BLUE WATERS
PETASCALE COMPUTING SYSTEM

NEIS-P^2 DIRECT FUNDING SUPPORT FINAL REPORT

Last Updated: 11.05.2014

NSF OCI-07-25070

Leadership-Class Scientific and Engineering Computing: Breaking Through the Limits

1. Introduction

The *Blue Waters' NCSA/UIUC Enhanced Intellectual Services for Petascale Performance (NEIS-P^2)* is a program that brings together Science & Engineering Team members, external experts, NCSA staff, Illinois faculty and students, and Cray/NVIDIA staff to enhance the functionality and performance of existing petascale applications; help the migration of other applications to petascale computers; and expand the petascale-ready community.

The NEIS-P^2 activities address major challenges that stem from the increased reliance on the use of parallelism and locality to achieve performance goals. These challenges range from scaling applications to large core counts on general-purpose CPU nodes, including making enhanced use of topology information, effectively using throughput-oriented numerical computing devices, using general-purpose and accelerated nodes of a highly parallel, heterogeneous system in a single, coordinated simulation to enhancing application flexibility for increased effective, efficient use of systems and efficient use of storage and I/O resources.

The program is organized in the following three major components. (1) The project provided direct funding and expert support to the PRAC Science & Engineering Teams that were approved to use Blue Waters to enable them to realize the full potential of the Cray XE6/XK7 system. (2) NEIS-P^2 supported activities to enable the general computational science and engineering community to make effective use of petascale systems such as the NSF Track 2 and Track 1 systems. (3) NEIS-P^2 facilitated the creation of new methods and approaches that will dramatically improve the ability to do sustained science on petascale plus systems. An additional program goal was to help train the next generation of computational experts through a coordinated set of courses, workshops and fellowships.

2. NEIS-P^2 Components

The NEIS-P^2 plan, initially presented to NSF in March 2012, was resubmitted to NSF in August 2012 per the recommendation of the review panel with a revised, more concentrated focus on the short term and long-term needs of the project. The three components of the plan are summarized below.

Direct Funding Support to PRAC Teams

This activity consisted of direct funding support to the PRAC science teams to re-engineer their applications, develop new algorithmic methods, and expand their use of heterogeneous computing to take better advantage of the Cray hybrid XE/XK architecture. The major objectives of this near term phase were to assist and support science teams in scaling their applications to use significant portions of Blue Waters and to use the innovative hardware and software technologies provided by the Blue Waters system, and to accelerate the likelihood of science success for NSF science teams. The intent was not only to help individual science teams but also to identify, develop and disseminate new methods and algorithms to benefit the wider community.

Community Engagement

The second component of the NEIS-P^2 program is aimed at establishing an expanded Community Engagement project that engages researchers, educators, HPC Center staff, campus staff, and undergraduate and graduate students across all fields of study. The Community Engagement Project is working with the entire computational science and engineering community to prepare current and future generations of scientists and engineers to make effective use of petascale systems as well as potentially provide insights for others systems up and down the scale (e.g. Track 3-scale systems, Track 2 systems, other resources available to the NSF community).

This multi-year effort is focusing on workforce development, encouraging additional science teams to apply for allocations on the Blue Waters system now that it is in full service operation, and on broadening

participation and expertise in petascale computing. This effort will help raise awareness of the Blue Waters system, its benefits to advancing science, engineering and education, and help build a larger and more diverse petascale research and education community.

Petascale Application Improvement Discovery (PAID)

The third component is the Petascale Application Improvement Discovery (PAID), which will facilitate the creation of new methods and approaches that will dramatically improve the ability to achieve sustained science on petascale systems. It will include external experts, NCSA staff, Illinois faculty and students, Cray/NVIDIA staff, academic partners, national laboratories, and software providers. The PAID will focus on four general components: "Application Functionality & Performance Using Accelerators", "Application Flexibility: Topology and Load-balancing", "MPI-based programming models" and "IO and Data Movement", which are based on analysis of subject matter and methods used by the component 1 projects. By providing new methods and technological insights, PAID will, at the same time, assist the general computational science community in making effective use of systems at all scales. See the separate document on the NEIS-P^2 program and PAID in particular.

3. NEIS-P^2 Component 1 – Direct Support to PRAC Teams

3.1. Summary of Proposal Activities, and the Submissions and Awards Process

The implementation plan for the Component 1 of the NEIS-P^2 program was articulated in a change request (CR-064) that was approved by NSF in February 28, 2012. Funding was to be provided to all existing PRAC teams that would be interested in participating in this effort. The proposed budget was set at $75K per PRAC team (approximately ½ a Post-Doctoral FTE), which translated into a total direct funding of $1.95M.

The Blue Waters project Office developed a template for the individual scope of work, centered around four major deliverables:

- First Quarter deliverables and meeting – Focus: Initial Design
- Second Quarter deliverables and meeting – Focus: Initial prototype
- Third Quarter deliverables and meeting – Focus: Prototype implementation
- Fourth Quarter deliverables and meeting – Focus: Final implementation

After CR-064 was approved, the Blue Water Project Office sent an invitation on March 13. 2012 to all PRAC Principal Investigators asking them to submit a statement of work in one of the areas:

- Scaling applications to large core counts on general-purpose CPU nodes and large scale storage.
- Effectively using accelerators, e.g., NVIDIA GPUs or Intel MICs.
- Using general purpose and accelerated nodes in a single, coordinated simulation.
- Enhancing application flexibility for more effective, efficient use of systems.

