FACETS (Framework Application for Core-Edge Transport Simulations) Performance Analysis

ParaTools, Inc., 2836 Kincaid St., Eugene, OR 97405, info@paratools.com

Final Report for DOE DOE Co-operative Agreement No. DE-FC02-07ER54910

PI: Dr. Allen D. Malony
Team members: Dr. Sameer Shende, Mr. Alan Morris,
Mr. Wyatt Spear, Dr. Kevin A. Huck

Executive Summary

The goal of the FACETS project (Framework Application for Core-Edge Transport Simulations) was to provide a multiphysics, parallel framework application (FACETS) that will enable whole-device modeling for the U.S. fusion program, to provide the modeling infrastructure needed for ITER, the next step fusion confinement device. Through use of modern computational methods, including component technology and object oriented design, FACETS is able to switch from one model to another for a given aspect of the physics in a flexible manner. This enables use of simplified models for rapid turnaround or high-fidelity models that can take advantage of the largest supercomputer hardware. FACETS does so in a heterogeneous parallel context, where different parts of the application execute in parallel by utilizing task farming, domain decomposition, and/or pipelining as needed and applicable.

ParaTools, Inc. was tasked with supporting the performance analysis and tuning of the FACETS components and framework in order to achieve the parallel scaling goals of the project. The TAU Performance System® was used for instrumentation, measurement, archiving, and profile / tracing analysis. ParaTools, Inc. also assisted in FACETS performance engineering efforts.

Through the use of the TAU Performance System, ParaTools provided instrumentation, measurement, analysis and archival support for the FACETS project. Performance optimization of key components has yielded significant performance speedups. TAU was integrated into the FACETS build for both the full coupled application and the UEDGE component. The performance database provided archival storage of the performance regression testing data generated by the project, and helped to track improvements in the software development.

Table of Contents

1. Abstract

The goal of the FACETS project (Framework Application for Core-Edge Transport Simulations) [1] was to provide a multiphysics, parallel framework application (FACETS) that will enable whole-device modeling for the U.S. fusion program, to provide the modeling infrastructure needed for ITER [2], the next step fusion confinement device. FACETS is highly flexible, through use of modern computational methods, including component technology and object oriented design, to facilitate switching from one model to another for a given aspect of the physics. This feature enables the use of simplified models for rapid turnaround or high-fidelity models that will take advantage of the largest supercomputer hardware. FACETS does so in a heterogeneous parallel context, where different parts of the application run in parallel using task farming, domain decomposition, and/or pipelining as needed and applicable.

An integral part of the FACETS development is the coupling of existing core and edge simulations, with the transport and wall interactions described by reduced models. More detailed wall interactions as well as greater coupling complexity, including the addition of near-first-principles computations of turbulent transport in the core and the edge, could then be added. This development plan was created to provide early delivery of capability, with continual delivery of new capability throughout the project. The principle investigator for the FACETS project is John R. Cary, Tech-X Corporation [3]. FACETS funding partners include the United States Department of Energy Office of Science [4], Office of Advanced Scientific Computing Research [5], and the Office of Fusion Energy Sciences [6]. More information about the project can be found at http://www.facetsproject.org.

A strong, interdisciplinary team of computational physicists, applied mathematicians, and computer scientists was assembled for this project in order to address the issues of physics model construction, dynamical systems coupling and solving, and software methodologies and application performance. To that end, ParaTools, Inc. [7] was tasked with supporting the performance analysis and tuning of the FACETS components and framework in order to achieve the parallel scaling goals of the project. The TAU Performance System® [8] was used for instrumentation, measurement, and profile / tracing analysis. ParaTools, Inc. also assisted in FACETS performance engineering efforts.

2. Introduction

Extending scientific simulations into the petascale performance regime presents a significant software engineering challenge. The scale of the computation and data manipulation is enormous. Performance problems that are insignificant on a small number of processors can grow to unmanageable proportions, preventing good performance on larger numbers of processors. Poorly thought out designs and algorithmic choices can easily lead to overheads and computational complexities that exceed the capabilities of current technologies as problem size or complexity are increased. Petascale performance demands create petascale constraints on design and development of simulation software. These levels of performance cannot be reached by accident, only by intention. A careful process of performance engineering must be integral to the design process if a project is to reach petascale performance. Successful scaling requires more than strong single CPU performance. It requires efficient scalable algorithms, I/O, data structures, and communication pattern design.

The performance engineering work within this project began with initial performance testing. For this testing, a test case was identified for each of the key FACETS codes to guide the analysis and optimization efforts. The first consideration is that the selected test cases should be representative of the actual usage of the codes. For those codes that permit selection of the solution algorithms, the test case should be constructed to configure the algorithms applicable to problems of interest – it makes no sense to optimize cases that will not be run. The test cases were chosen to be of the smallest problem size that yields performance profile data representative of production runs. The codes were instrumented with TAU and baseline performance profiles were obtained for the test cases. Scaling and I/O requirements were assessed and the key performance-limiting algorithms were examined for efficiency and compatibility with the target architectures at production scale.

Once the baseline performance measurements were obtained, they were analyzed in relation to the capabilities of the target architectures. Additional performance experiments were run to explore the impact of platform-specific capabilities. Next we will identified the most fruitful avenues for proceeding with performance improvements and establish both objectives for performance improvements and requirements for absolute performance. All data from performance experiments were maintained in the TAU performance database.

