Performance Intropection for Exascale Computing (PIE)

Simulation supported science is of increasing importance for many global-scale grand challenge problems such as understanding climate change, efficiently modeling nuclear fusion, predicting the effects of coastal processes such as hurricanes or oil spills, or investigating the origins of our universe. At the same time, the accessibility and utility of the ultrascale HPC systems that drive simulation science are in danger of eroding. The usefulness of petascale resources is already limited to science teams who can both deal with the complexities of expressing their science goals in HPC form and deal with complex performance optimization scenarios which vary across architectures and across runs. As we move toward exascale, without research and development effort applied to these two issues the utility of exascale systems will be of increasingly limited scientific and social relevance.

We propose a program of research that will make it easier for scientists to express their problems in programmatic form suitable for HPC, and also to make performance optimization more accessible and productive. These two efforts respectively form the HPC entry points and outputs that are growing steadily more complex.

We will then apply these methods to driver problems in coastal modeling and general relativistic astrophysics, aiding ongoing research in our collaboration, which will ensure that our methods will in the end be practically applicable. Our program of research will bring automatic code generation and scalable performance introspection to exascale.

Performance Introspection for Exascale Computing (PIE)

1 Overview

In recent decades simulation-based science and engineering has become a dominant research trend in many fields. For global-scale grand challenge problems – whether understanding climate change, efficiently modeling nuclear fusion, predicting the effects of coastal processes such as hurricanes or oil spills, or investigating the origins of the universe in which we live – it is crucial to be able to perform high fidelity simulations that comprehensively model the many involved physical processes and span different temporal and spatial scales.

Creating simulation codes for any of these examples involves developing highly complex software environments from the runtime system level to the application level and understanding the complex computer architectures in which applications execute. One also needs to bring together experts from different domains, integrating expertise in domain science, computer science, applied mathematics, software engineering and collaborative environments. Further, as computational science moves from petascale to exascale, the challenges posed by heterogeneous system architecture and extreme scales require us to rethink the way simulation models are being built.

Our plan of research and development addresses these issues by bringing together an experience consortium[1] to focus on the development of an extensible application component framework which provides automatic code generation directly from scientific equations combined with performance introspection to allow an application to adapt to its own dynamic runtime characteristics. Science drivers include global scale problems in coastal science and relativistic astrophysics that use community toolkits [1, 2].

Our approach is based on existing, well-used, open-source technologies developed by our consortium:

- `Cactus` [10, 3], developed at LSU and AEI, is an open-source component framework fundamentally designed for the collaborative development of large-scale, complex, applications including relativistic astrophysics [7], computational fluid dynamics, reservoir simulations, coastal science and computer science (see Fig. 1). These applications all share underlying computational components in `Cactus` such as parallel I/O, coordinate systems, adaptive mesh refinement, and message passing. `Cactus` also provides introspection capabilities to its components [13].

- `IPM` [14, 4] is a highly efficient integrated performance monitoring tool developed at LBL. It will enable simulation models being built with PIE to automatically adapt to load imbalance situations and to dynamically select from algorithm variants based on performance feedback delivered to the application at runtime.

- `Kranc` [11, 5] and `Simflowny` are two synergistic tools developed at AEI and UIB respectively which are both able to automatically generate code for `Cactus`. `Kranc` has been used to construct production-quality relativity codes now freely available as part of the community Einstein Toolkit [1]. `Simflowny` is a model management tool based on a formal representation of physical models based on PDEs, physical problems, and discretization techniques.

[1]LSU (Louisiana State University), AEI (Max Planck Institute for Gravitational Physics), SOTON (Southampton University), LMU (Ludwig-Maximilians-Universität München), LBL (Lawrence Berkeley Laboratory), UIB (University of the Balearic Islands), BSC (Barcelona Supercomputing Center)

