National Computational Infrastructure for Lattice Gauge Theory
SciDAC-2 Closeout Report

A proposal in response to Office of Science Notice DE-FG02-06ER06-04 and Announcement Lab 06-04: Scientific Discovery through Advanced Computing.

Lead Institution: Fermilab
Batavia, IL 60510

Lead Principal Investigator and DOE Contact: Paul Mackenzie
Address: Theoretical Physics Department
Fermilab
Batavia, IL 60510
Email: mackenzie@fnal.gov
Phone: 630-840-3347

Office of Science Programs Addressed: High Energy Physics and Nuclear Physics

Office of Science Program Office Technical Contacts: Lali Chatterjee, Ted Barnes, and Randall Laviolette

Participating Institutions and Principal Investigators:

Physics:

Boston University*, Richard Brower † ‡
Brookhaven National Laboratory*, Frithjof Karsch † ‡
Columbia University, Norman Christ †
Fermi National Accelerator Laboratory*, Paul Mackenzie † ‡
Indiana University*, Steven Gottlieb ‡
Massachusetts Institute of Technology*, John Negele † ‡
Thomas Jefferson National Accelerator Facility*, David Richards †‡
University of Arizona*, Doug Toussaint ‡
University of California, Santa Barbara*, Robert Sugar † ‡
University of Utah*, Carleton DeTar ‡
University of Washington, Stephen Sharpe †

Computer Science:

DePaul University*, Massimo DiPierro ‡
Illinois Institute of Technology*, Xian-He Sun ‡
University of North Carolina*, Rob Fowler ‡
Vanderbilt University*, Abhishek Dubey ‡

* Institution submitting an application
† Project Principal Investigator, Member of the USQCD Executive Committee
‡ Institution Principal Investigator

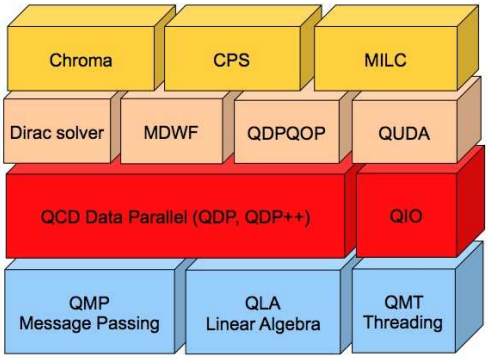

Figure 1: Levels of the QCD API

Under its SciDAC-1 and SciDAC-2 grants, the USQCD Collaboration developed software and algorithmic infrastructure for the numerical study of lattice gauge theories. This work was carried out jointly by high energy and nuclear physicists within USQCD, in collaboration with applied mathematicians and computer scientists. The software and its documentation is publicly available at the USQCD software web site *http://www.usqcd.org/usqcd-software*. The code has been widely adopted within the United States, and is used extensively abroad. It has been instrumental in our effective use of leadership class computers, and of the dedicated computers funded for USQCD through the LQCD Computing Project. The committees which review our hardware program on a yearly basis have consistently emphasized the importance of the work done under our SciDAC grants, and the need for their continuation.

1 Project SciDAC-2 Close Out Report

1.1 The QCD Applications Programming Interface

Under our SciDAC-1 and SciDAC-2 grants, the USQCD Collaboration created the QCD Applications Programming Interface (QCD API), a unified programming environment that enables its users to quickly adapt existing codes to new architectures, easily develop new codes and incorporate new algorithms, and preserve their large investment in existing codes. It has greatly facilitated the efficient use of leadership class computers and commodity clusters. The QCD API was developed as a layered structure which is implemented in a set of independent libraries. It is illustrated in Fig. 1, which shows the three levels of the API and the application codes that sit on top of them. Extensions to these libraries and the maintenance of the API code is an ongoing activity as computer architectures and algorithms change. The API is a critical software underpinning for all of our community application codes, which requires maintenance, testing, version control, documentation and distribution. During SciDAC-2, the components of the API that had been developed in SciDAC-1 were ported and optimized for new architectures (as described in this subsection) and extensions for new architectures and new algorithms were added (as described in the next subsection.)

Level 1 of the API provides the code that controls communications and the core single processor computations. To obtain high efficiency, sometime much of this layer has to be written in hardware specific assembly language; however, versions exist in C and C++ using MPI for transparent portability of all application codes.

Message Passing: QMP defines a uniform subset of MPI-like functions with extensions that (1) partition the QCD space-time lattice and map it onto the geometry of the hardware network, providing a convenient abstraction for the Level 2 data parallel API (QDP); (2) contain specialized communication routines designed to access the full hardware capabilities of computers, such as the Blue Gene line, and to aid optimization of low level protocols on cluster networks. New versions are developed as needed to accommodate changing architectures and algorithms. For example, as discussed below, hooks to combine message passing and threaded code are being added, as is the ability to work with multiple lattice geometries, which is needed for multigrid and domain decomposition algorithms.

Linear Algebra: All lattice QCD calculations make use of a set of linear algebra operations in which the basic elements are three–dimensional complex matrices, elements of the group SU(3). These operations are local to lattice sites or links, and do not involve inter-processor communications. The C implementation has about 19,000 functions generated in Perl, with a full suite of test scripts. The C++ implementation makes considerable use of templates, and so contains only a few dozen templated classes (the required specific classes are generated on demand by the compiler). For both C and C++ it is important to optimize the code for the most heavily used linear algebra modules.

Data Parallel Interface: Level 2 (QDP) contains data parallel operations that are built on QLA and QMP. QDP allows extensive overlapping of communication and computation in a single line of code. By making

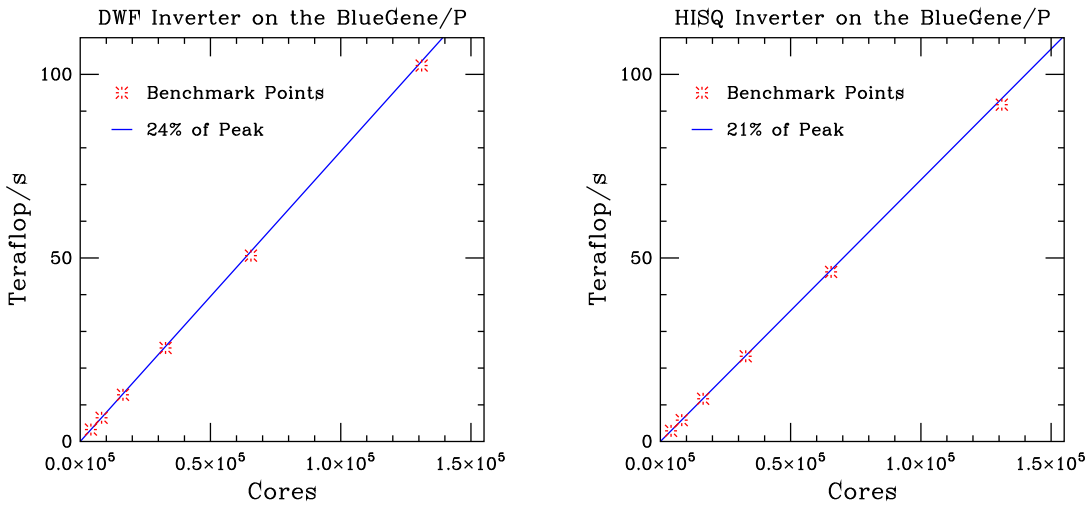

Figure 2: Performance of the Dirac solver on the Blue Gene/P in Tflops as a function of the number of cores for DWF quarks (left panel) and HISQ quarks (right panel). The red bursts are the benchmark points, and the solid blue lines indicate 24% and 21% of peak, respectively. These are weak scaling tests with the number of lattice points per core being held fixed at 6^4 for the DWF solver, and 8^4 for the HISQ solver.

use of the QMP and QLA layers, the details of communications buffers, synchronization barriers, vectorization over multiple sites on each node, etc. are hidden from the users, allowing them to focus on the physics, rather than the subtleties of parallel programming. QDP significantly accelerates the process of developing new codes and optimizing existing ones. It also lowers barriers for entry into the field by graduate students, postdoc and senior scientists from other fields.

Optimized Subroutines: Level 3 (QOP) consists of highly optimized code for a limited number of subroutines that consume a large fractions of the resources in any lattice gauge theory calculation. Most notable among these is the subroutine for the solution of the linear, sparse matrix equations involving the Dirac operator discussed in Sect. 3. To obtain the level of performance at which we aim, it is necessary to optimize these subroutines for each architecture. These routines are generally written with extensive assembly language coding, either employing hand coding or specialized tools, such as Bagel [1] and QA(0) [2], which were developed to generate optimized codes. The data mapping and cache efficiency is extensively tuned. In Fig. 2 we show the performance of the Dirac solver for DWF and HISQ quarks on the Blue Gene/P.

Data Management: QIO enables users to read and write the different types of files that arise in our work in standard formats. It supports a logical partitioning of the computer into I/O partitions with one core per partition handling I/O for the data in just that partition. Thus, in a suitable files system our codes can read and write data in parallel from/to a single file, or in any file system from/to multiple files, and these files can be flattened into one large one offline on a single processor machine. There are no unusual memory requirements for this process. By tuning the size of the I/O partitions, we can maximize the I/O bandwidth and avoid contention. In order to maximize the physics output from the very large computational resources that go into the generation of gauge configurations, we share all gauge configuration files that are created with USQCD resources. To enable this sharing we have created standards for file formats, which QIO adheres to. In addition, we are charter members of the International Lattice Data Grid (ILDG), which established a basic set of meta-data and middleware standards to enable international sharing of data [3, 4], which are also adhered to by QIO.

Application Codes: There are three large, publicly available application code suites developed by members of USQCD that take advantage of the QCD API. Chroma was built directly on QDP++, while the Columbia Physics System (CPS) and the MILC code predate the API, but incorporate key features of it. As a result, all three applications suites benefit immediately from any extensions to or optimizations of the QCD API. Among them, these suites contain all of the codes required for the QCD configuration generation and measurement campaigns we intend to carry out over the next three years. The application code suites and their documentation can also be found at the USQCD software web site.

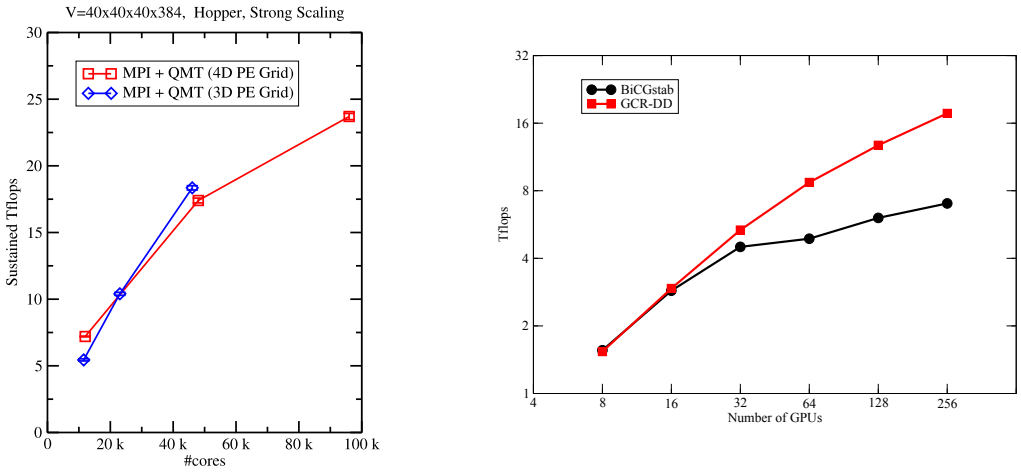

Figure 3: Strong scaling tests of the Wilson-clover inverter with threaded codes. On the left, results on the NERSC Cray XE6, Hopper, using hybrid code with MPI and the QMT library; and on the right, results on the LLNL Edge cluster for up to 256 GPUs with MPI and CUDA threads.

1.2 Recent Extensions of the QCD API

Although the QCD API and the application codes are highly portable, as we move to new computers, we typically have to upgrade the Level 1 and QOP routines. The advent of computer nodes with large numbers of cores and the use of GPU accelerators on the nodes have required that we develop threaded versions of our codes. Furthermore, we, and others in our field, regularly develop new algorithms which must be integrated into the API and the application codes. These developments require that we continually upgrade and extend the QCD API. Here we give a few highlights of this phase of our work.

Hybrid MPI/Threaded Code: It seems clear that in the near future computer nodes will contain large numbers of cores, and that for such machines we will need to employ a hybrid programming model in which communication between nodes is programmed in MPI or QMP, and work on nodes is performed with threaded code. At this early stage we do not believe that a "one size fits all" approach is possible, so we are experimenting with a variety of them. We have obtained early access to the Blue Gene/Q because members of our collaboration at Columbia University and Brookhaven National Laboratory, and our international collaborators at the University of Edinburgh, worked with colleagues at IBM on its design. They are well along in the development of code for domain wall fermions, and find that a hybrid MPI/OpenMP approach works well. They have also produced a highly optimized DWF solver using the Bagel tool [1], which produces assembly code for the Blue Gene/Q's PowerPC processors. Similar code for HISQ and Wilson-clover quarks will follow. For GPU accelerators, we are using CUDA threads on the GPUs combined with POSIX threads on the CPU, and MPI between nodes, while for computers with Intel and AMD multi-core processors, such as the Cray XE series, we have implemented a new threaded library, QMT. Our long range goal is to provide a single uniform data parallel interface so that the applications programmer does not need to be aware of the details of the hybrid code. In Fig. 3 we show strong scaling results for threaded code on NERSC's Cray XE6, Hopper, and on the Edge cluster at LLNL.

The QUDA GPU Library: Starting in 2008, we have explored high performance Dirac solvers in CUDA on NVIDIA GPUs [5]. This effort was initially supported by NSF funding, but has rapidly expanded into a major SciDAC project with the development of the QUDA (QCD in CUDA) library [6, 7, 8], and the rapid deployment of GPU accelerated clusters at Jefferson Laboratory and Fermilab. Our ability to respond rapidly to this new architecture demonstrates the advantage of our clear factorization of Level 3 solvers in the QCD API. At present the QUDA library has expanded to include all Dirac solvers used in QCD (Wilson-Clover, HISQ/asqtad, domain wall and twisted mass). The result has been a dramatic improvement in price/performance for a range of analysis work that is dominated by Dirac solvers. The most recent advance has been the extension of code from single to multiple GPUs. The multiple-GPU codes enables us to analyze the full set of lattices sizes generated by USQCD members with excellent weak scaling. In a

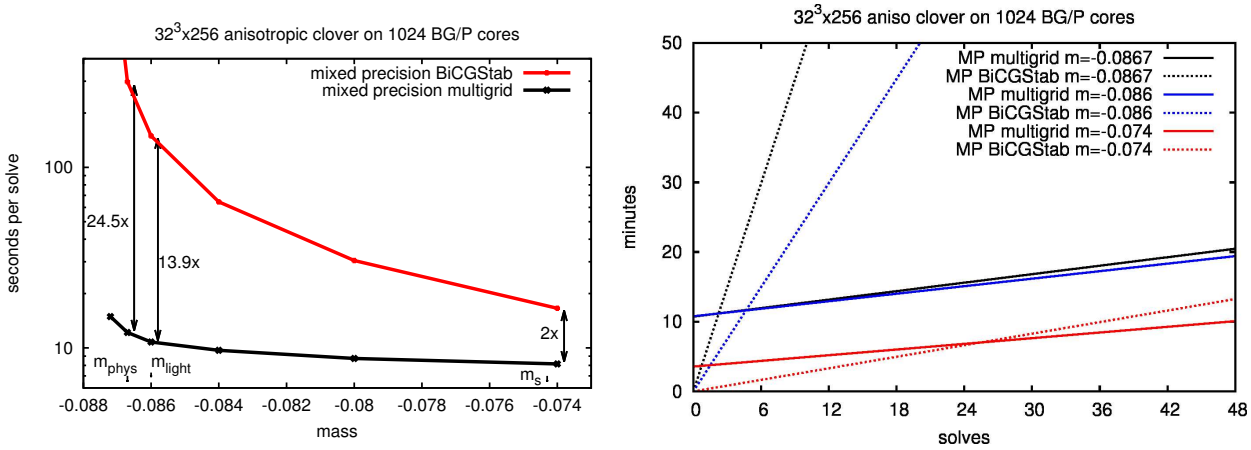

Figure 4: The left panel shows the marginal wall-clock per solve for the multigrid algorithm compared with our best BiCGStab Krylov solver on the Blue Gene/P for a $32^3 \times 256$ lattice with the Wilson-Clover Dirac operator. The right panel shows the total time including the setup for multiple solves on the same configuration by the multigrid and BiCGStab algorithms as a function of the number of solves [9].

paper presented to Super Computing 2011 we demonstrated that we have achieved good strong scaling on up to 256 GPUs for the HISQ/asqtad and Wilson-Clover solvers running on the Edge cluster at LLNL [9].

Solvers for the Dirac Operator: The solver for the lattice Dirac operator has traditionally been a dominant focus of algorithm and specialized software development because of its central role in all QCD codes. A large variety of Krylov solvers have been used with the conjugate gradient and BiCGStab being the current work horses in many production codes. Data layout to improve cache behavior and hand coded assembly kernels are commonplace. For example the Möbius [10, 11] domain wall fermion (MDWF) solver uses Morton ordering in its internal data representation [2], and the QUDA code employs specializes mappings and a novel mixed precision schemes from half (16 bits) to single (32 bits) to double (64 bits) in order to provide double precision accuracy with reduced data traffic between the processor and the memory. A new area of activity beginning to show great promise is the use of multigrid methods [12]. Lattice gauge theorists have attempted to apply multigrid methods to QCD for over twenty years [13]. In collaboration with applied mathematicians from TOPS, we have finally succeeded in formulating an adaptive multigrid solver for Wilson-clover [14]. In the left hand panel of Fig. 4, we show the speedup in the time for one additional solution provided by the multigrid solver compared with our best BiCGStab Krylov solver – nearly a 25x speed up as we move to the physical light quark mass limit. The multigrid algorithm has an overhead to construct its preconditioner, and in the right hand panel of Fig. 4 we show the number of solves with different right hand sides needed to amortize this overhead sufficiently so that the multigrid solver outperforms the BiCGStab one. In some measurement routines, such as those involving disconnected diagrams, hundreds of solves are required on each configuration, so the multigrid algorithm already offers a major improvement over BiCGStab. For the physical light quark mass, the crossover occurs for two or three solves, leading to the possibility of using multigrid in our configuration generation work. This is the beginning of a new opportunity for multi-level algorithms for other parts of our code, and will become increasingly important as the quark masses are reduced and lattice sizes increased. In this same spirit, we are exploring and implementing a variety of "deflation" and Schwarz domain decomposition methods [15].

