Overview of Research Activities: Fall 2023 – Spring 2025

Presenter: Prof. Patrick Bridges

Recommendations from 2023 Annual Review

Software use cases being explored

- Previous Mini/Proxy Applications
  - **CLAMR**: LANL cell-based adaptive mesh refinement (AMR) mini-app, using the **L7** communication framework
  - **MiniAero**: Manetevo mini-application that does Navier-Stokes equations for explicit unstructured finite volumes
  - **Comb**: LLNL communication performance benchmarking tool
- Current Mini/Proxy Applications:
  - **Hermes:** New global spatial sorting benchmark
  - **MiniGhost/CabanaGhost:** Existing and in-development stream-based regular halo exchange benchmark
  - **Beatnik:** New fluid interface benchmark to exercise FFT and mesh/particle remap communications
  - **CabanaMD/CabanaPD/CabanaMPM:** Cabana particle/mesh molecular dynamics, peridynamics, and material point benchmarks
- Current Applications/Libraries
  - **xRAGE**: LANL Eulerian radiation/hydrodynamics code
  - ***hypre***: LLNL library of preconditioners and solvers, e.g. multigrid methods for the large, sparse linear systems of equations
  - **Kripke**: LLNL neutron transport proxy
  - **Parthenon**: LANL performance portable block-structured adaptive mesh refinement framework
  - **HOSS:** LANL hybrid multi-physics software package using a range of element-based methods
- Upcoming/In-progress Mini/Full Applications: Sandia **EMPIRE/SPARC** via MiniEM/MueLu and Ifpack2 in Trilinos

Define any new MPI abstraction terms

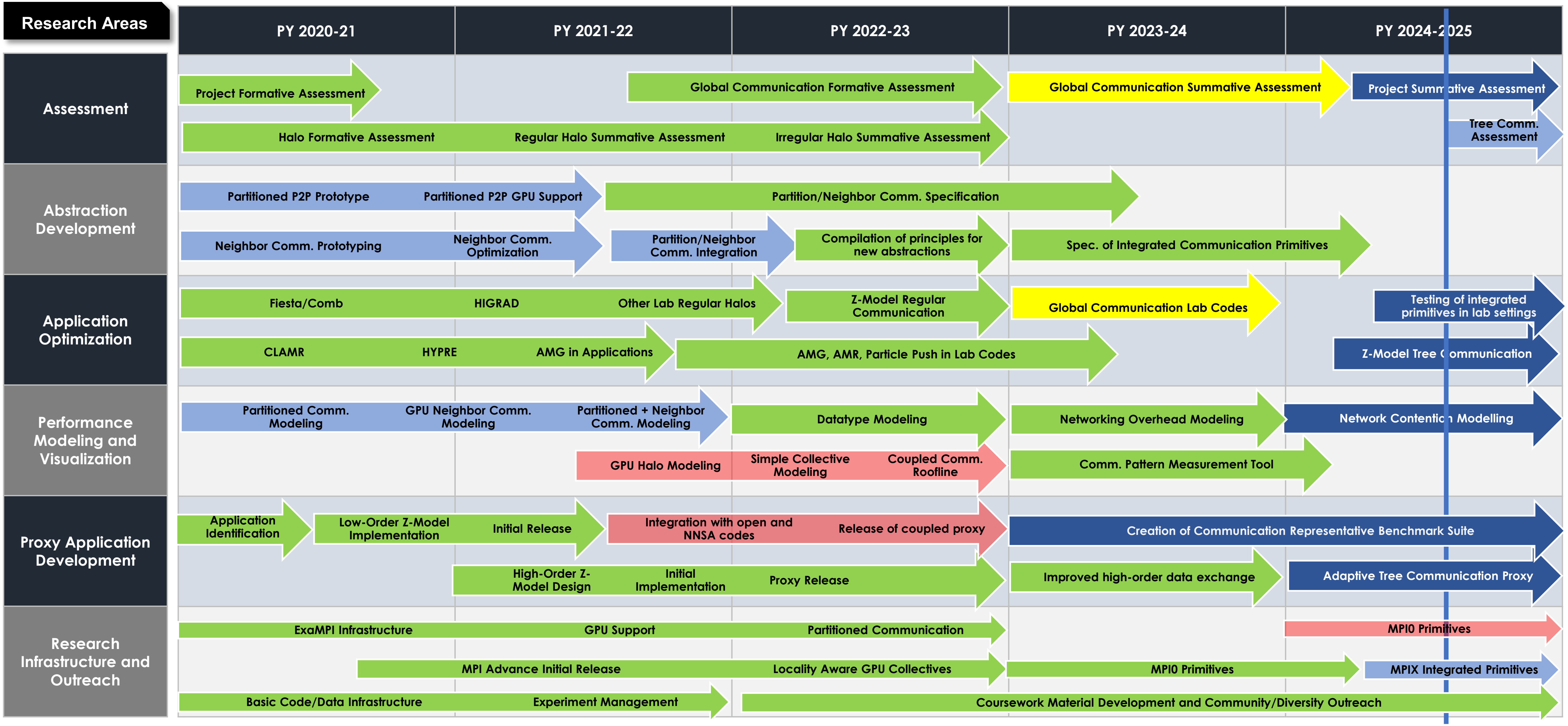
- **ExaMPI**: Our C++ research MPI implementation
- **Beyond MPI:** Application-oriented abstractions that leverage our findings, unconstrained by legacy MPI. Current focus in Kokkos, Cabana, and Trilinos
- **MPI Advance:** Application-oriented abstractions that extend and push the frontier of existing abstractions while respecting/extending MPI legacy interfaces
- **MPI0 (now RAPIDS):** New low-level communication primitives for use by library writers for building new abstractions

