pNFS Lustre layout draft 05 and beyond

IETF 87, nfsv4 WG – Berlin, July 31, 2013

Sorin Faibish - EMC
Peng Tao - EMC

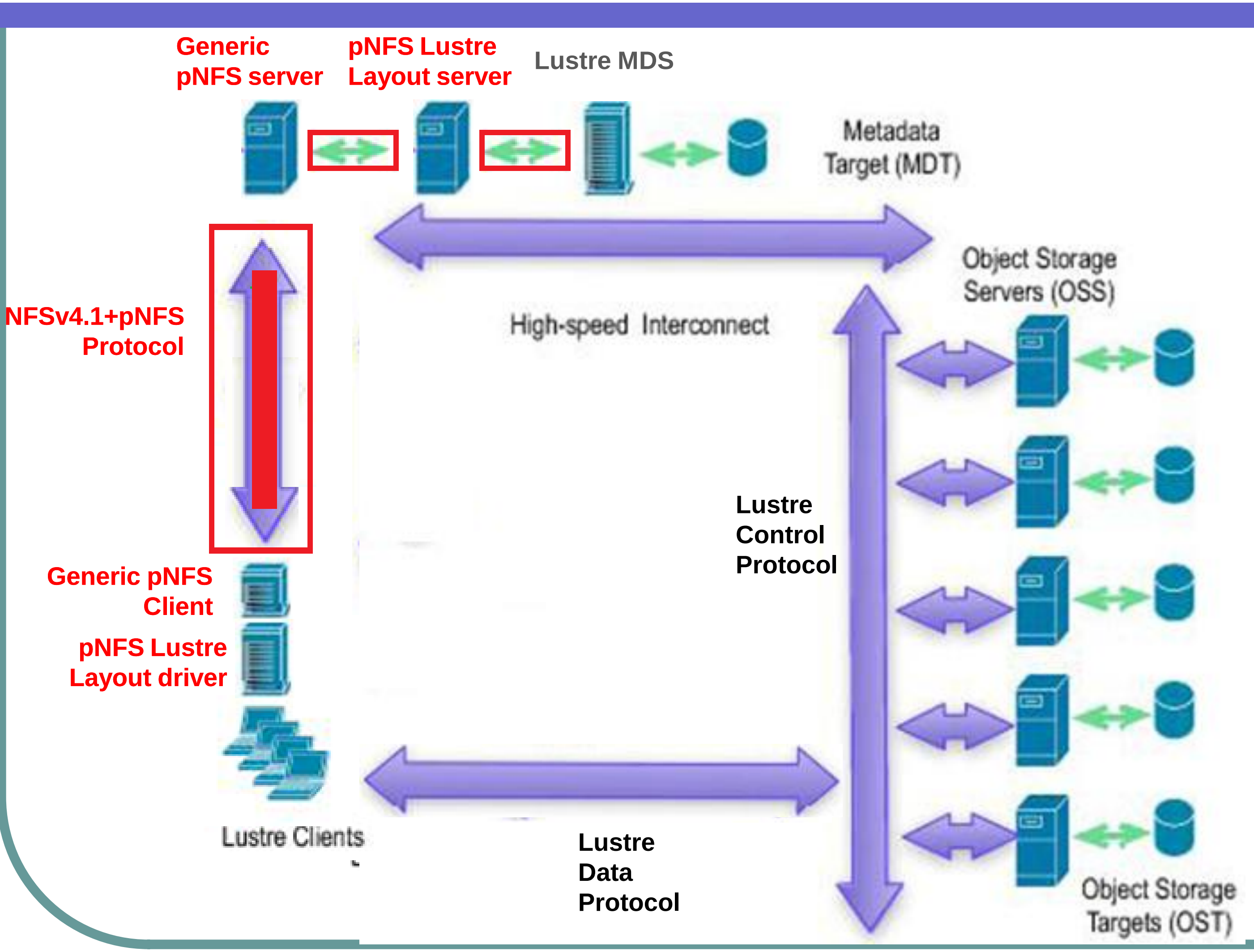
Agenda

- Update to existing Lustre layout draft-05
- New idea for LNET- based layout

Current Lustre Layout draft

- Revision 05 changes
  - Added references to LDLM, LOV and LND
  - Added a diagram and detailed explanation of Lustre layout being a shim layer on top of Lustre client and server (next slide)
  - New updated reference to documents on lustre.org
  - Added reference to Lustre source code in Whamcloud git tree and kernel git tree

pNFS Lustre client/server additions



Draft based on Lustre 2.4

- Wrapper Lustre MD inside pNFS MD
- Use Lustre as data protocol between Client and OSS
- Lustre client is into Linux kernel staging area updated for 3.11
- Draft based on current Linux kernel client implementation
- Draft updated to 2.4 Lustre client

Implementation direction

- Shim layer on server to translate from Lustre layout and pNFS MD
- Shim layer on client to translate pNFS MD to Lustre layout

Next draft changes (new draft?)

