MPI Hybrid -- Endpoint

Status 3/7/10

(Marc Snir)

General

- Full proposal on supporting multiple “MPI agents” within same address space
  - MPI extensions (heavily discussed / revised)
  - OpenMP binding (some discussion)
  - PGAS bindings (no discussion yet)
- Next:
  - Circulate with PGAS crowd
  - Provide more examples

Initialization

- Two phase:
  - MPI_INIT_ENDPOINT

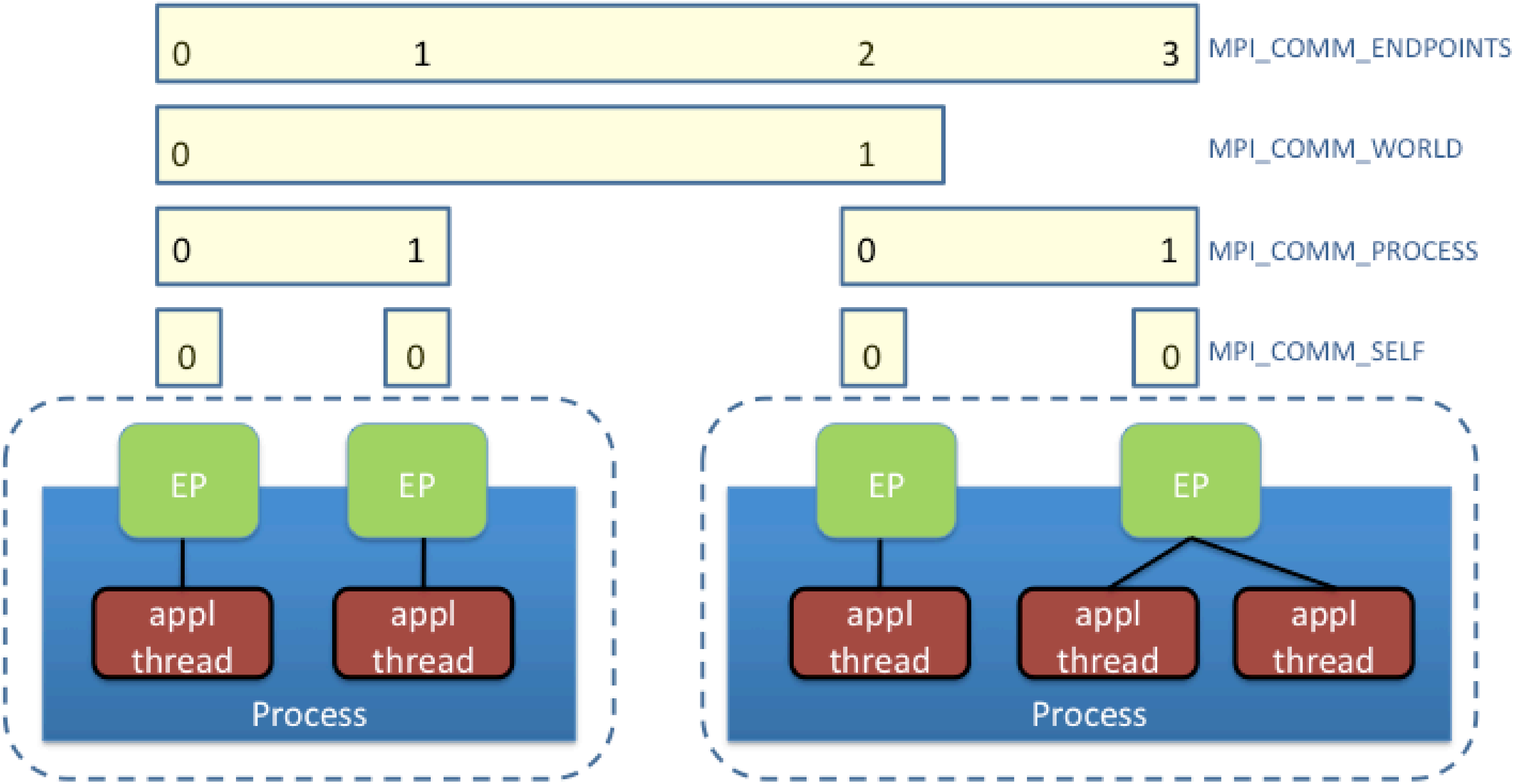
    Returns info on max #endpoints, #procs, process_rank

    (info needed to decide how many endpoints to create)
  - MPI_ENDPOINT_CREATE

    Returns endpoints
  - MPI_THREAD_ATTACH/MPI_THREAD_DETACH

    Attaches/detaches thread to/from endpoint

After Endpoint Creation



Compatibility with Current MPI

- "Legacy" code needs no change
- "Legacy" code can invoke libraries that use multiple endpoints provided that
  - Endpoints are created at initialization time
  - Endpoints are passed to library initialization routine
- Libraries can be written to be agnostic to number of endpoints per address space
  - Different MPI agents (using different endpoints) communicate only via MPI

