Renewal of SciDAC Grant National Computational Infrastructure for Lattice Gauge Theory

Lattice QCD Executive Committee

R. Brower, (Boston U.) N. Christ (Columbia U.), M. Creutz (BNL),P. Mackenzie (Fermilab), J. Negele (MIT), C. Rebbi (Boston U.),S. Sharpe (U. Washington), R. Sugar (UCSB) and W. Watson, III (JLab)

December 28, 2003

1 Introduction

We request a two year renewal of our SciDAC Grant *National Computational Infrastructure for Lattice Gauge Theory*, for the period July, 2004 to July, 2006. During the first two and half years of this grant, we have made very significant progress in constructing the computational infrastructure needed by the U.S. lattice gauge theory community for the study of quantum chromodynamics (QCD), the theory of the strong interactions. We have designed a QCD Applications Program Interface (QCD API) which will enable members of our community to make efficient use of terascale computers, and we have built prototype clusters to test this software and to study optimal architectures for QCD simulations. At the same time, but with separate funding, physicists at Columbia University in collaboration with colleagues at IBM have completed development work on the QCDOC, a special purpose computer designed specifically for the study of QCD.

During the renewal period we propose to optimize the QCD API for large scale parallel applications, port major applications to it, and develop the grid tools needed to implement a distributed national computing facility for the study of QCD. We also propose to continue to study cluster optimization with the aim of being in a position to build highly efficient multi-terascale clusters in 2005 and beyond. We are seeking separate funding for the deployment of terascale hardware and its scientific utilization. In particular, we propose to install a ten teraflop/s sustained QCDOC at Brookhaven National Laboratory (BNL) in 2004, and multi-teraflop/s clusters at Fermi National Accelerator Laboratory (FNAL) and Thomas Jefferson National Accelerator Facility (JLab) in subsequent years.

On February 6, 2003 the DOE convened a panel of physicists and computer scientists chaired by Frank Wilczek to review our overall project. A copy of the panel's report is provided in Appendix A. Among the panel's comments of particular relevance to our SciDAC grant were:

- "The scientific merit of the suggested program is very clearly outstanding."
- "It is proposed to pursue two separate hardware tracks, one using specially designed systemson-a-chip that leverage industrial intellectual property cores, the other using general-purpose computing systems. ... We therefore feel it is prudent, as well as interesting, to pursue both tracks, at least until a clear winner or a synthesis emerges."
- "The software development component of the proposal is also novel in this context and extremely important. ... The pursuit of two separate hardware tracks will aid in the development of robust, portable software. If successful, the software component could be very valuable both in itself and as a model for other scientific enterprises."
- "The proposed programs are of considerable interest from the point of view of computational science, since they could provide convincing models and demonstrations of the use of cost effective special architectures for scientific problems."
- "Both the proposers and the DOE should recognize that this is an endeavor that is not likely to be exhausted in 4 years or even in 10."

Our scientific objectives were set out in detail in our original SciDAC proposal. In brief they are to understand the physical phenomena encompassed by quantum chromodynamics, and to make precise calculations of the theory's predictions. This requires large scale numerical simulations within the framework of lattice gauge theory. Such simulations are necessary to solve fundamental problems in high energy and nuclear physics that are at the heart of the Department of Energy's large experimental efforts in these fields. Computational facilities capable of sustaining many tens of teraflop/s are needed to achieve our near term scientific goals.

Major goals of the DOE's experimental program in high energy and nuclear physics are to: 1) verify the Standard Model of High Energy Physics, or discover its limits, 2) understand the internal structure of nucleons and other hadrons, and 3) determine the properties of hadronic matter under extreme conditions. Lattice QCD calculations are essential to research in all of these areas. Our objective under the SciDAC Program is to create the computational infrastructure needed to study QCD. We expect that the scientific research which will make use of this infrastructure will be funded under the DOE programs in high energy and nuclear physics.

Our software and hardware development efforts are entering an important new stage. The basic design of the software has been completed and development clusters have been deployed at FNAL and JLab. The QCDOC ASIC, mother board and daughter board have been tested, and the first multi-processor QCDOC has been assembled. The core components of the software have been developed, tested and benchmarked. They will enable the U.S. lattice gauge theory community to use the planned ten teraflops OCDOC and the prototype clusters as these machines come on line. However, these components do not constitute the fully integrated software infrastructure needed to sustain the broad objectives of the national program in lattice gauge theory. Additional work is required to achieve a full data parallel interface, and to port large production codes to it. Further work is also required to optimize the QCD API for large scale parallel applications on the platforms for which it is targeted. Our plans call for the construction of a distributed national facility with major computational resources at BNL, FNAL and JLab. Additional effort is required to develop a uniform and convenient user environment for this distributed facility. Ongoing work is also needed on software engineering to release and maintain the SciDAC QCD software. This work includes code management, regression testing, and porting of the software to additional compilers, interconnects and processor nodes. We also propose work in the area of algorithmic design, performance analysis and common standards for run-time and interactive environments and data grid tools. The very considerable software effort we propose for the renewal period will yield a robust, production environment for the national lattice gauge theory community.

In Section 2 we outline the software development work carried out to date or scheduled to be completed prior the end of the current grant, as well as the work we propose for the renewal period, July 2004 to July 2006. In Sections 3 and 4 we describe the status of the cluster effort and the QCDOC, and our hardware plan for the next two years. In Section 5 we briefly outline the management structure of our project, and in Section 6 we set out the proposed budget.

2 Software Development

The goal of the Lattice QCD software infrastructure project is to create a unified programming environment which will enable the U.S. lattice gauge theory community to achieve high efficiency on the computer architectures targeted by our project, the QCDOC and optimized clusters, and on commercial supercomputers. It is important that users be able to develop new applications rapidly, and that the large investment that has been made in existing codes be preserved.

Lattice gauge theory calculations are well suited to massively parallel computers. They employ regular grids or lattices, so the computational load can be balanced by assigning equal numbers of lattice points to each processor. In updating variable on a given lattice site or link, one needs data from a limited number of neighboring lattice sites. Only when one or more of these neighboring sites is on a different processor than the one being updated is inter-processor communications necessary. Thus, communications are regular and predictable, and can usually be overlapped with computation. These simplifying features of QCD calculations must be taken into account when designing hardware and software if high efficiency is to be obtained. The QCD API does this through the three level structure shown in the chart below. At level 1 are the message passing API (QMP), which handles inter-processor communications, and a library of single site linear algebra routines (QLA) common to most lattice gauge theory calculations. Level 2 (QDP) contains data parallel operations which are built on QMP and QLA. A very large fraction of the resources in any lattice QCD simulation go into a few computationally intensive subroutines, most notably the repeated inversion of the Dirac operator. To obtain the level of efficiency at which we aim, it is necessary to optimize these subroutines for each architecture. These highly optimized subroutines constitute Level 3. They can be called from QDP or from C and C++ applications. Each component of the QCD API is described in detail in a subsection below.

Level 3			
Dirac Operators, CG Routines, etc.			
QDP lib Level 2			
Data Parallel QCD Lattice Operations			
(overlapping Algebra and Messaging)			
e.g. $A = SHIFT(B, mu) * C$; Global sums, etc			
Lattice Wide Linear Algebra	Lattice Wide Data Movement		
(No Communication)	(Pure Communication)		
e.g. $A = B * C$	e.g. Atemp = SHIFT(A, mu-dir)		

QCD–API Level Structure

QLA lib	Leve	vel 1 QMP lib	
Single Site & Vector Lin Alg API		Message Passing API	
e.g. SU(3), Dirac algebra, etc.		(Maps Lattice into Network)	

Richard Brower, the Software Coordinator, has overall responsibility for the software effort. He

is assisted by the Software Committee, whose membership is given in Section 5. The Software Committee operates by weekly conference calls and face to face Software Design workshops that have been held at JLab on November 8–9, 2001, at JLab on February 2, 2002, at MIT on June 26, 2002, and most recently at FNAL on February 20, 2003. Individual software developers participate in the calls and meetings as needed.