- Remove unnecessary Lustre client components
  - ldlm (Lustre layout uses NFS lockd)
  - mdc/mgc (Lustre layout uses pNFS MD to transfer layout information between client and server)
- Facilitate and simplify implementation on other OS than Linux
  - BSD, Solaris

New draft

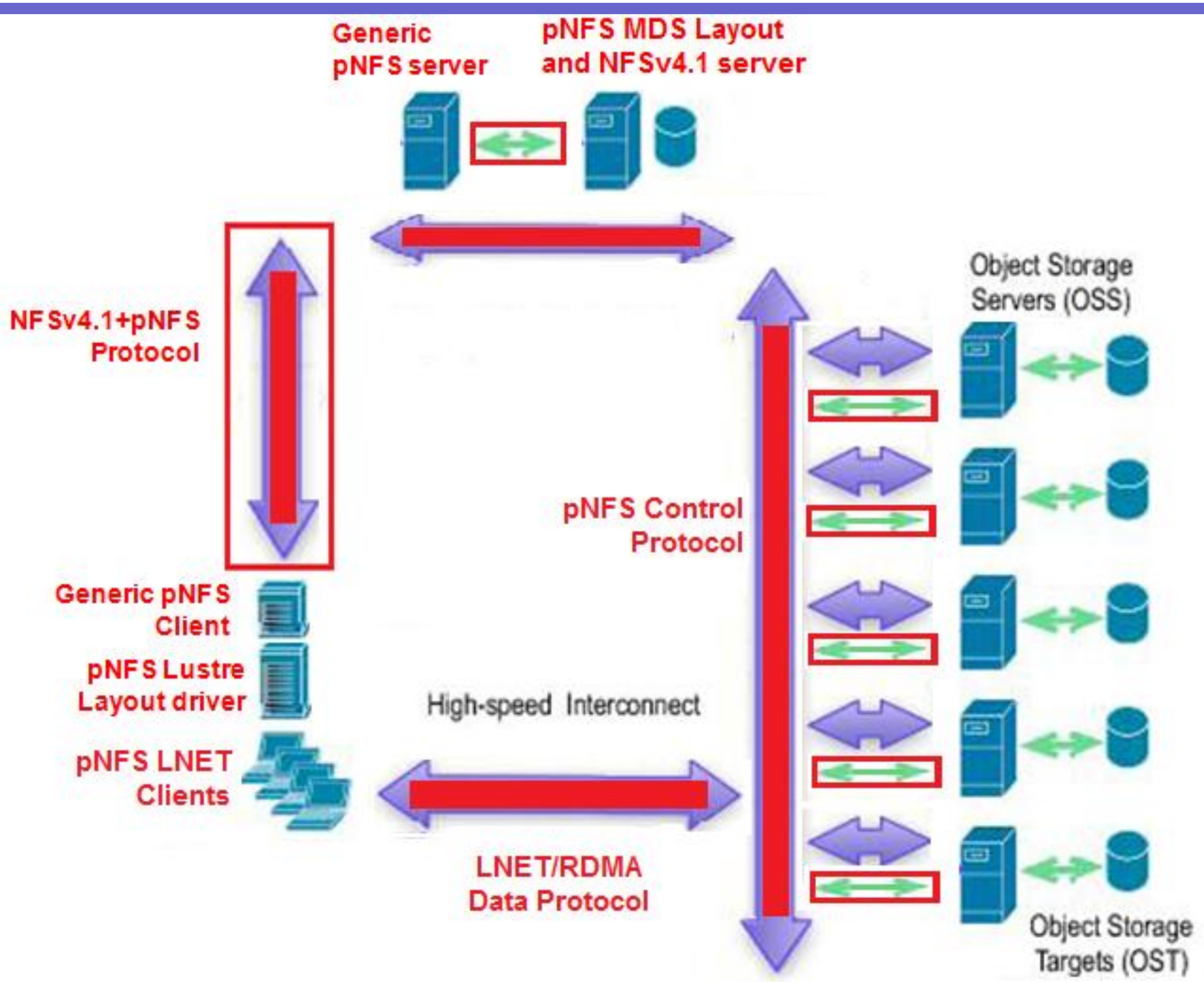
New idea for LNET- based layout

Alternative Layout summary

- Remove the entire Lustre client/server layout and replace with pNFS client/server layout
- Use only Lustre LNET data protocol and replace layout and control protocol with pNFS object protocol (maybe)

Alternative Layout features

- Define simple layout information in Lustre MD
- Do not wrap Lustre MD inside but use pNFS attributes
- Still use Lustre transport protocol as data protocol: TCP and RDMA


Generic
pNFS server
pNFS MDS Layout
and NFSv4.1 server
NFSv4.1+pNFS
Protocol
Object Storage
Servers (OSS)
pNFS Control
Protocol
Generic pNFS
Client
pNFS Lustre
Layout driver
High-speed Interconnect
pNFS LNET
Clients
LNET/RDMA
Data Protocol
Object Storage
Targets (OST)

Alternative Layout value

- Define and use low level Lustre client interface to talk to Lustre OSS
  - Use only Lustre data servers and data protocol
  - Use new control protocol from pNFS MDS to OSS
- Take advantage of fast scalable LNET RDMA implementation.

Alternative Layout client

- On client side, pNFS layout information are translated into Lustre stripe information.
- MD wise, simpler than current Lustre layout, thinner than current draft on client side.
  - Use pNFS MDS not shim on Lustre MDS
  - Will support NFSv3 fallback for NFS clients access as well as non-pNFS Lustre clients

Alternative Layout server

- On data server side, just like current layout, shim layer to translate between pNFS MD and Lustre layout.
- pNFS MDS <-> OSS control protocol will be based on Lustre protocol but needs to be changed to pNFS ACL and security model.
- Fencing will be done by OSS adding a thin clustering module

Future direction

- Build a very high performing and scalable pNFS layout
  - Address current pNFS RDMA performance and scalability limitations
  - Use a thinner layered data protocol based on RDMA verbs: Draft-hilland-RDDP-verbs-00.txt defines an abstract interface to a RDMA enabled NIC(RNIC).
  - Will require a new RDMA verbs draft replacing the abandoned draft implemented as a combination of the RNIC it's associated firmware and host SW.

Discussion

- Next steps:
  - Write new draft pNFS lnet and/or verbs(if/when available?) layout
  - Restart/update/re-write verbs draft outside nfsv4 WG – volunteers welcome
    - Support of the WG is needed
  - Will depend on draft of RDMA verbs if we go on this path – advice
- Q&A