Computational Resources 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 17, 2002

Abstract

We propose to take the next major step in our initiative to construct the computational infrastructure needed by the U.S. lattice gauge theory community for large scale numerical studies of quantum chromodynamics (QCD), the fundamental theory governing the strong interactions. In particular, we request funds to construct and operate a specially designed computer, the QCDOC, capable of sustaining 1.5 terasflops for the study of QCD. The QCDOC, like all hardware constructed in this initiative, will be made available to the entire U.S. lattice community. We also request funds to support the additional personnel made necessary by the enhanced scope of our work during the 2003 calendar year. We are currently developing the software needed for terascale simulations of QCD, and building and testing prototype clusters optimized for QCD under a SciDAC grant. We describe all of the work planned for 2003 in this proposal. This work constitutes an essential step towards the multi–teraflops facilities we plan to construct in 2004 and beyond.

Contents

1 Introduction **1**

2 The QCDOC **4**

2.1 Overview 4

2.1.1 Background 4

2.1.2 Motivation 6

2.1.3 Specific objectives 6

2.2 QCDOC ASIC Design 8

2.3 Construction and testing of the QCDOC prototype 12

2.4 Construction and testing of the 1.5 Tflops machine 16

2.5 Software specific to the QCDOC 19

2.5.1 Overview of QOS 20

2.5.2 QOS Status and Schedule 20

2.6 Operation of the 1.5 Tflops machine 22

2.7 Budget 23

3 Commodity Clusters **23**

3.1 Architectural Approach 24

3.2 Overview 24

3.2.1 Cluster Experience 26

3.3 Myrinet Clusters 26

3.3.1 48 Node Dual 2.0 GHz P4 Cluster 27

3.3.2 128 Node Single 2.0 GHz P4 Cluster 27

3.3.3 Performance of current clusters 28

3.3.4 Planned 128 Node Dual 2.2 GHz P4 Cluster 28

3.4 Network Research and Development 28

3.4.1 Gigabit Ethernet 29

3.4.2 Gigabit Ethernet Cluster 34

3.5 FPGA based Network Interface 35

3.6 Next Generation Clusters 35

3.6.1 Summer 2003 Cluster . . . 36
3.6.2 Fall 2003 Cluster . . . 36
3.7 Operating and Batch Systems . . . 36
3.7.1 Cluster Facilities Administration and Operation . . . 37

4 SciDAC Software Infrastructure Project 38
4.1 Overview . . . 38
4.2 Integration of software and hardware development . . . 39
4.2.1 Network Communication . . . 40
4.2.2 Optimization of Linear Algebra . . . 41
4.2.3 Data Parallel Framework and Porting Application Codes . . . 42
4.2.4 Data Management . . . 43
4.2.5 Execution Environment . . . 43
4.2.6 Summary of Software Infrastructure subtasks through FY03 . . . 44
4.3 Tests of Software Development and Final Evaluation . . . 45
4.3.1 Software Development Tests . . . 46
4.3.2 Final Milestones . . . 47

5 Management 47
5.1 Project Manager . . . 49
5.2 QCDOC Project Management . . . 50
5.2.1 QCDOC design and development . . . 50
5.2.2 QCDOC 1.5 Teraflops user facility . . . 50
5.3 Cluster Management . . . 50

6 Budget 51

A Physics Goals and Required Computational Resources 53
A.1 Precision Testing of the Standard Model . . . 53
A.2 The Quark Gluon Plasma . . . 57
A.3 Structure and Interactions of Hadrons . . . 59

B QCDOC Architecture 65

B.1 Overview of ASIC components 66
B.2 Prefetching EDRAM Controller (PEC) 69
B.3 Serial Communications Unit (SCU) 70

C Senior Personnel **73**

1 Introduction

The goals of our research are to understand the physical phenomena encompassed by quantum chromodynamics (QCD) 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 the 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 teraflops are needed to achieve our near term scientific goals. Most members of the United States lattice gauge theory community have been working together for several years to develop unified plans for creating such facilities. The hardware we propose will consist of a mix of the QCDOC, a specially designed computer that combines computation and communication capabilities on a single chip; and commodity clusters, optimized for the study of QCD. The software infrastructure needed to achieve very high efficiency on these platforms is under development with support from the Department of Energy's SciDAC program.

In this proposal we request fiscal year 2003 funds to build a 1.5 teraflops sustained QCDOC. This computer will be located at Columbia University, and, like all hardware constructed in this project, will be available for use by the entire U.S. lattice gauge theory community. In order to place this proposal in context, we provide an overview of our plans and set out in detail all of the work to be done in 2003. A Gantt chart showing the timelines for our work is given below.

Major goals of the DOE's experimental program in high energy and nuclear physics are to: 1) verify the the Standard Model of High Energy Physics, or discover its limits, 2) determine the properties of hadronic matter under extreme conditions, and 3) understand the structure of nucleons and other hadrons. Lattice QCD calculations are essential to research in all of these areas. The terascale computing facilities we propose are required if these calculations are to reach the needed accuracy in step with the experiments they support. Our scientific goals were set out in detail in the strategic plan for the development of computational infrastructure that we submitted to the Department of Energy in the Spring of 2002. In Appendix A of this proposal we provide an updated description of our scientific objectives, identify projects that will receive high priority for early use of the proposed facilities, and estimate the resources that will be required to make substantial progress on these projects. As is discussed in the Appendix, the advent of terascale computing facilities coupled with experiments currently in progress, offers the prospect of major advances in our understanding of QCD.

The size of the computing resources we seek to construct is driven by our scientific objectives. It is clear from the discussion in Appendix A that in order to provide support to the experimental programs in high energy and nuclear physics in a timely fashion, and to keep pace with the ambitious plans of our colleagues in Europe and Japan, the U.S. lattice QCD community requires computing resources capable of sustaining tens of teraflops within the next few years. Resources of this magnitude are not currently available at national supercomputer centers, and would be extremely expensive to provide on general purpose machines in the required time frame. However, by taking advantage of simplifying features of lattice QCD calculations, such as regular grids and uniform, predictable communications between processors, it is possible to construct computers for lattice QCD that are far more cost effective than general purpose supercomputers, which must perform well for a wide variety of problems including those requiring irregular or adaptive grids, non-uniform communication patterns, and massive input/output capabilities. In addition, lattice gauge theory calculations require significantly less memory than most large scale applications,

which also serves to reduce the cost of computers dedicated to this field relative to those that must serve a broad range of disciplines.

We have identified two computer architectures that promise to meet the needs of lattice QCD. One is the QCDOC, the latest generation of highly successful Columbia/Riken/BNL special purpose computers, which is being developed at Columbia University in partnership with IBM. The other consisits of commodity clusters, which are being specially optimized for lattice QCD at Fermi National Accelerator Laboratory (FNAL) and Thomas Jefferson National Accelerator Facility (JLab). This two track approach will position us to exploit future technological advances, and enable us to retain the flexibility to invest in the hardware that will maximize the scientific output at each stage of the project. Furthermore, it ensures a robust national research effort in the face of unforseeable circumstances in either track. Each architecture has its own strengths, and either may prove optimal for different aspects of our work. The QCDOC project is expected to provide very powerful computing platforms within the coming year at a cost of approximately $1 per sustained Mflops. This platform is likely to be particularly effective in generating the computationally expensive dynamical quark lattices that are crucial for our research. The clusters will allow us to take advantage of the rapid advances in the commodity computing market to build increasingly cost-effective machines over time. The well developed software packages and flexible communications systems of clusters make them particularly advantageous for more complex physics applications and for the development of new algorithms and computational techniques crucial for the advancement of our field. During 2003 and 2004 we will focus on the QCDOC, because it is expected to provide the most capable and cost-effective hardware during this period. Extrapolations indicate that clusters may surpass the QCDOC in cost effectiveness thereafter, so we anticipate switching our focus to them for 2005 and 2006.

The concept of a topical computing facility, as set out in the Office of Science's computing plan, is particularly well suited for lattice QCD. We propose to construct a distributed facility with major hardware located at BNL, FNAL and JLab. Initially BNL will focus on the QCDOC, while FNAL and JLab will concentrate on clusters. This approach will allow us to take advantage of the very considerable expertise at each of the participating laboratories, while building platforms of the appropriate size to meet our research objectives. We plan to build the initial 1.5 Tflops QCDOC at Columbia University in early 2003 to enable the machine designers to construct and debug it in house. We propose to construct a larger QCDOC at BNL in 2004. The development work on the QCDOC, and plans for the construction and operation of the 1.5 Tflops machine are described in Section 2. The construction and operation of prototype clusters, and our plans for building terascale clusters in the future are described in Section 3.

The construction and maintenance of computational infrastructure for lattice QCD will require a sustained, long term effort. Development work for the QCDOC is being funded through the HEP base program, the RIKEN Laboratory of Japan, and the UKQCD lattice gauge theory collaboration of Britain. The corresponding work for the clusters is funded through our SciDAC grant. The SciDAC grant provides funds to develop software that will enable the U.S. lattice QCD community to exploit both types of computers productively. The software effort is described in Section 4, with special emphasis on the work to be carried out in 2003, and the relationship between this work and the hardware being developed during the same period. The testing and evaluation of hardware and software will be critical for this project. The software is being validated and benchmarked as it is produced. Before building any new, large computer platform, we will construct a smaller prototype to evaluate suitability for our research, and to determine the performance of production codes.

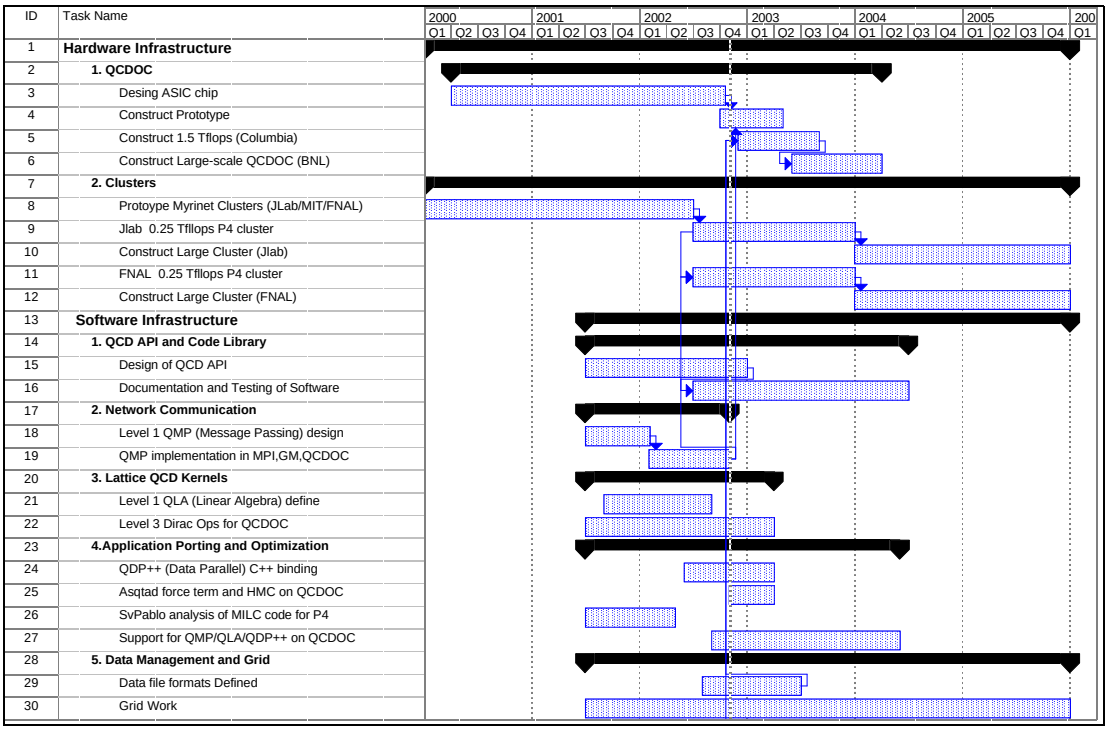

Figure 1: Overall schedule for the project

We will select machines (and their configurations) to maximize physics production, and will target software developments to areas with the greatest potential for further performance gains. Our plans for testing and evaluating hardware and software are set out in Section 4.3. The project to construct, staff and operate a distributed computing facility for lattice QCD will be managed under the administrative structure created for our current SciDAC grant. The Executive Committee will have overall responsibility for the project. The Scientific Program Committee will monitor the scientific progress of the project, and provide leadership in setting new directions. It will allocate the resources on all hardware built in this project. The Oversight Committee will review progress in implementing the plans of the collaboration, assist in the evaluation of hardware and software, review plans for new acquisitions, and make recommendations regarding alternative approaches or new directions. Members of the committees are listed in Section 5, where project management is discussed in greater detail. Virtually every senior member of the U. S. lattice gauge theory community is participating in this project at some level, as are a number of computer scientists and computer engineers. The senior members of the collaboration are listed in Appendix C.

Although we describe the full scope of our activities in this proposal, we only request funding in it for the construction of the 1.5 Tflops QCDOC, and additional personnel, not covered by our SciDAC grant, but necessary due to the increased scope of our project. We discuss the budget in Section 6.

Our project is a specific realization of the strategy for U.S. leadership in computational science set out by Argonne and Lawrence Berkeley National Laboratories in their proposal *Creating Science–Driven Computer Architecture: A New Path to Scientific Leadership*. In this proposal ANL and

LBNL emphasize the advantages of optimizing the architectures of high performance computers for scientific research in general, just as we have emphasized the advantages of doing so for research in lattice QCD. We participated in the preparation of the ANL/LBNL proposal, and we plan to work with these laboratories and their other collaborators on this broader project. In particular, the QCDOC is on the direct design path of the IBM Blue Gene family of computers, which play a major role in the ANL/LBNL plans. We will make QCDOC hardware available for testing and benchmarking by the architecture development team being assembled by ANL and LBNL, enabling it to test this architecture before Blue Gene hardware becomes available. We will work together on operating systems and other issues, so that the ANL/LBNL project can gain from the experience we have with the QCDOC, while we benefit from their great expertise in computer science. In addition, we plan to participate in the hardware evaluation efforts proposed by ANL and LBNL. We are in a particularly good position to do so because members of our community have experience with a wide range of computer architectures, have usually been among the first to develop code for new ones, and have extensive experience in testing and benchmarking. Thus, we believe that we can contribute to, and benefit greatly from being part of a broad DOE initiative.

2 The QCDOC

2.1 Overview

2.1.1 Background

As has been discussed above, the regular character of lattice QCD calculations can be exploited to achieve very significant cost/performance advantages. While the most demanding calculations easily require thousands of closely-coupled Gigaflops-scale processors, the underlying nearest-neighbor communication pattern allows the use of a mesh-style network architecture. Such mesh networks can provide both low latency and high bandwidth at very reasonable cost. By exploiting the present capabilities of commercial chip design, it is possible to fabricate an individual computing node which includes this network electronics and is contained in a single chip. It is this combination of processor, network and memory on a single chip which lies at the heart of the QCDOC (Quantum Chromodynamics on a Chip) architecture.

Targeting a cost per sustained performance of $1/megaflops, the QCDOC design exploits a single processing node which occupies less than 10 in^3 (including up to 2 GBytes of off-chip memory) and consumes roughly 5 watts. This combination of price, power and packaging permits the economical construction of the very large, 20K processor, 10 Teraflops machines required for significant advances on many of the critical physics topics addressed by this proposal.

The group centered at Columbia has pioneered the design and construction of special purpose machines for the study of QCD. The present QCDSP machines at Columbia and Brookhaven have provided a sustained 300 Gflops for lattice QCD calculations for the past four years and even now represent one of the two largest facilities for lattice QCD simulations internationally. (The QCDSP machine at BNL won the Gordon Bell prize for price performance at the November 1998 Supercomputing Conference, SC98.) With substantial funding from the U.S. DOE, the RIKEN Laboratory of Japan and the UKQCD collaboration, the design of this next QCDOC machine, begun in the fall of 1999, is now nearing completion with first prototype chips expected in February 2003.

This design work represents a very fruitful collaboration between Columbia, RIKEN, UKQCD and the IBM corporation which is exploiting this architecture for its own future research machines (`http://www.research.ibm.com/bluegene/BG_External_Presentation_January_2002.pdf`, page 6).

The QCDOC architecture is a natural evolution of that used in the QCDSP machines. Individual processing nodes are PowerPC-based and interconnected in a 6-dimensional mesh with the topology of a torus. By using only four of the six dimensions provided by the network, it will be possible to partition a single large QCDOC machine into independent sub-machines, should it be desired to run separate jobs on smaller geometries. A second, Ethernet-based network provides booting and diagnostic capability, as well as more general I/O. This Ethernet-based network permits each node in the machine to be independently addressed allowing versatile access to/from each node and the use of a powerful, multi-node source-level debugger. The entire computer will be packaged in a style that provides good temperature control, a small footprint and easy accessibility. Central to this design is IBM's technology which makes possible the high-density, low-power combination of an industry-standard RISC processor with 64-bit floating point, embedded DRAM, 500 MHz communications and the wide array of pre-designed functions needed to assemble the complete, functioning unit.

This carefully balanced architecture permits demanding, full QCD calculations to be distributed over many processors, providing the network bandwidth required for the resulting, large surface to volume ratios. Since the logical design is now complete, the performance of real code can be demonstrated with single-cycle-accurate simulation. The 500 MHz floating point unit achieves 84% efficiency for the Wilson Dirac operator when operating from cache for double-precision arithmetic, *i.e.* 0.84 Gflops. Using the large 8 GByte/sec bandwidth between processor and on-chip memory, this same Wilson Dirac operator can be evaluated with 78% efficiency. Finally with latencies in the range of a few hundred nsecs and an aggregate off-node bandwidth of 12 Gbits/sec, current gate-level simulations achieve a 49% efficiency for the Wilson Dirac operator in the most demanding case of 2^4 sites/node. The performance of machine-wide global sums is also important for efficient use of a large machine. Special broadcast and 'store-and-forward' functions have been implemented. Benchmarks running on the complete, functional simulator achieve 9μ sec for a 64-bit global sum when run as would be required on an $8^4 = 4096$-node machine.

The SciDAC software development effort described in the Section 4 is critical to the effectiveness of the proposed QCDOC machines for the U.S., community-wide lattice QCD research program. The creation of standard, community-endorsed communication interfaces as well as both high- and low-level arithmetic routines provides an efficient path for the evolution and standardization of the large and valuable U.S. QCD code base, allowing its efficient execution on QCDOC machines as well as specifically optimized cluster network architectures. The resulting efficient interoperability will provide significant advances in scientific productivity for the entire U.S. lattice QCD effort and should be a template for international standardization as well. The combination of an industry standard RISC, PowerPC processor, and an attached, 64-bit IEEE floating point unit with this SciDAC software development effort will provide a complete, standardized software environment. The processor and floating point unit are supported by both GNU and IBM 'C' and 'C++' compilers. For example, the generic IBM PowerPC XLC compiler produces object code which evaluates the Wilson clover term with 30% efficiency when run on the QCDOC chip simulator (code produced by the GNU compiler is roughly 1/2 as efficient).

2.1.2 Motivation

The rapid construction of this 1.5 Tflops QCDOC development machine is an important step in the creation of a distributed topical computing center for the study of QCD. This machine will be built and debugged by the QCDOC design team and will provide:

1. A significant platform to be used by the U.S. community for the frontier research problems described in this proposal.
2. A testbed for the QCDOC-specific implementation of the general SciDAC software now under development.
3. A first opportunity to test and refine the user support required for such a topical, non-commercial installation.
4. A serious, user-driven environment which will allow the creation and development of the hardware debugging and maintenance procedures required for a tera-scale computer with thousands of computing nodes.

This development machine provides the platform for achieving the milestones outlined later in this proposal. Once these milestones have been reached, we plan to construct within six to nine months the much larger QCDOC computer at Brookhaven. The user support and software and hardware maintenance for this larger machine will be provided by the same Brookhaven team that is responsible for the successful operation of the development machine. The software infrastructure and maintenance procedures developed during the initiation of the development machine will then be ready to support full and efficient utilization of the much larger, follow-on Brookhaven computer.

2.1.3 Specific objectives

This proposal covers four major objectives that must be achieved in FY2003 to prepare for the FY2004 installation of a large U.S. QCDOC machine at Brookhaven:

1. QCDOC design.
2. Construction and testing of the QCDOC prototype.
3. Construction and testing of the 1.5 Tflops machine.
4. Development of QCDOC-specific software.

