

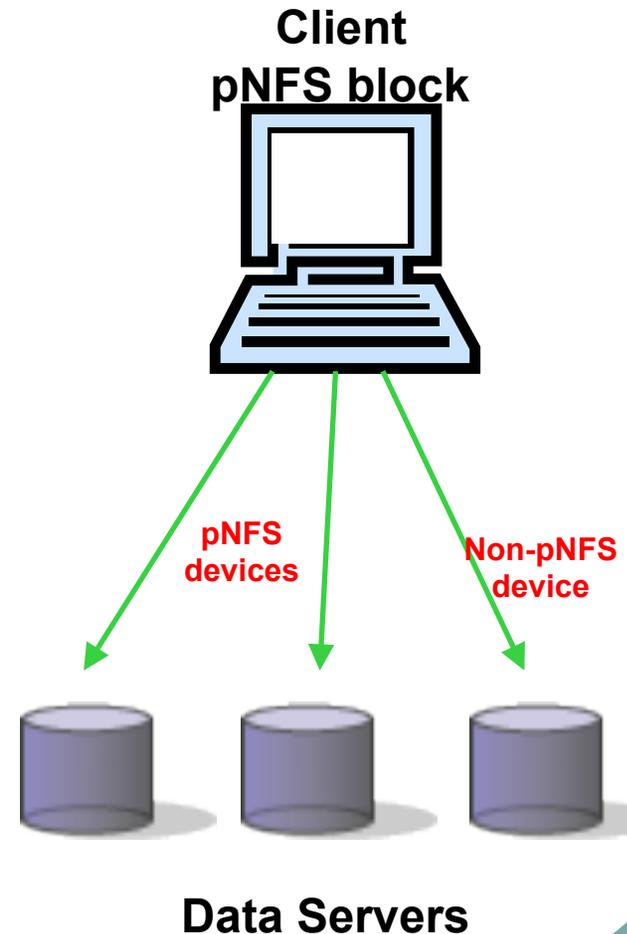
pNFS block signature/disk protection

IETF 79 NFSv4 WG Meeting, November 9, 2010

Sorin Faibish – EMC (sfaibish@emc.com)
Jason Glasgow - Google

Problem Statement

- Client cannot differentiate pNFS devices from non-pNFS devices
- pNFS devices are discovered after mount time via GETDEVICEINFO
- Before pNFS mount kernel/apps may write to pNFS SCSI devices and destroy pNFS FS
- No way to protect pNFS devices when there are multiple paths to same device
- There is no protocol way (5663) to identify devices used by pNFS
- Problem observed when complex volumes support was implemented



Additional Issues

- Current block protocol signature defines deviceid but no pNFS specific name
- Location of signature only communicated to client on GETDEVICEINFO call at mount time
- Even if there was a pNFS specific signature it cannot be found before mount; client doesn't know the MDS IP
- Signature can be vendor specific but client doesn't know it relates to pNFS
- Even if location is known, there is no write protection mechanism for pNFS devices

Possible Solution

- Need to extended RFC5663
 - Define fix location/s (configuration)
 - Define pNFS signature: “pnfs block device”
 - Recommend protection mechanism
- Will have to be vendor independent (generic name)
- Extend signature to indicate that a disk is used by pNFS
- Use offset (from beginning or end of disk), a length, and an array of bytes same as deviceid
- Possible use multiple matches to uniquely identify a pNFS device

Different Approaches

1. Clients can have configuration file that specifies the signatures to protect.
2. Clients could come pre-configured with a protection file that recognizes "most signatures"
3. pNFS servers could define a function that will return such signatures. Every time a client does a mount, the client could retrieve the signatures (will be maybe too late) , and match the configuration file.

WG Asks and Q&A

- **WG Asks:**
 - Extension to block or 4.2 feature
 - Can use a configuration file
 - Can use fix location
 - Can add pre-mount call