Application Software Case Studies in FY04 for the Mathematical, Information and Computational Sciences Office of the U.S. Department of Energy

November 15, 2004

K. Roche[1], J. Drake[2], P. Jones[3], D. Dean[4], J. Blondin[5], C. Ballance[6], M. Pindzola[7], C. DeTar, J. Osborn[8], R. Brower[9], H. Neff[10], B. Sugar[11]

[1]Computer Science and Mathematics Division, National Center for Computational Sciences, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6173 Oak Ridge, TN 37831-6008, U.S.A Email:rochekj@ornl.gov

[2]Computer Science and Mathematics Division, Climate Dynamics Group, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6016 Oak Ridge, TN 37831-6016, U.S.A Email:drakejb@ornl.gov

[3]T-3, Los Alamos National Laboratory, MS B216, Los Alamos, NM 87545, U.S.A Email:pwjones@lanl.gov

[4]Physics Division, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. BOX 2008, MS - 6373 Oak Ridge, TN 37831-6373, U.S.A. Email:deandj@ornl.gov

[5]North Carolina State University, Department of Physics, 2700 Stinson Drive, P.O. Box 8202, Raleigh, NC 27695, U.S.A Email:john_blondin@ncsu.edu

[6]Rollins College, Department of Physics, 1000 Holt Avenue, Winter Park, FL 32789, U.S.A. Email:ballance@vanadium.rollins.edu

[7]Auburn University, Department of Physics, 206 Allison Lab Auburn, AL 36849, U.S.A. Email:pindzola@physics.auburn.edu

[8]University of Utah, Physics Department, 115 S 1400 E Suite 201, Salt Lake City, UT 84112-0830, U.S.A Email:(detar,osborn)@physics.utah.edu

[9]Boston University, Department of Physics, 590 Commonwealth Avenue, Boston MA 02215, U.S.A Email:brower@bu.edu

[10]Boston University, Center for Computational Science, 3 Cummington Street, Boston MA 02215, U.S.A Email:hneff@bu.edu

[11]University of California Santa Barbara, Department of Physics, Broida Hall, Building

Contents

1 Problem Statement **7**

2 Software Development for the Community Climate System Model Codes in FY03 - FY04 **8**
2.1 Overview of the SciDAC CCSM Software Project 8
2.2 CCSM Software Development Path 10
2.3 Performance Metrics 12
2.4 Hardware Bottlenecks 12
2.5 Case Study: The Community Atmospheric Model 13
2.5.1 Scientific Goals and Preparation for the IPCC Simulation Project 13
2.5.2 Resolution for Regional Impacts 14
2.5.3 Hardware Example 14
2.5.4 Performance Gain 14
2.5.5 Discussion 14
2.6 Case Study: The Parallel Ocean Program 14
2.6.1 Science Goals and Eddy Resolving Simulations 14
2.6.2 Software improvements 16
2.6.3 Performance Gain 16
2.6.4 Discussion 16

3 Software Development for the Nuclear Shell Model Monte Carlo Code in FY03 - FY04 **17**
3.1 Overview 17
3.2 Physics Gains in FY04: Example 18
3.3 Monte Carlo Computation in the Nuclear Shell Model 20
3.4 Performance 21
3.5 Discussion 22

4 Software Development for the Virginia Hydrodynamics Code in FY03 - FY04 **23**
4.1 Overview 23
4.2 VH-1 at the beginning of FY04 24
4.2.1 State of the Code 24
4.2.2 State of the Data Pipeline 24

572, Santa Barbara, CA 93106-9530, U.S.A Email:sugar@savar.physics.ucsb.edu

4.2.3 Scientific Output 24
4.3 VH-1 at the End of FY04 25
4.3.1 State of the Code 25
4.3.2 Performance 25
4.3.3 State of the Data Pipeline 26
4.3.4 Scientific Output 27
4.4 Discussion 29

5 Software Development for the R-Matrix with Pseudostates Codes in FY03 - FY04 30
5.1 Overview 30
5.2 Code Structure of the RMPS codes 30
5.3 Computation and Performance for the RMPS codes for FY03 - FY04 32
5.4 Discussion 38

6 Software Development for Lattice Gauge Theory in FY03 - FY04 39
6.1 Overview 39
6.1.1 SciDAC Software Project 39
6.1.2 Lattice Gauge Theory Software Development Path . . 40
6.1.3 Performance Metrics 41
6.1.4 Hardware Bottlenecks 41
6.2 Case Study: QCD Thermodynamics 42
6.2.1 Physics Goals and Impact on RHIC 42
6.2.2 Setting the Lattice Spacing 43
6.2.3 Hardware Example 43
6.2.4 Performance Gain 44
6.2.5 Discussion 44
6.3 Case Study: B Decay 44
6.3.1 Physics goals and coordination with experiment 44
6.3.2 Parameter Choices 45
6.3.3 Hardware Example 46
6.3.4 Performance Gain 46
6.4 Discussion 46
6.5 Case Study: Lattice Generation 47
6.5.1 Why Generate Lattices? 47
6.5.2 Hardware Example 48

6.5.3 Performance Gain 48
6.5.4 Discussion 48
6.6 Prospects for Future Hardware 49
6.6.1 QCDOC 49
6.6.2 Clusters 49

7 Closing Comments and Summary 50
7.1 CCSM 51
7.1.1 Q3, FY04 51
7.1.2 Q4, FY04 51
7.2 SMMC 52
7.2.1 Q3,FY04 53
7.2.2 Q4,FY04 53
7.3 VH-1 53
7.3.1 Q3,FY04 53
7.3.2 Q4,FY04 54
7.4 RMPS 54
7.4.1 Q3,FY04 54
7.4.2 Q4,FY04 55
7.5 QCD 55
7.5.1 Q3,FY04 55
7.5.2 Q4,FY04 56

List of Figures

1 Single processor comparison over platforms for the column radiation model. 11
2 CAM 2.0 in the current fiscal year on the IBM p690 system at ORNL. Distributed and shared memory optimizations are studied here. 15
3 The evolution of performance of the POP code due to software and hardware improvements on the Cray X1 over the last year is shown. 17

4 Shown in the figure is the evolution of shape for three nuclei in the fp-gds shell model space. These nuclei are (from left to right) ^{68}Ni (very neutron rich), ^{72}Ge (stable), and ^{80}Zr (neutron deficient) at temperatures (from top to bottom) of $T = 2.0MeV, 1.0MeV$ and $0.5MeV$ (near the ground-state). Each has a fixed neutron number of $N = 40$. The evolution of shape as a function of temperature in the beta (deformation) and gamma (indicating the type of deformation, whether prolate ($\gamma = 0$), oblate ($\gamma = 60$) or triaxial ($\gamma = 30$)) plane is shown. Here, $x = \beta sin(\gamma)$ and $y = \beta cos(\gamma)$ is plotted. ^{80}Zr exhibits extreme deformation that lasts to high temperatures, while Ni is quite spherical and Ge is in between. 20
5 Snapshot of the turbulent stellar core flow beneath the supernova shock wave resulting from stationary accretion shock instability (SASI). 28
6 Code flowchart : including FY03/04 developments up to Q3 FY04 31
7 The plot on the left depicts the typical relative values and distribution of the matrix elements involved in *stg3r*. On the right, the diagonalisation of a real symmetric matrix as a function of time and processor. This corresponds to a 238 term C^{2+} calculation. Timings were carried out on Seaborg. The top graph on the right has a fixed Hamiltonian matrix size, varying the number of processors and the bottom varies the matrix size with a static number of processors. 33
8 Total radiative power loss for lithium in which the (green dots) Born approximation (Cowan code), the distorted wave (dashed, LANL code), and RMPS (Li^{2+} - 57 term (2004); Li^{+} = 101 term (2003); Li = $\sim$ 60 terms(2004)) results were used in the excitation rates for every ion stage of lithium. The question to address was whether the underlying electron impact excitation method affected the radiative power loss of lithium. The RMPS are 2 day calculations that are more computationally demanding but give a more accurate result. The simpler perturbative methods have limitations. There is up to a 44% difference even in this fairly simplistic system between DW and RMPS. 35

9 Cross sections computed for the highly ionized system Fe^{14^+} are compared for the DARC (jj coupling) and parallel Breit-Pauli (JK coupling) formalisms. 37
10 Flowchart of parallel Dirac-Fock scattering codes -changes through Q4 FY04 38

1 Problem Statement

The current report is intended to provide information on the effectiveness of the computational implementation of a handful of Office of Science applications over the duration of the current fiscal year (October 2003 - October 2004). One hopes to determine if the computational science capabilities have improved either by simulating the same problem in less time or simulating a larger or more physically realistic problem in the same time. The hardware platform is fixed unless explicitly stated otherwise. Thus, the dynamics reported for each of the applications are attributable to software, algorithm, and language / compiler changes, enhancements, and development *or* the addition of new physics.

It is noted that the particular applications investigated are all of relevance to the DOE SC missions. The level of maturity of the application codes ranges from still in development in the physics sense to having been refined for the last thirty years. The information gathered attempts to convey the science objectives of each application and how meeting these objectives has evolved this fiscal year.

One thing that is important to understand is how to pose questions that are meaningful when making inquiries such as this one. It is difficult to convey the effectiveness of software for an application by studying how well it has performed according to some hardware system observables such as flop rate or communication bandwidth or the number of processors it is able to utilize in a parallel computation. It is potentially meaningless to do so if there do not exist well defined scientific objectives that can be directly measured within a production run of the application. Understanding the coupling of such science-based metrics to effective system utilization is becoming increasingly important as hardware systems become increasingly larger and more expensive. Furthermore, efficient utilization of high performance computing systems may not be the primary goal in science cases of extreme importance to the nation's energy or security policies. Instead, it may be of greater importance to determine if the science objectives can ever be met on existing platforms by measuring scalable versions of a problem, or by determining if progress is being made at all with the advent of increasingly more powerful systems.

There do exist SC projects dedicated to measuring, modeling, and attempting to understand this complex process. Projects such as the DOE Matrix, the SciDAC Integrated Software Infrastructure Centers, and Early

Evaluation programs at our national laboratories and universities are some of these efforts. Most of these efforts are immature and the parameter space that is being interrogated is extremely complex.

It is the larger goal of DOE SC software (e.g. the SciDAC program) not to merely optimize execution efficiency of its applications but to accelerate scientific discovery. To this end, the standardization of the software infrastructure is designed to promote both the rapid development of new application codes and greater ease of porting these applications to more powerful high performance architectures as they evolve. The scientific goals can only be reached if the applications scientists are able to focus the majority of their effort on physics and not computational tools.

2 Software Development for the Community Climate System Model Codes in FY03 - FY04

2.1 Overview of the SciDAC CCSM Software Project

The Japanese Earth Simulator (GS40) has challenged the U. S. climate-modeling community and computer industry to accelerate the development of both the models and the computers required to run them, to keep our country competitive in both climate science and policy decisions. The SciDAC project Collaborative Design and Development of the Community Climate System Model for Terascale Computers has the goal of providing a performance portable, state of the science model suitable for use by the climate research community and for climate change studies in support of DOE and other agency missions. The FY2004 development has been aggressive in that it was decided to choose to support the U. S. participation in the Intergovernmental Panel on Climate Change (IPCC) with simulations of future scenarios under a variety of green house emission scenarios. This required rapid development of additional modeling capability to answer the specific science questions posed by Working Group One as well as aggressive software engineering targeting the available computer hardware so that simulation throughput and schedules could be met.

The SciDAC project Collaborative Design and Development of the Community Climate System Model for Terascale Computers had its initial fo-

cus on performance portability. As many of the software modifications and refactoring for performance portability have been completed, more time has been devoted to specific optimizations supporting the IPCC runs and toward development of new science capabilities in the modeling system. The collaborators at the National Center for Atmospheric Research, who coordinate the code development and lead the community of university researchers, have introduced a software engineering discipline to help manage the set of distributed developers and to engineer the code for maximum scientific use with excellent performance on platforms of interest. Supporting the community development process has been a high priority and allowing an open design process with contributions of entire component models by DOE laboratories (LANL in particular) has given new meaning to the interagency support to community model development.