In the remainder of this section we will describe these steps in detail, the specific tasks that must be accomplished to complete them, the schedule expected to be followed and the budget required for each. The overall schedule for the project is summarized in Figure 2.

In addition to the DOE-funded activities proposed here there are two other important collaborative components of the QCDOC effort. The first is a RIKEN/Brookhaven/Columbia collaboration which is also providing funding and personnel for the design and development of the machine and

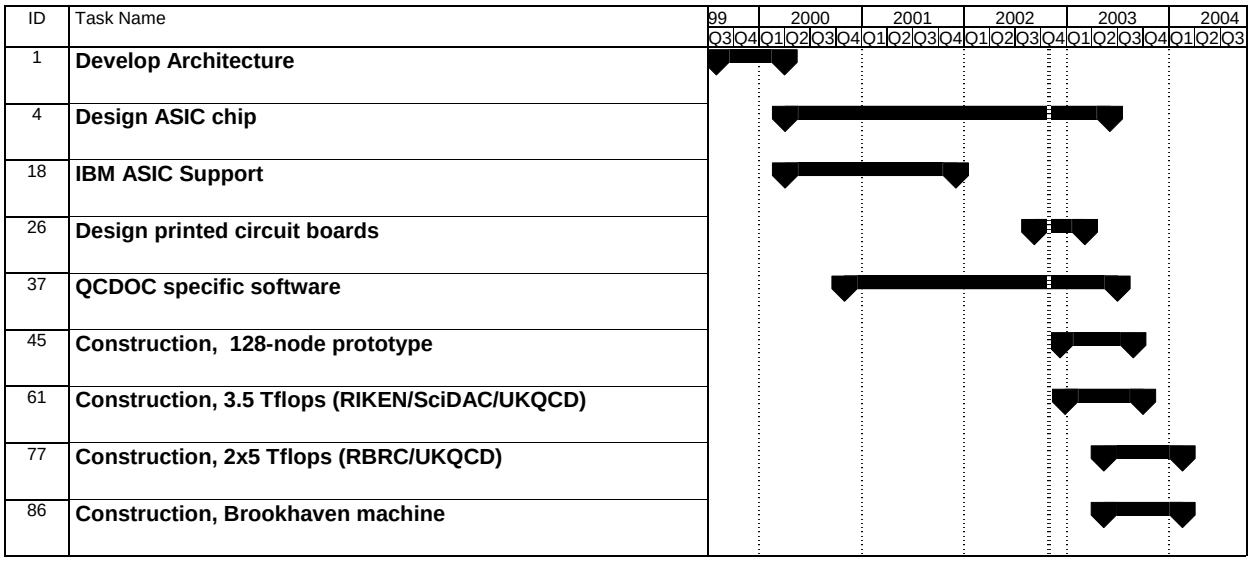

Figure 2: Overall schedule for the QCDOC component of this proposal. Note, the construction start shown for the large Brookhaven machine is the earliest possible permitted by the hardware development schedule.

includes funding (already in place) for a 5 Tflops machine to be installed at Brookhaven as a major research tool for the RIKEN BNL Research Center (RBRC). The second is a UKQCD/Columbia collaboration which is providing resources for the machine design and the construction of a 1 Tflops development machine at Columbia and a 5 Tflops machine located in Edinburgh. While these two added collaborations are not directly involved in this proposal, we will include in this section a description of those portions of the project being carried out by these collaborations, their contributions to the overall budget and their position in the schedule.

As can be seen from the schedule in Figure 2, the QCDOC-related activities addressed in this proposal divide into three stages: design, construction of a 128-node prototype and construction of three, Teraflops-scale development machines. As is discussed below and shown in Figure 2, these three stages are somewhat overlapping with the long-lead-time components required for a later stage procured during an earlier stage as soon as adequate testing has been completed to justify such a step.

The first, design stage includes design of the ASIC, various printed circuit boards and cabinets. Here ASIC refers to “applications specific integrated circuit”. As is described below, this single chip contains a complete computational node for the QCDOC machine, excluding the extra off-chip memory. This design activity, including a possible ASIC respin discussed below, is scheduled to be complete by the middle of 2003. The second stage of the project, beginning in October of 2002, is the construction of prototype hardware. This begins with the smallest number of components needed to perform a meaningful test and expands to include a 128-node prototype machine. This phase is scheduled to have verified the ASIC and two of the three printed circuit boards by the beginning of May, 2003. The third stage is the construction of three development machines, each dedicated to the software development and scientific objectives of each of the groups participating in the QCDOC project: RIKEN (1 Tflops), UKQCD (1 Tflops) and SciDAC (1.5 Tflops). The 1.5 Tflops, SciDAC development machine is the major component of this proposal. This development

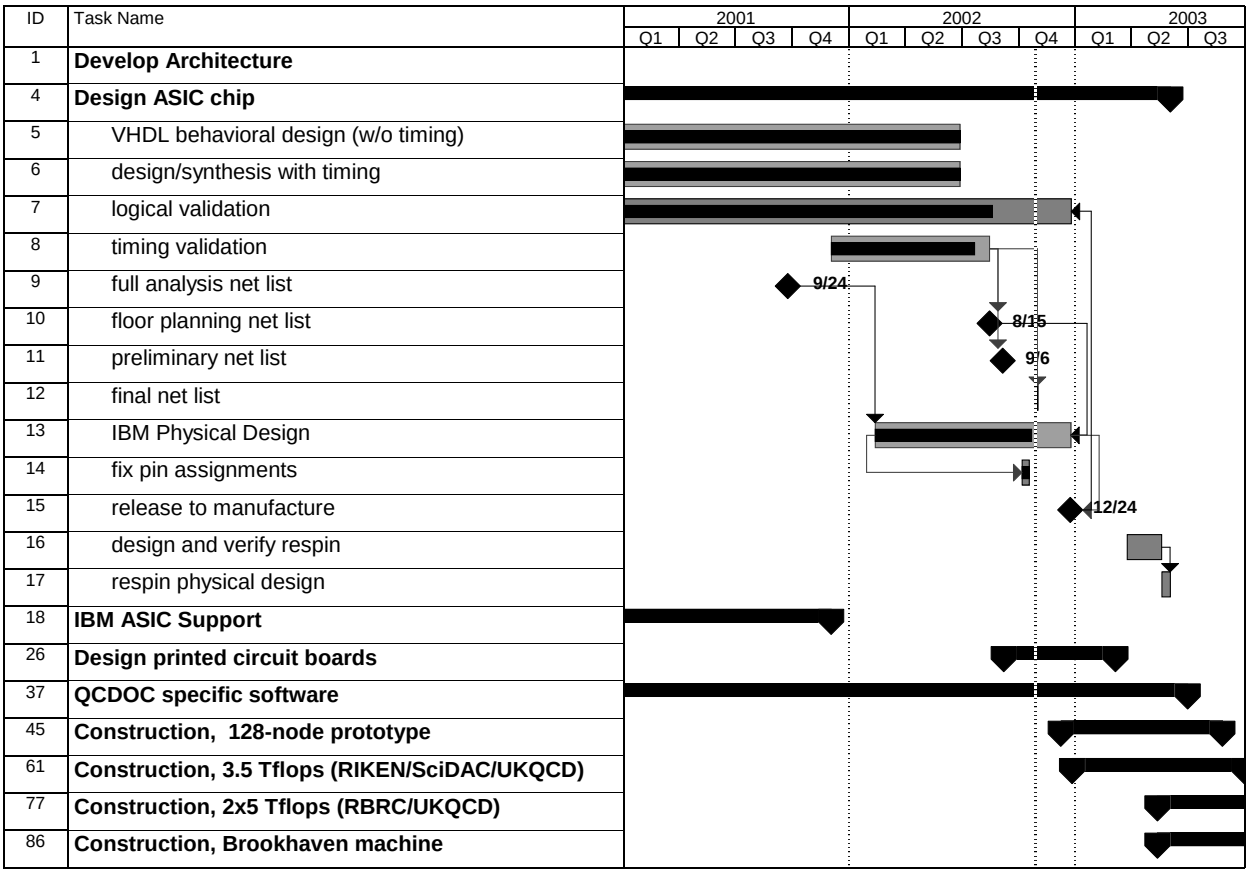

Figure 3: Design schedule for the QCDOC ASIC. Note, the complete schedule shown here allows for the possibility that a second ASIC design will be needed to correct faults in the first version and a second prototype production run must be completed (ASIC respin). Here tasks 16 and 17 are those that would be required by a respin.

machine construction, scheduled to complete in September of 2003, will demonstrate the effectiveness of a large-scale machine and work out last difficulties in large-scale construction. The final phase of the project involves the construction of the large RIKEN, UKQCD and SciDAC machines described above.

2.2 QCDOC ASIC Design

The design of the QCDOC machine has two major components: the design of the ASIC and the design of the printed circuit boards and cabinets which make up the rest of the computer. The design of the ASIC is by far the largest and most complex task and is discussed in this subsection. The printed circuit and cabinet design is included in the next subsections addressing the prototype and development machines.

In Figure 3 we show the expanded schedule for the completion of the design of the QCDOC ASIC, the most complex and risk-prone aspect of the QCDOC machine. As can be seen from that figure this design effort has been underway since the end of 1999 and is now nearly complete.

As is detailed in Appendix B the ASIC has four distinct subsystems. Three for whose design we are responsible, the prefetching EDRAM controller (PEC), the serial communications unit (SCU) and the Ethernet/JTAG interface (EJTAG). The fourth subsystem comprises all of the standard IBM library modules included in our system. Figure 14 in Appendix B shows how these components are interconnected.

The PEC is the most complex and timing-critical unit which we have designed. This design work was done at IBM/Yorktown Heights. It was completed many months ago but is still undergoing substantial tests: both tests run on that unit alone at Yorktown Heights and Columbia and testing at Columbia as part of the entire design. This design has also been simulated using a gate-level netlist, including component timings. The timing assertion files have been written for this portion of the design and, with those assertions, the design passes IBM's static timing requirements at 430 MHz, worst case. Note, in this discussion and that to follow, we refer to the speeds of the various components of the ASIC as determined for "worst-case" conditions: worst-case voltage, temperature and process. The final speed of the complete chip will be determined by that of the slowest component under a combination of "best-case" voltage and temperature since we can carefully control the chip's environment and "nominal-case process" since we may be able to selectively remove the slowest examples.

The SCU is next in order of complexity, however this runs at 1/8 the speed of the PEC with the complex phase detection circuitry required for the 500 MHz serial communication contained in IBM, hard-core libraries. This unit was completed in June 2002 and has been extensively tested both separately and in the full system. Timing assertions have been written for this portion of the design and it functions at 500MHz, worst case, without difficulty.

The EJTAG unit is based on an FPGA designed at Yorktown Heights using the Verilog design language. We have six functional boards containing this FPGA in our lab, connected to the Internet and both 405 and 440 development hardware. This has been extensively tested and is in continuous use for OS development. We contracted with a portion of IBM's Server Division in Rochester MN to port the Verilog EJTAG design to the VHDL language on which our design is based. This design was completed last fall and has been extensively tested. We have held a design review (12/01), directed by the designer of the original FPGA and carefully carried out extensive added tests which he suggested. These included the use of both good and corrupted Ethernet packets, exploring the complete functionality of the Rochester version of the design in our simulator environment. Much of this testing involves the simulation of two of our ASIC's with the first sending Ethernet/JTAG packets to control the second. This system has been stable, without errors for many months. Operating at 25 MHz, asynchronously with the rest of the chip, this unit provides no timing difficulties.

The IBM components of the ASIC design are numerous and some are quite complex. Much of our initial design effort was devoted to understanding their functionality and correctly integrating them in the design. Most of this integration job has been finished for nearly a year and has been extensively tested. Of particular concern has been the complex, 128-bit PLB bus. This is a split transaction bus with a very elaborate protocol. However, IBM's implementation of this bus as well as our connections to it have worked very well from the start with very few problems (although we have had to deal with a series of minor bugs in the direct connection between this bus and the EDRAM provided inside the PEC). Specification of the timing requirements for these IBM-supplied components is a daunting task which has been carried out by IBM engineers at Yorktown Heights and Burlington, VT for the EDRAM and by our Raleigh, NC sales engineer for everything

else. This is the final portion of the timing specification which was completed in the middle of August (8/02).

The physical design work being done at IBM is nearing completion. We have a final choice of chip size (12.59 mm$\times$12.59 mm), a specific pin-out for our 624-pin BGA package and a complete floor plan which places our components on the chip. Our design has passed both the floor planning and preliminary netlist stages which means that the IBM-developed floor plan meets timing requirements. The Raleigh design team is now implementing the clock and reset trees and doing the wiring using our preliminary netlist. The final design work, based on our final netlist, is scheduled to complete during December with IBM anticipating a "Release to Manufacture" (RTM) sometime before the end of December as shown in Figure 3. We are scheduled to receive prototype chips in 54 days after RTM, toward the end of February.

Perhaps the most important part of this ASIC design effort is the design, implementation and execution of a thorough testing strategy. Since the simulation of our design runs very slowly (at about a one Hz) we are limited in the number and length of tests that can be run. The number of separate simulation jobs that can be run simultaneously is limited by our ten licenses for the external memory simulation software. However, using Columbia workstations and a 6-processor SUN Enterprise server that has been lent to us by Brookhaven, we can run ten such simultaneous jobs. Added simultaneous testing jobs focusing on modules within the design that do not require the DDR memory can also be run. The tests are of three types: i) Tests of individual components. ii) Tests of particular functions of the complete design. iii) Tests of realistic application code running on the complete design. While the bulk of these three styles of tests are run on the behavioral description of the design, we have run some tests, especially those of individual sub-modules, on the gate level description of the design, including timing. We will discuss each of these tests in turn.

The tests of the individual components were written as those components were designed and have been preserved so they can be re-run if design changes are made. Two of those tests are sufficiently elaborate to be useful in the final testing of the design. The first is the test of serial communication between two of the twelve send-receive pairs that make up the serial communication unit. These tests transfer a file of 64-bit words from one unit to the other, running in both directions. In addition, random single bit errors are introduced to demonstrate that the error correcting circuitry works. In order to further increase the realism of this test, random interrupt supervisor packets are continually sent in both directions. As part of our complete testing program, we are running this test for many hours on both the behavioral and gate-level versions of the design allowing a wide range of possible patterns to be tested.

An even more elaborate test environment has been created for the PEC and attached EDRAM. This allows a combination of simulated 440 memory operations from both the fast direct connection between the 440 and the PEC and from the 440, through the PEC to the PLB and from there to the PLB port on the PEC. These simultaneous operations use constrained but otherwise random memory addresses and directionalities to test both correct operation of the EDRAM and the prefetching logic, with a particular focus on coherency issues. These tests have been run for thousands of hours at Yorktown Heights. They have also been ported to the simulation environment at Columbia were continuous testing is also underway, running on eight independent workstations until the chip manufacturing begins.

The next style of tests are run on the entire design but are focused on the functioning of particular

units. Tests of the Ethernet JTAG interface demonstrate that boot code can be successfully loaded and that the unit will function without problem even in an environment with corrupted Ethernet packets. Tests for the ECC unit for the EDRAM and the various programmable functions of the DDR controller are also required to verify that those aspects of the design work. Both the global and partition interrupts have their own testing suite while extensive testing of SCU transmission with the overlapping of all possible types of messages (data, supervisor, interrupts and various acknowledgments) with the random introduction of single bit errors is important. Since this later test explores a variety of relative timings, long runs of 1-2 weeks are of clear value. These tests were initially developed to demonstrate that the corresponding modules worked correctly within the larger design. However, they have been integrated into a global testing structure so they can be run automatically as part of the final testing process.

Finally and most important, high-performance QCD application code is run on the complete design. This comprises more than half of our testing and provides a reasonable demonstration that the final design will do the job for which it was created. The complex pattern of operations that has been optimized to execute with maximum performance is a very demanding test of the design. We believe that this ability to run such optimized production code is a special advantage of our application-directed chip design effort. Since tests of type ii) and iii) are driven by code that is executed by the 440, it is relatively straight forward to run those same tests on actual hardware once the corresponding code has been loaded into the physical memory. Thus, these tests for the ASIC design also form the heart of our testing strategy for the hardware that will be built during the prototype phase.

The two remaining critical dates in this ASIC schedule are our submission of the final netlist and IBM's completion of the physical design. The final netlist represents our last chance to easily correct design errors and should be submitted when we are assured that our design has been thoroughly tested. The high-performance physics code, essential for this final testing, was finished a few months ago. This testing of our final ASIC design is now being actively carried out by four postdocs and three graduate students at Columbia. We expect that the faults visible to our tests have now been found and corrected, making November 1^{st} a reasonable date for this milestone. We will continue to actively test the design until the actual chip manufacturing begins. However, correcting faults found at this late stage may delay the schedule and/or incur added charges.

As is discussed below, we have allowed for the possibility that important faults are found in the first ASIC's that are manufactured. These could be caused by errors in our design not found during testing or errors in IBM's physical design or manufacturing. These errors would be discovered in the prototype construction phase and would stretch the ASIC design effort beyond that required to complete the first chips. It is to represent this possibility that the ASIC design activity stretches until June of 2003, extending well beyond the start of prototype and 1.5 Tflops construction. Depending on the source of such errors, as much as an additional $221K may be required for this "respin" of the ASIC if the fault can be corrected by modifications to the metal layers, funds which have been included in the overall budget. The largest possible respin charge would be the full ASIC NRE of $340K. This has not been allowed for in the budget and but could be accommodated by a resizing of the various planned machines.

The budget for the ASIC development effort is shown in Table 1. This $1,471K expense was divided among the U.S. DOE, RIKEN and UKQCD as $369K, $400K, and $702K respectively. Since this portion of the project is nearly complete, this budget is quite precise.

Item	Cost ($K)
ASIC design software	$20
ASIC NRE	$340
Power PC 440 Core Fee	$50
Core access	$100
ASIC Engineering support	$100
T.J. Watson design support (serial)	$150
T.J. Watson design support (PEC)	$223
Rochester/T.J. Watson design support	$234
RAID system	$24
Sun memory (Clearpoint)	$9
ASIC respin charge	$221
Total	$1,471

Table 1: Breakdown of the design cost for the QCDOC ASIC. These costs have been covered by earlier funding and are not part of this proposal.

2.3 Construction and testing of the QCDOC prototype

In Figure 4 we show our planned schedule for the construction of the prototype QCDOC machine. This has two major components: the design of the printed circuit boards and the assembly and test of the boards and chips. We will exploit the simplicity of a multi-daughter board system that can be realized using a single mother board and perform our first tests using a small number of assembled daughter boards, one assembled mother board and a single-mother board test fixture. By placing even a pair of daughter boards in six different locations, we can verify the complete daughter board communications circuitry permitting the assembly of more daughter boards. Thus, we will initially fabricate eighty daughter printed circuit boards, perhaps four printed circuit mother boards and two printed circuit single-slot backplanes. We would then assemble five daughter boards, one mother board and one backplane. These units would then be thoroughly tested to demonstrate the full functionality of the QCDOC ASIC, including booting, general Ethernet communication, inter-node communication, external DDR DIMM access and flawless execution of all codes developed to test the ASIC using the simulator. When this verification of the ASIC design has been successfully completed, expected toward the end of March 2003, we would begin procurement of the ASICs for the development machine. We would also proceed to construct and test a fully populated, 128-node system. This unit, to be completed in April 2003, will provide a reasonably complete testbench for the daughter board, mother board and high-speed connectors and cabling as well a modest platform to explore the performance of the QCDOC architecture.

