**OFI WG Data Storage / Data Access Subteam Weekly telecom – 02/17/2015**

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

**Agenda**

* roll call
* Discuss Open Intel Questions regarding the Requirements Document
* Stan reviewed his proposed fi\_getinfo API
* Participation from the Lustre (LNET) community – Chet has contacted the Lustre team inside Intel. No news yet as to when they will attend.

**Open Intel Requirements questions**

* Bernard agreed that some of the requirements should be prioritized in to “must have” or “nice to have” including:
	+ Work Request Execution Ordering section
	+ Buffer Completion Level section
* Requirements for data visibility versus data persistency. Consider adding to the descriptions to make it clear which is driving the requirement including:
	+ Work Request Execution Ordering section
	+ Buffer Completion Level section
* The group helped Intel come up with some specific Use Cases to cover some the requirements. Do we want to add Use Cases to the requirement or perhaps just add some text for these:
	+ RDMA Atomics – Shared Locks Use Case
	+ Fusing Section 1.6 – Compare and swap of remote memory Use Case. Only write the data if the read succeeded.
	+ Mixed SGLs Section 7.2 – Networking SW Use Case - Pushing pre-built static network packet header from DRAM followed by a data payload from somewhere else.
* Detailed fi\_getinfo( ) API review led by Stan
	+ General KOFI Architecture Notes
		- Need to consider kernel versus user application API. While we are starting with the application interface, we will need to tailor APIs to the kernel.
		- Today Linux drivers don’t usually issue fabric or domain level calls. PCI add one/remove one is a lower interface. The top edge of that provider will top to the KOFI framework module and the top of that will expose the KOFI under discussion.
		- KOFI is similar to the application verbs library and supports multiple interfaces and HW.
		- Newer devices may not have anything to do with IB so the IB verbs layer is problematic for those types. Make KOFI more message passing specific and less device specific.
		- Most difficult part is getting the info and understanding the fields of interest when parsing the data
	+ fi\_getinfo( ) API specific notes:
		- Not all providers fill in all fields in the get info – need to add more information on what fields will be required to support for each type of provider
		- Reviewed Reliable Datagram Sockets (RDS) example of Get Info
			* We need to add IB example
			* Sample utilizes ipv4 addresses which in turn will utilize an IB RC Reliable Connection type from node to node
		- For an IB provider, the domain is the PD. For another device the domain could be something different, socket domain would be similar to the HCA – Ethernet NIC, each PCI card would be a separate domain
		- This API does not issue a bind as part of the call and left to the provider to handle
			* We need to up-level the interface to make it clear what the provider is doing for you.
			* The provider layer needs to be as thin as possible.
			* Provider does bin, builds WQ request, and sends the request – IB specifics are hidden from the application. Info is saved behind the scenes and that info is passed through once you define an end point, binding, etc…
		- Data transmission attributes are calls and are basically the same no matter what device type you are talking to.
			* Need to up-level the interface.

**Participation from the Lustre (LNET) community**

- Doug Oucharek has agreed to join the group.

**Meeting Coverage for 2/17 and 2/24 meetings**

- Chet Douglas and Frank Yang have graciously agreed to chair these two meetings (Chet: 2/17, Frank 2/24).

**Agenda for next meeting**

- Continue Review of FI\_GETINFO (Stan)

**Next regular telecom**

Next meeting: Tuesday, 2/24/15.

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>