FIC: Center for Understandable, Performant Exascale Communication Systems

Prof. Patrick G. Bridges, University of New Mexico

Prof. Purushotham Bangalore, University of Alabama at Birmingham

Prof. Anthony Skjellum, University of Tennessee at Chattanooga

August 18, 2020

Center Overview

- Research Focus: Optimized, performance-transparent communication systems for NNSA exascale applications
- Goal: Realize revolutionary communication and runtime systems for emerging applications and architectures
  - Efficiency: Fully leverage system resources–heterogeneous processors, network offload, abundant parallelism, and complex memory systems
  - Optimization: Manage trade-offs between bandwidth, message rate, concurrency, and synchronization inherent in modern architectures
  - Co-Design: Inform application scientists and system designers of communication system impact on performance
  - Reproducibility: Support continuous, reproducible evaluation and innovation in application, runtime system, and hardware design

Current Communication Systems Inefficient

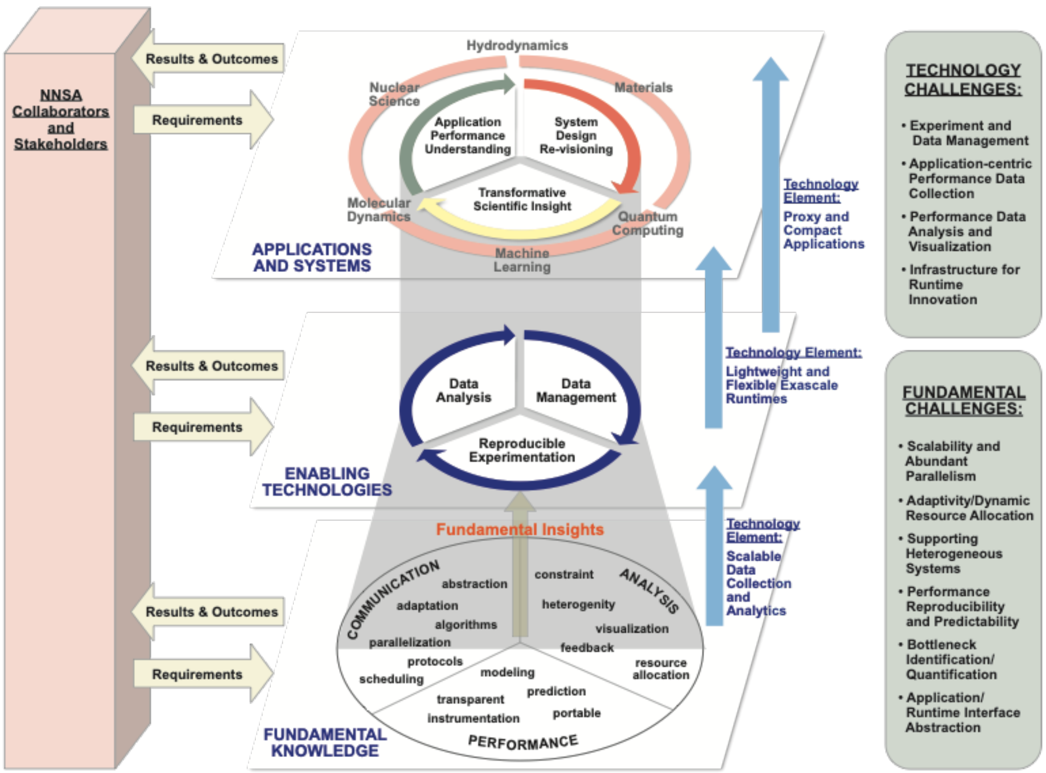
- Current HPC communication systems are incremental outgrowths of single-threaded computing communication systems
- Optimizing communication systems requires managing complicated application/system software/hardware tradeoffs

Application and system designers have questions like:

- **Will the communication system limit performance on or portability to current and next generation systems?**
- **How do I effectively choose or balance between loop-level application synchronization and communication system-level synchronization?**
- **If I invest in rearchitecting my application's compute/communication strategy, how much would time-to-solution improve now or in the future?**
- **Can the communication and/or runtime system actually *mitigate* potential performance problems in my application?**

