MPI Tools Breakout

Scalable Tools Workshop 2023
Group Lead: Martin Schulz

https://bit.ly/stw-mpi

Participants

- Martin Schulz (TU-Munich)
- Sameer Shende (U. Oregon)
- Laksono Adhianto (Rice)
- Jonathon Anderson (Rice)
- Emmanuel Oseret (Univ. Paris-Saclay)
- Jan Laukemann (NHR@FAU)
- Jesus Labarta (BSC)

Topic List

- Impact of MPI Sessions
  - Tracking of open sessions
  - Changing process sets / Malleability
- GPU bindings for some MPI calls
- ABI
- QMPI Interface
- Define performance variables and events for MPIT interface
  - Common semantics that can be added to the standard
  - Similar to OMPT

MPI ABI and Sessions

MPI Sessions

- Separated init/finalize for libraries
- Waiting for usage in actual codes
- State of the art: sit and wait :)

MPI ABI

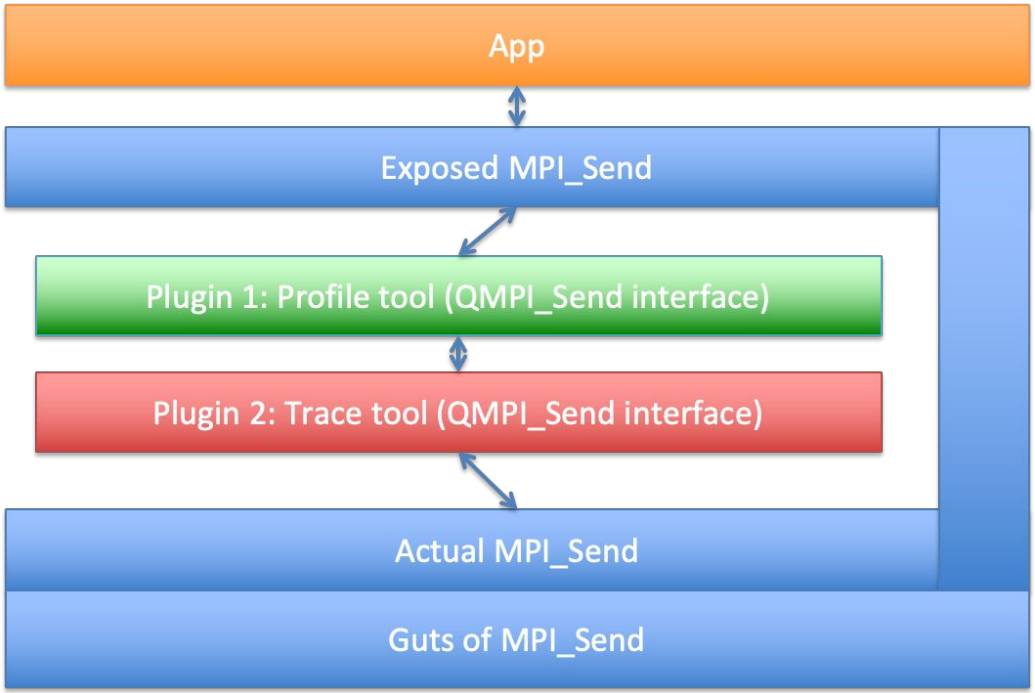
- Currently under consideration, but looking good
- Possibly for MPI 4.2
- Issues found:
  - MPI_KEYVAL_INVALID is not listed as compile time constant in 2.5.4
    - Used in 7.7.2 / Example 7.7.6
  - MPI_SESSION_NULL/COMM_NULL
    - Example 11.9 uses it as static initializer
-

QMPI Interface

https://github.com/mpi-forum/mpi-issues/issues/638

https://github.com/pmodels/mpich/pull/5035

Interactions with AI?
Depends on progress with ABI



MPI_T_* Interface / Standardized Variables

We should have some defined names

- Similar to OMPT where this was very successful
- Alias interface may be neededl

Do we need to better specify MPIT & Threads?

- Happens before relationship for threads and variable updates?
- Relationship to the “logically concurrent” discussion

Need to clarify threading also for event callbacks

- Needs to be clearly documented to not confuse tools
    - Especially if this comes from a background thread
    - In this case, need to be able to attribute back to user/app thread
- May need to extend context information on called events

Possible targets for standardized events and variables

Stalls

- “Waiting time” in the library when no progress is made
- How to define progress?
- Can we borrow the definition from OpenMP?
    - Waiting for barriers / synchronization
    - Housekeeping

One current approach:
TAU adds barrier to distinguish syn time and comm time (not in MPI, yet)

- “MPI Collective Sync” timer (also CrayPAT had the same)
- Only focus on “ALL” collectives

Possible targets for standardized events and variables

Messaging events

- Message send / copy to wire
- Message receive / copy from wire
  - Would be helpful, but may be difficult
  - What is the exact definition, e.g., with RDMA?
  - Must have opt-out model
- Portable definition?

Buffer allocation events

- Internal buffers
- BSend buffers
- Memory allocation kinds
- Size

Possible targets for standardized events and variables

Queue size for sending, receive, UMQ variables

- Logical queues in MPI
- Phyiscal queue in NIC / injection efficiency

Suggestion: “Total bytes queued up”

Number of collectives offloaded to a NIC/Switch

- Is this too specific?

More generic:

- Provide usage percentage
- Ability to query a list of resources
- Possibly make this a new class of PVARs
- Tools can then simply list all resource usages by looping of PVARs of that class

New feature requests:

Node-level barriers

- Used for load balancing on node
- Used for runtimes that can re-distribute threads for work on node
- May be needed to be subsetted for individual groups
    - Different code modules co-running on same node

May be solvable via topology split operation

- There exists a split type for shared memory, but may not be what you need
- Need names for “node” or other HW types to be able to have a better controls of types of sharing