The TAU Performance System [8] is a portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, Java, Python. TAU (Tuning and Analysis Utilities) is capable of gathering performance information through instrumentation of functions, methods, basic blocks, and statements. All C++ language features are supported including templates and namespaces. The API also provides selection of profiling groups for organizing and controlling instrumentation. The instrumentation can be inserted in the source code using an automatic instrumentor tool based on the Program Database Toolkit (PDT) [9], dynamically using DyninstAPI [10], at runtime in the Java Virtual Machine, or manually using the instrumentation API. Hardware counter support is provided through PAPI [11].

TAU's profile visualization tool, ParaProf [12], provides graphical displays of all the performance analysis results, in aggregate and single node/context/thread forms. The user can quickly identify sources of performance bottlenecks in the application using the graphical interface. In addition, TAU can generate event traces that can be displayed with the Vampir [13], Paraver [14] or JumpShot trace visualization tools.

TAU also includes PerfDMF [15], a performance data management framework for archival storage of performance data and associated metadata and PerfExplorer [16], a framework for data mining and analysis support for large-scale multidimensional performance data, including parametric study support. PerfExplorer provides both an interactive interface and scripting support, which is key for automated performance regression testing. Regression-testing procedures in FACETS included performance profiling and analysis to monitor the ongoing evolution of the performance issues so that problems are noticed early rather than late in the development process. The performance database played a key role in the management of performance data throughout the development life cycle.

3. Public Benefits

The Department of Energy's mission is to advance the national, economic and energy security of the United States. Fusion Energy has been identified as having the potential to play a key role in meeting the nation's future energy needs. This is a complex, international endeavor, with the next magnetic-fusion plasma confinement device, ITER, costing of order $10 billion. Prior to the construction of any such large device, there is a need to understand performance as it relates to device parameters in order to arrive at an optimum device for demonstrating the next step on the road to fusion energy. As performance is related to the rate at which heat and particles are lost to the wall, one would like, therefore, to be able to compute the transport, the rate at which heat and particles are lost to the wall, from first principles, for any of the various configurations contemplated for ITER and, ultimately, a fusion power plant. FACETS will allow optimization of ITER operation and ITER-developed physics understanding through comparative ease of computational diagnostics.

4. Technical Approach

This section describes the main ParaTools activities in the FACETS project. Section 5 discusses our accomplishments in detail.

4.1. TAU Integration with FACETS Component Codes

In the first year of the project, ParaTools focused on the integration of the TAU performance system with the FACETS code components in core transport and edge transport. The goal was to establish a common infrastructure for performance measurement and experimentation that would serve as the foundation for performance engineering of FACETS component coupling. The specific core transport codes were to be drawn from the NTCC [17] library, pTRANSP [18] development, and CWIM [19] project. In the case of edge transport, the codes included UEDGE [20], TEMPEST [21], and XGC [22]. ParaTools interacted with FACETS computational scientists to identify performance observation requirements and prioritize work on specific codes. In addition, we worked with the TOPS project [23] to determine how to measure the performance of TOPS solvers.

4.2. Performance Data Mining and TAU Integration with Coupling Interface

To evaluate the performance consequence of core-edge, transport-turbulence, and edge-wall couplings, it is important to understand the performance space of the constituent codes. ParaTools built a performance database from the component characterization results and developed performance data mining procedures to extract properties for performance engineering decisions during the coupling process. The TAU ParaProf and PerfExplorer tools were updated to support these analysis procedures. The second part of work involved the integration of TAU instrumentation and measurement in componentization and coupling methods. The goal was to define and implement a performance observation strategy for the FACETS multi-component coupling framework that can provide measured data for simulation performance analysis.

4.3. Integrated Performance Engineering of FACETS Coupled Simulations

Based on the integration of TAU with the FACETS coupling framework and the baseline performance characterization of coupled simulations, ParaTools develop a performance engineering environment with ORNL for the evaluation and optimization of FACETS coupled simulations. As part of this environment, we constructed a harness for performance experimentation that allows performance parameters and factors to be selected for evaluation during a simulation run. In addition, we extended the FACETS performance database for storing coupled performance results and enhance data mining procedures for coupling-oriented analysis. Testing of FACETS coupled simulations helped to validate the performance engineering infrastructure and demonstrate its capabilities to FACETS scientists. All experimental results were maintained in the database unique to each system platform. Throughout the project, the TAU PerfDMF performance database served as a repository of experiments and enabled further data analysis as new results are added. In particular, these results were useful for historical performance regression analysis.

4.4. Runtime Performance Tuning of FACETS Coupled Simulations

The FACETS performance engineering work is complemented by research in computational quality of service (CQoS) in component-based systems. Based on our involvement with CCA researchers in this area, we looked into how to apply CQoS technology to implement support for runtime performance tuning of coupled fusion simulations. This involved the FACETS components and framework teams to decide what sensors and actuators to make available at runtime, and how component adaptations will occur. These decisions were considered for extensions to the TAU component measurement system for runtime performance observation and query, and component control and replacement, as integrated with the FACET components and coupling framework. The performance models are instrumental in this work. Experiments then demonstrate the validity of the runtime tuning approach with fusion simulations.

