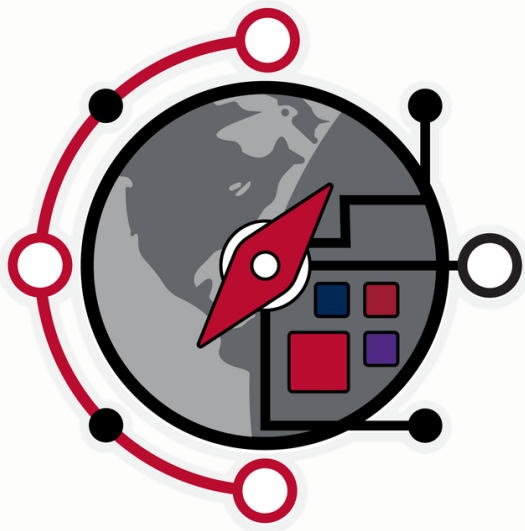




Center for
Optimized Modern
Parallel Adaptive
System Software

Plans for Year 2 and Beyond

PROF. PATRICK BRIDGES · UNIVERSITY OF NEW MEXICO
COMPASS 2026 ANNUAL REVIEW · OCTOBER 29, 2026 · 2:45 PM

Year 2 (FY27) on the Roadmap

	Year 1 (FY26)				Year 2 (FY27)				Year 3 (FY28)				Year 4 (FY29)				Year 5 (FY30)			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
AI Performance / Modeling	Analyze AI model sensitivity				Model data collection & augmentation				Decision under uncertainty				Model testing & validation							
					Research model uncertainty quantification				Research model transfer learning				Enhanced model for optimization integration							
Cross-Layer / Optimization	Assess sparse solver optimizations		Assess Beatnik optimization				Layer-by-layer optimization		Search-based optimization				Integrate model uncertainty				Fully integrate models			
			Scheduler optimization sensitivity				Assess NNSA app optimization opportunities													
Solver Design	Beatnik fast multipole support		Z-Model adaptive meshing		Compare Krylov solver variants				Parallelize topological transitions				Specialize models for physics problems				Integrate models into production codes			
	Develop compressible interface model			Model topological transitions				Multi-scale reactive interface methods				Parallelize multi-scale reactive solver								
				Parallelize compressible interface model						Combine Krylov enhancements into unified solver										
System Software	MPI/Kokkos stream triggered communication				Additional MPI/Kokkos extensions								Time-based communication primitive optimization							
	Create initial RAPIDS primitives				Additional RAPIDS extensions								Cross-layer control of system software APIs							
	Kokkos memory layout optimizations			Develop System Tuning APIs																
Integration & / Assessment	Design app/system workflow			Assess Z-Model/FIESTA				Assess Krylov Variants												
	Design/deploy system infrastructure				Assess NNSA application optimization								Assess integrated hydro codes							
Education & / Workforce	Create bootcamp materials			Develop tradeoff analysis materials				Create AI model course materials				Offer courses with integrated materials								
			Deploy bootcamp modules		Create conference tutorial materials				Offer HOTI/SIAM/SC tutorials											

Highlighted: every task active in Year 2. The dashed line marks today.

Year 2 Goals and Major Activities

Area	Year 2 milestone	Major activities
AI Modeling	Non-intrusive data collection and augmentation; uncertainty in model inputs and outputs	Data collection and augmentation; model UQ; extend models to detect and localize performance problems
Cross-Layer	Top-down and/or bottom-up layer-by-layer optimization of application and system hyperparameters	Finish scheduler sensitivity (Q2); layer-by-layer study (Q3 on); optimization selection under uncertainty
System Software	Demonstrate MPI, Kokkos, and RAPIDS API extensions in an NNSA-relevant code	MPI/Kokkos and RAPIDS extensions, including MailBox; memory layouts; begin system tuning APIs
Solvers	Benchmark standard, communication-avoiding, and related Krylov methods on emerging architectures	Preconditioned Krylov comparisons; topological transitions; parallel compressible interface model
Integration	Assess 3D Z-model vs. LANL Fury: rising bubble/plume with shear and stratification	Z-model assessment; begin NNSA application optimization; agentic adaptation framework prototype
Education	Course materials and tutorials on numerical vs. system-level trade-offs	Trade-off analysis materials; begin conference tutorial materials

