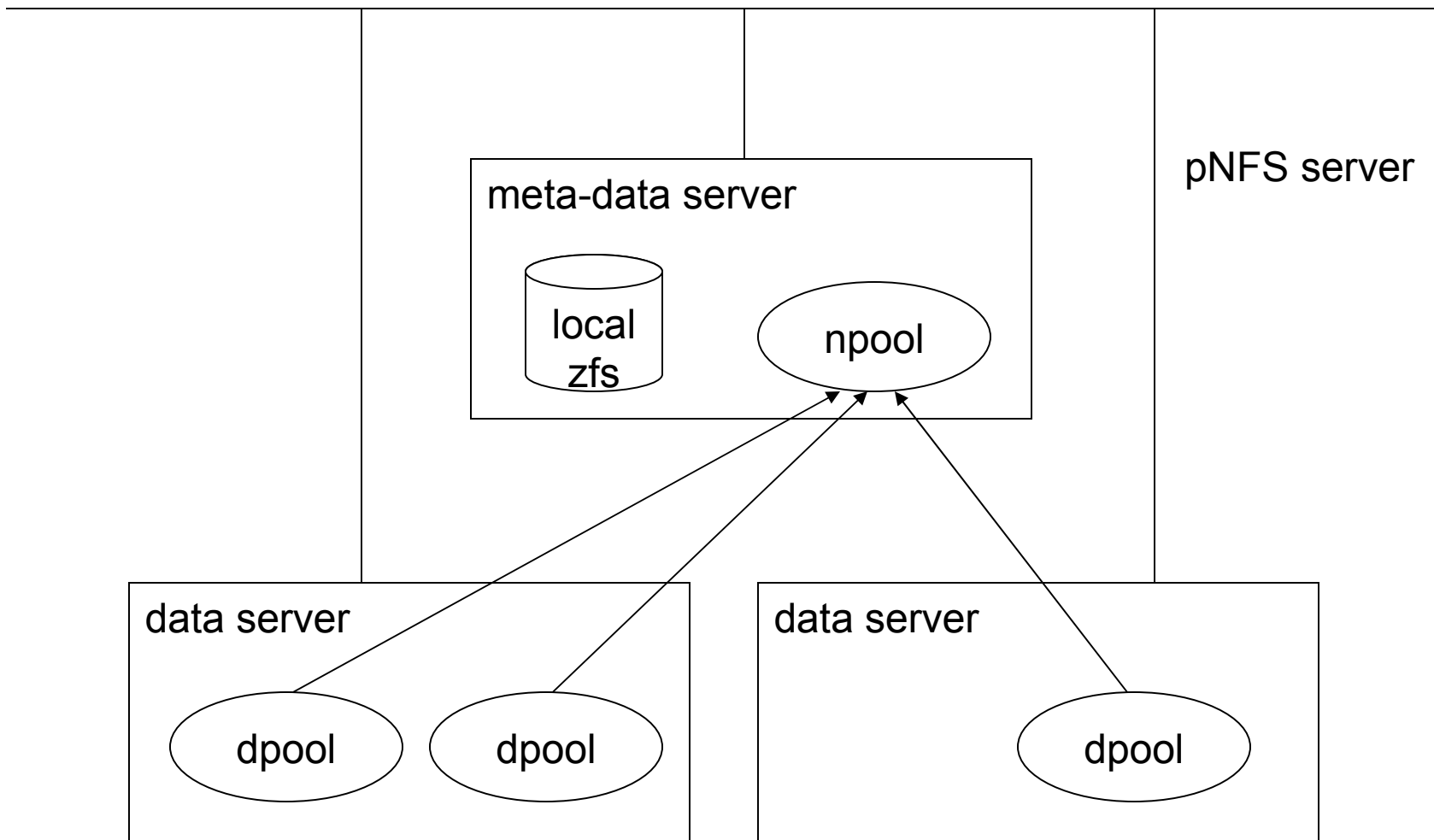
- Commands: pnfs, pnfsalloc, nfsstat
- nfsstat extended for nfsv4.1 and control protocol statistics
- `nfsstat -layout <file>` provided for visibility



STORAGE DEVELOPER CONFERENCE

Where The Storage Development Community Connects

2007





pnfs command

- pnfs ds create
- pnfs ds destroy
- pnfs ds add
- pnfs ds remove
- pnfs ds list
- pnfs online
- pnfs offline
- pnfs ds set <property=value>
- pnfs ds get all
- pnfs ds import
- pnfs ds status



pnfs command (con't)

- pnfs mds create
- pnfs mds destroy
- pnfs mds add
- pnfs mds remove
- pnfs mds list
- pnfs mds online
- pnfs mds offline
- pnfs mds status
- pnfs mds set <property=value>
- pnfs mds get all



pnfs command example

- `pnfs ds create zpool poolDSA c0t0d0 c0t0d1`
- `pnfs ds online poolDSA`
- `pnfs ds set mds=192.168.0.100`
- `pnfs ds create zpool poolDSB c1t0d0 c1t0d1`
- `pnfs ds online poolDSB`
- `pnfs ds set mds=192.160.0.100`
- `pnfs mds online dsa:poolDSA`
- `pnfs mds online dsb:poolDSB`
- `zpool create -o pnfs=on npool raidz2 c0t0d0 c0t0d1 c0t0d2`
- `zfs -o sharenfs=on /npool`



pnfs command example

- Create data server dpool (at DSA)
 - `pnfs ds create -o mds=192.168.0.100 poolDSA zpoolname`
- Create data server dpool (at DSB)
 - `pnfs ds create -o mds=192.168.0.100 poolDSB`
- Create meta-data server npool
 - `pnfs mds online dsa:poolDSA`
 - `pnfs mds online dsb:poolDSB`
 - `zpool create -o pnfs=on npool raidz2 c0t0d0 c0t0d1 c0t0d2`
 - `zfs -o sharenfs=on /npool`



pnfsalloc command

- pnfsalloc add *policy-group filename*
- pnfsalloc replace *policy-group filename*
- pnfsalloc [remove | list] *policy-group*
- pnfsalloc [clear | list]
- pnfsalloc explain *proposed-filename*

Status

- pNFS prototype code available for recent OpenSolaris builds
- Continue to update to follow NFSv4.1 I-D updates and interoperability testing
- Latest downloads (source/binary) (OpenSolaris build 69 / draft-10) available at:
<http://opensolaris.org/os/project/nfsv41/downloads>
- Alias for questions/discussion/contributions:
nfsv41-discuss@opensolaris.org



STORAGE DEVELOPER CONFERENCE

Where The Storage Development Community Connects

2007



pNFS in OpenSolaris

Spencer Shepler (blogs.sun.com/shepler)

Siddheshwar Mahesh

Sun Microsystems

<http://opensolaris.org/os/project/nfsv41>