Research Infrastructure

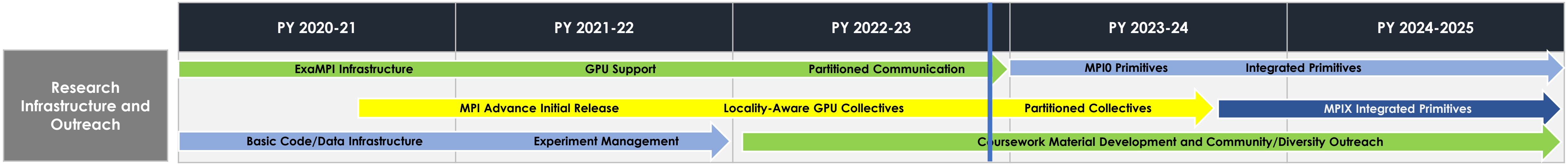
Derek Schafer (dschafer1@unm.edu)

University of New Mexico
Center for Advanced Research Computing

September 28th, 2023

Introduction

- Key Topics Outline:
  - ExaMPI
  - MPI Advance



PY 2020-21

PY 2021-22

PY 2022-23

PY 2023-24

PY 2024-2025

Research Infrastructure and Outreach

ExaMPI Infrastructure

GPU Support

Partitioned Communication

MPI0 Primitives

Integrated Primitives

MPI Advance Initial Release

Locality-Aware GPU Collectives

Partitioned Collectives

MPIX Integrated Primitives

Basic Code/Data Infrastructure

Experiment Management

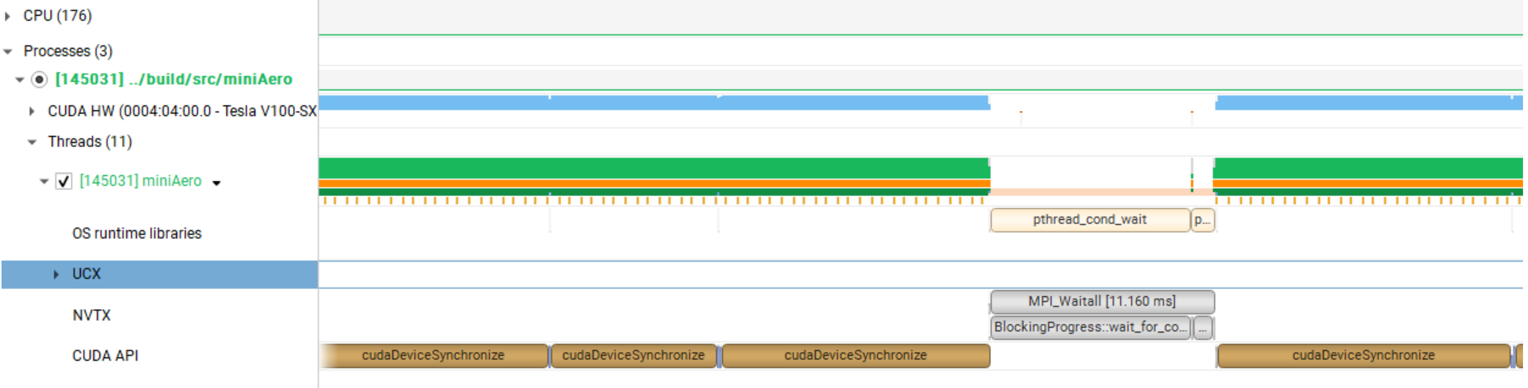
Coursework Material Development and Community/Diversity Outreach

ExaMPI – A Quick Recap

- Modern C++, Research MPI implementation
- Key features:
  - Smaller code base
  - Internals are C++ based
  - Strong progress first, with weak progress also an option
  - Most of the common MPI 3.1 functions, most of the new MPI 4.0 features
- Designed for flexible experimentation within MPI implementations
- Interactions with various collaborators, including some outside this center
- Still missing support for I/O and one-sided