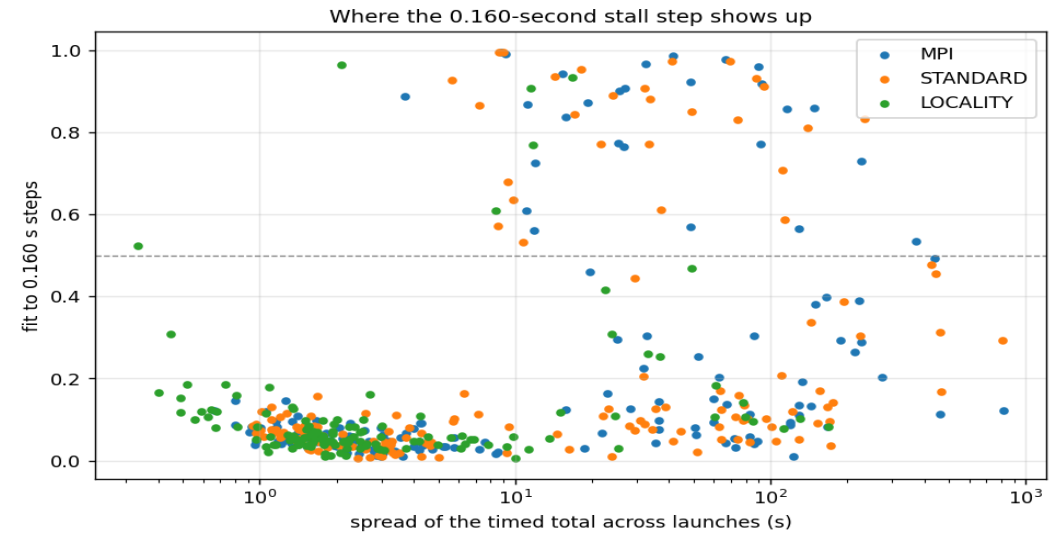
Bold activities are new or expanded since the proposal, driven by Year 1 findings.

Finding: AI Models Can Find Performance Problems

Performance models built for prediction also **detected and tracked down an in-cast congestion problem on Dane**

- 1 Detect**
Runs that fall well outside model predictions are flagged
- 2 Quantify**
Configure or create benchmarks to estimate how much time is lost, and how often
- 3 Localize**
Source Code Archeology: PSM2 in-cast traffic with dense communication patterns exhausts communication contexts and trigger160ms PSM2_ERRCHECK handler

Year 2: turn one case study into a general diagnosis capability across codes and systems