If all proceeds without major difficulty, three of the four critical components, ASIC, daughter board and mother board will be verified by the beginning of May, 2003. This verification will include a successful demonstration of the physics code available at this stage in the project from both the Columbia/RIKEN/UKQCD and the broader SciDAC software efforts. In particular, the high performance Dirac inverters that have been developed to test the ASIC and verify the QCDOC architecture should run reliably and show their predicted performance. At that time the construction of the 1.5 Tflops development machine can begin. While difficulties with the printed circuit boards would likely cause a delay of perhaps two months, a fault with the ASIC would require a redesign and remanufacture. This eventuality is shown explicitly in the schedule. Referred to as a respin

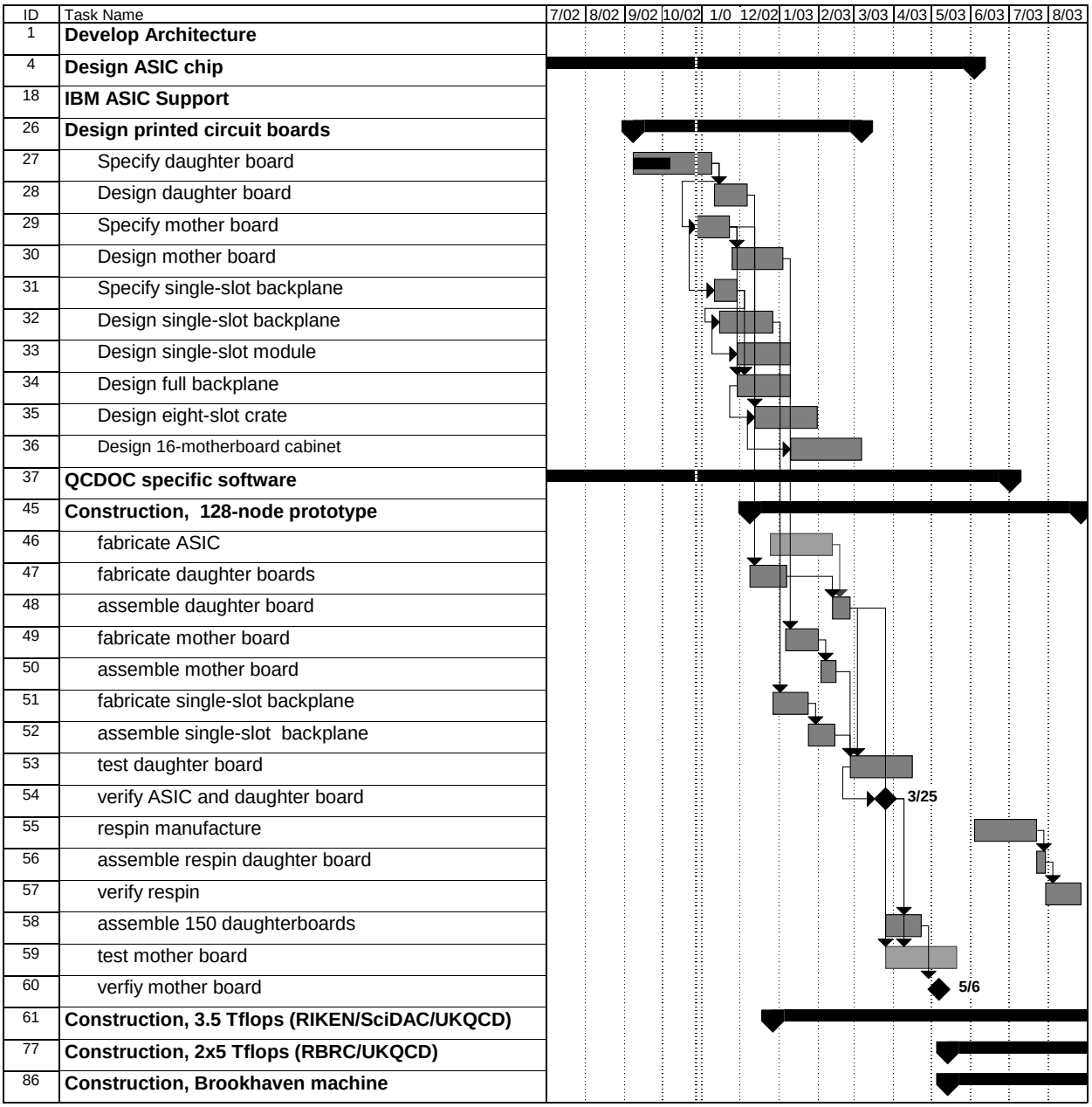

Figure 4: Construction schedule for the QCDOC prototype machine.

in the schedule, this could add as much as a six month delay. Of course this is the worst case and assumes that all steps must be repeated after the new version of the ASIC is available. Depending on the nature of the fault, there may be much testing of the printed circuit boards that could be done, reducing that delay, in the most favorable case to three months.

While it is not possible to guarantee that such a respin can be avoided, our testing strategy is designed to minimized the likelihood that this will be necessary. By testing the chip with complete applications code, something that is typically not done in commercial ASIC projects, and by being extremely careful that these tests are run on the final design, we significantly increase the chance that the chip will actually do what we need it to when it is manufactured. Our earlier QCDSP experience supports this approach. There were errors in the original QCDSP ASIC, occurring in portions of the design that were not tested at the end of the design process. However, these errors did not significantly interfere with running the applications code that had been used in the final tests. As a result the initial chip was satisfactory and no respin was performed. As a further precaution, we have been very deliberate to preserve hard-wired JTAG accessibility to the 440 core within the ASIC and to provide individual reset control for each of the components within the design. This will increase the probability that if there is a problem with the initial ASIC we will be able to diagnose the problem directly with our prototype chips.

The budget for the prototype phase of the project is composed of three parts. The first covers the hardware needed for software development. These costs are listed in Table 2. This hardware, which has already been procured and put into operation, provides a realistic, 4-node prototype system with a 2-processor Sun host and gigabit Ethernet connections. Here the four nodes are made up of 405 PowerPC processors rather than the planned 440 units. These are reasonably similar and this four-processor system provides an essential tool for software development and testing.

Item	Cost ($K)
RISCWatch debugger	$75
Sun Ultra Sparc host	$12
Gigabit switches	$2
2 Sun PCI Gigabyte adapter	$3
4 PPC405 development systems	$13
PPC440 development system	$4
JTAG/Ethernet boxes	$8
Total	$117

Table 2: Cost of the hardware needed to support the design and implementation of the QCDOC operating software. This cost has been covered by earlier funding and is not part of this proposal.

The second portion of the prototype budget includes the costs of printed circuit and cabinet design and the associated non-recurring engineering costs (NRE) required for initial manufacturing and assembly as well as some test equipment. This table also includes the costs of fabricating and assembling five-daughter boards, one mother board and one single mother board backplane and cabinet. For speed and some economy, the costs of the fabrication of the printed circuit daughter boards includes all those needed for the first 128-node prototype. These costs are given in Table 3.

The third aspect of the prototyping costs is the construction of the 128-node machine that will be used to demonstrate the operation of a significant multi-node system. This will provide a platform

Item	Cost ($K)
Daughter board design	$10
Daughter board NRE	$3
Daughter board fabrication (150)	$2
Daughter board assembly, (5)	$1
Digital oscilloscope	$10
Pulse generator	$2
Mother board design	$15
Mother board NRE	$5
Mother board fabrication (3)	$3
Mother board components (1)	$2
Mother board assembly (1)	$3
1-mother board backplane design	$5
1-mother board cabinet design	$1
1-mother board backplane NRE	$3
1-mother board cabinet fabrication	$3
Backplane design	$10
Crate design	$5
Backplane NRE	$3
Total	$86

Table 3: The cost for design and manufacture of the components required for the first few-daughter card prototype test of the ASIC, daughter board and mother board design. This cost has been covered by earlier funding and is not part of this proposal.

permitting the Collaboration to perform extensive tests of the application specific software, which is being developed with support from our SciDAC grant (see Secs. 4 and 4.3). These costs of this component of the prototype effort are listed in Table 4. Finally in Table 5 we gather together the prototype costs shown in the three earlier Tables 2, 3 and 4. The total cost for this prototyping phase have divided between the U. S. Department of Energy and the UKQCD collaboration as $147K and $206K respectively.

Item	Cost ($K)
Nodes (150)	$105
Motherboard (2)	$12
Backplane(2)	$20
Crate	$10
Cables (10)	$3
Total	$150

Table 4: Breakdown of the cost for constructing the 128-node prototype machine. This cost has been covered by earlier funding and is not part of this proposal.

Item	Cost ($K)
Development hardware	$117
NRE and first prototype	$86
128-node prototype total	$150
Total prototyping costs	$353

Table 5: Summary of the costs associated with the QCDOC prototypes. Shown are the totals from Tables 2, 3 and 4. These costs have been covered by earlier funding and are not part of this proposal.

2.4 Construction and testing of the 1.5 Tflops machine

In the subsections to follow we will describe the concrete elements of this proposal for the construction of the 1.5 Tflops development machine, added support for the development of operating and diagnostic software, and the provision of user support and hardware maintenance.

The QCDOC development machine whose construction is being proposed will be a system made up of a 3K-node machine housed in three cabinets and a second, single-crate configuration, capable of holding an additional 8, 64-node mother boards which will be available for code development and short test runs.

The QCDOC architecture is optimized for cost-efficient lattice QCD calculations on very demanding problems. The high-bandwidth, low-latency grid-based network communications permits a large number of processors to be applied to a single simulation. The machine's features can be summarized as follows (a more detailed description can be found at http://xxx.lanl.gov/abs/hep-lat/0110124):

- Each processing node is comprised of a single chip and a DDR SDRAM memory module.
- The processing nodes are joined into a 6-dimensional nearest-neighbor network with 0.5 Gbit/sec communications provided in each direction between each neighboring pair of processors. During typical operation this network allows the machine to be divided into 4-dimensional partitions running independent jobs.
- The custom-designed processor chip contains:
  - A 440 PowerPC embedded 32-bit RISC processor with a 1 Gflops, double precision IEEE auxiliary processor. The processor is Book-E compliant (a standard for embedded PowerPC processors) and produces floating point results which are bit-by-bit compatible with those produced on a MAC. The unit has a 32K-word instruction and 32K-word data cache and powerful memory management capability.
  - A 4 MByte embedded memory with 8 GByte/sec bandwidth to the processor. This memory is sufficient to hold many lattice QCD problems entirely and a DMA unit is available to provide automatic transport of code or data between this memory and the external DDR SDRAM.
  - All logic needed to support the 6-dimensional network communication. This includes a low-latency store-and-forward functionality needed for efficient broadcasts and global sums on this mesh network.

– A Fast Ethernet controller allowing host-node communication using this standard and economical protocol.

– An Ethernet/JTAG controller which accepts Ethernet-embedded JTAG commands. This allows the host to boot and debug individual nodes using the low-level JTAG control provided for the 440 processor. This also permits a multi-node implementation of IBM's RISCWatch debugger providing a convenient and powerful debugging tool for both hardware, OS and user code.

Table 6 below lists the major components of this development machine, their quantities and estimated costs.

Item	Unit cost	Cost ($K)
3400 nodes	$284	$964
54 motherboards	$3,300	$178
7 backplane	$5,564	$39
3 cabinets	$15,200	$45
324 cables	$265	$86
1 crates	$5,175	$5
Ethernet switch		$64
5 TByte disk storage		$50
host computer		$40
technical assistance		$29
Total		$1,500

Table 6: Breakdown of the $1.5M development QCDOC machine cost.

The construction schedule for this development machine is presented in Figure 5. Construction is planned to begin at the end of December 2002 with a single 8-mother board backplane and cabinet funded by RIKEN. (We will also construct an ancillary 4-motherboard machine, funded by UKQCD, which will be debugged and put into service as soon as this first 8-motherboard machine is stable.) This step will verify the backplane and the operation of a system of this size. Once successful operation has been demonstrated, we move to full-scale construction in the beginning of May. In addition to the 1.5 Tflops machine addressed by this proposal, two other 1.0 Tflops development machines will be constructed as part of the RBRC and UKQCD projects. These three machines are represented in the schedule of Figure 5as the single 3.5 Tflops entry.

The schedule presented in Figure 5 represents a careful attempt to complete the construction of these development machines as quickly as possible and at the same time to minimize the risk that monies will be spent on components that are later discovered to be flawed. Thus, long lead time items such as the ASIC's will be procured before the initial mother board tests are complete but after the ASIC has been thoroughly demonstrated to be appropriately functional. Similarly, the schedule calls for 8-slot crate production to begin at the end of December 2002, well before the testing of the 128-node prototype is complete in May. However, the power supply procurement and cabinet construction are long-lead-time items and should be started as soon as the required printed circuit board and cabinet design have advanced to an appropriate stage. Thus, these advance procurements are done "at risk" with the expectation that even if a difficulty were to be uncovered in later testing, a significant faction of this advanced investment may be recouped by use in an altered

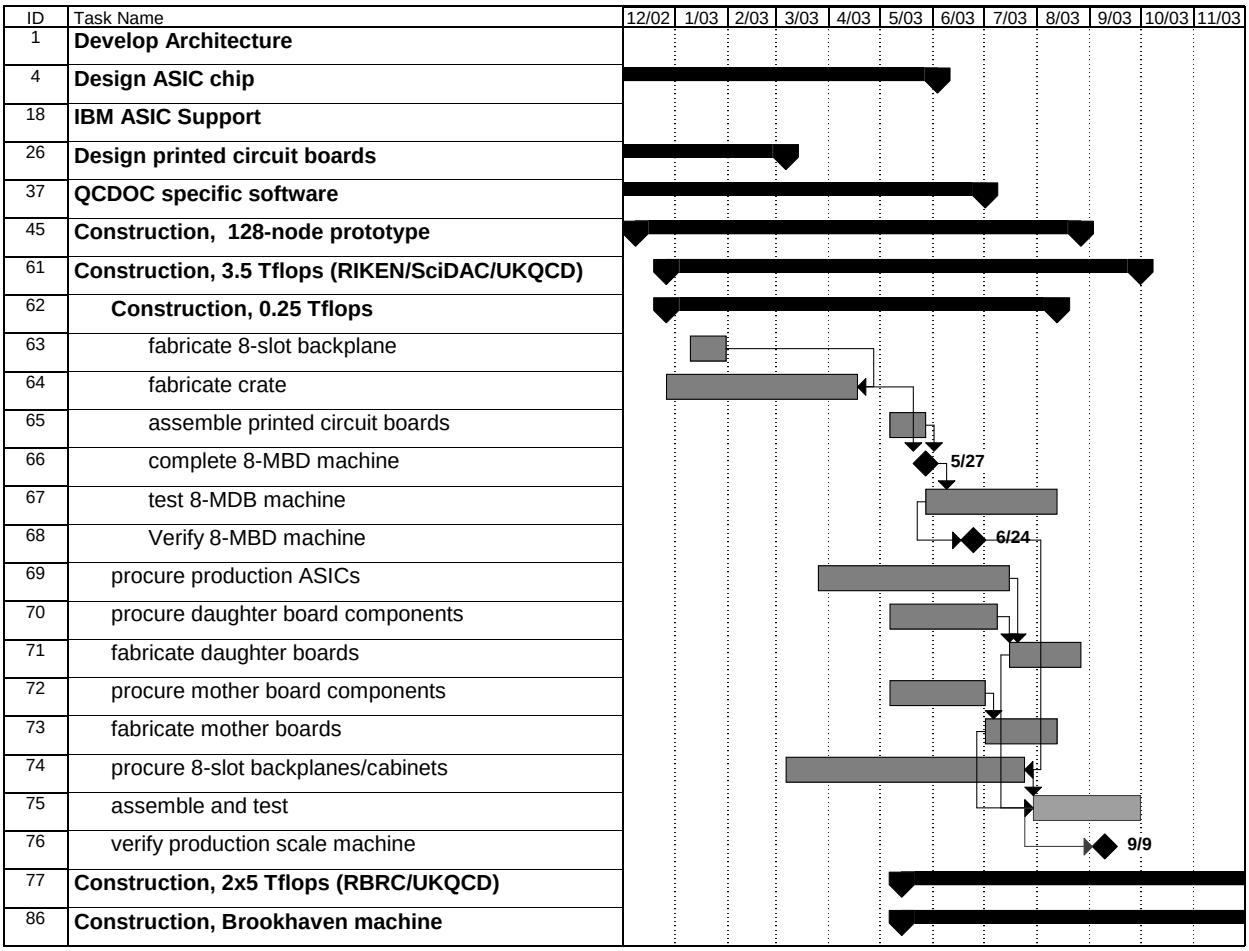

Figure 5: Construction schedule for the QCDOC development machines.

version of the design. Given the rapid advance of technology, it is very important to complete a project of this sort in a timely fashion and the added months of early physics research that can be accomplished are believed to justify the risks described.

This construction of large-scale development hardware by the design team is essential for the efficient remote construction of the planned large U.S.-supported QCDOC machine at BNL, the 5 Teraflops QCDOC machine for the RBRC and the 5 Teraflops UKQCD machine. While the Columbia faculty, students and postdocs assume the major responsibility for the construction of these machines, technical personnel from both Brookhaven and Edinburgh will also participate in their assembly and debugging.

This schedule will be reviewed as the design is completed, testing begins and costs become more certain. For example, by May of 2003, a decision must be made regarding the size of the external memory cards. The present budget anticipates $65 for 128 Mbyte 184-pin, DDR SDRAM registered DIMMS with ECC. Should funding permit, larger units may be purchased. The SciDAC Committee of PI's will establish a "QCDOC subpanel" including two members from Columbia and three from the larger collaboration, to make these decisions.

2.5 Software specific to the QCDOC

Critical to the timely, successful operation of this development machine is the operating system and diagnostic software. For a machine with thousands of processors, the operating system software must not only provide convenient user access to the machine for running applications, it must also provide monitoring and reporting on the hardware status to allow rapid isolation and repair of faulty components. From our experience with QCDSP, we have considerable experience upon which to build the operating system and diagnostic software for QCDOC. The parts of this experience which must be tailored and adapted to QCDOC will proceed hand-in-hand with the various steps in the hardware assembly outlined in this proposal. Certainly for the debugging and diagnostic parts of the operating system software, trials on early QCDOC hardware will be vital to ensure that the software is handling the tasks we demand correctly.

The development of QOS, the QCDOC operating system and diagnostic software, is currently the focus of four individuals: Prof. Mawhinney of Columbia, Drs. Balint Joo and Peter Boyle of Edinburgh and Xiaodong Liao, a Columbia graduate student. Dr. Chulwoo Jung, a SciDAC staff member at Brookhaven, is also spending a portion of his time on this and a new SciDAC staff member, Dr. Konstantin Petrov is beginning to help with this activity. During the last few months, Dave Stampf from BNL, along with other members of his group, have begun meeting with the operating system team and will help with user interfaces to the QOS. We expect additional help with this task from interested Columbia University graduate students.

However, given the critical role that the development of this software plays in both the completion and testing of the QCDOC machines and in the flexible support of a general user community, we propose to recruit two postdoctoral fellows to work specifically on these tasks.

2.5.1 Overview of QOS

The QOS is readily described in terms of two major subsystems: the host side OS (QOS-H) which runs on the host front-end and the machine side OS (QOS-M), running on the QCDOC ASIC. Users, as well as batch queuing systems, interact with QOS-H, while QOS-M provides the execution environment for user applications along with hardware monitoring. These two major subdivisions of QOS work via a physical Ethernet connection for data transfer and communication.

Both the QOS-H and QOS-M halves of the operating system can be further divided into boot-time and run-time components. The boot-time part of QOS-H and QOS-M provides automatic testing of all the hardware components of QCDOC, determination of machine geometry, cataloging of serial numbers of hardware components in use and tests of basic I/O channels. For particularly difficult hardware problems, the boot-time QOS can invoke ever more fine-grained JTAG commands to interrogate the processor, possibly leading to a RISCWatch debugging window attached to a particular processor. A later step in the boot-time procedure is the loading of the run-kernels to the QCDOC ASIC and verification of their functionality.

The run-time components of QOS-H and QOS-M will proved convenient user access to QCDOC for calculations and I/O. The QOS-M run-time kernel will be a small, essentially real-time kernel, providing the user environment for program execution. By keeping the kernel small, high-reliability is easier to achieve. The kernel will support the standard library calls needed for C and C++ programs, but will not have fork() or exec() capabilities, since any required multitasking can be handled by the host and queuing system. From our experience with QCDSP, it is vital that the kernels maintain constant monitoring of the state of the hardware. With many thousands of nodes, there will be failures and the run kernels provide the mechanism for rapid diagnosis and replacement of faulty hardware. This capability is vital for effective use of QCDOC and cannot be provided by any existing software kernels. Through kernel-system calls, users will be able to access both host file systems and the parallel file system of QCDOC.

Users will interact with QCDOC through the QOS-H part of the QOS. This software will allocate and coordinate the partition of the machine a user has access to. The QOS-H will keep information about the hardware status of each node, provide basic standard I/O to/from any collection of nodes on QCDOC and invoke any needed diagnostic tools required when a machine fault is detected. Multiple means of access to QOS-H are currently planned. Minimally there will be command line access through a basic terminal window and socket access from scripting languages (Perl, etc.). Web access is also envisioned, contingent upon the detailed evolution of the plans of the Software Committee.

