**OFI Data Storage / Data Access Subteam Weekly telecom – 02/21/2017**

**DS/DA Shared Documents:** <http://downloads.openfabrics.org/WorkGroups/ofiwg/>

**Agenda**

* roll call, agenda bashing
* enhancements to kfabric

**Very light attendance, so the meeting was foreshortened substantially. No recording was made.**

**Minutes from last meeting (2/7/17) attached below as a reminder.**

**Possible Enhancements to kfabric, driven by the Lustre discussion on 1/21/17**

* Memory Registration. One of the most complex parts of verbs, and getting more complex as time goes on.
	+ Memory management becoming increasingly important, and one area that is the source of many errors.
	+ Per Stan, much of the complexity of memory registration has been pushed down into the provider. In a nutshell, buffer, length and few other things, and you get back a single key representing those regions.
	+ That being said, it appears that kfabrics already addresses much of the memory registration complexities.
	+ An example was offered of changes in the interfaces due to memory registration in the RDMA subsystem. These changes inject confusion that creates problems for e.g. the Lustre community. Would very much like a stable, managed API.
	+ This exposed a long standing issue with Linux – ABIs are not standardized and are subject to change at any time. So unless your device driver is upstream, you are subject to being broken at any time.
	+ Another example: Global memory regions recently, and unexpectedly, disappeared. All of this stuff should be hidden underneath the covers; changes
	+ Because kfabric is not tied to any particular provider, there is the potential that it could serve to stabilize the API from the perspective of the consumer.
	+ The point – By looking at the problems that consumers, e.g. Lustre are having with the verbs model, kind of implies a requirement for a better way of doing things via e.g. kfabrics. Literally spending months solving problems created by the Linux kernel people.
	+ The existing RDMA subsystem really is a ‘red-headed stepchild’ within the kernel – no documentation, interfaces are constantly changing, not coded in a way that is generic and secure, things change on a whim by adding things on as a wart.
* Whatever became of the idea of a ‘killer app’ that would justify pushing kfabric upstream?
	+ The discussion on 1/21 was motivated by the idea to just develop kfabric for e.g. Lustre, and see if that produces enough pull to eventually cause it go upstream.
	+ This still seems like the right way to go, but no concrete plan has emerged to drive it.
* Is kfabric at the correct level of abstraction for I/O? Or does the fact that it essentially mimics libfabric take us down a path of greater complexity than is needed for an API for I/O?
	+ Existing LNET/LND i/f is fairly basic, with simple send/receive and error information with much of the complexity buried in the LND. Connections are created on demand based on e.g. a profile created at the beginning, but the connection not actually created until someone uses it.
* CPU/NUMA affinity – increasingly important for Lustre in an era of many cores/sockets.
	+ Would like hints from the consumer to indicate the CPU which is hosting the memory region being accessed. The hint would be included in kfabric with the objective of passing hints to the provider.
* Question from Ira – why can’t some of this be done with verbs today?
	+ Ans: it can be, but in the end you wind up re-inventing the kfabric interfaces but with a different name.
	+ Assertion – current verbs-based data structures need to be changed. Expect to see some OPA patches.
* Workshop (again)
	+ Opportunities for improving kfabric,
	+ Useful for Doug to describe some of the difficulties Lustre has encountered w/ verbs,
	+ What would this change mean for Lustre
	+ E.g. the difficulties that Lustre encountered with memory registration and how kfabric would help resolve those issues.

**Webex Recording:**

**(no recording was made of this meeting.)**

**Next regular telecom:**

Next meeting: Tuesday, 03/07/17

8am-9am Pacific daylight time

**Logistics:**

See the OFA’s central calendar (<https://openfabrics.org/index.php/ofa-calendar.html>) for current meeting logistics.