**Agenda**

* F-2-F Agenda
* Versioning – Bob Russell et al
* Fabric interfaces information structure: struct fi\_info / fi\_getinto()

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

**F-2-F** **Date 8/19, 20, Location – Hillsboro, OR**

Suggestion: Add a discussion of test frameworks to allow testing without having to rely on someone’s job launcher and to identify a specific framework (Unit test, Functional test, etc) that would need to be built.

For existing verbs, testing is done as a proprietary function of vendors. In the case of ibverbs, people used the existing microbenchmarks (e.g. perftest does some basic verbs testing) as the model for writing their applications.

We need both, example code implementations, and some sort of unit test or validation.

AR – Howard to verify he can travel, if so he may be able to lead this.

Susan Coulter has also volunteered to take up this topic.

Frank Berry has an RCG who is digging into some examples, and will be able to provide some illustrations a) how this is done in verbs, and b) now this is how it’s done in libfabrics. The parallels are imperfect because verbs does not have some capabilities (e.g. RD) that are available in libfabrics.

**Versioning – Bob Russell et al**

The idea is to settle on a way to incorporate future revisions and build it in from the beginning. For example, when you change or update a data structure? This led to the notion of a self-identifying data structure.

Two possibilities: if you change a data structure, add a new function call or use the built-in size parameter to signal the size of the data structure.

Indirection (double function calls), and extra instructions in the code path, are to be avoided in the name of performance.

Discussed using existing compiler linking mechanisms to enable this. The basic idea is to link the application and the provider, allowing the linker to resolve version levels. But doesn’t this require the provide to be compiled into the library?

Is it a requirement to support out-of-tree providers?

Enforce the requirement that a provider is aware of the version of the middleware layers, therefore guaranteeing that provider will setup the data structures appropriately. Provider registers the middleware version against which it was compiled.

Providers would need to report to libfabrics what version it was built with; libfabrics would fail to load if the version was not correct. Thus compatibility between the provider and libfabrics/ middleware would be enforced.

Following the meeting, Jeff Squyres posted this email to the reflector:

“At the end of the call today, I mentioned that the dlsym-like mechanism for getting vendor extensions could be a little tricky, depending on whether the provider was linked inside libfabric.<ext> or whether it was dlopened by libfabric.<ext>.

Sean asked to see an example so that we could talk though it.

After a little more thought, I realized that my assertion about it "being a little tricky" is wrong. The dlsym-like mechanism should work exactly the same, regardless of whether the provider is linked inside libfabric.<ext> or is dynamically opened.

So... no example needed.”

**Agenda for next meeting**

Resume detailed examination of other interfaces in detail.

**Next regular telecom**

Next meeting: Tuesday, 8/5/14

9am-10am Pacific daylight time

**NOTE: We have shifted over to using WebEx. Please let us know if you don’t have the new meeting invitation.**

-------------------------------------------------------
To join the online meeting (Now from mobile devices!)
-------------------------------------------------------
1. Go to <https://cisco.webex.com/ciscosales/j.php?MTID=m4be80dd52b5aafabb061680546ee195c>
2. Enter your name and email address.
3. Enter the meeting password: libfabric
4. Click "Join Now".

-------------------------------------------------------
To join the teleconference only
-------------------------------------------------------
1. Dial into Cisco WebEx (view all Global Access Numbers at
<http://cisco.com/en/US/about/doing_business/conferencing/index.html>
2. Follow the prompts to enter the Meeting Number (listed above) or Access Code followed by the # sign.

San Jose, CA: +1.408.525.6800 RTP: +1.919.392.3330

US/Canada: +1.866.432.9903 United Kingdom: +44.20.8824.0117

India: +91.80.4350.1111 Germany: +49.619.6773.9002

Japan: +81.3.5763.9394 China: +86.10.8515.5666