**Agenda**

* Agenda bashing
* Complete MPI requirements discussion
* Rename the group?
* Next steps – mapping out future direction beyond requirements gathering

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

**Continuing MPI requirements – Jeff Squyres. Resuming from 2/4/14 meeting**

Slide 30 – other things that MPI wants

* If n/w header knowledge is required, don’t mandate a specific header. Provide a run-time header to query the header details.
* Request ordered vs unordered delivery
* Completions on both sides of a write.

Slide 31

* Listen on a specific network (well-known) address
	+ 20 MPI processes per machine, 20k machines – each one is going to open at least one endpoint. Would be nice to assign a predictable port to each process. This allows dramatic reductions in startup times.
	+ How to handle multiple jobs on one machine? This would require the scheduler to tell each process. Focus is on starting up a single large job on a single large machine.
* Start-up processing has become a significant hot button.
* Find a way to scale the size of incoming receive buffers. Allow receiver providers to consume buffering directly related to the size of incoming messages.

Slide 32

* Generic completion types such as aggregated completions or vendor-specific events (allowing vendor extensions wherever possible).
* Out-of-band messaging, allowing messages to be sent with variable service guarantees.

Slide 33

* Scatter-gather lists of greater than one. Noncontiguous sends, receives and RDMA operations.
* Access to underlying performance counters. Figure out why performance is not what is expected.
* Set/get network quality of service

Slide 34

* a richer set of atomics.
* Determine if atomics are coherent with the processor via a run-time query.

Slide 35

* Offset-based vs virtual addressing, where offset means an offset from a known virtual address. The known virtual address could be known only at the destination.
* Some mechanism to discover if virtual addressed-based communication is better/worse than using offsets.

Slide 36

* Remote keys assigned by the application, allowing the application to distribute the key it expects.
* Specify arbitrarily-sized atomics.
* Flexibility in specifying ordering of atomics, or to know if ordering is even supported.

Slide 37

* n/w topology awareness and discovery to support topology-aware processes. Important: provide user-level read-only access to this topology information.

Slide 38

* w/o tag matching, MPI frequently sends two buffers – one for header, one for payload. Note that the headers may not be of fixed lengths, but can be bounded.
* Thread safety – MPI sometimes needs this, sometimes doesn’t. Sometimes in a single process. Allowing choice at run-time would be useful.
* Checkpoint/restart – make it safe to close stale handles and reclaim resources. For example, if you try to restart a saved checkpoint on a different node, likely to get all kinds of seg faults. (This requirement may not be well-formed.)

Slide 39

* Don’t assume:
	+ Max size of transfers
	+ Location of the memory translation unit.
	+ All comms buffers are in main RAM
	+ API handles refer to unique hardware resources
* Be “as reliable as sockets”

**Next meeting**

PGAS discussion

Bin list

* Consider re-naming the group to improve googleability?
* Next steps forward

Logistics

Tuesday, 2/25/14

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