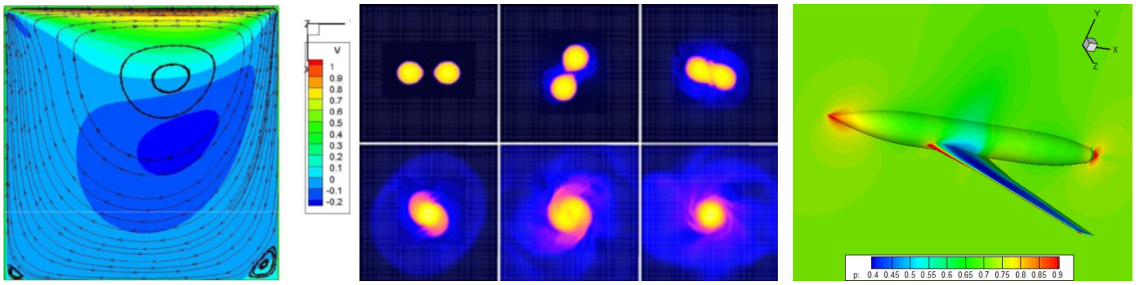

Figure 1: Example results from Cactus-based simulations in various science areas: teaching CFD basics (lid-driven cavity flow), relativistic astrophysics (merging neutron stars), and compressible CFD (flow around an airfoil).

2 Joint Research Plan

PIE will fundamentally address the accessibility and utility of extreme scale HPC systems for simulation science, providing technologies for automated code generation that will allow scientists to express their scientific problems in a high level form, and performance introspection that will integrate with applications to enable scalable exascale simulations. The four core research and development targets of PIE (see Fig. 2) are:

1. Robust Problem Formulation (SOTON, AEI): *Develop procedures to translate governing equations into a robust formalism for automated processing and implementation as simulation framework components.*

Most simulation codes are highly specialized to a particular system of equations, application area and discretization approach. Bringing these codes to the exascale involves significant research and development where it is difficult to leverage approaches from other applications. Our project involves a common underlying computational framework (`Cactus`) whose components are shared between different applications making advances in one area immediately available in others. Further, PIE involves applied mathematicians, who will investigate formalisms of governing equations that can be robustly discretized and integrated using coupling mechanisms into multiphysics simulations. SOTON will lead this effort levering their experience with high accuracy algorithms for complex multiphysics problems, such as the interaction between fluid and elastic matter interfaces in binary neutron star mergers [6, 9].

2. Automated Code Generation (AEI, LMU, UIB): *Generate software components from equations, enable automated optimisations, allow tuning for different hardware architectures and different parallel paradigms (MPI, OpenMP, hybrid, UPC, accelerators, etc.)*

Extending the capabilities of `Kranc` and `Simflowny` will allow application code authors to focus on the application-level aspects of the problem – the equations being solved – without having to worry about the details of the numerical implementation or performance characteristics [12]. In PIE, AEI will provide additions to `Kranc` so that any generated code can interface with available performance-related APIs and choose different algorithms and implementations of the equations based on that performance information. UIB will collaborate with AEI on a formal representation for model specification, and provide broad outreach to other applications through the general `Simflowny` system which will be extended using approaches developed using `Kranc`.

3. Performance Introspection (LMU, LSU, LBL, BSC): *Integrate run time performance information for highly concurrent, multicomponent simulations into the simulation framework to*

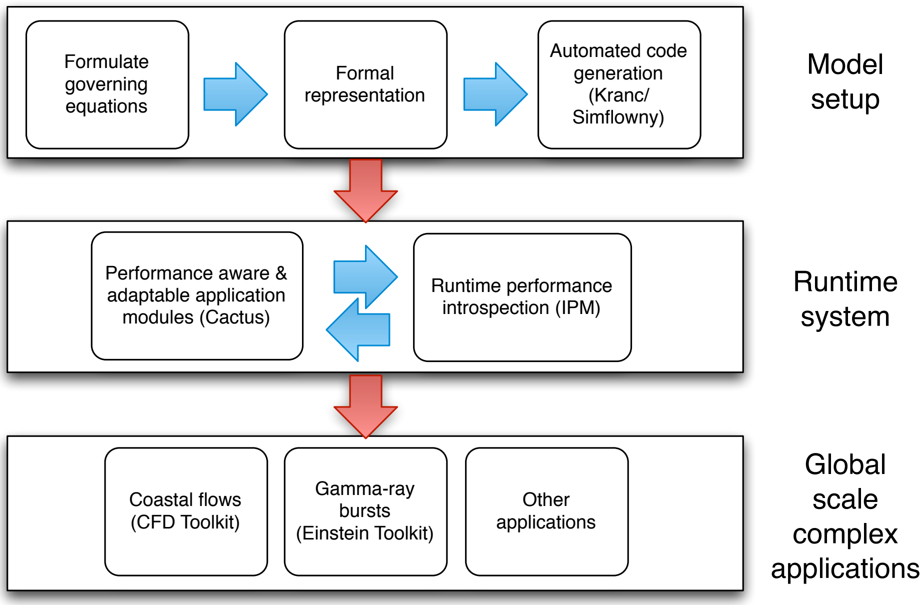

Figure 2: The PIE approach includes problem formulation, code generation, and performance introspection for global scale complex applications.

support dynamic decision making, dynamic optimizations, automated application level performance problem detection and reporting.