4.5. FACETS Performance Engineering Technology Delivery

Our development of performance engineering technology for FACETS will be available to the community in the final phase of the project. This includes the FACETS performance database, as well the experimentation and analysis tools developed for the project.

5. Technical Accomplishments

5.1. TAU Integration with Facets Component Codes

In the early stages of the project, the ParaTools team focused on the integration of the TAU performance system with the FACETS code components, and demonstrated the use of TAU within the different programming environments such as Python, C++ and Fortran. Specifically, the gsslvex and xpoisson aspects of the FACETS C++ framework were instrumented and measured. At the Python level, the FACETS coupling framework, Numpy, pyMPI and PETSc were instrumented and measured.

The NUBEAM [24] and UEDGE component applications were instrumented and measured. A scalability analysis of up to 128 processors was performed on NUBEAM.

The CORE solver in the FACETS framework was measured in a strong scaling study, and it was determined that the scaling was not good – the performance actually fell off when increasing the number of processors from 64 to 128. Two key optimizations were implemented – the first removed a redundant flux computation, and the second replaced synchronous blocked communication with non-blocking sends. Both overall performance and scaling was improved through these optimizations. Figure 1 shows the scaling performance of the CORE solver on Bassi, an IBM p575 POWER 5 system at NERSC [25].

Finally, the initial performance database was established, and performance data access was provided to the group through the TAU portal, a web interface through which the performance data can be browsed and the TAU analysis tools can be launched. With the performance database in place, performance regression testing was initiated. The FACETS project team was also trained in the use of TAU performance measurement and analysis tools.

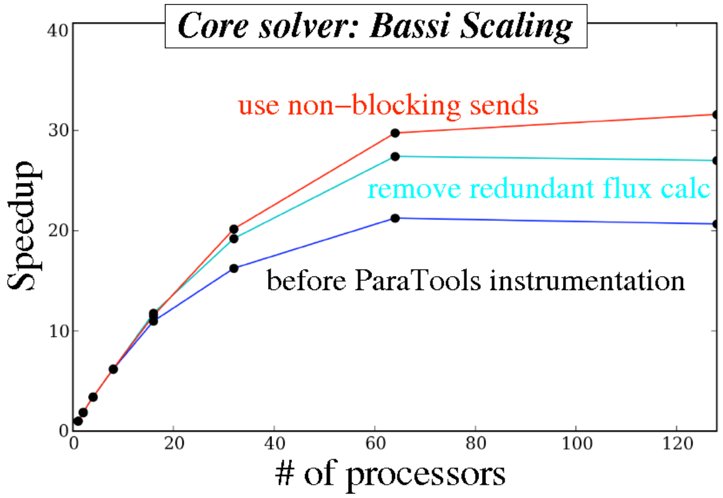

Figure 1: Scaling performance of NUBEAM on Bassi. The original performance is shown as compared to the performance with two key optimizations.

5.2. Performance Data Mining and Initial Performance Regression Testing

With the establishment of the performance database, ParaTools began the process of automated performance measurement and analysis. The ParaTools team integrated the TAU performance tools into FACETS and implemented automated scripts running a scaling study of the core solver on Bassi at NERSC. The scripts were executed weekly, and the performance charts are posted automatically on the web at http://www.facetsproject.org/wiki/TAURegression. Figure 2 shows an example regression chart, demonstrating the weekly performance of the solver as code changes are made within the project. NERSC decommissioned Bassi in April of 2010, and regression testing was migrated to Franklin, a Cray XT4 system at NERSC.