Other recommendations

- Modify the software stack slide to convey your evolution in thinking about MPI abstraction layers, and what is being accomplished with each abstraction – **done**, see later slides
- Provide a table of proxy apps, e.g. Beatnik, being used to test various abstractions – in progress
  - Regular halos: Comb, MiniAero, CabanaMPM, CabanaPD, MiniGHOST/CabanaGHost
  - Irregular halos: AMG2023, HYPRE SuiteSparse, MiniEM, Trilinos Ifpack 2 solve, New Irregular exchange benchmark
  - Global exchanges: Beatnik (FFT, particle/mesh redistribution), Hermes (new global spatial sorting benchmark)
- Develop MPI communication pattern extraction tools – **done**, see later slides

Current Roadmap and Changes

Updated 5-year Project Roadmap



Changes/Challenges/Status

- Personnel
  - Jason Stewart hired as full-time staff (UNM)
  - Thomas Hines left project (UA)
  - Hiring full time staff at UA for final software development integration (MPIAdvance)
  - Rebalancing/rescheduling work between UA and UNM due to personnel changes
  - Multiple lab internships completed (Evelyn, Nicole, Grace, Nick, Gerald, Jackson, Mike)
- Vendor interaction
  - NDAs executed with AMD, HPE
  - Speed of interaction/meaningful response has improved, assisting abstraction development
- Final software integration focusing on well-defined through-lines/libraries to demonstrate value of fundamental research results
  - MPI Advance, Cabana, Kokkos abstractions for final integration
  - Integrating into Trilinos, Hypre, HOSS, Parthenon, xRage, various benchmarks
  - Examining options for follow-on translational/applied research

High-level Research Goals/Accomplishments Fall 2023 to Spring 2025

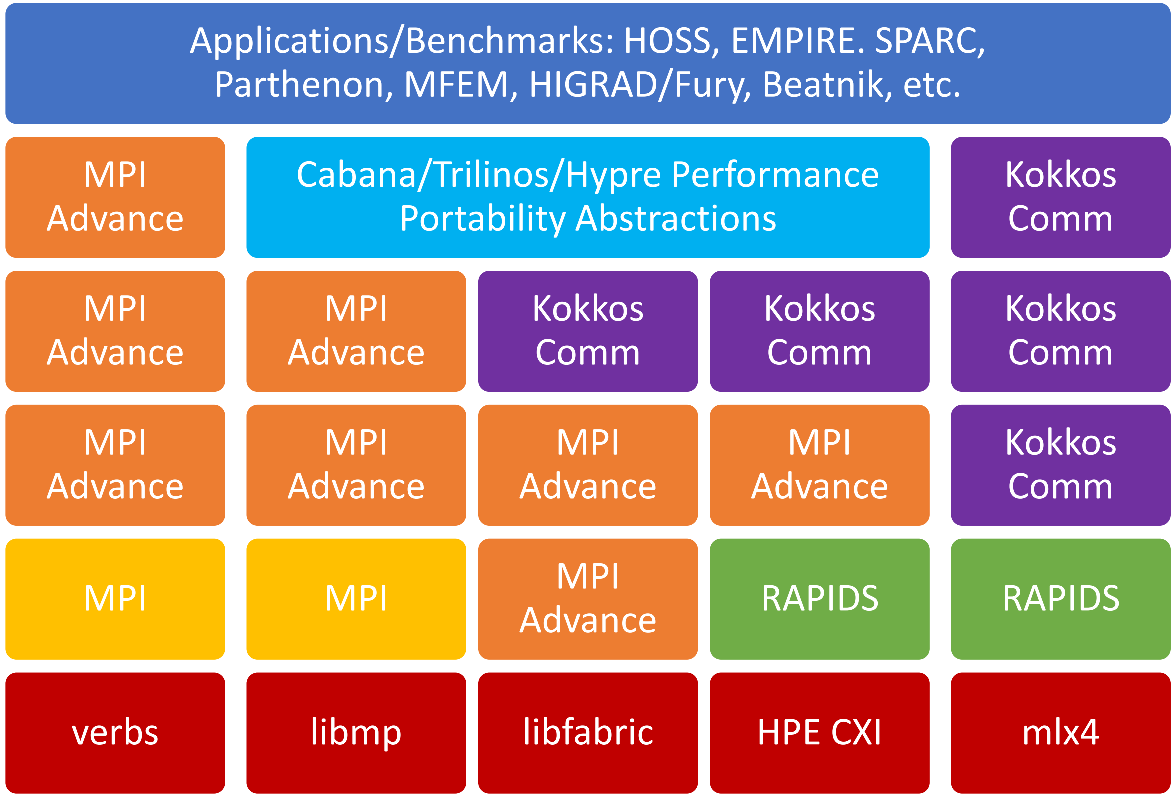
Overarching Research Questions

Original Goal: Research, demonstrate and deploy better communication abstractions that make NNSA mission applications faster, more predictable, and easier to write