Twenty one PRAC teams responded to the invitation. The Blue Waters Project Office, the Science and Engineering Application Support (SEAS) team and the UIUC Office of Sponsored Programs and Research Administration (OSPRA) worked with the 21 PRAC teams who applied for NEIS-P^2 BW funds to refine the Statements of Work and develop their budgets. Several PRAC teams decided to spit the work and funding between two collaborating institutions. This resulted in 25 new sub-agreements.

The subawards were submitted for approval to NSF. NSF approved these plans on May 1, 2012. The approval process also included formal Fastlane submissions for adding the PRAC institutions as Blue Waters subawardees. Upon receiving NSF's approval on June 28, 2012 to issue additional sub-awards, OSPRA coordinated with the various institutions to execute the agreements. The time to execute these agreements varied greatly from institution to institution, with the earliest subaward executed on July 15 and the last one on November 29, 2012. The delays in executing the subawards had implications of the work schedule for the respective teams.

3.2. Managing the Program and Tracking Deliverable Progress

The NCSA has a staff member in the SEAS group specifically assigned to each of the PRAC teams known as that PRAC's Point Of Contact. The PRAC teams are encouraged to contact their POC for questions or problems they have with the Blue Waters system and bring them any concerns they might have. Staff members in the SEAS group are typically the POC for two or three PRAC teams, but since those staff members have expertise in HPC systems and parallel programming, they have an immediate contact that can help them with their optimization efforts as part of NEIS-P^2.

The progress of the NEIS-P^2 PRAC projects was tracked through internal deliverables, which were subject to a formal certification process to ensure quality and accountability. The NEIS-P^2 deliverable reports were submitted directly to NCSA via their internal wiki. The reports were initially reviewed by the POC and then approved by someone else within the NCSA staff to ensure that the reports met guidelines. The certification of milestones and deliverables not only ensures that work is completed properly; it also contributes to intra-project communication.

3.3. NEIS-P^2 Research Symposium

The NCSA held a Symposium[1] for presentations of the NEIS-P^2 efforts and results on May 20 and 21, 2013 at NCSA (https://bluewaters.ncsa.illinois.edu/web/portal/symposium-may-2013). Some member of each of the PRAC teams attended and presented progress in their NEIS-P^2 efforts. Those reports constituted the Stage 4 progress reports for that work.

The goal of the Symposium was to encourage dissemination of the work performed as part of the funding, but also to encourage conversation between the participants in the program and with members of the HPC community. The presentations were made available to the general public on the Blue Waters portal. The following table shows the number of downloads for each presentation made available on the portal for the period of May 20/21 to November 5th, 2014. The rate of downloads has been anecdotally observed to be fairly constant over time. The range in apparent popularity across the presentations has not been examined but demand for the Klimeck, Wang, Wilhelmson and Karimabadi teams work is nearly 2x the average download rate of the rest of the teams. The Schulten presentation did not appear as available for download till a much later date due to the heavily animated presentations contained. The Woodward presentation was not made available.

Table 1. Downloads of PRAC Symposium presentations from May 20 to November 5, 2014.

PRAC PI	Downloads
Bartlett	188
Campanelli	274
Cheatham	297
Diener	365
Gropp	225
Jordan	106
Karimabadi	407
Klimeck	515
Mori	117
Nagamine	272
O'Shea	235

[1] https://bluewaters.ncsa.illinois.edu/web/portal/symposium-may-2013

Randall	221
Schulten	108
Sugar	271
Voth	192
Wang	505
Wilhelmson	391
Woodward	not posted
Yeung	285

Following the workshop, the PRAC teams submitted their final reports (Milestone 5). Upon review these reports were made publically available on the Blue Waters portal (https://bluewaters.ncsa.illinois.edu/neis-p2-final-reports)[2].

4. Results

The results of the program are not limited to the new methods, new implementations or improvements to applications as documented in the presentations and reports. A significant result of the program is knowledge exchange where researchers in different fields of science and engineering recognize common problems and engage in discussions that would not have normally taken place. A particular example is the Karimabadi team and their recent conversations with the Yeung team's efforts with large-scale 3D FFTs. The two teams started a collaboration in which they plan to work with optimized FFT libraries for each of their own applications.

The individual PRAC group final year-end reports are posted on the Blue Waters portal. Each PRAC team also has an executive summary below, for details of each team's work please refer to the full technical reports online.

Bartlett - Super instruction architecture for Petascale Computing

The Bartlett team used the domain specific programming language Super Instruction Assembly Language (SIAL) to facilitate porting the electronic structure computational chemistry code ACESIII to GPU platform on Blue Waters supercomputer. Due to the architectural design of the Super Instruction Architecture, only low-level computational kernels, or super instructions, had to be rewritten. The rest of vast science code written in SIAL required only the addition of high-level directives to indicate the computationally intensive parts of the code that should be done on a GPU. If a GPU is not present, the conventional CPU route will be executed. As a result, the new ACESIII code runs without errors on nodes that do not feature GPU accelerators. Speedups on Blue Waters from utilizing GPUs relative to CPUs in the range of 2.0-2.2 for a CCSD calculation and from 3.4-3.7 for CCSD(T) have been obtained.