ExaMPI New Features

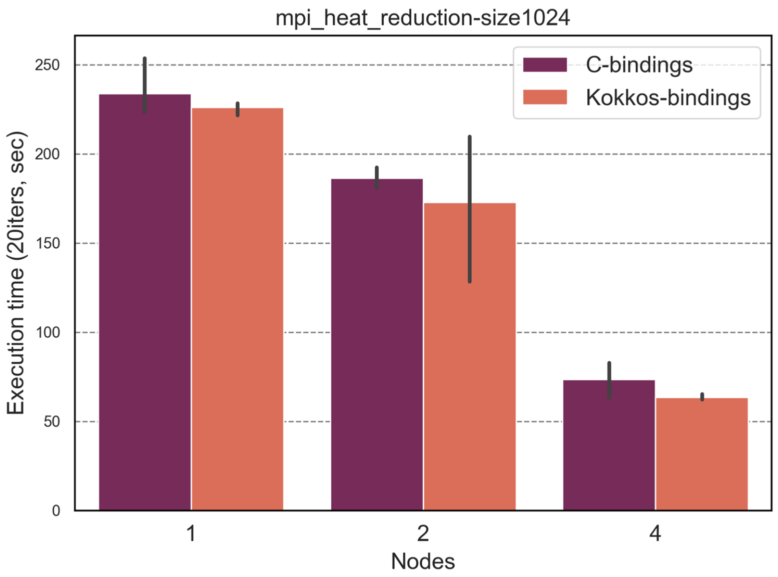
- More GPU optimizations (running with more applications)
- Caliper integration from last year
- Integration with more libraries:
  - PMIx – ExaMPI can now be ported to more systems quicker
  - UCX
- Added additional MPI functions:
  - MPI Op create, MPI Probe, MPI Scan, and other small functions
  - Now up to ~180 MPI functions



ExaMPI + Kokkos (Poster)

Led by: Evan Drake Suggs (UTC) w/ Sandia

- Paper accepted and presented at EuroMPI 2023
- Leverages ExaMPI’s experimental C++ bindings