As the center has progressed, we refined to two research questions:

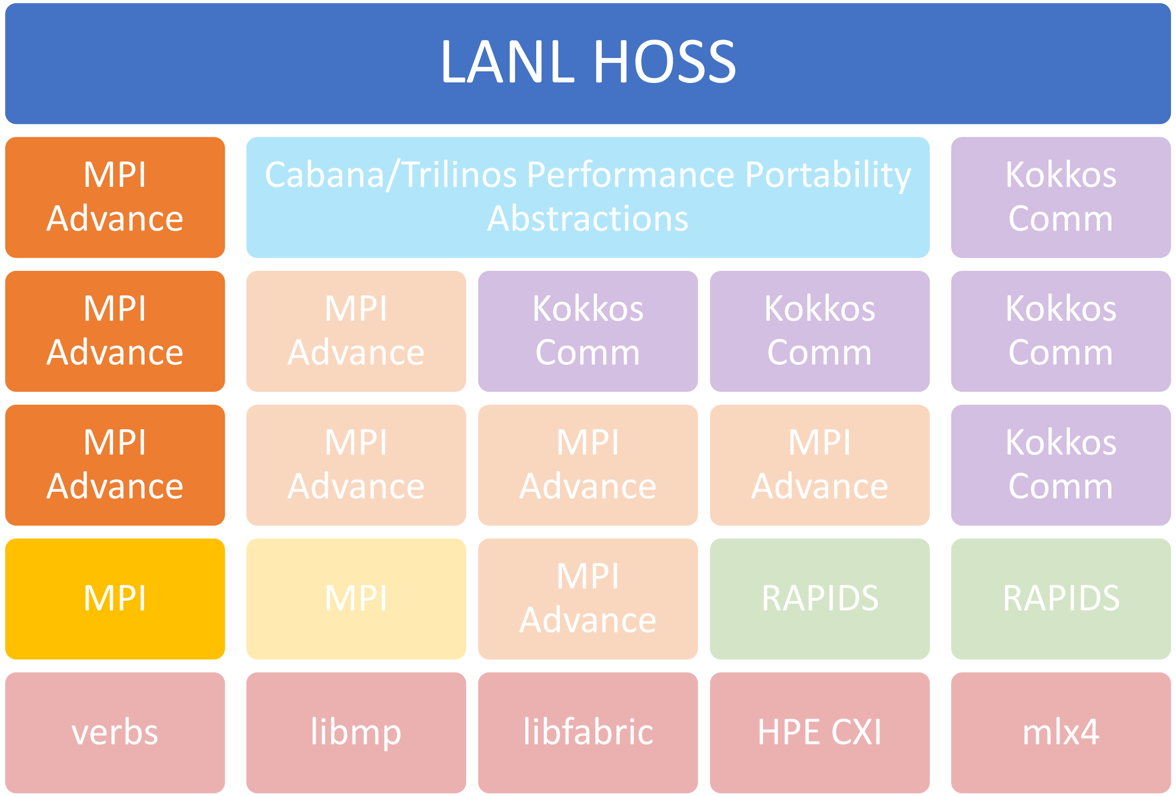
1. What novel communication abstractions and associated optimizations are needed for modern HPC applications/systems?
2. How can we model, predict, and assess the potential impact of these primitives on production applications? (Years 3-5)

Q1: Communication Abstractions Needed



- Goal: to develop new communication abstractions for DOE applications and libraries
- Development at each level is driven by careful assessment, benchmarking, and modeling
- Through the first two and a half years, focus was mostly on Advance-level primitives
- Shifted focus to include performance portability frameworks, Kokkos Comm, and RAPIDS abstractions in the last two years

