Tools interface WG

What interfaces are available for HPC?

- CTI
- OMPT/OMPD
- Launchmon?
- Perf
- Papi
- Ptrace
- Ld_audit
- MPIT/QMPI/MPIR
- Sdt systemtap
- caliper
- What’s missing? Who needs to look into this?

MPIR replacement

- No clear direction from MPI tools wg
- OpenMPI has MPIR on deprecated/deleted
- PMIx working on an alternative – not great buy-in
  - Debugger involvement is wrong approach

CTI changes?

- Perftools did not traditionally use MPIR
  - Abstraction to job launch
- Ability to bootstrap any TBON style communication trees
  - Launchmon has this
  - Not just tool daemon launch, but ”tool launch”
- Query a location to write to file system?
  - Hard problem to solve in the generic sense
- Query compute node host OS
- Default compiler for tools should match C++ used by lib fe
- Launch with bootstrapper process
  - CTI does the argv parsing

Tools in the cloud

- Cloud/container solutions for daemon launch?
  - Investigate partial solutions that might already exist
  - Kubernetes inserts gdbserver into container instance
  - Container isolation?
  - AI: Andrew and Ben to continue conversation via email
- Issues of compatibility with containers
  - MPI compatibility
  - Tools compatibility: FE with different OS into ABI incompatible container
- Push intelligence down into container
  - Thread_db access occurs into container
  - Intelligence moves downward

Tools in the cloud

- App is container but user owns the node
  - Potential use case
- Ptrace reaches through into the container?
- MPI injected into container – libc not forward compatible

What is needed in tool interfaces

- Tool interfaces should…
  - Be C based ideally
  - If conditions are right, use C++
  - Thread safety is a secondary requirement
- Launchmon
  - C based API/C++ implementation
- Multi-tool support
  - Proper isolation
  - Come up with scheme for unique/ephemeral ports
  - Built key sharing implementation into interface

What is needed in tool interfaces

- Performance tools..
  - Use files on lustre and use post processing
  - Use custom bootstrappers
    - Script or executable run before application to modify environment of application
  - Use MPI for communication
- By having a formal interface it allows us to…
  - Facilities can have center wide policies
    - SLURM plugin can call CTI via hpctoolkit for example
  - User doesn’t have to modify job scripts to embed tool support
  - Much better place for mixing tools: hpctoolkit and totalview at the same time

What is needed in tool interfaces

- Thorny problem is multiple debuggers at the same time
  - Consensus is may not be applicable to CTI
- CTI FE query what is available on backend
  - Separate interface is a better place for this