Remaining Work to be Done: Summer to Fall 2025

Presenter: Prof. Patrick Bridges

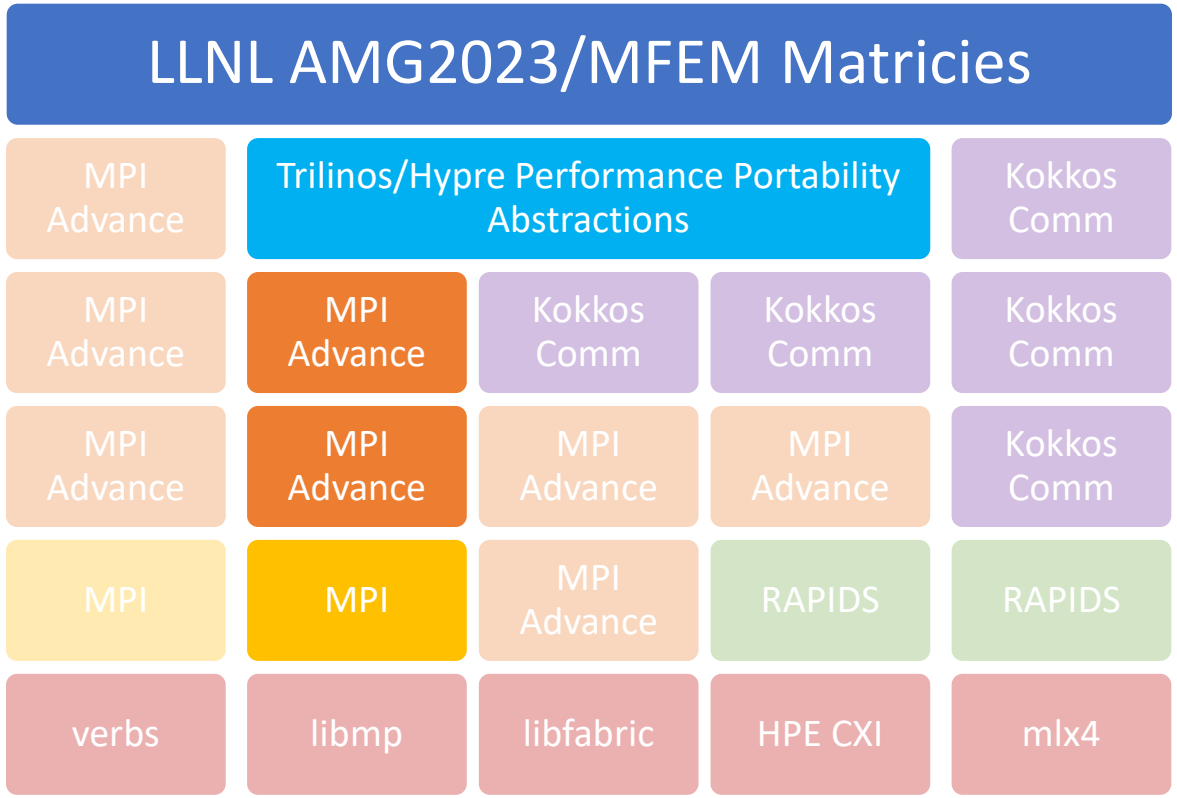
Remaining Work: Integration, Assessment, Publication, and Release Engineering

- Assessment and Integration
  - Finish integrated examples of APIs, tools, and workflows
  - Assess/demonstrate impact of new APIs, tools and workflows on more production codes
- Publication and Dissemination
  - Preparing two to three major (SC/IPDPS/HPDC) publications
  - Preparing workshops and tutorial materials demonstrating tools and techniques
- Release Engineering
  - MPI Advance and Beatnik
  - Benchmark suite documentation/collection
  - PR submissions and integration to key collaborator codebases

Integration and Assessment (1)