The software code and documentation can be found at http://www.lqcd.org/ and the working documents of the Software Coordinating Committee are at http://physics.bu.edu/~brower/SciDAC/scc.html. A useful set of introductory slides from the Software Tutorial given on Feb 22, 2003 at FNAL are posted at

http://physics.bu.edu/~brower/scidac_software_tutorial.

The basic design of the QCD API has now been completed, and the Software Coordinating Committee is nearing completion of the basic design of the I/0 routines required for the planned data archives, and grid tools. During the renewal period the focus of the software effort will shift from design and prototyping to full implementations, optimization, testing and support for terascale production work. In addition, we anticipate an on going need to respond quickly to new directions in physics, algorithms, hardware and grid tools. During the 2004–2006 time period, we must simultaneously maintain a production environment for terascale computations, and continue the development of the software and hardware infrastructure.

2.1 Optimized Network Communications–QMP

QMP defines a uniform subset of MPI-like functions equivalent to those used in existing QCD application code. In addition QMP extends this core set of MPI functions in two areas: (1) it partitions the QCD space–time lattice and maps it into the geometry of the hardware network, providing a more convenient abstraction for the Level 2 data parallel API (QDP); (2) it contains specialized routines designed to access the full hardware capabilities of the QCDOC network and to aid optimization of low level protocols on networks in use and under development on the clusters.

Release 1.0 of the QMP message passing API is published on **lqcd.org**, along with complete documentation. It includes: (1) a message passing library design and binding for both C and C++, (2) code implementing QMP atop MPI so that application codes can be ported and run anywhere linked to the MPI implementation, (3) an implementation of QMP atop GM to provide higher performance than MPI for clusters using Myricom's Myrinet interconnect, and (4) an implementation atop VIA and gigabit ethernet to support the new gigabit ethernet mesh cluster at JLab. There is a basic test suite to verify each implementation.

An implementation of QMP for the QCDOC is nearing completion. The QCDOC has important hardware functionality, such as the ability to start twenty-four different communications with a single CPU instruction, and persistent storage in the communications hardware of the data pattern for repeated communications transfers. These features are supported in the QMP implementation for the QCDOC by incorporating the QCDOC's native operating system communications calls. At present this implementation contains the nearest-neighbor functionality of QMP, and has been tested on both the QCDOC ASIC simulator and initial hardware. A complete implementation is

expected by January of 2004.

Prior to the start of this project most QCD code was written in C or C++ using MPI for communications. It is important that the investment in this large body of code be preserved, so a design requirement of QMP has been that C and C++ code must run with good performance on the target architectures simply by replacing MPI calls by corresponding QMP ones. The MILC Collaboration's code was chosen to test this requirement. It is written in C, is freely available, and is used by a significant fraction of the U.S. lattice gauge theory community. Bringing up the MILC code on the QCDOC was a particularly important test of QMP, because the architecture of this machine is significantly different from that for which the code was originally designed. In addition, concerns are sometimes expressed regarding the porting of large code bases to special purpose machines. The MILC code has now been run on both the QCDOC simulator and on initial hardware, obtaining a performance of approximately 20% of peak for key sections with local volumes as small as 4^4 lattice sites per processor. For comparison, the same code typically obtains 10% to 15% of peak on commercial supercomputers with performance falling rapidly for local volumes less than 8^4 lattice sites per processor. It is, of course, highly advantageous to work with small local volumes, since one can then bring large numbers of processors to bear on individual problems. These tests indicate that QMP enables the MILC code to take advantage of the exceptional communications system of the QCDOC with very little effort by the applications programmer. One expects similar results for other C and C++ codes.

With the dramatic improvement in price/performance of commodity processors in recent years, network interconnects have become a critical limiting factor in the design of clusters. We therefore project extensive network software experimentation and development during the renewal period, as we prepare for the deployment of terascale clusters in 2005 and beyond.

We are presently working on the efficient use of gigabit ethernet inter–connects with fixed grid topologies. During year three of this project, software for the gigabit ethernet mesh will be further optimized. Additional network technologies must also be investigated, and additional network–specific implementations of QMP will likely emerge. For example, although Infiniband can be tested initially via the MPI implementation QMP, it is one likely target for optimization in the renewal period.

In order to enhance our ability to optimize QMP and port it to new network architectures, we plan to develop an improved testing procedure for the API. Our computer science colleagues at the University of Illinois¹ have carried out preliminary work aimed at creating an exhaustive test suite for the QMP API. A structure has been established that defines a standard for current and future tests, as well as the format for test reporting, generated as an XML file, which can be viewed with any web browser that supports style sheets (e.g., Internet Explorer). This approach is similar to that used in the tests of the MPICH toolkit, a popular MPI implementation from Argonne National Laboratory. The first version of the test suite comprised functional tests, and it is being extended now to include performance tests. During the renewal we will complete a full version of the QMP test suite, including performance tests aimed at measuring the latency and bandwidth provided by the QMP implementation under study. The testing procedure for QMP will be reviewed by the

¹Daniel Reed and Celso Mendes will move from the University of Illinois to the University of North Carolina on January 1, 2004. They will continue to work on this project at their new institution.

Software Committee, and emulated for other SciDAC software libraries where deemed appropriate.

2.2 Linear Algebra Kernels–QLA

All lattice gauge theory 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 lattices sites or links, and do not involve inter–processor communications. We have gathered them together into a single level 1 library called QLA. The QLA routines can be used in combination with QMP to develop complex data parallel operations in QDP or in existing C or C++ code. There are both C and C++ versions of QLA. The C implementation has on order 24,000 functions generated in Perl, with a full suite of test scripts. The number of functions in C++ implementation of QLA is considerable reduced by making extensive use of the language's class structure and of operator overloading. We have made heavy use of "Expression Templates" employing a tool called PETE from LANL in writing the C++ version of QLA.

Considerable improvement in the performance of lattice QCD codes can be obtained by optimizing key linear algebra kernels. The SIMD facilities present on many of the processors likely to be used in our clusters provide a means for doing so. Examples of these instruction sets are SSE on Intel x86 processors and Altivec on IBM PowerPC processors. Although mathematics libraries exploiting these instruction sets are readily available, they optimize calculations involving large matrices, and do poorly on the small SU(3) matrices required for lattice QCD calculations. Consequently we have written a number of the most frequently used routines ourselves, employing the SSE instruction set. The best performance is obtained when when these routines are expressed in an inlined form, directly in C/C++ code.

The impact of this optimization effort is illustrated in Fig. TUNING, where we show results for improved staggered fermions on the next successor in the P4 processor family used in previous machines, a 2.8 GHz Pentium 4 with an 800MHz front side bus. This figure displays several important features. First, note the difference of more than a factor of two between the lowest, unoptimized curve and the totally optimized top curve. Second, observe that for large lattice size per processor, when the problem no longer fits in cache, performance is memory bandwidth limited. Here, the 50% increase in the front side bus from 533MHz on the most recent SciDAC machine to 800MHz on this machine directly translates into a 50% increase in performance. Finally, note the dramatic performance increase that arises when the entire problem fits in cache.

Similar benchmarks comparing results for different processors and interconnects can be found at **lqcd.fnal.gov/benchmarks**. By contrast, tests with the MILC code indicate that the IBM xlC compiler is so efficient that optimization of the QLA library for the PowerPC processor used in the QCDOC is less critical. This question will be revisited after we have more detailed benchmarks on the QCDOC hardware.

Unfortunately, the semantics of access to SIMD instruction sets varies from compiler to compiler. In the case of the GCC compiler, the language extensions available for accessing SSE instructions are very awkward, and in most cases results in code that is difficult to understand, debug and maintain.



Figure 1: The single processor performance in Mflop/s of the MILC code for the improved staggered (Asqtad) inverter on a 2.8 MHz Pentium 4 processor with a 800 MHz front size bus, as a function of lattice volume for various levels of tuning. The impact of temporary variables, which minimize cache misses, SSE instructions and inlining are all illustrated. The sharp fall off in performance occurs when the lattice volume becomes so large that the problem does not fit into cache.