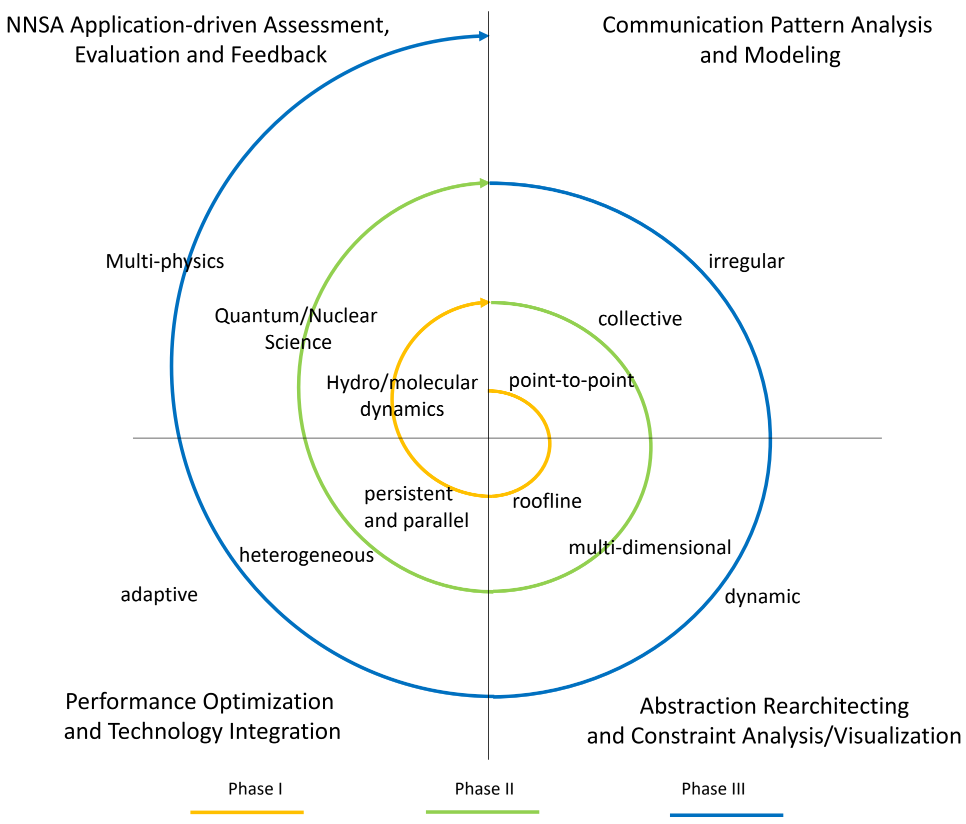
The only answer you can realistically get today is: **"Maybe?"**

Detailed Scientific Description



- **Fundamental Research**
  - Create new communication abstractions/optimizations
  - Deploy performance models to enable optimization and app/runtime co-design
- **Technical Infrastructure Innovations**
  - Communication abstraction prototyping
  - Data/experiment Reproducibility
- **Iterative application and system assessment**

Research: Communication Abstraction Innovation and Optimization



What communication abstraction will best match an application to the communication fabric's capabilities?

- Formative assessment
- Abstraction/Application Re-visioning
- Optimization/Integration
- Summative Assessment

Example: Thread/GPU Communications

Need some form of threading in modern HPC applications

- MPI+X implies the use of threading, for example OpenMP or CUDA
- Tasking runtimes, over-decomposition, and other recent messaging abstractions/implementations encourage concurrent communication

As a result, applications communicate at two extremes:

- Coarse application synchronization: Limit concurrency to maximize peak bandwidth; high network and CPU idle times
- Concurrent messaging: Minimize network/CPU idle times; high message rates/synchronization costs limit effective bandwidth

Need abstractions and optimizations that allow the communication system to balance these tradeoffs

Formative Assessment

- **Work with NNSA laboratory collaborators to identify applications to drive innovation**
  - Have initial assessment of 110+ open-source MPI applications in collaboration with Livermore National Labs
  - Starting with proxy applications to ease evaluation and prototyping
  - Examining creation of communication-representative mini-application
  - Move prototyped innovations into full NNSA applications
- **Use modern tools to quantify how communication abstractions do or will later limit application performance**
  - Variety of research and technical innovations in support of this
  - Previous examples showcase use for understanding MPI/threads interactions in stencil applications