Campanelli - Computational Relativity and Gravitation at Petascale: Simulating and Visualizing Astrophysically Realistic Compact Binaries

The primary goals of the project are to: 1) improve the runtime efficiency and load balance of Harm3d simulations; 2) parallelize Bothros with OpenMP and add new infrastructure for distributing I/O and data processing effort over several nodes; 3) implement OpenMP throughout GRHydro and evaluate its performance on Blue Waters. One of the earliest accomplishments was finishing implementing OpenMP directives in GRHydro, which was done in collaboration with the EinsteinToolkit consortium. As the research emphasis shifted more to Harm3d calculations, the team placed a higher priority on

[2] Two final reports are still missing (Columbia University and LSU). The Blue Waters Project Office has been in touch with the PIs (Dr. Ken Nagamine and Dr. Peter Diener) and the offices of sponsored research at these institutions regarding the outstanding obligations for these subawards.

accomplishing the Harm3d-related tasks. The Campanelli team designed, wrote, and tested a new load-balancing algorithm for distributing the computational effort more evenly over a simulation's processes. The algorithm decomposes the global domain non-uniformly, meaning that different processes are responsible for different numbers of cells. Using simulated per cell cost distributions, the load balancer yields a speedup of at least 1.5 to 2.5 times that of the existing, uniform distribution method. Further, it accommodates using an arbitrary number of MPI tasks, instead of having to use a number of processes that can evenly divide the entire domain's number of cells. The team is currently engaged in merging the stand-alone load balancing code with Harm3d and has made substantial changes to the code to use dynamic memory allocation instead of static memory allocation, and has verified that this new dynamic memory allocation code reproduces the same results as the original version of the code and scales as well---if not better---on Blue Waters than the original code. Once the load-balancing infrastructure is merged into Harm3d, full-scale simulations will be executed to verify its performance. The team has also performed extensive strong and weak scaling experiments on Blue Waters, finding that they can efficiently use up to---at least---50,000 cores at one time. Subroutines from Harm3d have been tested with OpenMP and the results imply that threading will not yield as profound a speedupas the load balancer. For this reason, the effort of adding OpenMP into Harm3d has been postponed until after the load-balancing infrastructure is in place.

Cheatham - Hierarchical molecular dynamics sampling for assessing pathways and free energies of RNA catalysis, ligand binding, and conformational change

The objective of this work is to enhance the AMBER code to support Blue Waters like system architectures. The proposed work involves (1) implementing and validating a multi-dimensional replica exchange methodology within PMEMD with a focus on optimized performance initially across a large set of GPU nodes and ultimately coordinated with the general purpose nodes, (2) validating the performance/efficiency and the ability to better sample or converge faster with the accelerated MD (aMD) code integrated into PMEMD followed by coupling aMD into the multi-dimensional replica exchange so that exchange can occur between different levels of acceleration, and (3) extending and simplifying the trajectory analysis through extensions to the CPPTRAJ program to facilitate the handling and workflow in interpreting the big data that will result from the large-scale simulations.

PMEMD was chosen for AMD+H-REMD modification as it has a significant speed advantage over SANDER. Currently only the CPU version of PMEMD works for AMD+H-REMD. M-REMD has been implemented in the Amber MD engines SANDER and PMEMD and makes use of the existing T-REMD and H-REMD exchange functions. Exchanges are attempted in each dimension in turn so that the first exchange is attempted in the first dimension, the second exchange is attempted in the second dimension, and so on. Amber NetCDF formats for both the trajectory and restart formats have been extended to support M-REMD and to aid in post-processing trajectories. CPPTRAJ has been modified to read M-REMD trajectory files.

The team was able to port the code and successfully run simulations on Blue Waters system. The various software packages enhanced through this work include: SANDER and PMEMD engines, the analysis software CPPTRAJ. Enhancements include the development of a multi-dimensional replica exchange molecular dynamics simulation method, changes to the Amber NetCDF trajectory and restart formats necessary to implement these methods, and modifications to the analysis software CPPTRAJ to recognize and process these coordinate files. These modifications are available in the current GIT master branch and will be made available to the community through the next official release.

Deiner (Schnetter) - From Binary Systems and Stellar Core Collapse To Gamma-Ray

To improve the scaling and performance of Cactus, an award-winning platform for physics simulations, the team developed and implemented in the sub-award project the following: (1) a new octree data structure to enable fast regridding in Carpet, (2) a new load balancing algorithm to improve load balancing in the regridding process in Carpet, and (3) CaKernel, a programming framework to support GPGPU in Cactus. The new data structure in (1) helps to keep track of refined points in Carpet. Here, the team designed and implemented a novel data structure based on storing the discrete derivative of the refined regions formed by the refined points from different refinement levels. It reduces e.g. a single cuboid to its eight corner points, and remains highly efficient for the grid structures typically encountered