Our strategy for implementing robust, maintainable SIMD code is to write the kernels using an assembler, rather than a C compiler. In the case of the Intel x86 architecture, we are using the NASM open source assembler. It has yielded very clean, maintainable code. We have written SSE routines for QLA in NASM for MILC code using a Perl machine translator to go from MILC NASM to inline GCC. During the renewal period we propose to combine these QLA routines with others written for QDP++ into a uniform library with consistent QLA semantics. Next we will implement this code for the new SSE3 instructions, which support complex arithmetic more directly. We will also extend the SIMD versions of QLA routines to double precision. We propose exploring a similar strategy for the Altivec instruction set, which is available on the PowerPC(G5) architecture.

2.3 QCD Data Parallel Interface–QDP

Level 2 of the QCD API is the data parallel interface QDP, which is built on top of the message passing (QMP) and linear algebra (QLA) libraries. It makes use of the local algebraic kernels of QLA and the communications routines of QMP to create lattice wide parallel operations. Complex expressions allow extensive overlapping of communication and computation in a single line of code. The objective is to enable new applications to be developed rapidly and to run on a wide range of architectures with high efficiency.

From the perspective of the applications programmer, QDP is the essential interface to the QCD

API. It allows the programmer to focus on physics without being concerned about the details of data movement or optimization. The hardware dependent features of QMP and QLA are transparent to the QDP programmer, and the highly optimized level 3 routines can be called directly from QDP. For these reasons, QDP code is straightforward to write, efficient, and highly portable. The MPI version of QDP enables it to run on commercial supercomputers and standard clusters. In addition, code can be developed and tested on single processor workstations. To appreciate the concise nature of code written in QDP++, the C++ implementation of QDP, consider a typical multiply and add algebraic expression,

$$a^{i}_{\alpha}(x) = U^{ij}_{\mu}(x)b^{j}_{\alpha}(x+\mu) + 2c^{i}_{\alpha}(x) \quad \forall x \in \text{even and} \forall i, \alpha,$$

which is found in code for the inversion of the Dirac operator. In this expression the Dirac spinor $a^i_{\alpha}(x)$ is to be evaluated at each even lattice site, x, by multiplying a Dirac spinor, $b^i_{\alpha}(x+\mu)$ from the nearest neighbor site, $x + \mu$, by an SU(3) gauge matrix $U^{ij}_{\mu}(x)$ associated with the lattice link between x and $x + \mu$, and then adding 2 times a Dirac spinor, $c^i_{\alpha}(x)$. μ is a vector along one of the four lattice axes of length equal to the lattice spacing. A lattice site is said to be even (odd) if the sum of its three space and one time components is an even (odd) multiple of the lattice spacing. (Einstein's summation convention implies a color contraction j = 1, 2, 3.) In QDP++ the code which performs this set of operations on all even lattice sites is

```
multild<LatticeColorMatrix> u(Nd);
LatticeDiracFermion a, b, c;
int mu;
    a[even] = u[mu] * shift(b,mu) + 2 * c;
```

This expression (and more complex variations) allow extensive overlapping of communication and computation in a single line of code. By making 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 user. The **[even]** target label and **shift** communication operator are examples of completely general user defined subsets and permutation maps included in the API. The implementation makes heavy use of standard operator overloading, as well as "Expression Templates" employing a tool called PETE from LANL, which eliminates temporaries and optimizes the performance.

A complete set of documents and the first code release for QDP/QLA with bindings in C and C++ is available at **lqcd.org**. During the renewal period a high priority will be placed on optimizing QDP for the QCDOC and for the terascale clusters we propose to build. Additional effort will go into code management and distribution, and into improving manuals.

One of our goals for QDP is to have it become the common programming environment for the entire U.S. lattice gauge theory community. This would facilitate the sharing of application codes, and reduce duplication of efforts. It would also be a boon to younger members of the field who would not have to learn new coding environments, or create their own, as they move from graduate school to postdoctoral positions to faculty positions. The sharing of application software has been a strong feature of the QCD community. The MILC Collaboration code and the Hadron Physics

Collaboration's SZIN code have been publicly available for many years, and are used by a significant portion of the community. With the QCD API we are now in a position to extend the sharing of code to the entire national community.

The development of application code will be an important activity during the renewal period. We envision two approaches. The Hadron Physics Collaboration is in the process of replacing the SZIN code with a new application base called Croma, which will be written entirely in QDP++, the C++ version of QDP. This effort will provide an example of a large, publicly available code written from the ground up in QDP. The MILC Collaboration has incorporated QMP and SSE versions of linear algebra routines into its code, and will include calls to the level 3 routines. This work already indicates that MILC and other C codes will run with high efficiency on the QCDOC, as well as on the optimized FNAL and JLab clusters. The MILC Collaboration has begun to compare the performance of critical components of its code with corresponding ones written in the QDP, and will write new code in QDP as appropriate. This effort will demonstrate an alternative route to the utilization of the SciDAC software, one that will enable rapid porting of existing codes to the targeted architectures.

Clearly, for QDP to gain wide acceptance, it must be proven to give high performance and increased convenience in terms of code development and portability. Once the development of application codes is under way, we plan to obtain feedback from programmers concerning these issues so that new releases of the QDP API code can evolve under the guidance of the community's experience. An example of QDP's potential is shown in Fig. 2, where we plot the performance of the conjugate gradient matrix inversion routine for improved staggered (Asqtad) quarks as a function of the local lattice volume. Results are shown for the standard MILC code and for QDP with and without SSE instructions. It should be noted that the MILC code has been optimized over a period of more than ten years, so the fact that the initial version of QDP slightly surpasses it is quite significant.

2.4 Level 3 Subroutines

The overwhelming fraction of the floating point operations in any lattice QCD calculation are consumed in a few computationally intensive subroutines. Foremost among these are the subroutines for the inversion of the quark Dirac operators, which account for 70% to 90% of floating point operations in typical computations. Level 3 of the QCD API consists of highly optimized versions of these critical subroutines. They can be called both from QDP and from standard C/C++ code.

The Scientific Program Committee has identified three quark actions as vital for initial projects on terascale computers: Wilson–Clover, Domain Wall, and Improved Staggered (Asqtad). Inverters for these three quark actions constitute the first set of level 3 routines. They are written in assembly language for the QCDOC. The critical part of these routines is the multiplication of a vector by the Dirac operator. Table 1 below shows the performance of this operation in Mflop/s per processor obtained on the ASIC simulator and confirmed on the initial hardware. The differences in performance have to do with the different ratios of floating point operations to data movement for the three actions. The peak speed of the processor is 1,000 Mflop/s, so the performance for the Wilson–Clover and Domain Wall quarks exceeds the design goal of 50% of peak. The Asqtad action puts a significantly higher demand on the communications system than the other actions, so



Figure 2: Comparison of performance of the MILC and QDP codes for the Asqtad inverter. Performance in Mflop/s per processor is shown as a function of the number of lattice sites per processor, L^4 , both with and without SSE instructions for each code. These tests used 16 processors on the FNAL cluster lqcd.

its performance on such a small local volumes is particularly noteworthy.

Up to now our prototype clusters have all employed Pentium 4 processors. Optimized inverters for Wilson–Clover, Domain Wall and Asqtad actions are being written for these processors using SSE instructions. The impact of our optimization program on the performance of these inverters is illustrated in Figs. 1 and 2. Many additional benchmarks, including performance for different processors and interconnects, can be found at the URLs **physics.bu.edu/ brower/tests** and **lqcd.fnal.gov/benchmarks**. During the renewal period we expect to extend the optimization program to additional processors, such as the G5. Furthermore, the development of new actions is one of the most important and actively pursued area of research in our field. We must be in a position to develop optimized inverters for new actions when they are ready for production work on both the QCDOC and clusters. In addition, for highly optimized actions, such as Asqtad, the calculation of the fermion force can become a major consumer of cycles. We will investigate the coding of level 3 routines for this operation as well. Methods and documentation need to be developed to streamline the optimization of level 3 routines.