2.5.2 QOS Status and Schedule

A substantial amount of work on the QOS has already been done. Since the QCDOC ASIC contains a standard 440 PPC CPU connected via Ethernet to the outside world, we have been able to do development work on the QOS using currently available hardware containing 405 and 440 GP (General Purpose) CPU's. This hardware does not contain a floating point unit, but this is of no real consequence for QOS development. Since we also have hardware boards with Ethernet/JTAG units implemented in FPGA's, we can manipulate the 405 and 440 GP boards through JTAG just as we will the QCDOC ASIC. The availability of this commercial hardware for QOS work should

allow us to have the QOS in a quite complete state before the first QCDOC ASIC has been built, with only a few low-level software driver routines being modified in the migration to QCDOC hardware.

In particular, a working version of the boot-time QOS-H and QOS-M software is already available. This allows a user on the host computer to reset and boot a collection of up to 4 PPC 405 boards. The QOS-H software uses Ethernet/JTAG packets to load a small boot kernel into the instruction and data cache of each board, which then enables the standard 100 Mbit Ethernet controller. At this point, the boot-kernel allows full read/write/execute functionality to the board from the host. For QCDOC, the various hardware tests would be loaded onto the machine and run at this time. The first versions of these tests will be extensions of the existing tests we are using to debug the QCDOC ASIC. Thus, much of this work is already done. We expect additional tests will likely be needed as we tune our debugging strategy for QCDOC. The existing working version demonstrates all the steps necessary for basic booting and debugging of QCDOC. This version must reach its final form by the time the first ASIC's arrive, since the software will be required to test the functionality of the ASIC.

A commercial SMP processor is planned for the QCDOC host computer. Possibly as many as 10 independent 1 Gigabit Ethernet links from the host may be used to provide quick response for booting and diagnostic software. Part of the work done to date on the QOS involves the development of a threaded packet driver library on the host to allow the SMP to utilize the bandwidth of the independent Ethernet links. Further work and testing will be done to ensure that the available bandwidth is efficiently used.

For the initial testing of the QCDOC ASIC's, not much software beyond the existing boot-time parts is needed. With read/write/execute functionality, very general programs can be run. We have already implemented a RPC stack required to NFS mount host file systems on the 405 GP boards using only the features in the boot-time components of the QOS. Our QOS software activity is currently focused on the development of the run-time parts of QOS.

The run-time parts of QOS may continue to improve indefinitely, but a major goal of this proposal is a stable, convenient version of QOS for general users to use the 1.5 Teraflops development machine. For the 1.5 Teraflops machine, the QOS will support both command line and script based user control of QCDOC, facilities for compiling and running generic C and C++ code using the QCD-API QMP interface for inter-node communication, single-node debugging windows using IBM's RISCWatch debugger and file I/O to both the host front end and the QCDOC file system. Some of the user interface features will be driven by the developments in the Software Coordinating Committee. In addition, the development machine provides a perfect opportunity to test our debugging strategies in the context of a larger user community, to ensure that they are in place before a topical center is built at BNL.

During the remainder of calendar 2002, we will be finalizing the boot-time portions of the QOS, with focus shifting primarily to the run-time portions. Work is currently underway to finalize the internal features of the QOS: data objects, functionality and protocols. The low-level packet libraries from the existing boot-time portions will likely only need slight modifications to reach their final form. Since the majority of the features of QOS can be completed on the 405 GP boards, this software should be ready to aggressively test and use the first QCDOC ASIC's.

In Figure 6 we show the planned schedule for the development of the various parts of the QCDOC-specific software described above. The completion dates are coordinated with the planned avail-

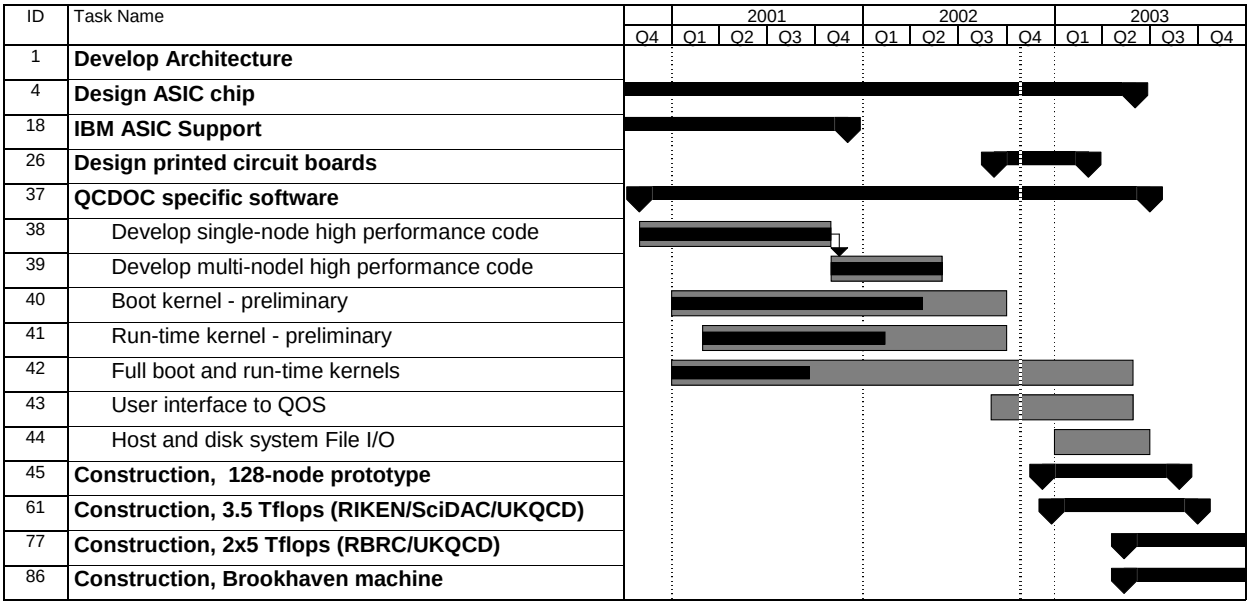

Figure 6: Schedule for development of the QCDOC-specific software.

ability of the QCDOC hardware to first permit efficient and thorough testing of the ASIC and multi-node booting and communication networks. Later steps insure that the needed software base is in place to support physics production jobs on the 1.5 Teraflops machine.

2.6 Operation of the 1.5 Tflops machine

Extensive user support and hardware maintenance is essential if this QCDOC development machine is to realize its potential as a substantial resource for the U.S. lattice community, is to provide a successful demonstration for the future multi-teraflops BNL topical computer and is to serve as a testbed for the development of these services which will be even more critical for this larger machine.

This support will be provided by four personnel: the Brookhaven SciDAC-funded staff member and postdoc, an additional software engineer and Brookhaven hardware technical support, budgeted at 1 FTE. In addition, system administration for this development machine will be provided by the Columbia/RIKEN system manager, Dr. Zhihua Dong. Central to this effort, but funded by independent SciDAC funds for the coming year, are Dr. Chulwoo Jung, who joined the project one year ago as a Brookhaven staff member and Dr. Konstantin Petrov, a Brookhaven postdoc who just arrived June 1st. (Chris Miller, a third SciDAC-funded post-bach left the project at the end of August and begun graduate study in Physics at Columbia.) Dr.'s Jung and Petrov represent an extremely valuable component of the planned user support. Both are practicing lattice theorists who will have an in-depth knowledge of QCDOC operating and applications software as well as the SciDAC software infrastructure. These individuals will be able to contribute effectively to further software development as planned by the collaboration as well as to provide very knowledgeable software/applications support to new users of the QCDOC development machine.

A significant resource of the sort being proposed must be run in a steady, reliable way, delivering

high utilization. Of course during the first few months of burn-in, there are likely to be unexpected problems which will interfere with the continuous running of applications code. During this period, the Columbia students, faculty and postdocs who have constructed the machine will naturally take responsibility for debugging and fixing problems. However, after this initial period it is important that we put in place a procedure for routine repair and maintenance that does not rely on continual attention from Columbia physics personnel. This hardware maintenance will be provided by the same Brookhaven team of three hardware technicians who currently supports the QCDSP machines. During the first year of operation this support is budgeted at 1 FTE. These individuals will provide on-site hardware support at Columbia, both performing needed maintenance and becoming experienced at the debugging and repair of this new machine.

Finally, in order to provide the needed level of user support for this development machine and an opportunity to develop the software infrastructure that will be required for the larger BNL machine, we will recruit a software engineer, expert in web-based applications and control, who will be responsible for implementing the batch and queuing systems specified by the Collaboration's software coordinating committee and creating an automated user interface. In fact, Brookhaven has already identified a high-level, Java expert whom we expect to be able to recruit for this position.

2.7 Budget

In Table 7 below, we present the budget needed to support the QCDOC-related effort proposed here.

Item	**Cost**
FY 2003	
1.5 Teraflops development machine	\$1500
Personnel:	
OS/diagnostic software postdocs, 2 FTE	\$170
Software engineer	\$180
BNL technicians/Facility Management	\$150
Total	\$500
Total	**\$2000**

Table 7: Budget required for QCDOC construction and support (all numbers are given in thousands of dollars) for FY2003. This table lists the QCDOC-specific funds requested in this proposal.

3 Commodity Clusters

Driven by the opportunities for utilizing commodity components for highly cost effective lattice computation, as explained in the introduction we will also aggressively pursue the development of commodity clusters during the time span of this proposal under the support of present SciDAC funding. In response to new opportunities that open up even more cost effective possibilities than available at the time of the SciDAC proposal, we are expanding the scope of the cluster research and development to explore more cost effective network technologies.

3.1 Architectural Approach

It is useful at the outset to define the range of commodity components we are considering for clusters optimized for lattice QCD. For the computational nodes, the current commodity marketplace provides both processors and motherboards that meet our computational needs and price requirements. For the interconnect, our QCD application has driven us to explore a broader range of options. These include fully commodity components like Gigabit Ethernet, where competition between multiple vendors in the marketplace drives the price down. They also include semi-commodity components like Myrinet network interfaces and switches, which while technically adequate, do not benefit from competition-driven prices. Finally, we also include the use of commodity field programmable gate arrays, FPGA's, which can be mounted on appropriate PCI cards with a short development time.

Clusters based on commodity components offer many advantages. Market forces are producing rapid gains in processor and memory performance, with Moore's law increases in processor performance of the order of 60% per year, and with the Pentium 4 yielding exceptional performance for QCD. The market for interconnects is growing and the technological options are increasing. The continuous appearance of new hardware with components improving with Moore's law makes possible the continuous improvement of computing facilities without the need to alter applications software. The cluster software environment makes use of mature operating systems, development environments, and batch systems that are already developed and in the public domain. In addition, small university systems and desktop machines can be made identical to the large production facilities, facilitating software development.

For some important applications, commodity hardware already delivers better than $1/MF, and is expected to continue to improve. For example, the valence quark propagators calculated on the moderately-sized unquenched configurations that are becoming common will use a significant portion of our computer time. These often fit into a small number of nodes, with many separate configurations being processed in parallel. Current clusters already provide efficient and cost effective platforms for such calculations. Furthermore, the flexible communications systems of clusters may be particularly effective for lattice QCD actions that are highly nonlocal. Since commodity facilities are expected to be continuously upgraded with improving components, by 2004 the fraction of the workload for which clusters are optimal is likely to be significant and growing.

3.2 Overview

The SciDAC funded work on cluster development is being undertaken collaboratively by Fermilab and JLab/MIT. The scope and schedule of the development effort is shown in Fig. 7. Prototypes for the development of very large clusters are being built, investigating the range of node and network options described above. Initial systems exploring the use of both single and dual Pentium 4 nodes with Myrinet have been built and are being actively used both for physics production and the study and optimization of performance. Procurement is beginning for a larger dual Pentium 4 system with faster processors.

More cost effective alternatives to Myrinet have been vigorously investigated. As a result, a gigabit ethernet cluster is planned for early 2003 and a research and development project will explore the use of FPGA's. Emerging technology is being investigated for a new generation of clusters in mid

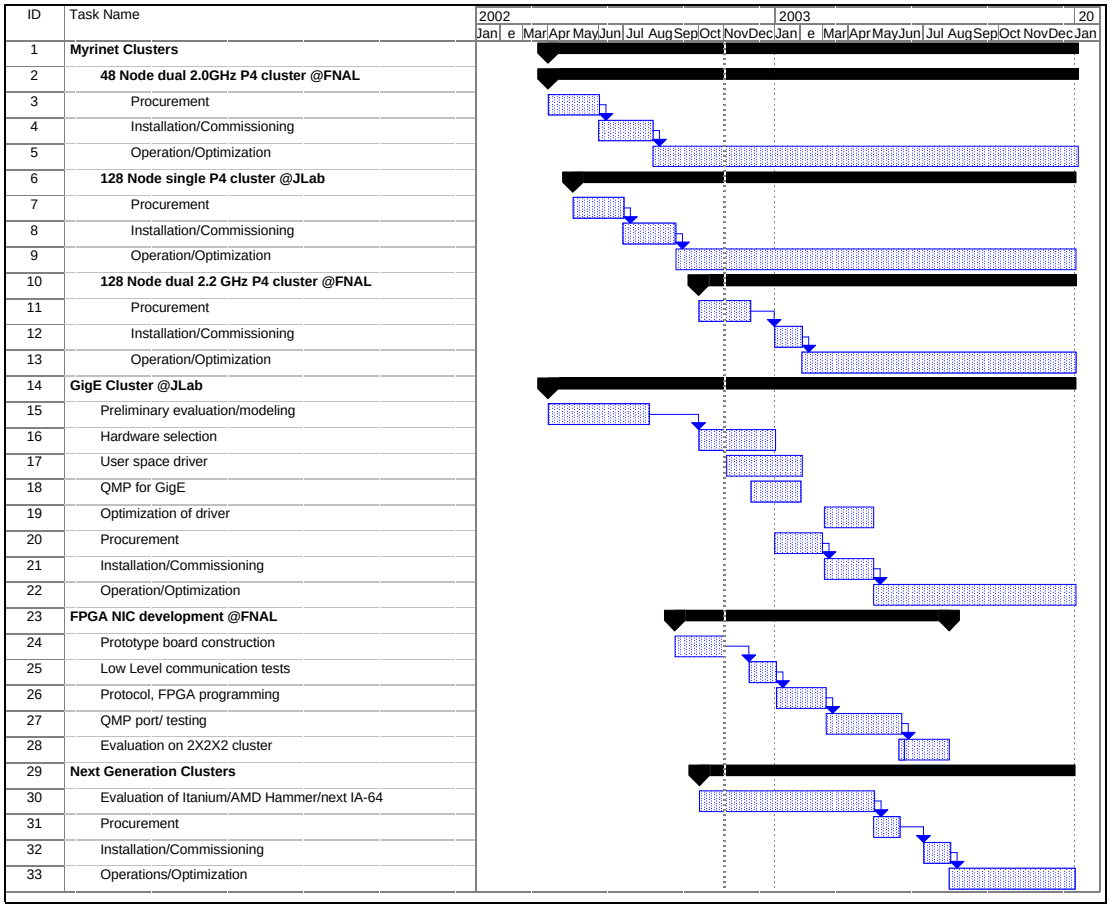

Figure 7: Schedule for research, development, and construction of clusters

2003.

It should be emphasized that all the development systems will be significant computing facilities in themselves and will be employed nearly full time for important physics production computing as well as for the development work. Indeed, Fermilab and JLab are currently working with the Scientific Program Committee to make the recently commissioned Myrinet clusters described below available to the national SciDAC user community. This production use of the facilities is essential to ensure that the user environment matures as the performance increases.

3.2.1 Cluster Experience

It is important to note that this SciDAC cluster project is built on previous, long-term cluster efforts at Fermilab and JLab/MIT that recognized the potential of commodity clusters for lattice field theory, built and operated clusters to explore their use for lattice QCD, and provided the foundation and expertise for the present initiative.

In collaboration with the MIT Laboratory for Computer Science and Sun Microsystems, in 1996 lattice physicists began the development of highly optimized code on a Sun SMP cluster that achieved performance of 50 % of peak. Jefferson Lab and MIT have operated and optimized performance on Alpha-based clusters with Myrinet interconnects since 1999, purchased with Laboratory and University start-up funds. Jefferson Lab studied both a sixteen node cluster of single-processor alphas and an eight node cluster of dual processor machines. MIT explored a twelve node cluster of 4-way alpha SMP's. Their combined efforts thus addressed both internode and intranode optimization.

Fermilab has operated an 80 node commodity cluster for lattice QCD production since January of 2001. Purchased prior to the SciDAC grant with supplemental DOE funds plus in-kind Fermilab support, the cluster consists of dual 700 MHz Pentium III computers, each with 256 MBytes of memory, a 20 GB disk, and 256 KBytes of cache per processor. The nodes are interconnected via a Myrinet fabric, using 80 ports of a 128-port Myrinet switch.

These cluster development efforts provided essential experience in optimizing the cost performance of clusters for lattice QCD and in developing cluster software for QCD and establish a proven track record of success. The current software development program is strongly influenced by this experience.

3.3 Myrinet Clusters

The technology of choice for the first procurment of clusters beginning in March 2002 was Pentium 4 processors with a Myrinet interconnect. Fermilab and JLab undertook a coordinated program exploring different aspects of Pentium 4 clusters while providing substantial research facilities to lattice theorists. In this section, we describe the clusters, elaborate on the schedules summarized in Fig. 7, and discuss the manpower required.

3.3.1 48 Node Dual 2.0 GHz P4 Cluster

Fermilab is focusing its attention on exploration of the dual processor P4 design space, exploiting the exceptionally high performance that the SSE instructions provide for lattice QCD and the fact that for problems in which a 4^4 sublattice can be assigned to each processor, the problem fits in cache and memory bandwidth does not impede the utilization of the second processor. The first cluster built with SciDAC funds was a cluster of 48 dual processor node (lines 2-5 of Fig. 7) . Procurement occurred March-May of 2002 with installation in June, commissioning in July, the beginning of operation in August and the beginning of optimization efforts in September. Work by Holmgren, Singh, Sergeev, and Toma amounted to 0.4 FTE.

Each node contains two 2.0 GHz Xeon processors, each with 512 KBytes of cache, 1 GByte of memory and a 20 GB disk. The nodes are interconnected via Myrinet, using the remaining 48 ports of the switch shared with the 80 node Pentium III cluster. The 48 nodes were purchased with cases large enough to house both a Myrinet card and four additional gigabit ethernet cards. As discussed below, tests with gigabit ethernet cards on16 of the nodes will be valuable in assessing the performance of gigabit ethernet with dual processor nodes.

Physicists from Fermilab and the MILC collaboration are currently using the cluster for physics production. In addition to advancing the physics program of the DOE in fundamental studies of weak matrix elements, this production is providing valuable experience in running and optimizing a variety of lattice applications and in influencing the design of the national SciDAC software.

3.3.2 128 Node Single 2.0 GHz P4 Cluster

Complementary to the Fermilab effort, Jefferson Lab is exploring the single processor P4 design space, building a larger cluster than would have been affordable with dual processor nodes. Jefferson Lab installed a 128 node cluster of single processor Pentium 4 nodes with a Myrinet cluster interconnect (lines 6-9 of Fig. 7). Procurement was carried out in May -June of 2002 with installation in July, commissioning in August, the beginning of operation in September and the beginning of optimization efforts in October. Work by Akers, Chen, Edwards, Watson and computer center staff totalled 0.5 FTE.

Each node has a 2.0 GHz Xeon processor, 512 KByte cache, 512 MBytes of memory and a 20 GB disk. More than 4 terabytes of file server space is available to hold input and output files for jobs. Two additional compute nodes are mounted in the racks as hot spares, and these are used as interactive nodes when not serving as replacements. The 128 node cluster uses all ports of a single chassis Myrinet switch, and the 2 spares are directly connected by a myrinet cable to allow interactive testing of parallel jobs with Myrinet. This 128 node cluster is currently running test physics production programs to optimize its use for calculations of the large space-time lattices required to study hadron structure in the physically relevant regime of light quarks. For the present size cluster and applications of physical interest, the number of lattice sites that must be treated by each processor do not fit in cache, so the application is presently memory bandwidth limited and a second processor per node would not be cost effective for these problems.

In addition to the 128 node prototype production facility, Jefferson Lab has also installed an 8 node R&D cluster (configured as 2x2x2) to prototype gigabit ethernet mesh clusters. An additional 2 nodes are linked point-to-point for driver code development. This pair is currently on loan to MIT

where it was used to experiment with user space communications and provided the measurements discussed below that motivate the exploitation of gigabit ethernet for the next generation machine.