The project has had an active role with several participants in the CCSM Software Engineering Working group. The management of CVS repositories with version control, change review boards, design documents, testing and validation procedures have required significant time and effort from all involved, but there is likely no other way to do it and the group has been very productive under tight deadlines.

The CCSM is a coupled climate system model consisting of atmosphere, ocean, land and sea ice components. Each model has been developed separately, often at different institutions, and runs in coupled mode using a coupler to exchange and regrid outputs and inputs needed for exchange between the components. The entire system is integrated in time typically for 100-200 years taking 10 minute timesteps. It is the long integration, as well as the complexity of the computation, that qualifies climate simulations as a terascale challenge.

The SciDAC Earth System Grid (ESG) project is a national infrastructure supporting the distribution and analysis of simulation model output of the CCSM. It provides for secure automatic migration of data sets between LBNL, ORNL, NCAR and LANL. The project has worked closely with the ESG as major new simulations for the Intergovernmental Panel on Climate Change Fourth Assessment simulations have started production.

The CCSM3.0 code was released in June of 2004 and is being heavily used to study the commitment scenarios of the Intergovernmental Panel on Climate Change (IPCC). Both ORNL and NERSC have joined NCAR in providing machine resources to perform the simulations. Simulations are progressing and runs should conclude in November. A special issue of the

Journal of Climate is being devoted to description of the CCSM3 and of the results being obtained. A special issue of the *International Journal of High Performacne Computing Applications* is also planned to describe the parallel algorithms and software design of the CCSM3. The documentation for these publications began in Q4 of FY04.

2.2 CCSM Software Development Path

The Community Climate System Model can be described as a physically based model of the earth's circulations and energy transport. It is based on partial differential equations and physical processes parameterized for the scale of interaction represented in a discrete representation of the atmosphere, ocean, land and sea ice. The software and simulation science is based on the following:

1. Model Equations - Nonlinearly coupled general circulation models representing atmospheric flow over realistic topography and free surface oceanic flow over realistic bathymetry. Moist air thermodynamics includes subgrid scale convective adjustments, cloud physics, full spectral columnar radiation absorption and balance. Thermohaline equations of state with ocean eddy parameterizations of diffusion. Sea ice flows with visco-plastic ice rheology. Fixed ecological and land use areas with consistent energy, biochemical fluxes and plant responses. Fresh water balance through river routing from soil hydrology.

2. Model Boundary Conditions - variable solar input, atmospheric CO2 level, volcanic aerosol concentrations, stratospheric ozone.

3. Initial Conditions - temperature, pressure, velocity, moisture and salinity from historic (or present) climatology.

4. Output - over one hundred 2-D and 3-D fields representing instantaneous states or monthly averages. The volume of output depends on resolution. Typical production runs are 1870-present (validation) and present - 2200 (scenario).

5. Resolution - Atmosphere (spectral T85 =1.4degrees, 10min timestep), Ocean and ice (1 degree).

Three dynamical cores for numerical innovation in the atmosphere are supported. Each provides a unique capability, but each also has deficiencies. The finite volume dycore is being developed to support simulations that must incorporate chemical cycles. (Please see the closing section of this document for updates about the fvd.) The desire is to add these capabilities in order to properly simulate the carbon cycle with a full complement of green house gasses as well as sulfate aerosols and air quality chemicals such as ozone.

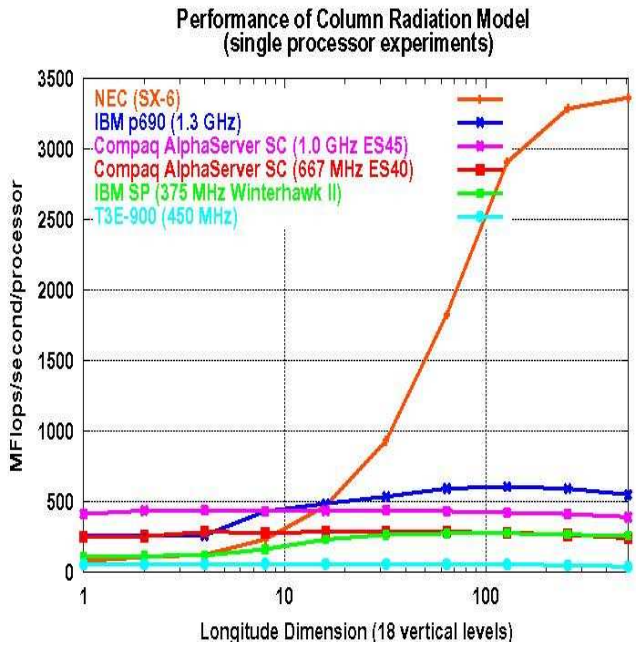

Figure 1: Single processor comparison over platforms for the column radiation model.

The Community Land Model provides the terrestrial component of the coupled system and is being extended to model carbon fluxes with dynamic vegetation processes and realistic land use patterns.

The POP ocean model from LANL is based on free surface formulation using innovative grid structures to get good arctic ocean circulations and sea ice interactions. A new formulation based on a hybrid vertical coordinate system is being developed in the HYPOP code.

A dynamic sea ice model (CICE) based on visco-plastic rheology is also included in the coupled system.

Vectorization and cache friendly data structures that are adjustable by the user provide much of the performance portability of the CCSM. The project has been able to support the user community and ongoing simulations with a single source code that runs well on all the supported computer platforms. The approach has been put to the test in FY2004 to address the reappearance of vector architectures in the Cray X1 and the NEC SX systems. Simulations are planned on each of these systems. Though it took some work, the layered architecture with utility layer for machine dependent parts minimized the effort in porting to new platforms while maintaining performance on the CCSM standard production platforms at NERSC, NCAR and ORNL. Using a library layer with communication supporting MPI, MPI2, SHMEM, Co-Array Fortran interfaces allows some customization for the specific architecture without affecting the model layer and code readability.

2.3 Performance Metrics

There is no single kernel that can be optimized for the atmospheric code. The dynamics calculation is a spectral transform that involves Fast Fourier Transforms and Legendre function transforms to discretize and solve the semi-implicit equations. Thus, optimized library routines are used where available. The atmospheric physics code represents the column radiation balance and moist processes in the vertical. Long wave and short wave absorption are calculated across a number of spectral bands for the chemical composition of the simulated atmosphere. These calculations are combined with representation of sub-grid scale physics that must be parameterized based on observed, measured data. A column physics benchmark has been developed (PCRM) to indicate the performance of the column radiation within the atmospheric model. The production throughput of component and coupled models is measured in simulated years per day as the primary performance metric.

2.4 Hardware Bottlenecks

Each component model has application characteristics that can utilize specific machine architectural features. The relevant algorithmic characteristics are listed with some comments on sensitivity to machine architectural features.

1. POP ocean code - finite difference with halo-updates and conjugate gradient barotropic solver
2. CSIM ice model - finite difference with halo-updates and incremental remapping advection
3. CAM atmosphere - vector radiation/physics calculations, spectral dynamics, semi-Lagrangian advection with halo updates
4. CLM2 - pointer data structure, required a complete re-write to get an explicit loop in each processing routine to get vector performance
5. CPL6 - coupler utilizes a distributed sparse multiply for interpolation and field regridding in the nxm data transfer
6. Fast network bandwidth allows good distributed performance on spectral methods and data transpositions especially important in CAM
7. Low network latency allows fast collectives and halo updates especially important in POP and CICE.

2.5 Case Study: The Community Atmospheric Model

2.5.1 Scientific Goals and Preparation for the IPCC Simulation Project

The IPCC Simulation project attempts to provide policy makers with a variety of possible climate futures based on carbon dioxide emission scenarios that correspond to possible economic and technological futures. The standard case is termed 'business as usual' and consists of a 1in atmospheric CO2 with the consequent warming of the earths land and ocean surfaces. Other scenarios are based on carbon stabalization at specified levels and explore the resulting climates.

In order to prepare the CCSM for use in the IPCC Simulation Project a variety of things needed to be added to the code. These included historical volcanic dust and sulfate emissions, GHG emissions, sulfate chemistry, and cloud water prognostics.

2.5.2 Resolution for Regional Impacts

After roughly 10 years at a resolution of T42 (2.8 degrees) (due to lack of improvement in computer interconnection bandwidth), the IBM p690 the resolution to T85 has doubled. This new model allows better resolved regional climates and some effect of hurricanes and extreme weather. The arctic simulation is one region that improved dramatically with the increased resolution. This model also has a better El Nino Southern Oscillation than previous US models.

Long wave radiation balance (and bias) is more physically realistic than CCSM2.

2.5.3 Hardware Example

IBM p690 located at Oak Ridge National Lab consists of 27 nodes each with a 32 way shared memory processor. We use a hybrid MPI OpenMP programming paradigm that allows us to specifically target the machine strengths for distributed memory and shared memory optimization.

2.5.4 Performance Gain

The main code optimizations that were introduced were the load balancing of the chunks, new communicators in atmosphere and with the land model, and a new spectral domain decomposition that allows more fine grain parallelism as well as a smaller memory footprint.

2.5.5 Discussion

With these optimizations the IPCC coupled simulations were run at 5 years /day on the IBM p690. This is allowing completion of the runs within the allotted time.

The same data structures have also been vectorized for the Cray X1 with an observed throughput of 20 years per simulated day at T85 resolution.

2.6 Case Study: The Parallel Ocean Program

2.6.1 Science Goals and Eddy Resolving Simulations

In order to get the surface wind-driven circulation of the oceans correct, it is necessary to resolve mesoscale ocean eddies with spatial scales of 10-50km.

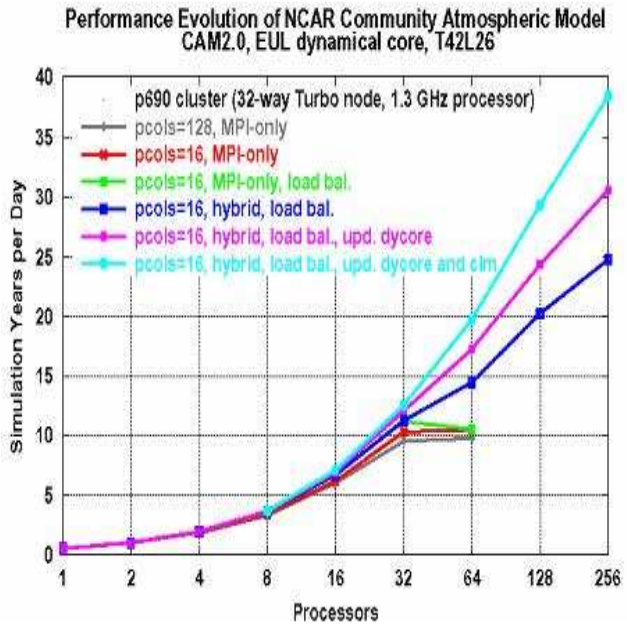

Figure 2: CAM 2.0 in the current fiscal year on the IBM p690 system at ORNL. Distributed and shared memory optimizations are studied here.

Progress in the efficiency of the POP ocean code and in computer capability permitted high resolution simulations of the ocean, resulting in greatly improved representations of ocean circulation. Simulations of the North Atlantic basin at 1/10 degree (4-10km) resulted in very realistic simulations of Gulf Stream separation and subsequent structure around the Grand Banks. The first global simulations at 1/10 degree oceans showed similarly improved representations of the Kuroshio current off Japan, though the Gulf Stream in this simulation was not as realistic as those in the basin-scale simulations, illustrating that eddy-resolving resolution is a necessary, but not sufficient condition for improving ocean simulations.

At climate timescales necessary for IPCC assessments, high resolution simulations like those above remain too computationally expensive. Instead, resolutions of one degree (100km) are used and the effects of mesoscale eddies must be parameterized using computationally intensive schemes like the Gent-McWilliams eddy parameterization.

