**Agenda**

* Discuss Redhat feedback (Doug Ledford)
* Ordering discussion (Howard Pritchard)

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

**Feedback from Redhat – see email thread “[ofiwg] presentation from today's WG meeting”**

Re; event driven interface w/callbacks.

Q: What exactly are you allowed to do in a callback? Malloc? Send new messages? The risk of getting into too deep leading to re-entrance issues.

A: Some work has been done on this, but not released. Don’t actually run the callback, but schedule a thread to execute the callback as a single purpose thread. In theory this should eliminate many of the so-called corner cases but may have some fairly minor performance implications.

Q: Would this create some caching effects? Why not just stick with a polling model? The MPI community has explored this and rejected it (more than once).

A: Never intended as the only programming model. Use a polling model or otherwise for performance, and the callback model for ease of programming.

Q: Difference between the library allocating the thread vs the application allocating the thread?

A: Could be done either way, in the former the library would have to allocate buffers.

Q: What’s the advantage of putting this functionality in the library? Could be very complicated from a number of perspectives…Seems like this really belongs as a ULP layered on top of the basic library.

A: Generally agree

Q: What type of applications are you targeting with this idea of a callback API?

A: targeted at things like messaging classes of applications, different from an MPI kind of application.

A: high performance applications using libevents is another example.

Summary of the feedback is that this should be layered on top of libfabric, but with a close binding (still built as a layered library).

**Ordering Discussion – see slides ofiwg\_WQE\_order\_pg.pptx distributed via the mailing list.**

Objective is to create a clear model for ordering, above and beyond what is available on the man pages today. Want to give consumers some knobs and controls.

Existing man pages are a starting point – needs more discussion to fill this out, which is the point for these slides.

The slides contain a suggestion in the form of a table for ordering for libfabric. Should such a table be included in the man pages?

It would be desirable to select the default behavior that is appropriate for mainstream applications.

Sean plans to attempt to integrate Howard’s proposal to see how it works out. The question of how this is documented remains somewhat cloudy – a voluminous notes section on the man page?

**Agenda for next meeting**

**Next regular telecom**

Next meeting: Tuesday, 5/6/14

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.**