Example Research: Abstraction Innovation

- **Identified 9 different aims to attack legacy performance problems in exascale communication abstractions**
- **For threaded communication, the main issue is tightly coupling host processing with network data movement**
  - Aim 1: Attack "Cost of Portability"
  - Aim 2: Attack Latency and Synchronization
  - Aim 4: Enhance Overlap of Communication and Computation
- **Example Plan of Attack**
  - Decoupling host data processing from network data movement gives runtimes opportunities to improve communication performance
  - Model, analyze, and optimize implementation to balance bandwidth, latency, message rate, runtime overheads, application synchronization
  - Effectively communicate abstraction tradeoffs to applications/runtimes

Example Research: Communication System Performance Models

- **High-level Problem: Spend effort wisely on communication optimization in applications and runtimes**
  - Mapping communication improvement to application performance traditionally very difficult
  - Communication primitives hard to tune for modern networks/systems
  - Myriad examples of communication system performance tuning that doesn't significantly improve application performance
- **Example Questions:**
  - Would replacing large sends with partitioned sends improve application performance?
  - When should we move data from the sender to the receiver to balance buffering, synchronization, and network bandwidth?
- **Leverage high-fidelity models**
  - Various stochastic models to quantify when threads reach barriers/sends
  - Quantifies marginal bandwidth/synchronization/latency tradeoffs

Example Research: Quantify Application/ Communication System Interactions

- **Goal: Understanding how changing communication plans or primitive tradeoffs impacts real applications**
  - Applications have multiple interacting performance bounds
  - Bounds vary by system, architecture, and optimization
- **Approach: Integrate network performance into roofline performance models**
  - Communication rooflines similar to memory rooflines (operational intensity vs. FLOPS) for simple point-to-point
  - Different curves for per-message and per-byte operations (latency vs. bandwidth) – an extra ½ dimension in the analysis
  - Collectives and threading bring in synchronization and imbalance issues – full extra dimensions in the roofline analysis,
  - Stochastic analysis mentioned previously can be used to create approximate compute imbalance/synchronization rooflines.

Technical Innovations

- **Provide clear development/testing/productization pathways**
  - ExaMPI -> Open Source Messaging Systems -> Production software
  - UNM testbed -> Lab experimental systems -> Production systems
  - Mini-applications -> Compact applications -> Production applications
- **ExaMPI – infrastructure prototyping of new abstractions**
  - Support messaging innovation on modern hardware without the legacy costs of current communication frameworks (OpenMPI, MPICH, GASNet)
  - Integrate instrumentation for use with LDMS for systematic profiling
- **Experiment Reproducibility and Data Management**
  - Leveraging containers, modern build and CI systems, experiment management, and data management tools to increase reproducibility
  - Also building on Jupyter notebooks to facilitate easy management and sharing of monitoring, modeling, and analysis results

Evaluation/Feedback Plan

- **Formative and Summative Assessment drive entire process**
  - Also drives refinement of modeling and technical innovations
  - Collaborations with NNSA personnel essential to both types
- **Identify, integrate, develop communication-representative mini-app**
- **Re-vision, Model, Optimize, and Evaluate increasingly complex communication abstractions and implementations**
  - Partitioned Communications and other P2P abstraciotns with a focus on GPUs
  - Collective communication abstractions (with a focus on irregular communication)
  - Multi-physics communication abstractions (e.g. mesh-to-mesh translation)
- **Demonstrate/evaluate innovations in increasingly complex settings**
  - Develop in mini/proxy applications to and transition to complete applications
  - Move from UNM testbeds to NNSA experimental systems to DOE production systems
  - Lab residencies by project personnel and collaborations essential for integrating into more complex codebases

