**Agenda**

* Accelio as a model for libfabric (Liran Liss)
* Next steps (Paul, Sean)

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

**Next steps**

To continue making progress, it would be desirable to map out at least some of the key architectural elements remaining to be addressed (we have already looked at, but not necessarily concluded, ordering, callbacks and progress). To identify the most obvious ‘line-of-sight’ issues, we plan to devote a meeting (likely 5/27) to brainstorming. To support that effort, Sean is planning to refresh the current object architecture and present it at the next meeting (5/20) with the thought that we can use the currently defined classes and the methods associated with each class as a starting point. The goal of the next two meetings is to identify open issues, not necessarily to solve them on the spot (that’s the next step).

**Accelio APIs Overview – slides from MLNX presented on-line**

* “application facing APIs”, related to the discussion of controlling progress.
* RPC library – request/response, one-way messaging, duplex sessions
* Object model
  + Dispatcher context; dispatcher uses callbacks to the calling application
  + Session – refers to one endpoint. A session can contain multiple connections, maintains event handles
  + Server context – aka passive side session
  + Connection – associated with both a session and a dispatcher
  + Message – contexts for request and responses
  + Memory region object
* Library is event based
  + Asynchronous – library invokes application callbacks
* APIs – Session APIs, Connection APIs, Data Path…and events associated with each.
* Memory management
  + Includes ring buffers for headers & sm messages, RDMA buffers for large I/O

Q: what is the main difference between Accelio and DAPL?

A: Accelio is middleware over RDMA\_CM and verbs, nothing to do with DAPL. DAPL is very similar (a subset of) to verbs and provides services at the same level as verbs. Doesn’t include event loops, or memory management, or thread management, etc. Accelio, being “application facing”, is more abstract from the underlying hardware. In Accelio, the target was RPC and storage. It is layered on top of verbs, so it is backward compatible and can include new features that are added to the low level library.

Q: What are MLNX’ plans for this? Proprietary? Open Source?

A: Already open source, available on github. (<https://github.com/accelio/>)

**Agenda for next meeting**

5/20 - Review updated architecture

5/27 – Begin discussion to identify open architectural issues

6/3 – Oracle: requirements

**Next regular telecom**

Next meeting: Tuesday, 5/20/14

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.**