Sean Hefty (Intel), Paul Grun (Cray), Mark Atkins (IBM), Frank Berry (Intel), Hal Rosenstock (Mlnx), Jim Ryan (Intel), Zarka Cvetanovic (HP), Rupert Dance (SFI), Jimmy Hill (Oracle), Bernard Metzler (IBM), Howard Pritchard (Cray), Pradeep Satyanarayana, Jeff Squyres (Cisco), Bob Russell (UNH), Bob Woodruff (Intel), Sayantan Sur, Todd Rimmer (Intel), Don Wood (Intel), Doug Ledford, Ken (Mercury Systems), Dave Goodell (Cisco), Brad Benton (AMD), Christoph Lameter (Jump Trading), Bill Snapko (SGI), Avnesh Pant (Oracle)

**Agenda**

* Regular meeting times
* Review policies and procedures (deferred until next meeting)
* Continue Sean’s presentation from last week

**Meeting times** – everyone is asked to fill out the doodle poll, beginning Tuesday November 12. Meetings will be bi-weekly.

**Process & procedures** – Will be presented at the next BoD meeting for approval.

**Sean’s slides – beginning with slide 36**

There had been a request to publish headers, but not done yet.

Fabric Information – related to RDMA getaddrinfo. Allows discovery of what kinds of service are available from providers.

Base Fabric Descriptor – struct fid is the basic object. Each fid has four basic ops – close, bind, sync, control. Very object-like. The objective is to reduce/eliminate branches.

FI-Communication – defines sets of operations (e.g. msg, cm, rdma, tagged…) New sets can be added, and new capabilities can be added to any given set.

Question – how to expose synchronous/asynchronous – all ops shown so far are asynchronous.

Not making any assumptions at this point about threaded/non-threaded. Should support both as a function of application requirements.

**Use of built-in library versioning?** No obvious objections, but keep the discussion open. Want to be able to build against a newer version of the library, but running against an older version. (This requirement was described by Intel MPI team). This places at least some burden on the application; need to hear from the application community on this. The ability to run against older versions implies some ability for an app to query capabilities. This querying would be done once, so unless it appears in the data transfer path it shouldn’t be a performance impact.

This discussion remains open for awhile, pending further input from application community.

Suggestion was to use dl open to solve the problem of extensions and versioning. No conclusion yet.

**slide 27**

Resource domain contains an event collector (an abstracted version of an event queue) and an address vector (encapsulates the basic ideas of an address query). The address vector cleans up the layering such that the application no longer need be concerned with address lookup which is pushed down to the provider.

**slide 30 - event collector**

Proposing to separate out errors from completions. This differs from verbs. Similarly, wait objects are created to give the user control over waits.

Suggestion – in some cases you want the thread to wait, but another thread to poll. In theory, the proposed mechanism supports this.

**Next meeting:**

Policy and procedures

SC13 BoF

Verbs compatibility