in Cactus. This data structure employs sweeping algorithms and dimensional recursion for set operations on the refined regions. Using this, the group was able to improve scalability of the infrastructure by more than an order of magnitude, and can now run with up to 16k cores with mesh refinement and more than 260k cores on unigrid domains. The new load balancing algorithm in (2) improves load balance in the regridding for Carpet. It has been developed and implemented to change the way Carpet domain decomposes the new grids. Currently, the new algorithm is only used in fixed mesh refinement, where significant speed improvements have been observed in some case. This algorithm is significantly slower than the old one, and needs further improvements for it to be in full adaptive mesh refinement runs. The newly designed and implemented tool CaKern mentioned in (3) is a programming abstraction in the Cactus framework to enable automatic generation from a set of highly optimized templates to simplify code construction. It makes it easy to write and execute computational kernels, as well as to optimize the kernels without changing the kernel code itself. To solve the Einstein's equation using CaKernel on GPUs, the Diener team took a high level description of the equation in a Mathematica script as the only input, and then used the Kranc code generation package to generate CaKernel code. They observed that the GPU code from CaKernel runs about twice as fast as the fully optimized CPU code. Effort is on-going to integrate CaKernel with newer version of Carpet.

Gropp - System Software for Scalable Applications

This project aimed at improvements to MPI to enable various optimizations including topology aware mapping of MPI processes on the Gemini network of Blue Waters, optimization of NAMD/Charm++ over MPI on Blue Waters, and integration of GPU and MPI related data movement for performance and programmer convenience.

Current HPC systems utilize a variety of interconnection networks, with varying features and communication characteristics. MPI normalizes these interconnects with a common interface used by most HPC applications. However, network properties can have a significant impact on application performance. The team explores the impact of the interconnect on application performance on irregular/anisotropic networks, such as the Cray Gemini network on Blue Waters, which provides twice the Y-dimension bandwidth in the X and Z dimensions.

On systems with GPUs, such as Blue Waters, current hybrid programming models require the user to explicitly manage the movement of data between host, GPU, and the network, which is both tedious and inefficient. The team developed a unified programming interface, MPI-ACC that provides a convenient and optimized way of end-to-end data communication among CPUs and GPUs. This support brings users highly optimized data transfers involving GPU memory spaces while insulating them from its development complexity. Both contiguous and non-contiguous (data types) data transfers are fully supported. Team is currently working on its integration into the MPICH code.

Jordan - Petascale Research in Earthquake System Science on Blue Waters (PressOn)

The Jordan team is using the Blue Waters system to improve their estimates of potential seismic hazards and their characteristics for numerous sites across southern California. They ran simulations at frequencies up to 5 Hz using three different codes: their finite-difference AWP-ODC, finite-element Hercules, and their newest CUDA-based AWP-ODC-GPU codes. Using the combined capabilities of Blue Waters, XSEDE resources, and local resources, they were able to explore alternative representations of California geological models, including two SCEC Community Velocity Models, CVM-S and CVM-H, as well as statistical augmentations to these deterministic models to represent small scale, near-surface heterogeneities. They completed their first survey of southern California sites and seek to double the resolution of their seismic hazard predictions with a second study to be performed exclusively on Blue Waters.

Karimabadi - Enabling Breakthrough Kinetic Simulations of the Magnetosphere via Petascale Computing

The Karimabadi team is using Blue Waters to study the Earth's magnetosphere, the understanding of which is necessary for the accurate prediction of space weather. As their large simulations can require tens of thousands of nodes, the performance of their algorithms and codes is of the utmost importance, and was thus the focus of their NEIS-P^2 project. The three main goals of their project were to develop a

scalable Poisson solver needed for a semi-implicit differencing algorithm, to explore the use of higher order particles to reduce numerical heating and improve energy conservation, and to develop a GPU implementation of their particle code. They obtained excellent results in all three areas. Their new Poisson solver scales to ~10,000 cores instead of only ~500, and their new semi-implicit algorithm should allow them to run up to 28x faster than their current code, after all optimizations are complete. Use of higher order particles greatly improves energy conservation at the cost of a relatively small increase in run time. And their GPU code runs several times faster than the CPU version. All of these results are useful enough that the team is incorporating the techniques in their production codes. Additionally, they plan to submit at least three papers based on their findings. For further details and benchmarking results, see the full report online.

Klimeck - Accelerating Nano-scale Transistor Innovation

The Klimeck team is using Blue Waters to study nano-electronics. Their original proposal called for porting their code, NEMO5, to GPUs, but this objective was dropped due to a lack of funding. The NEIS-P^2 program, however, has reinstated that opportunity. Their project consists of porting NEMO5 to the Blue Waters XK7 Kepler GPUs, followed by the addition of a load balancer that distributes work between both the CPU and GPU on each XK7 node to maximize the use of all available resources. So far, the team has struggled with their initial GPU port. They encountered unexpected difficulties with their customized PETSc library build. NEMO5 is unique in that it links to two PETSc builds. Significant progress was made such that the necessary components and libraries (MAGMA) that NEMO5 requires for running on GPUs have been compiled successfully on Blue Waters, and the team has successfully built a NEMO5 GPU executable. NEMO5 was able use the PETSc-MAGMA interface but for only double or complex PETSc builds so a change was made to use MAGMA directly from NEMO5. The PETSc-MAGMA will be part of the PETSc 3.5 release and will benefit other PETSc users.. Performance of LU factorization and linear solves using MAGMA were compared to PETSc builtin and MUMPS respectively.. Design issues in the load balancer have been resolved with MAGMA support for multiple process access to the GPU. For additional details on their accomplishments and plans, see the full report online.