Cost effective terascale clusters may well make use of multi–processor SMP nodes, so multi– threaded bindings for QLA, QMP and level 3 routines have been anticipated in our software design. We plan to implement multi–threaded versions of the code when the price of commodity SMP

Quark Action	Local Volume	Mflop/s per node
Wilson–Clover	2^{4}	560
Wilson-Clover	4^{4}	590
Domain Wall	2^{4}	470
Domain Wall	4^{4}	535
Asqtad	4^{4}	440

Table 1: Performance of QCDOC assembly code for the multiplication of a vector by the Dirac operator, for the three quark actions that will be used in initial projects on the QCDOC. Performance is from tests on the ASIC simulator, which were confirmed on the initial QCDOC hardware. The performance for Domain Wall quarks was taken from that on standard Wilson quarks, which is expected to be identical.

nodes warrants it.

2.5 Execution Environment

Don Holmgren at FNAL and Chip Watson at JLab, and their respective groups have been working together to develop the systems tools and software needed to run large clusters. These include software to monitor hardware, update BIOS, etc. This effort takes advantage of the experience of the FNAL staff on operations and the JLab staff on networking. Work on the QCDOC operating system is a major software task of the BNL/Columbia group. The OS for QCDOC has made substantial progress allowing the compilation of the entire Columbia Physics System (CPS) and the MILC code.

Efforts are now beginning at FNAL and JLab to build a unified user's environment with a goal of presenting to the users identical batch environments, identical commands for interacting with disk and tape resources, and identical development environments. Deployment and testing of the first version of this common user environment has begun at FNAL and JLab. The second iteration will aim at a common run-time environment and a common user interactive environment. As this effort matures, the BNL, FNAL and JLab facilities will be combined into a *Lattice QCD Meta-facility*, including data grid capabilities described below, plus virtual batch queues for job submission. This computational grid will leverage work within PPDG on the specification of high level job descriptions, and will ultimately address the issue of sending jobs to the most effective platform based upon domain specific parameters, such as lattice sizes and LQCD algorithms, as well as the more common batch criteria such as system load.

2.6 Data Grid Tools

The Software Committee is nearing completion of the basic design of the I/O routines consistent with plans for future data archives, and grid based tools. The design includes binary file formats

much like those used in the NERSC QCD archive, Metadata for physics parameters, and XML based I/O standards. We are founding members of the International Lattice Data Grid (ILDG) project, which was established to enable sharing of QCD data internationally. Members of the Software Committee, who serve on the ILDG Metadata and Middleware Groups, are working to insure consistent interfaces and common standards.

Important work remains to establish an international grid–of–grids for lattice data. ILDG has already adopted in principal the Storage Resource Management (SRM) system developed across multiple projects. JLab's participation in SRM developments was done as a part of the SciDAC Particle Physics Data Grid project, and that experience is being carried over into the LQCD Sci-DAC work.

Additional work in defining the necessary components of the ILDG grid remains, and members of the software committee are participants on the ILDG architecture working group. As for the SRM, there is agreement that it will be based upon web services, and include domain specific meta data catalogs to allow discovery of valuable data sets.

The ILDG metadata working group is in the process of defining the meta data to be archived with the data, and that meta-data specification will also form the basis for cataloging and query operations. We will undertake the development of a production data grid environment, with tools for entering, managing, and searching via meta-data. Care will be taken to guarantee interoperability with our ILDG partners.

We will also set up the first components of a data grid (storage management, file transfer) between the two cluster sites (JLab and FNAL). This effort will include implementing SRM v2.1 specification atop the sites' tertiary storage management. It will also address issues arising from FNAL's use of Kerberos. Extension of this capability to the QCDOC facilities at BNL and Columbia will follow quickly in the next stage. In addition the plan includes intelligent dispatch of jobs to all the sites within the meta center, based upon application domain parameters.

2.7 Profile and Performance Analysis Tools

Following the initial award of the SciDAC project, computer scientists in Dan Reed's group at the University of Illinois applied a high level performance analysis toolkit (SvPablo) to analyze the performance of the MILC code. The MILC code was instrumented and detailed hardware performance was captured using SvPablo's interface to the University of Tennessee PAPI hardware performance counter toolkit. Initial methods and result were presented at Supercomputing 2001. The profiling tool has been extended to the message passing API (QMP) as well.

During the renewal period propose to enhance and expand the functionality of the performance analysis tools developed to date, and extend them to support other platforms, including QCDOC, and other codes. The new version of the PAPI (v.3.0) hardware counter toolkit, which was officially released during the Supercomputing conference, in November 2003, will fully support the Pentium 4 processor. Using funds from another project, we will incorporate that PAPI version to

SvPablo. We plan to install this new, integrated SvPablo version on the FermiLab lqcd cluster, and we will extend our analysis of the MILC code on that cluster by capturing relevant performance data from hardware performance counters. Using such data, we expect to conduct optimizations on specific parts of the MILC code, guided by the detailed measurement of its computation and communication characteristics.

We also plan to apply the full-featured SvPablo toolkit to analyze the performance of the two other major QCD codes considered by the lattice gauge theory collaboration, namely the Columbia Physical System (CPS) and Croma. Similarly, we plan to port at least part of SvPablo's functionality to that platform. This will enable cross–platform comparison of the two major consortium computing platforms–commodity clusters and the QCDOC.

2.8 Algorithmic Development

If past history is a guide, new algorithms will be as important as faster hardware in advancing research in lattice gauge theory. Consequently the software infrastructure must be flexible enough to accommodate the evolution of QCD algorithms. Even on Teraflop/s or Petaflop/s platforms the central algorithmic problem faced by our field will almost certainly continue to be the performance of the Dirac inverter, which is critical for including the effects of light sea quarks. We need both improved algorithms for existing applications, as well as radically new approaches for problems outside the reach of current methods, such as simulations at finite chemical potential. These problems pose a fundamental mathematical challenge with strong relations to analogous problems in other areas of science and applied mathematics. SciDAC offers an ideal setting for this type of algorithmic research by encouraging interdisciplinary collaborations.

Brower and Rebbi, who have explored multi–grid methods extensively in the past decade, are beginning to work with applied mathematicians from the TOPS multi–grid algorithm team to explore new approaches to solve the critical problem of accelerating the inversion of the Dirac operator. David Keyes and Steve McCormick have expressed their interest and a test case using the two– dimensional Schwinger model is being pursued for an initial study of multi–grid Dirac inverters in gauge backgrounds. Also, the new SciDAC postdoctoral fellow at Boston University, Hartmut Neff, brings new expertise to the problem of projecting out low eigenvalues which are responsible for critical slowing down. This has proven to be a useful "preconditioner" for the Dirac inverter in some instances, and should be explored further in the multi–grid context.

Recently new possibilities for applying multi–level or domain decomposition methods have begun to be explored by Martin Lüscher, who has introduced a blocking method based on the "Schwarz alternating procedure" that shows some promise for improved performance on P4 clusters due to better locality for cache and communication. Possible extensions to the stochastic (or Monte Carlo) part of the problem are being explored, as well as a new multi–level approach to the full partition function (with Fermionic determinant). This approach represents a combination of domain decomposition ideas and stochastic multi–grid in the spirit of multi–grid modification of the Swendsen-Wang cluster algorithm by Achi Brandt. In short, there is a new set of very attractive ideas that certainly deserve careful examination. A joint effort is being started by Brower, Neff and Rebbi in collaboration with the multi–grid team in TOPS. If there is some initial success, we would recommend that this project be expanded in future software plans.

3 Cluster Development

By 2005, the rapid pace of processor and interconnect technology development will allow us to construct specially configured clusters for lattice QCD which will, at a scale of multiple teraflop/s, provide additional gains in price/performance. In order to deliver the highest performance, additional prototyping work must be done in 2004-2005 under this SciDAC grant so that we continue to track emerging technologies in preparation for full scale deployment. In this section we describe these near term explorations, and the optimization strategy and reference platforms for 2005 that are being targeted.