2.6.2 Software improvements

As mentioned above, flexible blocking data structures have been introduced in version 2.0 of POP. Such blocking enables performance portability by sizing the blocks for cache or vector performance and distributing the blocks more flexibly across a machine for better load balancing and hybrid parallelism. For the ocean, such data structures have the additional advantage of eliminating blocks that are only land.

2.6.3 Performance Gain

For the 1/10 degree model, the new blocking structure on SGI Origin class machines resulted in up to a 30% improvement. Performance at lower resolutions was not substantially improved by the new structure as block sizes were already small and fewer opportunities for land point elimination are available at such coarse resolution.

Performance portability was effectively demonstrated by the performance of POP on the Cray X1 vector machine. Performance of POP on this machine exceeds that of cache-based machines by factors of 6-10. Performance on the X1 at the coarse one-degree resolution even exceeded the Earth Simulator, though the Earth Simulator will probably still be superior at higher resolutions where it can take advantage of the longer vector lengths.

2.6.4 Discussion

The performance of POP on the Cray X1 at one degree resolution typical of IPCC simulations is dramatically better than any other machine available. As the vector length grows for higher resolution simulations, the NEC vector architecture shows significant improvement in efficiency over the 1 degree code. High resolution 1/10 degree simulations will continue to be a focus in the next year with the hope of eventually using such high resolution at climate timescales.

At the end of Q4, the the process of implementing the POP decomposition scheme into the sea ice model started. The outcome of this model change will yield a substantial improvement not only in the ice model, but also a decrease in the amount of information transferred within the coupled system.

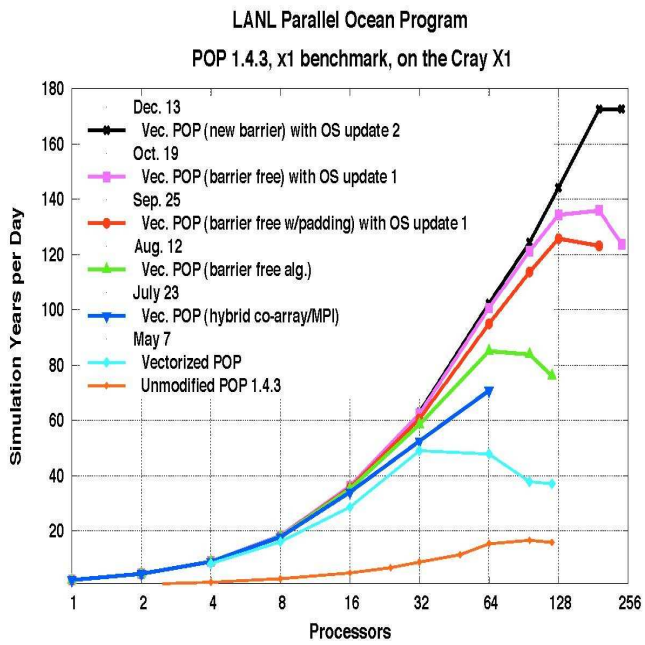

Figure 3: The evolution of performance of the POP code due to software and hardware improvements on the Cray X1 over the last year is shown.

3 Software Development for the Nuclear Shell Model Monte Carlo Code in FY03 - FY04

3.1 Overview

The Shell Model Monte Carlo (SMMC) code was developed to enable calculations for nuclear structure. A number of recent developments impose new and more stringent tests of our ability to describe nuclear structure. Heavy-ion induced reactions allow the study of nuclear behavior at extremes of temperature, angular momentum, or proton-to-neutron ratio. Increasingly precise experiments with electron, pion, kaon, and nucleon beams probe new modes of excitation. As our understanding of supernova and nucleosynthesis is refined there is a corresponding need to know more precisely the relevant nuclear properties. With the advent of the Rare Isotope Accelerator, the SMMC methods will become one of several principle codes that may be used to theoretically investigate the very unstable and interesting nuclei to be

studied at RIA.

The range and diversity of nuclear behavior have naturally engendered a host of models. Short of a complete solution to the many-nucleon problem, the interacting shell model is widely regarded as the most broadly capable description of low-energy nuclear structure, and the one most directly traceable to the fundamental many-body problem. Unfortunately, the combinatorial scaling of the many-body space with the size of single-particle basis or the number of valence nucleons restricts exact diagonalization to either light nuclei or heavier nuclei with only a few valence particles.

The SMMC methods circumvent some of these difficulties while retaining the rigor, flexibility, and predictive power of traditional shell-model calculations. These methods are based on a Monte Carlo evaluation of the path integral obtained by a Hubbard-Stratonovich transformation of the imaginary-time evolution operator. The many-body problem is thus reduced to a set of one-body problems in fluctuating auxiliary fields. The method enforces the Pauli principle exactly and the storage and computation time scale gently with the single-particle basis or the number of particles.

Monte Carlo evaluation of the path integral requires several steps. The evolution operator is an exponent of a matrix representation of the one-body Hamiltonian which describes how particles behave in the fluctuating fields. There are two one-body Hamiltonian matrices (one for protons and one for neutrons) for each time slice of the path integral. If the imaginary time, which is measured in units of inverse energy, is taken to infinity, then the ground state information on the system is recovered; otherwise, the information on the thermal average of the system is obtained. Because of the natural gap between ground and excited states in nuclear systems (particularly in those with an even number of particles), an imaginary time of 2 MeV^{-1} (in nuclear units) allows for ground state calculations.

3.2 Physics Gains in FY04: Example

Shown in the figure is one recent application in the fp-gds model space. No other application that includes shell-model correlations can perform these types of calculations. The three nuclei studied have neutron number N=40 (naively a closed core since the neutrons would occupy the fp-shell model states). In these nuclei, the evolution of shape as a function of particle number and temperature were studied.

It is shown that for an extremely deformed nucleus (such as ^{80}Zr) , the

nucleus maintains its ground state shape up to high temperatures. The two other systems, ^{72}Ge and ^{68}Ni, do not maintain their shape to such high temperatures.

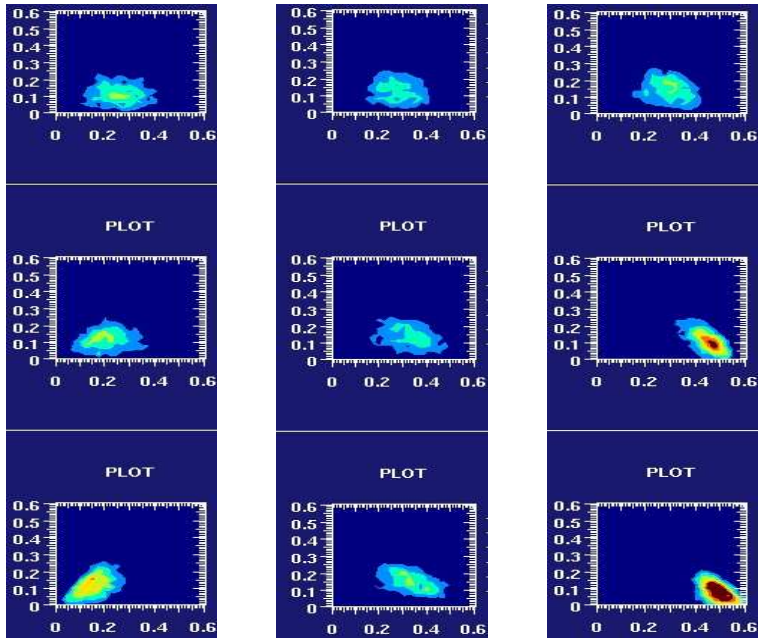

Figure 4: Shown in the figure is the evolution of shape for three nuclei in the fp-gds shell model space. These nuclei are (from left to right) ^{68}Ni (very neutron rich), ^{72}Ge (stable), and ^{80}Zr (neutron deficient) at temperatures (from top to bottom) of $T = 2.0MeV, 1.0MeV$ and $0.5MeV$ (near the ground-state). Each has a fixed neutron number of $N = 40$. The evolution of shape as a function of temperature in the beta (deformation) and gamma (indicating the type of deformation, whether prolate ($\gamma = 0$), oblate ($\gamma = 60$) or triaxial ($\gamma = 30$)) plane is shown. Here, $x = \beta sin(\gamma)$ and $y = \beta cos(\gamma)$ is plotted. ^{80}Zr exhibits extreme deformation that lasts to high temperatures, while Ni is quite spherical and Ge is in between.

3.3 Monte Carlo Computation in the Nuclear Shell Model

The imaginary time (see Overview) is discretized, typically in steps of 1/32 which obtains reasonable convergence. Thus, the following operational steps for each Monte Carlo sample are performed. First, the one-body Hamiltonian matrix (which depends on the fluctuating auxiliary fields) is generated.

This matrix is then exponentiated to obtain the matrix representation of the evolution operator for a given imaginary time step. All such time slices must subsequently be multiplied together to produce the imaginary time evolution matrix, U. Next, the determinant of (1+U) is computed in order to formulate the probability that is used to determine whether to accept or reject the Metropolis random walk [1] for a given auxiliary field. These steps are performed several times per recorded sample to obtain statistically independent samples.

The SMMC code performs separate Metropolis random walks on each processor, making the code rather efficient in communications overhead and embarrassingly parallel in computation. For each sample, the code performs a global reduce operation (this has the effect of synchronizing the code when each sample is taken). The global reduction is over only about 50 variables -observables such as energy, level-occupation numbers, and various transition operators.

3.4 Performance

The SMMC code has matured significantly in the last 10 years. Published benchmarks from 1997 indicate the average MF rating (per processor) was 36 Mflops on the IBM-SP2 Thin66. Today, the code (with continuing software and hardware performance enhancements) performs at roughly 300 Mflops/processor (and up to 350 depending on the application) on Seaborg at NERSC. The code is written in FORTRAN and utilizes MPI for global operations.

Several benchmark calculations have been computed over the course of the last year using the SMMC code. These calculations were performed on very large systems consisting of either 30 (gds) or 50 (fp-gds) single-particle states for both neutrons and protons. All calculations were performed on Seaborg. For comparison, the corresponding matrix diagonalization that would be required in standard shell-model diagonalization procedures would be of rank $1.3E10$ for gds or $2E16$ for fp-gds. Depending on the type of measurements taken, the performance of the code is given in the following table (data taken with POE on Seaborg).

Note: A change from MP_PIPE_SIZE=16 to MP_PIPE_SIZE=32 was necessary to make the final 2048 processor run.

Procs	Job Size/MC samples	Wall Time	Flop rate	Aggregate Rate
16	Ni68, fp-gds / 32	0.5 hrs	315 MFlops	5 GFlops
128	Ni68, fp-gds / 4096	5.8 hrs	271 MFlops	34.7 GFlops
256	Ni56, fp-gds / 5120	3.3 hrs	285 MFlops	72.9 GFlops
256	Ni68, fp-gds / 5120	3.3 hrs	284 MFlops	72.7 GFlops
256	Ni78, fp-gds / 5120	3.3 hrs	283 MFlops	72.4 GFlops
512	Mo92, gds / 16384	0.7 hrs	311 MFlops	159 GFlops
1024	Mo92, gds / 32768	0.7 hrs	317 MFlops	325 GFlops
2048(mp=16)	Mo92, gds / 65536	0.9 hrs	241 MFlops	493 GFlops
2048(mp=32)	Mo92, gds / 65536	0.74 hrs	298 MFlops	610 GFlops

3.5 Discussion

The signature of the Metropolis algorithm is that information needs to be synchronized due to at least two or three (implementation dependent) conditional statements in any given iteration.

The SMMC code utilizes the parallelization whereby ensembles of the path integral formulation are distributed over processors and computed in a completely concurrent mode. Once each processor computes its portion of the integral locally (which may require parallel execution as the physics is refined), the information needs to be accumulated globally so that each processor obtains the updated physical state. Thus, enhancements to the SMMC are largely limited by algorithms that perform global operations over all processors involved in the computation. The code relies upon the MPI implementation that exists on the system for these global operations. The implementations are platform dependent as well as being limited by hardware capabilities. Algorithms for global operations need to be rethought each time the number of processors is scaled up for a system due to system latency and other system scaling effects.