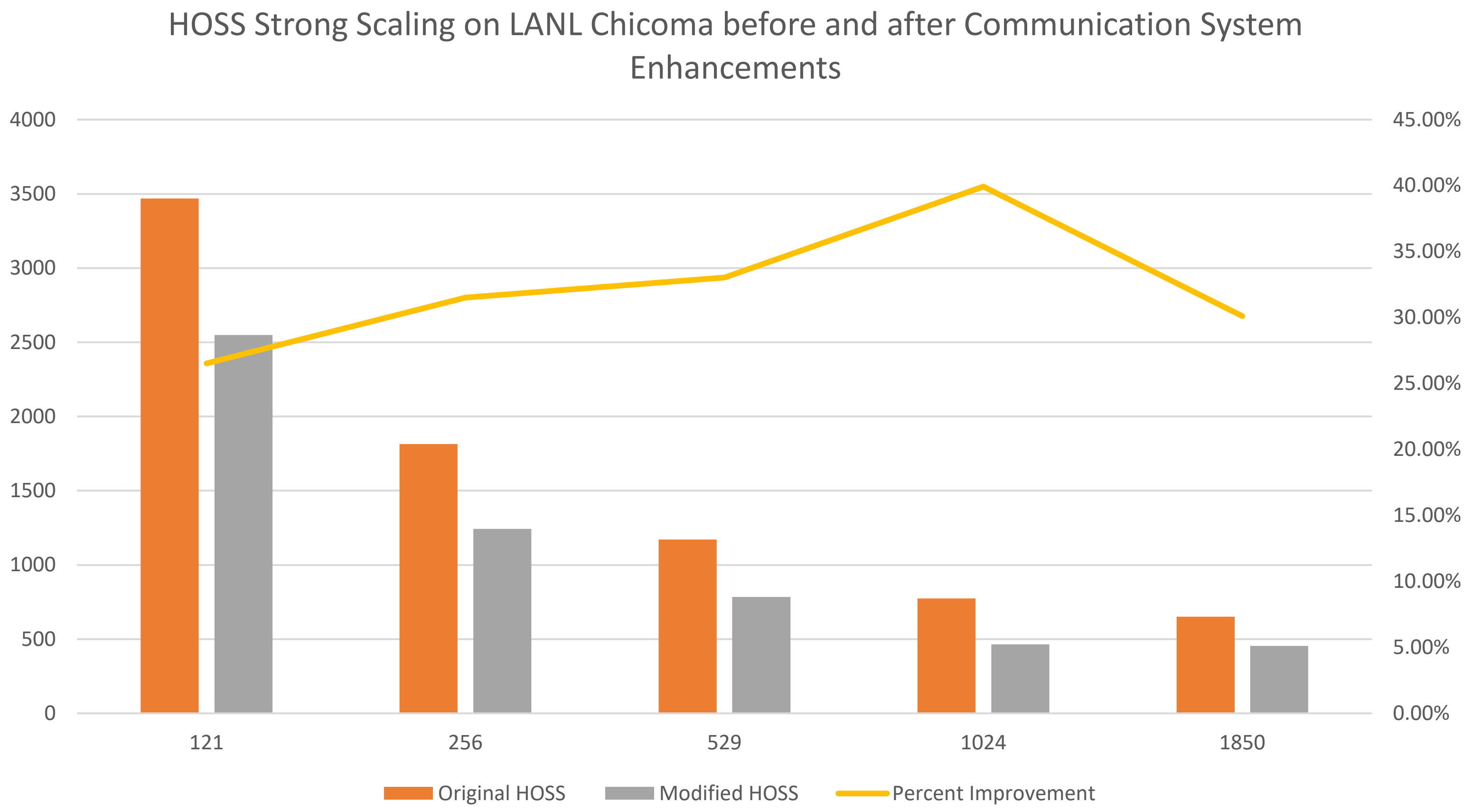
Example 1: HOSS on Modern Networks



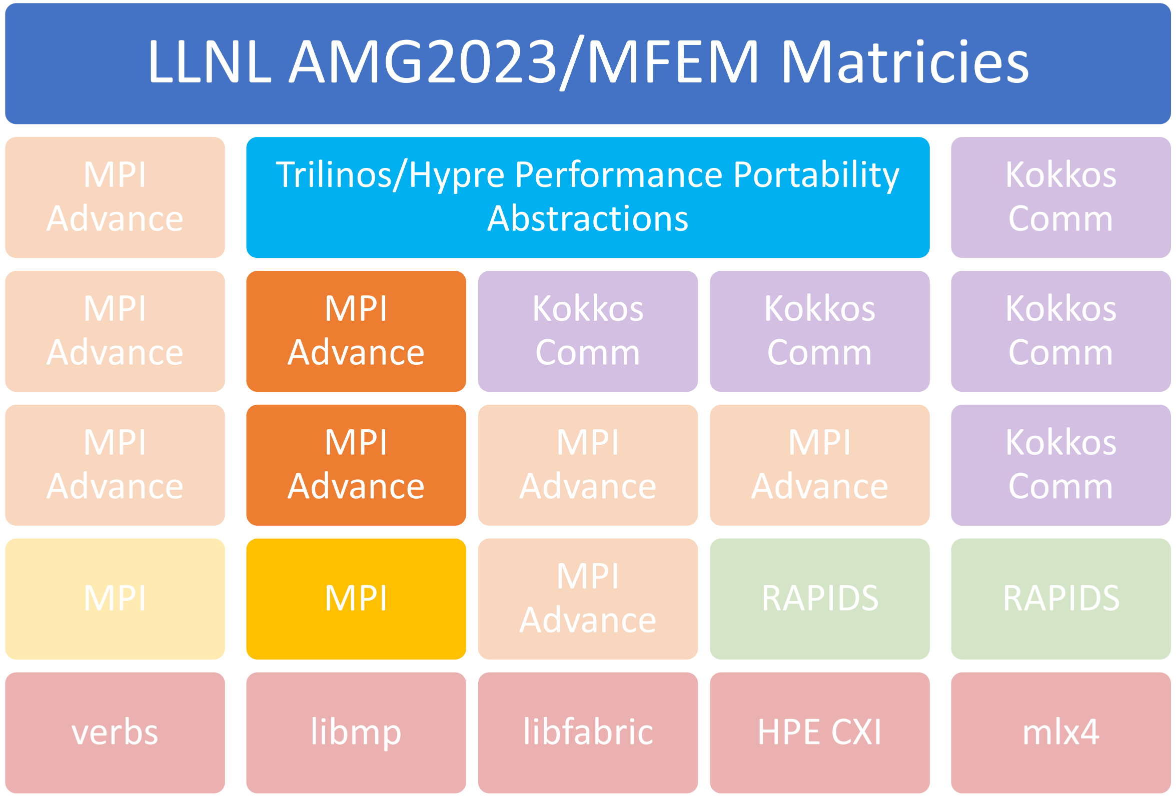
- As part of the LANL GPU porting effort, we examined HOSS communication handling
- HOSS has a custom communication scheduling engine to manage complex FE object pack/exchange/unpack with highly varied numbers of objects per node
- Engine was designed to support complex neighbor exchanges for a wide range distribution of communication patterns/loads
- Changed network/CPU balance and improved HOSS load balancing resulted in >50% CPU idling idling during communications