Lamm - Computational Chemistry at the Petascale

The Lamm team ported parts of the popular quantum chemistry code GAMESS US to C++ and CUDA on Blue Waters to take full advantage of both CPU and GPU architectures. The new C++ routines were first optimized for CPU architectures, producing novel algorithms, and then the code was ported to GPU accelerators. The development produced Rys polynomial two-electron integrals, Hartree-Fock (HF), second order perturbation theory (MP2), and coupled cluster theory (CCSD(T)) for closed-shell formalism optimized for the GPU platform. The obtained speed-up of the HF C++ CUDA code was 2 - 3 fold against the CPU-only FORTRAN-77 code. The new codes were officially released in the May 2013 release of the GAMESS US package.

Due to staffing problems, the SOW was modified (CR-076) descope the tasks named "Evaluation of GPU-based solvers" and "Conditional compilation". The scope of the primary task "Hybrid GPU/CPU quantum chemistry routines" was expanded to include code for two-electron integrals. The budget was reduced accordingly.

Mori - Petascale plasma physics simulations using PIC codes

The Mori PRAC team is dedicated to using the Blue Waters system to explore new particle-in-cell (PIC) algorithms that will take advantage of the unique hardware configuration on Blue Waters. They obtained a NEIS-P^2 allocation to fund work to : (1) Explore the use of SSE/AVX extensions to speed up single node performance of the OSIRIS code, and (2) Extend the development of their MPI/GPU PIC framework (UPIC). The team has been able to run their code at the full scale of the machine and achieve > 1 PF performance. Their performance and analysis are in the full report online.

Nagamine - Peta-Cosmology: galaxy formation and virtual astronomy

In the field of cosmological galaxy formation, the load balancing and scaling of cosmological hydro codes to large core counts have been one of the major problems. To alleviate these problems, the team

developed the HECA (Hierarchical Ensemble Computing Algorithm) technique, and has been making a transition to utilize the zoom-in technique as a primary mode of work. In HECA, the team carries out an ensemble of zoom-in simulations concurrently, which allows them to avoid the communication overhead in computation, and the problem becomes close to `embarrassingly parallel'. The idea is that it might be faster to carry out multiple zoom-in simulations separately for many refined regions, rather than performing a full cosmological simulation for the entire volume, if the communication overhead is relatively high. The team has done some test runs with GADGET-3 on Blue Waters and a few other machines to test this hypothesis. This technique should apply equally well to both GADGET and Enzo. In the largest test run the team performed 1,909 zoom-in simulations concurrently using 61,088 CPUs with no observable slowdown of the individual runs, demonstrating that application can scale up to large processor counts using HECA. The team can achieve a higher resolution in HECA than in full-box cosmological hydro simulations, and it is now possible to scale up to arbitrarily large processor counts. By running multiple zoom-in runs concurrently, team can simulate a statistically relevant sample of galaxies at high resolution, alleviating the earlier problem of small samples for zoom-in simulations. Furthermore, because each zoom-in simulation is much cheaper than the full-box cosmological run, different physical models can be implemented and tested. The team extended the initial condition generator for the Enzo AMR code as well. The team tested a hybrid version of GADGET-3 (OpenMP+MPI), which might give 20% increase in performance that will aid in their production runs on the Blue Waters system..

O'Shea / Norman - Formation of the First Galaxies: Predictions for the Next Generation of Observatories

The Enzo team is developing a "petascale Enzo" fork of the Enzo code base called Enzo-P, using a new, highly scalable AMR framework called Cello that has been developed concurrently. Cello implements a "forest of octrees" AMR approach, and Cello is parallelized using Charm++ rather than MPI. The team is excited to have reached the milestone in Enzo-P / Cello development where Enzo's PPM hydrodynamics and PPML ideal MHD solver kernels can be applied to problems using both Charm++ for parallelism and Cello's forest-of- octree adaptive mesh refinement. Preliminary performance tests also look promising, especially given that the initial implementation errs on the side of correctness versus performance. Since the performance work has been necessarily pushed back due to software development delays, the team still has much work to do in performance measurement, analysis, and optimization. Some immediately known performance improvements include the following:

- Allow for simultaneous coarsening and refining of CommBlocks to eliminate the synchronization between phases.
- Remove or eliminate quiescence detection calls, which introduce unnecessary global synchronization and imbalance in number of messages sent.
- Do not send restricted child Block data to parent CommBlocks in p_child_can_coarsen(), since the data sent is never accessed unless all eight child CommBlocks are able to coarsen.
- Remove the need for synchronization between multiple adapt phases at startup. This is a one-time cost, but the results show that it is possible.
- Only refresh ghost zones once per cycle instead of twice. Currently twice is required since the mesh refinement criteria are based on the relative slope of the density field, which requires accessing ghost zones.
- Remove the restriction on ProlongLinear that requires an even number of ghost zones, so that Enzo-P can run with PPM's 3 ghost zones rather than the current 4.
- Support adaptive time-stepping, which will reduce computation (hence power consumption) by O(L) on average for an AMR mesh with L levels. Currently a single global timestep is used.