It is suggested that the SMMC code be written to be fault tolerant. This will prepare the code for future hardware systems that will couple tens of thousands of hardware components in the parallel execution of an application.

The SMMC code identifies number of nucleons, number of shells, number of Monte Carlo samples, and number of imaginary time steps as parameters that affect the performance and scientific quality of a production scale run.

4 Software Development for the Virginia Hydrodynamics Code in FY03 - FY04

4.1 Overview

VH-1 is a hydrodynamics code based on the PPM (piecewise parabolic method) algorithm, capable of simulating three-dimensional turbulent stellar flows with high accuracy and little dissipation (in other words, it is able to track features very well over long simulations), and accurately resolving strong shock waves within a couple of numerical zones. This combination of accurate treatment of shocks and turbulent flows is critical to the problem stated below.

The problem is to compute three-dimensional simulations of "spherical accretion shocks" (SAS) around a compact stellar core. The SAS serves as an idealized model of the supernova shock wave trapped inside the core of the progenitor star, with the outer core of the star accreting onto the shock. While the actual mechanism driving the explosion of a massive star may involve a complex interplay of all four forces of nature, the goal in this work is to understand how the shock wave that forms at a radius of about 200 km from the center of the star is able to assist the explosion through multidimensional effects. Virtually all 2D simulations of core-collapse supernovae have shown that this shock becomes highly distorted after a few hundred milliseconds. Our early 2D work with SAS models has shown that this physical situation is unstable, and leads to a growing, asymmetric shock wave: the Spherical Accretion Shock Instability, or SASI. High-resolution, three-dimensional simulations to understand this accretion shock are critical in order to eventually understand the full core-collapse supernova mechanism.

The scientific objectives are:

1. To understand the origin of SASI and the conditions under which it may affect the evolution of the nascent shock wave in core-collapse supernovae.

2. To discover if (and how) the dynamics of a non-spherical shock can assist the explosion of a core-collapse supernova. This amounts to understanding the linear and non-linear evolution of the SASI, respectively.

4.2 VH-1 at the beginning of FY04

4.2.1 State of the Code

VH-1 is a mature code that has been run on DOE, NSF, and NASA supercomputers for the past decade without significant changes. It has been used for this supernova application the past three years primarily on IBM SP Power3 machines at NERSC, ORNL, and the North Carolina Supercomputing Center. The performance of the production code on these machines is consistently around 70,000 zone updates per second per processor (roughly 175 Mflops/processor), with nearly linear scaling up to about 500 processors. The original set of 3D SAS runs used 480^3 grids running for 40,000 timesteps on 480 processors, for a total run time of about 36 hours.

However, to accomplish the scientific objective, a move to increasingly large computational grids (of order 1 billion zones) was required. This could be done on Seaborg, but the efficiency of the code was dropping with such large grids, the I/O was becoming a bottleneck with more than 500 processors, and the total run time became prohibitive. In anticipation of running on the Cray X1 at ORNL, VH-1 was prepared for that platform.

4.2.2 State of the Data Pipeline

By the end of FY03, some 3D accretion shock models on a grid with 100 million zones had been computed, producing almost 1TB of data per run. The data for a given time snapshot was over 2GB which is too large to analyze and visualize on a desktop workstation. The EnSight visualization software was utilized in parallel mode on Seaborg. However, this software is restricted to batch mode and as such the data was not able to be effectively explored. This experience inspired new efforts on data management, analysis, and visualization of terabyte-scale datasets for the project that would be carried out in FY04.

4.2.3 Scientific Output

With the 3D simulations run on the IBM SP Power3 machines it was confirmed that the spherical accretion shock instability (SASI) discovered in 2D axisymmetric simulations does in fact operate in 3D. However, because of the limited grid size the evolution of the shock once it became distorted and began to grow in size (transitioning from the linear to nonlinear stage) was

not tractable. Extensive testing in 2D has shown that one needs a 3D grid with at least a billion zones to correctly model the evolution of the shock out to an average radius that is 3 times larger than the initial radius of the stalled spherical shock. 3D simulations on such larger grids were required to learn if this instability could produce a growing, asymmetric shock capable of blowing out of the core of the progenitor star.

4.3 VH-1 at the End of FY04

4.3.1 State of the Code

With a few small modifications to VH-1, reasonable speeds and scaling on the Cray X1 were obtained. The corresponding "production speed" of 70,000 zone updates per second per processor on the IBM SP3 is 1,140,000 zone updates per second on the X1, or the Cray X1 is about 16 times faster. More importantly, the Cray X1 performs better on larger grids (up to 1,459,000 zone updates per second for a billion-zone grid). Furthermore, by running on fewer processors there are no cumbersome I/O bottlenecks. This code performance makes it possible to run billion-zone models in a reasonably short period of time (e.g., 48 hours wall clock for 50,000 time steps on 200 processors). A single large 3D simulation of the SASI on the Cray X1 has been computed, ending on a final grid of 500 million zones. The early, linear evolution was computed on smaller grids for efficiency. VH-1 can evolve to larger grids, but this particular run reached a maximum shock radius and then retreated, negating the need for grids with more than a billion zones. This has been an important development.

4.3.2 Performance

For the following tables there is a rough conversion of about 2,500 flops for each zone update. Thus, these numbers can be represented in terms of flops/processor, or total flops. For instance, the last case in table two has 250 processors. Thus, 250 pes x 1257,000 zone updates x 2,500 flops/zone update = 785 Gflops.

Scaling for 600^3 zones on the Cray X1 at ORNL:

Processors	Kilozone Updates/s/pe
5	1064
10	1110
30	1142
50	1130
100	1170
200	1104

Scaling for 1000^3 zones on the Cray X1 at ORNL:

Processors	Kilozone Updates/s/pe
20	1473
50	1395
100	1495
200	1295
250	1257

Memory usage is not a limiting factor because a total memory requirement of tens of GBytes is distributed over hundreds of processors.

4.3.3 State of the Data Pipeline

Today, a production run executes on a grid of one billion cells on the Cray X1 and typically generates (4-6)TB of data. While a single run takes less than 40 hours of wall-clock time on the Cray X1 at ORNL, it takes many weeks to transport this much data from ORNL to the data analysis cluster at NC State University. The data pipeline is still in development, but it is functional today.

The current approach includes an integration of networking tools (LoRS), I/O libraries (HDF5, and later netCDF), parallel visualization software (EnSight), and a local Linux cluster. The simulation data generated on the Cray X1 is moved from ORNL through the HPSS to a project cluster (all behind a ORNL firewall) to NC State University, parsed and distributed to the nodes of the data cluster, and remains available for interactive visualization as long as desired. This pipeline is not yet automated and requires constant monitoring, but it satisfies the critical need of interactivity with TB datasets. The speed of data movement and processing must be improved at all stages, beginning with movement of data off of the Cray X1 to some platform that can provide continued interactive parallel access to the data.

At the end of Q4 a new implementation was tested whereby the data is pushed to NC State from ORNL directly from the Cray X1. This resulted in promising results to be fully implemented in the code. The gains are reported in the closing section.

Please read the discussion of this section for more details on data movement.

4.3.4 Scientific Output

The current state of 3D supernova simulations within TSI is focussed on the dynamics of the stalled accretion shock, and how the instability of this shock might help generate an asymmetric explosion capable of explaining the observed asymmetry of many supernovae. Production runs in Q3 and Q4 have provided some tantalizing science results.

The data from more production runs is needed before the 3D evolution of spherical accretion shocks can be understood. Nonetheless, early runs have shown the same rapid linear growth of the SASI as seen in previous 2D and 3D runs, but once the shock had grown to about twice its original size, the dominant sloshing mode (l=1 in spherical harmonics) was disrupted and the shock collapsed back to its original radius. While these results are still not understood, it is clear that the nonlinear evolution of the supernova shock wave in three dimensions can be different than seen in two-dimensional simulations.

Next, because of the data bottleneck described in the previous section, some production runs were made in late Q4 without attempting to store the full dataset. In these runs, only a few gross features of the flow are recorded -the remaining data is discarded. One of those global features, the angular momentum imparted to the proto-neutron star at the heart of the explosion, has provided an exciting new result. The unstable accretion shock can evolve in such a way that it generates a global vortex capable of spinning up the interior flow to extreme rotation rates.

This is a remarkable result. The implosion of a stationary star can leave behind a neutron star spinning with a period of roughly 50 ms! This value is comparable to the spin rates observed for young radio pulsars [12] associated

[12] Radio pulsars were discovered in 1967, and quickly associated with rapidly rotating, magnetized neutron stars. These stars are more massive than our Sun, yet are small enough to fit between Raleigh and Chapel Hill. They are believed to be born in core-collapse supernovae, as is clearly the case with the pulsar in the Crab supernova remnant.

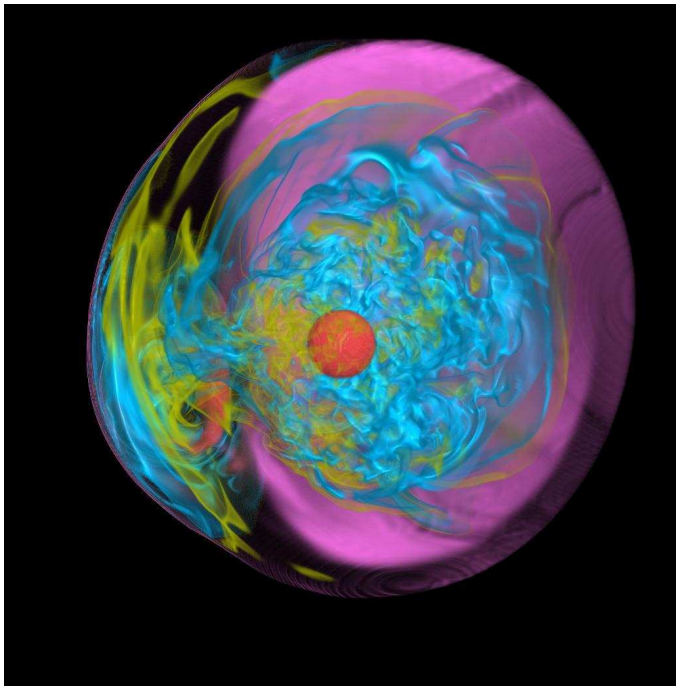

Figure 5: Snapshot of the turbulent stellar core flow beneath the supernova shock wave resulting from stationary accretion shock instability (SASI).

with supernova remnants. Note that this effect can ONLY be seen in 3D simulations.

We now have an exciting new discovery, but to understand the physics of this phenomena we must be able to examine the full data, and in particular examine the dynamical evolution of the accretion shock generating the vortical flow. As such, our scientific discovery process is now clearly limited by the rate at which we can move terabyte-scale datasets from the high-performance computing platform (uniquely capable of running the simulations) to a local data analysis cluster (providing unique dedicated resources for analysis) where interactive inquiry of the data can lead to meaningful scientific discovery.

The initial periods are believed to be in the range of 30-60 msec. Why are they born spinning so fast?

4.4 Discussion

VH-1 has evolved over the current fiscal year by being prepared for a new architecture, being ported to the new architecture, and producing a data analysis process that is important not to be ignored. Further optimization on the Cray X1 is still ongoing.

The concept of sharing large data sets over geographically distributed computing centers is not a new problem. In a nutshell, once the data hits the network, it is reduced to the network flow problem. However, the implementation here is unique for a number of reasons. First, the data is generated behind a restrictive firewall on a DOE production system that does not allow for a quick dissemination of the data to the wide area network. Next, once the data reaches the other side of the firewall, it is moved over the network by hopping through a system of distributed disks and broken into files that are stored at the final destination potentially over a large assembly of disks. This approach has not yet been fully studied for efficiency.

Now, although parallel streams usually only yield a large benefit with wide area transfers, attributes of the Cray X1 make it useful for both LAN and WAN transfers. A problem stems from the Cray X1's relatively poor scalar performance. A single Cray X1 SSP drives a TCP stream significantly slower than can a typical Intel processor. This is exacerbated by the dynamics of TCP in a wide area environment, resulting in especially poor WAN performance. The Cray X1 does, however, have many processors and a significant amount of I/O capability to throw at the problem. Sending the file over parallel TCP streams alleviates the scalar problem with parallelism, achieving more reasonable LAN and WAN transfers.