Example 1: HOSS on Modern Networks

- Legacy C/MPI code so focused on MPI/MPI Advance Primitives
- Changed HOSS to a bulk-synchronous packing plus neighbor collectives for data exchange
- 25-50% improved strong scaling on LANL-provided test problem (bore hole fracture)
- **Deployed in production at LANL**
- **Collaboration with Ryan Marshall, former CUP-ECS Postdoc and Current LANL Technical Staff Member**

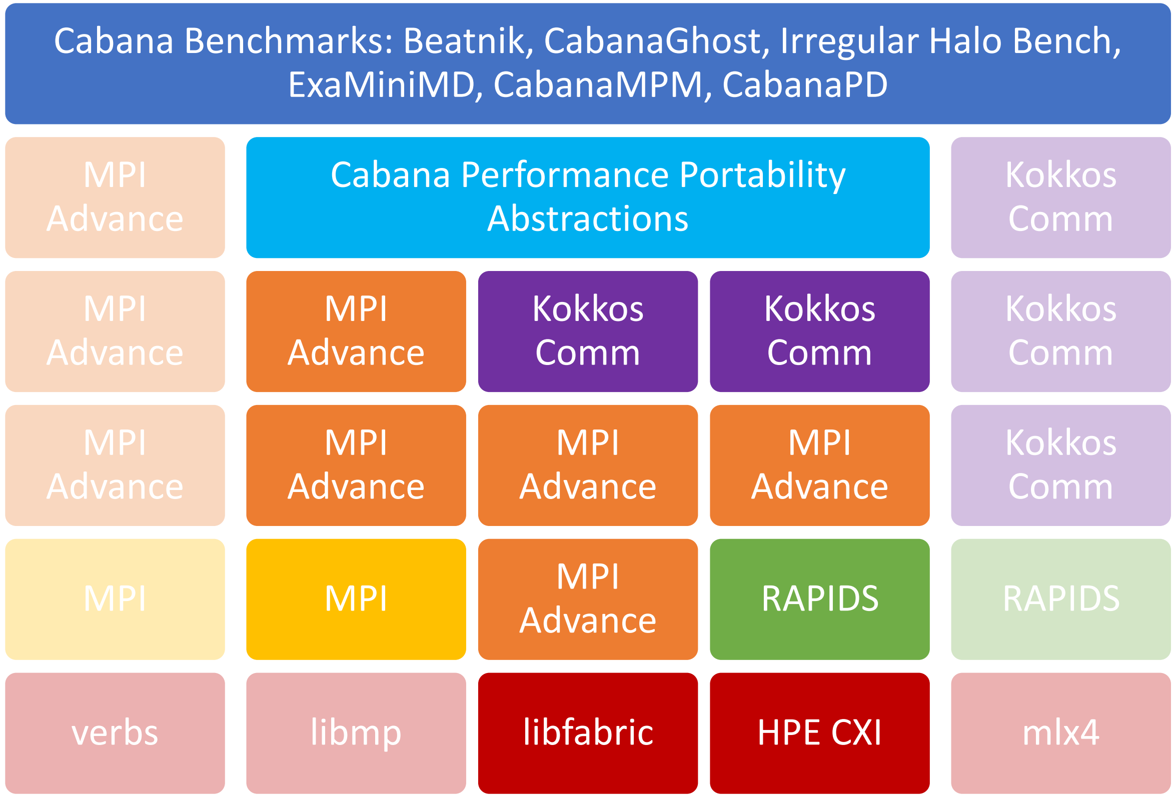


Example 2: Hypre Communication



- Examining broad range of communication optimizations in Hypre, with higher-level communication interfaces
  - Receiver-driven neighbor discovery
  - Locality-aware neighbor exchange
  - Early communication/computation via row/column partitioning
- Again primarily focused on breadth of APIs and optimizations to make sure we cover the needs of a wide range of applications
- But testing on state-of-the-art systems and problem
- Similar approach planned for Trilinos this summer with a focus on matricies from EMPIRE/MiniEM/MueLU and SPARC/ifpack2

Example 3: Stream-Triggered C++ Communication Primitives



- Goal: Examine impact of deep application/network co-design on diverse communication performance
- Cabana benchmarks (some developed in-house) push a wide range of local and global communication patterns
- Hand developed stream-triggered interfaces for Cray Slingshot NIC (and general CUDA/MPI systems)
- Targeting both Kokkos Comm and MPI primitives
- Hope to extend/integrate via later development for Trilinos, other backends

