**Agenda**

* Agenda bashing
* Continue feedback on MPI requirements
* Next steps – mapping out future direction beyond requirements gathering
* Rename the group?

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

**Interest Groups**

Lively offline discussion about renaming the group to achieve two goals: 1. Make the group more easily google-able, 2. Adopt a name that more closely reflects our actual work

**MPI Update – Jeff Squyres Cisco (see slide deck 2014-01-21-mpi-community-feedback.pptx)**

**MPI wants – verbs improvements (cont’d from last week – slide 18)**

* MTU should be an int (not an enum)
* Timeouts for connection requests to avoid the need for a separate progress thread for connections, or a connection manager that completes asynchronously
* All operations need to be non-blocking
* Specify buffer/length as functions, instead of as structs
* Ability to query how many credits currently available in a QP (especially completion queue)
* Remove concept of “Queue pair” – standalone send channels and receive channels. Queues are good, but de-couple the two queues comprising a pair.
* Completion at target for an RDMA write – wrt w/immed comes close, but is limited. Want a genuine completion, prefer without having to post a receive. This would be the first completion that doesn’t require a linkage to a prior work request. Unclear what information needs to be conveyed in the “completion”. Probably just reporting a write to some memory location probably isn’t sufficient. There may be a problem here if there is no upper limit on the number of outstanding events. However, there are other areas where we create back pressure in response to an overflow e.g. on-demand paging.
* Query if loopback communication from one process to another is supported. Given two linux processes, can I queue up an RDMA to you (on the same server).
* Define what function is required versus optional. Verbs functionality is wildly different for each provider, whether it is different transport implementations (iWARP vs IB) or different vendors. Verbs was supposed to be the unifying API, but misses at some points, which means that MPI ends up providing the same functionality everywhere regardless of hardware/platform. There is a bit of schizophrenia in the MPI community – some members want everything provided completely in high level interfaces, others really just want fine-grained features coupled with capability flags.
  + This led to a deeper discussion of multiple ‘levels’ of APIs – primitives vs higher level functions, and allowing the vendor to provider either or both.
* Better fault and error isolation

**Standardized High-level interfaces – from one MPI camp**

* Tag matching, non-blocking collectives, remote atomics, etc. MPI community asking to be involved here.
* Tag matching – “incoming flow steering on steroids with wildcards”. Want to steer it to a process and to a s/w construct within a process. Need a diagram to illustrate this.

**Vendor-specific interfaces – the flipside**

* Direct access to vendor-specific features – any lowest common denominator API may not produce the most efficient implementations.
* Implies a requirement for a robust “capabilities” query, APIs to query what devices and providers are available at run-time. Vendor-specific extensions.

**Memory Registration**

* Is memory registration required (run-time query)?

**Fork Behavior**

* MPI does not define what happens on a fork, but folk history has produced a set of colloquial behaviors.
* In child, all mem accessible, network handles are unusable/stale, can re-initialize the network to get new handles.
* In the parent: continue on independent of the child.

The balance of Jeff’s material deferred until the 2/18 meeting

**Next meeting**

* SNIA presentation on non-volatile memory

Bin list:

* Complete MPI requirements discussion
* Consider re-naming the group to improve googleability?
* Steps forward beyond requirements gathering.

Logistics

Tuesday, 2/11/14

9am-10am Pacific time

<<https://meet.intel.com/sean.hefty/D14C7J10>>  
<https://meet.intel.com/sean.hefty/D14C7J10>  
  
Join by Phone  
1-888-875-9370  
+1(916)356-2663 (or your local bridge access #) Choose bridge 5.  
Find a local number<<https://dial.intel.com>>  
  
Conference ID: 746347966