It is likely that great improvements will be gained by applying parallel algorithms that allow multiple processors to access fragments of a large data set that have been strategically placed over multiple data storage disks. Regardless, unless the network hardware is dedicated -it is not in this case- there is an indeterminacy involved in the movement of data in the wide area.

5 Software Development for the R-Matrix with Pseudostates Codes in FY03 - FY04

5.1 Overview

The R-matrix with Pseudostates (RMPS) codes are a serial/parallel suite of codes used to study the electron excitation/ionization of light fusion related species through to heavier complex targets such as Fe. High quality electron impact excitation/ionization atomic data is still the foundation of collisional-radiative modeling used to support the interpretation of fusion experiments. Fundamentally, a new parallel suite of codes has allowed for an improved description of the atomic target, and for a study of the effects of including highly-excited Rydberg states and the target continuum in energy regimes above the ionization threshold.

The physical goals are:

1. To extend these calculations to more complex species that require semi-relativistic effects included within the description of the target and incident electron.

2. In the long term, to produce a single comprehensive electron excitation/ionization package that allows for non-relativistic, semi-relativistic and fully relativistic calculations.

3. To ensure that complete sets of resonant-resolved excitation cross sections and groundstate/metastable ionization cross sections are formatted and integrated within collisional-radiative modeling packages.

5.2 Code Structure of the RMPS codes

All the codes in the RMPS suite are written in FORTRAN. For the parallel codes, MPI is utilized as well as the libraries required to utilize ScaLAPACK. These are the PBLAS (Parallel Basic Linear Algebra Subprograms) and the BLACS (Basic Linear Algebra Communications Subprograms) libraries. The PBLAS on Seaborg relies on BLACS and the IBM ESSL (Engineering and Scientific Subroutine Library) libraries (on other systems the serial BLAS and the LAPACK libraries are required). The BLACS is built directly upon the MPI implementation.

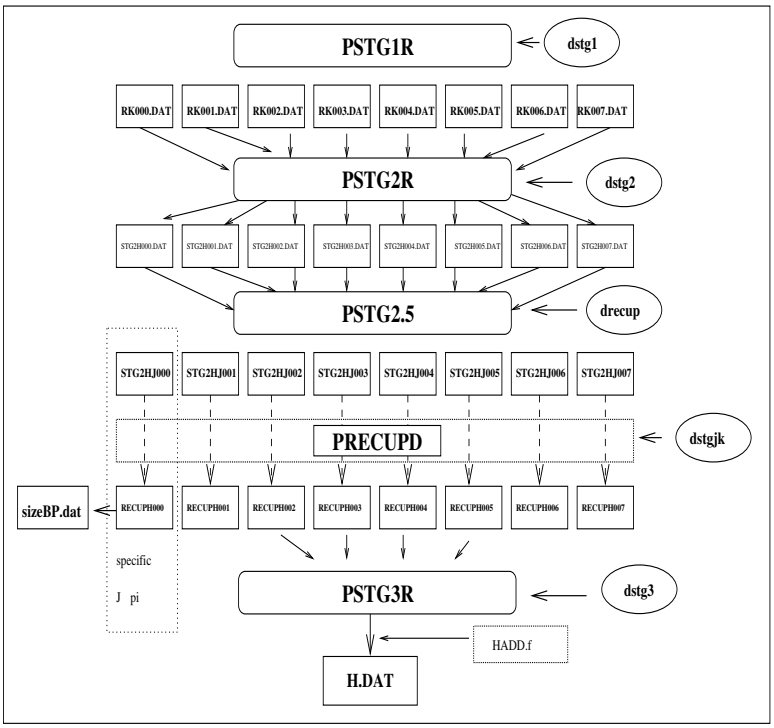

Figure 6: Code flowchart : including FY03/04 developments up to Q3 FY04

The code flowchart allows for the tracking of software development down to the subroutine level in the RMPS codes. In what follows, relevant stages are described coupling their purpose with software implementation and performance where it is particularly relevant.

- The RMPS scattering package requires the radial orbitals derived from an atomic structure package, such as AUTOSTRUCTURE. There is little benefit in parallelizing this fast serial code.

- *stg1r.f/pstgr.f* -generates a continuum basis to represent an incident electron over a user defined energy range. Subsequently, every one and two electron integral (of which there are billions) used to create an $N+1$ electron Hamiltonian is generated. In the parallel version of the code, blocks of integrals are distributed over 16-32 processors which independently write separate files. The parallel code is very efficient and runtime is less than 1 hr for a typical calculation.

- *stg2r/pstg2r.f* - generates all the angular algebra and constructs the elements of the $N+1$-electron Hamiltonian.
  The parallel code is moderately efficient as the angular algebra is distributed in symmetry blocks of 16-64, which are completely independent of each other. The low end of that scale allows for use on small Beowulf clusters of 10-32 processors.
  These codes exhibit the greatest degree of time variation across architectures, as not only do they have to interrogate multiple (DIRECT access) files of perhaps 1 Gb from *pstg1r.f*, but also write passing binary files of even larger magnitude. A SGI Altix with fast communication between nodes performs better than a 26 processor Opteron Linux cluster, in which the master node struggles manage the I/O demands even with a 1 Ghz switch. Further code profiling and optimization is still required.

- *stg3r/pstg3r.f* - computes the matrix diagonalization of a real symmetric matrix. The most computationally demanding problem involves the matrix diagonalization of a dense real symmetric matrix for which all the eigenvalues and eigenvectors are required. The shape of the undiagonalised matrix prohibits the use of banded matrix techniques.

  The ScaLAPACK routine *pdsyevd* has been utilized for this operation. A matrix of rank 50K typically requires 30 minutes on 1024 processors on Seaborg, however a significant time (10-15 mins) is required to read in and distribute the matrix elements across processors.

- *stgf.f/pstgf* - produces the final observable, whether it is an electron excitation cross section or an ionization cross section.
  Computationally this stage is embarrassingly parallel. Each energy point of the incident electron is distributed across processors in a cyclic manner, with an emphasis on load balancing. The distribution can become a challenge when narrow resonance structure has to be resolved.

5.3 Computation and Performance for the RMPS codes for FY03 - FY04

The first fully parallel versions of the code stabilized in the FY03, and hydrogenic targets through to beryllium were investigated. In providing performance metrics for atomic scattering there is variation in performance results

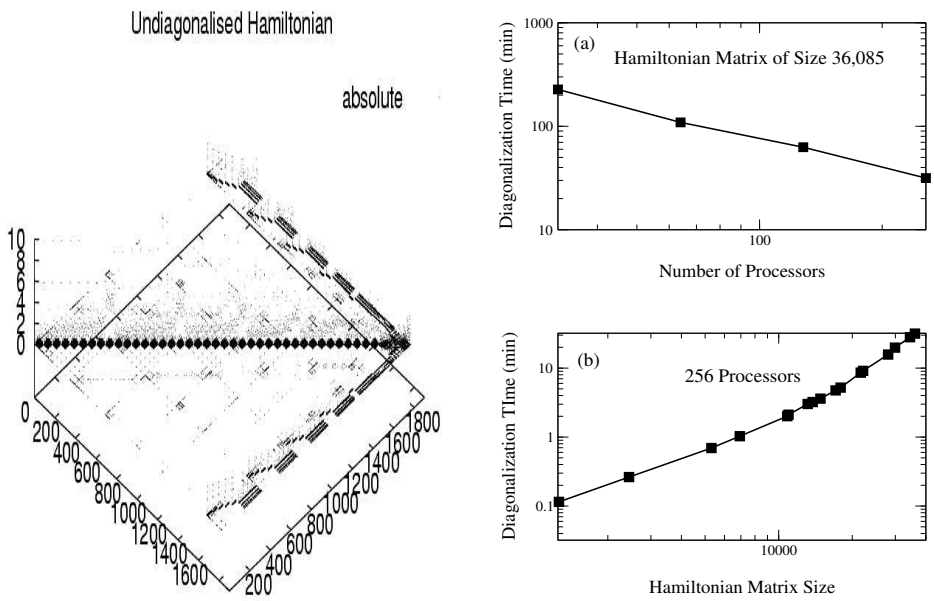

Figure 7: The plot on the left depicts the typical relative values and distribution of the matrix elements involved in *stg3r*. On the right, the diagonalisation of a real symmetric matrix as a function of time and processor. This corresponds to a 238 term C^{2+} calculation. Timings were carried out on Seaborg. The top graph on the right has a fixed Hamiltonian matrix size, varying the number of processors and the bottom varies the matrix size with a static number of processors.

dependent upon the complexity of the target. A typical calculation in FY03 was a 238 term C^{2+} electron-impact excitation (see figure).

In FY04, the focus has been on developing new additions to an emerging suite of parallel codes to address required physics before refining and improving the computational methods in the existing codes. The R-matrix method requires the repeated diagonalisation of matrices. It stands that serial calculations have already reached a computational limit (see J Phys B: At Mol Opt Phys 36 p455 third paragraph). These limits can reduce the accuracy of the target and the energy range of the incident electron. As such, a parallel relativistic R-matrix BPRM (Breit-Pauli R-matrix) code was required. Major coding developments (*pstg2.5* and *precupd* -see flowchart) to enable the *simultaneous* formation of every Hamiltonian matrix and the

the implementation of the parallel diagonalization for these heavier atomic systems had to be undertaken. Thus, the greatest benefit has been to extend the codes to heavier atomic systems with the inclusion of semi-relativistic effects. The codes are being tested.

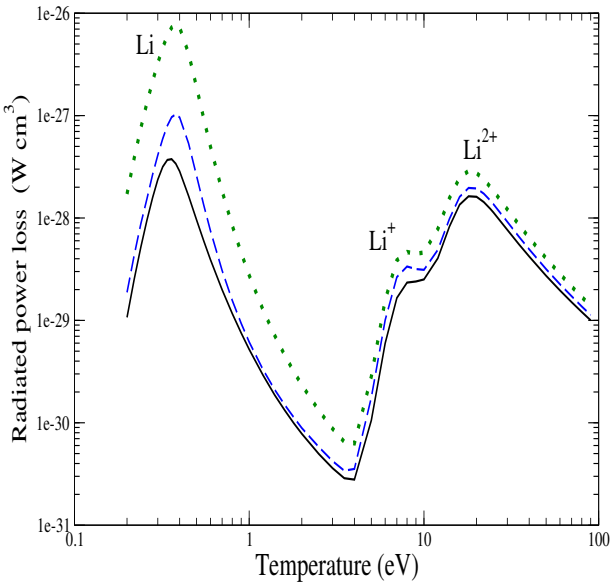

Figure 8: Total radiative power loss for lithium in which the (green dots) Born approximation (Cowan code), the distorted wave (dashed, LANL code), and RMPS (Li^{2+} - 57 term (2004); Li^{+} = 101 term (2003); Li = $\sim$ 60 terms(2004)) results were used in the excitation rates for every ion stage of lithium. The question to address was whether the underlying electron impact excitation method affected the radiative power loss of lithium. The RMPS are 2 day calculations that are more computationally demanding but give a more accurate result. The simpler perturbative methods have limitations. There is up to a 44% difference even in this fairly simplistic system between DW and RMPS.

The contributions and priorities :

1. The introduction of semi-relativistic transformations to the existing parallel code that allows for excitation/ionization of heavier complex targets such as neon. These results would support the neon-gas puff experiments conducted to study the disruption migration in the EDFA-JET tokamak at Culham and improved energy and particle confinement in the D-III-D tokamak at General Atomics in San Diego. Theoretically, it has only recently been computationally feasible to attempt semi-relativistic RMPS calculations with coupling to Rydberg and target continuum states included. The largest calculation to date involved a

matrix of rank 48712. Thus, in FY04 improvements in the RMPS code suite have scaled previous BPRM calculations by a factor of almost 5.

2. The simplification of the parallel codes to the level of the serial versions, with the number of processors being the only additional variable required by the user.

