**Agenda**

* Continue working through list of issues identified at the F-2-F

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

**Cont’d’ Review output from the MPI point-to-point pseudo-code group**

**See the detailed minutes from Tuesday, 8/19/2014 PM** [**https://www.openfabrics.org/downloads/OFIWG/Hillsboro%20F2F%202014-0819/OFWG\_minutes\_2014-08-19-PM.docx**](https://www.openfabrics.org/downloads/OFIWG/Hillsboro%20F2F%202014-0819/OFWG_minutes_2014-08-19-PM.docx)

**Reviewing MPI Initialization pseudo code group list of issues**.

AV Map vs AV Index (Charles Archer) – the question is whether both are needed. The conclusion is that both are needed.

Resolve shared memory (Charles) – some work remains to be done here; if there are multiple tasks per node, want to be able to share addressing information among multiple processes. Need to make sure the API is right to allow that sharing. Don’t currently know if it’s correct or not. AV Map seems pretty straightforward, AV Index will be more complex.

These are not read only, if doing MPI spawn, have to be able to add entries, which implies a need for locking. It is not a ‘create once’ kind of structure.

Three key issues:

* need a locking mechanism to allow table updates,
* the size of the table is not deterministic,
* need some kind of garbage collection to keep the table from growing monotonically over time.

Locking could possibly be ddone above the API. There is an API for cleaning up, and there is an API definition for allowing sharing.

In short, are the APIs correct, and can the code be written without being overly complicated, both from an application and a provider perspective?

Address Compression (Jeff Squyres) – MPI may need to address many many many peers (1000s is not uncommon today, 10,000s is coming and we should plan for 100,000’s). May want some kind of mechanism to represent a large number of peer addresses in a small amount of memory space.

Q: is this compression in the way the address information is stored, or in the way the application refers to it?

A; Yes. Almost certainly the app will want to be able to pass an address down based on a regular expression or something. There is also an intent for an application to be able to specify a range of addresses (doesn’t work at present).

AV Map would give the fastest access, AV Index would have the smallest footprint.

Currently the API returns an 8 byte value to the app. It’s possible that a provider may not require all 8 bytes. If the provider can somehow indicate that it only uses 4 of the 8 bytes, the application may be able to optimize around that. Consider a mechanism, such as a field, to allow the provider to indicate the address size,

Need a thread-safe way of multiplexing streams across a single connection (Charles A., Jeff S.) – Currently, you have to create a new endpoint, including exchanging endpoint addresses, for each thread or process. Is there some way to create a shorthand version to address multiple threads or processes behind a single endpoint? In OSI terms, it’s similar to a session – want to be able to create multiple sessions over a single connection.

This appears to be an important emerging usage model.

Absent the ability to do this, each thread would have to maintain its own table of endpoint addresses and so on.

We are agreed that this is probably an important feature, but at present don’t have a good way to address it.

**AR – Put this on the GitHub issues list for resolution.**

**Next steps –**

We need to find a way to make progress toward the end of the year goal, which means that work needs to get done outside of the weekly concalls. The following were suggested as areas for focus or tools to help us to focus:

1. Close out on the ‘endpoints table’ created at the F-2-F. See the minutes from Wednesday, 8/20/14 for a view of the current table: <https://www.openfabrics.org/downloads/OFIWG/Hillsboro%20F2F%202014-0819/OFWG_minutes_2014-08-20.docx>
2. Use the GitHub issues list to assign owners and drive issues to closure. This requires some effort to bring the current issues list up to snuff.
3. Fill out and complete the existing (but non-working) sockets model provider.
4. Need a set of higher level documentation to describe libfabrics functionality, e.g. what is RMA and when and how is it used?

Here is the current Minimum Functionality table from the f-2-f:

**Minimum Required Functionality constituting Rel 1.0**

* **In libfabrics core:**
	+ **Code is complete for all defined endpoint types and operations**
	+ **The list of endpoints to be included is to be decided in upcoming meetings**
	+ **There is at least one provider for each endpoint type**

**Atomic CM Msg RMA Tagged Trigger**

**11 EP-Dgrm [\*] [F] [1] [\*] [\*] [\*]**

**12 EP-Msg [p] [1] [1] [1] [?] [?]**

**13 EP-Pkt [\*] [\*] [F] [\*] [\*] [\*]**

**14 EP-Raw [\*] [\*] [F] [\*] [\*] [\*]**

**15 EP-RDM [1] [F] [1] [1] [1] [F]**

F- future

p-partial implementation

\* - n/a

1 – required for rel 1.0

? – not clear yet

**Agenda for next meeting**

Work through the ‘endpoints table’ from the face-to-face.

**Next regular telecom**

Next meeting: Tuesday, 9/23/14

9am-10am Pacific daylight time