Improved Hybrid Monte Carlo Evolution: Besides the Dirac solvers, the other major consumer of floating point operations in lattice field theory codes is the symplectic integrator for the molecular dynamics equations that arise in the hybrid Monte Carlo algorithms used to generate gauge ensembles. Over the period of the SciDAC grants, a major advance has been the development of the Rational Hybrid Monte Carlo (RHMC) [16], which is implemented in all of our major application codes. It typically results in a two to four times speedup in the generation of gauge configurations. An even higher order symplectic Force Gradient integrator has been designed [17], which promises further improvements in the next generation of gauge configurations on very large lattices.

A Appendices

A.1 References Cited

[1] P. Boyle, *The Bagel web site*, *http:www2.ph.ed.ac.uk/ paboyle/bagel/Bagel.html* .

[2] A. Pochinsky, *Writing efficient QCD code made simpler: QA(0)*, PoS **LATTICE2008**, 040 (2008).

[3] M. G. Beckett, B. Joo, C. M. Maynard, D. Pleiter, O. Tatebe and T. Yoshie, *Building the International Lattice Data Grid*, Comput. Phys. Commun. **182** (2011) 1208 [arXiv:0910.1692 [hep-lat]].

[4] See International Data Lattice Grid (ILDG) website: *http://www.usqcd.org/ildg/*

[5] K. Barros, R. Babich, R. Brower, M. A. Clark and C. Rebbi, *Blasting through lattice calculations using CUDA*, PoS **LATTICE2008**, 045 (2008) [arXiv:0810.5365 [hep-lat]].

[6] M. A. Clark, R. Babich, K. Barros, R. C. Brower and C. Rebbi, *Solving Lattice QCD systems of equations using mixed precision solvers on GPUs*, Comput. Phys. Commun. **181**, 1517 (2010) [arXiv:0911.3191 [hep-lat]].

[7] S. Gottlieb, G. Shi, A. Torok and V. Kindratenko, *QUDA programming for staggered quarks*, PoS **LATTICE2010**, 026 (2010).

[8] R. Babich, M. A. Clark and B. Joo, *Parallelizing the QUDA Library for Multi-GPU Calculations in Lattice Quantum Chromodynamics*, arXiv:1011.0024 [hep-lat].

[9] R. Babich, M. A. Clark, B. Joó, G. Shi, R. C. Brower and S. Gottlieb, Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11), 11 (2011), arXiv:1109.2935.

[10] R. C. Brower, H. Neff and K. Orginos, *Moebius fermions: Improved domain wall chiral fermions*, Nucl. Phys. Proc. Suppl. **140** (2005) 686 [arXiv:hep-lat/0409118].

[11] R. C. Brower, H. Neff and K. Orginos, *Moebius fermions*, Nucl. Phys. Proc. Suppl. **153**, 191 (2006) [arXiv:hep-lat/0511031].

[12] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn and C. Rebbi, *Adaptive Multigrid Algorithm for Lattice QCD*, Phys. Rev. Lett. **100**, 041601 (2008) [arXiv:0707.4018 [hep-lat]].

[13] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick J. C. Osborn,and C. Rebbi, *Adaptive multigrid algorithm for the lattice Wilson-Dirac operator*, Phys. Rev. Lett. **105**, 201602 (2010) [arXiv:1005.3043 [hep-lat]].

[14] J. C. Osborn, R. Babich, J. Brannick, R. C. Brower, M. A. Clark, S. D. Cohen and C. Rebbi, *Multigrid solver for clover fermions*, PoS **LATTICE2010**, 037 (2010) [arXiv:1011.2775 [hep-lat]].

[15] M. Luscher, *Solution of the Dirac equation in lattice QCD using a domain decomposition method*, Comput. Phys. Commun. **156**, 209 (2004) [arXiv:hep-lat/0310048].

[16] M.A. Clark, A.D. Kennedy and Z. Sroczynski, *Exact 2+1 flavour RHMC simulations*, Nucl. Phys. (Proc. Suppl.) **140**, 835 (2005); M.A. Clark, Ph. de Forcrand and A.D. Kennedy, *Algorithm Shootout: R versus RHMC*, PoS LAT2005, 115 (2005).

[17] A. D. Kennedy, M. A. Clark and P. J. Silva, *Force Gradient Integrators*, PoS **LAT2009**, 021 (2009) [arXiv:0910.2950 [hep-lat]].

1 Boston Univesity Software Co-ordination Activities

At Boston University, Richard Brower serves as the Chair for the Software Coordinating Committee and as a member of the USQCD Executive Committee. For the 6 years of the SciDAC-2 grant, a major activity for Brower has been monitoring the work of the ten universities and three national laboratories that are funded to carry out work under this grant. The Software Committee with all SciDAC developers has weekly teleconferences and annual face to face software workshops to plan and track the projects. In addition reports on the software status are given during our annual USQCD "All Hands Meetings" and at many lattice workshop such a the "Lattice Meets Experimentalist" series covering the 4 major topical areas: the Intensity Frontier, the Energy Frontier, Nuclear Structure and High Temperature QCD.

In his capacity as Software Co-ordinator, Brower obviously contributes to help guide the projects activities at the other participation institutions. These contributions are however limited for the most part to helping to find ways to strength the interaction between experts in each project and to promote code development, testing and implementation into productions. Beyond this largely administrative role, the SciDAC project at Boston University has made significant contribution, initiating and sustaining three important but more restricted software projects:

- (i) The development of the high performance GPU software library called QUDA [1] (or QCD in CUDA) for NVIDIA accelerated systems [2, 3, 4],
- (ii) The development of multi-scale algorithms for lattice Dirac solvers [5, 6]
- (iii) The extension of lattice field theory method [7] to address Beyond the Standard Model (BSM) strong dynamics that may underly the Higgs phenomena under active experimental investigation at the Large Hadron Collider.

Over the periodic of this grant, each of these areas has developed into larger projects with an expanded team of developers at other participating SciDAC-2 institutions.

Under joint funding from SciDAC-2 and the National Science Foundation, Boston University has been fortunate to have a series of remarkably talented postdoctoral fellows: James Osborn (now the Computational Scientist at the ALCF and a Fellow of the Computation Institute at The University of Chicago), Michael A. Clark (visitor at Harvard-Smithsonian Center for Astrophysics and now full time software engineer at NVIDIA continuing with GPU lattice field theory software), Ron Babich (visitor at the Pittsburgh Super Computer Center and also full time at NVIDIA with continuing collaboration on lattice field theory software), Saul Cohen (now at University of Washington and INT in Seattle). The current Boston University postdoctoral fellows in lattice field theory are Michael Cheng and Oliver Witzel contributing to GPU, multigrid and BMS software development. Meifeng Lin was supported under the SciDAC-2 NFE period until March 14, 2013 before joint James Osborn at ANL under SciDAC-3.

2 QUDA: GPU software development

In the summer of 2008, Rebbi and Brower enlisted a graduate student in statistical physics, Kipton Barros, to explore the GPU architecture for lattice field theory. In collaboration with our postdoctoral fellow, Mike Clark, he obtained a performance in excess of 100 Gigaflops on a single 240 core Nvidia GTX280 GPU. Specifically, at Boston University methods developed by Mike Clark and Ron Babich using multiple precision solvers [3] and a variety of tricks to reduce bandwidth to the device memory on the GPU card have demonstrated an additional factor of two to three in performance for the Wilson inverter. The result has been the development of a software library, QUDA [9] for QCD Dirac solvers on GPUs, which has enabled substantial new analysis capability on the ARRA GPU cluster funding at Jefferson Laboratory and the LQCD funded GPU cluster at Fermi Laboratory. The CUDA coding effort has expanded to include work at Jefferson Laboratory by Balint Joo, Jie Chen and Robert Edwards focused on integration into production code and extensions to multi-GPU codes, work by Joel Giedt at Rensselaer focused on the Domain Wall operator, and work by Steve Gottlieb and Guochun Shi at NCSA and Carleton DeTar and Justin Foley at Utah on the MILC staggered code [8]. (See full QUDA team at *https://github.com/lattice/quda*.) The goal is to develop a new GPU implementation of the QCD API, which promises to reduce the cost/flop substantially both for analysis and lattice generation code [9]. The major contribution of the QUDA library under SciDAC-2 was to the analysis code using the Wilson-clover solver on the anisotropic Wilson lattices used for the excited state calculations at Jefferson Laboratory. In addition it has laid the ground work for application to Lattice generations for Wilson for the nuclear program and Staggered fermion for MILC program at the Intensity Frontier. Under SciDAC-3 the goal it full production code scaling to O(1000) GPUs for the INCITE program on the Titan Cray/GPU machine at Oak Ridge and NSF PRAC program on the Blue Waters installation at NCSA.

3 Multigrid Algorithm Development

Brower, Rebbi and David Keyes have led an effort in collaboration with TOPS applied mathematicians to employ multigrid methods for lattice QCD. After nearly four years of effort this team, which includes Mike Clark, Ron Babich and applied mathematicians James Brannick (Penn State), Steve McCormick (Colorado University) and others, constructed the first successful multi-grid lattice Dirac inverter [5, 6, 10]. The inverter for Wilson Fermion propagators demonstrates uniform convergence in the chiral limit, and already shows a factor of ten to twenty five improvement in execution time on large state of the art lattices. James Osborn and Andrew Pochinsky had designed and implemented an extension to the QDP API to accommodate multiple lattices, and, in collaboration with Clark and Saul Cohen, implemented a Level 3 multigrid inverter for the Wilson-clover operator in production code. Saul Cohen, who joined the BU group in the summer of 2009, has formulated the first multigrid inverter for domain wall fermions [11] which will be a major protect to realize in production under SciDAC03. Recently simple version of Schwarz domain decomposition (or block Jacobi) preconditioner has been applied to communication mitigation for the multi-GPU solver for Wilson and Staggered lattice fermion allow for good strong scaling to 256 GPUs [4]. This is crucial step in the extension software for full gauge generation on capability platforms for the Titan and Blue Waters 2

hybrid systems.

4 Beyond the Standard Model (BSM) Software

In the last few years the Lattice Field Theory research focus at Boston University has made a transition from lattice QCD to the study of new strongly interacting gauge theories for BSM studies at the TeV energy scale. Nearly five years ago Appelquist, Brower, Fleming, Osborn, Rebbi and Vranas have formed the Lattice Strong Dynamics collaboration (`http://www.yale.edu/LSD`) aimed at exploring non-perturbative scenarios beyond QCD, which may well be part of the new physics discovered at the LHC. A large range of options are described in an early white paper [12] in 2007 and an initial workshop on *Lattice Gauge Theory for LHC Physics* was held at Livermore May 2-3, 2008, followed workshop at Boston University Nov. 6-7, 2009 , *Lattice Meets Experiment 2010: Beyond the Standard Model* at FNAL on October 14-15, 2011. and most recently at University of Colorado at Boulder Oct 2627, 2012, http://www-hep.colorado.edu/ schaich/lat-exp-2012/. at University of Colorado at Boulder Oct 2627, 2012. The initial projects were chosen in areas close to QCD itself [13] to mitigate the risk due to the need to develop, test and verify a range of new codes and algorithm. Boston University's focus has been on the *S*-parameter [14], which places one of the most stringent constraints on technicolor models and problem of disconnected diagrams, specifically computing the *ss* condensate in the proton needed to estimate the cross section for the direct detection of SUSY neutralino as a possible candidate for dark matter. Both of these project are limited by the cost of the domain wall Dirac solvers. Collaboration with Andrew Pochinsky at MIT on a fast code [15] for the Möbius domain wall algorithm [16, 17, 18, 19] and the multi-GPU implementation on QUDA are two steps to address this problem.

After the discovery of the Higgs scalar at 126 Gev last July 4, 2013 the projects in the BSM strong dynamics has rapidly refocused on specific theories and generic mechanism that are most likely to produce a light scalar consistent with this new experimental reality. This interaction between experimental discoveries and non-perturbative theoretical calculation for composite Higgs theories is of course the classic interaction required to make progress at the frontier of any new physics. During period of NFE (No-fund Extension) for the SciDAC Year 6 Funding, Boston University was able to complete support the Meigfeng Lin to bring to collaborate with James Osborn on the first prototype of a Framework for Unified Evolution of Lattices (FUEL) as planned in our SciDAC-2 grant. This tool is designed to enable more the rapid development and testing of new algorithms for configuration generation based on top level control using the scripting language Lua [20] . The goal is to have this tool in a beta version to be help develop new BSM software and extend the Multigrid algorithms for lattice evolution code. The actual use of the FUEL framework is beyond the scope of SciDAC-2 but is expected to play a central role during SciDAC-3. In addition under the SciDAC-2 NFE, prototype for the use of the NERSC archive a lattice repository for the ILDG (International Lattice Data Grid) was advanced with the help of Oliver Witzel in collaboration with FNAL.

5 List of Publications

[1] The QUDA (QCD in CUDA) library and github repository is at *https://github.com/lattice/quda*

[2] K. Barros, R. Babich, R. Brower, M. A. Clark and C. Rebbi, "Blasting through lattice calculations using CUDA," PoS **LATTICE2008**, 045 (2008) [arXiv:0810.5365 [hep-lat]].

[3] M. A. Clark, R. Babich, K. Barros, R. C. Brower and C. Rebbi, "Solving Lattice QCD systems of equations using mixed precision solvers on GPUs," Comput. Phys. Commun. **181**, 1517 (2010) [arXiv:0911.3191 [hep-lat]].

[4] R. Babich, M. A. Clark, B. Joo, G. Shi, R. C. Brower and S. Gottlieb, "Scaling Lattice QCD beyond 100 GPUs," arXiv:1109.2935 [hep-lat].

[5] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn and C. Rebbi, "Adaptive Multigrid Algorithm for Lattice QCD," Phys. Rev. Lett. **100**, 041601 (2008) [arXiv:0707.4018 [hep-lat]].

[6] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick J. C. Osborn,and C. Rebbi, "Adaptive multigrid algorithm for the lattice Wilson-Dirac operator," Phys. Rev. Lett. **105**, 201602 (2010) [arXiv:1005.3043 [hep-lat]].

[7] R. C. Brower, "Lattice gauge theory for physics beyond the standard model," *44th Rencontres de Moriond on QCD and High Energy Interactions, La Thuile, Valle d'Aosta, Italy, 14-21 Mar 2009*

[8] R. Babich, R. Brower, M. Clark, S. Gottlieb, B. Joo, and G. Shi, Progress on the QUDA code suite, Proc. XXVIII International Symposium on Lattice Field Theory(Lattice 2011) (2011).

[9] R. Babich, M. A. Clark and B. Joo, "Parallelizing the QUDA Library for Multi-GPU Calculations in Lattice Quantum Chromodynamics," arXiv:1011.0024 [hep-lat].

[10] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, S. D. Cohen, J. C. Osborn and C. Rebbi, "The role of multigrid algorithms for LQCD," PoS **LAT2009**, 031 (2009)

[11] S. Cohen, R. C. Brower, M. A. Clark and J. C. Osborn, Adaptive Multigrid Algorithm for Domain-Wall Fermion Inversion in Lattice QCD" Proc. XXVIII International Symposium on Lattice Field Theory(Lattice 2011) (2011).

[12] See whitepages at USQCD website: *http://www.usqcd.org/collaboration.html#2007*

[13] T. Appelquist, A. Avarkian, R. Babich, R. C. Brower, M. Cheng, M. Clark, S. Cohen, G. Fleming, J. Kiskis, E. Neil, J. Osborn, C. Rebbi, D. Schaich and P. Vranas, "Toward TeV Conformality," Phys. Rev. Lett. **104**, 071601 (2010).

[14] T. Appelquist, R. Babich, R. C. Brower, M. Cheng, M. Clark, S. Cohen, G. Fleming, J. Kiskis, E. Neil, J. Osborn, C. Rebbi, D. Schaich and P. Vranas, "Parity Doubling and the S Parameter Below the Conformal Window," Phys. Rev. Lett. **106**, 231601 (2011) [arXiv:1009.5967 [hep-ph]].

[15] Pochinsky, Andrew, "Writing efficient QCD code made simpler: QA(0)", PoS LATTICE2008, 040. Also see *http://www.mit.edu/ avp/mdwf/* Jung:lat11

[16] R. C. Brower, H. Neff and K. Orginos, "Möbius fermions: Improved domain wall chiral fermions," Nucl. Phys. Proc. Suppl. **140** (2005) 686 [arXiv:hep-lat/0409118].

[17] R. C. Brower, H. Neff and K. Orginos, "Möbius fermions," Nucl. Phys. Proc. Suppl. **153**, 191 (2006) [arXiv:hep-lat/0511031].

[18] R. Brower, R. Babich, K. Orginos, C. Rebbi, D. Schaich and P. Vranas, "Möbius Algorithm for Domain Wall and GapDW Fermions," PoS **LATTICE2008**, 034 (2008) [arXiv:0906.2813 [hep-lat]].

[19] Yin Jung, "Improved DWF Simulations: Force Gradient Integrator and the Möbius Accelerated DWF Solver" Proc. XXVIII International Symposium on Lattice Field Theory(Lattice 2011) (2011).

[20] See the programming language Lua website: *http://www.lua.org*

[21]

[21] R. C. Brower, C. E. DeTar, R. G. Edwards, D. J. Holmgren, R. D. Mawhinney, W. Watson and Y. Zhang, "National software infrastructure for lattice quantum chromodynamics," J. Phys. Conf. Ser. **46**, 142 (2006).

SciDAC-2 Project – The Secret Life of Quarks – National Computational Infrastructure for Lattice Gauge Theory

BNL closeout report

Lead Institution: Fermi National Accelerator Laboratory (FNAL)
Batavia, IL 60510-5011.

Lead Principal Investigator and DOE Contact: Paul Mackenzie
Address: Fermi National Accelerator Laboratory
106 (WH 3E)
Batavia, IL 60510-5011.
Email: mackenzie@fnal.gov
Phone: (630) 840-3347

Office of Science Programs Addressed: High Energy Physics and Nuclear Physics

Office of Science Program Office Technical Contacts: Amber Boehnlein and George Fai

Participating Institutions and Principal Investigators:

Physics:

Boston University*, Richard Brower † ‡ and Claudio Rebbi †
Brookhaven National Laboratory*, Frithjof Karsch † ‡
Columbia University*, Norman Christ † ‡
Fermi National Accelerator Laboratory*, Paul Mackenzie † ‡
Indiana University*, Steven Gottlieb ‡
Massachusetts Institute of Technology*, John Negele † ‡
Thomas Jefferson National Accelerator Facility*, David Richards † and William (Chip) Watson ‡
University of Arizona*, Doug Toussaint ‡
University of California, Santa Barbara*, Robert Sugar † ‡
University of Utah*, Carleton DeTar ‡
University of Washington, Stephen Sharpe †

Computer Science:

DePaul University*, Massimo DiPierro ‡
Illinois Institute of Technology*, Xian-He Sun ‡
University of North Carolina*, Rob Fowler ‡
Vanderbilt University*, Theodore Bapty ‡

* Institution submitting an application
† Project Principal Investigator, Member of the USQCD Executive Committee
‡ Institution Principal Investigator

Brookhaven National Laboratory and Columbia University: The software development performed at BNL under SciDAC-2 had originally been coordinated by Michael Creutz. Since 2010 work done under the SciDAC-2 extension has been coordinated by Frithjof Karsch. Most of the actual software writing at BNL and their integration into USQCD software packages has been supervised by Chulwoo Jung. Under SciDAC-2 and its extension three Research Associates were employed at BNL that contributed to the software development and data organization for USQCD: Enno Scholz (FY06-FY08, now at University of Regensburg, Germany), Oiver Witzel (FY09-FY11, now at Boston University), Yu Maezawa (since FY12). In addition Efstratios Efstahiadis was partially supported through the SciDAC-2 grant and worked at BNL as an IT Professional to take care of user support for the QCDOC installation at BNL (FY07-FY10). He is now a Technical Director for High Performance Computing Facility at the New York University Center for Health Informatics and Bioinformatics.