Major API Development and Deployment Thrusts

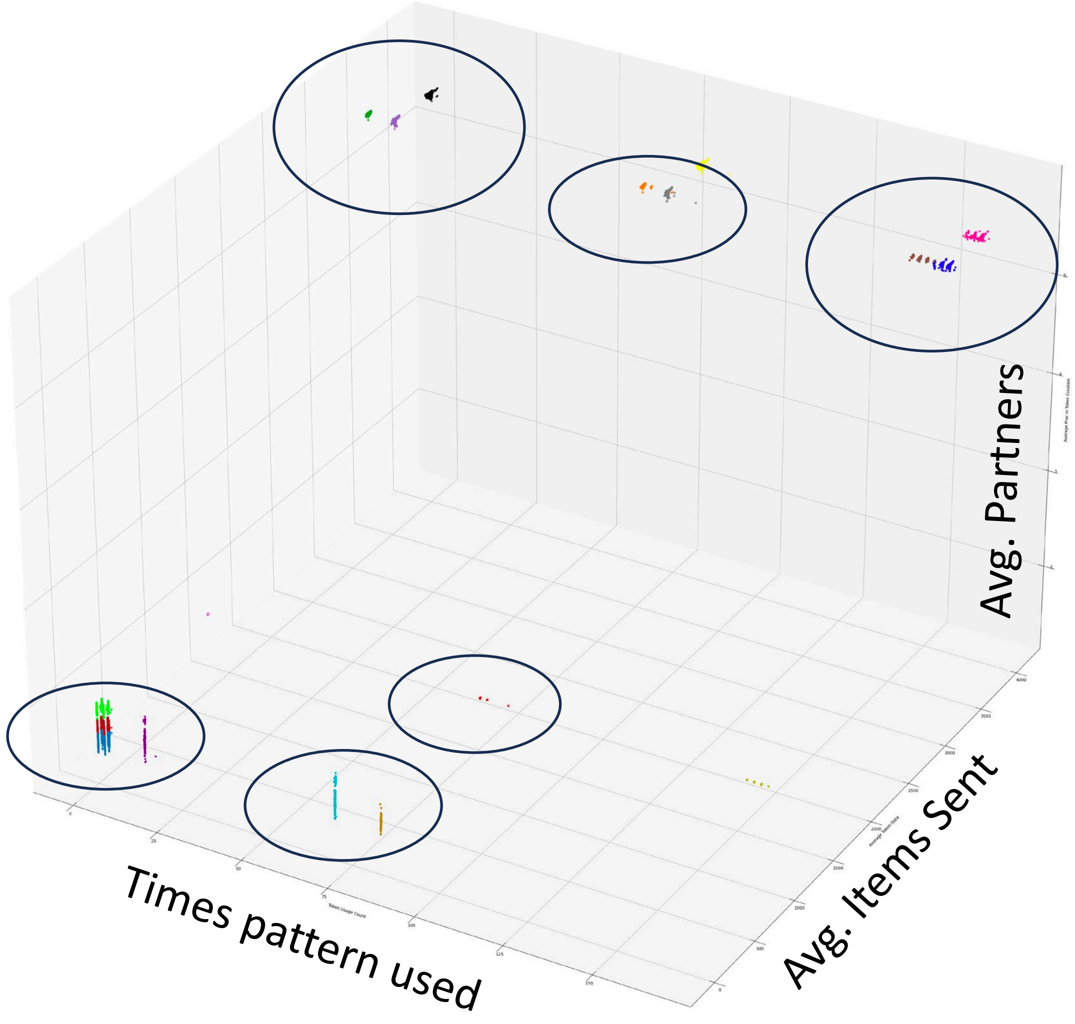
- C++ Portability Library Primitives:
  - New C++ Communication interfaces for Kokkos, Cabana and Trilinos
  - Expanded Cabana primitives and optimizations to integrate in Trilinos, xRage, etc.
- MPI Advance
  - MPI Stream Triggering Abstractions: Cabana and Kokkos
  - Optimized Neighbor Discovery/Comm.: Cabana, Hypre, HOSS, xRage, and Trilinos
  - Optimized Global Communication: Global sorts, Mesh/Particle Redistribution, FFTs
- RAPIDS Low-level Primitives: Abstractions for modern architectures
  - Current focus on persistent channel that amortize away dominant costs in current networks
  - Complements other research (LCI focuses on many-task systems, MVAPICH on MPI)

Q2: Which Abstractions and Optimizations, and Where and When?

- Communication studies are painful!
- Hard to use real applications and input decks for communication development
  - Require significant expertise to build, configure, run, and scale
  - Wide range of communication techniques and behaviors make it hard to isolate, understand, and optimize specific patterns
  - Communication tightly enmeshed with complex compute makes it difficult to try out novel communication abstractions
- Creating representative communication proxy applications is hard
  - Communication-representative mini-applications work in this area: LAMMPS/miniMD, HACC/SWFFT (Aaziz et al., 2018), CTH/miniAMR (Aaziz et al., 2019), others
  - New miniapps with interesting input decks if you know where to look (ATS 5 acquisition benchmarks): MiniEM, AMG2023, Block AMR and Neutron Transport benchmarks, etc