The prototypes are being used to study the capabilities and limitations of cluster systems. As the most recent example of these studies, JLab is now in the process of commissioning a novel gigabit ethernet mesh cluster which achieves higher per node bandwidth than the most common cluster interconnect (Myrinet) at a lower cost, by exploiting six gigabit ethernet links in a three–dimensional mesh configuration.

In January, 2004, approximately 80 new systems will be procured at Fermilab. The most likely purchase will be single processor Intel systems, with 800 MHz front side buses, and PCI-X provided by the new Canterwood-ES chipset. Pending the results of testing in December, the processor will be either the new "Prescott" Pentium 4 with SSE3, or the current Pentium 4 processor. These systems will replace the dual 700 MHz Pentium III nodes purchased in 2000 with pre-SciDAC supplemental DOE funds and in-kind Fermilab contributions. We will re-use the existing Myrinet 2000 fabric from that dual Pentium III cluster. This will be the first example of part of our cluster strategy - the re-use of high performance network fabrics. Due to Moore's law performance increases, after 3 years of operation the computers in a given cluster have only 25% of the performance of new machines. However, our high performance network fabrics historically have had excess bandwidth, and sufficient latency, and so perform well for five to six years. This re-use represents a substantial cost savings.

In January, hardware to investigate two alternate networking architectures will also be purchased. First, blades providing 32 ports of gigabit ethernet will be used in one of our existing Myrinet switches, in order to investigate whether single and dual gigabit connections per computer configurations are cost effective on the smaller physics runs common on our cluster, such as valence quark propagators. Second, a small Infiniband switch, approximately 24 ports, and matching host channel interfaces will be purchased. Infiniband is the most promising switched network architecture in the future, and acquisition of this small fabric will allow the collaboration to assess operational aspects of Infiniband and to port the SciDAC communications API (QMP).

A significant boost in I/O bandwidth and a reduction in communications latency is anticipated with the introduction of PCI Express in mid-2004. We are also likely to see significant cost reductions in two alternative processor architectures, Intel's IA64 (Itanium) and IBM's PPC970 (called G5 by Apple). Both of the latter have shown promise in our limited investigations. PCI Express will

not be supported on Itanium in 2004; it is not known at this time whether it will be supported in either the AMD Opteron or IBM PPC families in 2004. However, these three alternatives (AMD Opteron, IBM PPC, and Intel IA64) are of sufficient promise that we can propose at this time two significant cluster acquisitions at JLAB and FNAL in late spring and late summer of 2004. One cluster would investigate PCI Express and the most promising supporting architecture, likely dual Xeon or dual Opteron, and the other would investigate, if appropriate, another processor family (PPC, Opteron, or Itanium) and the best matching communications fabric, possibly optimized for large, single-processor clusters.

3.1 Cluster Optimizations

From the SciDAC research already completed, it is clear that there are two regimes for running clusters. For very large physics problems, which cannot fit into the aggregate cache size of the cluster CPUs, lattice algorithms are completely memory bandwidth limited. In this situation, multiple processors on a shared memory bus (as in the Xeon architecture) are not efficient (bandwidth starved), and so single processors deliver the best price performance (NUMA architecture machines, as in the Opteron, may allow efficient multi–processor nodes even for these problems). On the other hand, for smaller physics problems where cache residency is feasible, multi–processor compute nodes are more cost effective than single processor nodes (lower cost per processor). Similar optimization tradeoffs in cost and performance for network links may prove to be equally significant.

One advantage of the cluster approach is that the national lattice project can deploy multiple clusters with different optimizations, and intelligently steer specific applications onto the most cost effective platform (meta–center operations described above).

The prototyping work done under the SciDAC grant will provide the needed parameterization of clusters for lattice QCD. During 2004, the national community will develop a good understanding of the total workload for the following 2–3 years which will allow the selection of specific clusters for deployment in 2005 and beyond.

3.2 2005 Cluster Deployments

Current trends, plus the optimization possibilities described above, lead to a plan to deploy multiple clusters in 2005 exploiting different optimization strategies to most effectively cover a broad range of physics topics. In addition, clusters have a non-linear scaling term which becomes important above 1024 nodes, which similarly leads to the decision to deploy several clusters.

The exact details of the cluster procurements for 2005 cannot be given now, in that it is impossible to predict the market, and know which technologies will gain sufficient market share to profit from volume shipments (as gigabit ethernet has in the last 3 years).

Reasonable estimates based upon individual extrapolations in processor speed, cache size, memory bus speed, and I/O interconnect bandwidth and latency yield an expected price/performance of less than \$0.8/Mflop/s at a scale of several teraflops in 2005. Under an only slightly optimistic assumption that quad processor nodes will become commodity by 2005, even more favorable price/performance, as low as \$0.40/Mflop/s, can be anticipated for some significant portion of the physics program. This price performance would be achieved for problem that fit into cache (hence multi-processor is optimal) and when a high performance network is exploited (Infiniband) to produce a lattice QCD /it SuperCluster."

As a concrete example, an important physics goal in the 2005 time frame will be to calculate hadron spectroscopy and structure with chiral fermions on a sufficiently large lattice to accommodate pions as light as 220 MeV. One means to accomplish this goal that has been analyzed is using domain wall fermions on a $32^3x48x24$ lattice. Using the performance analysis described in the December 2002 proposal "Computational Resources for Lattice Gauge Theory" and conservative market assumptions, a Gigabit Ethernet cluster of 2048 single processors in 2005 would sustain 4.5 Tflop/s in single precision at a cost of \$0.68 per sustained Mflop/s. An alternative architecture (a SuperCluster) of 1048 quad processors in the same time frame is estimated to sustain 7.7 Tflop/s at a cost of \$0.41 per sustained Mflop/s. Smaller applications (lattices) would achieve similar price performance on an appropriately scaled machine (partition).

The prototype clusters planned for 2004 will be essential for accurately analyzing these performance metrics for selecting the large 2005 clusters and optimizing their architecture.

4 The QCDOC Project

A central objective of the SciDAC software effort is to provide a standardized software environment that is able to effectively support multi-teraflops hardware platforms that are highly costeffective for QCD. As has been discussed above, the first multi-teraflops production machine which our collaboration plans to construct is a 10 Tflops QCDOC machine at BNL. Thus, an important component of the software effort is directed at efficiently supporting this architecture. In this section we will describe the current status of the QCDOC project. Further details as well as pictures of the prototype hardware can be found in Appendix C.

The QCDOC architecture is based on a single-chip computational node which contains an embedded RISC 440 PowerPC processor with a 1 Gflops IEEE double precision floating point unit, 4 Mbytes of on-chip memory and communications hardware providing 0.5 Gbit/sec, bi-directional serial communication in each of the twelve directions permitting a large parallel machine to be constructed as a six-dimensional mesh. (Each node also contains an off-chip standard memory module of size that can be selected between 64 Mbytes and 2 Gbytes.)

4.1 Hardware

The design and development of the QCDOC architecture and hardware is in the final debugging stages and construction of large-scale machines is about to begin. The heart of the machine is an applications specific integrated circuit (ASIC) which is being built by IBM. The first of these components were received at the end of May. Two of these QCDOC ASICs are mounted on a daughter board. Daughter boards with mounted ASICs have been available since June. The QCDOC ASIC was tested quite thoroughly in a single-node mode during the summer. All tests to date have been successful, so that no faults have been found in this complex, 50 million transistor part. Some minor faults were found in the daughter card, but these could be fixed by rework and engineering changes have been made to the daughter board design so future production will not have these difficulties.