Software developers and lattice field theorists at Brookhaven National Laboratory (BNL) work in close collaboration with colleagues at Columbia University, the RIKEN BNL Research Center [1] (RBRC), and the University of Edinburgh. Our SciDAC-2 grant has been used to support application software, in particular the Columbia Physics System (CPS), on the BlueGene/L and BlueGene/P, and future BlueGene architectures. In particular, the development of CPS was crucial for the running of highly optimized code on the QCDOC at BNL and its successful exploitation. Over the entire funding period, until 2010, users support for the QCDOC had been provided by Stratos Efstathiadis and guidance in the development of highly optimized code for this machine has been provided by Chulwoo Jung. QCDOC is a prototype of BlueGene/L that has been developed by IBM and its design received crucial input from the Columbia and BNL groups. BNL operated the QCDOC until the end of 2011.

CPS has been further developed and has been ported to other BlueGene installations. This code has been extensively used on the NYBlue BlueGene/L and BlueGene/P computers at BNL and the BlueGene/L at Livermore and the BlueGene/P computers of the ALCF at Argonne. At Argonne, CPS is routinely used for the generation of lattices and propagators on 32k-core partitions of BlueGene/P. Chulwoo Jung at BNL has been responsible for the majority of the on-going work in optimizing CPS for the BlueGene architectures, as well improving QMP and QIO for these machines. Over the whole period of funding Jung and Stratos Efstathiadis (until 2010) have provided support for USQCD users of USQCD's QCDOC computer at BNL, which were operating until the end of 2011. This included QIO work for that hardware platform. Oliver Witzel at BNL has been improving the heavy quark measurement capability of CPS, managing the conversion of ensembles generated by the RIKEN, Brookhaven Columbia (RBC) and UKQCD collaborations to International Lattice Data Grid (ILDG) format. He has also been working on a local database to manage configurations as moving configurations between computing locations has become a non-trivial task, given their size. The new database will allows us to monitor their storage locations and history.

During the extension of SciDAC-2 we continued the projects mentioned above, with a primary focus on porting CPS to the BlueGene/Q. This platform is expected to be a major source of computing time for USQCD, starting later in FY2012 or FY2013. Working efficiently on the BlueGene/Q architecture requires codes that enable many tens of threads to run on each node. Columbia and BNL, along with colleagues at Edinburgh, have been heavily involved in the development of the IBM BlueGene/Q computer and three prototype racks are operated at BNL since 02/2012. Peter Boyle at Edinburgh has written an extensively optimized Domain Wall fermion inverter for it, which has been run the first hardware. Jung has also ported the full CPS Rational Hybrid Monte Carlo configuration generator code to the BlueGene/Q with OpenMP threading in place for the performance critical parts. This RHMC code is being run on the first BlueGene/Q hardware. These software developments, done in an early stage of the hardware development, ensured that CPS, as well as QIO, QMP and QDP++ are now available and tested on the BlueGene/Q. The high-performance, threaded parallel transport, a central software piece in any lattice QCD code, has been developed by Jung and Boyle All this is now available for USQCD researches and can be used to run production code on BlueGene/Q. A complete configuration generation codes for BlueGene/Q has been prepared and the porting of measurement codes to this multi-threaded environment has started. Central tasks in this effort have been taken over by RBC physicists who were not SciDAC-2 supported, such as Robert Mawhinney (Columbia University), Taku Izubuchi (RBRC), and Chris Dawson (now University of Virginia) and Eigo Shintani (now Tsukuba University, Japan).

New software developments, started during the SciDAC-2 extension period, focus on the improvement of

[1] Supported by The Institute of Physical and Chemical Research of Japan

simulation capabilities with chiral fermions such as domain-wall, overlap, or Mobius fermions. These newly started algorithmic and software oriented developments are part of our effort to provide optimized measurement routines for the BlueGene/Q and get prepared for large scale usage of multi-GPU systems. These new developments include:

1. Improvement of algorithms to analyze the domain wall Dirac operator in the four dimensional formulation, equivalent to the original five dimensional formula. This allows for a smaller memory footprint but will lead to larger computational costs, which is advantageous when using memory band width limited computing resources such as GPUs.

2. We started to implement EigCG for domain wall fermions, which is found to accelerate the Dirac equation solves by a factor seven when applied on currently typical lattice size and quark masses. Also a new algorithm (MADWF), which uses Mobius fermions as an approximation for domain wall fermions in the preconditioning step is under development.

3. We improved the eigenvalue solvers used for chiral quark propagators. An implicitly restarting Lanczos algorithm with polynomial acceleration is being developed and will be implemented in CPS for both original domain-wall fermion and Mobius fermions. In addition to previous studies, the algorithm now is capable of shifting spectrum regions so that the different parts of spectrum can be solved efficiently with independently running programs. The solved eigenvectors can be compressed and decompressed to save on storage and I/O time.

4. We implemented low mode deflation techniques in CPS, which allows us to accelerate the Dirac equation solves by a factor of more than ten for our typical parameter choices. We also implemented low mode averaging, which allow us to reduce the statistical error with no bias. Extensions of the low mode averaging will be explored.

In addition to the software developments performed within the framework of CPS the BNL group also worked during the last two years on the development of software suitable for QCD thermodynamics calculations on GPU enhanced hardware. In close collaboration with Bielefeld University we developed code for the inversion of staggered Dirac matrices on GPUs and build a complete Hybrid Monte Carlo simulation program that is capable to run entirely on GPUs and produces gauge field configurations for many of BNLs thermodynamics projects. All finite density QCD calculations are now routinely done on GPU clusters.

National Computational Infrastructure for Lattice Gauge Theory
SciDAC-2 Closeout Report
Workflow and Reliability Subprojects

A proposal in response to Office of Science Notice DE-FG02-06ER06-04 and Announcement Lab 06-04: Scientific Discovery through Advanced Computing.

Lead Institution: Fermilab
Batavia, IL 60510

Lead Principal Investigator and DOE Contact: Paul Mackenzie
Address: Theoretical Physics Department
Fermilab
Batavia, IL 60510
Email: mackenzie@fnal.gov
Phone: 630-840-3347

Office of Science Programs Addressed: High Energy Physics and Nuclear Physics

Office of Science Program Office Technical Contacts: Lali Chatterjee, Ted Barnes, and Randall Laviolette

Participating Institutions and Principal Investigators:

Physics:

Boston University*, Richard Brower † ‡
Brookhaven National Laboratory*, Frithjof Karsch † ‡
Columbia University, Norman Christ †
Fermi National Accelerator Laboratory*, Paul Mackenzie † ‡
Indiana University*, Steven Gottlieb ‡
Massachusetts Institute of Technology*, John Negele † ‡
Thomas Jefferson National Accelerator Facility*, David Richards †‡
University of Arizona*, Doug Toussaint ‡
University of California, Santa Barbara*, Robert Sugar † ‡
University of Utah*, Carleton DeTar ‡
University of Washington, Stephen Sharpe †

Computer Science:

DePaul University*, Massimo DiPierro ‡
Illinois Institute of Technology*, Xian-He Sun ‡
University of North Carolina*, Rob Fowler ‡
Vanderbilt University*, Abhishek Dubey ‡

* Institution submitting an application
† Project Principal Investigator, Member of the USQCD Executive Committee
‡ Institution Principal Investigator

1 Reliability Project

As part of this project work, researchers from Vanderbilt University, Fermi National Laboratory and Illinois Institute of technology developed a real-time cluster fault-tolerant cluster monitoring framework. This framework is open source and is available for download upon request. This work has also been used at Fermi Laboratory, Vanderbilt University and Mississippi State University across projects other than LQCD.

In future, we will release the framework on a public website. Attached below, is an summary of the work performed. Finally, a list of publications generated during this work is attached.

1.1 2006-2008

During the first year, Vanderbilt and Fermilab evaluated the suitability of a number of solutions OpenNMS and AWARE frameworks as a complete solution for cluster control and monitoring. Both proved to not match our requirements well. However, OpenClovis, an implementation of open specifications for service availability produced by the Service Availability Forum (SAF), did provide suitable functionality to support our needs for both controls and monitoring. However, it was no longer under development nor was it in a form which could be deployed on the existing systems.

Over the summer, the work was started by doing a preliminary investigation to classify the messages sent on the LQCD cluster support email list. This was the first step in automatically locating error-related messages that appear on the email lists.

Then the team from Vanderbilt University started working on an automated fault monitoring and mitigation system for large Lattice QCD clusters. In collaboration with colleagues at FNAL they developed and deployed sensors written in python language to monitor the critical health parameters for the nodes in the cluster. These sensors intelligently capture system health parameters and periodically report values outside of specified normal ranges to a regional node, which is responsible for storing the health data.

One of the initial problems faced during this phase was the affect of sensor executions on the physics computations being performed on the cluster. In experiments it was observed that the asynchronous execution of sensor on cluster nodes caused significant performance impact on a large MILC configuration generation job stream. This indicated the need for coordination in monitoring. The Vanderbilt team then started investigating various optimization and synchronization techniques to minimize these impacts.

FNAL and Vanderbilt also devised a classification system for monitoring data and a schema for storage in a relational database. We completed a prototype system for transfer and automatic storage of all health and cluster-related monitoring information in a database maintained at FNAL.

The personnel who contributed during this period to the cluster reliability subproject are:

- Ted Bapty, Vanderbilt
- Abhishek Dubey, Vanderbilt
- Sandeep Neema, Vanderbilt
- Don Holmgren, Fermilab
- Jim Kowalkowski, Fermilab
- Nirmal Seenu, Fermilab
- Amitoj Singh, Fermilab

1.2 2008-2009

The main part of the work in this phase was the implementation of inter-node synchronization of monitoring agents to minimize jitter that can affect performance of parallel codes. This work resulted in a publication [1]. Also, the existing code that implements heartbeat sensors was refined to increase robustness.

The prototype system (implemented in Python) for transfer and storage of monitoring information developed in the previous year was expanded, and installed in the production system at Fermilab. In addition to node health information, this system was expanded to store process accounting data and data related to batch system and MPI jobs. The latter required modifications to MPI job launch software; from this stored data, the status of the executions of all binaries associated with an MPI job can be correlated with the cluster hardware and batch system information.

This information was used by the SC LQCD II hardware project at Fermilab to determine, report, and optimize time lost on failed MPI jobs. The data accumulated over the last two years has been used in simulations to explore whether reliable predictions of node failure can be made based on various sensor readings.

During this period the team also began the design of actuators, codes which take mitigating action when failures are detected or anticipated based on the automated analysis of monitoring data. The crucial use cases and behavioral requirements for various actuators were investigated and documented. One of the key problems observed during this stage was the communication overhead. Based on the observed traffic patterns, it was decided that a hierarchical configuration (under investigation) will be necessary for scaling to thousands of nodes.

Additionally, we started to work closely with the workflow subproject during these years. An important requirement of the workflow subproject (see Fermilab and Illinois Institute of Technology, above) is that automated workflow execution must be able to recover from participant (atomic tasks, such as jobs in the batch system) failures. During this year, the reliability subproject started to define what it means for participants to succeed or fail in terms of preconditions, post-conditions and invariants.

Several papers including [2, 3, 4] were published during this phase.

The personnel who have contributed during the past period to the reliability subproject are Ted Bapty, Abhishek Dubey, and Sandeep Neema at Vanderbilt, and Jim Kowalkowksi, Nirmal Seenu, Amitoj Singh, and Don Holmgren at Fermilab.

1.3 2009-2010

For the reliability sub-project, the period was spent upgrading the monitoring system developed in the previous years to a modern message passing system based on OMG Data Distribution Services DDS, prototyping plotting and interaction interfaces, and working through the details of providing fault tolerance to a workflow system.

Early in 2009 we met with the CiFTS team (Coordinated infrastructure for Fault Tolerant Systems) from Argonne National Lab to evaluate their reporting API and the sample communications implementation they provide. Their API is of interest to us because it is targeted at HPC applications, and there appears to be buy-in by some low-level software library providers (such as the Ohio State University MPI over Infiniband project, mvapich). CiFTS was a young project at the time of the meeting, with little to no testing on clusters (most testing had been on supercomputers); our team decided to perform an evaluation on clusters. Vanderbilt personnel did preliminary robustness and performance runs on their cluster, as well as an API evaluation. The CiFTS communications software as implemented lacked the robustness needed to serve as the underlying messaging system for LQCD. The API, however, can be used, and an implementation for LQCD message passing using the API has been defined. Consequently, the resulting LQCD system will also be able to receive information from other software, such as mvapich, which reports according to the CiFTS API.

Our previous prototype monitoring/control system, which uses syslog-ng as a communication mid-

dleware layer, has been extended; this prototype system is used on the LQCD production clusters at Fermilab.

During this period, we started the work on the next version of the monitoring framework that used publish-subscribe middleware built upon the OMG Data Distribution Service standard (DDS).

This approach was taken because it allowed us to design actuators (reactive software components) as separate processes that listen to specific command data or message topics. This approach results in robust and efficient bidirectional communication. Moreover, it eliminates single points of failure because of the Reliability QoS provided by DDS. We are currently working on completing the first version of this subsystem using OpenSplice's DDS implementation. This system will serve as the underlying distributed message passing system for both the cluster reliability software system, and for communications within the LQCD workflow system.

During this period, some integration between the workflow team and the reliability team was done. This resulted in the extension of the LQCD workflow management system to become a collaboration of several components: a workflow manager, a workflow instance manager, and participant managers that track the state of job execution using timed state machines as separate threads running on the local machines. Vanderbilt applied its expertise in developing these state models to further the design of both the reliability and workflow projects. The APIs for all abstract components have been developed to include aspects necessary to ensure reliable operation: preconditions, postconditions, and invariant checks.

Exploratory work in user interface also took place during these years, including the use of Matlab, ISIS GME, and other web-based tool to show active displays of performance and alarm conditions, and also to send commands into the system for control purposes.

The personnel who have contributed during the past period to the reliability sub-project are Ted Bapty, Abhishek Dubey, and Sandeep Neema at Vanderbilt, and Jim Kowalkowksi, Nirmal Seenu, and Amitoj Singh at Fermilab. Community interactions include the projects: CiFTS, OpenSplice DDS (Data Distribution Service), OpenDDS, the Fermilab component of the JDEM SOC (Joint Dark Energy Mission Science Operation Center) project, LSST Controls Group (Large Synoptic Survey Telescope), SC2009.

1.4 2010-2012

On the cluster reliability sub-project, a joint effort of Fermilab and Vanderbilt, from February 2010 through August 2010 we continued our work extending a monitoring system based upon the OMG Data Distribution Service (DDS) standard. This approach allows us to design reactive software components as separate processes that listen to specific command data or message topics.

This resulted in a fault-tolerant distributed monitoring approach (RFDMon). RFDMon can be used for measuring system variables (CPU utilization, memory utilization, disk utilization, network utilization, etc.), system health (temperature and voltage of Motherboard and CPU) application performance variables (application response time, queue size, and throughput), and scienti?c application data structures (PBS information and MPI variables) accurately with minimum latency at a speci?ed rate and with controllable resource utilization.

This framework is designed to be tolerant to faults in monitoring framework, self-configuring (can start and stop monitoring the nodes and configure monitors for threshold values/changes for publishing the measurements), aware of execution of the framework on multiple nodes through HEARTBEAT messages, extensive (monitors multiple parameters through periodic and aperiodic sensors), resource constrainable (computational resources can be limited for monitors), and expandable for adding extra monitors on the fly. Since RFDMon uses a Data Distribution Services (DDS) middleware, it can be used for deploying in systems with heterogeneous nodes. Additionally, it provides a functionality to limit the maximum cap on resources consumed by monitoring processes.

We also started work on a model-based, front-end tool using the ISIS Generic Modeling Environment (GME). GME is good for designing system assemblies made up of distributed software components with well-defined interaction behaviors. The modeling and design environment allows a developer

to define process monitoring and control components, along with their resource constraints and relationships, without specifying implementation and deployment details. The component models define the customization, deployment properties and communication methods for a given platform. They also define interaction ports for a component, which include the types of data that the component produces and consumes, along with any other commands that the component may accept or emit. The tool chain allows for a model parser to generate source code for connecting the modeled components using available systems such as DDS and deployment of the modeled configuration on a physical cluster.

During this time, we also developed a formal state machine model for scientific workflow and reliability systems. This includes the use of Vanderbilt's Generic Modeling Environment (GME) tool for code generation for the production of user APIs, code stubs, testing harnesses, and model correctness verification. It is used for creating wrappers around LQCD applications so that they can be integrated into existing workflow systems such as Kepler.

Finally, during this period Fermilab personnel developed a system that tracks in close to real time the status of the various Torque batch systems used on the LQCD productions clusters at the site. This system consists of programs that actively monitor the accounting logs of Torque, noting all job transitions (queued, modified, started, ended, and so forth) and the characteristics of all jobs (nodes, times, resources) in a database. We conceived of this database as a resource required by any workflow system for determining the current and past states of jobs and nodes. However, the database and a large set of queries have proven very useful for many aspects of monitoring and administrating the LQCD clusters, including tracking progress against allocations, troubleshooting jobs and nodes, and managing allocated projects.

The personnel who have contributed during the past period to the reliability sub-project are Ted Bapty, Abhishek Dubey, and Sandeep Neema at Vanderbilt, and Jim Kowalkowksi, Randy Herber, Don Holmgren, Nirmal Seenu, and Amitoj Singh at Fermilab.

Community interactions include the projects: CiFTS, OpenSplice DDS (Data Distribution Service), OpenDDS, the Fermilab component of the JDEM SOC (Joint Dark Energy Mission Science Operation Center) project, LSST Controls Group (Large Synoptic Survey Telescope).

2 Workflow

The goal for the scientific workflow project is to investigate and develop domain-specific workflow tools for LQCD to help effectively orchestrate, in parallel, computational campaigns consisting of many loosely-coupled batch processing jobs.