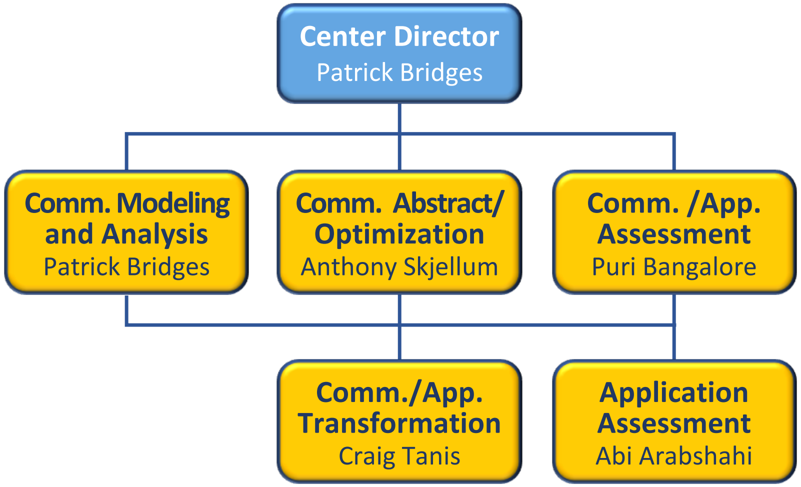
Lab Interaction Plan

- **Strong collaboration with NNSA personnel essential to project success**
  - Identify code bases with highest need for communication innovation
  - Collaboratively envision new abstractions
- **Build on existing academia->NNSA student development pipeline for recruitment/placement**
  - Initial training (courses, mentorship) at Universities
  - Mid-term students develop expertise at Universities and lab summer placements
  - Students finish research and dissertations on year-round lab internship
- **Long project PI record of successful collaboration with all three NNSA laboratories**
  - Many publications, project, contributions toward lab milestones
  - Long record of placement of students to DOE lab staff positions

Management Plan – Center Organization

Responsibilities divided between three Universities

- UNM (Lead) – Leadership, Communication/ Performance Modeling, NNSA Collaboration Lead
- UAB – Communication/ Application Assessment
- UTC - Communication Abstraction/Optimization



- **Leverage state-of-the-art collaboration tools for project management, software release**
  - Slack, Zoom
  - Docker, Singularity
  - Github, Jenkins/Travis, Spack

Management Plan - Milestones

Concrete Milestones for Each Repeated Phase of Assessment, Research, Integration, Feedback

- New phase (partitioned/GPU communication, collective, irregular) starts every 18 months
- Each phase lasts 24 months; overlap feedback/revision of previous phase with formative assessment of following phase
- Infrastructure/assessment milestones during center initiation

	Year				
	1	2	3	4	5
Center Infrastructure Standup					
Source/Data Sharing and Management Infrastructure	X	X			
Communication System Research Infrastructure Development	X	X			
Overarching Assessment/Broadening Activities					
Project Formative Application Assessment	X				
Communication-Representative Mini-App	X	X			
Phase I(a) - Partitioned Communication					
Improvement of Partitioned Communication Abstractions	X				
Model/Analysis of Partitioned Communication Performance	X				
Prototype Implementation of Partitioned Communication Abstraction	X	X			
Technology and NNSA Code Integration	X	X			
Summative Application Performance Assessment	X	X			
Phase I(b) - GPU Communication					
Creation/Optimization of New Communication Abstraction	X	X			
Model/Analysis of GPU Communication Performance	X	X			
Prototype Implementation of GPU Communication Abstractions		X			
Technology and NNSA Code Integration		X	X		
Summative Application Performance Assessment		X	X		
Phase II - Collective Communication Deep Dive					
Integrate Feedback/Design Improvements from Phases I(a) and I(b)		X			
Formative Application Communication/Performance Assessment		X			
Creation/Optimization of New Communication Abstractions		X	X		
Model/Analysis of Collective Communication Performance		X	X		
Technology and NNSA Code Integration			X	X	
Summative Application Performance Assessment				X	
Phase III - Irregular Communication and Multi-physics Deep Dive					
Integrate Feedback/Design Improvements from Phases I and II			X		
Formative Application Communication/Performance Assessment			X		
Creation/Optimization of New Communication Abstraction			X	X	
Model/Analysis of New Communication Abstraction Performance			X	X	
Technology and NNSA Code Integration				X	X
Summative Application Performance Assessment					X

Table 1: Five Year Roadmap