Once the software and operations environments are developed and stable, both the Fermilab and Jefferson Lab clusters will be opened up to the larger SciDAC collaboration and transition to production running, with occasional periods scheduled for performance testing and optimization.

3.3.3 Performance of current clusters

Since the P4 is a cache based processor, the performance of a single node is a strong function of the size of the problem each node handles (that is, the size of a single lattice site times the number of lattice sites per processor). For example, with a 4^4 local lattice using standard Wilson fermions, the problem fits in the 512 KB cache, and memory bandwidth constraints are not important. Single precision performance of a single processor node for this problem exceeds 2.5 Gflops for a 2.0 GHz processor. For larger problems (above 8^4 per processor) the problem is memory bandwidth limited, and the P4 achieves only 1 GFlop. One question to be addressed by the 128 node prototype is the extent to which the high performance for cache resident problems can be preserved across the cluster. Initial results are very encouraging: even without fully optimized communications software, for a problem in which the lattice is distributed with 4^4 sites per processor and for communications in 3 dimensions, application of the Wilson Dirac operator yielded 1.5 Gflops/processor. Part of the degradation is due to data moving out of cache for communications, and part is due to message processing overhead, which will be reduced in the coming year. This encouraging result already shows that communications does not completely destroy the high performance of cache resident calculations and motivates research to explore how high we can push the in-cache performance. In a similar fashion, the 48 node dual cluster at FNAL will be used to explore the extent to which cache performance can be achieved in a cluster of SMP nodes, where two processors are sharing the (constrained) memory bandwidth.

3.3.4 Planned 128 Node Dual 2.2 GHz P4 Cluster

As the next stage in its study of dual processor P 4 Myrinet clusters, this Fall Fermilab will acquire a cluster of 128 Xeon systems with higher clock speed (2.2 or 2.4 GHz, depending upon pricing). This cluster will allow us to test in-cache scaling on faster processors, and also to measure SMP scaling on both in-cache and out-of-cache codes and the effects of competition for the Myrinet network.

The planned schedule is shown on lines 10-13 of Fig. 7. Procurement is planned during October and November, 2002, with installation and commissioning in January, 2003, and operation beginning in February. Work by Holmgren, Singh, Sergeev, and Toma is estimated to be 0.7 FTE.

3.4 Network Research and Development

One important goal of the cluster project is to prepare for single applications running on very large clusters, in many cases with the lattice in-cache. On such systems, extreme demands will be placed on the network, including a need for high bandwidth and low latency in the presence of very high

message rates, exceeding 10^4/sec. At the same time, it is essential to our cost/performance objectives that the network not dominate the system cost. Myrinet already accounts for half the node cost for medium size systems, and its timetable for reaching very large systems cost effectively is not clear. In this section we describe joint investigations of alternatives to Myrinet by Fermilab and JLab/MIT supported under SciDAC to capitalize on new opportunities. Two research and development efforts are being undertaken to address our immediate and long term network needs: building a gigabit ethernet cluster and evaluating an FPGA based network interface.

3.4.1 Gigabit Ethernet

In this section, we describe the motivation for considering gigabit ethernet, measurements and analyses that indicate that it is more cost-effective than Myrinet, and estimates of its price performance.

Performance Measurement and Modelling

The in-cache performance of lattice QCD on current generations of commodity processors is considerably higher than the out-of-cache performance, currently differing by a factor of more than 2.5 (single node performance). This factor will grow in future years, since the clock speed of processors tends to grow much faster than the speed of the memory bus. Therefore our goal is to build systems that can run significant lattice calculations in-cache.

Preserving this high performance in the face of communications is a challenge, as shown in Fig. 8. This figure shows the single-precision performance of a key kernel in lattice calculations, multiplication of a vector by the Dirac operator, as a function of the local problem size on the Myrinet cluster at Jefferson Lab. The topmost solid line is the measured single node performance, going from 2^4 mesh points per processor on the far left to 16^4 on the far right. The drop in performance as the problem no longer fits in the level 2 cache is very clear. Note that a 4^4 problem almost perfectly fills the cache, and that the high performance cache effect persists for problems even twice as large as the cache.

The lower solid curves show the impact of turning on communications in 1, 2, and 3 dimensions. The drop in performance is due to a combination of the software overhead in dealing with communications and the need to wait for the arrival of data from adjacent nodes. Myrinet does not have enough bandwidth to support the high performance achieved on a 4^4 or smaller local problem.

The dotted lines in this graph show the performance predicted by a spreadsheet model for this problem. The spreadsheet takes into account various communications effects: the amount of data to send, bandwidth limits, latency caused by copying the buffer between host memory and the Myrinet card, the number of floating point operations to be performed, processor performance as a function of problem size (reflecting the cache effect), and performance degradation as data moves between cache and memory for I/O. It models the processor performance as a function of problem size as a trapezoidal step from in-cache to out-of-cache performance, with the left edge of the sloped step equal to the cache size and the right edge of the sloped step equal to twice the cache size (from a fit to the data with communications in 3 dimensions). Model values for bi-directional bandwidth and link latency are from low level network performance diagnotic programs. A constant overhead term is obtained by fitting the fall off in single node performance as the problem size is shunk from 4^4 to 2^4.

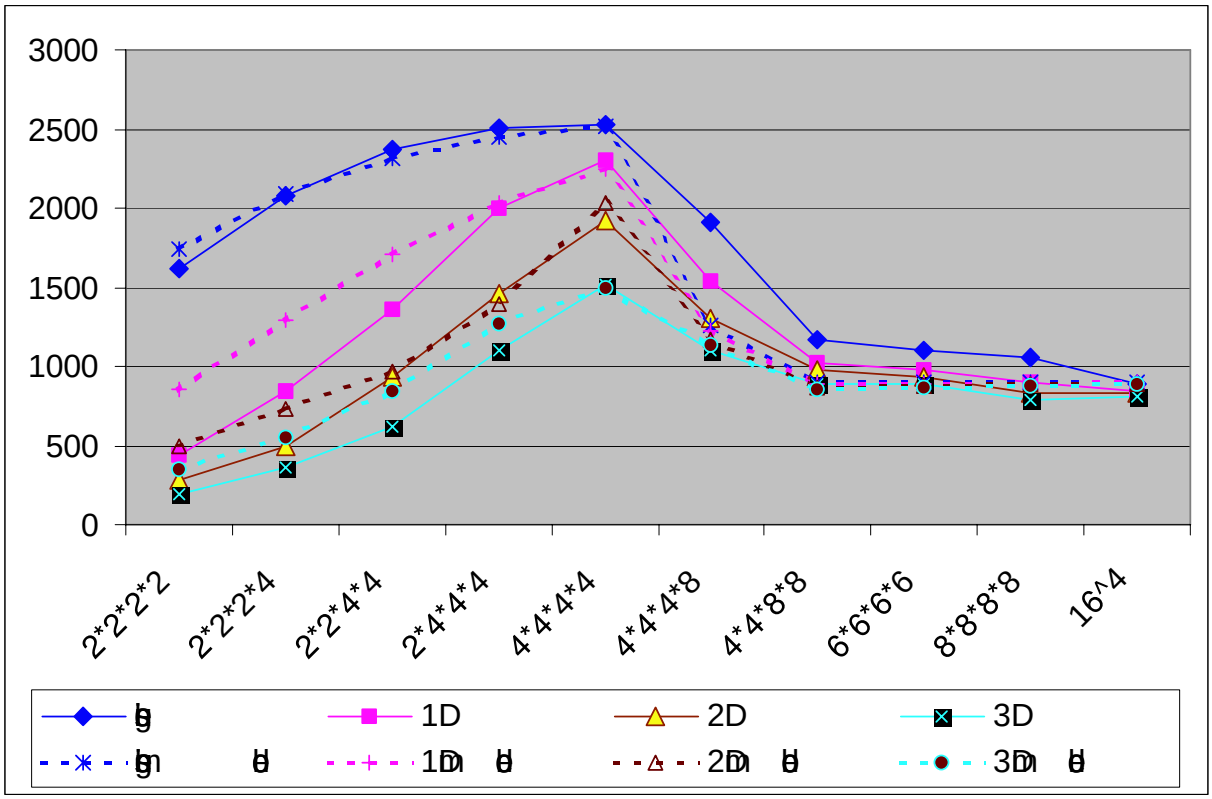

Figure 8: Measurements of the performance of the application of the Dirac opertor to a vector on a Myrinet P 4 cluster and comparison with the model described in the text.

As can be seen in the figure, the model shows very good agreement with the data. There are two notable deviations. It underpredicts the single node performance for problems just slightly larger than the cache. An excellent fit can be had for this case by setting the point at which the processor performance falls to the out-of-cache value (right edge of the trapezoid) to be three times the cache size instead of twice the cache size. Apparently communications degrades the effectiveness of the cache. We have chosen to set the model paramenters to match the (degraded) communicating performance so that we underpredict the single node performance, but do a good job predicting the cluster performance.

For large clusters, the time to execute a global sum is an important constraint. We have measured this time as a function of cluster size for 2 nodes through 128 nodes, and it behaves consistent with the B-tree summing algorithm used, with the latency set to 10 microseconds. This is consistent with the network measured 9 microseconds, plus a microsecond of adding and bookkeeping for the next step.

Since we do not yet have a fully optimized multi-processor implementation of multiplication by the Dirac operator, the potential performance of an SMP node was checked by running two copies of the single node code, again varying the problem size. For small problems (inside cache), an almost perfect speed-up of a factor of 2 was observed on a dual processor machine. For large problems, no gain was observed (each ran at 1/2 speed), reflecting that the problems is entirely memory bandwidth constrained. This data is incorporated into the spreadsheet model as a linear speedup for small problems, and a linear slope down to the memory constrained limit for large problems. (Additional data for dual processor clusters will be obtained in the next quarter.)

With the model paramenters now correctly predicting the performance of the current commodity clusters, we can run what-if scenarios to estimate the behavior of other clusters. From these studies, two parameters are clearly critical: the aggregate I/O bandwidth, and the message latency. It is clear that running lattice QCD in-cache will require more bandwidth than myrinet currently provides. Also, from a performance per dollar point of view, a lower cost network is also desireable.

Gigabit ethernet is an attractive alternative because the cost of the network interfaces has fallen considerably in the last 2 years. Large switches remain expensive, but lattice QCD is a grid based problem, so multiple point-to-point links can be used instead of a switch, an approach already demonstrated by Fodor *et. al.* [1, 2]. Multiple links also has the advantage of providing high bandwidth (at low cost).

Gigabit Ethernet Measurements

Using the pair of nodes at MIT with Netgear gigabit etnernet cards and GAMMA software, a study was completed by Pochinsky [3] that measured the send latency, sustained bandwidth, communication memory bandwidth, and processor load during communication. Two key results are shown in Fig. 9. The gigabit ethernet send latency is seen to be 2.4 μsec and the single-direction bandwidth is seen to be 110MB/sec for a single card. The true latency, or half round trip time, is 5.7 μsec and the full bandwidth is 110 + 110 MB/sec. In addition, the communication bandwidth of 220 MB/sec represents an 8% degradation of the cpu memory bandwidth of 2800MB/sec, and the degradation of a computationally intensive SU(3) matrix multiply is 19%.

Comparison of Gigabit Ethernet and Myrinet

For purposes of a go/no go decision on changing from Myrinet to gigabit ethernet, it suffices to consider a worst case analysis. We are currently evaluating cards that may well be superior

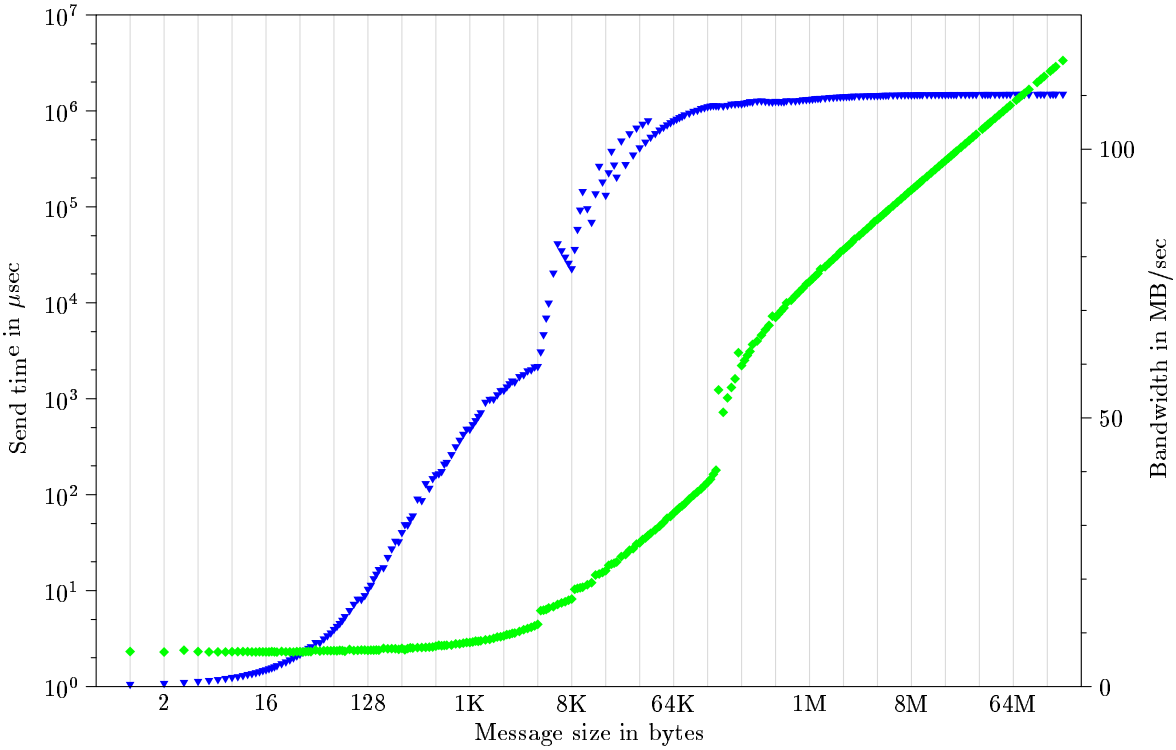

Figure 9: Performance of gigabit ethernet on a pingpong test. The send time is denoted by the diamonds and shown on the left scale and the bandwidth, defined by half the total send plus receive time, is denoted by triangles and measured on the right scale.

to Netgear and software improvements in the final user space driver should be possible relative to the performance of GAMMA. A simple back-of-the-envelope calculation shows that with the performance measurements described above, a three dimensional gigabit ethernet mesh is more cost effective than Myrinet. Consider a cache resident problem, for which the communications bandwidth is irrelevant and the processor degradation on gigabit ethernet is 19% compared with 5% for Myrinet. The true latency is 5.7 μsec for gigabit ethernet compared with 7-9 μsec for Myrinet. With gigabit ethernet, the bandwidth on 6 110MB/sec links is expected to be considerably higher than the 250MB/sec for Myrinet and the true latency is lower. Hence, with superior bandwidth and latency, the only disadvantage of gigabit ethernet relative to Myrinet is the larger processor degradation, so the worst case (over) estimate of performance degradation is 14%. However the current price per node of a Myrinet cluster is approximately \$1400 (processor node) + \$1400 (Myrinet) = \$2800, whereas with gigabit ethernet cards costing \$100, the corresponding cost is \$1400 + \$600 = \$2000, for a cost savings of 29%.

Model Analysis of Gigabit Ethernet Performance

A more detailed analysis of the performance of gigabit ethernet meshes is provided using the spreadsheet model described in the previous section and the measurements of ref. [3].

Performance of lattice codes on clusters is a strong function of the problem size per node, and a weaker function of the number of nodes for fixed problem size per node. There are many factors that compete in complicated ways, including whether the problem fits into cache (favoring a small

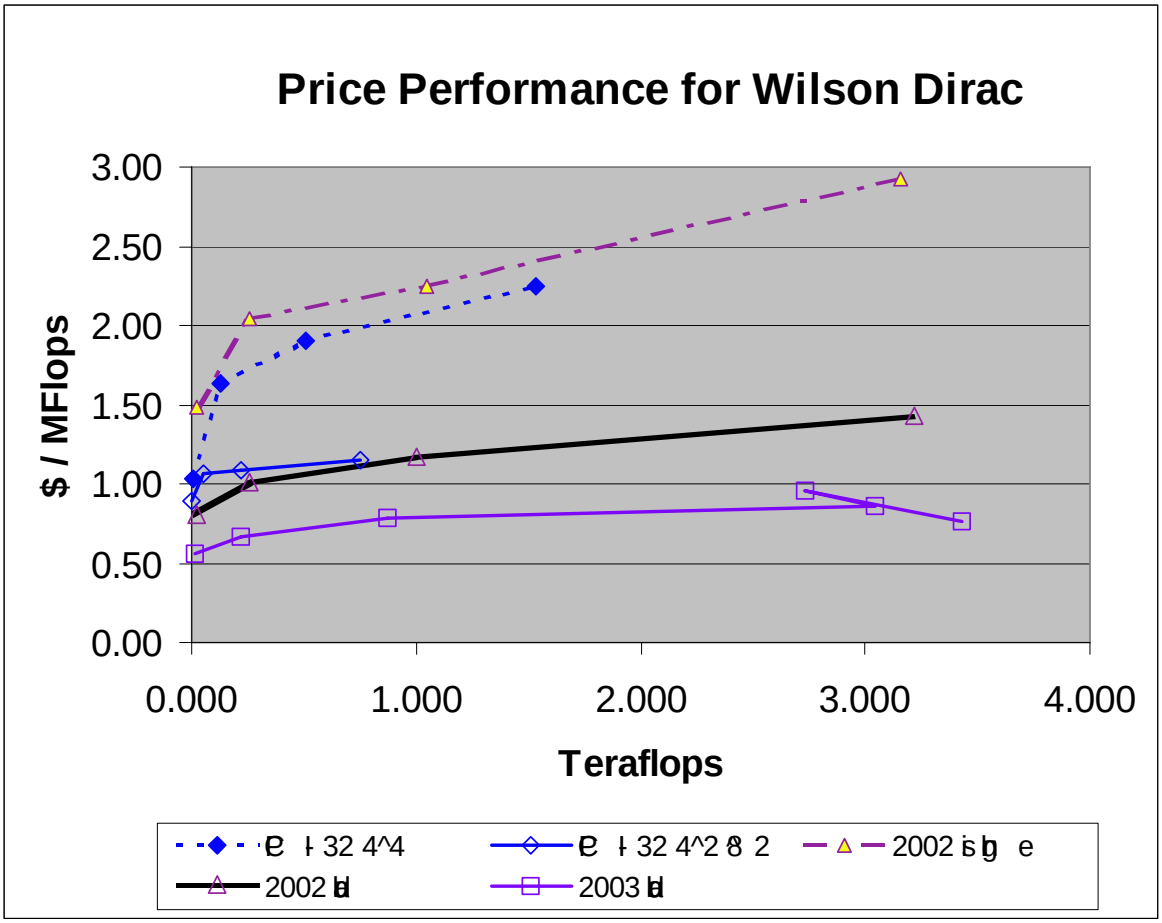

Figure 10: Projected price performance for conjugate gradient calculation of Wilson fermion propagators with gigabit ethernet using the performance model described in the text

local problem), and the surface-to- volume ratio (favoring a large local problem). Cache sizes tend to increase with each passing year, and so the optimal problem size per processor likewise changes. As already described in connection with Fig. 8, on today's processors, the "sweet spot" is at 4^4 for the Wilson Dirac operator. By 2003, cache sizes will increase 50%, and the optimal problem size will become $4^3 \times 8$ per processor.

To show how this competition of factors affects the cost performance of current and future clusters, the cost per Mflops calculated in the spreadsheet model for the full conjugate gradient single precision calculation of propagators for the Wilson Dirac operator is shown for several cases in Fig. 10. Each trace in the Figure shows a different model cluster running a fixed local problem size and a total problem given by $L^3 \times 2L$, for L=8, 16, 24, and 32. For L=8, a 2D mesh is used; for L=16 and 24 a 3D mesh is used, and for L=32 a 4D mesh is used (except for the PCI-32 traces, which can't support a 4D mesh).

The trace with open diamonds shows a cluster of inexpensive PCI-32 nodes running with 4x4x8x8 per node. Such clusters have received a great deal of international attention since Fodor *et.al.* [1, 2] demonstrated that they could actually be built for less than 1$ / Mflop. The trace with solid

diamonds shows what happens if one attempts to run this cluster at 4^4 per node. For such a cluster, the performance for a given L is roughly a factor of 2 higher, but the cost is a factor of 4 higher, so it is not a good choice for production running. These consumer systems are very I/O constrained, and so do not operate efficiently on local lattices smaller than $4^2 \times 8^2$ per node. Note that in the work of Fodor *et.al.*, even larger local lattices (8^4) were used to push the cost per megaflop below $1, but with the result that the performance overall for a given physics problem was fairly low.