Major requirements for an LQCD workflow system include: a system to manage input metadata, e.g. physics parameters such as masses, a system to manage and permit the reuse of templates describing workflows, a system to capture data provenance information, a systems to manage produced data, a means of monitoring workflow progress and status, a means of resuming or extending a stopped workflow, fault tolerance features to enhance the reliability of running workflows. Requirements for an LQCD workflow system are available in documentation.

2.1 2006-2007

Fermilab and the Illinois Institute of Technology have worked very closely together since the start of the SciDAC-II LQCD project. Since September 2006, the subproject surveyed existing active workflow projects. A workshop with the VDS (now called Swift) workflow project was held at Fermilab on December 18. Video conferences were held in early 2007 with the Askalon and Kepler workflow project teams. A detailed requirements document for an LQCD workflow system was prepared jointly by IIT and Fermilab [5, 6]. This document had been used in discussions with other workflow projects and served well as a learning tool so that the various parties can understand one another. The Karajan execution engine was installed on one of the Fermilab clusters, and one of the weak decay analysis workflows had been coded in the Karajan language as a test case.

Work after the initial meetings with workflow projects was concentrated on understanding Swift and Askalon in depth. Both of these workflow systems have been installed on the Kaon cluster at Fermilab. As both workflow systems rely on GRID (Globus) tools for dispatching work, some effort was required to layer Globus tools or interfaces on top of the Torque (PBS) resource scheduler used on Kaon. In the case of Swift, we worked with the Swift developers to debug a PBS "provider" for their workflow execution engine (Karajan). This work included debugging and modifying the launcher (mpirun_rsh) from the MVAPICH MPI used on the Kaon cluster.

In both Swift and Askalon, we prototyped a few simplified LQCD workflows, specifically, configuration generation and heavy-light two-point analysis. A number of feature mismatches between these workflow systems and the LQCD workflow requirements were identified. We established working relationships with both the Askalon and Swift development teams, and both groups began working with us to evolve their products to be better suited to LQCD workflows.

The personnel who contributed during this period to the workflow subproject are:

- Jin Hui, IIT
- Pr. Xian-He Sun, IIT
- Don Holmgren, Fermilab
- Jim Kowalkowski, Fermilab
- Luciano Piccoli, Fermilab
- Nirmal Seenu, Fermilab
- Jim Simone, Fermilab
- Amitoj Singh, Fermilab

2.2 2008

Fermilab and IIT participated in the developments from November 2007 until October 2008. The efforts this year on the workflow subproject began with the emphasis on implementing candidate LQCD workflows in two existing systems, Swift (U. Chicago) and Askalon (Innsbruck), which met a number of the requirements that we had established and documented in the first year of the subproject. The candidate workflows were configuration generation [7] and heavy-light two-point analysis. Also, at the February LQCD software committee workshop in Boston, the subproject demonstrated the Chroma regression test suite implemented as Askalon and Swift workflows [8].

During the first six months, the subproject team interacted frequently with both the Swift and Askalon development groups. For Swift, the interactions focused on refining features of the Swift language so that the candidate LQCD workflows could be described completely and in a less awkward fashion. For Askalon, the work involved installation and configuration on Fermilab machines, as well as the implementation of the candidate workflows in the Askalon language. The workflow subproject produced a written report that discusses the suitability of the Kepler, Triana, JBPM, Swift, and Askalon workflow systems to our documented LQCD workflow requirements [9].

Neither Swift nor Askalon, nor the other workflow systems evaluated fully met the requirements for adequately describing and controlling LQCD analysis campaigns. Specifically, we found that they were inadequate in terms of their abilities to parameterize LQCD campaigns, to record and provide detailed run time histories, to record and provide provenance information, or to interact with secondary data storage systems. The focus of the subproject had now shifted to developing a workflow system that relies on front and back-end systems that wrap, and are independent of, existing workflow engines. The front-end software provides the parameterization features. The back-end software provides history, provenance, information necessary for job recovery, and structured storage (and access to) scientific data products. Prototype implementations of both front-end and back-end software were written using Ruby on Rails. The wrapped software was permitted

to be any of the various existing workflow engines (e.g. Swift/Karajan, Askalon, Kepler, Ruote BPM). The wrapped engine layer supervised the dispatch of individual jobs to our batch systems using a set of instructions which were assembled from our front-end system. The engines potentially permitted dispatching of jobs to systems on external clusters or grids. We used Ruote BPM as the workflow engine in our investigations.

Community interactions of the workflow subproject included the various meetings and phone conferences with the Swift and Askalon teams, papers and/or posters for SuperComputing 2007, Lattice 2008, eScience 2008 Conference (SWBES Workshop), participation in tutorials and workshops at SuperComputing 2008, and a talk submission to CHEP09.

The personnel who have contributed during the past period to the workflow sub-project are Xian-He Sun at the Illinois Institute of Technology (IIT), Luciano Piccoli of IIT and Fermilab, and Don Holmgren, Jim Simone, Jim Kowalkowksi, Nirmal Seenu, and Amitoj Singh of Fermilab.

2.3 2009

For the workflow sub-project, the past period of November 2008 through January 2010 was spent iterating on earlier developments, including the design and initial implementation of new subsystems, and continuing the search for better tools. Early in the period, effort included construction of a virtual machine-based test facility, and the review of the prototype LQCD workflow system described and implemented during the previous year (Nov 1, 2007 - Nov 1, 2008). This prototype "wraps" an existing non-LQCD workflow engine (Ruote) with front- and back-end software that parametrize an LQCD user's workflow, provide persistence (job execution history, provenance, and information needed for recovery), and provide structured storage and access to scientific data products. The prototype LQCD workflow system was tested with an additional simple LQCD workflow. The effort then shifted into continuing the evaluation of other existing workflow systems, with the goal of finding a better workflow engine to wrap. The last half of the year was spent in the design of workflow system components and their interactions, and in producing a complete specification and initial implementation of a parameter set language capable of configuring all aspects of LQCD jobs.

The virtual machine (VM) test facility was created to evaluate workflow systems and their interactions with batch submission software. Many of the workflow systems previously evaluated (e.g., Kepler, Askalon) had the disadvantage of being tied to GRID software, making it difficult to install, maintain, and operate in our non-GRID environment. The VM test facility provides the environment needed to do testing without perturbing the existing LQCD production systems. This helped in our understanding of how computing cloud systems might be used to check failure handling software, and it also helped to do configuration and software testing involving multiple nodes without use of the LQCD production clusters.

The design and code review of our prototype workflow system took place in March 2009 [10, 11]; the review covered the basic workflow information storage system that was designed and started during the previous year, and also the use of the Ruote workflow engine. The major conclusions were that (1) the system provided a good start, (2) the design needed to be better documented, (3) several aspects of the design needed to be clarified, and (4) some restructuring of the code was necessary. The review report also called out the need for better testing. In response to the review, documents were added describing the systems design and operation.

Shortcomings in the Ruote workflow engine became evident during the review and during the implementation of additional test workflows. Pegasus was evaluated as a possible replacement. Lack of dynamic and flexible scheduling features, and lack of expressiveness in the Pegasus workflow language, resulted in that version of Pegasus not being able to satisfy enough of the LQCD requirements for adoption as the workflow engine.

Summer 2009 was spent developing designs for necessary workflow components and making a more complete and concise definition of concepts, APIs, and component behaviors. State models were developed to express the correct behavior of running workflow "participants" (individual tasks or functions within the workflow). Parts of the reliability sub-project became coupled to the workflow

project, including the message passing subsystem (see the Vanderbilt/Fermilab subsection). At present the reliability and the workflow designs are moving forward together, and joint meetings are held weekly.

We realized in early fall of 2009 that trying to simultaneously solve problems involving the wrapped workflow engine, and problems involving the front and back-end software, was too difficult and was hindering overall progress. We elected to narrow temporarily the scope of work and concentrate on the front and back-end software, wrapping a more limited but functional - and quite old - existing LQCD workflow engine (Runjob). As the first task, we concentrated on the complete specification of a configuration language suitable for LQCD and the tools to manipulate it. The design work was finished by the end of the year, and the first version of a parser and manipulation library is now complete. Simultaneously work was started to improve the design of the parameter set storage system in order to accommodate the new parameter sets from the configuration language, and also to clean up the code and the interface of the parameter set storage software (as suggested by the earlier review).

The personnel contributing to this workflow sub-project effort during the past period are Xian-He Sun at the Illinois Institute of Technology (IIT), Luciano Piccoli of IIT and Fermilab, and Don Holmgren, Jim Simone, Jim Kowalkowksi, Nirmal Seenu, Amitoj Singh, and Randolph Herber of Fermilab. Community interaction include the following groups: CIFTS (Argonne), Pegasus (ISI at University of Southern California), Kepler (UCSD), Fermilab component of the JDEM SOC (Joint Dark Energy Mission Science Operation Center) project, ESO workflow (European Southern Observatory).

2.4 2010-2011

For the workflow sub-project during February 2010 and August 2010, a large portion of our effort was spent implementing the designs completed during the previous year. These 2009 designs include workflow system components and their interactions, and a complete specification and initial implementation of a parameter set language capable of configuring all aspects of LQCD jobs. The reliability and the workflow implementations moved forward together, and joint meetings continued to be held weekly. In the fall of 2009, we elected to narrow the scope of work and concentrate on the front and back-end software, wrapping a more limited but functional - and quite old - existing LQCD workflow engine (Runfile).

We designed and implemented a system to manage parameters (e.g. masses, algorithmic parameters) that are inputs to LQCD production campaigns. The system is capable of tracking and versioning parameter values through a relational database. A language superset of the JSON data-interchange format was developed to represent parameter sets as easily readable text. A fully functional parser, available in Ruby (as a scripting language), javascript, and in C++, was implemented. A user interface tool was developed by a Fermilab summer intern, who built a GUI editor to manipulate parameter sets based upon the Adobe Air stand-alone web application development platform. This tool was designed to be used to manage collections of parameter set files that collectively form a single configuration of any complex calculation campaign. It eases the maintenance of parameter set files that are stored in version control systems such as SVN and compliments the long-term storage database tools that are better suited for production running and provenance tracking.

Parameter values must typically be substituted into the text input prompts of existing LQCD applications such as MILC. We developed a system to create and manage text templates that represent the inputs to LQCD applications. A given template can be stored as a text in a relational database or as text in a file. Before an LQCD application is started, its input lines would be created by binding parameter values from a parameter set to a specific template. In general, both the number of input lines to an application and their content may depend upon parameter values, for example, when an application must process each item from a list of parameter values. Hence, the system permits the embedding of language constructs such as function calls and flow-control (e.g. "if-then-else" and "foreach" loops) in templates.

Interactions with the JDEM-SOC (Joint Dark Energy Mission - Scientific Operations Center)

project increased during this time. JDEM-SOC team members joined the weekly LQCD meetings and contributed to the workflow engine investigation. They have demonstrated the use of Kepler as a workflow engine. Members from both teams started to define a working model of the interactions between three critical pieces: the program (e.g. MILC), the actor (the management piece of this job), and the workflow engine scheduling element ("director" in Kepler terminology).

The personnel contributing to this workflow sub-project effort during the past period are Luciano Piccoli of IIT and Fermilab, and Don Holmgren, Jim Simone, Jim Kowalkowksi, Nirmal Seenu, Amitoj Singh, and Randolph Herber of Fermilab. Community interaction include the following groups: Kepler (UCSD), and Fermilab component of the JDEM SOC (Joint Dark Energy Mission Science Operation Center) project. Early in the year Piccoli left IIT and Fermilab. Herber has taken on some of his work.

As of July 2011, formal definitions of scientific workflow system components and functions were created to clarify and further the requirements list, and also to reconcile them with other Fermilab projects that involve workflow systems. In addition, an analysis of several common MILC input decks was done to discover configuration practices and rules for this tool. This resulted in finding a few anomalies in production run scripts and also the release of a tool to help diagnose configuration problems. This work also resulted in new documentation for configuration card ordering, dependencies, and desk building rules. A new real use case was also documented and added to the collection of workflows.

We formalized the parameter definition and management system (named fhicl). Fhicl started out as being largely JSON compatible, and has diverged only slightly to accommodate LQCD and other HEP requirements. The fhicl language and C++ binding were already being used within other Fermilab HEP Intensity Frontier projects as a common language for configuring scientific applications. A file system storage and retrieval system was added. These tools were packaged for use as a library. Many features were added, including parameter value substitution, file includes, and parameterized values. A Python binding was delivered for processing fhicl files. The Ruby binding was improved.

A template engine was developed to move LQCD analysis applications toward being usable in workflow systems. This engine was integrated into the "run_onium" scripts. It permits sophisticated parameterization, including generation of blocks of input desk cards using looping constructs. Such a facility permits removal of a portion of the hardcoded processing logic. Tools were also developed to help with integration with the fhicl parameter definitions. A workflow participant (scientific application participating in a workflow) protocol was developed, documented, and tested. Several LQCD application-specific graph execution engines were put together to test this protocol.

We evaluated Kepler as a candidate workflow execution engine and modeling environment and worked with the Kepler team to bring a demonstration system up. This work was done in conjunction with another Fermilab project.

A workflow modeling and execution system was established using mature Vanderbilt ISIS Generic Modeling Environment (GME) toolkit. The system was demonstrated using one of the sample use cases weve collected. The system incorporated and tested several of the concepts and protocols weve developed, including the participant protocol and state machine process behavior management. GME allows for multiple aspect modeling, so the correct program behavior (states) can be modeled independent of the workflow components (participants), and also independent of the deployment on a cluster (node assignments).

3 Achievement highlights

3.1 Extension period

The extension period lasted from August 2011 through December 2011. During that time, the Ruby gem system was set up for deployment of the Ruby code. The Ruby and Python parsers for FHICL were documented. Most of the available time was spent on documenting, expanding, and

debugging the monitoring database for tracking jobs and the associated command line interface tools.

3.2 Uses outside LQCD

3.2.1 FHICL

FHICL, introduced as part of this these projects, was introduced to the Intensity Frontier (IF) experiments at Fermilab. After a series of refinements to the syntax and semantics of the language, it was accepted into the main offline software framework (called art) used by the IF experiments. It is currently used as the configuration language for mu2e, NOvA, g-2, and the experiments using the LArSoft (Liquid Argonne Software) package. A full-featured C++ binding was developed and is distributed as a standard package.

3.2.2 Wrapper protocol

The wrapper protocol that was started within the LQCD workflow project permits embedding (or adopting) existing external applications into a workflow execution engine and processing environment. The wrapper protocol was taken and refined for use in the JDEM (Joint Dark Energy Mission) Science Operation Center, a Cosmic Frontier project assigned to Fermilab. The protocol was expanded to use FHICL. More recently this protocol was also used as the basis for the engineering of the DES (Dark Energy Survey) workflow software framework. The design elements were reused because they demonstrate a clean separation between coordination functions and functions that carry out the science.

3.2.3 Problem analysis reuse

In addition to the wrapper protocol, the basic set of requirements, goals, and system architecture for the LQCD workflow and reliability projects have been reused in discussions with additional experiments. The requirements and the system architecture formed a starting point for work that was done on JDEM. They have also served as a starting point for the reengineering discussions with DES. They have been heavily relied on in discussions with LSST on a potential level-3 toolkit project.

3.3 Summary

- Implemented a software system to manage parameters. This includes a parameter set language based on a superset of the JSON data-interchange format, parsers in multiple languages (C++, Python, Ruby), and a web-based interface tool. It also includes a templating system that can produce input text for LQCD applications like MILC.
- Implemented a monitoring sensor framework in software that is in production on the Fermilab USQCD facility. This includes equipment health, process accounting, MPI/QMP process tracking, and batch system (Torque) job monitoring. All sensor data are available from databases, and various query tools can be used to extract common data patterns and perform ad hoc searches. Common batch system queries such as job status are available in command line tools and are used in actual workflow-based production by a subset of Fermilab users.
- Developed a formal state machine model for scientific workflow and reliability systems. This includes the use of Vanderbilt's Generic Modeling Envirnment (GME) tool for code generation for the production of user APIs, code stubs, testing harnesses, and model correctness verification. It is used for creating wrappers around LQCD applications so that they can be integrated into existing workflow systems such as Kepler.

- Implemented a database system for tracking the state of nodes and jobs managed by the Torque batch systems used at Fermilab. This robust system and various canned queuries are used for many tasks, including monitoring the health of the clusters, managing allocated projects, producing accounting reports, and troubleshooting nodes and jobs.

4 List of Publications

[1] A. Dubey, G. Karsai, and S. Abdelwahed, "Compensating for timing jitter in computing systems with general-purpose operating systems," in *ISORC*, 2009, in press.

[2] A. Dubey, S. Nordstrom, T. Keskinpala, S. Neema, T. Bapty, and G. Karsai, "Towards a model-based autonomic reliability framework for computing clusters," in *EASE '08*, 2008, pp. 75–85.

[3] A. Dubey, S. Neema, J. Kowalkowski, and A. Singh, "Scientific computing autonomic reliability framework," in *eScience*, 2008, in press.

[4] L. Piccoli, A. Dubey, J. N. Simone, and J. B. Kowalkowlski, "Lqcd workflow execution framework: Models, provenance and fault-tolerance," *Journal of Physics: Conference Series*, vol. 219, no. 7, 2010. [Online]. Available: http://stacks.iop.org/1742-6596/219/i=7/a=072047

[5] LQCD Workflow Team. (2006, December) Lqcd workflow functional requirements. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6332/FunctionalRequirements.pdf

[6] ——. (2009, February) Glossary of workflow terms. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6363/LQCD-Glossary.pdf

[7] ——. (2008, March) Confgen analysis: Managing parameters, provenance, and secondary products. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6338/ConfGenAnalysis-v2.pdf

[8] ——. (2008, February) Workflow systems for lqcd. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6348/LQCD-Workflows-SciDAC-swc-2008.pdf

[9] ——. (2009, October) Workflow evaluation: Swift and askalon. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6334/WorkflowEvaluation.pdf

[10] E. N. W.E. Brown, D. Holmgren and M. Paterno. (2009, March) Lqcd workflow (code) mini-review. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6336/MiniReviewMarch09.pdf

[11] L. Piccoli. (2009, February) Lqcd workflow code review intro. slides. [Online]. Available: https://cdcvs.fnal.gov/redmine/attachments/6366/LQCD-code-review-slides.pdf

[12] R. Mehrotra, A. Dubey, S. Abdelwahed, and W. Monceaux, "Large scale monitoring and online analysis in a distributed virtualized environment," in *Engineering of Autonomic and Autonomous Systems (EASe), 2011 8th IEEE International Conference and Workshops on*, april 2011, pp. 1–9.

[13] R. Mehrotra, A. Dubey, J. Kwalkowski, M. Paterno, A. Singh, R. Herber, and S. Abdelwahed, "Rfdmon: A real-time and fault-tolerant distributed system monitoring approach," , 10 2011.

