The Use of Split Processes in Virtualizing the Access to Hardware Resources

Yao Xu
Northeastern University
xu.yao1@northeastern.edu

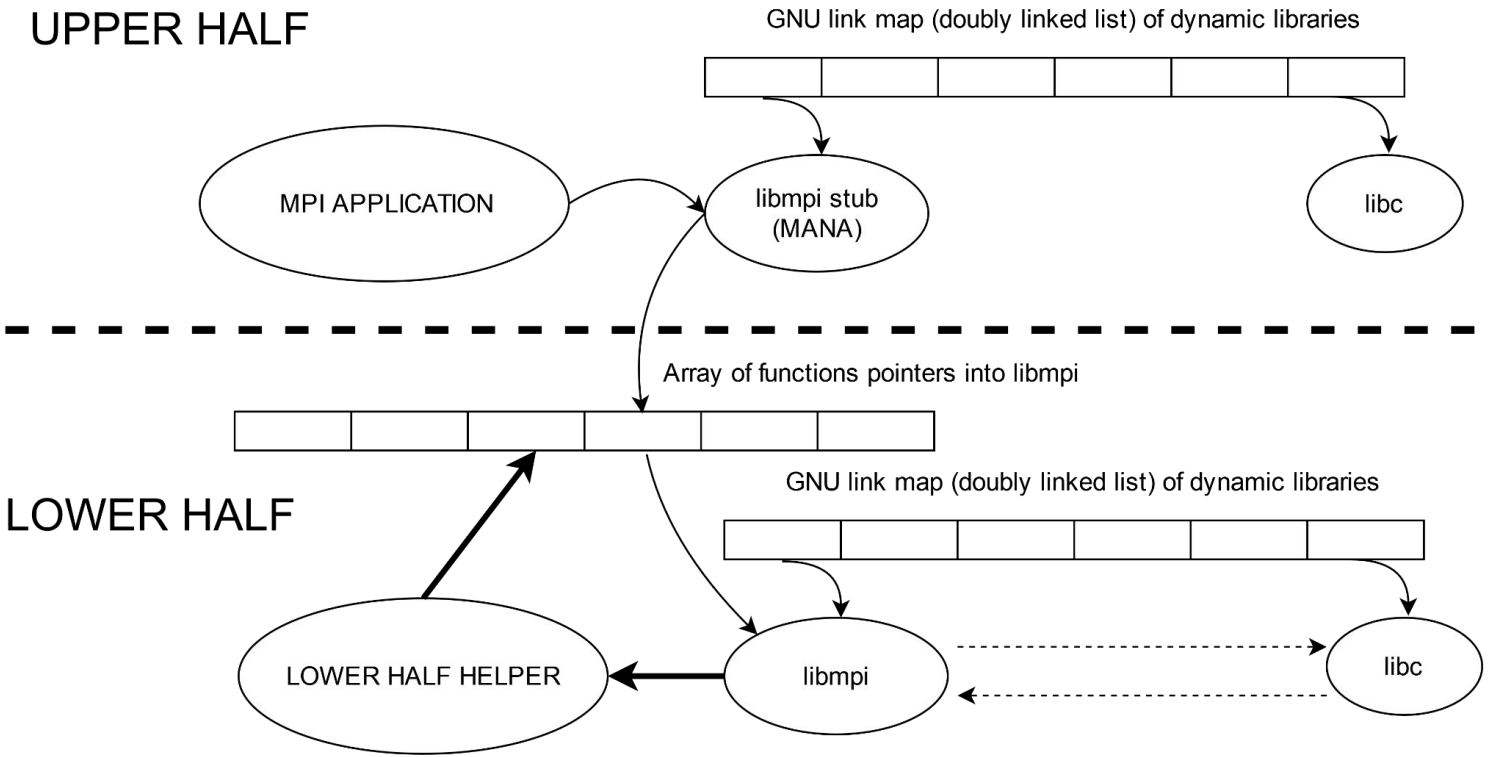
DMTCP: Distributed MultiThreaded Checkpointing

- Checkpoint/Restart (C/R) tool that can transparently checkpoint a threaded or distributed computation into disk
- Requires no modifications to user codes or to the Linux kernel
- Requires no root privilege

Challenges in Transparently Checpoint MPI

- Different MPI implementations
  - Open MPI
  - MPICH
  - Cray MPI
  - ...
- Different networks
  - InfiniBand
  - Cray GNI
  - TCP/IP
  - ...

Split Process



MANA: MPI-Agnostic Network-Agnostic

- Designed by Rohan Garg et al.
- A plugin for DMTCP
- Can checkpoint and restart with different MPI implementations or networks
- Latest MANA 2.0 has been tested with HPE Cray MPI and MPICH on multiple clusters, including the Cori supercomputer at NERSC

CRAC: Checkpoint-Restart Architecture for CUDA

- Designed by Twinkle Jain et al.
- Low runtime overhead (approximately 1% or less)
- Support for scalable CUDA streams
- Support for the full features of Unified Virtual Memory

Future of Split Processes

- Applications in more areas

Proposal for a new abstraction: Medium-Weight Process (MWP)

- Hardware support for easier development and debugging

Medium-Weight Process (MWP)

An MWP is like a LWP (Thread), but heavier.

Each MWP has:

- Individual text segment
- Individual virtual memory mmap regions

Between MWPs:

- Shared memory with other MWPs
- One MWP’s text segment can call another MWP’s text segment directly
- One single FS register and thread-local region for the entire process

Wish List for Kernel/Hardware Support

- Tag virtual memory regions for upper/lower half
  - Avoid merging of regions from upper and lower halves (if same attributes)
  - What if the upper half asks the lower half to allocate memory? (e.g., MPI_Alloc_mem)
  - Tagged TLB: one TLB entry for lower half & others for upper half
- Kernel loader
  - Native support for loading two programs into one virtual memory space
- Debugging
  - Currently breakpoints in debuggers can be used in upper or lower half, but not both.
  - Can we make the debugger have a view of both halves?
- Hardware acceleration
  - Treating paging independently for upper and lower halves

Thank you