3. The upgrade of a complimentary suite of R-matrix codes (referred to as non-exchange codes) to treat the high angular-momentum portion of the calculation. This allows for a more efficient determination of these contributions, which dominate the cross sections at high energies.

Comments regarding the developments through Q4:

- The serial version of the Dirac Atomic R-matrix Codes (DARC)[7][8] have been developed since the early 1980's. However, the completion of our parallel Breit-Pauli suite of codes has allowed the simple integration of this package within our own that takes the full advantage of the ScaLAPACK matrix diagonaliser. It extends the capabilities of our existing package by allowing us to study electron scattering by complex heavy atomic targets with nuclear charge greater than Z=35.

- The first stage was to convert the Hamiltonian matrix elements given in jj coupling by DARC to JK coupling format used by the parallel Breit-Pauli suite in the formation of Hamiltonian matrix. This is achieved by the conversion code dto3/pdto3 written by Prof. Keith Berrington. Investigating cross section differences for heavy nuclear targets by the two independent codes is a future project, but initially convergence in results at moderately low Z for a highly ionized system such as Fe^{14+} was carried out. As figure 9 shows, the differences in the cross sections between the two codes is minimal, provided that good atomic structure in used in both approaches.

- Secondly, consistent with the parallel Breit-Pauli package which requires the formation of many symmetric Hamiltonian matrices, which then are subsequently diagonalized, the DARC codes employ the same procedure. The elements of every Hamiltonian matrix are now all calculated *concurrently*, and with the parallelization of the conversion code DTO3, are also *concurrently* transformed to JK coupling. These two codes are called *pstg2d* and *pdto3*. Fig 6 illustrates how they integrate into the previous flowchart.

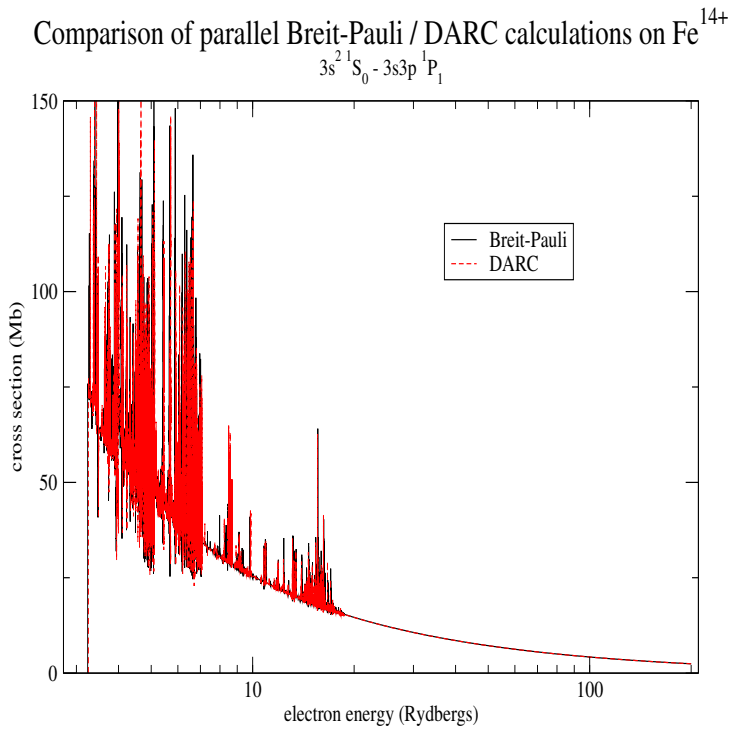

Figure 9: Cross sections computed for the highly ionized system Fe^{14+} are compared for the DARC (jj coupling) and parallel Breit-Pauli (JK coupling) formalisms.

- The existing serial version of the DARC codes were profiled extensively before parallelization and changes made to increase efficiency. Both serial and parallel versions of the code are written in generic fortran 90/MPI and should be platform independent. All testing has been carried out on a 26 processor Opteron linux cluster.

- From a computational perspective, there appears to be undesirable blocking when increasing number of processors try to access DIRECT access FORTRAN file, which affects the overall scaling of the parallelization. Of course, these effects are less acute on large supercomputers which have better communication.

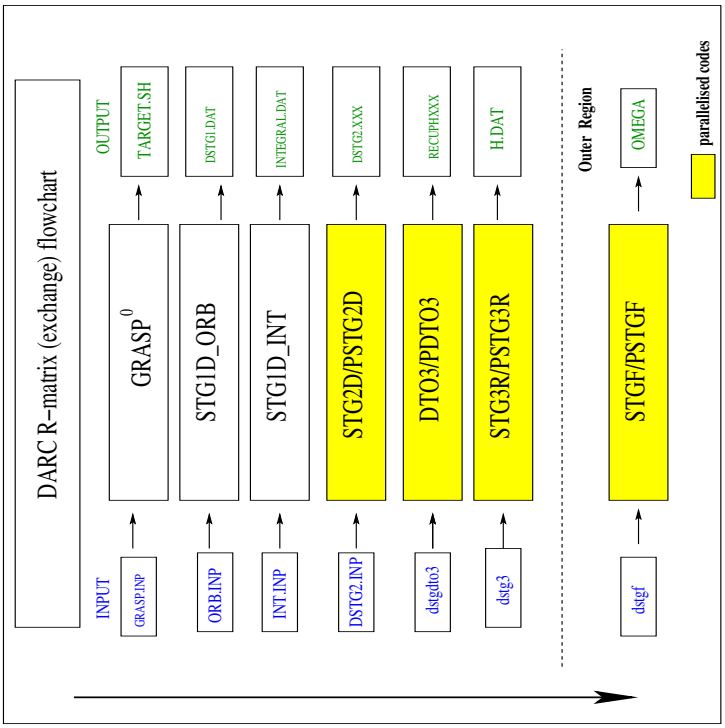

Figure 10: Flowchart of parallel Dirac-Fock scattering codes -changes through Q4 FY04

5.4 Discussion

In FY04, the RMPS code team has improved the physics base of their parallel suite of codes by the introduction of semi-relativistic transformations of the target and incident electron. These changes merge into the existing LS coupled codes, already being used for light fusion species.

One of the aims has been to quantify the impact of the sensitivity of the total radiative power loss of light fusion elements to the underlying electron-impact excitation method used. Traditional simpler methods (plane-wave born/disorted wave) exhibit significant differences to the RMPS results for even simple atomic targets such as lithium.

From a physics perspective, the next challenge is to include the relativistic effects of the incident electron, which are not accounted for presently. However, the codes should be valid for a incident electron energy of a few eV

to several hundred Rydbergs for the majority of cases.

The existing code base is well documented and is organized in a manner that will allow outside help when optimizing routines or porting the codes to new platforms. There is a strict dependence on existing dense linear algebra libraries. The routines have not been optimized for block size, number of processors, or logical rectangular grid aspect ratio (the number of processor rows to processor columns) for these libraries. This step will yield potentially large performance gains. This stage of the RMPS codes will scale in terms of floating point operations on systems where level three BLAS scale. However, the routines will continue to experience performance degradation as larger problem sizes are considered since there is a heavy IO stage whereby a number of files (the number gets larger as the problem complexity increases) is read as input and written as intermediate output.

6 Software Development for Lattice Gauge Theory in FY03 - FY04

6.1 Overview

6.1.1 SciDAC Software Project

The three-year DOE SciDAC program for the Infrastructure for Lattice Gauge Theory is bringing unprecedented coherence to the software development effort of the US lattice gauge theory community. It is beginning to have a major impact on the ability to use present and emerging hardware resources effectively, including switched clusters, mesh-networked clusters, and the QCDOC. The impact on the performance of the MILC code, an extensive publicly available code for lattice gauge theory calculations, is discussed.

The SciDAC software project is creating a library of basic utilities to facilitate the writing of highly portable application code for addressing specific physics questions. The QCD API has a three level structure:

Level 1:	QMP: Message Passing & QLA: Per site Linear Algebra
Level 2:	QDP: Data Parallel lattice wide operators & I/0
Level 3:	Optimized CG Inverters written in assembly languages

These layers include QMP, a message passing interface which hides all network specific details and a single-processor linear-algebra interface. At Level

2 the basic data parallel protocol for the MPP linear-algebra API is provided with datatypes tailored to the needs of lattice gauge theory. Also there is an I/O interface with standardized file formats. For the most important rate-limiting high-level subroutines, such as the conjugate gradient inverters, the level 3 SciDAC code will include custom assembly-language code optimized for each specific architectures. These are callable from the QDP API through a uniform interface. Replacing the level 3 library by C or C++ routines and the QMP library by its MPI implementation, all applications written to the SciDAC standard are entirely portable.

An applications programmer chooses between a C (QDP/C) or a C++ (QDP++) implementation at level 2. In either language the API provides a standard interface for the development of codes specific to the physics under investigation. The standardized file formats with built in XML tools designed in collaboration with the International Lattice Data Grid (ILDG) are designed to facilitate sharing of expensive file archives across a wide community and reduce duplication of effort.

6.1.2 Lattice Gauge Theory Software Development Path

The SciDAC software effort brings together three previously independent application code archives: the C/MPI language MILC collaboration code, historically the most extensive and most portable publicly available code for lattice gauge theory, the C/Macro language SZIN code, in use by several groups, and the C++/Assembly language Columbia Physics System code primarily for the use of the Columbia/Brookhaven collaboration. The MILC code is beginning to be ported to the QDP/C library as manpower permits. So far, a couple of the time-critical components have been rewritten to use QDP/C. In the future a continued transition to QDP/C is foreseen. The SZIN code is being replaced by C++ Chroma, which is built on the SciDAC QDP++ interface. A wide community of QCDOC and cluster users are beginning to use this Chroma code in their research. The Columbia Physics System will also share some SciDAC software components to promote portability.

Porting of the MILC code to the QCDOC was accomplished initially by simply replacing its message passing layer with the SciDAC (QMP) message passing layer. However, considerable further attention to optimization was required. This work inspired improvements and performance gains in MILC code used on clusters.

The examples below document improvements in MILC code performance on MPP architectures during the current fiscal year. All of these improvements were made possible by SciDAC software development and the optimization needed to ready the MILC code for the QCDOC.

6.1.3 Performance Metrics

By far the most time consuming operation in nearly every lattice gauge theory calculation is the solution of a very large sparse linear system. This comes from the discrete approximations to the elliptic partial differential operator for the Dirac equation that governs the motion of quarks. While the physics applications vary, nearly all of them use one of a small handful of conjugate gradient or Krylov space solvers. So the performance of the solver is our standard benchmark. With our current preferred algorithm of the MILC research team, the computation of the "fermion force" term is the second most time-consuming operation. To get an honest measure of performance, all routines are timed using the system wall clock and the performance in Megaflops per processor (MFlops per processor) is reported, a number that can be compared directly with peak single-processor performance.

6.1.4 Hardware Bottlenecks

Data movement has become the key consideration in selecting hardware and in setting the operating parameters for lattice QCD.

The most frequently executed kernel in lattice quantum chromodynamics (QCD) multiplies a three-dimensional complex matrix times a three dimensional complex vector with accumulation. In single-precision that operation must move 4 bytes per 3 flops from memory to cache. Since in recent years, processor speed has grown far faster than memory bus speed, the codes are typically memory bandwidth limited on current processors. Adding a second processor to an MPP node usually results in only a small gain (in the 20% range) in performance, since both processors typically share the same memory bus.

The four-dimensional space-time lattice is distributed among processors through regular domain decomposition. The sparse matrix solver requires exchanging values on the faces of the local volumes. Computation on interior sites can be overlapped with communication. To optimize performance, the size of the local volume can be adjusted so that communication time

matches local computation time. The QCDOC will deliver high throughput by achieving very low message passing latency among many cheap processors, thus allowing small local volumes. On clusters, one achieves similar throughput with fewer, but more powerful processors, somewhat higher latency, and larger local volumes.