[14] A. Dubey, S. Nordstrom, T. Keskinpala, S. Neema, T. Bapty, and G. Karsai, "Towards a verifiable real-time, autonomic, fault mitigation framework for large scale real-time systems," *Innovations in Systems and Software Engineering*, vol. 3, pp. 33–52, March 2007.

[15] A. Dubey, L. Piccoli, J. B. Kowalkowski, J. N. Simone, X.-H. Sun, G. Karsai, and S. Neema, "Using runtime verification to design a reliable execution framework for scientific workflows," in *Proceedings of the 2009 Sixth IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, ser. EASE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 87–96. Online: http://dx.doi.org/10.1109/EASe.2009.13

[16] S. Nordstrom, A. Dubey, T. Keskinpala, R. Datta, S. Neema, and T. Bapty, "Model predictive analysis for autonomic workflow management in large-scale scientific computing environments," in *EASE '07*, 2007, pp. 37–42.

[17] L. Piccoli, X.-H. Sun, J. Simone, and et al., "The lqcd workflow experience: What we have learned," in *SuperComputing 2007*, 2007.

[18] L. Piccoli, J. Simone, J. Kowalkowski, and et al., "Tracking lqcd workflows," in *Lattice 2008*, 2008.

[19] L. Piccoli. (2006, October) Workflow project. Online: https://cdcvs.fnal.gov/redmine/attachments/6341/wf-slides-boston.pdf

[20] L. Piccoli and J. Kowalkowski. (2008, September) Workflow project status. Online: https://cdcvs.fnal.gov/redmine/attachments/6350/SciDAC-SEP-11-2008.pdf

1 MILC institution closeout: Arizona, Indiana, Utah

The MILC collaboration consists of approximately eight senior and nine junior members at ten institions, mostly in the USA. [1] Almost all of its scientific work is done with the MILC code, an integrated package of some 200,000 lines of scientific application codes and a library of generic supporting codes. It has been in use worldwide and freely available to the public since the early 1990's, and has grown and evolved over the years to meet our evolving physics goals and rapid changes in computer architecture and capability.

Under SciDAC-2 three MILC-collaboration institutions, the University of Arizona, Indiana University, and the University of Utah, were responsible for a large number of important revisions to the code that exploit the evolving SciDAC code suite. These institutions have also contributed important new SciDAC modules. These changes have made the MILC code much more flexible, they have allowed us to keep abreast of our changing scientific objectives, and, consequently, they have enabled rapid scientific progress.

Over the six years of the SciDAC-2 grant, the MILC institutions received support for approximately 1/2 postdoctoral research associate each. We list accomplishments and publications resulting from this support:

- Arizona
  - Implementation of the highly improved staggered quark (HISQ) algorithm. This algorithm has become a mainstay of our current physics efforts. It is yielding greatly improved control of discretization errors, which, in turn, is producing higher precision results [1, 2, 3, 4, 5].
  - Porting the HISQ algorithm into SciDAC Level 3. This step is essential for taking advantage of architecture-specific optimizations in the SciDAC code suite [6].
  - Production of a large set of gauge field configurations with the HISQ action. They are the basis for many physics analyses, including spectroscopy, decay constants, and a wide variety of weak

[1] A. Bazavov, R.S. Van de Water (Brookhaven), C. Bernard, M. Lightman (Washington U.), C. DeTar, J. Foley, L. Levkova, M. Oktay (U. Utah), J. Kim, D. Toussaint (U. Arizona), S. Gottlieb, R. Zhou (Indiana U.), U.M. Heller (APS), J.E. Hetrick (U. Pacific), J. Laiho (Glasgow U.), J. Osborn (Argonne), R.L. Sugar (UC, Santa Barbara)

matrix elements [7]. We publish worldwide the gauge field configurations we produce [8].

- Indiana
  - Integration of the MILC code with the SciDAC QUDA multi-GPU asqtad fermion solver. This gives our code a factor of four or more speedup for much of our analysis projects. [9, 10, 11, 12, 13]. The code has been put into production measuring electromagnetic splittings of hadron masses [14].
  - Development of an improved gauge-force module.
  - Development of code to do the spectroscopy of baryons containing a charm or bottom quark [15, 16].
- Utah
  - Two major revisions of the MILC code suite that provides support for the HISQ algorithm, greatly expanded support for hadron interpolating operators, the FFTW (Fourier transform) package, the PRIMME (eigenvalue) package, random color-wall sources, momentum twists, open-meson code for B Bbar mixing calculation, HISQ equation of state, etc. [17]. The new features of the code are yielding new physics results [18, 19, 20, 21, 22, 23, 24, 25, 26].
  - A user manual for the MILC code suite [17].
  - Development of QUDA GPU HISQ link-fattening and fermion force modules. These are the first steps toward developing the ability to do full hybrid molecular dynamics calculations on GPU's.
  - Testing the improved heavy-quark Oktay-Kronfeld action [27].

2 Publications and conference proceedings resulting from this work

References

[1] A. Bazavov et al. HISQ action in dynamical simulations. *PoS*, LATTICE2008:033, 2008.

[2] A. Bazavov et al. Progress on four flavor QCD with the HISQ action. *PoS*, LAT2009:123, 2009.

[3] A. Bazavov, C. Bernard, C. DeTar, W. Freeman, Steven Gottlieb, et al. Simulations with dynamical HISQ quarks. *PoS*, LATTICE2010:320, 2010.

[4] A. Bazavov et al. Scaling studies of QCD with the dynamical HISQ action. *Phys.Rev.*, D82:074501, 2010.

[5] A. Bazavov et al. Properties of light pseudoscalars from lattice QCD with HISQ ensembles. *PoS*, LATTICE2011:107, 2011.

[6] USQCD Software: `http://usqcd.jlab.org/`.

[7] A. Bazavov, C. Bernard, C. DeTar, S. Gottlieb, U.M. Heller, J.E. Hetrick, J. Laiho, L. Levkova, P.B. Mackenzie, M.B. Oktay, R. Sugar, D. Toussaint, and R.S. Van de Water. Nonperturbative QCD simulations with 2+1 flavors of improved staggered quarks. *Reviews of Modern Physics*, 82:1349–1417, 2010.

[8] Carleton E. DeTar. Sharing lattices throughout the world: An ILDG status report. *PoS*, LAT2007:009, 2007.

[9] A. Torok, S. Basak, A. Bazavov, C. Bernard, C. DeTar, E. Freeland, W. Freeman, S. Gottlieb, U.M. Heller, J.E. Hetrick, V. Kindratenko, J. Laiho, L. Levkova, M. Oktay, J. Osborn, G. Shi, R.L. Sugar, D. Toussaint, and R.S. Van de Water. Electromagnetic splitting of charged and neutral mesons. volume LATTICE2010, page 127, 2010.

[10] Steven Gottlieb, Guochun Shi, Aaron Torok, and Volodymyr Kindratenko. QUDA programming for staggered quarks. *PoS*, LATTICE2010:026, 2010.

[11] G. Shi, S. Gottlieb, A. Torok, and V. Kindratenko. Accelerating Quantum Chromodynamics Calculations with GPUs. In *Proc. Symposium on Application Accelerators in HPC (SAAHPC10)*, Knoxwille, TN, July 2010.

[12] G. Shi, S. Gottlieb, A. Torok, and V. Kindratenko. Design of MILC lattice QCD application for GPU clusters. In *Proceedings of the 2011 IEEE Parallel and Distributed Processing Symposium*, IPDPS '11. IEEE, 2011.

[13] R. Babich, M. A. Clark, B. Joó, G. Shi, R. C. Brower, and S. Gottlieb. Scaling Lattice QCD beyond 100 GPUs. In *Proceedings of the 2011*

ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11. IEEE Computer Society, 2011.

[14] A. Torok, S. Basak, A. Bazavov, C. Bernard, C. DeTar, et al. Electromagnetic splitting of charged and neutral mesons. *PoS*, LATTICE2010:127, 2010.

[15] Heechang Na and Steven Gottlieb. Heavy baryon mass spectrum from lattice QCD with 2+1 dynamical sea quark flavors. *PoS*, LATTICE2008:119, 2008.

[16] Heechang Na and Steven A. Gottlieb. Charm and bottom heavy baryon mass spectrum from lattice QCD with 2+1 flavors. *PoS*, LAT2007:124, 2007.

[17] `http://www.physics.utah.edu/~detar/milc/milc_qcd.html`.

[18] Richard Todd Evans, Elvira Gamiz, Aida El-Khadra, and Andreas Kronfeld. B-Bbar Mixing and Matching with Fermilab Heavy Quarks. *PoS*, LAT2009:245, 2009.

[19] C. Bouchard, A.X. El-Khadra, E.D. Freeland, E. Gamiz, and A.S. Kronfeld. B_s^0 and B^0 Mixing in the Standard Model and Beyond: A Progress Report. *PoS*, LATTICE2010:299, 2010.

[20] Jon A. Bailey et al. $B \to D^* \ell nu$ at zero recoil: an update. *PoS*, LATTICE2010:311, 2010.

[21] Jon A. Bailey, A. Bazavov, C. Bernard, C.M. Bouchard, C. DeTar, et al. $B_s \to D_s / B \to D$ Semileptonic Form-Factor Ratios and Their Application to BR($B_s^0 \to \mu^+ \mu^-$). 2012.

[22] C. Bernard et al. B Mixing in the Standard Model and Beyond: Lattice QCD. 2011.

[23] C.M. Bouchard, E.D. Freeland, C. Bernard, A.X. El-Khadra, E. Gamiz, et al. Neutral B mixing from $2 + 1$ flavor lattice-QCD: the Standard Model and beyond. 2011. 13 pages, 6 figures. Proceedings of the XXIX International Symposium on Lattice Field Theory - Lattice 2011, July 10-16, 2011, Squaw Valley, Lake Tahoe, California. Ver 2: New Figs. 2-4 and Table 4 corrects scripting error. Table 4 now includes results for BJU as well as BBGLN choice of evanescent operators.

[24] Si-Wei Qiu et al. Semileptonic B to D Decays at Nonzero Recoil with 2+1 Flavors of Improved Staggered Quarks. *PoS Lattice2011*, 2011.

[25] Daping Du, Carleton DeTar, Andreas Kronfeld, Jack Laiho, Yannick Meurice, et al. Semileptonic Form Factor ratio $B_s \to D_s/B \to D$ and Its Application to $BR(B_s^0 \to \mu^+\mu^-)$. 2011. 7 pages, 6 figures. Talk presented at The XXIX International Symposium on Lattice Field Theory - Lattice 2011, July 10-16, 2011, Squaw Valley, Lake Tahoe, California/ Minor errors corrected, references and graphs updated.

[26] E. Gamiz, C. DeTar, A.X. El-Khadra, A.S. Kronfeld, P.B. Mackenzie, et al. Calculation of $K \to \pi l \nu$ Form Factors with $N_f = 2 + 1$ Flavors of Staggered Auarks. 2011. 7 pages, 4 figures, presented at The XXIX International Symposium on Lattice Field Theory - Lattice 2011, July 10-16, 2011, Squaw Valley, Lake Tahoe, California.

[27] C. DeTar, A.S. Kronfeld, and M.B. Oktay. Numerical Tests of the Improved Fermilab Action. *PoS*, LATTICE2010:234, 2010.

National Computational Infrastructure for Lattice Gauge Theory
SciDAC-2 Closeout Report for MIT

Lead Institution: Fermilab
Batavia IL 60510
This Institution: Massachusetts Institute of Technology
MIT Principal Investigator: John Negele
6-315 MIT
77 Massachusetts Ave.
Cambridge, MA 02139
negele@mit.edu
Office of Science Program Contact: Randall Laviolette

This report summarizes the SciDAC-2 research accomplishments of Principal Research Scientist Andrew Pochinsky.

1 Möbius Domain Wall Fermions

The Möbius generalization[1] of the standard domain wall algorithm includes the standard Shamir domain wall implementation as a special case, but also offers the advantage of algorithmic speed-up when using the scaling parameter of the Möbius generalization. The Möbius DWF inverter, MDWF, has been implemented as a Level III library, reusing the infrastructure developed earlier for the regular Domain Wall fermion inverter. Like all software that has been developed at MIT under SciDAC, source code can be downloaded from the MIT SciDAC website[2]. The inverter is now fully integrated both into Chroma and into QDP/C, has been tuned for performance and scaling on the ANL BG/P, where it has been heavily exploited by ESP and INCITE projects, and is used on BG/Ls at BU and MIT. It has been used extensively in calculating the structure of the nucleon and other low mass baryons[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], as well as by the LSD Collaboration[16] to calculate the S-parameter[17] that is important in exploring physics beyond the Standard Model.

The code scales linearly to the full ANL BG/P, sustaining 23-24% in single precision and 18% in double precision. This linear speedup is made possible by a combination of using `qa0`[18], an optimization tool that is a precursor to Qlua, to aggressively reuse data in local computations and using Morton ordering to provide cache oblivious memory access.

MDWF is also used to accelerate the standard DWF inversion. With appropriate tuning, the Möbius action provides a good approximation to the Shamir action using a smaller number of steps in the fifth dimension. Thus, it provides an economical preconditioner for the conjugate gradient solver, which in practice, provides speedup by a factor of two for current USQCD domain wall lattices.

2 Clover Wilson Fermions

The Budapest-Marseille-Wuppertal (BMW) collaboration utilized an improved Clover Wilson action to calculate the spectrum of light hadrons down to the physical pion mass[19], which was recognized by *Science* as one of the top ten achievements of the year. Motivated by a collaboration to utilize their configurations for hadron structure calculations, a Level III library of solvers for Clover Wilson fermions was implemented. It built upon the existing inverter infrastructure, including `qa0` and cache-oblivious Morton layout, to develop an efficient mixed-precision conjugate gradient solver. In addition, the EigCG inversion acceleration algorithm[20] was implemented, which computes and saves low-lying eigenvectors of the Dirac operator during initial inversions on a given configuration, and then reuses them to accelerate the convergence of subsequent inversions on the same lattice. The resulting speed-up of the inverter by a factor of four to six has been crucial in facilitating an extensive set of hadron structure calculations down to the physical pion mass[21].

The library is available under an open source license on the MIT SciDAC web site[2].

3 Qlua

Lua is a modern high-level embeddable scripting language. Pochinsky has used it as a starting point for developing Qlua, a LQCD-centric domain specific language. Qlua[**?**] has become a highly flexible LQCD application framework that serves as a platform for algorithm development and code integration and is heavily used in production.

Extending basic Lua with parallel data types and operations makes Qlua a versatile tool for lattice QCD. It provides a user with full access to the QDP library while handling all house-keeping operations behind the scene. By relying on automatic memory management provided by the Lua runtime system, the memory footprint of production codes is considerably decreased. Using the full-blown scripting language for application level code allows users to quickly adapt to constraints of the queueing and file systems at ANL's BG/P and increases their ability to harvest otherwise unutilised cycles on the machine, thus providing faster turnaround time for production runs and avoiding waste of an expensive and valuable national resource. Both the MDWF and Clover inverters are fully integrated into Qlua and are used in production both on the national leadership class and USQCD resources.

Three primary uses of Qlua are the following:

1. Algorithm development platform
Qlua serves as an algorithm development platform that liberates the programmer from tedious book-keeping and thus optimizes the use of human time. To facilitate faster and more general inverters, Qlua and QDP have been extended to handle multiple lattices, which is one feature that is required for mutligrid algorithms that will be essential for calculations with large lattices at the physical pion mass. Parallel operations have also been extended beyond $N_c = 3$ colors, which is the second generalization required for multigrid algorithms and which is also required for applications studying physics beyond the Standard Model .

2. Code testbed
In development of highly optimized libraries, code testing and verification play an important role in the development cycle. As the complexity of algorithms grows, correctness tests must handle more corner cases and explore a larger parameter space. Qlua helps the development process by (a) providing an easy-to-use platform for reference implementation of the algorithm via the QDP primitives that can be directly compared to the optimized implementation and (b) a simple way to write large number of test cases to verify the implementation.

3. Production scripting
Use of Qlua as the application level script in production runs greatly improves the ability to respond to the changes in computing environments. Deployment on all machines from leadership-class to laptops makes moving from concept to production a simple exercise in changing problem size from a toy model to the real application.

Three years of experience using Qlua for production by the LHP Collaboration have shown Qlua to be an effective domain specific language for use in SciDAC. Building on the fact that QDP hides the gory details of a massively parallel computer behind a simple data parallel abstraction, Qlua is the logical extension of the idea by combining QDP with a modern programming language. Experience has shown that a high-level language at the application level improves users productivity and that automatic memory management is a must for scientific HPC. It has also shown that some data parallel and domain specific features are better expressed if the system design is integrated with these features from the beginning instead of adding them to an existing language.

References

[1] R. C. Brower, H. Neff and K. Orginos, "Mobius fermions: Improved domain wall chiral fermions," Nucl. Phys. Proc. Suppl. **140**, 686 (2005) [hep-lat/0409118].

[2] All source code, including qlua, mdwf, and clover are available at: `https://lattice.lns.mit.edu/trac/downloads`

[3] B. U. Musch, P. .Hagler, M. Engelhardt, J. W. Negele and A. Schafer, "Sivers and Boer-Mulders observables from lattice QCD," Phys. Rev. D, in press. arXiv:1111.4249 [hep-lat].

[4] S. N. Syritsyn, J. R. Green, J. W. Negele, A. V. Pochinsky, M. Engelhardt, P. .Hagler, B. Musch and W. Schroers, "Quark Contributions to Nucleon Momentum and Spin from Domain Wall fermion calculations," arXiv:1111.0718 [hep-lat].

[5] C. Alexandrou, E. B. Gregory, T. Korzec, G. Koutsou, J. W. Negele, T. Sato and A. Tsapalis, "The $\Delta(1232)$ axial charge and form factors from lattice QCD," Phys. Rev. Lett. **107**, 141601 (2011) [arXiv:1106.6000 [hep-lat]].

[6] C. Alexandrou, G. Koutsou, J. W. Negele, Y. Proestos and A. Tsapalis, "Nucleon to Delta transition form factors with $N_F = 2 + 1$ domain wall fermions," Phys. Rev. D **83**, 014501 (2011) [arXiv:1011.3233 [hep-lat]].

[7] C. Alexandrou, T. Korzec, G. Koutsou, J. W. Negele and Y. Proestos, "The Electromagnetic form factors of the Ω^- in lattice QCD," Phys. Rev. D **82**, 034504 (2010)

[8] J. D. Bratt *et al*. [LHPC Collaboration], "Nucleon structure from mixed action calculations using 2+1 flavors of asqtad sea and domain wall valence fermions," Phys. Rev. D **82**, 094502 (2010) [arXiv:1001.3620 [hep-lat]].

[9] C. Alexandrou, G. Koutsou, T. .Leontiou, J. W. Negele and A. Tsapalis, "Axial Nucleon to Delta transition form factors on 2+1 flavor hybrid lattices," Phys. Rev. D **80**, 099901 (2009) [arXiv:0912.0394 [hep-lat]].