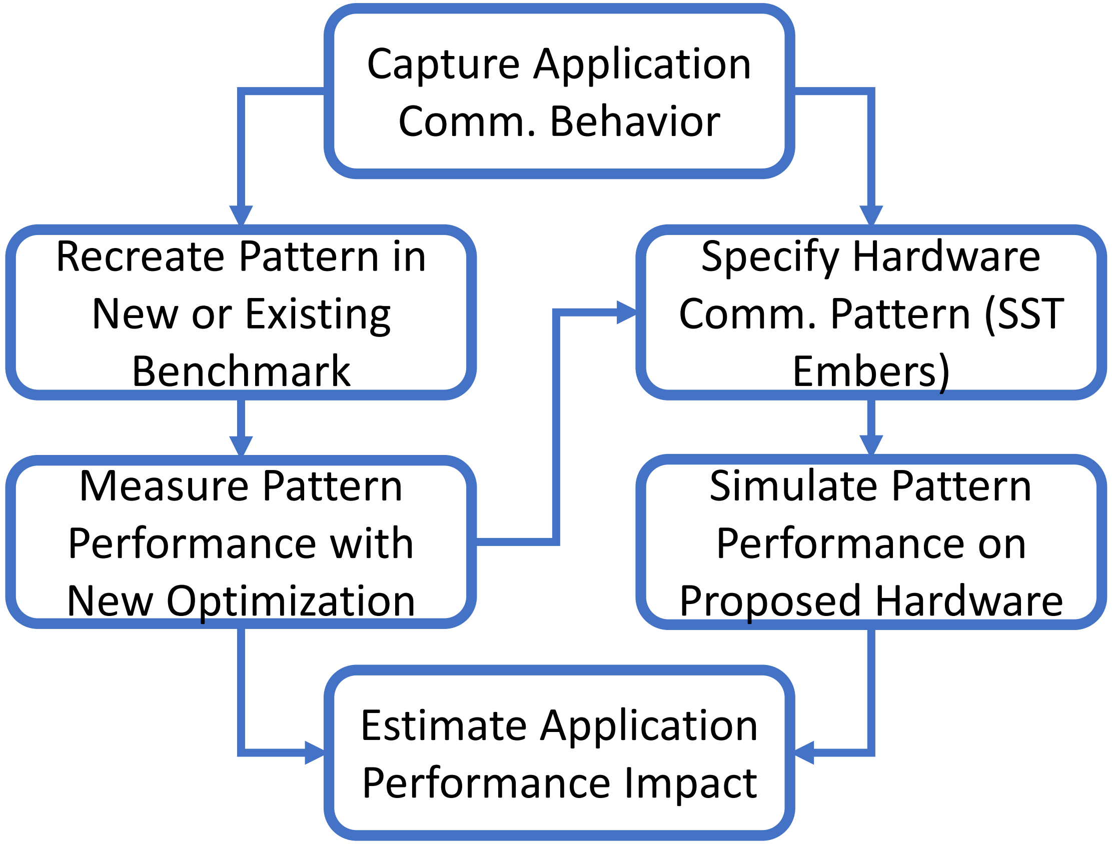
Q2: Which Abstractions and Optimizations, and Where and When?

- Y2 studies prototyped key capability
  - Capture real communication patterns from production applications (xRage)
  - Replay communication pattern
- Goal: Develop a workflow around this
  - Profiling tools generally capture communication patterns
  - Set of representative algorithm and pattern/statistics replay benchmarks
  - Tools and analyses to predict performance impact of changes on communication patterns and applications



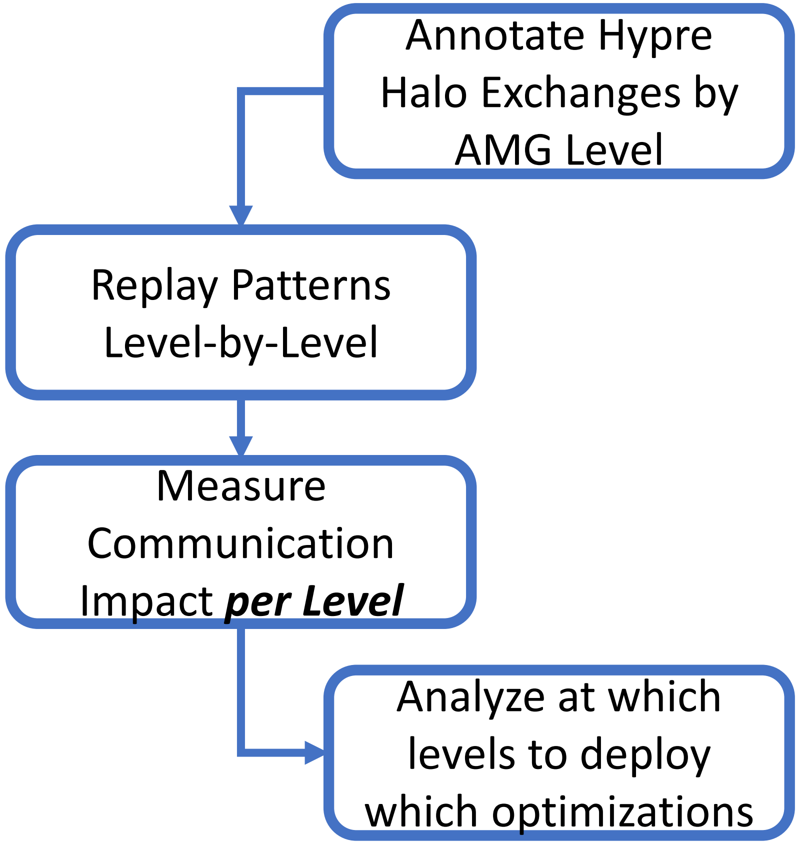
Developing and Deploying this Workflow to Drive Communication Abstraction Design