Lattice gauge theory code scales very well on switched commodity clusters, at least as far as they have been tested to a few hundred nodes. At 128 nodes the per-node performance is down by about 25% over single-node performance. The departure from linear scaling in the processor number is due to the need to compute global sums in the CG iterations and thus this effect can be reliably parametrized to confidently extrapolate performance numbers to larger networks with known latency characteristics.

6.2 Case Study: QCD Thermodynamics

6.2.1 Physics Goals and Impact on RHIC

At extremely high temperature or high pressure protons and neutrons dissolve into a quark-gluon plasma. This remarkable state of matter prevailed in the early universe moments after the Big Bang. It may also occur today in the cores of highly compact stars. An intensive experimental effort is underway at the Brookhaven Relativistic Heavy Ion Collider to create a "mini-Bang" through collisions of large nuclei. Sorting through the debris of the collisions, searching for evidence of quark-plasma creation is a difficult task that requires a solid theoretical understanding of the signals for plasma formation.

Only through *ab initio* lattice QCD calculations can a few key, quantitatively reliable predictions about the properties of the plasma be made: the temperature and order of the phase transition, the equation of state, and the likelihood of strangeness fluctuations. Numerical lattice gauge simulations are limited to studying these properties in static thermal equilibrium, whereas experimental conditions are necessarily dynamic. Phenomenological models bridge the gap between lattice QCD and the experimental conditions. The solid predictions of QCD are fundamental to these models.

6.2.2 Setting the Lattice Spacing

The traditional formulation of lattice QCD lends itself naturally to a study of matter in thermodynamic equilibrium. The lattice has three spatial dimensions and one "Euclidean" time dimension. The temperature is controlled by two parameters, namely the coupling strength of the gluons and the extent of the lattice in Euclidean time. To obtain a solid understanding of the plasma one wants to be sure to have good control of any discretization artifacts. With today's improved algorithms, we should be simulating at a lattice spacing close to 0.1 fm. That, in turn tells how to set the lattice size and gluon coupling. With a time axis of four sites, the choice of our European competitors, we get a rather coarse lattice spacing of about 0.3 fm at the phase transition. If the time axis has eight sites, the choice in the example below, the lattice spacing is half as big, namely, 0.15 fm. Of course, simulating with larger lattices requires more computer resources. Since simulations over the range of experimentally accessible temperatures and over a range of quark masses are needed, the problem requires a supercomputer.

6.2.3 Hardware Example

The MILC code performance on the NCSA Xeon cluster, installed during this fiscal year, is examined. The table below lists some hardware characteristics and the number of processors and problem size for this benchmark.

Location	NCSA
Machine name	tungsten
Processor	Xeon
Processor speed	3 GHz
Memory FSB	533 MHz
Theoretical peak	4.3 GB/s
Communications	Myrinet
Processors	32
Nodes	16
Problem size	$16^3 \times 8$
Precision	single

6.2.4 Performance Gain

During the past fiscal year the MILC single-mass CG inverter (old code) was replaced with an inverter written with the QDP/C library (new code). The numbers below show the performance improvement of this subroutine in MFlops/processor and percentage gain.

old	new	gain
408	471	15%

During the past fiscal year the previous MILC fermion force subroutine was replaced with a new version that incorporates optimization strategies under development for the QCDOC.

old	new	gain
360	538	50%

6.2.5 Discussion

Over past years the MILC collaboration has worked hard to optimize the C-code Asqtad single-mass CG inverter, so getting an improvement in the 10-15% range for our chosen production parameters is very good. The fermion force term is relatively newer, so stood to gain far more from further attention.

On a per-processor basis, 8% of theoretical peak processor performance is obtained, assuming uninterrupted SSE operation. However, taking the formula for memory access, it is seen that between the two dual processors one gets 30% of the theoretical peak memory bandwidth. For this subvolume of size $8^3 \times 4$ used in this computation, interprocess communication reduces single-node performance by about a factor of two. On a single node one would get about 60% of the theoretical peak memory bandwidth, which is close to a practical sustained maximum.

6.3 Case Study: B Decay

6.3.1 Physics goals and coordination with experiment

An essential part of the search for physics beyond the Standard Model is the careful measurement of its several dozen input parameters, including quark, lepton, and boson masses and their couplings. A precise knowledge of these

parameters provides clues to deeper physics and constrains proposals for more fundamental theories. Among the least well known of these parameters are the CKM matrix elements that determine weak interaction transitions and annihilations among the different quark flavors. Particularly poorly known are the matrix elements for the heaviest quarks, such as the bottom or b quark. To determine these matrix elements, one measures the lifetime and propagation of the B meson, composed of a b quark and a light antiquark.

A major experimental effort is under way at SLAC, Cornell, and in Japan at KEK to study the properties of the B meson, particularly its decay rate. Much like a hydrogen atom with the electron replaced by a light antiquark and the proton replaced by the heavy b quark, it decays when the light quark encounters the b quark and the weak interaction takes place. The annihilation products are leptons. Alternatively, it decays when the b quark decays on its own to a lighter quark plus leptons with its satellite light antiquark behaving as a spectator. Either way, effects of the weak interaction are modified by the much stronger interactions between the heavy quark and the light antiquark. So to put it succinctly, the CKM matrix elements are obtained by dividing the measured decay rates by strong interaction factors. Lattice gauge theory provides the most reliable strong interaction numbers. Consequently the precision of the determination of a CKM matrix element is controlled by the product of the precision of the experimental measurement and the precision of the numerical simulation.

In several important cases the precision of experimental measurement has outpaced the precision of lattice calculations. For example, the decay $B \to D\ell\nu$ determines directly the V_{bc} matrix element. The experimental error is approaching 2%. With state-of-the-art lattice calculations we hope to achieve 5%. With major computer resources over the next few years we may be able to match the experimental error.

6.3.2 Parameter Choices

This project is one of several making physics measurements on an archive of lattice gauge configuration files. Currently the largest of these lattices is $40^3 \times 96$, which have only a small ensemble. The benchmarks were obtained with the more numerous $28^3 \times 96$ lattice size. Double precision is required for technical algorithmic reasons and costs a factor of about two in performance. Since several masses are needed, there is a substantial gain over several repetitions of a single-precision single-mass inversion.

6.3.3 Hardware Example

The MILC code performance on the Fermilab SciDAC Xeon cluster, in operation during this fiscal year, is examined. The table below lists some hardware characteristics and the number of processors and problem size for this benchmark.

Location	Fermilab
Machine name	lqcd
Processor	Xeon
Processor speed	2.4 GHz
Memory FSB	400 MHz
Theoretical peak	3.2 GB/s
Communications	Myrinet
Processors	32
Nodes	16
Problem size	$28^3 \times 96$
Precision	double

6.3.4 Performance Gain

In this case the double-precision multi-mass variant of the CG inverter is examined. During the past fiscal year the MILC multi-mass CG inverter (old code) was replaced with an inverter written with the QDP/C library (new code). The numbers below show the performance improvement of this subroutine in MFlops/processor and percentage gain.

old	new	gain
117	231	97%

6.4 Discussion

The MILC multi-mass inverter has been used only for a few small projects in the past, so has not been subjected to intensive optimization. Double precision is approximately twice as expensive as single precision and benefits even more from efficient use of cache. It is believed that much of the performance gain here comes from more efficient cache loading.

On a per-processor basis one gets 10% of double-precision peak. However, taking the formula for memory access, it is seen that between the two dual processors one gets 38% of the peak memory bandwidth.

6.5 Case Study: Lattice Generation

The SciDAC collaboration, representing essentially the entire US Lattice Gauge Theory community, has been working with counterparts in the United Kingdom, Europe and Japan to develop a common system for sharing lattice files internationally. In Q4 FY04 the ILDG committee has agreed upon and published a metadata standard for describing the parameters and conditions under which the files are generated. That is a crucial first step in creating a searchable database for the archive. Work is continuing on international standards for the middleware system for retrieving files from the archive. The ILDG archive system was inspired by the current DOE "Gauge Connection" archive of lattice files housed at NERSC. The middleware project makes use of tools developed under the companion SciDAC project, Particle Physics Data Grid. The ability to share files will accelerate the progress of science and benefit researchers on all sides of the exchange.

6.5.1 Why Generate Lattices?

The previous example illustrates the importance of generating an archive of lattice gauge configuration files or "lattices" for short. They are the starting point for a wide variety of physics projects, including measurements of the masses and other properties of hadrons, examining the structure of the vacuum state of QCD, and measuring weak interaction decays as in the previous example.

Once generated, these files can be shared throughout the lattice gauge theory community. As a part of the International Lattice Data Grid (ILDG), the SciDAC project is adopting the MILC practice of making lattices openly available once they are generated. Thus a significant portion of the initial SciDAC hardware resources, particularly at ORNL, have been used to make important additions to the archives.

While most of the software development over the past fiscal year has been devoted to clusters and preparation for the QCDOC, the codes are portable and optimization strategies are transferable as the following example illustrates. The QDP/C code suite was ported to the ORNL IBM SP named "cheetah" and gave modest improvement in speed.

6.5.2 Hardware Example

The MILC code performance on an ORNL IBM SP, in operation during this fiscal year, is examined. The table below lists some hardware characteristics and the number of processors and problem size for this benchmark.

Location	ORNL
Machine name	cheetah
Processor	Power 4
Processor speed	1.3 GHz
Memory bus	433 MHz
Theoretical peak	6.9 GB/s ??
Communications	Federated switch
Processors	64
Nodes	2
Problem size	$20^3 \times 64$
Precision	single

6.5.3 Performance Gain

During the past fiscal year the MILC single-mass CG inverter (old code) was replaced with an inverter written with the QDP/C library (new code). The numbers below show the performance improvement of this subroutine in MFlops/processor and percentage gain.

The MILC code is being modified to support the SciDAC libraries.

old	new	gain
450	495	10%

6.5.4 Discussion

The MILC collaboration has worked hard for many years to optimize the C-code Asqtad single-mass CG inverter, so getting an improvement in the 10% range for our chosen production parameters is very good.

These experimental changes began in FY04 and are being incorporated in the next release version of the MILC code. These modifications allow all MILC code applications to run on the QCDOC including those in the case study.

6.6 Prospects for Future Hardware

6.6.1 QCDOC

The QCDOC is currently under construction and is scheduled to become generally available to the US lattice community in the winter of 2005. Part of the acceptance tests for the QCDOC required benchmarking the MILC single-mass CG inverter on 128-node prototypes. These tests compared the "out-of-the-box" single-precision MILC C code performance with the single-precision QDP/C version and the highly optimized double-precision assembly-coded inverter. To be conservative, the results below are quoted for the maximum 450 MHz rating, and for a local volume of 6^4 lattice sites per processor.

old	new	gain
90	171	90%

The old code is written in standard C for single precession and the new code in QDP/C for single precision. These numbers should be compared with hand coded assembly language code for double precision that yields 37% of peak or 333 flops/processor.

Because of the tight design of the communications fabric, extremely good scaling is expected. With the assembly-coded version, the planned 12,288 node machine is expected to sustain an aggregate power of 4 TeraFlops at double precision for this application. The other major version of the CG inverter uses Domain Wall fermions which is expected to give better performance with the prices performance meeting the design goal of $1/Mflops for some important applications.

During the past quarter, work has focused on readying code for the first production runs on the QCDOC, scheduled for Q1 FY05. This work includes the completion and testing of the QCDOC implementation of the message passing layer, the runtime environment, and the highly optimized "Level 3" code for the most important rate-limiting parts of the code.

6.6.2 Clusters

The performance of commodity components is expected to continue to improve at the traditional Moore's law rate, at least over the next few years. It is expected that improvements in memory access rates and communications latency will have the largest impact on the performance of lattice gauge

theory codes. Don Holmgren at Fermilab (FNAL) and Chip Watson at Jefferson Lab (JLab) have done an analysis of the performance of MILC C-code and Chroma C++ code respectively with SSE enhancements. Based on this data and performance models they have projected cluster performance over the next few years. The estimates from the MILC code at FNAL start from single-processor benchmarks of the currently available 2.8 GHz Intel Prescott P4E with an 800 MHz FSB, which in a single-processor-per-node configuration and a 14^4 local subvolume, is expected to sustain 1.2 GF/s-proc on 128 processors at a current total price of $1.58 per sustained MFlops (including boards, packaging, cabling, and network). Over the next couple of years processor speeds and memory access rates will increase and competition provided by Infiniband will bring down network latencies. Holmgren makes a conservative projection of 3.0 GF/s-proc for single-mass CG inversion by late 2006 at a price of $0.47 per MFlops. With the variant of Dirac fermions used currently in the Chroma application codes at JLab, one sees even better performance with nearly $1.00 per Mflops for the domain wall inverter assembly code at present and similar or better extrapolations into the future.

