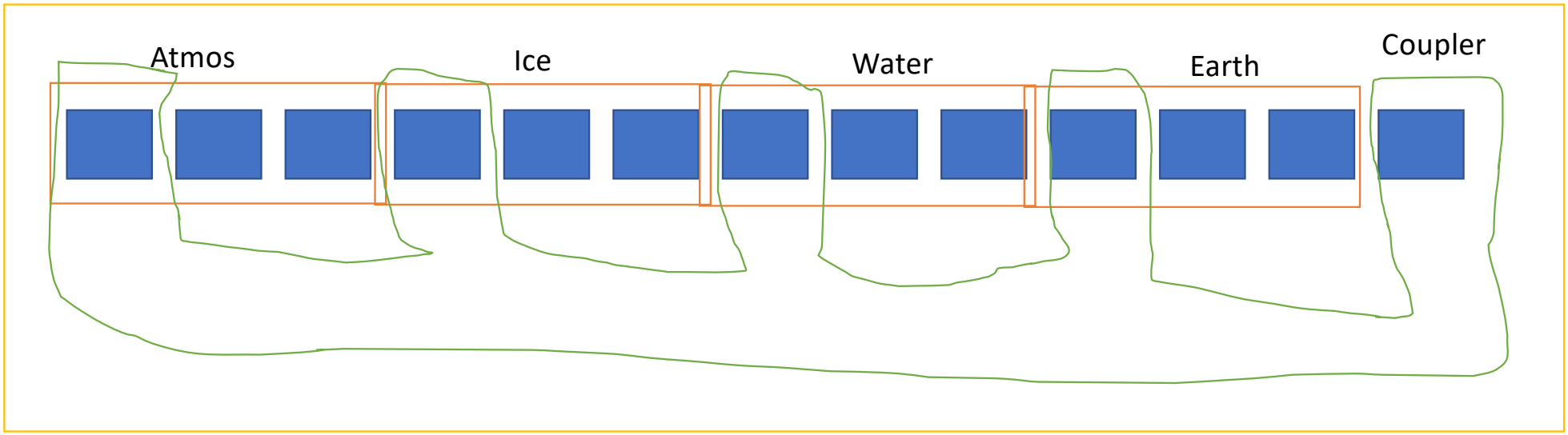
Modern MPI & Tools

Sessions, Malleability and Other Fun Stuff

Session Model for Multi-Set Comms



- Advantages in MPI 4.0
  - Enable more scalability in the library by avoiding global objects
- Advantages in MPI 5.0++
  - Options for Isolation, Malleability and FT

Issue 1: Lack of Global Init and Finalize

- Would want global entry point for tool initialization
  - Problem: that may not be possible, as it doesn't exist

- At least: Would want local entry point for tool initialization
  - Would want to know when first MPI routined is called
    - Without having to intercept every routine
  - Could be done with ”standardized” MPI_T event
  - Would have to be registered in a static global initializer

- Finalize: use exit handler
  - Problem: order of exit handlers / is MPI still viable
  - Consider MPI_T with “standardized” event

Per Session Tools

- Need to intercept Initialization
  - Identification of Sessions
  - Or perhaps just really the Comm create
- Careful when using Finalize
  - Can be local operation, but collective ops should be possible
- General question: is this sufficient for most/all tools or do tools need to exist that can capture truly global behavior?
  - Answer: probably the latter, at least for some

Interactions between Tools and Sessions

- Would be nice to have a PMPI/QMPI tool stack per session / per process
  - But how would be associate them?
  - May be just a wishlist item (as a spatial optimization)

- Would like to get a process set that covers all currently valid processes
  - Is this even possible? Probably not, especially not with a collective call-in point

- Tools may need to change to move from processing at finalize to postprocessing?
  - Or ending processes may have to hand over data (similar to apps)

Issue 2: Malleabilty

- Need a “global” (within a bubble, regional) sync point during negotiation/acceptance (even if app only uses P2P exchange) where a tool can call collective operations
  - Tools can use that to redistribute their data
  - Regional: all processes involved in communicators affected by the change in resources
- Important: With multiple “bubbles” (app/library/tool/…) changes in one need to be followed by all others

Wishlist for the MPI Forum

- MPI_T Events for Init and Finalize

- Query session ID from other objects
  - From communicators, groups, windows, files, requests, …
  - “Normal” MPI function
  - MPI_GET_SESSION_FROM_COMM

- Translate ranks across sessions
  - Should we make an exception for these MPI routines?
  - Doesn’t impact resource isolation

- Prototype implementations of MPI that do not just map to WPM (World Process Model)
  - Ability to test against

- Common tool support library to track/intercept/…
  - Could be part of the new QMPI interface as a “side library”
  - Need to capture common practice first -> Tools WG

- Handle introspection interface