**OFI WG Data Storage / Data Access Subteam Weekly telecom – 04/14/2015**

**OFIWG Download Site:** [www.openfabrics.org](http://www.openfabrics.org) 🡪OFED/OFA Resources 🡪 OpenFabrics Interfaces WG

**Agenda**

* roll call, agenda bashing
* Storage requirements for Lustre – Doug Oucharek
* Continue review of key objects and methods – Stan Smith

**Next steps on SNIA stuff**

- first order of business is to track the completions described by Bernard

- would really like the persistent level guarantee

- some additional ancillary requirements, but most already probably met

- SNIA team is currently wrapping up a white paper on e.g. error handling

- To participate in SNIA TWG requires joining SNIA and signing an IP agreement that requires that you will disclose any IP that goes into the spec.

**Lustre parallel filesystems – Doug Oucharek**

**See Doug’s slides, 2015\_OFI\_WG\_Lustre.pptx on the download site**

[**http://downloads.openfabrics.org/downloads/ofiwg/dsda\_rqmts/**](http://downloads.openfabrics.org/downloads/ofiwg/dsda_rqmts/)

Quick review of materials from OFA workshop

-10,000 – 100,000 clients (or more)

- all should have equal access to the filesystem

- if a client locks a file for too long, the server will evict the client. This could cause the death of the job in question. An eviction is often the first way the filesystem knows that there’s a problem.

- evictions are bad.

Usage models

- In the big data space, there is an available Hadoop adapter for using Lustre.

- In the cloud, Lustre appliances exist today as a series of VMs allowing HPC clusters in the cloud.

- Currently this is Ethernet based, but the cloud community may be interested in moving to IB.

Client/server view

- Connectivity – a client will create an RC connection to each server comprising the filesystem. From a network perspective, the client connects to the OSS, but from a filesystem perspective, the client is actually communicating with the OST, using the network service provided by the OSS.

- Today, when using Enet, it’s straight TCP/IP. For RDMA over Enet, RoCE seems to work with Lustre, as does iWARP.

- A significant issue in the Lustre space is the use of pings to determine connection health. One possible help would be a high priority health message. Lustre is looking at building a health network using the Gossip protocol.

- Concerned that RC connections won’t scale far enough – reliable datagrams?

Flow control

- Currently self imposed – client and server manage credits. Adding queues adds latency which is okay for Lustre, but dropped messages are not. Dropped messages cause evictions, which are bad.

Routing

- Really need a single OFED build that works equally well regardless of the verbs provider, e.g. prefer OFED instead of MOFED.

- Routing using LNET over TCP/IP uses the same route selection protocols as LNET over IB – LNET looks at a TCP socket connection the same way as an IB QP connection.

Network Stack

- Everything in Lustre is in kernel space – not using user level libraries. Getting things into the kernel is sometimes a challenge.

- Lustre client / server exchange RPC messages.

- Next layer down is the LNET layer – buffering, flow control, etc

- Next layer down is the LND layer – today its socklnd, o2iblnd, gnilnd. You tell LNET what type of network layer you are talking to, and it loads the correct module (socklnd or o2iblnd or Gemini).

- LNET is fully asynchronous.

- To add OFI, we would want a new kofilnd layer. Lustre community will have to communicate to its clients which lnd layer to use, and why. Performance? Flexibility? Scalability?

Requirement Summary

- Equal core access to network interface – deal with explosion of cores.

- Encryption support – hooks or callbacks into the OFI stack would be useful.

- Latency support 🡪 FMR or equiv. FMRs are handy with high latency connections to keep the pipe full.

- VM support. Because Lustre is running in the cloud.

- SLA with fabric. Allow the fabric to throw away packets if the SLAs are exceeded.

- Single OFED stack optimized for all supported interfaces, preferably upstream (kernel.org)

- Fault/error counters from the fabric. A callback when error counters overflow would be helpful to allow the router to choose more intelligently between available paths.

- High priority messaging e.g. for a health network

- Currently only do ‘put’ operations using RDMA writes. This is because RDMA reads have a resource limitation associated with them. In some cases an RDMA write is sub-optimal. It would be optimal if there were no limitations, but also good if the limitations on read resources were observable, or even better controllable. o2iblnd today supports only puts, ‘gets’ don’t work.

- reliable data grams.

**Agenda for next meeting**

* Continued discussion on KOFI data transfers (Stan)

**Next regular telecom**

Next meeting: Tuesday, 4/21/15.

8am-9am Pacific daylight time

**NOTE:** We have switched over to using Webex (courtesy of Cisco). The URL for joining meetings is:

<https://cisco.webex.com/cisco/j.php?J=200935598&PW=67935ad6df07030d5f05044a5b0f>