**OFI WG Weekly telecom – 11/04/2014**

**Potential Agenda Topics:**

* Detailed inventory of the current state of the generic sockets provider
* End-to-end flow control – someone volunteered to try to capture a table of possible options
* Credit discussion – we discussed this two weeks running – conclusion?
* Release process – this discussion produced several ARs:
  + Investigate using github to publish man pages
  + Look into applying mechanisms used by OpenMPI to our release process
* Interfaces and structures for reporting topology data

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

Credit discussion: still somewhat stalled. Back off and solve some of the underlying problems first e.g. metrics to be used for specifying the magnitude of the resource at the endpoint - Number of queue entries? Memory space? Number of SGL elements?

An inventory of the real types of operations occurring across the API:

1. Buffer size – data transfer into/out of privileged buffers, e.g. the sockets model

2. Number of Transactions – e.g. the IB model counts the number of requests posted to a queue. Requires allocating space for the largest possible number of SGLs, which may be inefficient, especially if most operations consume only one or two SGLs.

3. Size of each Transaction – e.g. the US NIC method counts the number of scatter/gather entries that can be posted. Similar to the IB model, but finer granularity.

Fixed sized or variable sized entries? If fixed size, we can use a simple count, if the entries are variably sized, it becomes a bit more complex, potentially requiring some math to calculate if there is sufficient resource available for the next operation.

Objective: keep the accounting math out of the application. The application simply wants to know if the next post of an operation will succeed or not. What about a call that allows the application to query the amount of queue space left? This complicates completion queuing, and it raises the question of how the provider tracks free space.

One thought: what about providing a memory location that is W/O by the provider and R/O from the application, indicating remaining queue space available?

It appears as though sockets is likely to use an EAGAIN model because it’s difficult to indicate the buffer space remaining

There are four elements that contribute to calculating available queue space:

1. the operation itself

2. # of scatter/gather elements

3. the size of any inline data

4. alignment factor (round up to the next whole buffer size)

Liran suggested a mechanism whereby an application posts a template, or sample request and the provider returns a value indicating how much resource is required to fulfill the request. Liran agreed to summarize the idea in email posted to the mail list.

Agenda for next week:

-TBD

**Next regular telecom**

Next meeting: Tuesday, 11/11/14

9am-10am Pacific daylight time