The next two traces to consider are the 2002 single processor (triangles with dot-dash line) and 2002 dual processor (triangles with solid line) PCI-X based clusters. These systems are more expensive per node, but have much better I/O capability. Observe that only the dual processor system is predicted to be cost effective for cache resident problems, a prediction that will be tested on the new cluster to be installed at Fermilab this Fall. (There may be some issues with dual processors that the current spreadsheet does not correctly model. For production running on the current prototype clusters, the problems don't fit into cache, and so single processor systems are more cost effective. The prototype duals are primarily for cache resident scaling studies).

The lowest trace (open squares) shows predictions for dual processor P4 clusters in 2003, where the memory sub-system will be 533 MHz instead of 400 MHz, and clock speeds will also be higher. However the most significant improvement will be the larger cache (0.75 MB), which will support cache resident running at a more favorable surface-to-volume ratio. The model is most sensitive to the message latency and overhead, and as a way of showing the uncertainties due to this parameter the last trace shows the effect for L = 32 of varying the latency and overhead together between 6 usec and 10 usec.

These calculations look very encouraging, and the SciDAC funded developments this coming year will allow us to validate these predictions, and further refine the model for future year predictions.

3.4.2 Gigabit Ethernet Cluster

Based on the results described above, our current plan is to build a 256 node cluster with a three-dimensional gigabit ethernet mesh at Jefferson Lab. (lines 14-22 of Fig. 7). At present, we expect the machine will use dual processor P4's if tests of dual processors at FNAL confirm good use of the second processor on cache resident problems, will have a 533 MHz memory sub-system, and will be configured as an 8x8x4 3D mesh. This machine will allow a somewhat larger set of applications to run in cache, and will allow prototype scaling of gigabit ethernet meshes to large sizes.

The preliminary evaluation described above has been in progress since March, 2002, and we plan to continue evaluation and testing of hardware through November 2002. Assuming hardware is available that meets the expectations described above, the first version of the user space driver and QMP for gigabit ethernet will be developed November through February 2003, procurement will begin in January, installation and commissioning are planned for March, operation is anticipated for April, and optimization of the second release of the driver will occur March through May. This work will be performed by Akers, Chen, Edwards, Pochinsky, and Watson, corresponding to 1.25 FTE.

3.5 FPGA based Network Interface

In order to support very large clusters, it may be necessary to reach performance levels not achievable by commodity network cards. Furthermore, it is desirable to have network features that are unavailable in gigabit ethernet, such as hardware support for a global sum, and for non-nearest neighbor communications. Therefore, we propose to take advantage of a timely opportunity to explore use of a high performance custom network interface that takes advantages of the simplifications of lattice QCD. A prototype of such a custom network interface card (NIC) is being developed at FNAL as part of a data acquisition system development activity and Fermilab will produce a small number of these first generation cards for evaluation by our lattice QCD collaboration. This card uses a recent Lattice/Orca ORT82G5 FPGA, and provides eight 2.5 Gbit links with 8/10 encoding on a PCI-64/66 card. This performance level allows effective prototyping of the technique. If this development is successful, it would lead naturally into the production of a high performance PCI-X card in FY04.

To take advantage of this important opportunity, which arose after initial SciDAC funding, it will be necessary to develop custom firmware, user-space driver code, and QMP implementation code. Assuming the hardware development proceeds appropriately, the custom firmware will be developed using 0.7 FTE of an electrical engineer contributed by Fermilab and redirecting 0.3 FTE of SciDAC support. If the firmware development is successful, 0.6 FTE of resources from the current SciDAC grant at Fermilab will be redirected to develop the user-space driver code. If the firmware and user-space driver are successfully developed, Jefferson Lab will provide the necessary time by Jie Chen, who is already experienced in QMP development, to perform the QMP implementation. The schedule, is shown on lines 23-28 of Fig. 7. Note that there are decision points after the conclusion of low level communication tests in December 2002 to decide if the firmware should be developed, after the FPGA programming is complete in March to see if the driver should be developed, and when the driver is complete in May to see if the QMP port should be performed.

3.6 Next Generation Clusters

The final stage of procurements utilizing SciDAC funding is planned for the Summer and Fall of 2003. We plan to continue our successful pattern of staggered acquisition to obtain the most complete access to rapid developments in the marketplace.

At present, the most promising candidate CPU architectures for the Summer and Fall FY03 cluster prototypes are Itanium 2 and its successor, “Madison”, the AMD 64-bit chips known as “Hammer”, and the next generations of Intel Xeon and Pentium 4. As with all cluster procurements, there is a great deal of schedule risk associated with the production schedules of new processors, as well as with the suitability of their support chipsets. Frequently a given combination of chipset and processor are unsuitable for lattice QCD because memory bandwidth, i/o performance, and floating point capabilities are badly out of balance. For example, the original Intel Xeon chipset, i860, exhibited very poor PCI bus performance; consequently, we delayed acquisition of a Xeon cluster until the next generation chipset, E7500. Because of this schedule risk, the dates shown may shift considerably.

Synthetic benchmarks run on 900 MHz Itanium 2 systems have shown performance boosts in main memory of 300-400% over the current generation of Xeon processors running at 2.0 GHz. How-

ever, detailed, low level code optimizations were necessary to achieve this level of performance. If compilers for this processor family do not improve, or if the low level optimizations prove too labor intensive or difficult to maintain, we will build the FY03 clusters with one of the other architectures.

The other 64-bit architecture expected in the time frame of this procurement is the "Hammer" family from AMD. Few details about performance or compiler availability are currently known. As a fall-back, we could choose one of two new generations of Pentium 4 processors planned by Intel - SMP capable Xeon CPUs with 533 MHz memory buses, and uniprocessor-only Pentium 4 CPUs with 667 MHz memory buses.

3.6.1 Summer 2003 Cluster

The first of these next-generation clusters will be built at Fermilab, and the planned schedule for this cluster is shown in lines 14-22 of Fig. 7. The primary decision point is the processor decision at the end of April 2003, procurement would occur in May and June with installation in July, commissioning in August and operation in September. Holmgren, Singh, Sergeev, and Toma would provide a total of 2FTE's of effort on this cluster.

3.6.2 Fall 2003 Cluster

The second cluster of this generation will be built at Jefferson Lab approximately three months later as shown in lines 35-39 of Fig. 7. The network and processor choice will be influenced by the experience with the JLab GigE cluster and the outcome of the NIC development project, as well as the evolution of processor technology. It is our intent to coordinate the Fermilab and Jlab acquisitions for the maximal benefit of the overall project. Our present plan is to decide on the technology by the end of July 2003, carry out procurement in August and September, install hardware in October, commission in November, and begin operation in December. Akers, Chen, Edwards, Pochinsky, and Watson are expected to provide 2 FTE's of effort.

3.7 Operating and Batch Systems

The clusters at Fermilab and Jefferson Lab run RedHat Linux, and on both systems, a number of important operating system and hardware management tasks are automated. Extensive use of network booting allows automated, unattended, and parallel installation of system images, as well as firmware updates. Each computer can be remotely reset or power cycled via commands given over serial lines or ethernet connections, independent of the state of the operating system. Software monitors each node for health, alarming on high temperature conditions and fan failures.

The batch system for the clusters is Open PBS. Fermilab is exploring use of the Maui scheduler, and the Fermilab Enstore mass storage system provides access to robotic tape systems. Enstore gives users a transparent interface to data stored on tape, with individual files appearing as directory entries in a standard Unix file system.

Jefferson Lab is currently exploring the use of a scheduler (Underlord) they have developed. In addition, a web services based data grid provides tools to manage a distributed set of files in a

single namespace. This software, developed at Jefferson Lab, leverages the lab's participation in the Particle Physics Data Grid Collaboratory (DOE SciDAC program). Jefferson Lab and MIT are working together to test this software, and drive the deployment of a meta-center that will provide seamless sharing and access to data and computational resources at both sites. A more complete description of this software, and how these capabilities will play a part in the LQCD distributed terascale facility, is given in the software section below.

3.7.1 Cluster Facilities Administration and Operation

Fermilab and Jefferson Lab have operated production lattice QCD clusters during the last two years. Consequently many administrative tools and procedures required for operations as user facilities are already in place, including:

- written procedures for obtaining accounts for off-site users.
- accounting tools for generating cluster usage reports.
- access from the "home" node to tape mass storage facilities.
- scripts for users to call from batch jobs to correctly launch lattice QCD codes.
- web-based displays of cluster status, including batch queues, node utilization, and hardware status (temperatures, fan speeds).
- alarms to administrators based on hardware and software conditions.
- software tools to issue commands in parallel to groups of nodes, including nodes belonging to a specific user job.
- software tools to replicate files to groups of nodes.
- software tools and procedures for automated operating system installation and configuration, and for automated BIOS and other firmware installation.
- user and administrator mailing lists and web based archives.
- dispatching of calls for assistance from users to personnel responsible for administration.

In order to support the full range of envisioned SciDAC operations, the following tasks will be accomplished in the coming year:

- transition of day-to-day administrative tasks to administrative groups. This will require documentation and training.
- modification of accounting procedures and batch systems to support the cluster allocations assigned by the Science Committee.
- expansion of support. Full production facilities should eventually provide 24x7 support for simpler problems, and 8x5 support for difficult issues.

- further addition of disk space for data.
- as needed, procurement of more capable ”home” nodes.

References

[1] Z. Fodor, S. D. Katz and G. Papp, arXiv:hep-lat/0209115.

[2] Z. Fodor, S. D. Katz and G. Papp, arXiv:hep-lat/0202030.

[3] A. Pochinsky, “GigE and Xeon” http://www.mit.edu/~avp/lqcd/GigE/report.pdf

4 SciDAC Software Infrastructure Project

4.1 Overview

The goal of the U.S. SciDAC software infrastructure project is to create a unified programming environment to achieve high efficiency on the multi-Terascale computer architectures targeted in this plan. By the creation of standards for communication interfaces, optimized low level algebraic kernels, optimized high level operators and other run-time functions, the valuable U.S. base of application codes can be easily ported and extended as the computer architectures evolve over time, without wasteful or duplicative effort of the theoretical physics community. Further, in light of the two architecture approach for a distributed topical center for QCD, it is essential to be able to rapidly move application codes between hardware platforms in order to balance the load and optimize performance system wide.

The focus of the first year of this project has been to define the overall structure of a QCD Application Programming Interface (API), and to flesh out the first two critical components of that API: a highly optimized communication (message passing) layer (QMP), and a linear algebra layer (QLA). The QCD API, illustrated in the block diagram below, provides the software matrix within which many of the other tasks are found. The full range of software development as define in the SciDAC proposal comprises six major tasks:

1. Design of QCD API and code libraries
2. Optimization of network communication
3. Optimization of QCD kernels
4. Porting and optimization of application codes
5. Development of data management and data grid tools
6. Development of the execution environment

A more detailed breakdown into subtasks is given in Sec 4.2.6 below. Published standards and code releases are posted at the URL www.lqcd.org. Although the SciDAC software infrastructure will not be completed until the end the SciDAC grant, priorities have been set so that the critical components will be ready in time to fully support the QCDOC developoment machine and clusters coming on line in FY03.

QCD-API Level Structure

Level 3
Dirac Operators, CG Routines, etc. (Optimized Plugins for critical sections)

QDP_xxx	Level 2
Data Parallel API: QCD Lattice Operations (overlapping Algebra and Messaging) e.g. A = SHIFT(B, mu) * C	
Lattice Wide Linear Algebra (No Communication) e.g. A = B * C	Lattice Wide Data Movement (Pure Communication) e.g. Atemp = SHIFT(A, mu-dir)

QLA_xxx	Level 1	QMP_xxx
Linear Algebra API: Single Site & Vector e.g. SU(3), Dirac algebra, ...		Message Passing API: (Maps Lattice Geometry into Network)

4.2 Integration of software and hardware development

Our plan for FY03 requires a careful co-ordination between software and hardware development. The central software components are:

1. Optimized message passing layer (QMP)
2. Optimized core low level linear algebra routines (QLA).
3. High level Dirac inverters optimized for QCDOC and clusters.
4. Standardized I/O routines and data file formats.
5. Performance milestones for core routines and full application codes.

An overview of the schedule for software development in FY03 is given in the Gantt chart in Fig 11. The synchronization of software development tasks in FY03 with hardware deployment proceeds in two phases.

The first phase, begun September 2001 using the QCDOC ASIC simulator and small clusters, is the optimization of Level 1 code (QMP message passing and QLA linear algebra routines). These software components will be fully benchmarked, optimized and hardened first on the prototype

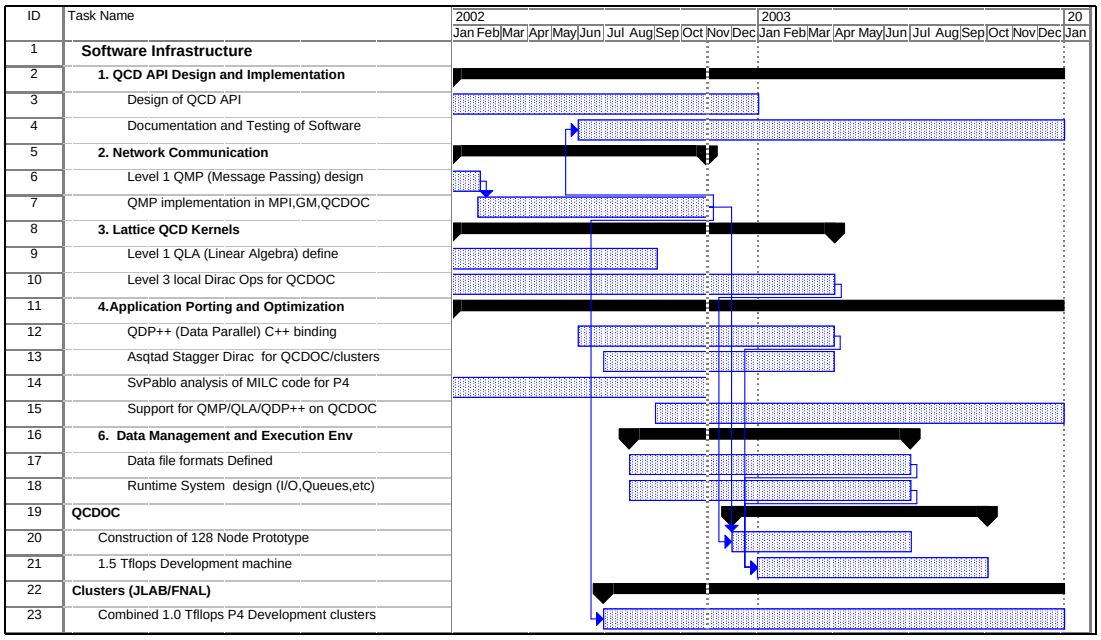

Figure 11: Overall schedule for SciDAC software in relationship to the deployment of QCDOC and cluster development machines in FY 03.

clusters at JLab and FNAL, and then on the 128 node QCDOC prototype, with both platforms operating by June 1, 2003.

The second phase from June 1, 2003 to Dec 31, 2003 requires efficient full scale application codes to do production work on the 1.5 Tflops QCDOC development machine and the 0.25+ Tflops clusters. This necessitates that the major large scale applications designated by the Program Committee be ready to run by June 1, 2003 and that additional applications be developed as needed. In addition to enabling very significant physics production work in FY03, this phase will allow for further optimization/testing/evaluation of SciDAC software and evaluation of the programming and execution environment required for a national center in lattice QCD with several multi-Terascale production machines.

4.2.1 Network Communication

It was clear from the beginning of the project that the message passing API (QMP) represented the single most time critical component. This layer 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: (i) QMP 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); (ii) specialized routines are 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. The initial stage for the documentation and implementation of the message passing layer (QMP) has been largely the responsibility of Watson, Edwards and

Chen at JLab in collaboration with Mawhinney and Chulwoo Jung at Columbia/BNL, who are assuming the task for the native QCDOC implementation. The QMP definition, documentation and code are on the project website *http:://www.lqcd.org*. The released code includes a pure MPI implementation for portable code development, optimized implementation for Myrinet (in native GM code) and a native implementation for QCDOC, whiich has been tested on the ASIC simulator. Note that the design, documentation, implementation and testing procedure for QMP has set the framework and methodology for all the code modules that follow.

By itself, this message passing infrastructure (QMP) permits one to quickly port C or C++ application codes to run on the QCDOC and the clusters. Once these codes are linked to optimized Level 3 Dirac inverters, reasonably efficient physics production running becomes possible. However, there are several additional tasks needed in this area during FY03. On the QCDOC, optimized routines for non-nearest neighbor communication are still needed for the non-local Dirac operator, Asqtad. We anticipate the development of many non-local operators which will employ these extended communication kernels. A test code suite for QMP is being designed by Mendes at Illinois and Simone at FNAL for all implementations, present and future. As described in the Commodity Cluster Sec 3.4 and 3.5, if, as is expected, the GigE and the FPGA NIC networks prove to be more cost effective than Myrinet, we will work on QMP implementations for these networks under our SciDAC grant.

4.2.2 Optimization of Linear Algebra

The linear algebra routines (QLA) and their optimization on the Pentium 4 are primarily the responsibility of DeTar, Pochinsky, Holmgren and Osborn. The ”streaming SIMD” features of Pentium processors, known as the SSE and SSE2 instructions, are essential for high performance. By coding the fundamental QLA routines using SSE instructions on a 1.7 GHz Pentium 4, the single node performance in single precision for the Wilson Dirac operator in cache exceeds 2.5 Gigaflops. The full library for QLA routines is being generated in C using software tools in Scheme and Perl. Upon completion of this task by Jan 1, 2003, the optimized SSE library will replace core routines in QLA to enhance the performance on the Pentium 4 clusters. Implementation of the QLA routines on QCDOC is under development by Mawhinney, Jung and Petrov with special emphasis on optimizing of the core routines. Analysis of the impact on performance and practical pursuit of higher effeciency at this level is a major task prior to June 1, 2003, when the QCDOC development machine is to begin operation.

The Scientific Program Committee has identified three quark actions as being essential for early work on Terascale computers: Wilson clover, domain wall, and improved staggered (Asqtad). The first two of these have already been hand coded for the QCDOC, as has the conventional staggered quark action. These Dirac inverters are also being optimized for clusters as part of the code development work at JLab and FNAL. The Asqtad action, which is a less local version of the conventional staggered action, is also required. Asqtad is being implemented by a joint effort of SciDAC developers in Columbia and Utah. Gregory at Utah has adapted the MILC Astad operator to use the QMP routines and Chulwoo at Columbia is running this code on the QCDOC ASIC simulator. At the same time the Asqtad fermion force term is being written by Gregory in the C implementation of QDP. Once this stage is completed, both the Dirac operator and the force term will be optimized as Level 3 QCDOC modules callable from any C or C++ application code. The optimization of the Asqtad inverter is a priority which will be completed by the end of the first

quarter of 2003 in time to be tested on the prototype QCDOC, and to run production code on the QCDOC development machine by June, 2003.

This Asqtad implementation is a typical example of the collaborative process between experts in low level code optimization and the general lattice QCD community, which the SciDAC software project seeks to establish as the norm. Clearly there will be an ongoing need to develop optimized plug-in (Level 3) routines in order to adapt quickly to new fermion and gauge actions for QCD, and similar quantum field theories that are needed to go beyond the Standard Model. Thus, a critical design objective of the software infrastructure project is to combine rapid prototyping in the convenient QDP (data parallel) interface with the subsequent coding of assembly language modules for the QCDOC and clusters when new algorithms are required for large scale production codes. However it should be noted that the overwhelming majority of application code (by line count) can remain in QDP (or even legacy C or C++) without substantial impact on the throughput of simulations.

4.2.3 Data Parallel Framework and Porting Application Codes

QCD applications have the fortunate characteristic that they have a natural framework well suited to a data parallel interface, and a small number of linear algebra operations that must be optimized to obtain excellent performance, the inversion of the Dirac operator being the most notable. Consequently it is possible to write the vast majority of application code (by line count) in a uniform data parallel language. The SciDAC software project has defined a full data parallel QCD interface (QDP) to promote a community wide standard for code development. The C and C++ implementations for the Level 2 QDP interface will be available in serial form by January 1, 2003 so that software developers can use them to port legacy codes, or write new application codes. A major task for software development starting in January, 2003 is to produce a robust parallel implementation of QDP by June, 2003 so that code written to this interface will run on all SciDAC prototype hardware.

