Discussion Notes: MPI Low-level Access

Goal:
Ability to have greater visibility into what MPI is doing and whether it's doing something to use MPI. From a user perspective. To optimize it's use.

- Detect excessive blocking delay
- Load imbalance
- Are you resource constrained - exhausting the buffers
- Connection constrained
-

We need to add to the MPI specs through the MPI forum. But we need to build a case for it – easy to integrate, well defined, use cases and that there is utility (people will use).

Would like a model to get global state information. What are the states? Do we set up generic states that can be defined by too developers as an add on to MPI?

- working
- waiting
- idle
- extensible

Support for other "state variables" (e.g., number of requests, number of UMQ entries)?

Should we get timers, counters, global variables? For what? What are they?

Publish events that MPI implementation supports collecting information on; a basic publish/subscribe interface; any predefined information types?

What was the cost of using Peruse? Was it the callbacks or the potentially large number of supported events? Other? What parts of Peruse were useful and should be maintained (at least in a similar way)? Access to environment variables?

Are callbacks an effective way to gather information? Are they cheaper? Used extensively within MPI community in the MPI implementations.
There have been objections to the use of callbacks in PERUSE from the MPI implementer community. What are the technical reasons? If we rely on callbacks: what can we actually allow within a callback?

How about Query capabilities? Global variables vs. query functions? Macros?

What is the method of collecting?

How do we get the data? Register – ability to turn things on or off.
Support for interrupts at counter overflows? Same problems as with callbacks?

Can the MPI implementations collect time for events and report back when it's considered safe.

How can you tell if you are exhausting resources?
Can we gather a % of a resource? We would need to know what the resources are.

Can you only get info from your own thread or can you do more global polling (or query specific other threads).

Do you need to get information before MPI_Init and after MPI_Finalize

Use cases:

How do we define what are performance problems?

State transition from waiting to working.

Detect connection cache trashing

Other issues for MPI Forum:

Could we add something to the standard to eliminate the need for binary rewriting for P^NMPI (not have PMPI call necessarily call into the MPI library)?

Subsetting proposal that would make it acceptable to build MPI libraries that do not include the profiling interface

Piggybacking support within the standard? Perhaps as part of a revised datatype management?

> Investigated as part of the FT group at the MPI forum