The Solaris pNFS Data Server

Sam Falkner
Sun Microsystems, Inc.

Agenda

- High Level Design
- Control Protocol
- ZFS
- Administrative Model

High Level Design

- Not a global file system
  - Each data server only has its own data
  - Nodes may be both data server and metadata server if desired
  - Nodes may be in a high availability configuration

Control Protocol

- RPC based
- RDMA desirable, but not required

Control Protocol Goals

- Lazy: don't do anything until needed
- Virtualized: don't tie entities to their current physical instantiation
- Versioned: don't force an "upgrade the whole world at once"

Example Scenario

Register Storage

- Data server calls DS_EXIBI with the metadata server
  - Like EXCHANGE_ID
- Data server calls DS_REPORTAVAIL
  - Report a data set that's available
  - MDS returns MDS_SID (virtual storage identifier)

Client Creates a File

- Client creates a file and does a layoutget
- Data server not involved yet, due to the lazy aspect of the control protocol

Client Writes to Data Server

- Data server does not find (filehandle, stateid, credential) tuple in its cache
  - Data server calls DS_CHECKSTATE to validate
- Data server extracts an object id from the filehandle
  - But this is the metadata server's object id

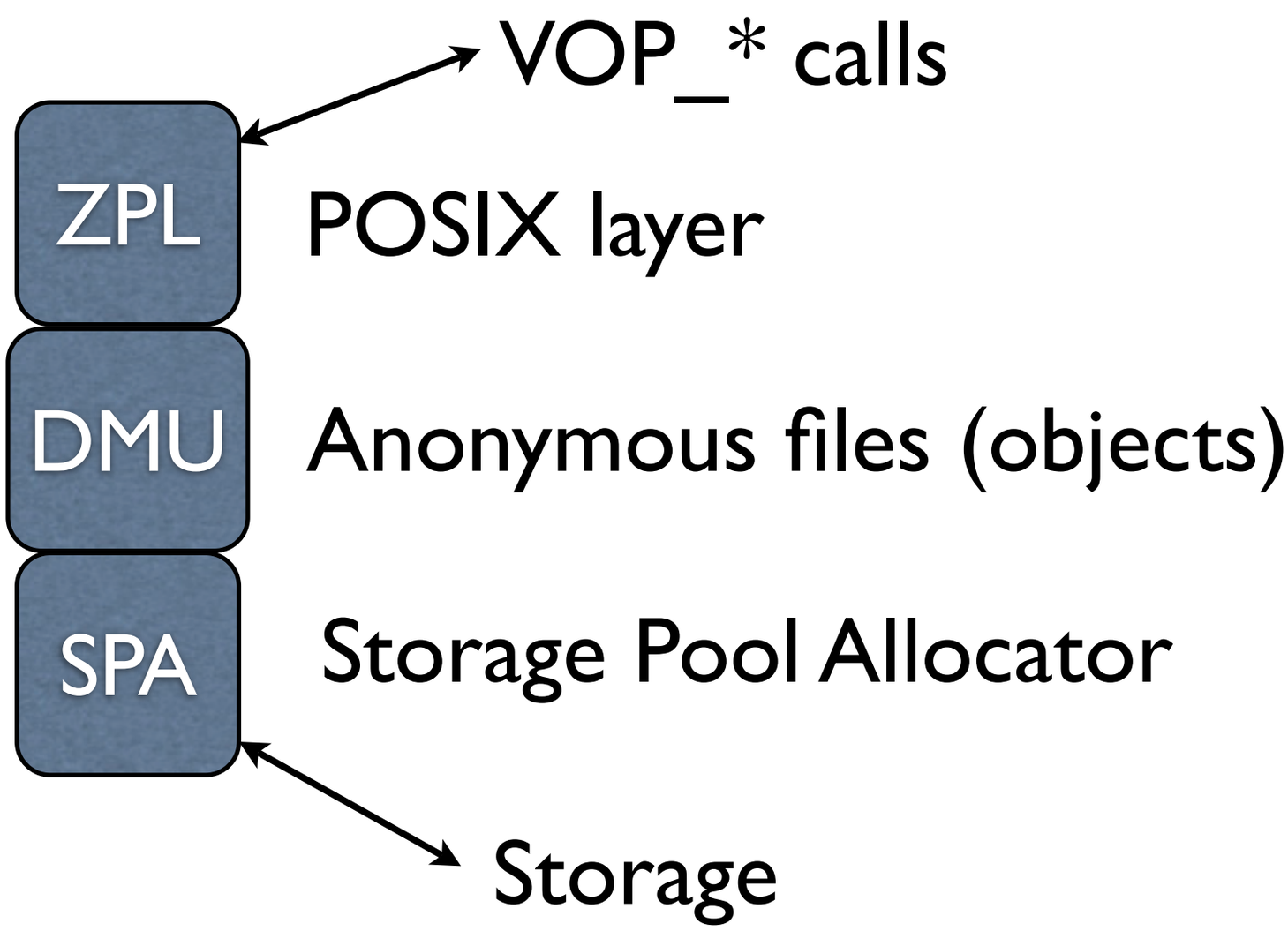
Data Server Writes

- Data server discovers that there is no object corresponding to this object id
- Data server creates new object, and stores the mapping of the metadata server object id to the physical object
- Write proceeds...