The next phase of testing began in early September when three motherboards were assembled. These mother boards hold thirty two daughter cards and each mounts in a single-mother board cabinet. These cabinets contain the full clock distribution and serial communication systems of the final QCDOC computer, allowing all of these systems to be tested. After overcoming some initial difficulties, we now have two mother boards which function perfectly. These have run physics code for many tens of hours both separately and joined with 2-meter cables to form a 128-node machine. The longest run to date on this 128-node machine was for 14 hours. During this run no communications errors were found establishing a bit error rate of less than one in 10¹⁸. We expect to accept the ASIC as ready for large-scale production after performing additional long runs with a number of physics programs and comparing with workstation results.

4.2 Software

As has been described earlier, software development for QCDOC has been a major priority of the U.S. Lattice QCD SciDAC effort as well as the other collaborating groups at the RIKEN BNL Research Center and in the UKQCD. High-performance QCD kernels have been tested on a two-node system using 1-dimensional communication and show efficiencies as high as 50%, as were seen using the QCDOC simulator. While the QCDOC operating system is still being developed much progress has been made, permitting the compilation and single-node execution of both the entire Columbia Physics System (CPS) and MILC code.

4.3 QCDOC construction

The schedule for completion of the QCDOC project is shown in the Gantt chart in Figure 3. The QCDOC computer construction proceeds in two stages. Beginning in November we will construct the first 2000 nodes of the 10 Tflops machine, (as well as similar initial large-scale hardware for the RIKEN BNL Research Center and the UKQCD Collaboration). This partial machine will be made available for use by the U.S. collaboration as rapidly as possible. This will permit the collaboration to begin physics production running for two or three projects, to be determined by our allocation

procedure. It will permit the BNL/Columbia team a test run at providing user support for this scale of use. Production computer code from a number of groups within our collaboration will be used in a real environment. By making initial large-scale hardware available as quickly as possible, we will insure that the full machine can be used effectively as soon as it becomes available.



Figure 3: Gantt chart showing the QCDOC construction schedule. The 8-mother board machine is the final electronic prototyping step verifying a multi-mother board backplane.

As soon as this stage has been successfully brought up, we will begin full-scale production of the remaining 20,000 nodes which will be assembled at Brookhaven. While this construction is scheduled to begin in January of 2004, long lead-time parts will be ordered earlier, as soon as required by the construction schedule and validated by the present prototype testing.

The construction of this 10 Tflops machine will be completed by July of 2004 at which time fullscale physics production running will commence.

5 Management

Overall responsibility for this project is vested in the Lattice QCD Executive Committee: Richard Brower (Boston U.), Norman Christ (Columbia U.), Michael Creutz (BNL), Paul Mackenzie (Fermilab), John Negele (MIT), Claudio Rebbi (Boston U.), Stephen Sharpe (U. Washington), Robert Sugar (UCSB, Chair) and Chip Watson (JLab). The Executive Committee sets the project's goals, draws up plans for meeting these goals, and oversees progress towards meeting them. The Executive Committee has been carrying out these functions for over four years. It holds approximately two conference calls per month, and communicates via email between calls. A consensus has been reached on nearly all issues that have come before the Executive Committee. When consensus is not reached, decisions are made by majority vote, with the Chair's vote deciding the outcome in case of a tie. The Chair of the Executive Committee, Robert Sugar, serves as spokesperson and principal contact with the Department of Energy. Each institution receiving funds under this project has a principal investigator who has first level responsibility for work performed at his institution.

The Executive Committee has formed a number of committees to assist it in managing the project:

Scientific Program Committee: The Scientific Program Committee monitors the scientific progress of the project, and provides leadership in setting new directions. It solicits proposals for use of the

computational resources available to the collaboration, and allocates time on them in a fashion to achieve the greatest scientific benefit. At present, these resources consist of the FNAL and JLab clusters, and an allocation at Oak Ridge National Laboratory provided through the SciDAC Program. The Committee organizes an annual meeting of all lattice gauge theorists working on or planning to participate in the project order to review progress and plan future directions. Members of the Scientific Program Committee are Peter Lepage (Cornell U.), Robert Mawhinney (Columbia U.), Colin Morningstar (Carnegie Mellon U.), John Negele (MIT), Claudio Rebbi (Boston U., Chair), Stephen Sharpe (U. of Washington), Doug Toussaint (U. of Arizona) and Frank Wilczek (MIT).

Oversight Committee: The Oversight Committee is charged with reviewing progress in implementing the plans of the collaboration, reviewing plans for the development and acquisition of software and hardware, and making recommendations regarding alternative approaches or new directions for the collaboration. It meets via conference calls, which are scheduled so that the Committee can review on–going progress and planning, and provide timely advice before important implementation or procurement decisions are taken. The Chair of the Executive Committee participates in these conference calls to obtain the advice of the Oversight Committee at first hand, and the Software Coordinator and hardware developers participate as needed. The Chair of the Oversight Committee, Steven Gottlieb, maintains regular contact with all aspects of the project, to keep the Committee informed with developments, and to schedule meetings appropriately. The members of the Oversight Committee are Steven Gottlieb (Indiana U., Chair), Anna Hasenfratz (U. of Colorado), Greg Kilcup (Ohio State U.), Julius Kuti (UC San Diego), Rob Pennington (National Center for Supercomputer Applications), Ralph Roskies (Pittsburgh Supercomputer Center) and Terry Schalk (UC Santa Cruz).

Software Coordinator and Software Coordinating Committee: The Software Coordinator, Richard Brower, supervises the work of all software development teams, providing direction and coherence to the effort. Expanding on our original SciDAC proposal, he has developed a detailed set of tasks and milestones, which he monitors. He provides quarterly progress reports for the Executive Committee on the progress of the software effort. The Software Coordinator has set up a website, http://physics.bu.edu/~brower, on which all agenda, minutes and working documents of the Software Coordinating Committee are posted, and he has also established a mail archive, (qcdapi@physics.bu.edu), for interchange of information among all members of the collaboration.

The Software Coordinating Committee works with the Software Coordinator to provide overall leadership of the software effort. Its members are Richard Brower (Boston U., Chair), Carleton DeTar (U. of Utah), Robert Edwards (JLAB), Donald Holmgren (FNAL), Robert Mawhinney (Columbia U.), Celso Mendes (U. of Illinois), and Chip Watson (JLAB). It took the lead in designing the QCD applications interface and is overseeing its implementation on the two computing platforms targeted in this project. The Committee holds regular conference calls, and meets in person several times per year.

6 Budget

The overall budget for the two year renewal period is summarized in Table 2. Detailed budgets for the institutions which are to receive funds, along with the description of the work these budgets will support, can be found in the separate budget sheets. The funds support a total of 11.66 FTE, of which 9.36 FTE is devoted to the software effort, and the remainder to the evaluation of cluster components, and to the design, procurement and evaluation of clusters. It should be noted that a number of people who do not receive support from the grant make major contributions to the effort. The hardware funds are to continue our program of constructing prototype clusters which are used to test the software, and to determine optimum parameters for the terascale production clusters we propose to build in 2005.

Personnel Budgets			
Institution	FY04	FY05	
BNL	340	350	
Boston U.	205	169	
FNAL	417	430	
Indiana U.	56	58	
JLab	437	450	
MIT	265	217	
U. Arizona	49	49	
U. Illinois	175	181	
UC Santa Barbara	19	19	
U. Utah	51	52	
Total Personnel	2014	1975	
Hardware Budgets			
Institution	FY04	FY05	
FNAL	150	255	
JLab	130	235	
Total Hardware	280	490	
Total	2294	2465	

Table 2: Budgets for personnel and hardware in \$1,000. The personnel budgets support 9.36 FTE for the software effort and 2.30 for the evaluation, design and procurement of clusters and their components.

A Report of the Lattice Gauge Computing Review Panel

We were charged with assessing the proposals under three heads: intrinsic scientific merit, strength and significance as computer science, competitive position with respect to world activity. We were asked to consider each of these matters from a broad strategic perspective, and finally to make appropriate recommendations.