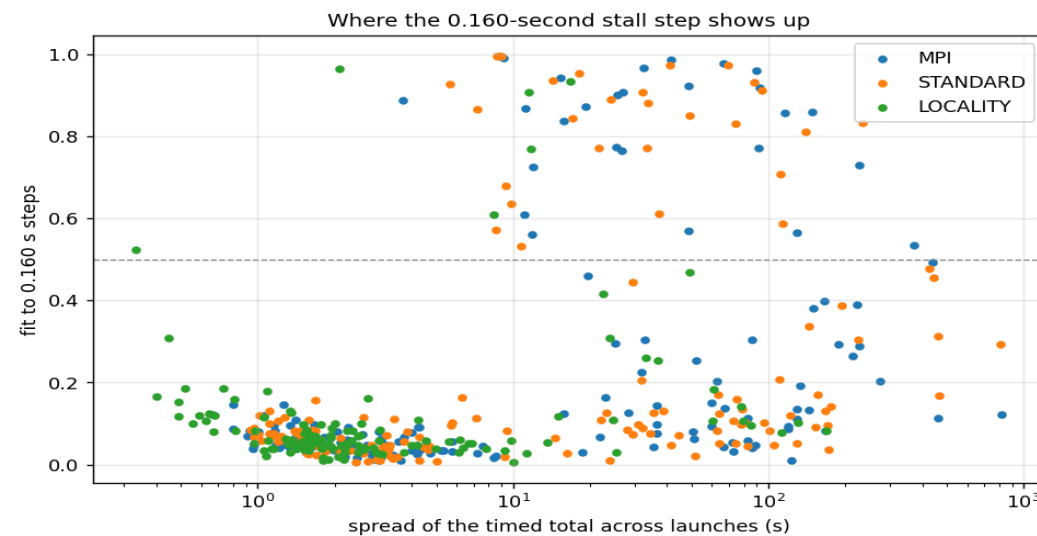


Finding: AI Models Can Find Performance Problems

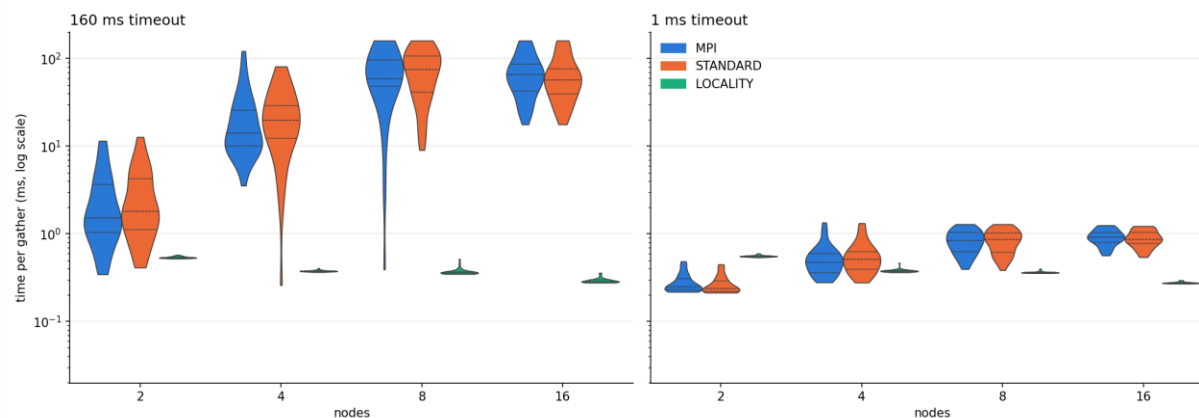
Performance models built for prediction also **detected and tracked down an in-cast congestion problem on Dane**

- 1 Detect**
Runs that fall well outside model predictions are flagged
- 2 Quantify**
Configure or create benchmarks to estimate how much time is lost, and how often
- 3 Localize**
Source Code Archeology: PSM2 in-cast traffic with dense communication patterns exhausts communication contexts and trigger160ms PSM2_ERRCHECK handler

Year 2: turn one case study into a general diagnosis capability across codes and systems

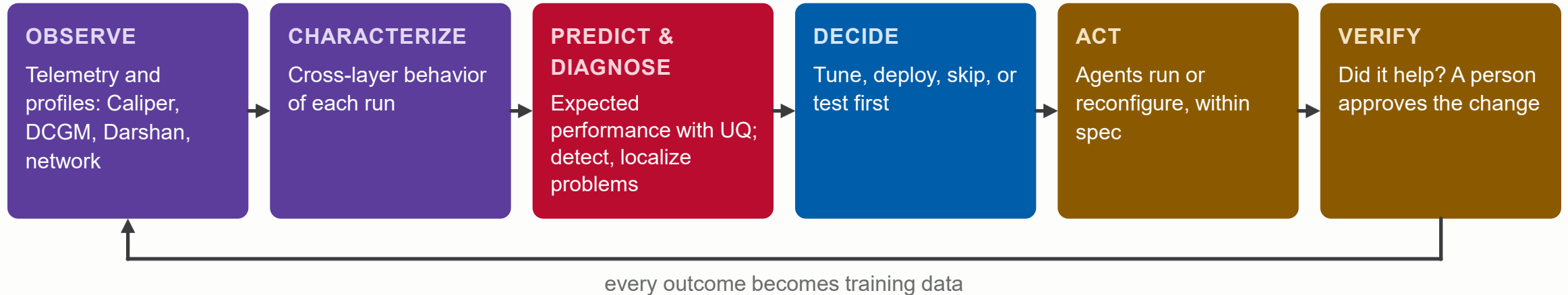


crankseg_2 .pm: time per gather, 160 ms vs 1 ms PSM2 error-check timer
40 launches per violin; inner lines = median (dashed) and quartiles (dotted)



Next: An Integrated Agentic Adaptation Framework

The three findings become one loop, with each step served by tools the center already has



GUARDRAILS Specs bound every action · typed MCP tools only · human approval · full provenance

APPLIED TO hypra/aCG and Krylov solvers · Beatnik (Cabana) and multi-scale hydro · [NNSA application]

Research Questions: Four Properties to Achieve

Automatic

How much of the loop can run without an expert stepping in?

Measure: share of steps automated; expert time per study

Systematic

Can it cover the optimization space reproducibly, with provenance for every decision?

Measure: coverage of the configuration space; reproducibility of results

Safe

How can we guarantee AI-driven actions stay within allocation, system, and policy limits on shared machines?

Measure: potential policy violations; node-hours spent per finding

Effective

Does it find and fix real problems faster and better than expert tuning?

Measure: speedup and time-to-diagnosis vs. expert-tuned baselines on NNSA-relevant codes

Year 2 Plan for the Framework (proposed)

Q1 · OCT–DEC

Shared data and tools

One store for runs, telemetry, and tuning results; characterization and launch as typed MCP tools

Q2 · JAN–MAR

Diagnose with uncertainty

Generalize the in-cast diagnosis; UQ in diagnosis and performance models; few-shot transfer to new systems

Q3 · APR–JUN

Decide and act

Autotuning and decision support as agent tools, tied to the layer-by-layer study; guardrails aligned with Lab AI guidance

Q4 · JUL–SEP

Pilot end to end

Full loop on our current benchmarks on Dane and Tuolumne, evaluated against the four properties