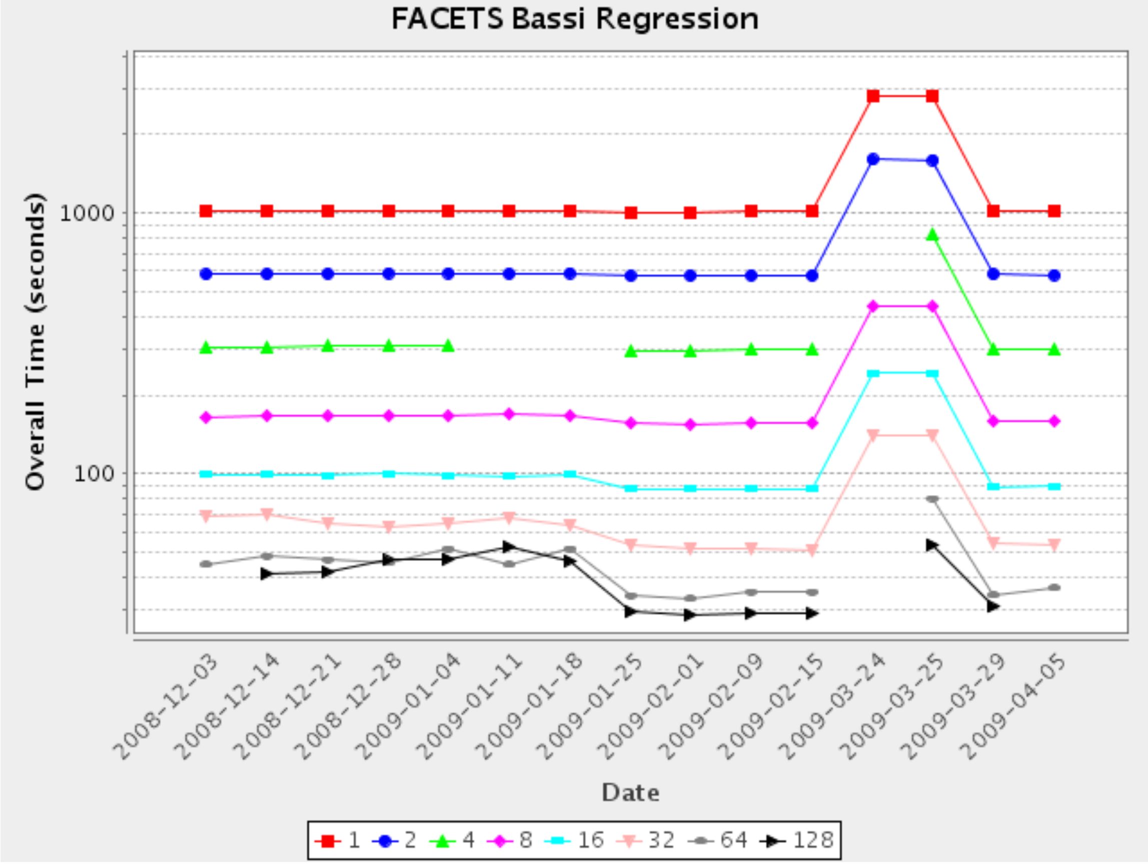

Figure 2: Performance regression testing of the FACETS Core Solver on Bassi at NERSC.

The ParaTools team executed the previously instrumented NUBEAM on Franklin, the Cray XT4 system at NERSC. Through measurement and analysis, the ParaTools team identified key areas (duplicate work, unnecessary synchronization) and worked with developers to reduce these bottlenecks and improve performance. A complete set of comparison charts before and after each optimization is posted at http://www.facetsproject.org/wiki/NubeamPerformance. Figures 3 and 4 show the difference between the first and final versions of the code in execution time. Notably, both the overall performance of the code and scaling performance improved.

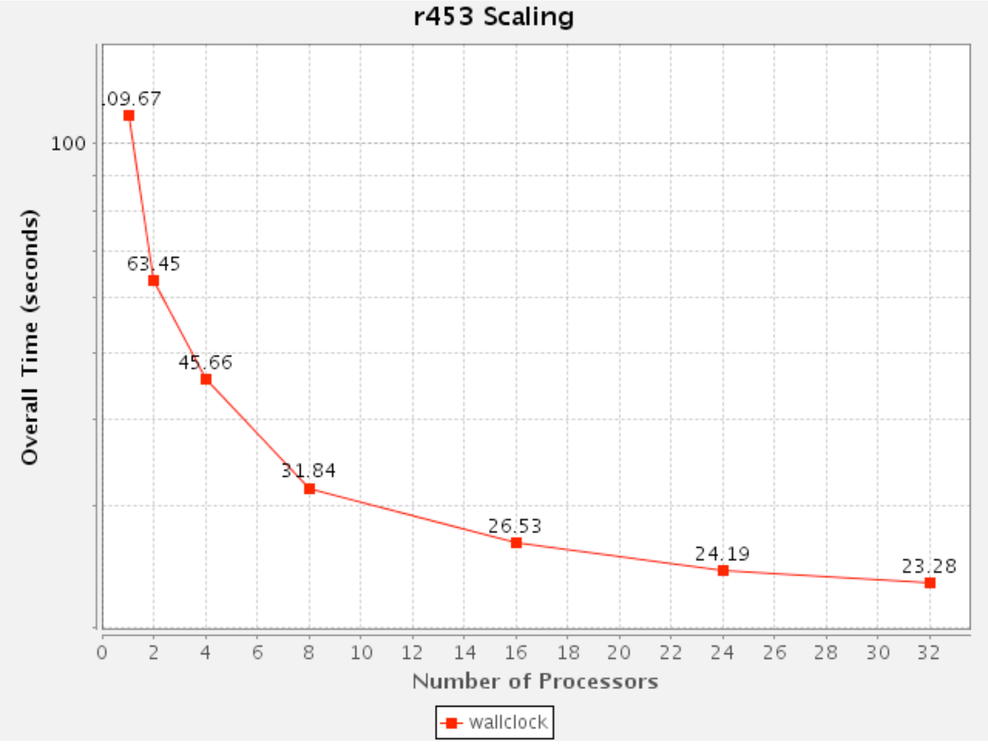

Figure 3: NUBEAM Scaling behavior with version r453 of the code.