We will deliver performance introspection to applications through an extension of highly efficient and lightweight monitoring techniques implemented in the Integrated Performance Monitoring (`IPM`) framework in use at DOE, NSF, and commercial computing centers. This will allow applications to query performance metrics while they execute, enabling `Cactus` to provide a feedback-directed mitigation of load imbalance situations and the automatic selection of algorithm and code variants generated by the `Kranc` framework.

In PIE, LBL will be a source of understanding the runtime needs of HPC applications. We will build an interface for communicating performance information on an as-needed basis to the `Cactus` runtime. Instead of making performance engineering decisions on an after-the-fact or per job basis, our approach offers the ability to make just-in-time decisions about algorithm and library level strategies. This perspective can provide great value to end users in that they are not required to decide about performance choices prior to issuing their workflows. LMU will cover the design and development aspects of adapting `IPM` for integration with `Kranc` and `Cactus`. Through the close connection between LMU and the Leibniz Supercomputer Centre (which is participating in the European PRACE project) the consortium will have access to emerging hardware platforms. BSC will contribute experience with advanced analytical methods for online data analysis.

4. Global Scale Challenges (LSU, AEI, SOTON): *Drive PIE research and development by real-world science use cases that address global grand challenge problems in science and engineering and represent a broad range of applications.*

A. Modeling Coastal Environments: `Cactus` is providing the framework for the development of a community toolkit for modeling coastal flows by the DOD-funded COMI project [8] (Allen is co-PI). COMI researchers are developing simulation codes for modeling the effects of hurricane driven coastal flows on the coastline of Louisiana and other coastal phenomena such as the movement of oil spills.

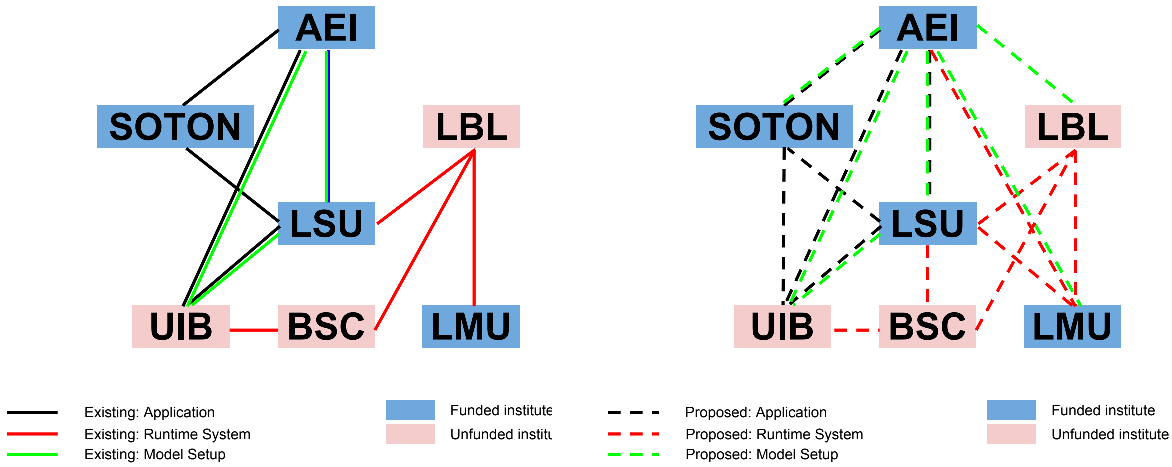

Figure 3: Current (Left) and proposed (Right) connections in the PIE consortium.

B. Modeling Gamma-Ray Bursts: LSU, AEI and SOTON are all involved in the development of multiphysics and multiscale systems to model astrophysical phenomena such as Gamma-Ray Bursts, some of the most energetic events in the universe, and likely prominent generators of gravitational waves. These groups all use the `Cactus` framework and the Einstein Toolkit. At LSU, this work is funded by an NSF PetaApps and NSF PRAC award.