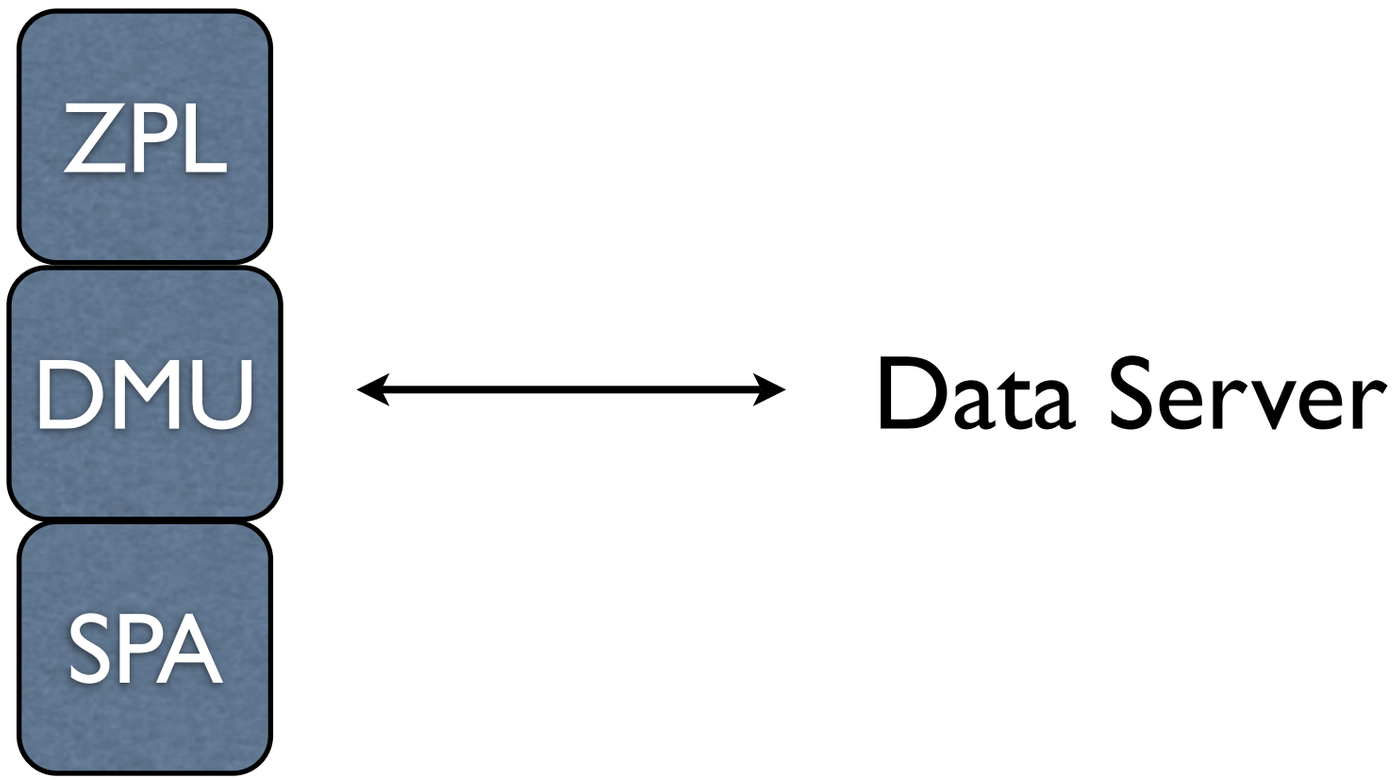
ZFS

- Initial implementation will be deployed on ZFS
- Other backing stores possible, of course

ZFS Architecture (simplified)



The Data Server and ZFS



DMU-based Data Server

- DMU provides anonymous files
- Not in the namespace
- No ACLs, modes, owners, groups, ...
- All other ZFS features are still there
  - snapshots, checksums, quotas, ...

Administrative Model

- Create a zpool
- Create a dataset of type “pnfsdata”
  - specify the metadata server

Example

```
# zpool create tank mirror c0t0d0 c0t1d0
# zfs create -t pnfsdata \
    -o mds=mymds,sharepnfs=on tank/pnfs
# zfs create tank/normalfs (If you wish...)
```

Questions?