The design objective for QDP is to enable new applications to be developed rapidly and to run with efficiency beyond the reach of generic C or C++ codes. Edwards, Mawhinney, Petrov and DiPierro are implementing QDP in C++, whereas DeTar and Osborn are implementing QDP in C. Both implementations conform to a single API and pains are taken to build them on a common set of lower level (Level 1) fundamental C functions. Much of the underlying development and optimization can be shared between the C and C++ variants. It is expected that in most instances legacy code will be migrated to the QDP only on a case by case basis, but new application code will adhere to the full API.

Obviously, the principle responsibility for porting application code to the API interface falls largely on the authors of the particular code base: SZIN (Edwards, Richards, Pochinsky), MILC (DeTar, Osborn, Toussaint, Gregory) and CPS (Mawhanney and co-workers). However, the Software committee acts to review progress and to promote a uniform methodology and shared optimization strategies to minimize duplication of effort. Harmut Neff, who begins as a SciDAC postdoctoral fellow in Nov 1, 2002 at BU, will focus much of his effort on providing developers with example code and documentation to support this goal.

4.2.4 Data Management

Task 5 also has essential elements that must be defined before 2003. The data file formats and headers will be defined in the last quarter of 2002 by adding I/O routine to the Level 2 API standards so that all application codes can share data files. The Software Co-ordinating Committee is actively working on this task with separate teams assigned to address the programming interface in QDP (Edwards, Pochinsky, Brower), File Formats (Mawhinney, DeTar) and Web access and archiving (Watson, Holmgren). Simone and Edwards are representing the project in the international data archiving project proposed by Richard Kenway of the United Kingdom.

This project is also studying the use of XML, both as a tool to control data file I/0, and to locate relevant data using web based search methods employing XQL. Subsequently there will be a gradual evolution to a lattice portal with multi-site replicated data for security and rapid retrieval. These tools should allow a natural exploration of web based computing, including batch systems and uniform scripting tools. Chip Watson at JLAB is an active member of the SciDAC Particle Physics Data Grid project, working with others to develop and deploy advanced data grid software. In addition we are actively participating in the proposed ILDG (International Lattice Data Grid). These connections will enable us to deploy, rather than re-invent, grid tools as they become available. Obviously this ambitious program needs to be carefully considered and prioritized, so that practical extensions are brought on line without losing focus on the immediate needs of the physics program. This will become a higher priority in FY04 to further facilitate operation of the combined computing resources at BNL, JLab and FNAL as a single topical center for QCD lattice research. Such a distributed meta-laboratory is clearly an ambitious project, but the relatively narrow focus and maturity of the QCD computational program make it an ideal testbed for more general grid based computing facilities.

4.2.5 Execution Environment

The Execution Environment is a another important area of software development for this project. This category of software includes a number of application modules beyond those previous listed, as well as the batch and grid environments in which the applications run.

The application run time environment includes error detection and reporting, and standards for application (data) check pointing procedures. Because each platform must give bitwise reproducible results, global sums and random number generators will be rigorously deterministic on all networks.

Standards for the batch environment will allow not only applications, but also production batch scripts to be moved quickly between the different platforms. This implies a certain amount of standardization of file system naming conventions (or discovery), and standardization of batch submission commands and options. Declaration of input and output dataset names will eventually use global file names, names which are independent of the site at which they are located.

Uniform file format standards and compatible batch environments are precursors to the larger goal of putting in place a web based (or data grid) system to allow the three laboratories to be viewed as a single topical national facility. This has substantial advantages. It allows both load balancing and optimization by directing jobs to the architecture on which they will run most cost effectively.

4.2.6 Summary of Software Infrastructure subtasks through FY03

Below we give a summary of the progress to date on the software tasks and projected completion dates through FY03 for work in progress. In this table "on going" refers to subtasks active through out FY03, and continuing as part of software development during the life cycle of each platform. Each sub task is assigned to a team, and reviewed by the entire Software Coordinating Committee in weekly teleconference meetings and quarterly workshops. A chronology of the minutes of all meetings and working documents can be accessed through the web page *http://physics.bu.edu/ brower* whereas released standards and code are publish on *http://lqcd.org*.

Task/Subtasks	• Start Date	• Final Date
1. QCD API and Code Library:		
1.1 Overall design of QCD API	• July 01	• Dec 02
1.2 Compliance, Testing and Benchmarking in Application Codes	• Jan 03	• on going
1.3 Code Libraries and Documentation	• June 02	• on going
1.4 Data Analysis Tools defined and code archive	• Jan 03	• on going
2. Network Communication:		
2.1 Level 1 QMP (Message Passing) design (done)	• July 01	• Feb 02
2.2 QMP implemented on top of MPI, GM, QCDOC (done)	• Feb 02	• Sept 02
2.3 Fat SMP extensions using threads	• (when required)	
2.4 Alternate Networks (e.g. GigE/ FPGA-NIC/etc)	• May 02	• on going
3. Lattice QCD Kernels:		
3.1 Level 1 QLA (Linear Algebra) fully defined	• Sept 01	• Sept 02
3.2 QLA library issued in C with SSE opt for P4	• July 01	• Dec 02
3.3 Level 3 Local Dirac Operators for QCDOC/clusters (done)	• July 01	• June 02
3.4 Asqtad Staggered Dirac & force terms for QCDOC/clusters	• July 02	• May 03
4. Application Porting and Optimization		
4.1 C binding for QLA & QDP	• Sept 02	• Feb 03
4.2 QDP++ (Data Parallel binding to C++) for QCDOC/clusters	• June 02	• Feb 03
4.3 Parallel Support for QDP on QCDOC/clusters	• Jan 03	• Aug 03
4.4 Analysis of MILC code using SvPablo for P4	• July 01	• Jan 03
4.5 Optimization of Application Code on QCDOC/clusters	• Sept 02	• on going
5. Data Management and Data Grid		
5.1 Data File Formats Defined	• Aug 02	• April 03
5.2 Grid Work (w. International Lattice Data Grid (ILDG))	• July 01	• on going
6. Execution Environment		
6.1 Run-time environment: I/O, Error detection standards	• Aug 02	• June 03
6.2 Define standards for Batch Environment	• Aug 02	• June 03
6.3 BIOS , Chipsets, Network booting, Machine configuration,..	• April 02	• ongoing

The SciDAC grant in FY03 supports QCD software infrastructure at the level of 9 FTE. These resources are complemented by roughly an equal number of physicists and software engineers with no direct SciDAC support. Thus the size of the group actively participating in the SciDAC software infrastructure work is roughly 20 individuals. The seven member Software Co-ordinating Committee meets weekly in conference calls to develop the overall design strategy. Subgroups take particular responsibility for separate tasks with intermittent participation of others to insure the unity of the overall design.

The software co-ordinating committee sets the deadline for deliverables to insure that critical components are available in time to keep the overall software and hardware project on track. On occasion this means reassigning effort in "burst mode" to avoid unanticipated delays on the critical path. This *modus operandi* has worked very well to date and we are confident that we have the means to deliver the software infrastructure as set out in the SciDAC grant.

4.3 Tests of Software Development and Final Evaluation

Our project follows an aggressive schedule of development and construction, which will take advantage of a unique window of opportunity to bring the U.S. lattice community back to the forefront of worldwide research. In order to reap its full benefits, it will be crucial to insure that the development of application specific software, as described in the previous section, meshes well with the development of hardware. Once the construction of the 1.5 Tflops QCDOC machine and of the prototype clusters have been finished, it will also be important to run extensive tests of complete application packages, to validate the suitability of such special purpose computers for lattice QCD calculations and lay the ground for the construction of dedicated 10 Tflops machines, as envisaged by our long range plan. For these purposes, we have devised a set of tests which will be conducted both during the development of our project and at its end, and which will be referred to as "software development tests" and "final milestones". The goals of these tests are quite different. The software development tests will be conducted in parallel with the construction of the QCDOC prototype and development machine, and of the clusters. Indeed, some are already being run, on the simulator in the case of the QCDOC. They will serve to make sure that the software development is on course, providing diagnostics for any aspect of it which may require remedies or extra efforts, so that full fledged applications may be run as soon as the development QCDOC and clusters are completed. The final milestones will involve running complete QCD application codes and will serve to validate the efficiency and flexibility of the architectures, as well as their ability to sustain the efforts of the diverse U.S. lattice community.

In making these tests we will take advantage of the existence of three large, well established code packages, which are freely available to the entire U.S. lattice gauge theory community: the Columbia Physics System (CPS), the MILC code and the SZIN code. Each of these codes runs on single processor workstations. The MILC and SZIN codes have been validated and have achieved excellent performance on a wide variety of massively parallel computers, while the CPS has been validated and achieved excellent performance on the QCDSP, the predecessor of the QCDOC. Thus, we are in a position to test the correctness of new hardware at both the single and multiprocessor level, and to obtain a broad set of comparative benchmarks. The three codes include among them nearly all applications and algorithms in common use by lattice gauge theorists. We

are thus convinced that their performance will be indicative of what will be possible on a wide range of lattice QCD applications with C or C++ codes conforming to the QCD API.

The ideal metric for judging hardware would be dollars per unit of science, but that cannot be measured directly. For example, several different actions are used to describe quarks on the lattice, and more are under development. The choice among them often depends upon the problem under consideration. Quark actions differ widely in the total number of floating point operations required to invert the Dirac operator, the most computationally intensive part of any lattice gauge theory calculation. They also differ in performance, as measured in Mflops per processor because of differences in the balance between floating point operations and data movement. Nevertheless, cost per sustained Mflops is a very important measure, once such differences are taken into account. The size of problems to which a platform can be applied, and the human resources required to create or modify code are also important considerations. The former is crucial for the large projects which are at the heart of our research, and the latter for the development of new algorithms and calculational methods, which is also a crucial component of our work. It is possible that no single architecture is optimal by itself for the entire range of jobs we envision. Thus, judgments regarding the suitability of hardware for our research must be based upon a spectrum of considerations. Final decisions will be made by the Executive Committee with advice from the Software, Scientific Program and Oversight Committees.

4.3.1 Software Development Tests

These tests will be performed throughout the design and construction of the QCDOC prototype, the 1.5 Tflops QCDOC development machine, and the prototype clusters. As mentioned above, they will serve to insure that the software developed under the auspices of our SciDAC funded project will meet the needs of the lattice community. They will also serve to provide additional diagnostics for the development of machine specific software, such as the run time operating systems. The tests will consist of the following:

- Using C or C++ compilers and the QMP communications libraries, compile complete production codes for the three quark actions specified by the Scientific Program Committee: clover Wilson, improved staggered and domain wall quarks. The CPS, MILC and SZIN source codes will be used for these tests. The compiled codes should correctly evolve full QCD systems using hybrid Monte Carlo and molecular dynamics algorithms, including standard measurements such as hadron propagators, yielding answers in agreement with code on conventional hardware platforms. Tests will be done on lattice sizes per processor such that, when scaled up to the anticipated full machine or usable subsets of the full machine, they span the range of lattice sizes needed for both high temperature simulations (small lattices of size $18^3 \times 6$) and for studies of pion cloud phenomenology at zero temperature (large lattices of size $32^3 \times 96$). These tests will be done, to the extent possible, on the QCDOC simulator, to validate the correctness of the results, and to obtain estimates of the performance on the actual machine. They will be repeated on the machine components, as they become available (single node, motherboard, partial or full prototype machine). The target is to achieve at least 25% of the designed Megaflops/dollar for the updating part of the above codes on lattices with $8^2 \times 16^2$ or fewer lattice sites per processor, optimizing only at the C or C++ level. (Targeted performance figures are for Wilson based quark actions. Actions based on

staggered quarks are expected to run approximately 30% slower). The ability to read and write lattices to disk should be demonstrated.

- The second test is to prepare and run (Level 3) optimized inverters for the three quark actions specified by the Scientific Program Committee. The inverters should be compatible with the corresponding suites of production codes, although it is permissible to reformat the data before and after the inversion. All of the inverters should achieve a performance of at least 80% of the designed Megaflops/dollar with the allowance for the difference between Wilson and staggered based quark actions noted above. The performance measurement must include communication and global sum overheads. These tests may be performed on lattices of limited size, but the scaling must be well enough understood to convince the Executive and Oversight Committees that scaling to a sustained Teraflops is achievable.

4.3.2 Final Milestones

The 1.5 Tflops QCDOC and 0.25 Tflops clusters must pass a rigorous set of milestones prior to the the deployment of large terascale platforms at the BNL, JLab or FNAL. These milestones will be used to test the ability of the hardware to advance our research, and to determine the appropriate mix of hardware for the national center for lattice gauge theory. The details of these milestones will be set out by the Executive Committee in consultation with the Program, Software and Oversight Committees. Their general form will be as follows:

- The milestones must demonstrate that both hardware and software are ready for multi-terascale deployment. Entire applications in the CPS, MILC, and SZIN codes, covering the full range of our research program, will be run and benchmarked. Tests will compare the performance of specially optimized (Level 3) inverters described in the previous subsection with that of inverters coded in pure C/C++ on top of optimized Level 1 routines, in the full data parallel API (Level 2). The maturity and stability of the runtime environment for data management, job queues, remote access, etc. will be evaluated. Statistics will be compiled for sustained production runs over a period of at least one quarter to measure overall throughput. The entire user community will be polled to ascertain the experience of users not intimately involved in either the hardware or software design.

- A measure will be made of the overall development environment for new and exploratory software that inevitably makes up a substantial fraction of lattice QCD applications in a research environment. The ongoing cost in terms of time and effort for the optimization of the critical components of production codes will be included in this assessment. The goal is to be sure that the computational environment allows for full access of the U.S. lattice gauge theory community of some 200 members, and that it fosters innovation in algorithms and new applications to particle physics phenomenology which are essential for progress.

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 (Fer-

milab), 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 three 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. He submits a quarterly progress report to the DOE. 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:

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.

Scientific Program Committee: The Scientific Program Committee monitors the scientific progress of the project, and provide leadership in setting new directions. It organizes an annual meeting of all lattice gauge theorists working on or planning to participate in the project to review progress and plan future directions. It has determined the use to which the collaboration's access to DOE supercomputers has been put, and it will allocate time for major projects on computers built by the collaboration. 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).

5.1 Project Manager

The construction and operation of terascale computing facilities constitutes a major increase in the scope of this project, which currently entails SciDAC funded software development and hardware prototyping. These added responsibilities will necessitate the addition of a full time project manager to the management team. The project manager will work with each site involved in the development and deployment of hardware and software infrastructure to facilitate achieving the goals of the project. She/he will be responsible for ensuring that the following specific tasks are accomplished:

- preparation of detailed planning documents for the project, which include a set of project milestones and an hierarchical list of tasks, with each task defined at a level that can be externally reviewed and with the sites and individuals responsible for each task clearly set out
- preparation and approval of proposal budgets consistent with the detailed planning documents
- internal project oversight and review, ensuring that funds are expended according to the project plan, and identifying weaknesses in the execution of the project plan that need to be addressed
- preparation of quarterly progress reports
- development of a project web site, containing information on the project, schedules of meetings, repository of project documents, etc.

The Project Manager will report to the Lattice QCD Executive Committee. She/he will work closely with the software coordinator and the site coordinators both to assist in defining milestones and infstrastructure deployment schedules, to ensure a high level of coherency across the project. She/he will work with the Oversight Committee to ensure that it has sufficient information to carry out its duties. The Project Manater, with guidance from the Executive Committee, will work with the software coordinator and site coordinators to address problems identified by the Oversight Committee or by external reviews. Finally, she/he will serve as a point of contact with the DOE on matters related to budget and schedule of all funded activities.

5.2 QCDOC Project Management

The management of the QCDOC component of this proposal divides into two parts. The first concerns the design and construction of the QCDOC computer, including construction of the 1.5 Teraflops development machine. The second addresses the operation of the 1.5 Teraflops machine as a production machine for the use of the U.S. lattice QCD community.

5.2.1 QCDOC design and development

The design and construction of the QCDOC machines that is already underway or proposed here will be carried out by the team at Columbia. This activity is managed by Professors Christ and Mawhinney with Christ carrying the major responsibility for the hardware design, procurement and construction and Mawhinney the design and implementation of the QCDOC-specific software. This hardware and software design, construction and implementation effort is managed through weekly meetings of all Columbia/BNL/RBRC participants, weekly conference calls with the SciDAC Software Coordinating Committee, bi-weekly software meetings between Columbia and BNL personnel, and monthly software teleconferences between Columbia and Edinburgh/EPCC.

The Oversight committee periodically reviews the progress and status of this portion of the project in conference calls with Professors Christ and Mawhinney. Further oversight is provided by a monthly video conference with the UKQCD Project Management Committee and bi-weekly meetings with the Director of the RIKEN BNL Research Center.

5.2.2 QCDOC 1.5 Teraflops user facility

As is discussed earlier in this proposal the QCDOC 1.5 Teraflops machine will be supported as national resource with the allocation of time determined by the Scientific Program Committee. Hardware support for the machine will be provided by the team of Brookhaven technicians who presently maintain the QCDSP machine at Brookhaven and will participate in the construction of this and the RBRC 1.0 Tflops development machine. This group is managed by Edward Mcfadden, the RIKEN Supercomputer Project Manager. Software and user support will be provided by the present BNL SciDAC team together with the BNL software engineer described above. This software development/user support group as well as the overall management of this facility will be the responsibility of a facility manager to be identified by the beginning of 2003.

5.3 Cluster Management

Fermilab and JLab will jointly develop cluster hardware for the collaboration, and each laboratory will host some of the jointly developed hardware. Paul Mackenzie is the Fermilab principal investigator, and has overall responsibility for the work carried out at the laboratory. He serves as the link to the Lattice QCD Executive Committee. Don Holmgren is the lattice QCD project leader for the Distributed Systems Projects Group, and has direct management responsibility for the computing work done at Fermilab. Amitoj Singh of the Distributed Systems Projects Group will have lead operational responsibility for the Fermilab clusters. Routine user administration of the systems

in use for production computing will be handled by the Intgrated Systems Administration Group, which administers other large systems at Fermilab.

Chip Watson is the Jefferson Lab principal investigator with overall responsibility for all of the software and cluster work done there, and is also a member of the QCD Executive Committee. Jie Chen and Walt Akers, members of the High Performance Computing Group, serve both as software developers and as system administrators for the SciDAC prototype clusters. Much of the account management, and all of the site security, disk backups, and silo operations are handled by the Computer Center staff. In addition, Robert Edwards and David Richards of the Theory Group serve as liasons with the lattice user community, sponsoring accounts and providing some assistance in getting going on the clusters, and in using SciDAC lattice software.

6 Budget

In this section we briefly discuss the components of the budget we request under this grant. In the sections above, we described the full scope of our work. However, we again emphasize that the development of the QCDOC, the construction of the prototype QCDOC and clusters, and the software development work are all being funded from other sources.

The largest item in our budget is $1.5M for the construction of the 1.5 Tflops QCDOC. The timely construction of this machine is vital to our overall effort. Although the 128 node QCDOC prototype will be sufficient to test the hardware and the applications code, a computer of the proposed size is essential to determine the suitability of the architecture and operating system to serve our diverse national community. For this determination it is necessary to have a powerful enough machine to support a substantial mix of problems in a production environment. Furthermore, even though we are aiming for facilities with one to two orders of magnitude greater throughput, the 1.5 Tflops QCDOC will already be the most powerful computer dedicated to the study of lattice gauge theory. It will enable major advances in our research.

In order to deal with the increased scope of our work associated with the testing, operation and production use of the new hardware, we request a total of $635,800 for personnel, beyond what is being provided by our SciDAC grant. The breakdown of these funds is as follows:

- As discussed in the previous section, the increase in scope of this project associated with the construction and operation of terascale computing facilities requires the services of a full time project manager. The hardware and software development which is being carried out at three national laboratories and seven universities with the support of our SciDAC grant, the DOE Office of High Energy and Nuclear Physics, and other sources, will also benefit significantly from the attention of a professional manager. We intend to search broadly for the project manager, and can envision he/she being housed at any of the participating institutions. In order to be concrete, we have tentatively placed funds for this position in the UCSB budget, but we intend to shift them to whichever institution eventually houses the project direct. We request $128,700 in salary and fringe benefits, $10,000 in travel, and $5,000 for supplies in support of this position. The total including overhead comes to $211,240.