Randall / Stan - PRAC Collaborative Research: Testing Hypotheses about Climate Prediction at Unprecedented Resolutions on the NSF Blue Waters System

The Randall team has ported key components of the Colorado State University (CSU) Global Cloud Resolving Model (GCRM) to the Blue Waters system. With the model on Blue Waters there are two levels of parallelism to explore. The first is a coarse grain MPI based communication between computational cores. This parallelism works in conjunction with global domain decomposition. The team's results show

scaling characteristics of the Blue Waters system to 40K cores, and include comparisons with other computer systems. The second level of parallelism is a fine-scale parallelism that utilizes the NVIDIA Kepler accelerators. This loop-based parallelism directly modifies the numerical operators used within the model. The team has shown that the parallel efficiency of the accelerator strongly depends on the problem size, and has devised modifications to the model to better utilize the accelerators.

Schulten - The Computational Microscope

The NAMD NEIS-P^2 team is dedicated to getting the best performance for large-scale simulations with NAMD on BlueWaters. They worked with an RA to implement an optimized cross platform replica exchange implementation in Charm++. In addition to improving performance and facilitating the future implementation of more elaborate Replica methods, the implementation resulted in a general-purpose processor set partitioning feature added to the production release of Charm++. The resulting implementation allows replica exchange computations (8 replicas of 1 million atom STMV), using the platform specific Gemini target of Charm++, to scale efficiently from 8 to 2048 XK nodes of Blue Waters with 10x better performance at 2048 nodes than the previous MPI only implementation. Detailed performance analysis and API documentation are in the full report online.

Sugar - Lattice QCD on Blue Waters

Lattice QCD is a numerical approach to the theory of the strong interaction. Calculations in this field answer fundamental questions about the nature of matter. Massive computational resources are needed to achieve these goals, and Lattice QCD is a major HPC application. In recent years, lattice calculations have benefited from the use of GPU accelerators. To fully utilize the GPU computational resources of Blue Waters, new algorithms must be developed to extend strong scaling in lattice calculations to many hundreds or even thousands of GPUs. In particular, the development of efficient many-GPU algorithms for solving the large-scale linear systems that arise in lattice calculations is essential. In this report, we describe the implementation of domain-decomposition-based linear solvers for GPU calculations using the HISQ lattice formalism. Our NEIS-P^2 funded work focused on the implementation of domain-decomposition-based linear solvers for GPU Lattice Quantum Chromodynamics (LQCD). As part of this effort, the team has implemented additive Schwarz preconditioning for the HISQ operator in the QUDA GPU library, and our code supports variable domain overlaps. Performance results obtained in multi-GPU calculations on Blue Waters are discussed in the attached Stage 5 report. The algorithms described here also serve as a starting point for the development of more sophisticated methods, which will be needed to effectively utilize future HPC technologies**.**

Tagkopolous - Evolution of intricate multi-scale biological systems

During the support period, the software suite EVE 3.0, a new version of the Evolution in Variable Environments (EVE) framework has been completed. The new parallel model includes an adaptive dynamic load balancer that is implemented based on the analysis of benchmarked prototypes presented in previous and current reports. The performance of the EVE code on the Blue Waters machine is significantly improved, and the code now scales up to 8,000 MPI processes and 128,000 organisms in a population. Every evolutionary experiment requires at least thirty-two independent technical replicate runs for statistics. Therefore, each experiment is potentially scalable up to 256,000 MPI processes.

The combined EVE code incorporates several parallel models: (a) the original model with a static load balancer; (b) a newly-developed adaptive dynamic load balancer with a non-fixed population size option; (c) AMPI compatibility; and (d) a serial version of the code for small local runs. The choice of the optimal model(s) depends on the size of a particular run and on available computational resources.

The C++ source code, compiled binaries for several standard architectures, and the user manual with samples are freely available from the project's website, along with a tutorial on how to use the tool (more information is available at http://www.tagkopolouslab.cs.ucdavis.edu/software). The EVE framework was successfully used recently in microbial evolution research and in the teaching of undergraduate students. More specifically, it was used in both Spring and Fall of 2012 in the ECS 124 "Theory and applications of bioinformatics" class, where approximately fifty undergraduate students used EVE for laboratory assignments related to microbial evolution and population diversity.

Voth - Petascale Multiscale Simulations of Biomolecular Systems

The effective use of modern supercomputing resources such as Blue Waters for coarse-grained (CG) models with implicit solvent presents significant challenges to orthodox molecular dynamics (MD) software. The Voth PRAC team created a de-novo design and implementation of a custom CG-MD code, designed to enable the simulation of cell-scale biomolecular systems using a combination of sparse memory techniques and computational load balancing via the Hilbert space filling curve (SFC). The CG-MD software is shown to demonstrate strong scaling behavior to over 130,000 CPU cores, even for a relatively small test system with extremely inhomogeneous particle distribution. Very low memory requirements are also demonstrated, indicating the CG-MD software will provide a valuable platform for future studies of highly dynamic, very large-scale biological phenomena.