[10] S. N. Syritsyn, J. D. Bratt, M. F. Lin, H. B. Meyer, J. W. Negele, A. V. Pochinsky, M. Procura and M. Engelhardt *et al*., Phys. Rev. D **81**, 034507 (2010) [arXiv:0907.4194 [hep-lat]].

[11] E. E. Jenkins, A. V. Manohar, J. W. Negele and A. Walker-Loud, Phys. Rev. D **81**, 014502 (2010) [arXiv:0907.0529 [hep-lat]].

[12] C. Alexandrou, T. Korzec, G. Koutsou, C. Lorce, J. W. Negele, V. Pascalutsa, A. Tsapalis and M. Vanderhaeghen, Nucl. Phys. A **825**, 115 (2009) [arXiv:0901.3457 [hep-ph]].

[13] C. Alexandrou, T. Korzec, G. Koutsou, T. .Leontiou, C. Lorce, J. W. Negele, V. Pascalutsa and A. Tsapalis *et al*., Phys. Rev. D **79**, 014507 (2009) [arXiv:0810.3976 [hep-lat]].

[14] A. Walker-Loud, H. -W. Lin, D. G. Richards, R. G. Edwards, M. Engelhardt, G. T. Fleming, P. .Hagler and B. Musch *et al*., Phys. Rev. D **79**, 054502 (2009) [arXiv:0806.4549 [hep-lat]].

[15] C. Alexandrou, G. Koutsou, H. Neff, J. W. Negele, W. Schroers and A. Tsapalis, Phys. Rev. D **77**, 085012 (2008) [arXiv:0710.4621 [hep-lat]].

[16] Qlua documentation, including building instructions and tutorials, are available at: `http://www.yale.edu/LSD/members.html`

[17] T. Appelquist, R. Babich, R. C. Brower, M. Cheng, M. Clark, S. Cohen, G. Fleming, J. Kiskis, E. Neil, J. Osborn, C. Rebbi, D. Schaich and P. Vranas, "Parity Doubling and the S Parameter Below the Conformal Window," Phys. Rev. Lett. **106**, 231601 (2011)

[18] A. Pochinsky, PoS LATTICE **2008**, 040 (2008).

[19] S. Durr, Z. Fodor, J. Frison, C. Hoelbling, R. Hoffmann, S. D. Katz, S. Krieg and T. Kurth *et al*., Science **322**, 1224 (2008) [arXiv:0906.3599 [hep-lat]].

[20] A. Stathopoulos, A. M. Abdel-Rehim and K. Orginos, J. Phys. Conf. Ser. **180**, 012073 (2009).

[21] J. Green, S. Krieg, J. Negele, A. Pochinsky and S. Syritsyn, PoS LATTICE **2011**, 157 (2011) [arXiv:1111.0255 [hep-lat]].

National Computational Infrastructure for Lattice Gauge Theory
SciDAC-2 Institutional Closeout Report

A proposal in response to Office of Science Notice DE-FG02-06ER06-04 and Announcement Lab 06-04: Scientific Discovery through Advanced Computing.

Principal Investigator: Chip Watson
Address: Thomas Jefferson National Accelerator Facility
Newport News, VA 23606.
Email: watson@jlab.org
Phone: (757) 269-7101

Lead Principal Investigator: Paul Mackenzie
Lead Institution: Fermi National Accelerator Laboratory (FNAL), Batavia, IL 60510.

Office of Science Programs Addressed: High Energy Physics and Nuclear Physics

Office of Science Program Office Technical Contacts: Ted Barnes and Lali Chatterjee

Participating Institutions and Principal Investigators:

Physics:

Boston University*, Richard Brower † ‡ and Claudio Rebbi †
Brookhaven National Laboratory*, Michael Creutz † ‡
Columbia University*, Norman Christ † ‡
Fermi National Accelerator Laboratory*, Paul Mackenzie † ‡
Indiana University*, Steven Gottlieb ‡
Massachusetts Institute of Technology*, John Negele † ‡
Thomas Jefferson National Accelerator Facility*, David Richards † and William (Chip) Watson ‡
University of Arizona*, Doug Toussaint ‡
University of California, Santa Barbara*, Robert Sugar † ‡
University of Utah*, Carleton DeTar ‡
University of Washington, Stephen Sharpe †

Computer Science:

DePaul University*, Massimo DiPierro ‡
Illinois Institute of Technology*, Xian-He Sun ‡
University of North Carolina*, Rob Fowler ‡
Vanderbilt University*, Theodore Bapty ‡

* Institution submitting an application
† Project Principal Investigator, Member of the USQCD Executive Committee
‡ Institution Principal Investigator

Work at Jefferson Laboratory (JLab) followed five main tracks: 1) Development of the Chroma software system on leadership platforms for gauge generation, ii) Developments in the QCD SciDAC API for data analysis driven by the need to meet JLab science mission needs, iii) exploitation of Graphical Processing Unit (GPU) technologies (driven by the American Recovery and Reinvestment Act - ARRA - procurement) iv) co maintenance and support of the SciDAC software infrastructure, in particular the Chroma Software System and its related modules such as QDP++ and v) algorithmic advances enabling both advances in our ability to generate gauge configurations and to make physics measurements on those configurations. We summarize the progress made over the project in each of these areas below.

Core to the lattice calculation of the spectrum of QCD has been the use of anisotropic clover lattices, with finer temporal than spatial lattice spacing. The JLab group developed optimized code for this action within the Chroma software suite, focusing in particular on the Cray XT leadership-class machines. As the lattice volumes increased - they are now as large as $48^3 \times 512$ - it was realized that relying on single precision arithmetic was no longer feasible, and that double precision was required to maintain the required reversibility in the molecular dynamics time stepping and force calculations resulting in roughly a factor of 2 slowdown in code performance from the single precision case. To ameliorate this we followed the success of the multi-precision techniques demonstrated by colleagues at Boston University in GPUs. These techniques (multiple precision through reliable updates) were ported into the Chroma software system. Additionally these were combined with ideas from reduced communication Krylov solvers such as the Improved Stabilized Bi-Conjugate Gradients (Improved BiCGStab or IBiCGStab), resulting in solvers which produce fully double precision results using predominantly single precision with the minimal number of global reduction synchronization points (Reliable Improved BiCGStab or Rel-IBiCGStab). In addition we performed parameter tuning of our molecular dynamics time stepping procedure, and integrated the QMT threading API pervasively through the performance intensive aspects of both the QDP++ and Chroma software packages.

A major effort by JLab has been on the use of databases, and the software produced by that effort was integrated into QDP++ and Chroma and is in production use. The use of these databases has been essential to exploit some of the algorithmic advances made under SciDAC-2, and in particular "distillation", described below.

The rise of GPU computing presented an unprecedented opportunity for highly cost effective capacity computing, as much as an order of magnitude more cost effective than existing alternatives, and this was enthusiastically embraced by the JLab group to exploit the GPU-accelerated cluster procured under the American Recovery and Reinsvestment Act, and subsequent installations both at FNAL and at JLab. In particular, we collaborated with the developers of the QUDA library and Boston University to incorporate several crucial extensions, most notably by integrating the Wilson-clover inverters from QUDA into the Chroma framework, and by parallelisation of the QUDA library onto multiple GPUs. These efforts are enabling us to analyse the full range of lattice sizes needed for our work, notably for the anisotropic clover program.

GPUs are now appearing in accelerated-architecture leadership-class computers, such as the Cray XK series; a member of the JLab team, Balint Joo, together with Mike Clark of Nvidia gained early access to the Jaguar/TitanDev 10-rack, 960-GPU system at OLCF, demonstrating scaling of the inverter to 768 GPUs. Finally, the JLab group partnered with Intel on evaluation of, and development of optimized code for, the Intel MIC (many-integrated-core) architecture.

The Chroma application suite is the principle framework for studies of the spectroscopy of hadrons and of their interaction, including the generation on leadership facilities of Wilcon-clover gauge configurations. It is widely adopted, with over 200 citations. Throughout the SciDAC-2 program, we devoted some fraction of our time to code maintenance and infrastructure tasks, as well as incorporating the advances outlined in this report.

In the area of algorithms and applications, Jefferson Lab has been at the forefront of developing a novel technique, called *distillation*, for analyzing the spectrum of excited states. The technique allows for unprecedented precision in studying excited states, and has been used in the most precise calculations of the spectrum of hybrid meson excited states made to date. The technique, however is rather expensive in terms of computation. The multi-GPU parallelization of the QUDA library, combined with its integration into the Chroma application framework, exploits the power of GPUs to enable precision calculations with the distillation technique. Other algorithmic advances under SciDAC-2 include the development of a new temporal preconditioner, in collaboration with Trinity College, Dublin, and the development of a deflated BiCGStab solver, in collaboration with the College of William and Mary.

The advances enabled by the SciDAC-2 project have greatly expanded the range of calculations relevant to the DOE nuclear physics program. An outstanding example is the calculation of the excited state spectrum of mesons and baryons with the quantum numbers of the states reliably identified, and with the important prediction of states of exotic quantum numbers in an energy range accessible to the future GlueX experiment at the 12 GeV Upgrade of Jefferson Laboratory.

UCSB SciDAC-2 Closeout Report

During the first two and a quarter years of this grant, Robert Sugar, the UCSB Principal Investigator, served as the lead principal investigator and Spokesperson for the project as a whole. As Chair of the USQCD Executive Committee he provided overall leadership and co-ordination of the effort. On January 1, 2009, Sugar stepped down as Chair of the Executive Committee and lead principal investigator of this grant, being replaced by Paul Mackenzie in both these capacities. However, he remained a member of the Executive Committee and a co-principal of this project, in which positions he continued to help provide leadership for the project.

Throughout the course of the grant, UCSB administered funds for travel not covered by grants to other participating institutions. These trips included visits of collaboration members to participating institutions for joint research work, and attendance at meetings directly related to the project, such as the series of International Lattice Field Theory Network workshops and those at which results from the project could presented. UCSB also administered travel funds for Sugar and for Principal Investigator Stephen Sharpe to cover trips associated with the project. During the last two years, UCSB also administered travel funds for Julius Kuti after he became a member of the USQCD Executive Committee with responsibilities for the project. In the original proposal, we estimated that twenty-one trips per year would be supported for a total of 126 over the six year period of the grant. To date, 108 trips have been supported in five and a half years. The remainder of the $165,000 allocated to UCSB is committed, and is expected to be expended by the end of the grant.

DOE Final Report
Project: DE-FC02-06ER41441

Massimo Di Pierro

April 6, 2012

DOE Award Number: DE-FC02-06ER41441
Name of Recipient: Massimo Di Pierro
Principal Investigator: Massimo Di Pierro
Consortium Name: USQCD Collaboration

1 License

2 Executive Summary

Our research project is about the development of visualization tools for Lattice QCD. We developed various tools by extending existing libraries, adding new algorithms, exposing new APIs, and creating web interfaces (including the new NERSC gauge connection web site). Our tools cover the full stack of operations from automating download of data, to generating VTK files (topological charge, plaquette, Polyakov lines, quark and meson propagtors, currents), to turning the VTK files into images, movies, and web pages. Some of the tools have their own web interfaces. Some Lattice QCD visualization have been created in the past but, to our knowledge, our tools are the only ones of their kind since they are general purpose, customizable, and relatively easy to use. We believe they will be valuable to physicists working in the field. They can be used to better teach Lattice QCD concepts to new

graduate students; they can be used to observe the changes in topological charge density and detect possible sources of bias in computations; they can be used to observe the convergence of the algorithms at a local level and determine possible problems; they can be used to probe heavy-light mesons with currents and determine their spatial distribution; they can be used to detect corrupted gauge configurations. There are some indirect results of this grant that will benefit a broader audience than Lattice QCD physicists and explained below.

3 Achievements and Comparison of Goals

The goal of this research project is the creation of a visualization toolkit that could be used to aid physicists in the analysis of data from lattice QCD computations. This goal has been successfully achieved as descibed below.

Links to code, images and video can be found at:

`http://latticeqcd.org`

The original grant proposal stated:

> Lattice QCD computations comprise multiple steps, creating very large datasets, but the final result is typically encompassed in a small set of numbers with the analysis performed in an automated way. While an automated procedure may be beneficial in efficiency, the ability to visualize the data being analyzed is important both as an aid to the analysis, and as a means of acquiring insight into the physics. [...]
>
> Crucial to the success of the graphics-visualization initiative will be a close collaboration between physicists to devise and interpret visualization of physically important quantities, and computer scientists to provide the appropriate visualization toolbox. Questions that visualization might address are many: can we understand how flux-tube formation observed with infinitely heavy quarks extends to hadrons where one or more of the quarks is light; what is the distribution of charge within a nucleon; can we display the distribution of spin and magnetism within a hadron? In the longer term, can we visualize the interactions of hadrons? Currently, no general-purpose package is available tailored to the display of lattice data. Thus a software package will be developed with a general GUI capable of reading a set of four-dimensional lattice quantities, and taking their ensemble average; performing a projection into a real four-dimensional vector; interpolating the 4-D vector into a continuous four-dimensional field; taking

three-dimensional slices of a four-dimensional field; displaying the data using density plots, iso-surfaces, and 2-D projections; and displaying the evolution of data, both in simulation time for four-dimensional quantities, and as the evolution of three- and two-dimensional slices in the remaining coordinates. The software will support two types of plug-ins: type-1 plug-ins that perform specific physics measurements and output a real 4-D vector, and type-2 plug-ins that take the interpolated 3-D field and generate specific types of plots.

Most of the research underlying this project will consist of identifying a set of physical measurements suitable to be implemented as type-1 plug-ins. The visualization techniques for the type-2 plug-ins are very similar to standard techniques used for representation of 3-D geophysical data and, when possible, we will incorporate existing libraries into the development of our plug-ins.

The system will be developed in C++ and take advantage of existing graphics and visualization libraries such the Trolltech QT libraries and the Visualization Tool Kit (VTK) library. The plug-ins will be callable from C or C++ code conforming to the QCD API, and will form another component of our Level 4 QCD Toolbox. The system will be capable of reading datasets in the SciDAC/ILDG format and the MILC format.

[...]

DePaul University will lead the design and development of a visualization tool for lattice QCD. Work will be done in collaboration with physicists involved in the project and with computer scientists at the University of North Carolina. The goals for the first year of the project are to identify and catalog the types of datasets to be visualized, identify appropriate smoothing and visualization algorithms, and develop a prototype interface. In subsequent years, plugins will be developed to read in the various types of datasets produced in lattice QCD simulations, and tools for manipulating the data in increasingly sophisticated ways will be created. A total of 1.08 FTE per year is budgeted for this effort.

A QCD physics toolbox will be constructed which will contain sharable software building blocks for inclusion in application codes, performance analysis and visualization tools, and software for automation of physics work flow. New software tools will be created for managing the large data sets generated in lattice QCD simulations, and for sharing them through the International Lattice Data Grid consortium.

Our basic toolkit consists of two parts. The first part has been implemented in the form of C++ libraries which are now included within the FermiQCD toolkit which is part of the USQCD Software Suite. These API allow the project of arbitrary fields into 3D and 4D scalar

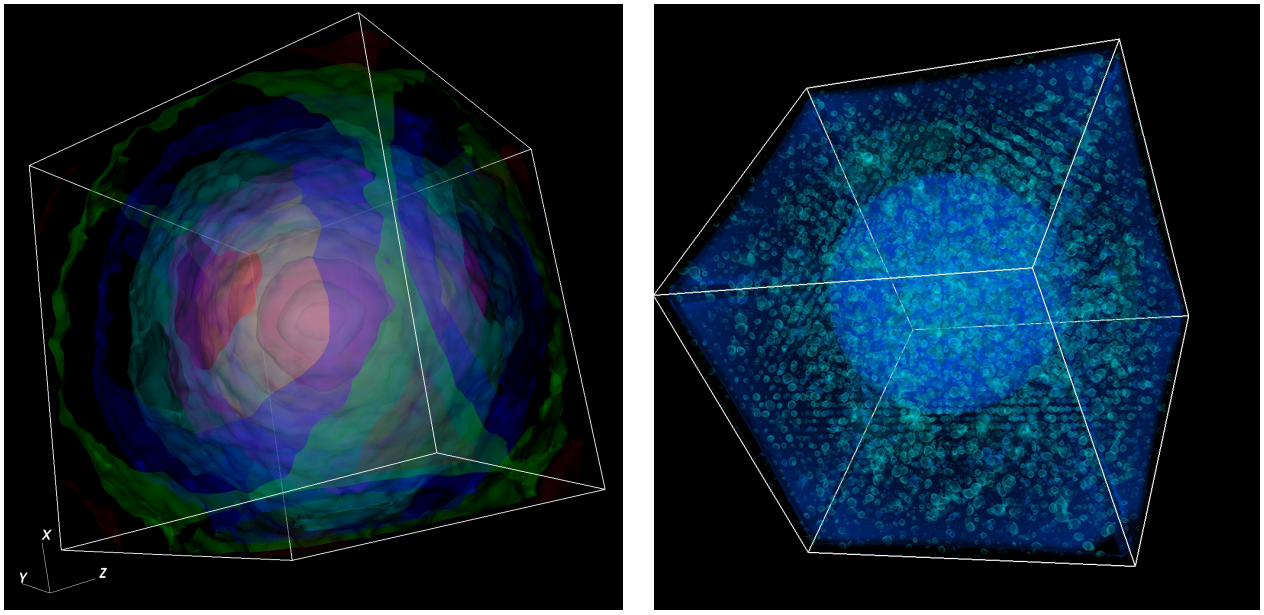

Figure 1: Example Images. The left one shows iso-surfaces for a quark wave function. The right one shows the density of HMC hits in presence of a lump of topological charge.

fields with can be visualized using open source toolkits like VisIt, Paraview, and MayaVi. The toolkit includes algorithms to project topological charge density, plaquette, Polyakov line components, quark propgators, meson propagators, and current insertions. It provides an API to create other custom operators and quark contractions and project/visualize them too. We also modified the minimum residue and the stabilized bi-conjugate inverter so that it is possible to observe the spatial effect of this algorithms and visualize their convergence for arbitrary sources. Our choice of the FermiQCD toolkit is motivated by the critical need to be able to perform visualizations for arbitrary $SU(N)$ gauge groups. FermiQCD is the only lattice QCD code, part of the USQCD Software Suite, that at this time supports abitrary gauge groups.

The second part of our toolkit consists of a collection of Python programs that interface with the C++ programs and make the system more accessible to scientists by implementing a typical workflow. Specifically we developed 7 different programs.

The first program (as required by the original grant proposal) has the ability to convert MILC/ILDG data (as well other data formats) into the format required for visualization. On top of that, as described below, the same program has the ability to download gauge configurations from the NERSC repository, which is the largest public repository of gauge

configurations in the United States.