Irregular Neighbor Optimization

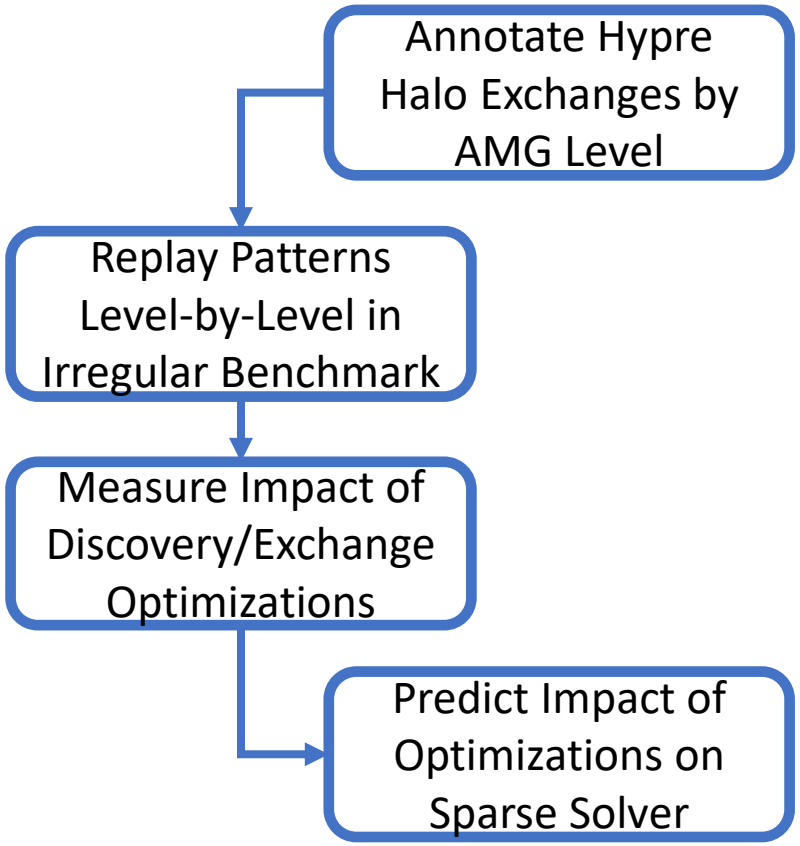
- Integrate multiple optimizations into production sparse solver
  - Focus on Hypre (MFEM/AMG2023)
  - Dynamic topology abstractions
  - Neighbor discovery abstractions
  - Locality-aware neighbor exchange
- Optimizes abstractions at the heart of multiple production codes



Integration and Assessment (2)

Communication Performance Prediction Workflow

- Demonstrate the ability predict the impact of irregular communication optimization on sparse matrix solvers
- Complements actual integration into Hypre solvers
- Informs efforts to potentially integrate into Trilinos and xRage