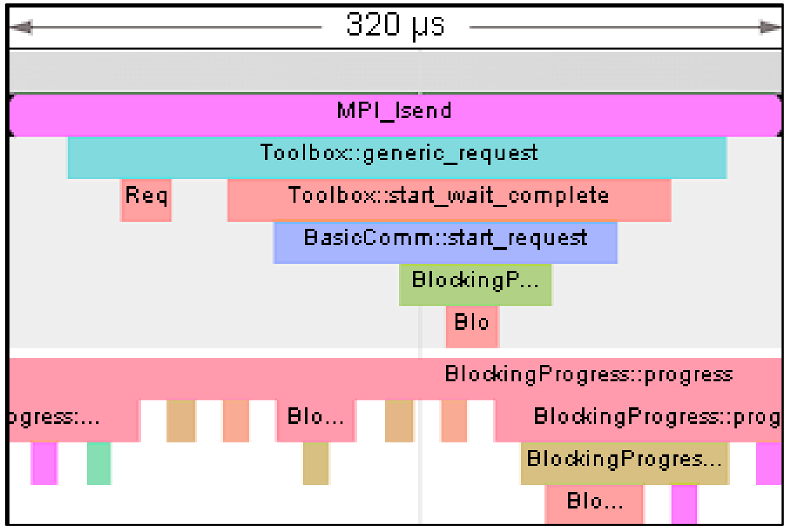


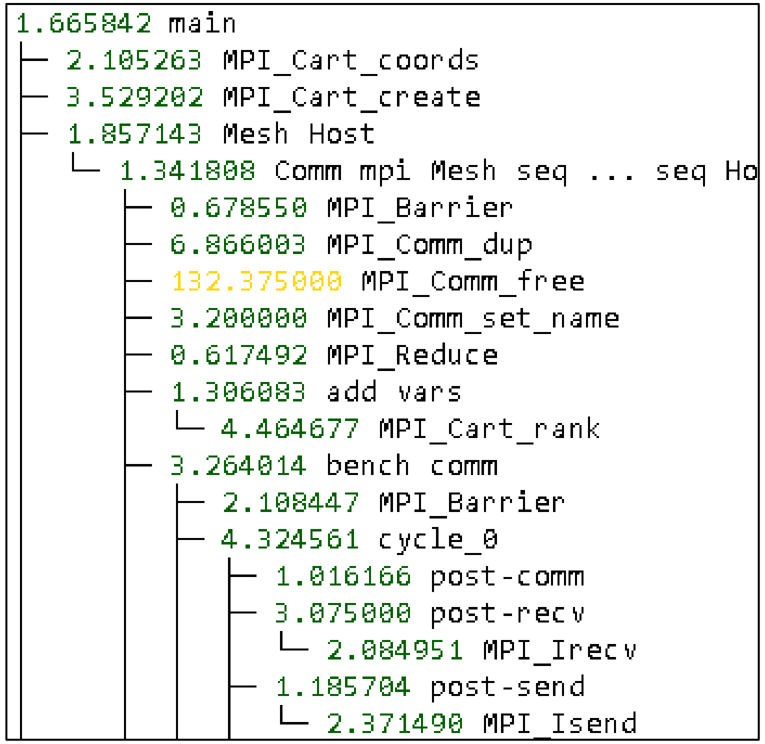
```
// old method
   int *recv_buf = (int*) malloc(n * sizeof(int));
   MPI_Recv(recv_buf, n, MPI_INT, 1, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
   Kokkos::View<int*> recv_check(recv_buf, n);
// new method
   Kokkos::View<int*> A(“New Method View”, n);
   MPI_Kokkos_Recv<Kokkos::View<int*>, int>(A, n, MPI_INT, 1, 0, MPI_COMM_WORLD);
// newer method
   Kokkos::View<int*> A(“New Method View”, n);
   MPI::Recv<Kokkos::View<int*>, int>(A, 1, 0, MPI_COMM_WORLD);
```

ExaMPI + Caliper (Poster)

Led by: Riley Shipley (UTC) w/ LLNL

- Polished work started last year
- Used to help optimize ExaMPI’s general performance
- Middle – Visualization of events in ExaMPI
- Right – Comparison of ExaMPI and Spectrum MPI running Comb (visualized with Hatchet)





```
1.665842 main
├─ 2.105263 MPI_Cart_coords
├─ 3.529202 MPI_Cart_create
├─ 1.857143 Mesh Host
│  └─ 1.341808 Comm mpi Mesh seq ... seq Ho
│     ├─ 0.678550 MPI_Barrier
│     ├─ 6.866003 MPI_Comm_dup
│     ├─ 132.375000 MPI_Comm_free
│     ├─ 3.200000 MPI_Comm_set_name
│     ├─ 0.617492 MPI_Reduce
│     ├─ 1.306083 add vars
│     │  └─ 4.464677 MPI_Cart_rank
│     ├─ 3.264014 bench comm
│     │  ├─ 2.108447 MPI_Barrier
│     │  ├─ 4.324561 cycle_0
│     │  │  ├─ 1.016166 post-comm
│     │  │  ├─ 3.075000 post-recv
│     │  │  │  └─ 2.084951 MPI_Irecv
│     │  │  ├─ 1.185704 post-send
│     │  │  │  └─ 2.371490 MPI_Isend
```

ExaMPI Next Steps

- Continuing to add support for MPI APIs on an app by app basis
- Combine with new MPI primitives experimentation, as appropriate
- Use as test bed for “HPC/MPI Class” (next presentation)
- Potential fully public release possible by center conclusion

MPI Advance

MPI Advance Recap

- A GitHub organization with a collection of MPI implementation-agnostic libraries showcasing new APIs or optimizations of current MPI APIs
- Current libraries:
  - MPIPCL (Partitioned Communication)
  - Locality Aware MPI
- Demonstrate feasibility of new ideas before acceptance by MPI Forum and adoption by MPI implementations
- Collect these ideas in one central place
- Accelerate adoption of community best practices into production applications

MPI *Advances* in the past year

- Use of MPI Advance in the community:
  - Presented at last two EuroMPI conferences
    - MPIPCL Tutorial last year
    - Locality aware paper last year
    - Short paper on overall MPI Advance collection this year
  - Hackathons
- Added support for more collective operations
- Added support for using GPU buffers (both CUDA and HIP)
- Locality-aware library integrated into Trilinos and HYPRE

