**Agenda**

* Agenda bashing
* Feedback from PGAS community
* 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

**PGAS Feedback – Howard Pritchard See uploaded slides ofa\_pgas\_frame\_wg – v2.pptx**

Brief SHMEM background

* Five key topics: endpoint considerations, memory registration, remote memory references.

Open SHMEM

* All ranks run the same program. Ranks are called “PEs”
* Two types of memory regions – data segments, symmetric heap
* SHMEM APIs today are blocking, would like to explore non-blocking APIs

UPC

* Compiler-based programming model, basically c with extensions.
* Thread-based (as opposed to rank based) Thread roughly equivalent to MPI ranks

Fortran 2008 (fka CoArrays or CAF)

* Also compiler based, one-sided

Different points of view

* As with MPI, the community is diverse, both from an implementer perspective and a user perspective.

Community Feedback

* Many requirements are the same or similar to MPI. Therefore focusing mainly on issues related to one-sided programming models.

Endpoint considerations

* Scalable endpoint memory usage
* Connectionless is preferred, but still reliable.
* If connected, need an efficient all-to-all primarily for start-up.
* Requirement for “thread hot” thread safety model, i.e., multiple threads on a given node should be able to send messages to multiple other endpoints, conceptually similar to having multiple QPs. Think of them as communication channels.

Memory registration

* Memory reg is a necessary evil, but given the necessity it should be scalable.
* On-demand paging – want flexibility to also support traditional pinned memory pages.
* Ability to register large amounts of virtual memory that is sparsely populated.
* “lazy pinning”

Small remote memory reference API requirements

* Need small non-blocking PUTS and GETS. Typically much smaller than MPI.
* PUT with various completion notification mechanisms

Ordering for small remote memory references

* Necessary to avoid Write-after-Write (WAW), Write-after-Read (WAR) and Read-after-Write (RAW) hazards.

Atomics

* Need more than FADD and CSWAP
* Need reliable AMOs, but there are cases where unreliable AMOs would be helpful.
* Cached Atomics coherency (cached on NIC)

Completion efficiency

* Ability to batch certain types of completions, same as described for MPI.

Large data transfers – RDMA

* Very similar to MPI requirements
* Option for fences between transactions
* Immediate data requirement (again similar to MPI request). Would like to use the immed data as inputs to invoke a function. Increase immediate data field to 64 bits. May want to improve completion semantics.

Collectives

* Collectives need to be defined.

Active Message Support

**Next meeting**

Bin list:

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

Logistics

Tuesday, 3/4/14

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