Wang - Enabling Large-Scale, High-Resolution, and Real-Time Earthquake Simulations on Petascale Parallel Computers

The Wang PRAC team is dedicated to using the Blue Waters system to solve the three-dimensional elastic seismic wave equation on unstructured tetrahedral meshes and solving large-scale linear equation problems in geophysics. Their NEIS-P^2 activities are focused on the design and implemention of two scalable algorithms for heterogeneous CPU-GPU supercomputing. One is a widely-used Sparse Equations and Least Squares (LSQR) solver, called SPLSQR, which is a numerical method for solving large-scale linear equation problems in an iterative way. The other is a community earthquake simulation software "cuDGSeis" for solving the 3D viscoelastic seismic wave equation using the discontinuous Galerkin method. The team has made significant strides in performance improvement (i/o improvements of 50x with mpi-io and appropriate lustre striping). They would like to have allocation time on Blue Waters to test their improved code at larger scale. Their performance and analysis are in the full report online.

Wilhelmson - Understanding Tornadoes and Their Parent Supercells Through Ultra-High Resolution Simulation/Analysis

CM1 is a parallel a three-dimensional, non-hydrostatic, non-linear, time-dependent numerical model designed for idealized studies of atmospheric phenomena. It is being used to simulate storms and tornadoes and is the model used in this proposal that focuses primarily on high performance I/O together with inline- and post- analysis/visualization. VisIt was selected for visualization of three-dimensional spatial data on Blue Waters and Matlab for two-dimensional data. Both of these visualization tools have parallel capabilities. Originally the data to be visualized was going to be moved from the memory of a running CM1 simulation to memory on the GPU side of Blue Waters for in-line visualization using Damaris. Some work with the Damaris group demonstrated this possibility for CM1 but it was concluded that the availability and maintenance of Damaris on Blue Waters was unclear. Further VisIt performance did not significantly benefit from access to the GPUs. Therefore, near real-time visualization became the emphasis with data from a running simulation first written to Blue Waters disk using HDF5 (3D) and pHDF5 (2D). The arrival of new data on disk is monitored and when ready it is read back into Blue Waters for visualization of the 3D data on CPU nodes and sent to another system for generation of 2D visualization and web-based display of both 2 and 3D visualizations.

A VisIt plugin was developed for the near real time and post analysis/visualization of simulation data. Up-to-date visualizations is made available on web pages by adapting software built for post-analysis of WRF simulation data but that now will update as the simulation proceeds. In addition work continued with Paul Woodward's team using a simplified version of CM1 for studying the impact of restructuring advection and turbulence calculations to improve computational intensity (number of computations per memory fetch). This paper discusses current accomplishments and progress.

Woodward - Petascale Simulation of Turbulent Stellar Hydrodynamics

Paul Woodward's PRAC is using the Blue Waters system at NCSA to study compressible, turbulent mixing of gases in the deep interiors of stars and also in the context of inertial confinement fusion (ICF). They have also worked under this subcontract with a simplified version of the CM1 weather code in an attempt to apply techniques from their code to the one being used by the PRAC team led by Bob

Wilhelmson. Their work with the PPM code culminated in December 2012, during the Blue Waters friendly user access period, when they carried out a simulation of an ICF test problem on a grid of 1.18 trillion cells at a sustained performance level of 1.5 Pflop/s (in 32-bit precision). Their multifluid PPM code ran at a sustained rate of 12% of peak performance in 32-bit mode, on 702,784 cores with one thread per core, 8 threads per CPU die, 4 MPI processes per node, 87,846 MPI ranks, and 21,962 nodes.

As a part of this project, Woodward worked with NCSA to enable his team's volume rendering utility, HVR, to run on the GPUs within the Blue Waters machine. This gave them an interactive volume rendering capability for petascale data that was demonstrated at the PRAC workshop at NCSA in May, 2013.

Woodward has also been working with Bob Wilhelmson and Leigh Orf to explore the generality of the briquette-based code structure used in PPM and the massively pipelined transformed code it enables to achieve high performance. In December 2012, Leigh Orf visited Minnesota and constructed a greatly simplified version of the CM1 code he and Wilhelmson are using on Blue Waters. Woodward has so far studied a direct application of their code transformation tool to a directionally split variant of the advection scheme used in CM1. They found that they could accelerate the execution of this algorithm in the same way and to essentially the same degree as the advection scheme in PPM. Woodward has studied the sound wave step in isolation also. Their most recent work with CM1 focuses on the turbulence calculation that forms the beginning of the grid cell update, and is the most challenging and computationally complicated portion of the code.

Yeung - Petascale Computations for Complex Turbulent Flows at High Reynolds Number

In this project, the team has explored various approaches to improve the performance of their DNS code, primarily to reduce the time spent in communication. Their code PSDNS, uses a pseudo spectrum algorithm which has the computation 3D FFT's computation as a key component. As a result, global communication is needed and is the bottleneck of the code. With the help of Cray application specialist Bob Fiedler, substantial reduction in time spent in communication due to 3D FFT, has been achieved through the use of Coarray FORTRAN (CAF) in alltoall operations. Following the success of using CAF in alltoall, the team has expanded the use of CAF to other global operations such as gather and scatter for performance gain. To make further improvements, they have been investigating the use of hybrid programming model with MPI and openMP. They are also studying the effects of overlapping communication with computation using openMP in various thread safety modes, using non-blocking MPI collectives, as well as using CAF. In addition to experiment with different programming models and strategies, the team is also involved in the study on performance and performance variability in relation to node locality on Blue Waters. It is found that the best geometry for their code is for the nodes to be in a collection of contiguous sheets that are large in the x and z direction, but small in y.