The second program is the main interface to the C++ algorithms. The program provides command line options to run physical algorithms such as the computation of the topological change density, plaquette, Polyakov, lines, quark propagators, meson propagators, currents, 4-quark operators. The program downloads, compiles and runs the requested algorithms. Each algorithm provides the option to save the computation steps in the VTK format for visualization. Not all the C++ algorithm are accessible via this interface and some requires explicit programming. Yet the provided code and documentation should be a sufficient example for the scientists to write their own customized code for other particular cases.

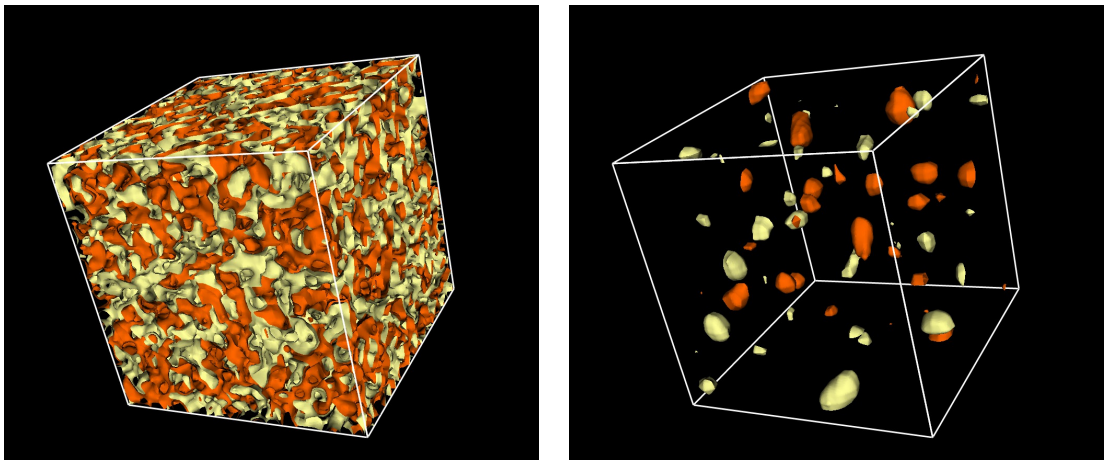

Figure 2: Example Images: They show the visualization of the topological charge at different cooling steps.

The thrid program perform the tasks of extracting information from the VTK files generated by the previous step, resampling them (to achieve better resolution in visualization), interpolating them (to make smooth visualizations and movies) and generating VisIt scripts. The VisIt visualization toolkit was developer by the Lawrence Livermore National Laboratory and it is a critical part of our workflow. It can be accessed via a GUI or programmatically via Python. Using a GUI to automate a workflow and process many files at once is not practical. It must be done programmatically. Our program generates VisIt scritps using meta-programming techniques so that QCD phycisicts do not have to code. By running these scripts scientists can directly obtain images and movies without programming.

The fourth program we develop allows to convert 3D VTK files into interactive web pages. The program opens the VTK files, identifies optimal tresholds for the iso-surfaces, computes said iso-surfaces and generates a 3D representation of the polygons in JavaScript. The output

consists of static HTML files which embed the 3D objects. They can be visualized with any browser and the viewer can rotate them with the mouse. While VisIt is a general purpose tool is very powerful which allows many more maniputaions of the data, the possibility of generating 3D interactive web pages opens the possibility for scientists to view the data without installing VisIt and to publish the data on the web for other people to see.

The other programs we created are beyond the scope of the original grant proposal but we felt they were necessary and part of a broader interpretation of the conept of visualization. In fact, not all visualizations are spatial visualizations. There is other information that is important to visualize, which often takes the forms of simple 2D plots but often is not looked at because of the extra work involved in doing so. Examples are autocorrelations between physical measurements on different gauge configurations, moving averages, distributions of bootstrap samples. Our other programs serve this purpose. One of the programs read the output of typical QCD algorithms, extract the numerical results for each gauge configuration, and combines them into a bootstrap analysis. The user can specify the expression to bootstrap using command line arguments without programming. The code generates CSV files storing data for the intermediate steps of the computation. The other two programs can read those CSV files and generate plots and fits from them automatically. CSV files can also be read by many third party analysis and visualization programs.

Consider for example the computation of a four-quark matrix element. It involves the computation of a two-point and a three-point correlation funciton for each gauge configuration and their bootstrap analysis. Our programs can perform this analysis in the standard way but they also generate autocorrelation plots for each two-point and three-point correlation function, moving averages for the ratio on different time-slices, and bin the distribution of the bootstrap samples.

Those programs have been documented in long technical document attached to this report [1].

The main obstacle to this research has been accessibility. Visualization is indeed useful not but the way QCD physicists normally approach lattice QCD computations. We have therefore put lots of extra work in making our programs accessible by creating web interfaces that could simplify the task. Another obstacle is that no computing time was allocated to this project. While this did not prevent us from achieving the task of developeing the tools, it did not allow us to move beyond the original stated goal and utilize the tools for obtaining more abituious scientific results which would have been computing intensive.

Nevertheless we have used our toolkit to produce scientific results. Specifically we collaborated with Chulwoo Jung at Columbia University, Mike Clark and Richard Brower at Boston Universty and looked at the autocorrelation of topological charge density over short HMC trajectories [15]. It is a well known problem that global topological charge has a long autocorrelation. We found that the local topological charge instead has very short autocorrelation

and therefore there is no measurable bias in production gauge configurations.

We also were able to observe the effect of a single instanton on a quark propagator and how its presence gives mass to the quark by increasing the exponential fall off of the propagator [1].

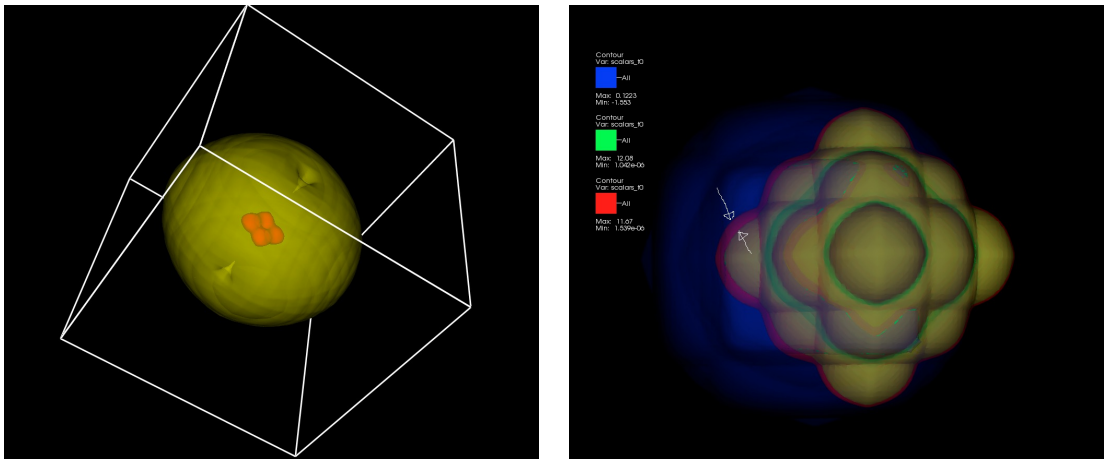

Figure 3: Example Images: left left image shows the density of a current inserted between to meson operators. The right image shows the shriking of a quark propagator (red) in presence of a localized topological charge (blue).

Anyway, that is beyond the scope of the current grant and more visualizations will be done in the future as computing time becomes available.

In our original proposal we stated that our tools would have a GUI based on the Qt toolkit. In the very early stages of our project we have revised that decision. On the one side Nokia, owner of Qt, decided to cut support for the library. On the other site it became evident that desktop GUI have became an obsolete technology giving way to modern web based interfaces. We have therefore put lots of extra work in this direction and we have created three web applications.

The first web application (nersc) [3] was developed in collaboration with the National Energy Research Scientic Center (NERSC) to replace their previous interface to the lattice QCD archive known as "gauge connection". The new system allows searching for gauge configuration, visualize statistics, and collaborate online by editing metadata in a wiki format. The system also allows batch downloads of the data using the program we described above. We have used our tookit to process many ensembles from the NERSC archive and generated movies of the topological charge density. This program is designed to be very general purpose and it can be used by communities other than Lattice QCD to publish their data online.

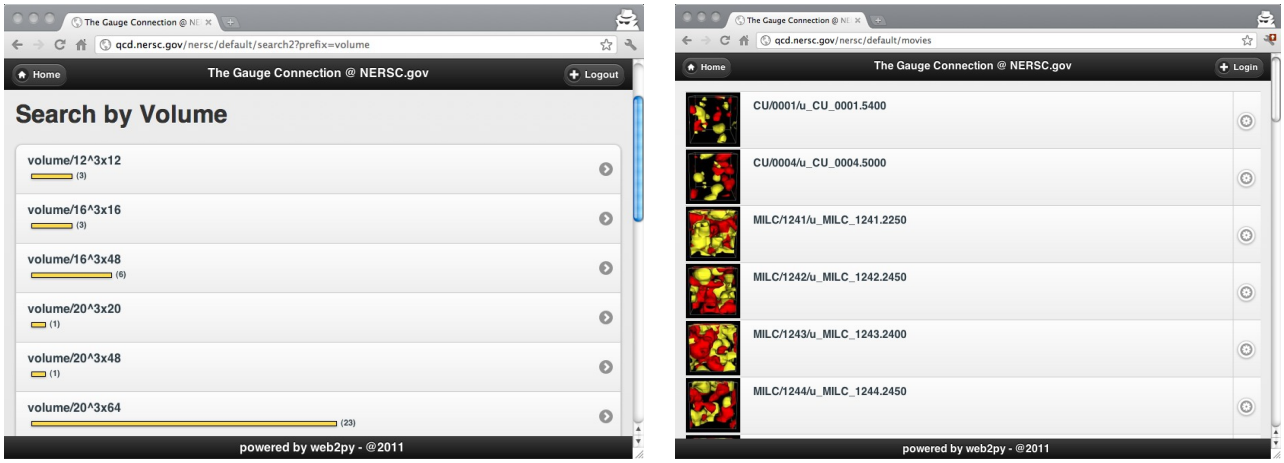

Figure 4: Screenshots from the NERSC "gauge connection" web site.

The second web application (vis) [6, 8] provides an interface to the algorithms and allows upload of gauge configurations and schedule visualization algorithms to run in backgroud. The system provides a web interface to the Portable Batch System (PBS) and to VisIt and. It can schedule both computations and visualizations. The results are displayed in a web page.

The third web application (mc4qcd) [7, 9] is an interface to our analysis and plotting tools. It allows users to upload the log file result of a physics runs, to extract data from it using pattern matching, and to perform bootstrap analysis. The reults of the analysis are stored and published online together with the replative plots (including autocorrelations, moving averages, distribution of bootstrap samples, and fits). Scientists in a group can track results and collaborate online by sharing data and comments.

In order to develop these tools, in the eraly stages of the project we have developed a set of libraries for creating online scientific applications called web2py. This project is not part of the goal of this grant but it turned out to be an important and necessary component to replace the obsolete GUI concept with the modern web based paradigm. This project took a life on its own and found applications beyound this physics project. It was released open source and it is now used by thousands of users and businesses worldwide. It won the Bossie Award for "best open source software development tool" in 2011 and the InfoWorld Best Technology of the Year Award in 2012. This provides an example of unexpected broader impact of DOE funded research. Although we consider this very important and we are proud of the result, since it falls outside the scope of the original grant, we omitted references to it in the rest of this report. Yet references are available upon request.

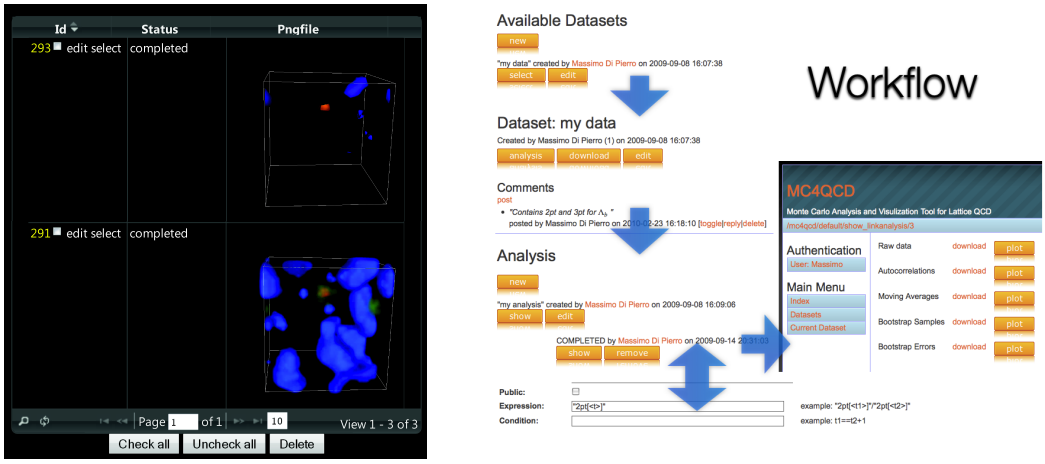

Figure 5: Screenshots from the VIS and the MC4QCD web applications respectively.

In 2010 we also contributed organize the 6th High End Visualization workshop in Obergugl, Austria.

Some of the visualization created with our tools were used for the opening video for the Lattice 2011 conference: `http://vimeo.com/25242353`

3.1 Summary of Code Created

The code written as part of this grant is published in the following repositories:

- `http://code.google.com/p/qcdutils/` It is the main toolkit, documented in [1].
- `http://code.google.com/p/fermiqcd/` This is a pre-existing C++ library for lattice QCD computations part of the USQCD Software Suite. The core visualization algorithms have been included here and distributed together. They are accessible via QCDUTILS.
- `http://code.google.com/p/nersc-data-publisher/` This is code behind the new NERSC "gauge connection" web interface. Develped in collaboration with James Hetrick (University of Pacific) and David Skinner (NERSC).
- `http://code.google.com/p/qcdvis/` This is a web interface to the visualization algorithms.

- `http://https://launchpad.net/qcdmc` This is a web interface to our analsys and plotting tools.

3.2 Published Web Sites

- `http://latticeqcd.org` This is the main front-end where we have published links to the code and some of our visualizations. More will be published as our tools is put into production. This site includes web interfaces to VIS and MC4QCD.

- `http://qcd.nersc.gov` This is the new NERSC "gauge connection" archive (developed in collaboration with NERSC). It also stores some videos created using our tools.

- `http://tests.web2py.com/ildg` This is a new proposed web site for the International Lattice Data Grid. It provides an interface for searching lattiec QCD data in a visual way.

3.3 Fostered Collaborations

During this research project we have collaborated with Prof. James Hetrick from the University of Pacific and David Skinner at NERSC to re-build the new "gauge connection" web site.

We collaborated with Prof. Werner Berger from the Center for Computation and Technology, Louisiana State University and together we organized the 6th High End Visualization Workshop.

We utilized the VisIt software created by the Lawrence Livermore National Laboratory. Although we did not interact directly with the authors we interacted inderectly by using various online resources generated for that project.

Finally we interacted with the rest of the USQCD and the ILDG collaborations, from which we received constant feedback and suggestions.

3.4 Personnel

This grant has funded the PI and some of the following graduate students, who contributed to this research:

- Nate Wilson

- Yaoquan Zhong

- Brian Schinazi

- Tony Garcia
- Chris Baron
- Vincent Havery

3.5 Published Papers

In the following bibliography we list all the papers published by the PI and supported directly by this research grant. We omitted papers and books published by the PI on other topics not directly related to the grant scope.

References

[1] M. Di Pierro "qcdutils" arXiv:1202.4813 [hep-lat]

[2] M. Di Pierro "Improving Non-Linear Fits" arXiv:1202.0988 [cs.NA]

[3] M. Di Pierro, J. Hetrick, S. Cholia, D. Skinner "Making QCD Lattice Data Accessible and Organized through Advanced Web Interfaces" PoS **LAT2011** (2011) arXiv:1112.2193 [hep-lat]

[4] E. T. Neil *et al.* [for the Fermilab Lattice Collaboration and for the MILC Collaboration], "B and D meson decay constants from 2+1 flavor improved staggered arXiv:1112.3978 [hep-lat]

[5] A. Bazavov *et al.* [Fermilab Lattice and MILC Collaborations], "B- and D-meson decay constants from three-flavor lattice QCD," arXiv:1112.3051 [hep-lat]

[6] M. Di Pierro, Y. Zhong and B. Schinazi, "Vis: Online analysis tool for lattice QCD," PoS **LATTICE2010**, 326 (2010).

[7] M. Di Pierro and Y. Zhong, "Analysis and visualization tools for Lattice QCD," PoS **LAT2009**, 038 (2009).

[8] M. Di Pierro, "Vis: QCD workflow and visualization tool" in Proceedings of the 6th High End Visualization Workshop, Dec 8th - 12th, (2010), Obergurgl (Austria), Lehmanns Media, ISBN 978-3-86541-361-1

[9] M. Di Pierro, Y. Zhong and B. Schinazi, "mc4qcd: Online Analysis Tool for Lattice QCD," PoS A **CAT2010**, 054 (2010) arXiv:1005.3353 [hep-lat]

[10] C. Bernard *et al.* [Fermilab Lattice Collaboration and MILC Collaboration], "Tuning Fermilab Heavy Quarks in 2+1 Flavor Lattice QCD with Application to Phys. Rev. D **83**, 034503 (2011) arXiv:1003.1937 [hep-lat]

[11] A. Bazavov *et al.* [Fermilab Lattice and MILC Collaborations], "The Ds and D+ Leptonic Decay Constants from Lattice QCD," PoS **LAT2009**, 249 (2009) arXiv:0912.5221 [hep-lat]

[12] T. Burch *et al.*, "Quarkonium mass splittings in three-flavor lattice QCD," Phys. Rev. D **81**, 034508 (2010) arXiv:0912.2701 [hep-lat]

[13] J. A. Bailey *et al.* [Fermilab Lattice Collaboration and MILC Collaboration], "Progress on charm semileptonic form factors from 2+1 flavor lattice QCD," PoS **LAT2009**, 250 (2009) arXiv:0912.0214 [hep-lat]

[14] T. Burch *et al.*, "Quarkonium mass splittings with Fermilab heavy quarks and 2+1 flavors of PoS **LAT2009**, 115 (2009) arXiv:0911.0361 [hep-lat]

[15] M. Di Pierro *et al.*, "Visualization as a tool for understanding QCD evolution algorithms," J. Phys. Conf. Ser. **180**, 012068 (2009).

