**Agenda**

* Application characterization and taxonomy – identify interested parties who can help define application requirements

**Next meeting** – Tuesday, 1/14/14 9:00AM PST

Will need to schedule a new meeting and send out updated bridge information.

**Application Taxonomy slides**

Identified interested parties for the various applications.

Jeff Squyres will try to pull together some slides on MPI requirements of an API. He will also speak with Nathan Young. The goal is to list requirements MPI has on an API, what it wants from the providers, and identify places where the current software stack matches well and mismatches with what MPI wants.

Howard Pritchard from Cray will attempt to do the same for PGAS. Their implementation does not use verbs, so might be a good candidate to understand what changes will be needed by the proposed framework. He anticipates needing ‘a few’ weeks to pull together the necessary material.

There’s some concern that the taxonomy may not quite capture database systems accurately. It was indicated that databases have a couple different requirement sets, depending on whether the data is in memory or storage. Panda, Krishna, and Bernard were indicated as contacts for details on exploring requirements for databases.

For storage, contacts were provided for Lustre. A mention was also made to support user space file systems, such as Ceph, which is currently targeting rsockets as its interface. Sean has Ceph contacts.

**Opens**

Pradeep mentioned that Bernard had a concern regarding compatibility. They have a j-verbs (java-verbs) implementation that calls verbs. They would like any new interface to support j-verbs. It was mentioned that the current solution is backwards compatible with libibverbs, but requires re-compiling. Compatibility has been taken into account, both in terms of applications and providers, to help smooth a transition to using new interfaces. This concern was captured in the ‘issues’ spreadsheet.

Another concern which was brought up was the limitation that the existing interfaces have using 32-bit values in places. For example, sizes are restricted to 32-bits. The new APIs use 64-bit (size\_t) values. This concern was also captured as a requirement of the new API.

There was a brief discussion and general agreement that objects in the new API be opaque. This is to avoid problems that exist with the current API that exposes internal data structures. This concern was captured as an issue.