Where the Framework Sits on the Roadmap

	Year 1 (FY26)				Year 2 (FY27)				Year 3 (FY28)				Year 4 (FY29)				Year 5 (FY30)			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
AI Performance / Modeling	Analyze AI model sensitivity				Model data collection & augmentation				Decision under uncertainty				Model testing & validation							
					Research model uncertainty quantification				Research model transfer learning				Enhanced model for optimization integration							
Cross-Layer / Optimization	Assess sparse solver optimizations		Assess Beatnik optimization				Layer-by-layer optimization		Search-based optimization				Integrate model uncertainty				Fully integrate models			
			Scheduler optimization sensitivity				Assess NNSA app optimization opportunities													
Solver Design	Beatnik fast multipole support		Z-Model adaptive meshing		Compare Krylov solver variants				Parallelize topological transitions				Specialize models for physics problems				Integrate models into production codes			
	Develop compressible interface model				Model topological transitions				Multi-scale reactive interface methods				Parallelize multi-scale reactive solver							
					Parallelize compressible interface model								Combine Krylov enhancements into unified solver							
System Software	MPI/Kokkos stream triggered communication				Additional MPI/Kokkos extensions								Time-based communication primitive optimization							
	Create initial RAPIDS primitives				Additional RAPIDS extensions								Cross-layer control of system software APIs							
			Kokkos memory layout optimizations		Develop System Tuning APIs															
Integration & / Assessment	Design app/system workflow				Assess Z-Model/FIESTA				Assess Krylov Variants											
	Design/deploy system infrastructure								Assess NNSA application optimization				Assess integrated hydro codes							
Education & / Workforce	Create bootcamp materials				Develop tradeoff analysis materials								Create AI model course materials				Offer courses with integrated materials			
			Deploy bootcamp modules						Create conference tutorial materials				Offer HOTI/SIAM/SC tutorials							

No new thrust: the framework connects existing tasks, from data collection and UQ through layer-by-layer and search-based optimization, system tuning APIs, and NNSA application assessment, to full model integration in Year 5

How the Plan Has Evolved, and Why

Area	In the proposal	Now	Why
Infrastructure	Integrate and extend existing tools	Research an integrated agentic adaptation framework on the center infrastructure	Evidence that AI diagnosis, agentic characterization, and agentic experiments are feasible
AI Modeling	Predict performance	Predict, and also detect and localize problems	Models found the in-cast problem on Dane
Model UQ	Year 1	Year 2	Initial assessment of model capabilities
Cross-Layer	Broad opportunity assessment by Y1 Q4	Scheduler and communication sensitivity first; layer-by-layer from mid-Year 2	TST input at mid-year; telemetry availability
Krylov	Compare and assess standard, Enlarged, and combined Krylov	Compare and assess Krylov solver variants, starting with preconditioned communication-avoiding and pipelined methods	Enlarged Krylov lacks a preconditioner

Beyond Year 2: The Path to Year 5

Year 3 · FY28

- Decisions under uncertainty; transfer learning
- Search-based optimization
- Multi-scale reactive interface methods
- Time-based communication primitives
- Begin assessing Krylov variants (Q3)

Infrastructure: more codes and systems [proposed]

Year 4 · FY29

- Model testing and validation
- Model uncertainty in optimization
- Models specialized for physics problems
- Cross-layer control of system software APIs
- Assess Krylov variants; begin integrated hydro assessment
- AI-model courses; HOTI/SIAM/SC tutorials

Infrastructure: drives cross-layer system software control [proposed]

Year 5 · FY30

- Fully integrated models
- Models in production codes
- Assess integrated hydro codes
- Courses with integrated materials

Infrastructure: the integrated, automated optimization system in production codes [proposed]

Biggest Risk and Most Pleasant Surprise

WHAT KEEPS US UP AT NIGHT

Trustworthy automation on shared Lab systems

AI-driven experiments must be safe for allocations and other users, and models need enough representative data, including telemetry we cannot yet get on Tuolumne.

Mitigation: guardrails and human approval from the start; small-scale pilots first; work with LC on telemetry access

WHAT LETS US SLEEP WELL

Our models found a real problem

Models built to predict performance detected, quantified, and localized in-cast congestion on Dane: a use we had not planned for, and one that points straight at the Year 2 direction.

Also: stream-triggering work reached the MPI Forum in Year 1

Three Takeaways

1 · KEY ACCOMPLISHMENT

CPU-free GPU communication

Co-designed with Lab staff, published at Cluster 2026, and on a standards path at the MPI Forum

2 · FUTURE PLAN AND CHALLENGE

A safe, effective agentic adaptation framework

Connect characterization, diagnosis, tuning, and agentic experiments into one loop that is automatic, systematic, safe, and effective

3 · DISCRETIONARY

Telemetry access

Job-level telemetry on Tuolumne would directly speed up both the cross-layer and diagnosis work