A.1 Scientific Merit:

The standard model provides a remarkably economical description of our current knowledge of the fundamental laws of physics. It is a great achievement, but no one believes that the standard model is a complete description of Nature, and there are several well-motivated theoretical proposals that suggest the existence of quantitatively small yet profoundly meaningful deviations from its predictions. Such deviations would be similar in character to the minute bending of light that provided crucial evidence for Einstein's general theory of relativity.

The search for deviations from the standard model is a major focus of experimental investigation in physical science. It drives the construction of powerful accelerators, intricate detectors, and sophisticated tools for data analysis, all at the frontier of human ingenuity. Many hundreds of millions of dollars per annum are invested in these activities. Yet in many cases interpretation of the experimental results is limited by the accuracy with which we can compute the consequences of the standard model. For although the equations of the standard model are quite precisely defined, it can be extremely difficult to solve them to an accuracy that does justice to what can be achieved experimentally. If we are to recognize small deviations from the standard model we must have accurate knowledge of what it predicts.

The only way physicists know to do the required calculations is to bring the full resources of modern computers to bear, using the methods of lattice gauge theory. Existing work allows one to estimate with considerable confidence what accuracy can be attained, as a function of available computer power. Several specific cases have been identified where a few teraflop-years will add significant value to completed or ongoing experimental projects. Many more become accessible at the level of tens of teraflop-years and beyond.

Besides the service they provide to searches for essentially new phenomena, calculations in lattice gauge theory are valuable in advancing our understanding of quantum chromodynamics (QCD) itself. QCD is a remarkably beautiful and successful theory. It seems certain to be the foundation of our understanding of the strong interaction, including the internal structure of protons and neutrons and the origin of nuclear forces, for the foreseeable future. But again, because of our limited ability to calculate, we have not yet fully exploited the potential of the theory to give insight into the internal structure of nucleons, the nature of nuclear forces, and the properties of particles containing heavy quarks. Computational work along these lines will enhance ongoing experimental programs at Jefferson Lab, Fermilab, SLAC, CLEO, and elsewhere around the world. Such work is also vital for justifying confidence in the claimed precision and accuracy of lattice gauge theory techniques, thorough validation in applications where the underlying fundamental physics is not in

question.

Finally, there are important potential applications of QCD in cosmology and astrophysics that require understanding the behavior of matter under conditions that are impractical to duplicate in a laboratory setting. Computer simulations, of course, are not so limited. The ultra-high temperature limit of QCD is used to model matter during most of the crucial first minutes of the big bang. Lattice gauge theory calculations have already made a major impact in this field. They suggested when as ordinary (hadronic) matter is heated it ionizes into quark-gluon plasma at a surprisingly low temperature. This prediction has now been broadly confirmed at the Brookhaven Relativistic Heavy Ion Collider (RHIC), and a fertile new field of extreme nuclear physics has opened up. The regime of ultra-high density is important for the description of neutron stars and supernovae. There is a beautiful analytic, semi-quantitative theory that predicts the behavior at "asymptotically" large densities, but we need much more comprehensive and accurate information to do justice to the astrophysics, and at present lattice gauge theory is our best long-term hope.

In short, we feel the scientific merit of suggested program is very clearly outstanding.

A.2 Strength and Significance of Computer Science:

The proposed programs are of considerable interest from the point of view of computational science, since they could provide convincing models and demonstrations of the use of cost effective special architectures for scientific problems. Indeed, development work on the QCDOC project has already influenced the architecture of the IBM BlueGene/L supercomputer project.

It is proposed to pursue two separate hardware tracks, one using specially designed systems-on-achip that leverage industrial intellectual property cores, the other using general-purpose computing systems. A specially designed system appears to offer substantial cost savings in the near term, but the cost effectiveness of general purpose computing systems may improve in the intermediate term. These two approaches are subject to quite different risks and opportunities. General purpose computing systems profit enormously from economies of scale in production, but of course they will not be designed with the needs of science or lattice gauge theory in mind, and their future development will depend on market forces that are difficult to anticipate. We therefore feel it is prudent, as well as interesting, to pursue both tracks, at least until a clear winner or a synthesis emerges.

The software development component of the proposal is also novel in this context and extremely important. In order to deliver on the scientific promise of the proposal, a much larger community than those actively engaged in the initial development will need to be engaged. This mandates implementation of interfaces using standard programming languages at the level of C/C++ and a plain- vanilla UNIX-based operating system at the earliest appropriate stage, with maximal transparency to the hardware. There must also be an adequate library of optimized functions for common computational tasks in lattice gauge theory, and adequate protocols for testing and validation of new contributions. This must all be well documented, in a form that will be accessible even to members of other scientific communities, such as condensed matter theorists or specialists in statistical mechanics, who attack mathematically similar problems. The proposers appear to be

well aware of the importance of the software component of the project, and work along these lines is already proceeding. The pursuit of two separate hardware tracks will aid in the development of robust, portable software. If successful, the software component could be very valuable both in itself and as a model for other scientific enterprises.

A.3 Competitive Position:

Since lattice gauge theory is in some ways a mature field, with established procedures and standards, it is not difficult to compare the relative power of different computational facilities. Sustained operation at the multi-teraflop level is both necessary and sufficient for the U.S. effort to match existing European and Japanese initiatives in the immediate future. A sustained program as outlined in the proposal should allow the U.S. to compete very successfully in hardware within a 2-4 year time frame.

We anticipate that the "open" software model will engage interest beyond the traditional lattice gauge theory community, broaden the scientific development base, and give a major additional edge.

We commend and encourage the remarkable collaboration in this project between industry leaders, specifically IBM, and the university and national laboratory scientific communities. Given the world leadership role of the U.S. computer industry, this is another great source of competitive strength.

A.4 Recommendations:

Several specific recommendations are embedded in the preceding sections. We will not repeat these, but will close with a few suggestions of a general nature.

The DOE could leverage the excellent scientific potential of this endeavor through a program of fellowships that would allow young people entering the field, which requires a significant start-up time and falls somewhat between traditional academic categories, some measure of freedom and security during the early, precarious parts of their careers.

The proposers should not be overly conservative in their funding requests. In particular, there should be realistic allowance for contingency costs, and adequate support staffing so that skilled physicists and computer scientists can employ their time efficiently.

Both the proposers and the DOE should recognize that this is an endeavor that is not likely to be exhausted in 4 years or even in 10. While the focus at this early stage is, quite properly, to get significant hardware and software up and running, it will be appropriate to keep the longer term in mind, and to begin serious long-term planning perhaps 2 years into the project, when enough practical experience will have been accumulated.

B Appendix: Cluster Hardware and Performance

This Appendix describes the prototype cluster hardware acquired during the initial SciDAC grant, summarizes its present performance on QCD applications, and describes performance analyses and projections for clusters to be built during the SciDAC renewal.

B.1 Current Clusters

Building on experience from pre-SciDAC clusters at Fermilab, JLab, and MIT, a sequence of four prototype clusters have been built at Fermilab and JLab:

- 48 node dual 2.0 GHz P4 Myrinet cluster at Fermilab, beginning operation in August 2002.
- 128 node single 2.0 GHz P4 Myrinet cluster at JLab, beginning operation in September 2002.
- 128 node dual 2.4 GHz P4 Myrinet cluster at Fermilab, beginning operation in January, 2003.
- 256 node single 2.66 GHz P4 Gigabit Ethernet mesh cluster at JLab, beginning operation in September 2003.

In our coordinated program to broadly explore architecture and technology options, the acquisitions have alternated between Fermilab and Jlab, and focused on complementary issues. Fermilab has concentrated on dual-processor nodes, which are most cost effective for cache-resident problems, and JLab has focused on single processor nodes and most recently, exploitation of Gigabit Ethernet, which are most cost effective for large applications that do not fit in cache. Figure 4 Shows the most recent clusters at Fermilab and JLab.