3 PIE Consortium and Workplan

We have assembled a uniquely experienced team of researchers with proven skills in large-scale scientific computing and collaborative software development too numerous to describe in this pre-proposal. In physics, LSU, SOTON, AEI and UIB already collaborate on scientific studies in computational relativity using `Cactus` and are core partners in the community Einstein Toolkit. In computer science, Munich, LBL and BSC provide experience and research in performance modeling and tool development. Fig. 3 shows the existing links between partners, and new connections that will be established to good effect in PIE.

The PIE project will fund postdocs at LSU, AEI, SOTON, and LMU who will collaborate closely to extend the open source `Cactus`, `IPM` and `Kranc/Simflowny` tools to provide capabilities for performance introspection for large-scale scientific applications. Externally funded applications in coastal science and relativistic astrophysics will provide the science drivers for this work. All software developed in this work will be released to the community under open source licenses.

Education and Outreach: We have a full educational, outreach, and collaboration management plan that will be part of our full proposal.

Involvement of Non-G8 and Other Unfunded Partners: PIE involves committed partners who will not receive funding from this award but provide important contributions: LBL/UCM develop the `IPM` software that will be used in PIE and have great experience in its use for application performance profiling; UIB will contribute to formal representation of models for automated code development and their general `Simflowny` system will be developed to include a range of applications and frameworks; BSC will contribute to the performance modeling activity and enable outreach to diverse applications.

References

[1] Einstein Toolkit, URL `http://www.einsteintoolkit.org/`.

[2] CFD Toolkit, URL `http://www.cfdtoolkit.org/`.

[3] Cactus Computational Toolkit home page, URL `http://www.cactuscode.org/`.

[4] IPM: Integrated Performance Monitoring, URL `http://ipm-hpc.sourceforge.net/`.

[5] Kranc: Automated Code Generation, URL `http://numrel.aei.mpg.de/Research/Kranc/`.

[6] L. Baiotti et al. Three-dimensional relativistic simulations of rotating neutron star collapse to a Kerr black hole. *Phys. Rev.*, D71:024035, 2005. eprint gr-qc/0403029.

[7] CIGR Collaboration. Community infrastructure for general relativistic MHD. Technical Report CCT-TR-2008-6, Louisiana State University, 2008. URL `http://www.cct.lsu.edu/CCT-TR/CCT-TR-2008-6`.

[8] C. Eskilsson, Y. E. Khamra, D. Rideout, G. Allen, Q. Q. Chen, and M. Tyagi. A parallel high-order discontinuous galerkin shallow water model. In *Proceedings of the 9th International Conference on Computational Science, Baton Rouge, LA. Lecture Notes in Computer Science Vol. 5544*, pages 63–72. Springer, 2009.

[9] B. Giacomazzo and L. Rezzolla. WhiskyMHD: a new numerical code for general relativistic magnetohydrodynamics. *Class. Quant. Grav.*, 24:S235–S258, 2007. eprint gr-qc/0701109.

[10] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf. The Cactus framework and toolkit: Design and applications. In *Vector and Parallel Processing – VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science*, Berlin, 2003. Springer.

[11] S. Husa, I. Hinder, and C. Lechner. Kranc: a Mathematica application to generate numerical codes for tensorial evolution equations. *Comput. Phys. Comm.*, 174:983–1004, 2006. eprint gr-qc/0404023.

[12] A. Hutanu, E. Schnetter, W. Benger, E. Bentivegna, A. Clary, P. Diener, J. Ge, R. Kooima, O. Korobkin, K. Liu, F. Löffler, R. Paruchuri, J. Tao, C. Toole, A. Yates, and G. Allen. Large-scale problem solving using automatic code generation and distributed visualization. Technical Report CCT-TR-2009-11, Louisiana State University, 2009. URL `http://www.cct.lsu.edu/CCT-TR/CCT-TR-2009-11`.

[13] E. Schnetter, G. Allen, T. Goodale, and M. Tyagi. Alpaca: Cactus tools for application level performance and correctness analysis. Technical Report CCT-TR-2008-2, Louisiana State University, 2008. URL `http://www.cct.lsu.edu/CCT-TR/CCT-TR-2008-2`.

[14] N. J. Wright, W. Pfeiffer, and A. Snavely. Characterizing parallel scaling of scientific applications using ipm. In *The 10th LCI International Conference on High-Performance Clustered Computing*, March 10-12, 2009.