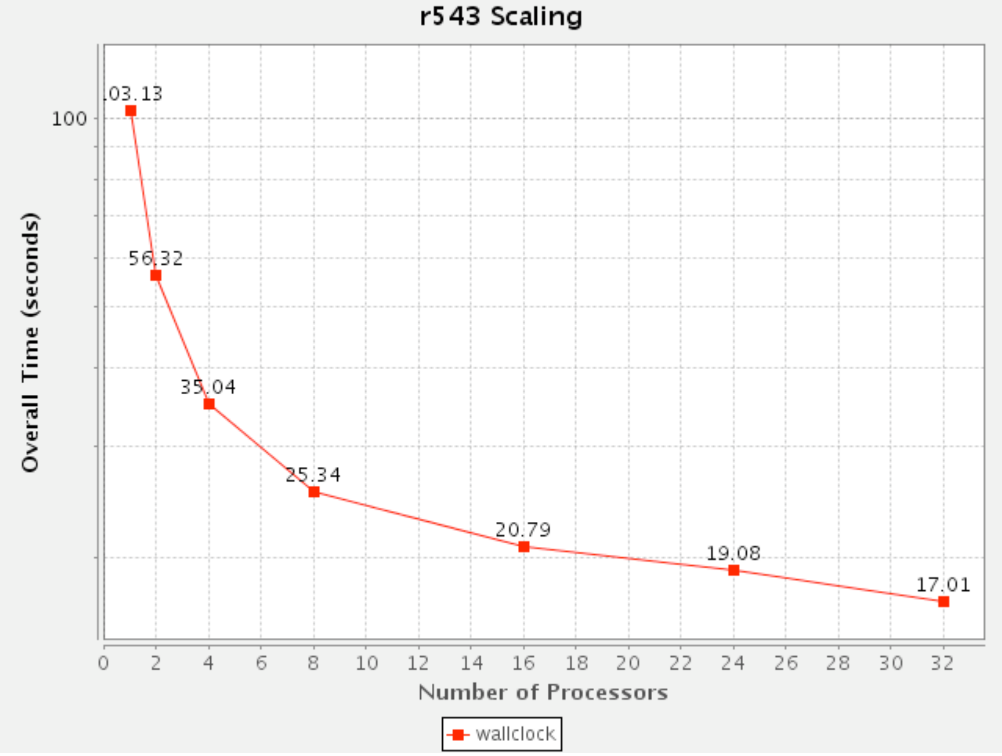

Figure 4: NUBEAM Scaling behavior with version r543 of the code.

Using the combination Python and Fortran instrumentation interface to TAU, the ParaTools team was able to generate combined performance charts showing integrated performance data in UEDGE. Complete results and instructions are posted at http://www.facetsproject.org/wiki/UedgePerformance. Figure 5 shows the performance of UEDGE when executing the test case “rdftest4_1”. Nearly all time in the tests is spent in the Fortran code, so it was determined that detailed measurement of the Python code was unnecessary – therefore the total time spent in Python is the “python” event. Loop instrumentation was used to isolate the time spent in the NEUDIF function.

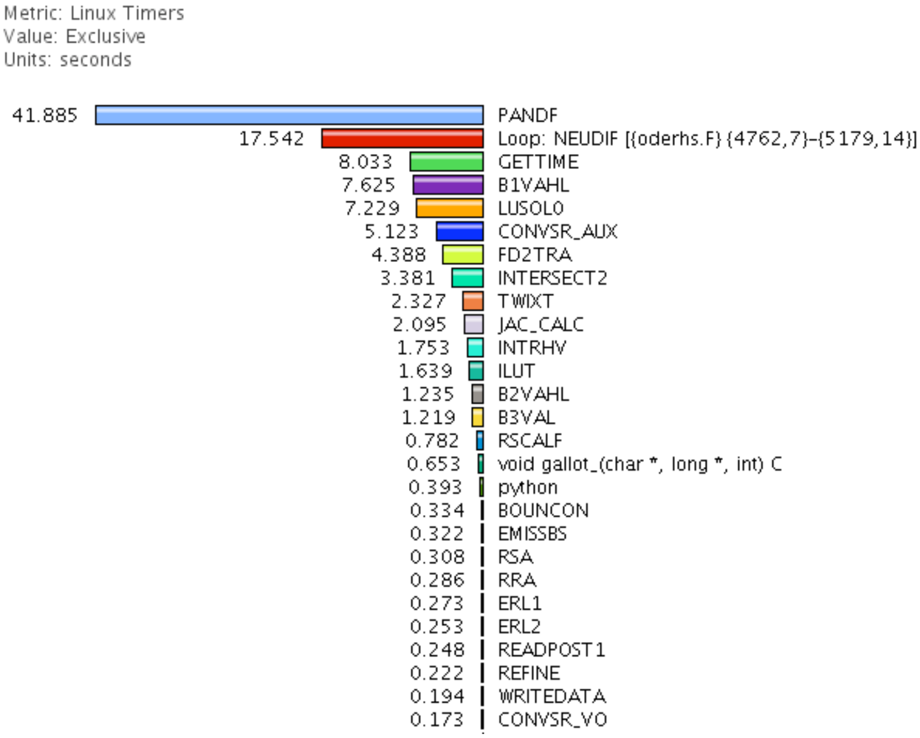

Figure 5: TAU performance profile of the UEDGE application executing the "rdftest4_1" test case.

5.3. FACETS Software Integration

The integration of the TAU performance tools into FACETS continued and TAU and PDT (Program Database Toolkit) are now part of the FACETS distribution. They are combined in a package called “metatau” that conforms to bilder (the FACETS build manager) configuration and building style. An instrumented version of FACETS is now built by default (using mkfcall.sh) on all machines. Also, a new FACETS configuration called “partau” (an extension of “par”) is now built for all machines, by default. It will generate performance data when run just the same as regular “par” version of FACETS.