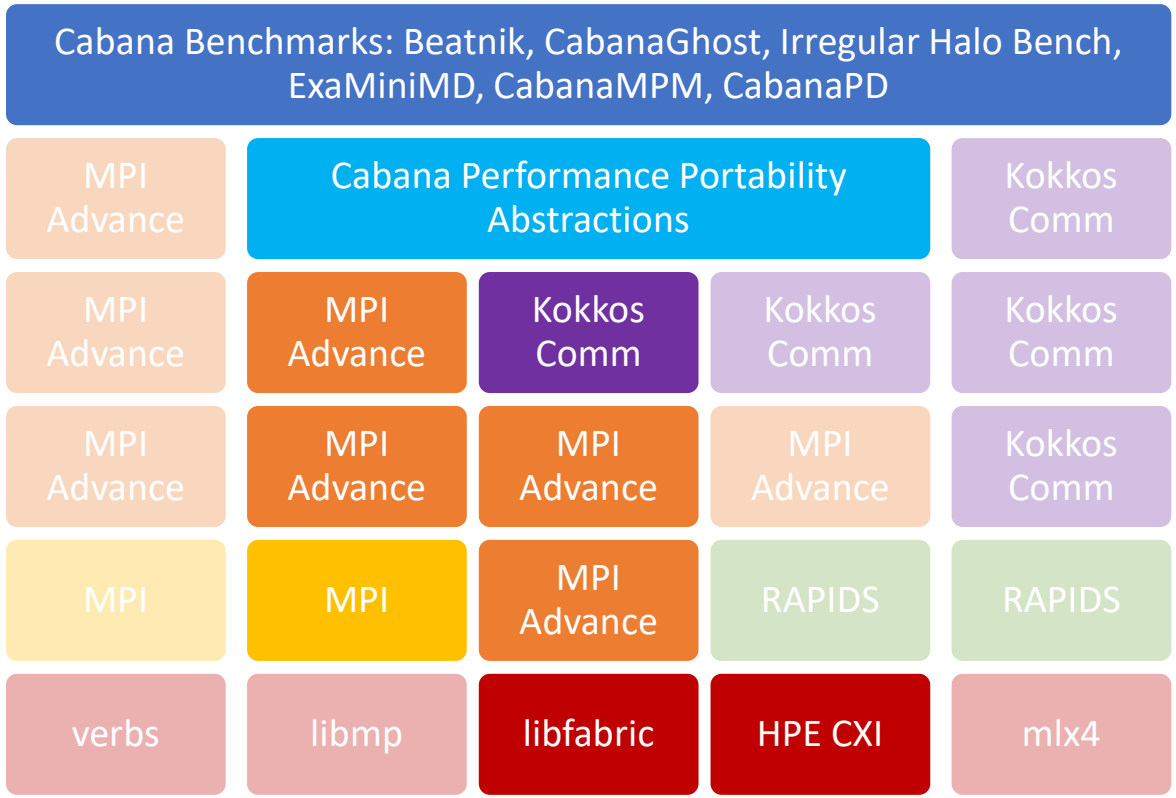


Integration and Assessment (3)

Stream Triggering Abstraction Co-Design

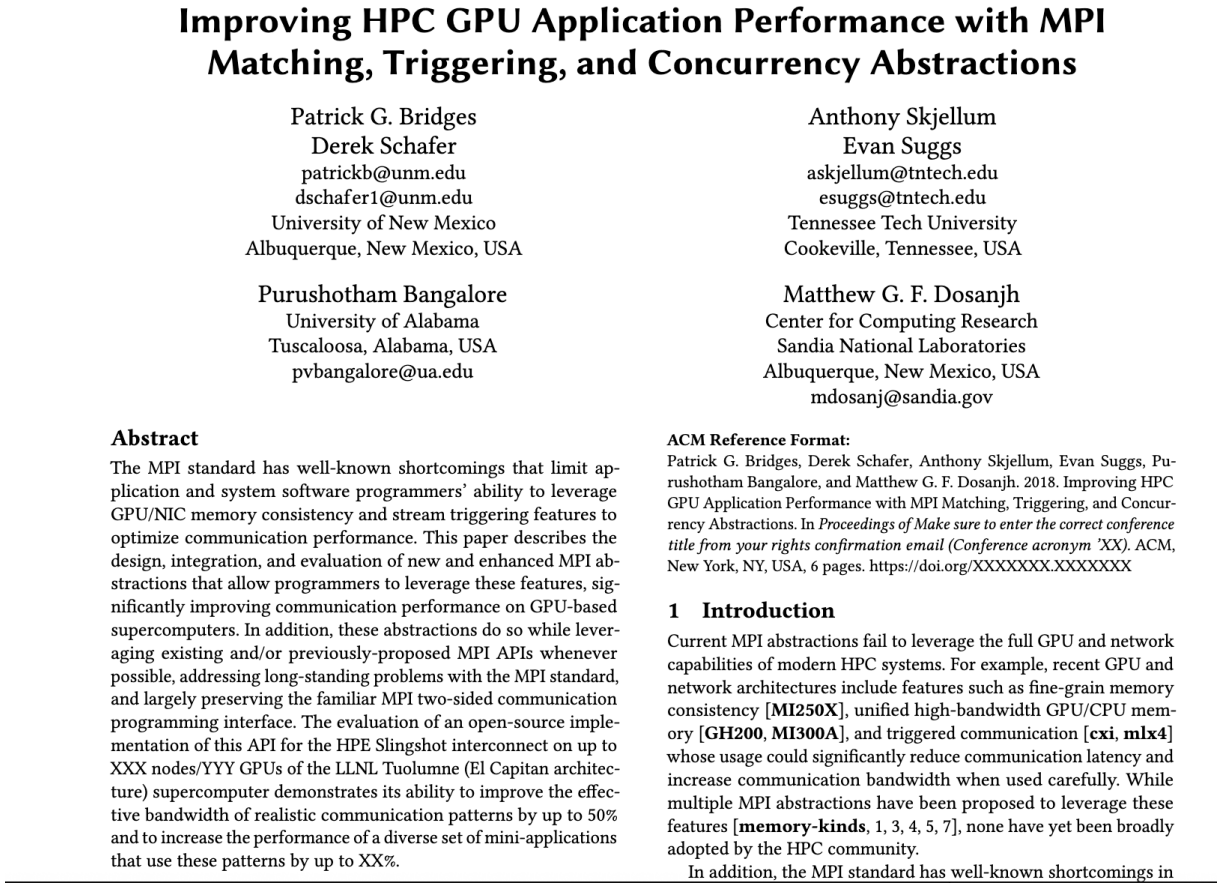
Integrate Cabana, Kokkos Comm, MPI Advance, MPI, and Slingshot APIs

