**OFI WG Weekly telecom – 05/19/2015**

**Agenda:**

* Roll call, agenda bashing
* Update on Q2 release – 1.0.1?
* Directed Receive – no open github issue (yet)

**Directed Receives – Sean**

- Primarily designed for tag matching, but also useful for normal messaging operations.

- Create a receive buffer for a specific source; and a first buffer as an “any source” buffer. What is the expectation in terms of the order in which those buffers are consumed?

- Tag matching is strictly ordered (it’s a queue).

- Should this be the same for messaging operations? Are there optimizations that would be possible if you knew you were only doing directed receives.

- What’s the use case for mixing these? There may not be one, or at least not one that is obvious (for non-MPI usages).

- If there are directed receive buffers posted for separate sources, there is no concept of ordering between the various directed receive buffers…each is a unique queue.

- One possible use case: Use the ‘any source’ buffer to receive anything that isn’t from a known source (the default case). If this is the case, need to be absolutely clear that an inbound message from a known source will go to the directed receive buffer first, with anything else going to the ‘any source’ buffer.

- In any case, there should be a more rigorous search for use cases.

- Current implementation is that the ‘any source’ buffer and any ‘directed receive’ buffers are treated strictly as a queue; e.g., if the ‘any source’ buffer is posted first, and a ‘directed receive’ buffer is posted second, any inbound message regardless of whether it was from a known source, will go to the ‘any source’ buffer first.

- Two options – conceptually one queue per source (plus an anonymous buffer), or a single queue.

- Is there actually a real use case for (non-tagged) directed receives? If not, should this function be deprecated in the next release? Better to eliminate unneeded functions earlier, rather than later.

- Suggestion is to post a message indicating the intention to deprecate this function and see if anybody notices or objects.

- Not clear if any provider, other than sockets, implements this. USNic, for example, does not (yet).

- Reconsider this at the next meeting, perhaps Howard’s use case is sufficient to justify retaining the capability.

**“Q2” release**

- probably okay to publish an rc1 at the beginning, targeting a release at the end of June, which would put us back on a quarterly schedule.

**Post Release 1.0 – See Sean’s email 4/30/15 – [ofiwg] post OFI 1.0**

Currently about 50 open issues in GitHub. May be able to hit one or two of these for the Q2 release, the rest should be addressed in the Q3 release.

\* Should releases be 1.1, 1.2, 1.3, etc. or 1.0.1, 1.0.2, 1.0.3?

- See Jeff S’s email on the subject, including a pointer to a website.

- Next release (Q2) could be either 1.1 or 1.0.1, depending on how much gets added. Bug fixes etc would be 1.0.1, major functionality would merit a 1.1.

- Dave - Cisco doesn’t have a specific set of features tied to any particular release.

-Sean – PSM provider will have only bug fixes, sockets provider may have new functionality.

\* Support more efficient formats for iovec's (32)

 What formats should be targeted?

 How are the new formats used with the existing APIs?

 Are new APIs required to use the new formats?

- Proposal is to create a ‘strided’ iovector

- May require some kind of a flag for the provider to indicate if it supports this or not.

- Can a stride be negative, i.e. stride backwards through a data segment?

AR – Sean – create a specific patch to be discussed.

\* Multicast support (1001)

 How are multicast groups joined?

 How are multicast addresses specified to the provider?

 Does multicast introduce a new set of APIs, or reuse/extend existing ones?

- Multicast existed originally in the API, but was taken out because of some confusion as to whether a given transaction is unicast or multicast.

- Original proposal assumed that a given endpoint would join only a handful of multicast groups.

- Are multicast groups defined as part of the address vector, or through some other mechanism?

- Do we need a new set of multicast functions, or can we re-use existing functions?

- What about establishing multicast capability as a property of the EP?

AR – Dave Goodell – bone up on IP/Enet multicasting

- Could define endpoints to be unicast only, multicast only, or both. This would eliminate the branch, except for the case where an EP is configured for both. This assumes that the address space is partitioned into a multicast space and a unicast space (assign a bit to indicate mcast).

- Are there use cases that require an EP to support both simultaneously? Christoph may have such a use case.

\* Topology (253)

 What does the topology data look like?

 How do providers discover and report topology information?

- an overview and some ideas may have been presented at the f-2-f, a proposal can be discussed next week.

\* AV support for exchanging provider specific data (298)

 Providers may be able to take advantage of apps that exchange

 data out of band. What would such an interface look like?

- This comes from a USnic discussion, originally raised by Reese.

- Example: an app using an IP address, puts the IP address into the AV, which gets resolved into a MAC ID. Rather than have each node do its own resolution, there may be mechanisms for doing the resolution once and exchanging that information out of band to help avoid e.g. ARP storms.

- Take this one offline for awhile, then bring it back here. (Goodell, Squyres, Hefty)

\* Domain events (244)

 What event types and data should be reported?

- may be related to topology,

- what events does it make sense to reflect upward and are of interest to the application?

\* Environment variables (177)

 There was a discussion for registering environment variables with the

 framework, so that all active variables could be exported by the

 framework and documented.

- Objective is to avoid a large number of env variables (especially undocumented ones!).

- The idea is to allow registration of a set of common environment variables which would be accessible for use by others. The basic idea is to limit the number of env vars.

- Seems like a really good idea, but needs more thought and/or a specific proposal.

Webex link: <https://cisco.webex.com/ciscosales/j.php?MTID=m9389b0513c9ae643d57e2381e254dcf5>
Webex password: ofi

**Future Agenda Topics:**

**OFIWG Download Site:** [www.openfabrics.org/downloads/OFIWG](http://www.openfabrics.org/downloads/OFIWG)

**Github:** <https://github.com/ofiwg/libfabric>

**OFI Software Download Site:** [www.openfabrics.org/downloads/OFI](http://www.openfabrics.org/downloads/OFIWG)

**Link to WebEx Recording** - [**Play recording**](https://cisco.webex.com/ciscosales/lsr.php?RCID=8eeb6b64afa546efaa22d703a9cba5ac)

**Next regular telecon**

Next meeting: Tuesday, 5/26/15

9am-10am Pacific daylight time