- Capturing Application Comm. Behavior
  - Irregular Halo Exchanges: Comm. Pattern Annotations for Caliper
  - New Comm. Benchmarks for more complex communication motifs
- Creating interfaces to generate SST Embers from captured patterns to simulate impact of hardware change
- Developing tools to support this workflow
  - Adopted Benchpark for experiment management
  - Creating Hatchet/Python analyses to make analyses done in xRage available generally
  - Working on performance estimation



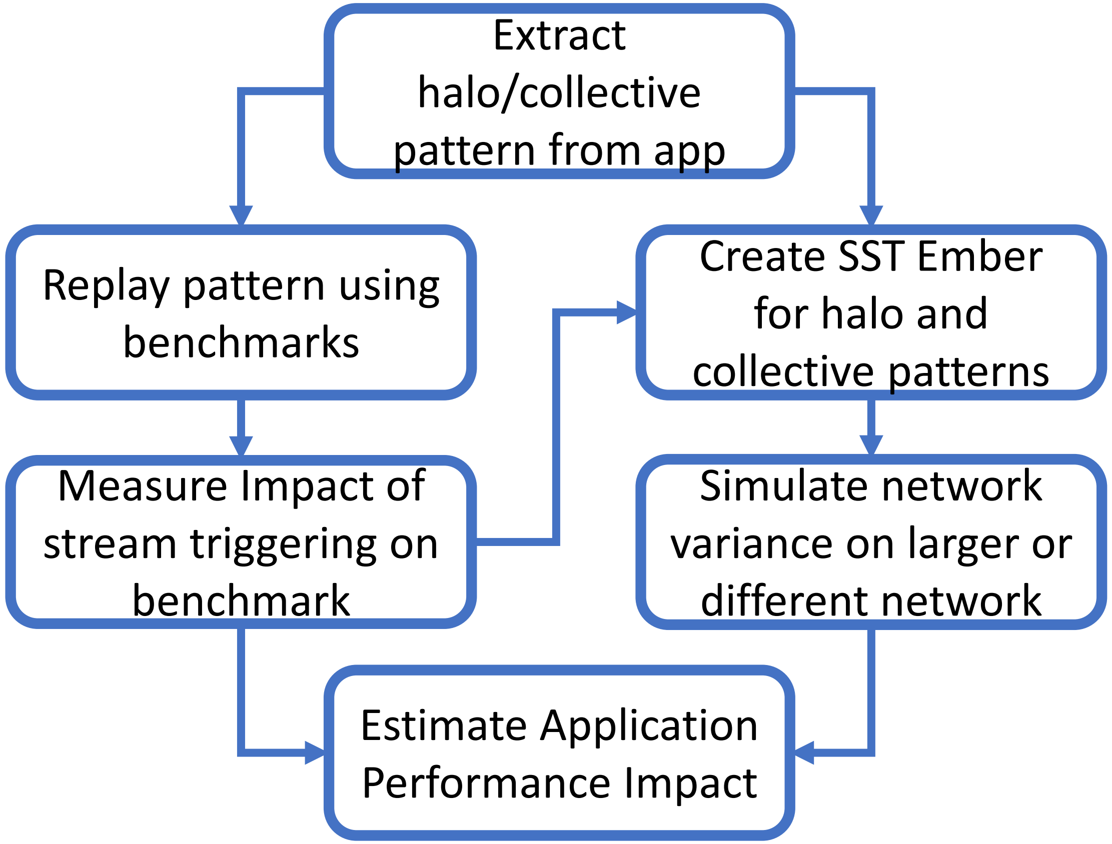
Example 1: Hypre BoomerAMG Analyses

- Optimization of Hypre highly dependent on problem structure
- Fine-grain communication patterns annotation enables careful optimizations targeting
- Motivates research on broad set of optimizations described earlier
  - Optimized Neighbor discovery
  - Locality-aware Neighbor Exchanges
  - Fine-grain data movement
- Hard-managed instance of this workflow in use in center today



Example 2: Stream Triggering Analyses

- Impact of effective stream triggering on applications still unclear
- Promising candidates when strong scaling
  - Halo Pack/send/unpack (MueLU, Ifpack2)
  - Latency-bound collectives (AllReduce, FFT)
- Can also depend on NIC and network performance (network variance, hardware vs. software collectives)
- Currently developing this workflow for in-depth stream-triggering exploration



Intentionally not spoiling all the details!

- Center student, research staff, and PIs have focused on these questions and approaches for the last two years
- Two student/research staff-focused sessions of lightning talks and posters presenting the detailed research toward these questions
  - Morning: Abstraction Development and Optimization Session
  - Afternoon: Measurement, Modeling, and Assessment Session
- Wrap up at the end of the day:
  - Education and Outreach
  - Remaining Research Tasks
  - Overall Contributions over life of the center