**Agenda**

* Agenda bashing
* Continue feedback on MPI requirements

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

**Interest Groups**

OpenFabrics not very Googleable – Consider changing the name? Work with Ken Strandberg to push ourselves up the stack? Continue this discussion offline, and add it to next week’s agenda.

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

**Quick re-cap**

High-level abstraction, could even be memory to memory

Connectionless

All communication is reliable, some ordering rules, all messages are typed.

Addressing is via ‘tuples’ called a communicator. Endpoints are processes, not servers.

Communications modes – blocking, non-blocking, point-to-point two sided and one-sided, collectives (broadcast, scatter, reduce, etc.

Async progression is required

**Different MPI Camps**, sometimes with varying requirements e.g. high level interfaces and very low level interfaces. This may be reconcilable through the fabric interface mechanism, i.e. because we the framework supports a variety of types of APIs.

**Basic Things needed – new material for this week**

* Message-based (stream are a huge pain)
* Should be very efficient in terms of latency, instructions in the critical path, zero copy, etc.
* Async operation – initate an operation, check for follow-on completion. Applies also to non-rendezvous operations
* One-sided ops (incl atomics)
* Two-sided semantics
* No concept of queues from the MPI API perspective, but from the hardware implementation point of view there is a notion of flow control (to avoid overrunning the h/w queues).
* Asynchronous progress independent of API calls – does not play well together with today’s verbs. For example, if there are multiple consumers (e.g. MPI and PGAS) in the same process.
* Scalable communications with millions of peers (processes) needed for both one-sided and two-sided semantics.

**Things Verbs does Right**

* Reliable and unreliable connections, connectionless, ability to target a remote memory target, signaling built in (e.g. RDMA write w/ immed) but want more.
* Ability to re-use “inline” buffers
* Both polling and blocking QP modes
* Discover devices, ports and their capabilities, but not tied to a specific hardware.
* Scatter/gather lists for sends, Atomics (but want more)

**Verbs doesn’t impose a bunch of restrictions**

**MPI wants – verbs improvements**

* MTU should be an int (not an enum)
* Timeouts for connection requests to avoid the need for a separate progress thread for connections
* All operations need to be non-blocking
* Specify buffer/length as functions, not as structs
* Ability to query how many credits currently available in a QP
* Remove concept of “Queue pair” – standalone send channels and receive channels
* Completion at target for an RDMA write – wrt w/immed comes close, but is limited. Want a genuine completion.

**Next meeting**

* Re-name the group to improve googleability?
* Continue feedback from the MPI community.
* Looking ahead – next steps

Logistics

Tuesday, 2/04/14

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