5.4. Automated performance regression.

Automated weekly performance experiments with the FACETS core solver ran on Franklin, Bassi, and other clusters for over a year and a half, and results are saved in a database to allow performance regression evaluation. The weekly job checks out a fresh version of the code and builds it on the NERSC Cray XT4 Franklin system. A strong scaling study is run for the core solver on the core-astra-cmp-fmcfm input, varying the processor count from 1 to 64. The purpose of the regression analysis is to see performance trends and identify anomalous behavior. Figure 6 shows a range of results for the scaling performance. Figure 7 identifies performance effects on MPI file synchronization caused by I/O contention in the Franklin system. Performance data from the automated regression analysis can be found on the FACETS webpage: http://facetsproject.org/wiki/TAURegressionFranklin.

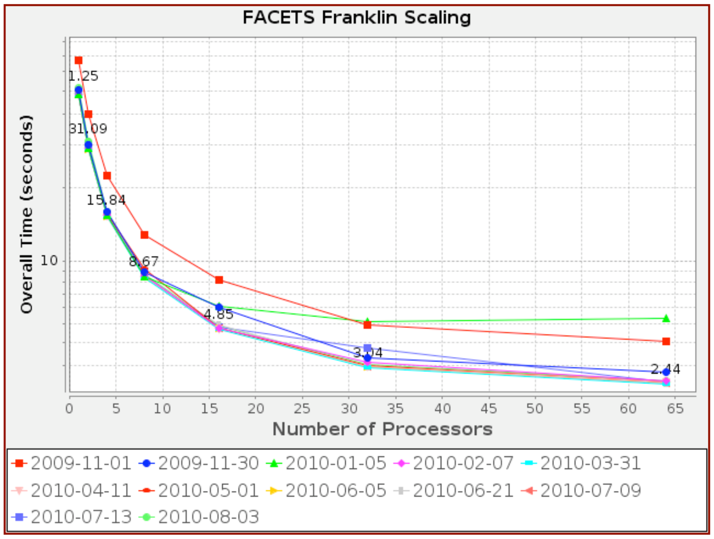

Figure 6: Scaling of the Core FACETS solver on Cray XT4.

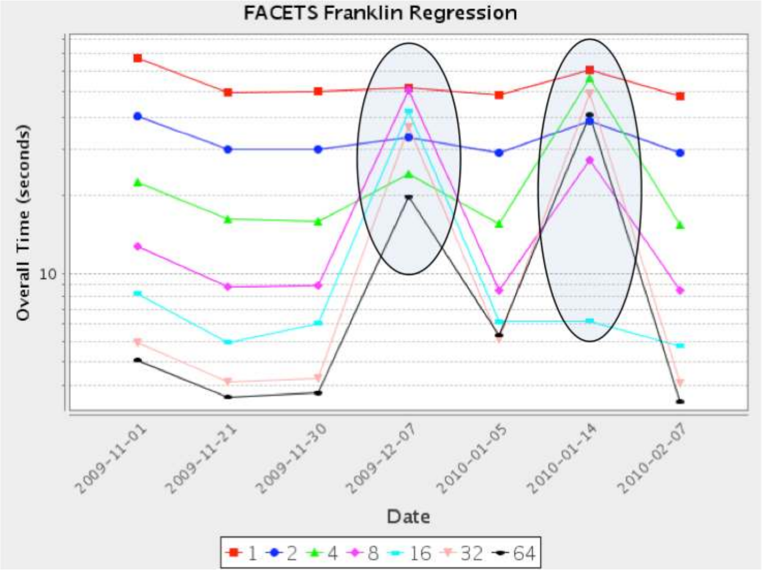

Figure 7: Performance effects on MPI file synchronization caused by I/O contention in the Franklin system.

5.5. Investigation of BG/P processor mapping.

FACETS performance on leadership-class platforms will be affected by the parallel architecture. For the BG/P, we experimented with the Gyro [26] code (on constProfiles shortScaling test cases) and different processor mapping to understand the relative effects of MPI rank assignment to nodes. The default mapping was showing poor scaling behavior. First, we tried some alternate template mappings and then a random one. After seeing the random mapping do better than the default, we surmised that by analyzing the communication patterns we could construct a better mapping. Indeed, the communication analysis paid off. The custom mapping shaves off 21% of the runtime for the 32,768-process case. Figure 8 shows the performance difference at larger numbers of processors for different mappings.