[16] C. Bernard *et al.*, "Visualization of semileptonic form factors from lattice QCD," Phys. Rev. D **80**, 034026 (2009) arXiv:0906.2498 [hep-lat]

[17] C. Bernard *et al.*, "B and D Meson Decay Constants," PoS **LATTICE2008**, 278 (2008) arXiv:0904.1895 [hep-lat]

[18] J. A. Bailey *et al.*, "The B —¿ pi l nu semileptonic form factor from three-flavor lattice QCD: Phys. Rev. D **79**, 054507 (2009) arXiv:0811.3640 [hep-lat]

[19] C. Bernard *et al.*, "The Anti-B $\rightarrow$ D* l anti-nu form factor at zero recoil from three-flavor Phys. Rev. D **79**, 014506 (2009) arXiv:0808.2519 [hep-lat]

[20] M. Di Pierro, "A Visualization Toolkit for Lattice Quantum Chromodynamics", in Proceedings of the 4th High End Visualization Workshop, June 18th - 22th, (2007), Obergurgl (Austria), Lehmanns Media, ISBN 9783865412164

[21] C. Bernard *et al.* [Fermilab Lattice, MILC and HPQCD Collaborations], "The decay constants f(B) and f(D+) from three-flavor lattice QCD," PoS **LAT2007**, 370 (2007).

[22] R. Todd Evans, E. Gamiz, A. X. El-Khadra and M. Di Pierro, "A Determination of the B0(s) and B0(d) mixing parameters in 2+1 lattice PoS **LAT2007**, 354 (2007) arXiv:0710.2880 [hep-lat]

[23] M. Di Pierro, "Visualization for Lattice QCD" PoS **LAT2007**, 331 (2007) `http://pos.sissa.it/archive/conferences/042/031/LATTICE2007_031.pdf`

The Secret Life of Quarks

Final Report, University of North Carolina at Chapel Hill.
DE-FC02-06ER41445, October 2006 – March 2012.

Robert J. Fowler
RENCI, University of North Carolina at Chapel Hill
(rjf@renci.org, 919.445.9670)

I. Overview

This final report summarizes activities and results at the University of North Carolina as part of the the SciDAC-2 Project *The Secret Life of Quarks: National Computational Infrastructure for Lattice Quantum Chromodynamics*.

The overall objective of the project is to construct the software needed to study quantum chromodynamics (QCD), the theory of the strong interactions of subatomic physics, and similar strongly coupled gauge theories anticipated to be of importance in the LHC era. It built upon the successful efforts of the SciDAC-1 project *National Computational Infrastructure for Lattice Gauge Theory*, in which a QCD Applications Programming Interface (QCD API) was developed that enables lattice gauge theorists to make effective use of a wide variety of massively parallel computers. In the SciDAC-2 project, optimized versions of the QCD API were being created for the IBM BlueGene/L (BG/L) and BlueGene/P (BG/P), the Cray XT3/XT4 and its successors, and clusters based on multi-core processors and Infiniband communications networks. The QCD API is being used to enhance the performance of the major QCD community codes and to create new applications. Software libraries of physics tools have been expanded to contain sharable building blocks for inclusion in application codes, performance analysis and visualization tools, and software for automation of physics work flow. New software tools were designed for managing the large data sets generated in lattice QCD simulations, and for sharing them through the International Lattice Data Grid consortium.

This effort was a project of the USQCD Collaboration, which consists of nearly all the high energy and nuclear physicists in the United States engaged in the numerical study of QCD. The Collaboration's web page www.usqcd.org contains information regarding its scientific objectives, membership, and initiatives. All software developed under this grant is publicly available, and can be downloaded from a link on the USQCD web site, or directly from the URL usqcd.jlab.org/usqcd-software. Ten universities and three national laboratories were funded on the project.

As part of the overall project, researchers at UNC were funded through ASCR to work in three general areas. The main thrust has been performance instrumentation and analysis in support of the SciDAC QCD code base as it evolved and as it moved to new computation platforms. In support of the performance activities, performance data was to be collected in a database for the purpose of broader analysis. Third, the UNC work was done at RENCI (Renaissance Computing Institute), which has extensive expertise and facilities for scientific data visualization, so we acted in an ongoing consulting and support role in that area.

This project paid for approximately 0.3 of an FTE staff member per year at UNC over most of its lifetime. Beginning in September 2011 the level of effort and spending rate was progressively reduced to permit continued participation in SciDAC QCD activities through the no-cost extension period.

II. Results and Findings

Throughout the course of the project UNC personnel served on the USQCD Software committee. This participation included active participation in meetings, both weekly telecons as well as semi-annual face-to-face project conferences. These interactions formed the basis for UNC personnel to act in a Computer Science consulting role on a variety of topics and to serve as a liaison between USQCD and the broader Computer Science research community.

During the first year of the of the project, computer scientists at the University of North Caroline focused on designing and developing a web based High Performance Computing (HPC) Performance Database targeted particularly on QCD applications. It was used for performance profiling of the latest releases of MILC code. The HPC Database is a web based infrastructure designed to store the performance data collected by our performance team at Renaissance Computing Institute (RENCI) for QCD applications on various high performance computing systems. It provided web interfaces for users to browse the performance data, perform statistical analysis and conduct performance comparisons. The goal was to create a knowledge base for maintaining and sharing the performance analysis results among the QCD community. The UNC team completed an initial implementation of the system. As of the end of the year, the database contained the performance files collected by running SvPablo instrumented MILC on several HPC systems during the year of 2006. All the performance data can be browsed via the web interface. In addition, two simple web queries are provided for fine grained data search. With partial funding from SciDAC PERI (Performance Engineering Research Institute) project, the HPC Database was merged with PERI Performance Database as part of PERI's Application Engagement effort. The current version of this is the TAU DB supported by the University of Oregon.

The database was used to collect results for a study of the performance scaling properties of MILC. The performance profiling was done using SvPABLO to instrument the FOR_ALL_SITES loops and the contained SU(3) operations in MILC using hardware performance counters to get definitive measurements of the effects of memory bandwidth and latencies of the on-node calculations. This was done on several classes of system, including a BlueGene/L, an SGI shared memory system, and several Linux clusters. The goal was to measure the performance "headroom" and to identify restructuring methods, either manual or compiler-based, to improve on-node performance and to reduce the performance "drop off" as the local domain size increases beyond the size of cache. This methodology separated the performance of phases that are constrained by on-node activity and nearest neighbor communication from from phases dominated by all-to-all communication. The general finding was that the nearest neighbor aspects scale linearly on all systems investigated. The on-node activities were found to be severely constrained by the system memory cache size, thus causing severe performance degradation for local problem sizes that don't fit in level 2 cache. The Krylov method solvers used for matrix inversion exhibit comparatively poor scaling with large numbers of processors because they eventually dominated by all-to-all communication. One conclusion is that even without changing the solver efficiency could be improved by mitigating

the "cache size cliff" to allow larger local sub-problems to be done efficiently. The results were shared with the MILC community. While the qualitative nature of the results were expected, this work helped to quantify the magnitude of the problems and to emphasize trade-offs and areas for improvement. Increased attention was paid to loop ordering, e.g., the Morton ordering used in the MIT Moebius inverter, and to tiling issues. It also helped to quantify the advantages of pursuing new inverter methods through the remainder of this project and its follow-ons.

One goal of this scaling study was to use measurements on multiple systems to quantify the relative importance of CPU and memory performance for QCD applications. In the end, CPU and memory performance, both latency and bandwidth, were all on co-linear curves for the systems available at the time.

The results were written up and circulated in Zhang *et al.* "Performance Characterization for HPC Systems and a QCD Application".

The HPC Performance Database was offered to the physics community to track performance of QCD applications. While it was found to be useful for the scaling study, for a variety of reasons it was not embraced by the physics community. Aspects of the database continue to be used in SciDAC PERI and SciDAC SUPER in the form of the TAU DB.

In the second year of the project (2007-2008), Zhang worked with the MILC collaboration to use the MILC code as one of the major components of the PERI performance database work, which is being used to drive auto-tuning efforts.

A second major effort was performance measurement, analysis, and tuning activities for the QDP++ and Chroma code base in coordination with with SciDAC PERI. This coordination benefited the QCD project by bringing to bear other researchers within PERI. It benefited PERI by presenting a set of challenging research problems not addressed by existing performance tools.

Chroma uses the expressive capabilities of the C++ programming language more aggressively than most other high performance scientific codes. Chroma was thus used to drive the extension of the analysis capabilities of HPCToolkit from Rice University.

QDP++ and Chroma are written using very expressive and powerful modern idioms in C++, specifically template meta-programming and expression templates. This is a very powerful approach that enables complex calculations to be specified in a concise manner, thus improving programmer/scientist productivity. The down side from the performance analysis perspective is that a single source line statement can expand through inlining and template into the equivalent of many hundreds of lines of code and can implicitly invoke C++ methods that recursively expand into many lines of code. Furthermore, after inlining, subsequent compiler optimization passes may reorder object code. This is important for efficiency, but it allows only very coarse grain performance analysis. Inserting instrumentation at the source level impairs performance by inhibiting inlining and other optimizations. Furthermore, instrumentation in inner loops can itself be very expensive. Sampling-based performance profiling tools can avoid the performance penalty, but needs a robust mechanism for attributing costs in optimized object code back to the original source code. Using driving problems from JLab, UNC and Rice addressed the issue of adding the necessary algorithms to the HPCToolkit performance tool to derive such attribution. This work has produced a masters

thesis (Nathan R. Tallent, *Performance Analysis of Optimized Code: Binary Analysis for Performance Insight*) and in two conference papers (Tallent *et al. Binary analysis for measurement and attribution of program performance.*, and Tallent em et al. *Diagnosing performance bottlenecks in emerging petascale applications.*) The work at UNC on this effort was funded through the QCD SciDAC project and the work at Rice was funded through SciDAC PERI.

RENCI personnel (Fowler and Porterfield) worked with JLab on the issue of multi-core performance of LQCD codes. Aspects of the analysis of threading strategies for a DSlash kernel on quad-core AMD chips, such as those used in the ORNL Jaguar (and other Cray XT systems) were reported at SciDAC 2008 in the form of a poster and a journal paper (Fowler *et al.*. *Performance engineering challenges: the view from RENCI.*) This entailed a comparison of three strategies for multi-core parallelism: "MPI everywhere". "MPI and OpenMP 2.5", and "MPI and QMT", where QMT is a threading library written at JLab specifically for QCD computations. Experimental results indicated that the "MPI everywhere" and "MPI and QMT" were competitive, but that "MPI and OpenMP" was considerably worse. We determined that relatively poor performance of OpenMP relative to QMT is due to cores going into and out of a low power mode when they are idle rather than simply going into an active idle loop. Version 3 of the OpenMP standard allows applications to explicitly request "busy waiting", so the comparative advantage of QMT may be diminished when OpenMP v.3 becomes widely available.

The results on synchronization discovered through studying Chroma were further applied under the aegis of SciDAC PERI to improve the performance of Robert Harrison's MADNESS chemistry code at ORNL. UNC and Rice worked on generalizing the analysis methods for diagnosing such problems. See (Tallent, Mellor-Crummey, and Porterfield. *Analyzing lock contention in multithreaded applications.*) for a description of some of the analyses and tool improvements.

Part of UNC's work plan for the project was to exploit RENCI's extensive scientific visualization resources and expertise. During this project year Dreher, Borland, and Fowler from RENCI worked with DiPierro of DePaul to produce prototype visualization capabilities. DiPierro has continued the work on his own with occasional consultation from UNC.

During the 2008-2009 project year, work at the University of North Carolina under the project continued to be focused primarily on performance measurement, analysis, and tuning activities in collaboration with the SciDAC Performance Engineering Research Institute.

During this project year we used UNC internal funding to supplement our SciDAC funding to permit us to continue our collaboration with Massimo DiPierro of DePaul on visualization.

The work on multi-core performance continued during the 2009-2010 project year. Because multi-core, multi-socket performance is often dominated by bottlenecks at resources shared between cores, it is necessary to use measurement and analysis methods that expose contention at those resources. Almost all existing performance tools using hardware performance counters are based on a "first person" perspective in which each process or thread executes performance instrumentation code that accesses virtualized hardware performance counters that are swapped when control is shifted between processes (threads). On most systems the operating system does not support the virtualization of performance counters that monitor off-core resources so these tools cannot provide the necessary information. Some systems do allow virtualized counters to view off-core

resources, but the resulting measurements are nearly impossible to interpret. Since each process observes the shared resource only when it is scheduled to run, no application process captures all of the events at the resource. On the other hand, during time intervals when multiple processes run concurrently on several cores, each event is redundantly counted by all the active processes. The resulting counts are of little use.

To meet these deficiencies, under the SciDAC PERI project UNC began development of an approach we call "Resource Centric Reflection" (RCR) to use a third-person approach to produce node-wide measurements of shared resources, to compute real time analyses of this information to translate raw counts into more useful quantities, and to share this information with system and application programs, including first-person performance tools such as the University of Oregon's TAU and Rice University's HPCToolkit. Based on this approach UNC researchers developed methods for characterizing the degree of memory concurrency supplied by a system (See Mandal *et al.*, 2010.) and under this project used the method to analyze the offered load from LQCD codes. We determined that for the generation of multi-core chips then in use that the QCD codes are memory latency bound, but that if the number of cores were to be increased, they would soon be limited by memory bandwidth unless there were dramatic changes in memory hierarchy design.

In the final year of the original plan (2010-2011) UNC continued to address issues of performance measurement and analysis on multicore and manycore systems in the context of QCD codes.

The UNC group participated in in the discussions regarding GPU algorithms and in the configuration strategies for the hardware that was deployed at JLAB. In support of this, we procured a standalone system with multiple GPUs and began development work on integrating multi-GPU performance monitoring with our framework for monitoring multi-core, multi-socket servers. The tool work done under a non-disclosure agreement with Nvidia that gave UNC access to their CUPTI performance interface to CUDA. Similar work was done at the University of Tennessee Knoxville and Rice University, but the agreements with Nvidia prevented us from sharing information and code among ourselves until CUPTI was made public after the end of this project.

The SciDAC multi-GPU lattice code library QUDA was the main driving application for starting the development of a comprehensive performance measurement environment to integrate Nvidia CUPTI measurements, first-person CPU call-stack profiling using Rice's HPCToolkit, and node-wide bottleneck analysis using UNC's RCRTool. This effort saw mixed success. The effort to get RCRToolkit and RCRTool to interoperate was successful and an analysis of the Chroma code was reported in (Mandal *et al.,* 2012). The versions of CUPTI available during this period, however, were limited to producing only relatively coarse measurements of GPU codes and required that the GPU code run synchronously with the CPU, *i. e.,* the CPU and GPU could not operate simultaneously. This was judged unacceptable for our purposes, so we were forced to abandon this line of development.

Midway through the final year, the participating institutions began to prepare for the end of the SciDAC 2 project and to prepare for SciDAC 3. Anticipating a gap between the programs and under advice from DOE ASCR, the institutions requested six month no-cost extensions and prepared a proposal for a funded extension year. UNC received a six month unfunded extension from March 2011 to Sept 2011. This was extended by another six months to March 2012. While some of the project partners were funded for the full year, UNC did not receive new additional funding.

In response, to keep the collaboration active, beginning in Sept. 2010 we progressively reduced our level of effort, thus stretching the remaining funds to permit continued participation in meetings and technical consultation with the QCD community.

During this period UNC continued to perform specific performance experiments in support of USQCD code developers, particularly with respect to details of memory hierarchy behavior and of the pipeline efficiency of the low level SU(3) operators used in LQCD calculations. The paper (Mandal *et al.,* 2012) uses one aspect of the memory behavior of Chroma as an example.

Discussions and performance experiments conducted during the extension year were also driven by the need to identify long-term strategic problems facing LQCD code development for emerging and future systems. As we move into the exascale era, the number of cores per chip will continue to increase while other aspects of the system such as available power, off-chip memory bandwidth, and interconnect bandwidth will be stagnant or will grow much more slowly. Such architectures will require highly-parallel code that can be vectorized to use pipelined, streaming functional units. The QUDA library for Nvidia GPUs is an example of software that has pioneered the structuring of code to have these properties. The methods used here were recognized by at least part of the USQCD community as being needed on other new system architectures such as the Intel Many Integrated Cores (MIC) and on main stream chip such as the Intel Xeon as well.

In contrast, the QDP++ library can be viewed as a very powerful data parallel, application specific language that has improved scientific productivity by making final application development much easier. Unlike other data parallel languages, such as High Performance Fortran, that were based on cutting edge compiler techniques, QDP++ is implemented using extensive C++ template meta-programming, including the use of expression templates. While this approach can generate relatively good code at the statement level, it cannot apply inter-statement, outer-loop, or global analyses and code generation methods the way a conventional compiler can. In particular, it cannot directly generate the kinds of streaming codes needed to get the best performance for GPUs and other emerging architectures.

Our discussions and groundwork during this period about future approaches to compilation for domain-specific QCD languages influenced the design of the QLUA code being developed at MIT. Through our interactions with USQCD researchers at JLab and Frank Winter of the University of Edinburgh, it has influenced the development of QDP-JIT. It also provided the basis for the strategy for code generation we are taking in the SciDAC-3 Project "Computing Properties Of Hadrons, Nuclei And Nuclear Matter From Quantum Chromodynamics" under funding agreement DE-SC000876.

Publications and Presentations

1. Robert J Fowler, Todd Gamblin, Allan K Porterfield, Patrick Dreher, Song Huang, and Balint Joo. "Performance engineering challenges: the view from RENCI" J. Phys: Conf. Ser, page 5pp, October 2008.
2. Nathan R. Tallent, John M. Mellor-Crummey, Laksono Adhianto, Michael W. Fagan, and Mark Krentel. Diagnosing performance bottlenecks in emerging petascale applications. In SC '09: Proc. of the 2009 ACM/IEEE 2009. ACM. (doi:10.1145/1654059.1654111).

3. Nathan R. Tallent, John Mellor-Crummey, and Michael W. Fagan. Binary analysis for measurement and attribution of program performance. In PLDI '09: Proc. of the 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 441-452, New York, NY, USA, 2009. ACM. Distinguished Paper. (doi:10.1145/1542476.1542526).

4. Nathan R. Tallent, John M. Mellor-Crummey, and Allan Porterfield. Analyzing lock contention in multithreaded applications. In PPoPP '10: Proc. of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 269-280, New York, NY, USA, 2010. ACM. (doi:10.1145/1693453.1693489).

5. Anirban Mandal, Rob Fowler, and Allan Porterfield. Modeling memory concurrency for multi-socket multi-core systems. In Proceedings of the 2010 IEEE International Symposium on Performance Analysis of Systems and Software, White Plains, NY, March 2010. IEEE.

6. Anirban Mandal, Robert Fowler, and Allan Porterfield. System-wide introspection for accurate attribution of performance bottlenecks. In Workshop on High-performance Infrastructure for Scalable Tools (WHIST), Venice, Italy, 2012.

7. Ying Zhang, Kevin Gamiel, Mark Reed, Morten Somervoll, Jeffery Tilson, and Daniel Reed. Performance Characterization for HPC Systems and a QCD Application, 2007. (Unpublished Report).