Because of the substantial performance improvement achieved by Cray's implementation using CAF instead of MPI_alltoall in matrix transposes for the PSDNS code, the SOW was changed (CR-070) to build on the success of using CAF. The second proposed programming strategy in the proposal, "Overlapping computation with communication by one-sided communication" was replaced with "Overlapping computation with communication by employing a CAF module for data transpose".

5. Lessons Learned

The results of the first component of the NEIS-P^2 program are being shared with the broader community of the Blue Waters users. The popularity of the presentations and eventually the reports indicate the interest in the areas of exploration and development made available by this type of funding opportunity. The implementations, experiments and changes to applications documented in the presentations and reports are valuable to other researchers in and beyond the communities represented by the PRAC teams. The visibility of the Blue Waters project and the accessibility of the material from the Blue Waters portal will ensure that the lessoned learned by the teams will not get lost.

Beyond the tangible deliverables of the work done by the teams that participated in this opportunity, the nature of the work contains information useful in directing future efforts within the NEIS-P^2 program. A review of these Component 1 reports identified several themes common to many of the projects.

Additional examination of work by the PRAC teams with their Point of Contact (PoC) was done in light of these themes. Categorization of the work into these themes revealed the following:

- Approximately ten teams invested in algorithmic development of existing applications in the following areas: conventional load balancing, use of FFTs, adaptive mesh refinement and graph solving methods.
- About twelve teams explored, enabled or enhanced accelerator technology support in their applications using OpenACC or CUDA.
- Five teams addressed some aspect of IO performance at the framework, file system or parallel implementation levels.
- Several teams considered ways to improve scalability through the use of topology awareness along with hybrid programming models (e.g. MPI + OpenMP).
- Teams also investigated various aspects of heterogeneous computing combining aspects of accelerator use with load balancing.

The assessment of the PRAC team needs was used as a basis for the work to be done under the 3rd component of the NEIS-P^2. The above-identified themes have been condensed into four areas that have overlap: topology awareness and load-balancing (including AMR), single source-code support for accelerators and use of accelerators at scale, efficient IO and data movement, programming models using MPI.

This information was used to motivate the final component of the NEIS-P^2 effort to coordinate and support collaborative projects of technology providers and science teams in the areas of the themes listed above. The goal is to enable functionality in existing applications that will facilitate the teams using the codes to fully utilize the resources provided by the Blue Waters system and other existing or future computational resources.

The information collected from the NEIS-P^2 component 1 reports was also used to structure training workshops for the PRAC science teams and the other science teams on Blue Waters. While the level of expertise varies greatly across the teams it was clear that providing hands-on workshop training for topics such as OpenACC and hybrid programming would help for teams moving towards those technologies. Two workshops took place since the analysis of the NEIS-P^2 Component 1 reports: December 2013 and October 2014. The former focused on traditional new user topics (compilers, running jobs, general debugging) but also provided an important presentation on proper performance comparison as recommended by members of the July 2013 NSF review panel. The presentation titled Performance Report Guidelines was prepared by three CS department graduate students participating in a HPC journal group lead by Bill Kramer that has been downloaded several hundred times. The report provides both serious and comical issues in the area of correct methodologies for performance reporting and comparison. The second workshop focused on more advanced topics with hands-on access to presenters, as suggested by previous attendees, on the topics of CUDA, OpenACC, OpenMP, performance tools and debugging. A findings report from the University of Illinois I-STEM team will be included in the December 2014 quarterly report.

The program overall was able to provide funding for 21 projects in a way the allowed for groups to explore novel techniques, new approaches or to complete work that was only partially implemented. This program responded to the general sentiment in the HPC community and particularly among the PRAC awardees that more funding and effort needs to go towards software and code improvement.

As evidenced by the final reports and the May workshop, this effort has been very successful. Not only did it benefit the individual teams, but it also triggered new collaborations and exchange of knowledge between participants. In addition the program advanced the interaction and cooperation between the PRACs and the Blue Waters team.

A very successful element of the program was the face-to-face workshop organized at NCSA. It provided a stage for each individual team to showcase their research results. It also offered a forum to discuss

common challenges and ideas for addressing them and provided networking opportunities for the PRAC team members.

Due to the large number of subawards, this initiative incurred a significant management administrative overhead, especially in the startup stage (execution of the subawards). This is especially significant given the fact that this was a one-year program. In spite of a very aggressive time scale for implementing this effort and the desire to have a common starting date for all awards, some institutions were slow in responding. This translated in delays in executing the subawards and therefore in the start dates. Once the uneven start dates appeared in the individual SOW schedules, it proved difficult to catch up and realign to the master schedule. The short-duration of the project also exacerbated the challenges related to staffing at the PRAC institutions and directly impacted the work of several teams.

The quarterly deliverable process of review and feedback made us think of a self-service webpage (on the Blue Waters Portal) for future interactions with the PRAC teams. Tracking based on wiki pages and email is not scalable, and does not provide simple tools for reminders, summarizing of status, etc.