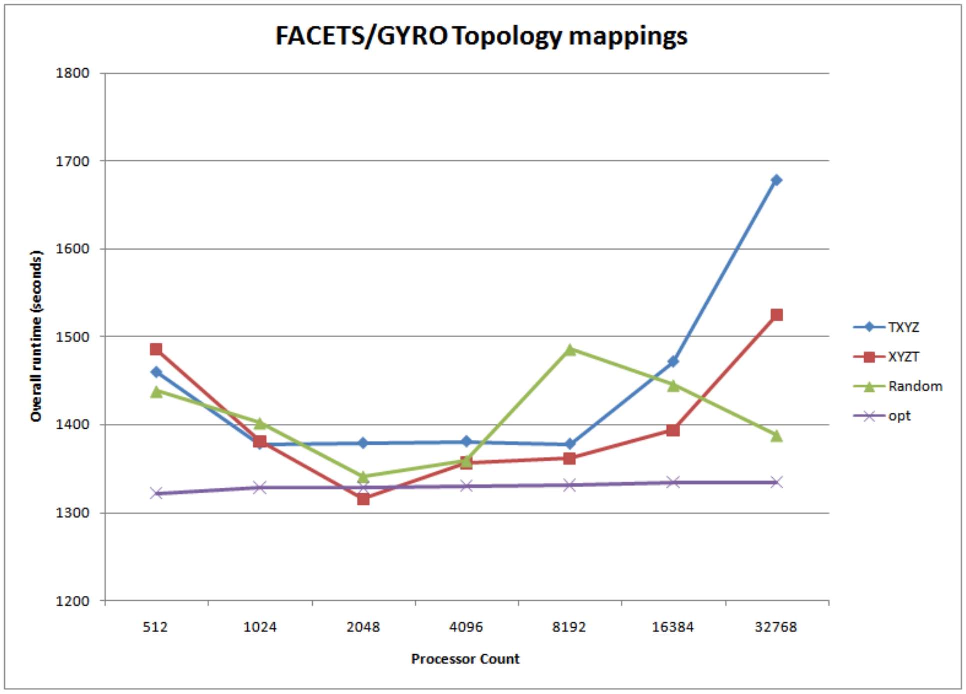

Figure 8: Scaling behavior of Gyro with different topology mappings. The custom "opt" mapping scheme shows the best performance and scaling behavior.

5.6. Testing of core-edge coupling.

Initial experiments were made with code-edge coupling using one process for each. The goal was to demonstrate instrumentation and measurement for a coupled execution, to show how performance events are captured on different components, and to observe the (simple) performance load balance. Figure 9 shows the performance of two processes, one assigned to core and one to edge. Core execution is significantly faster and waits for the edge process to finish.

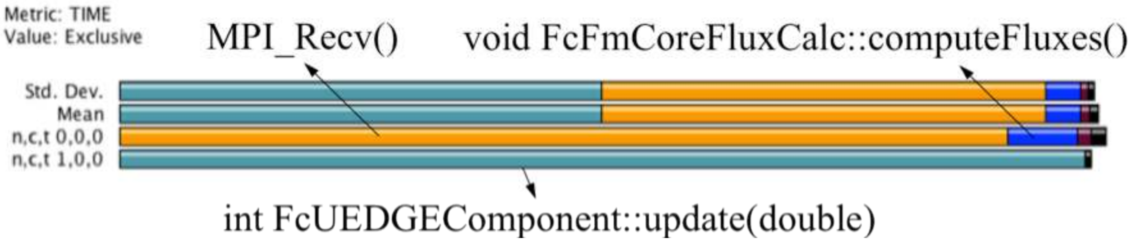

Figure 9: Core-Edge coupling behavior.

5.7. Continuing Work

The Cray XT4 system at NERSC, Franklin is scheduled to be decommissioned in April of 2012. For that reason, the FACETS project has migrated from Franklin to Hopper, a Cray XE6 system at NERSC. The ParaTools team has migrated the regression performance tests as well. Currently, porting issues related to building the complete FACETS framework have prevented full testing of the regression framework in this configuration, but when the current porting and compilation issues are resolved, the performance regression tests can resume. In the meantime, performance regression tests are performed on two clusters at the University of Oregon. The first machine, Mist, is a cluster of 16 Dell PowerEdge 1950 units with 128 Intel Xeon cores. The second machine, ACISS, is a computational cloud consisting of 196 HP ProLiant SL390 and DL580 compute nodes with various capabilities and 2682 total Intel X5650 and X7560 cores. While the performance data is not directly comparable to either Franklin or Hopper, continuing to execute the regression tests will reduce the eventual effort involved in completing the migration to Hopper. The performance database is actively maintained, and the TAU portal provides access to selected performance results.

The current FACETS build includes the latest released versions of both TAU (2.21.1) and PDT (3.17). Both the UEDGE component and the FACETS core solver have “partau” build configurations for automatically building the instrumented versions of the code.

6. Conclusions

Through the use of the TAU Performance System, ParaTools has provided instrumentation, measurement, analysis and archival support for the FACETS project. Performance optimization of key components has yielded performance speedups. TAU has been integrated into the FACETS build for both the full coupled application and the UEDGE component. The performance database has provided archival storage of the performance regression testing data generated by the project, and has helped to track improvements in the software development.

This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

7. Publications

J R Cary, J Candy, R H Cohen, S Krasheninnikov, D C McCune, D J Estep, J Larson, A D Malony, P H Worley, J A Carlsson, A H Hakim, P Hamill, S Kruger, S Muzsala, A Pletzer, S Shasharina, D Wade-Stein, N Wang, L McInnes, T Wildey, T Casper, L Diachin, T Epperly, T D Rognlien, M R Fahey, J A Kuehn, A Morris, S Shende, E Feibush, G W Hammett, K Indireshkumar, C Ludescher, L Randerson, D Stotler, A Yu Pigarov, P Bonoli, C S Chang, D A D'Ippolito, P Colella, D E Keyes, R Bramley and J R Myra, “Introducing FACETS, the Framework Application for Core-Edge Transport Simulations”, Journal of Physics: Conference Series 78 (2007) 012086, 2007.