Library with Endpoints Invoked in “Legacy MPI Code” – Using all Processes

```
MPI_INIT_ENDPOINTS(…);
MPI_ENDPOINT_CREATE(count, endpoints);
Lib_foo_init(count, endpoints,…,&context);
….
Lib_foo_invoke(context,…)
```

- Library invoked by one thread per process
- Can use, internally, multiple endpoints and threads per process

Same, but Using Only Some Processes

```
MPI_INIT_ENDPOINTS(…);
MPI_ENDPOINT_CREATE(count, endpoints);
Lib_foo_init1(count, endpoints,…);
….
Lib_foo_init2(comm,…,&context);
…
Lib_fool_invoke(context,…);
```

- Init2 can be invoked by a subset of processes
- Library is invoked once on each process in comm
- Can use multiple endpoints and threads on each of these processes

Same -- Variant

```
MPI_INIT_ENDPOINTS(…);
MPI_ENDPOINT_CREATE(count, endpoints);
Lib_foo_init(count, endpoints,…);
…
Lib_foo_invoke(comm,…)
```

- foo is invoked once on each process of comm
- foo creates, on first invocation, communicator that uses all endpoints at each of the processes
- foo reuses this communicator at subsequent invocations with same comm argument

Function Needed to Support Last Two Examples

MPI_ENDPOINT_ALLOCATE(comm, count, endpoints, &newcomm)

- Input communicator has one endpoint per process
- Output communicator has multiple endpoints per process

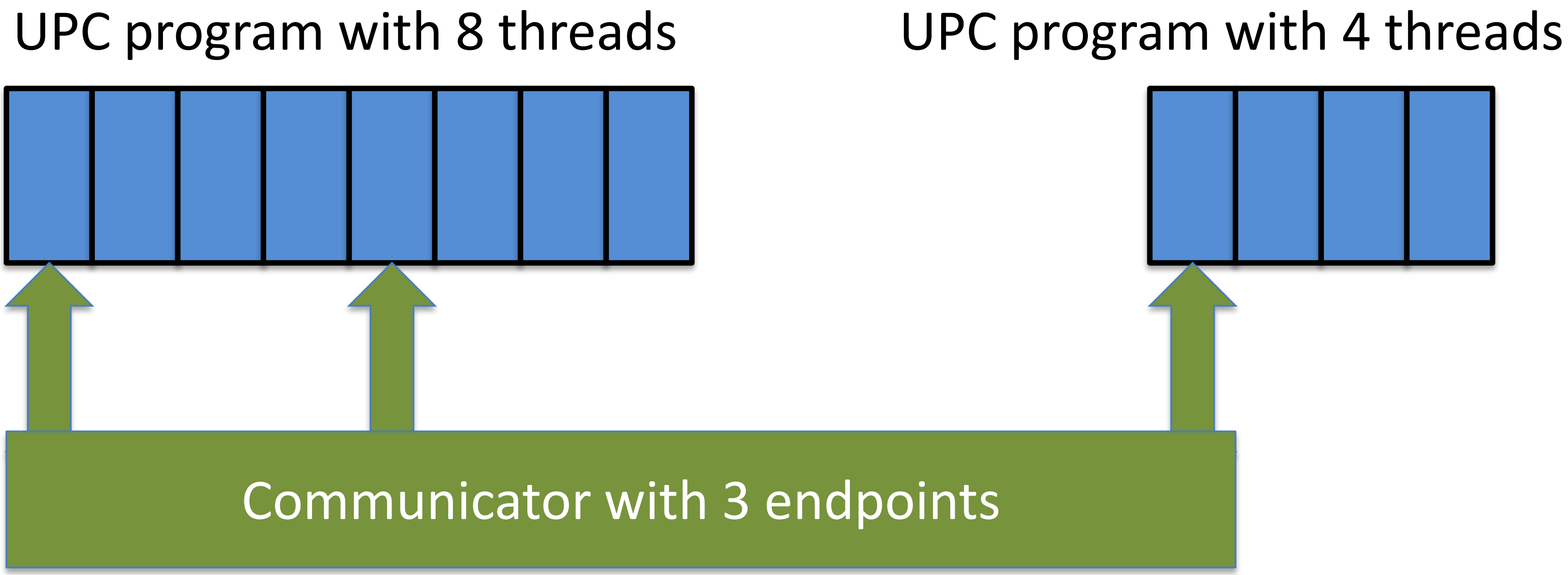
MPI+OpenMP/TBB/…

- Endpoints are attached by threads (not tasks)
- User has to ensure that task is not migrated to another thread after attach and before MPI call.

MPI+PGAS

- Can connect, using MPI, multiple UPC/CAF programs (teams)
- Each program can have multiple MPI endpoints
- Each endpoint is used only by one UPC thread (CAF image)
- User is oblivious to UPC/CAF choice of number of threads (images) per address space

Example


UPC program with 8 threads
UPC program with 4 threads
Communicator with 3 endpoints

- Programmer is oblivious to the number of OS processes used by each UPC program