7 Closing Comments and Summary

It is difficult to convey the effectiveness of application software according to popular measured metrics such as flop rate or computational efficiency alone. Observables measured during production scale runs should be put into the context of the scientific goals of the application. It is noted that for a set hardware system some codes will show a degradation in performance due to the introduction of new physics or the refinement of existing physics, will not show performance changes due to the nature of the underlying core algorithm, or will improve for whatever reasons.

The above stated, the following measures of *computational effectiveness* are reported for each of the applications described in this document. The scientific context of the measures is explained in the previous sections. Most of the codes in the study evolved from development and optimization of software on a target platform where quantifiable performance gains were well defined (through Q3 FY04), to a stabilized state ready for the study of physics during production runs (Q4 of FY04).

7.1 CCSM

7.1.1 Q3, FY04

At the T42 production resolution, the Community Atmospheric Model reports an improvement from 10 simulated years per day of computation to just over 38 simulated years per day of computation on the IBM p690 system, Cheetah, at ORNL. It is noted that for the T85 higher resolution runs, Cheetah was able to sustain 5 simulated years per day of computation. The Cray X1 system at ORNL can sustain 20 simulated years per day of computation at the same resolution.

Community Atmospheric Model with T42 resolution (Simulated years/day)	old	new	gain
	10	38	280%

For the Parallel Ocean Program, the 1 degree production scale runs on the Cray X1 architecture have evolved from 125 years per day of computation to 193 years per day of computation. The code also reports a 47.82% improvement in the total wall time of the combined barotropic and baroclinic code segments on Cheetah.

Parallel Ocean Program at 1 degree resolution (Simulated years/day)	old	new	gain
	125	193	54.4%

The CCSM3.0 code was released at the end of Q3 for production. This is the coupled, full-component code suite.

7.1.2 Q4, FY04

It was determined in late Q3 that the Cray X1 should be targeted for the coupled production CCSM code. This was based upon the fact that two key components of the application, CAM and POP, were ported with substantial performance gains in Q3 (as reported above).

There are not new numbers to quantify the gains in performance for Q4. The effectiveness of the software (effort) must be gauged otherwise.

In what follows, it is noted that *verification* implies that all parts (each component model, coupler, I/O, archiving scripts, the production environment) of the production code are functioning and that subsequent runs of the same problem instance produce the same output.

Validation is harder to achieve and hence more time consuming. It implies that the code is producing the correct model climate. Code that has been validated has necessarily been verified.

The substantial achievements in Q4:

- The production code has run IPCC scenarios essentially non-stop on the IBM systems at ORNL and NERSC.
- The production code port to the Cray X1 has completed verification.
- The production code port to the Cray X1 began but has not completed validation on the Cray X1.
- The CAM3 atmospheric model in conjunction with the CLM3 land model has cleared validation .

  ```
  A high resolution AMIP experiment is being developed
  for this case.
  ```

- The experimental branches of the code that are adding dynamic vegetation, land carbon accounting and a nitrogen cycle have been vectorized and ported to the Cray X1.
- The finite volume dynamical core of the atmosphere has been vectorized and ported to the Cray X1.

  ```
  This enables effective use of the atmospheric model with
  coupled tropospheric and stratospheric chemistry being
  exercised in a prototyping branch. As such, the change
  is not designed for standard IPCC runs.
  ```

- The new POP2.0 ocean code was integrated into the CCSM framework.

  ```
  This is the first (public) of several accepted
  deliverables for DOE's Climate Change Prediction
  Program.
  ```

7.2 SMMC

Shell Model Monte Carlo employs the mature (1953) Metropolis algorithm [1].

7.2.1 Q3,FY04

The code evolved from 20^2 to 50^2 spaces this FY enabling better resolution of previously studied nuclear shells or comparable resolution of more complex shell structures. For production scale runs on the IBM SP Power3 (Seaborg) at NERSC, the changes resulted in a decreased computational performance. The overall scale of the production runs allows almost a factor of 4 larger problem to be studied, however, when compared to this time last year. Please refer to the section on the SMMC code for details.

7.2.2 Q4,FY04

No changes are reported between quarters. The numbers are:

Shell Model Monte Carlo (Performance in MFlops/processor)	old	new	loss
	350	315	10%

7.3 VH-1

In the Virginia Hydrodynamics One code, the number of zone updates per second plus the grid dimensions plus the total number of time steps are the dominant factors that are used to determine the computational effectiveness. The scientific effectiveness of the code has been seen to be limited by the data pipeline since it generates output data during production at ORNL and does the data analysis at NCSU.

7.3.1 Q3,FY04

The performance gain during computation is captured in the following table for the Cray X1 at ORNL.

	old	new	gain
zone updates / second	1140000	1459000	28%
grid points	480^3=110592000 pts	1000^3=1000000000 pts	804 %
time steps	40K	50K	25 %

It is noteworthy that at the beginning of the FY, the production scale runs on Seaborg were approximately 480^3 grid points for 40K time steps at

a rate of 70K zone updates per second. Thus, the rough conversion for zone updates is 70K for Seaborg ::1140K for Cray X1.

7.3.2 Q4,FY04

The performance gain in the data pipeline is reported in terms of effective bandwidth:

	old (Mbps)	new (Mbps)	gain
Cray X1 to NC State	39	216	553.8%

If we consider 400Mbps (a number achieved from a machine at ORNL - not the Cray X1) to be the upper bound on available bandwidth for the path from ORNL to NCSU, then the improved software achieves roughly 54% of this *peak effective bandwidth* as opposed to less than (as of the end of Q3) 10% of it.

In the old implementation, data is written from the Cray X1 to the HPSS disk, pulled from the HPSS disk from a project cluster behind the CCS at ORNL firewall, and written to disk(s) at NCSU from there. The new implementation writes to the NCSU disks directly from the Cray X1.

The performance gains in the data pipeline cannot be understated. The movement of output data is now the limiting factor in studying an exciting new scientific lead discovered during a production run this quarter. (refer to the text)

7.4 RMPS

In FY04 the structure of the RMPS suite of codes changed significantly with the addition of semi-relativistic features (see discussion in text).

7.4.1 Q3,FY04

The parallel semi-relativistic code did not exist in FY03, and therefore the only meaningful metric reported for computational effectiveness is the scale of calculations that became feasible.

In FY03, a serial 89 level neon-like Fe calculation that required diagonalizing matrices of rank 10K was computed. As of Q3 FY04, a parallel 235 level study of neutral neon was achieved that required the complete computation of an eigenspace of rank 50K. The calculations in both cases were performed on Seaborg at NERSC.

7.4.2 Q4,FY04

The computational conversion of the Hamiltonian matrix elements given in jj coupling by DARC to JK coupling format used by the parallel Breit-Pauli suite in the formation of Hamiltonian matrix was completed. One immediate benefit of this development has been is two independent codes to verify results and insight into the measure of the typical variation between them. Secondly, electron scattering of highly-charged atomic systems, produces complex resonance structure that can dominate a cross section. To resolve these resonances fully the integrated DARC package benefits from the parallelization over incident electron energy achieved in $pstgf$. This was an important scientific development.

7.5 QCD

7.5.1 Q3,FY04

Through Q3 FY04 the MILC multi-mass and single mass conjugate gradient inverters (old code) were replaced with inverters written with the QDP/C library (new code). The numbers reported reflect the computational effectiveness of this, the dominant, subroutine for some of the platforms under investigation. Here sm implies single mass and mm denotes multi-mass production runs respectively.

	old	new	gain
NCSA Xeon Cluster (sm)	408	471	15%
Fermilab SciDAC Xeon Cluster (mm)	117	231	97%
ORNL IBM p690 (sm)	450	495	10%

It is noted that the MILC collaboration has worked hard for many years to optimize the single-mass CG inverter, so getting the reported improvements for production parameters is very good.

The previous MILC fermion force subroutine was replaced with a new version that incorporates optimization strategies under development for the QCDOC. An example of the effectiveness of developments this year is provided for the NCSA system here.

old	new	gain
360	538	50%

7.5.2 Q4,FY04

In Q4, the QCD codes have been in a production mode on the hardware platforms mentioned in the text. Thus, no further benchmarks have been generated on them. However, physics results are summarized according to their respective physics cases:

1. Case Study "QCD Thermodynamics"

   This project has (1) predicted the temperature of the transition between ordinary matter and the quark-gluon plasma and (2) predicted fluctuations in the "strange" matter content of the plasma at high temperatures [2]. Both of these results were obtained at an unprecedented lattice resolution of approximately 0.15 fm at the transition temperature, including the effects of strange quarks in the plasma. The next phase of this project includes a study of the equation of state of the quark-gluon plasma. These quantities are key ingredients in models of plasma formation and decay and help identify and analyze collisions that generate the plasma.

2. Case Study "B Decay"

   This project has predicted the decay rates for both D and B mesons into lighter strongly interacting particles plus leptons with an unprecedented accuracy of 11% [3]. Such decays are being studied intensively in the laboratory. Continued improvement coupled with high statistics experimental measurement will have a significant impact on the determination of several key constants of Nature. The next phase will push the lattice resolution in this study to 0.09 fm, permitting a further improvement in accuracy.

3. Case Study "Lattice Generation"

   This project continues to add to the lattice community archive, now approaching 3 Terabytes. We expect this archive to double in size every two to three years. As with all previous MILC collaboration files, the most recent additions will be published on the NERSC archive and eventually the ILDG archive mentioned above. These files are the basis for several efforts, including the "B Decay" project above. Other significant results of this quarter, also based on these lattices, are the determination of the up, down, and strange quark masses [4], a

determination of the mass of the omega minus baryon containing three strange quarks [5], and a determination of the electromagnetic form factors of the nucleon and pion [6].

In Q4 FY04 the ILDG committee agreed upon and published a metadata standard for describing the parameters and conditions under which the files are generated. That is a crucial first step in creating a searchable database for the archive.

Other measures of computational effectiveness and the relation of these to production scale runs are presented in the text.

References

[1] N. Metropolis,A. Rosenbluth,M. Rosenbluth,A. Teller,E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chemical Phys. **21**, 1087-1092, 1953

[2] C. Bernard *et al.* [MILC Collaboration], "Three flavor QCD at high temperatures," arXiv:hep-lat/0409097.

[3] C. Aubin *et al.* [MILC/Fermilab Collaboration], "Semileptonic decays of D mesons in three-flavor lattice QCD," arXiv:hep-ph/0408306. M. Okamoto *et al.* [MILC/Fermilab Collaboration], "Semileptonic $D \to \pi/K$ and $B \to \pi/D$ decays in $2+1$ flavor QCD," arXiv:hep-lat/0409116.

[4] C. Aubin *et al.* [MILC/HPQCD Collaboration] "First determination of the strange and light quark masses from full lattice QCD," Phys. Rev. D **70**, 031504 (2004).

[5] Doug Toussaint and C.T.H. Davies, "The Omega− and the strange quark mass," [arXiv:hep-lat/0409129].

[6] D. B. Renner *et al.* [LHP Collaboration], arXiv:hep-lat/0409130. R. G. Edwards [the Lattice Hadron Physics Collaboration]; arXiv:hep-lat/0409119.

[7] S. Ait-Tahar*et al.* 'Electron scattering by Fe XXII within the Dirac R-matrix approach' Phys. Rev. A 54, 3984-3989 (1996)

[8] www.am.qub.ac.uk/DARC

[9] S. Carter, N. Rao, Private Communication.