J R Cary, J Candy, R H Cohen, S Krasheninnikov, D C McCune, D J Estep, J Larson, A D Malony, A. Pankin, P H Worley, J A Carlsson, A H Hakim, P Hamill, S Kruger, M Miah, S Muzsala, A Pletzer, S Shasharina, D Wade-Stein, N Wang, S. Balay, L McInnes, H Zhang, T Casper, L Diachin, T Epperly, T D Rognlien, M R Fahey, J Cobb, A Morris, S Shende, G W Hammett, K Indireshkumar, D Stotler, A Yu Pigarov , “First results from core-edge parallel composition in the FACETS project”, Journal of Physics: Conference Series 125 (2008) 012040, 2008.

J R Cary, J Candy, J Cobb, R H Cohen, T Epperly, D J Estep, S Krasheninnikov, A D Malony, D C McCune, L McInnes, A Pankin, S Balay, J A Carlsson, M R Fahey, R J Groebner, A H Hakim, S E Kruger, M Miah, A Pletzer, S Shasharina, S Vadlamani, D Wade-Stein, T D Rognlien, A Morris, S Shende, G W Hammett, K Indireshkumar, A Yu Pigarov, H Zhang, “Concurrent, Parallel, Multiphysics Coupling in the FACETS Project”, Journal of Physics: Conference Series 180 (2009) 012056, 2009.

8. References

[1] The FACETS Project, http://www.facetsproject.org, 2012.

[2] ITER – The Way to New Energy, http://www.iter.org/, ITER Organization, 2012.

[3] Tech-X Corporation, http://www.txcorp.com/, Tech-X Corporation, 2012.

[4] Office of Science, http://science.energy.gov/, United States Department of Energy, 2012.

[5] Advanced Scientific Computing Research (ASCR), http://science.energy.gov/ascr/, United States Department of Energy, 2012.

[6] Fusion Energy Sciences (FES), http://science.energy.gov/fes/, United States Department of Energy, 2012.

[7] ParaTools, Inc., http://www.paratools.com, ParaTools, Inc., 2012.

[8] S. Shende and A. D. Malony, “'The TAU Parallel Performance System,” International Journal of High Performance Computing Applications, Volume 20

Number 2 Summer 2006. Pages 287-311.

[9] K. A. Lindlan, J. Cuny, A. D. Malony, S. Shende, B. Mohr, R. Rivenburgh, C. Rasmussen. “PDT: A Tool Framework for Static and Dynamic Analysis of Object-Oriented Software with Templates”, Proceedings of SC2000: High Performance Networking and Computing Conference, Dallas, November 2000.

[10] DynInst, http://www.dyninst.org/dyninst, University of Wisconsin, University of Maryland, 2012.

[11] PAPI, http://icl.cs.utk.edu/papi, 2011.

[12] R. Bell, A. D. Malony, and S. Shende, “A Portable, Extensible, and Scalable Tool for Parallel Performance Profile Analysis”, Proc. EUROPAR 2003 conference, LNCS 2790, Springer, Berlin, pp. 17-26, 2003.

[13] Vampir – Performance Optimization, http://www.vampir.eu/, 2012.

[14] Paraver, http://www.bsc.es/paraver, Barcelona Supercomputing Center, 2012.

[15] K. A. Huck, A. D. Malony, R. Bell, and A. Morris, “Design and Implementation of a Parallel Performance Data Management Framework,” Proc. International Conference on Parallel Processing (ICPP 2005), IEEE Computer Society, 2005.

[16] K. A. Huck, A. D. Malony, S. Shende, and A. Morris, “Knowledge Support and Automation for Performance Analysis with PerfExplorer 2.0”, Large-Scale Programming Tools and Environments, special issue of Scientific Programming, vol. 16, no. 2-3, pp. 123--134. 2008.

[17] NTCC Software Catalog, http://w3.pppl.gov/rib/repositories/NTCC/catalog/, Princeton Plasma Physics Laboratory, 2012.

[18] TRANSP, http://w3.pppl.gov/transp/, Princeton Plasma Physics Laboratory, 2012.

[19] Center for Simulation of RF Wave Interactions with Magnetohydrodynamics, http://cswim.org/, 2012.

[20] T.D. Rognlien, J.L. Milovich, M.E. Rensink, G.D. Porter, “A fully implicit, time dependent 2-D fluid code for modeling tokamak edge plasmas”, Journal of Nuclear Materials, Volumes 196–198, 1 December 1992, Pages 347-351, ISSN 0022-3115, 10.1016/S0022-3115(06)80058-9.

[21] TEMPEST: Kinetic Simulation of Boundary Plasma Turbulent Transport, http://www.mfescience.org/Tempest/index.html, Lawrence Livermore National

Laboratory, 2012.

[22] C. Chang, S. Ku, and H. Weitzner, “Numerical Study of Neoclassical Plasma Pedestal in a Tokamak Geometry”, Phys. Plasmas 11, vol. 2649, num. 2667, 2004.

[23] Towards Optimal Petascale Simulation, http://www.scalablesolvers.org, TOPS, 2012.

[24] NTCC NUBEAM Module, http://w3.pppl.gov/ntcc/NUBEAM/, Princeton Plasma Physics Laboratory, 2012.

[25] National Energy Research Scientific Computing Center, http://www.nersc.gov/, US Department of Energy Office of Science, 2012.

[26] The General Atomics TGYRO Code Suite (GYRO/NEO/TGYRO), https://web.gat.com/theory/Gyro, General Atomics, 2012.