B.2 Performance

Cluster performance will ultimately be measured by the sustained performance of the SciDAC test suite of physics applications on full scale multi–Tflop/s clusters. In the prototype phase, our strategy has been to use the cluster performance model described in Section 3 of the December 2002 proposal *Computational Resources for Lattice Gauge Theory*, to demonstrate its agreement with measurements of QCD physics code running on current clusters, and to use the model to estimate performance of future clusters based on the announced or estimated parameters of improved commodity components.

As a result of software development supported by the original SciDAC grant, Initial Level 2 QCD software applications are beginning to become available for production code. Figure 5 shows recent measurements of the performance of the Wilson inverter as a function of cluster size for 8^4 sites per node with first generation optimized code developed at JLab. In this measurement,



Figure 4: Most recent SciDAC clusters. Above: 128 node dual 2.4GHz P4 Myrinet cluster, commissioned at Fermilab in January 2003 . Below: 256 node single 2.66 GHz P4 Gigabit Ethernet cluster, commissioned at JLab in September, 2003.

the calculation was run on a single node with no communications, and then communications were successively turned on in one, two, and three dimensional tori as listed in the caption. Note that adding new dimensions of communications are the principal origin of the performance decrease, and that the actual increase in cluster size has a negligible effect on the global sums.

It is important to note that this is still an early stage of optimization, and additional cluster software development is being carried out as described in the text. One example is implementation of domain wall fermions, another SciDAC metric test. Whereas straightforward implementation of the domain wall Dirac operator with SSE instructions yields 804 Mflops on a single node for 8⁴ per site, software effort at MIT changing the SSE implementation to minimize memory traffic increases this performance to 1585 Mflops. Thus, it is clear that the initial level 2 implementation has not yet achieved its full potential, and that level 3 implementation will provide further opportunity for optimization.

Another important cluster accomplishment has been the parallel development at Fermilab of opti-



Figure 5: Initial performance measured on JLab GigE cluster for the Wilson inverter as a function of cluster size for 8^4 sites per node. The four points represent no communications, 1-D ring, 2-D 8×8 torus, and 3-D $4 \times 4 \times 8$ torus. The 3-D point corresponds to \$2.75 per sustained Mflops. The fall in performance is almost entirely due to turning on additional dimensions of communication.

mized code for improved staggered fermions. This application is typically more computationally demanding than Wilson or domain wall fermions, and an initial level 2 optimized inverter for improved staggered fermions runs at 752 Mflop/s/node, corresponding to \$4.12 per sustained Mflops, for a 14⁴ lattice on the Fermilab 128-node Xeon cluster. Figure 6 shows the evolution of performance of MILC staggered fermion code on Intel based clusters at Fermilab and elsewhere, and the projected performance for 2004. The late 2002 point is the 128-node Xeon cluster result noted above. The results shown are for non-cache resident lattices on the full system, one of the most demanding computational tasks. Fermilab's current job mix, calculating valence quark propagators on small numbers of nodes, delivers 30% better performance than the point shown. Wilson and domain wall fermions are less demanding and perform even better than the point shown.

Table 3 summarizes performance on the two JLab clusters, where the initial level 2 Wilson inverter has been benchmarked, and shows projections for future clusters planned for the period of the SciDAC renewal. Measured performance for the Wilson inverter of 619 Mflops on an 8⁴ lattice on the JLab Myrinet cluster corresponds to \$4.62 /Mflops and 703 Mflops on the GigE cluster corresponds to \$2.75 /Mflops. These initial results make us confident that fully optimized level 3 inverters on these machines will achieve the indicated targets of \$3 and \$2/Mflops respectively. On the basis of the performance model and the commodity components that are becoming available as described in the text, we expect that half Teraflops prototype machines supported by SciDAC in 2004 will reach \$1/ Mflops, and that full scale 4-8 Tflops production machines will be below \$1/ Mflops in 2005.



Figure 6: Price/performance of staggered fermion inverter on Intel clusters as a function of installation date. The measurements are for full clusters, running two processes per node.

Year	Cluster	TFlops	\$/Mflop/s	\$/Mflop/s
		sustained	expected Level 3	current Level 3
2002	128 P4 Myrinet	0.1	3	4.62
2003	256 P4 GigE	0.25	2	2.75
2004 Q3	Jlab, TBD	0.5	1	-
2004 Q4	FNAL, TBD	0.5	1	-
2005	FNAL, Jlab, TBD	4-8	< 0.6	-

Table 3: Current and planned SciDAC prototype clusters. The last column shows the cost per sustained Mflop/s for the Wilson fermion inverter with an initial implementation of level 2 QDP code. The preceding column shows performance expected with fully optimized software.

To identify the optimal commodity components for future machines, Fermilab has undertaken extensive testing of newly available technology, and results are available on the web page http://lqcd.fnal.gov/benchmarks/. Among the processors tested in addition to the Pentium 4, are the Opteron, Itanium-2 and G5. Because of the high visibility of the new G5 cluster at Virginia Tech, we show in Fig. 7 a comparison of the performance of a 2.0 GHz G5 with the 2.8 GHz Pentium 4 discussed above. Subsequent analysis has shown that although the G5 has many attractive features, its price performance for lattice QCD appears to be a factor of two less favorable than the Pentium 4.



Figure 7: Performance tests of two processors analyzed for the next Fermilab cluster, 2.8 GHz Pentium 4 with 800MHz front-side bus, and 2.0 GHz G5

C QCDOC Hardware Status

Figure 8 shows a picture of one of the first daughter boards and Fig. 9 a daughter board in a testjig. These testjigs were assembled by hand at Brookhaven. While they provided a critical opportunity for the initial testing of the ASIC, the configuration of the Ethernet wiring was not sufficiently electrically solid to permit an exhaustive test of the on-board Ethernet.

Figure 10 shows a fully-populated mother board while Figure 11 show a back-side view of the cabinet which holds a single mother board. The black cables are 1-meter examples of the cables that will interconnect the large machine. Here the cables provide a loop-back function allowing off-board communication to be routed back to this same mother board. Each cable contains 16 pairs so the 24 cable connections to a single mother board represent 384 pairs or 768 signals.

The first emphasis in our mother board testing has been the Ethernet system which has given more trouble than expected. Both the Ethernet repeater and physical layer interface chips that we are using have been quite sensitive to noise and difficult to configure. Careful adjustment of the clock signal levels and the drive strength of the repeater chips has been required. In addition, an extra EEPROM has been installed to permit a power-on configuration of the repeater chips that is more noise-immune than the default power-on state. It appears that these changes have produced a reliable system but more testing is needed and under way.

Possibly even more important than the Ethernet system is our custom serial communication network. Initial investigation also showed unreliable behavior of this system. Here the problem was traced to excessive jitter in the output serial data. This was caused by an IBM-recommended power filter circuit for the ASIC phase-locked-loops going into oscillation. This has been fixed and now



Figure 8: A daughter board which contains two independent QCDOC nodes.



Figure 9: A test jig holding one of the QCDOC daughter cards.



Figure 10: A fully-populated mother board holding 32, 2-node daughter cards.



Figure 11: Rear view of a single mother board cabinet. The 24 black cables connect the mother board back to itself as a 2^6 torus.



Figure 12: The serial data signal as seen on the pins of the receiving ASIC after passage through a six-meter cable. Both the amplitude and phase definition are excellent suggesting that the simple design of the serial communications system has adequate margin for very reliable behavior.

the jitter is at the design specification and the data signal strength and timing much better than required. Figure 12 shows the excellent signal present on the receiver input pins after passing through a six-meter cable. With this jitter problem corrected, the serial network has functioned perfectly. The single-bit error recovery feature was well tested (and verified) when the large jitter was present. With the jitter problem fixed we see no communication errors at all.

To our dismay, one of the 10 DC-to-DC voltage converters on one of the mother boards failed after about one month of testing and damaged both the mother board and six daughter cards. The apparent over-voltage condition caused by this failure is supposed to be impossible and Power-One, the manufacturer, is actively studying the problem. Fortunately, our two remaining mother boards have been sufficient for the needed testing of the ASIC and two additional mother boards will be available in the beginning of December to support the software development work of a larger fraction of the collaboration.