- Provide full co-designed state-of-the-art communication stack
- Assess in strong scaling cases
  - Regular halo benchmarks
  - SpMVs (small global reductions)
  - FFTs
- Tuolumne is the ideal target
- Demonstrates new abstractions requiring significant standardization/integration



Publication of Key Results

- Targeting capstone publications in major HPC publications
- Focusing on SC, IPDPS, HPDC
- One paper for each of the prior listed integrations
  - Stream Triggering
  - Communication Performance Prediction
  - Irregular Exchange Optimization



Improving HPC GPU Application Performance with MPI Matching, Triggering, and Concurrency Abstractions

Patrick G. Bridges
Derek Schafer
patrickb@unm.edu
dschafer1@unm.edu
University of New Mexico
Albuquerque, New Mexico, USA

Anthony Skjellum
Evan Suggs
askjellum@tntech.edu
esuggs@tntech.edu
Tennessee Tech University
Cookeville, Tennessee, USA

Purushotham Bangalore
University of Alabama
Tuscaloosa, Alabama, USA
pvbangalore@ua.edu

Matthew G. F. Dosanjh
Center for Computing Research
Sandia National Laboratories
Albuquerque, New Mexico, USA
mdosanj@sandia.gov

Abstract

The MPI standard has well-known shortcomings that limit application and system software programmers' ability to leverage GPU/NIC memory consistency and stream triggering features to optimize communication performance. This paper describes the design, integration, and evaluation of new and enhanced MPI abstractions that allow programmers to leverage these features, significantly improving communication performance on GPU-based supercomputers. In addition, these abstractions do so while leveraging existing and/or previously-proposed MPI APIs whenever possible, addressing long-standing problems with the MPI standard, and largely preserving the familiar MPI two-sided communication programming interface. The evaluation of an open-source implementation of this API for the HPE Slingshot interconnect on up to XXX nodes/YYY GPUs of the LLNL Tuolumne (El Capitan architecture) supercomputer demonstrates its ability to improve the effective bandwidth of realistic communication patterns by up to 50% and to increase the performance of a diverse set of mini-applications that use these patterns by up to XX%.

ACM Reference Format:
Patrick G. Bridges, Derek Schafer, Anthony Skjellum, Evan Suggs, Purushotham Bangalore, and Matthew G. F. Dosanjh. 2018. Improving HPC GPU Application Performance with MPI Matching, Triggering, and Concurrency Abstractions. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Current MPI abstractions fail to leverage the full GPU and network capabilities of modern HPC systems. For example, recent GPU and network architectures include features such as fine-grain memory consistency [**MI250X**], unified high-bandwidth GPU/CPU memory [**GH200**, **MI300A**], and triggered communication [**cxi**, **mlx4**] whose usage could significantly reduce communication latency and increase communication bandwidth when used carefully. While multiple MPI abstractions have been proposed to leverage these features [**memory-kinds**, 1, 3, 4, 5, 7], none have yet been broadly adopted by the HPC community.

In addition, the MPI standard has well-known shortcomings in

Workshops and Tutorials

- Draft materials for tutorials educating community how to use the abstractions and tools described in the previous slides
  - Using MPI Advance to improve application performance
  - Leveraging stream-triggered communication on modern hardware
  - Analyzing and leveraging communication optimization opportunities in HPC applications
- Initial offering to collaborating research groups at national labs
- Offer as paid tutorials a conferences once initial offerings tested

Software Release Engineering

- Hired additional research staff member with which we have previously collaborated for final software release pushes in multiple packages
- MPI Advance: CI Testcases, CMake and Spack Packaging, Documentation and associated training materials
- Cabana: Finalization of pull requests for multiple communication backends, new and optimized communication abstractions (collector, stream triggering)
- Caliper and Benchpark: Integration of histogram-based communication pattern collection into Caliper production and Benchpark benchmarks