Going Forward

- MPICPL to add Partitioned Collectives APIs
- Explore combination of the two libraries to support “partitioned neighborhood collectives” -> See Gerald’s poster
- Applications are already creating their own primitives
  - MPI Advance should be a common location that takes this knowledge and materializes it into useable abstractions more applications can use
  - From there, we can get community feedback and tweak and optimize
- Both libraries tie nicely into goals for remaining two years:
  - MPI-0 for trying “bottom up (lower level)” approaches
  - MPI Advance for trying “top down (application)” approaches
  - ExaMPI for tying the two together (middle layer)

Education and Outreach

Derek Schafer (dschafer1@unm.edu)

University of New Mexico
Center for Advanced Research Computing

September 28th, 2023

Outline

- Outreach & Participation in Community
- Hackathons
  - February 2023
  - September 2023
- Education Materials
  - Class development
  - Potential student assignments

Outreach

- Grace Hopper
  - Decided after feedback to look into hosting a tutorial on MPI/HPC
  - Unfortunately, dates of the conference overlapped with this meeting
  - Will aim for next year
- SIAM PP24 Mini-symposium proposal
  - Submitted a proposal
- Participation in various conferences, MPI Forum meetings

Cluster Team Support

- Funded UNM Staff/student who mentored team in the Winter Classic student cluster competition (5th place finish)
- We are looking into avenues for supporting and/or leveraging the efforts of the team
  - Provide more realistic benchmarks/tests to run
  - Having the students/mentors share their experiences in a colloquium/hackathon
  - Utilize material from their presentations in book/class
- Looking into advising a UA team for the next year

UNM HPC Competition Team

Photo courtesy of Stewart Copeland and Graphic Design by Carter Frost

Participation @ SIAM CSE 2023

- Optimizing Hypre Communication with Node Aware Parallelism
  - By: Gerald Collom
  - Mini-symposium: Krylov and Algebraic Multigrid Solvers at ExaScale
- Leveraging Modern MPI+GPU Communication Strategies
  - By: Derek Schafer
  - Mini-symposium: Recent Developments on GPU-Based Solvers in High-Performance Computing

SIAM Conference on Computational Science and Engineering (CSE23)

February 26 - March 3, 2023

RAI Congress Centre | Amsterdam, The Netherlands

CUP-ECS Seminars

- February 24th Colloquium
  - Early experience with Stream Triggered MPI on Frontier
  - *Jack Lange, Oak Ridge National Laboratory*
- February 10th Colloquium
  - MPI's Struggle With Threading
  - *Hui Zhou, Argonne National Laboratory*
- We are planning to continue seminars in the next year ☺

Hackathon – February 2023

- Goal: GPU Triggered communication
- Looked at two main libraries:
    - MPIX Streams (Argonne)
    - Cray's MPIX Streams
- Explored CUDA, HIP APIs to get familiar with current capabilities
- Had students work in Pulse benchmark and test out implementations
- Several lab personnel attended and participated in activities in addition to students from all three universities

Hackathon – September 2023

- Apply the refined center goals to address one of the “thorns” of MPI: Using graph topologies in communicators
- Technical Goals:
  1. Decouple topologies from MPI Communicators so that graph topology creation is more efficient. This in turn would make creating communicators with that topology more efficient.
  2. Extend the functionality of topologies to support things like “flipping the direction” of an existing topology.
- Students from all three universities participated

Hackathon Results

- Technical results were mentioned earlier today:
  - GPU triggering and current partitioning abstractions are tricky – hard to know what is best for applications
  - Initial prototype topology object is implemented in MPI Advance, and into HYPRE – timing results in progress
- Continuing to train students, participants on HPC research concepts as well as fundamental systems programming
- Help students understand and practice the full scope of center's activities
- Informs how center benchmarks should work, what paradigms they should be capable of performing

HPC Class Development

- Current timeline:
  - Collecting materials from various parallel classes taught by PIs
  - Finish develop of class materials and assignments by end of calendar year
  - Aim to teach first version in Spring 2024
- Class assignments (under refinement):
  1. Develop a basic P2P MPI implementation
  2. Add in non-blocking features, (weak) progress engine
  3. Add in strong(er) progress engine, collective functions
  - Assignments will likely use ExaMPI's runtime, standard C socket networking
  - Graduate students could potentially use UCX/Libfabric/Verbs networking instead
- Incorporate course materials into the open-source book in development

Thank you!