- Richard Brower, the software coordinator plays a crucial role in this project leading the soft-

ware effort. His responsibilities, which are already very time consuming, will increase as the QCDOC and prototype clusters come online. Through arrangement with Boston University, Professor Brower receives 50% release time from his academic responsibilities in return for the payment of 30% of his academic year salary by this project. This amount was paid by the SciDAC grant for the 2001-2002 academic year, and will be again for 2002-2003. However, there is a shortfall of $21,840 in salary support for the 2003-2004 academic year in the SciDAC budget. We request funds in this proposal to cover the salary shortfall, and an additional $5,000 for Professor Brower's travel expenses. The total cost including fringe benefits and overhead is $51,530.

- Steven Gottlieb, the Chair of the Oversight Committee also also plays a very important role in the project. He must stay abreast of all developments, keep his committee informed of them, and organize discussions within his committee as needed. His role will become even more time consuming as we begin to test hardware and software. Professor Gottlieb does not receive funding through the SciDAC grant. He was able to devote a great deal of time to the effort during the 2001-2002 academic year while on sabbatical leave at FNAL. To enable him to carry out his responsibilities, we propose to obtain 50% release time from his academic duties. Indiana University has agreed to provide this release time in return for the payment of 30% of his academic year salary from this grant. The total cost, including salary, $3,000 for travel, fringe benefits and overhead will be $56,445.

- As noted in Section 2 we request funding for four FTE to support the QCDOC. These include one FTE for hardware technical support, one FTE for a software engineer, and two postdoctoral research associates to work on operating system software for the QCDOC. The total budget for the four FTE comes to $500,000.

- Dr. Andrew Pochinsky plays a key role in the overall software effort, performance optimization, and porting of application code to the QCD-API. He currently receives ten months support per year from the SciDAC grant for this work. He has recently taken on new responsibilities in the evaluation of Gigabit Ethernet and development of the user space driver for the planned Jlab Gigabit Ethernet cluster described in Section 3.4. We request an additional two months of support for Dr. Poochinsky so that he continue this important work. With fringe benefits and overhead, the total cost is $27,700.

The table below summarizes the budget.

Item	**Cost**
1.5 Teraflops QCDOC	$1500.0
Project Manager	211.2
Software Coordinator	51.5
Chair, Oversight Committee	56.4
QCDOC Support Staff	500.0
Gigabit Ethernet Support Staff	27.7
Total	**$2346.8**

Table 8: Total budget request for FY 2003. All numbers are given in thousands of dollars.

A Physics Goals and Required Computational Resources

Our goal is to understand the physical phenomena encompassed by QCD, and to make precision calculations of its predictions. We will follow a multifaceted approach in which we calculate both quantities that can be compared with existing experimental results, so as to calibrate the accuracy of our methods, and use the same methods to predict quantities yet to be measured. In particular, our calculations will allow increasingly precise tests of the electroweak sector of the Standard Model, allow us to determine the properties of hadronic matter under extreme conditions, and understand nucleon structure and interactions. We describe each of these areas below. Furthermore, we expect the calculational techniques and computational infrastructure we develop for the study of QCD to be applicable to strongly coupled theories which go beyond the Standard Model, such as supersymmetric gauge theories, chiral gauge theories, and string theory. Indeed, work in these directions is already in progress by some members of our community.

A.1 Precision Testing of the Standard Model

A central focus of experiments at U.S. high energy physics facilities is precision testing of the Standard Model. The ultimate aim of this work is to find deviations from this model—a discovery which would require the introduction of new physical principles to describe matter at the shortest distances. Many of these tests require, in addition to precise experimental measurements, accurate evaluation of the effects of the strong interactions on processes induced by the electroweak interactions. Such an evaluation requires lattice QCD, the only known method which can systematically reduce all sources of error. These computations are one of the major physics focuses of our research plan, and a crucial companion to the U.S. experimental program in high energy physics.

The technical challenge is to calculate quantities known as "weak matrix elements" (matrix elements of electroweak operators between hadronic states). Each such matrix element, when combined with a particular experimental quantity, gives a direct measurement of an underlying parameter of the Standard Model. If multiple measurements of these parameters disagree, new physical principles are required. Table 9 summarizes the present situation for five key matrix elements. For four of the five, lattice calculations lag well behind experiment. [1] The impact of the larger lattice errors is shown in Figure 12. For the Standard Model to be correct, the parameters ρ and η are constrained to lie in the region of overlap of the solidly colored bands. The figure on the left shows the constraints as they exist today. The figure on the right shows the constraints as they would exist were the lattice gauge theory errors reduced to 3%, but with experimental errors unchanged. Reducing the errors to this size is the one of the goals of our calculations.[2]

Terascale computers will lead to an enormous advance in lattice calculations of such matrix elements, bringing the theoretical precision much closer to that of experiments. In Table 9 we give our estimates for the reduction in theoretical uncertainties resulting from calculations using two representative computational resources. The first assumes a a computer sustaining 0.5 teraflops

[1] The estimate of the present theoretical error in $\Delta M_{B_s}/\Delta M_{B_d}$ (20%) is larger than that quoted in our strategic plan [1]. It has been realized that the extrapolation in light quark masses is much more uncertain than previously thought [2, 3, 4].

[2] For a recent detailed review of the methodology of such calculations, their present status, and their implications for flavor physics see Ref. [5].

(Tflops) for a year, the scale of the calculations that will be possible with the proposed 1.5 Tflop QCDOC prototype. The second assumes a machine sustaining 10 Tflops for a year, which is the scale of the calculations we hope to undertake in subsequent years.

Such a reduction in errors approaches, and for some quantities attains, the level shown in the right-hand panel of Figure 12. Coupled with improvements in experimental results from the SLAC B-factory and the Tevatron B-meson program, our proposed calculations will lead to much more stringent tests of the standard model.

Measurement	CKM Matrix Element	Hadronic Matrix Element	Expt. Error	Current Lattice Error	Lattice Error 0.5 TF-Yr	Lattice Error 10 TF-Yr
ΔM_{B_d} ($\bar{B}B$ mixing)	$\lvert V_{td}\rvert^2$	$f_{B_d}^2 B_{B_d}$	4%	35%	18%	9%
$\Delta M_{B_s}/\Delta M_{B_d}$	$\lvert V_{ts}/V_{td}\rvert^2$	$f_{B_s}^2 B_{B_s}/f_{B_d}^2 B_{B_d}$	Not yet measured	20%	5%	3%
ε ($\bar{K}K$ mixing)	$\mathrm{Im} V_{td}^2$	B_K	2%	20%	10%	5%
$B \to \binom{\rho}{\pi} l\nu$	$\lvert V_{ub}\rvert^2$	$\left\langle \binom{\rho}{\pi} \right\rvert (V-A)_\mu \lvert B\rangle$	25%	Calc. in progress	15%	5–10%
$B \to \binom{D^*}{D} l\nu$	$\lvert V_{cb}\rvert^2$	$\lvert \mathcal{F}_{B\to\binom{D^*}{D} l\nu}\rvert^2$	2%	Calc. in progress	6%	3%

Table 9: *Impact of lattice QCD on the determination of CKM matrix elements. In the table above, f_X is the leptonic decay amplitude for the indicated meson, and B_X is proportional to the matrix element of $\Delta S = 2$ or $\Delta B = 2$ four–quark operators. The last two columns show the improvements in lattice errors that we estimate would be obtained with computers sustaining 0.5 and 10 Tflops for one year.*

There are many other matrix elements that can be calculated using lattice methods and used in a similar way to test the Standard Model. We collect these in Table 10. Terascale resources will lead to great improvement in the accuracy of the results for these quantities, and are an essential step towards reaching the desired accuracy.

There are also a large number of measured hadronic properties that will be calculable to high precision using a terascale facility, and which can be used to calibrate our methods. Such tests will be important demonstrations of the reliability of lattice calculations. Examples of properties that can be used for calibration include the masses and decay constants of B and D mesons and corresponding baryons, of charmonium and bottomonium states (bound states, respectively, of charm or bottom quarks and their antiparticles), and of hadrons composed of light quarks. An accuracy of a few percent is expected for some of these quantities with a terascale facility. We note that this calibration will be significantly sharpened by the CLEO-c program at Cornell [6].

An important feature of the lattice methodology is its flexibility. Particle physicists will no doubt discover other interesting matrix elements to calculate, and it is often the case that these can be

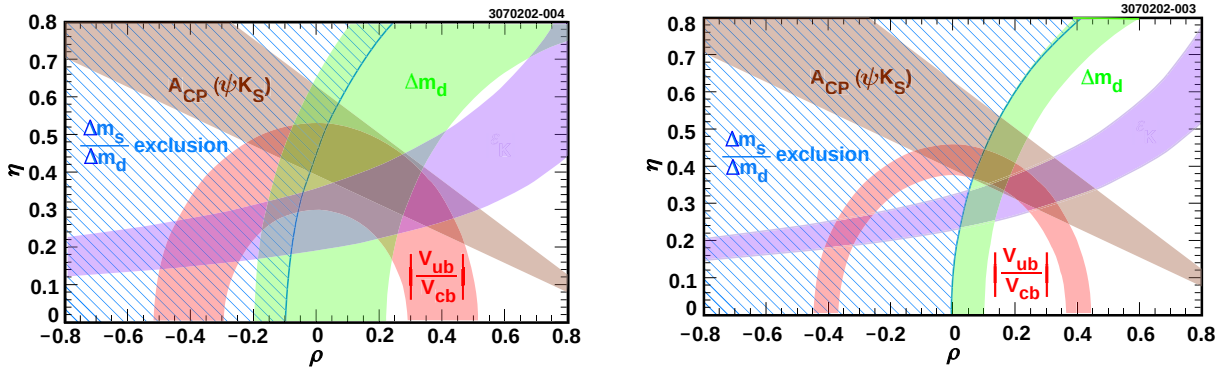

Figure 12: *Constraints on the Standard Model parameters* ρ *and* η *(one sigma confidence level). For the Standard Model to be correct, they must be restricted to the region of overlap of the solidly colored bands. The figure on the left shows the constraints as they exist today. The figure on the right shows the constraints as they would exist with no improvement in the experimental errors, but with lattice gauge theory uncertainties reduced to 3%. R. Patterson, Cornell University.*

Quantity	Importance
$f_D\sqrt{B_D}$	Needed to predict $D-\overline{D}$ mixing
$B \to (\pi,\rho)\ell\nu$ form factors	Improve determination of V_{ub}
$B \to (D,D^*)\ell\nu$ form factors	Improve determination of V_{cb}
$D \to \pi\ell\nu$ form factors	Alternative determination of V_{cd}
$B \to K^*\gamma$ and $B \to (K,K^*)\ell^+\ell^-$ form factors	Test standard model
Non SM $B-\overline{B}$, $D-\overline{D}$ and $K-\overline{K}$ matrix elements	Predict effects of new physics
CP-violation in $K \to \pi\pi$ decays	Test standard model
EDM of neutron induced by $\overline{q}\sigma_{\mu\nu}F^{\mu\nu}q$	Predict effects of new physics

Table 10: *Partial list of calculations of weak matrix elements that can be used to test the SM or predict the effects of physics beyond the SM.*

"piggybacked" on previous calculations with little overhead. Thus, the terascale facility we propose will provide a flexible database which can be reused repeatedly as new ideas appear.

Our calculations will also provide precise values for other fundamental parameters of the Standard Model—the quark masses and the strong coupling constant, α_S. Precise results for these are needed to differentiate between competing models of flavor physics and electroweak symmetry breaking, and lattice simulations provide the only method for doing such calculations. Indeed, the lattice results for the c and b quark masses are already very accurate (e.g. the error is 2% for m_b). The light quark masses—m_u, m_d and m_s—are more difficult to calculate, requiring extensive simulations with light dynamical quarks. Present errors, estimated to be 25% [7], will be substantially reduced by a terascale facility.

We now describe the basis for our estimates of the reduction in errors given in Table 9. The dominant sources of error in present calculations are (i) the use of dynamical ("sea") quarks which are much heavier than the physical up and down quarks; (ii) the need to extrapolate from non-zero

lattice spacing a to the continuum limit $a \to 0$; and (iii) the use of one-loop perturbative matching factors between lattice and continuum operators. To reduce the first error we need to use dynamical up and down quarks which are light enough that we reliably extrapolate to physical values. The precise minimum value needed depends on the rate of convergence of the chiral expansion and thus varies from quantity to quantity. It is generally expected that one requires the chiral expansion parameter $(m_\pi/m_\rho)^2$ to range down to ~ 0.1, and thus a minumum value $m_\pi/m_\rho \sim 0.3-0.4$. The size of the second error depends on the choice of gauge and fermion actions. In particular, the error can be reduced using the improved actions which have been developed over the last decade. For example, using unimproved staggered fermions to calculate B_K in the quenched approximation (no dynamical quarks), a reliable continuum extrapolation required lattice spacings as small as 0.05fm. Improved staggered fermions, for which the leading relative error is reduced from $(a\Lambda)^2$ (with $\Lambda \sim 1$ GeV) to $\alpha_S(a\Lambda)^2$ and $(a\Lambda)^4$, should have the same relative error for a minumum lattice spacing of 0.1fm. The use of such improved actions is an essential aspect of our calculations. The final error, due to matching factors, is more difficult to reduce. With staggered fermions, for which non-perturbative matching is difficult, the error from using 1-loop matching is $\sim \alpha_S^2 \approx 5\%$. When the other errors reach this level, it will be essential to push the matching calculation to two loop order. Methods for automating such calculations are under development. For Wilson-like and domain-wall fermions, non-perturbative methods are available which can greatly reduce this error.

Our first large scale calculations will likely use improved staggered fermions, since these are computationally less demanding than other discretizations. We focus on a particular variant, the "Asqtad" action recently developed by members of our group [8]. The strange quark will be held at its physical value, and the light quark masses lowered as far as possible. Resource requirements for this algorithm scale approximately as a^{-7} for fixed physical volume and m_π/m_ρ,[3] as $m_{u,d}^{-2.5}$ at fixed lattice spacing,[4] and proportional to the volume at fixed a and m. Based on existing calculations at $a \approx 0.1$ fm on lattices $3-3.5$ fm across, with $m_\pi/m_\rho \approx 0.33$ we can scale to lattice parameters needed for our weak matrix element calculations. In table 11 we give the resource requirements for three representative parameter choices. These are based on needing 100 independent lattices to obtain small enough statistical errors, and on 25 molecular dynamics trajectories between independent configurations. An overhead factor of 3 for masses other than the lightest one is included. We choose 3 fm for the lattice volume, at which value finite volume corrections in mesonic quantities should be much smaller than other errors, with the possible exception of the semileptonic form factors. We include a lattice spacing smaller than 0.1 fm because we expect to use only partially improved operators along with the improved action.

We see from the table that a facility which can deliver Tflops-years, such as our proposed prototype QCDOC machine, can reach the desired range of parameters. We will then be able, for the first time, to make quantitative estimates of the residual errors from the data itself. This is in contrast to present estimates of quenching errors (such as those in table 9) which are based on qualitative arguments. This is what makes Terascale computations in QCD so exciting.

Continuum and chiral extrapolations with staggered fermions are quite challenging because of the breaking of the so-called "taste" symmetry. It will thus be particularly important to repeat all calculations using other fermion discretizations. Generically, these are more computationally

[3] Four powers of a^{-1} arise from the change in number of lattice points, and one power each from the inversion of the Dirac operator, the decrease in step size, and the increase in autocorrelation time.

[4] One power of $m_{u,d}^{-1}$ from the the inversion of the Dirac operator, the second with the step size, and the square root with the correlation length, which is inversely proportional to the pion mass.

Spacing	Size	Minimum m_π/m_ρ	Tflops-years
0.1 fm	$30^3 \times 60$	0.4	0.07
0.1 fm	$30^3 \times 60$	0.3	0.3
0.067 fm	$48^3 \times 96$	0.4	1.5

Table 11: *Examples of CPU requirements for different lattice parameters, for 100 independent lattices.*

demanding, and will require an order of magnitude more computing power. Examples of such calculations are discussed in the following sections.

We close this section by stressing two points. First, we have been conservative in our estimates of computational requirements, since advances in algorithms may speed up the calculations in the future. Also, we use asymptotic scaling laws, which overestimates the time needed in the range of parameters we are considering. Second, we note again that the lattices which are generated can be used for calculating all the matrix elements of interested—they can be repeatedly reused.

A.2 The Quark Gluon Plasma

At low temperatures and densities, quarks and gluons are confined in elementary particles, such as neutrons and protons. At very high temperatures and densities one expects a phase transition or crossover from this ordinary strongly interacting matter to a plasma of quarks and gluons. Such a plasma is believed to have been a dominant state of matter in the early development of the universe, and a possible central component of neutron stars today. A primary physics goal of the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory is the discovery and characterization of the quark-gluon plasma. In order to confirm such an observation, it is important to determine the nature of the transition, the properties of the plasma, including strange quark content, and the equation of state. Lattice gauge theory has proven to be the only source of *a priori* predictions about this form of matter in the vicinity of the phase transition. We plan to use improved actions to undertake a detailed study of these issues.

Lattice gauge theory has already provided a wealth of qualitative information for zero baryon density (zero chemical potential), including an indication of the order and an estimate of the temperature of the phase transition as well as a characterization of the equation of state. However, simulations have, for the most part, only included the up and down quarks, whereas it is expected that the somewhat heavier strange quark also plays a crucial role. From universality considerations we expect that with two flavors of quarks there is no phase transition at zero baryon density for physical values of the up and down quark masses – merely a crossover. However, a strange quark could induce a first order transition, or move a second order critical point closer to physical quark masses. These effects may be of considerable importance to the phenomenology of the phase transition. Furthermore, up to now it has not been possible to carry out realistic QCD simulations at nonzero chemical potential. However, recent algorithmic developments [9] have opened the possibility of performing such simulations in the regime relevant to RHIC experiments. The further development and application of algorithms for the simulation of QCD at finite baryon density will be an important component of our work.

We propose to take advantage of recent progress in algorithms and improved actions, and the enormous power of the proposed facilities to carry out a definitive study of the quark-gluon plasma with a realistic quark ensemble, and vastly reduced discretization artifacts. The results are expected to give valuable assistance to the RHIC experimental program.

Our scientific objectives are the following:

1. Mapping of the phase diagram in temperature, chemical potential and quark mass for up, down, and strange quarks, including determining the order of the phase transition and temperature of the crossover.
2. Determining the equation of state of the plasma, including strange quark content.
3. Determining the quark number susceptibilities for up, down and strange quarks.
4. Predicting real-time excitations of the plasma.
5. Understanding the role of instantons in the phase transition.
6. Measuring the strength of the axial $U(1)$ anomaly.

The first four objectives have obvious relevance to the analysis of experimental results. The last two are needed for formulating phenomenological models that extrapolate to situations inaccessible to lattice gauge theory. Based on our extensive experience in the study of QCD thermodynamics with simpler actions, this will be a multi-year project, which will require long-term use of the proposed 10 Tflop/s Facility.

As with all lattice simulations, systematic errors primarily arise from three sources: (1) the lattice discretization, (2) finite volume, and (3) unphysically large quark masses. For example, previous thermodynamic studies of the effects of strange staggered quarks [10] and strange Wilson quarks [11] have been done with most of the pion masses far above their real-world values. While these studies are the best that could be done with the algorithms and computing power of the time, it is hard to do a realistic study of the effects of a (strange) K meson with its correct physical mass of 500 MeV, when most of the (non-strange) pions have comparable masses, well above their physical mass of 140 MeV.

To achieve our scientific objectives in a realistic simulation requires a combination of significantly improved algorithms and significantly enhanced computing power. Recently, we have developed substantial algorithmic improvements in the two major formulations for quarks (fermions) on the lattice: (1) Without requiring an excessively small lattice spacing, our improved staggered fermion actions [8] give vast improvement in the zero-temperature hadron spectrum, a feature essential on the low temperature side of the phase transition, and they give significant improvement in the quark and gluon dispersion relations, essential at high temperature [12]. Also, especially important, on the low temperature side, dispersion relations and flavor symmetry in the critically important pion sector are vastly improved. (2) The recently implemented domain wall fermion approach dramatically improves the Wilson fermion scheme. This new method preserves the flavor symmetry of the Wilson formulation, improves its relatively good scaling properties by removing all $O(a)$ errors both on- and off-shell, realizes the complete chiral symmetry group and supports the physics of the axial anomaly and the Atiyah-Singer theorem [13]. This physical chiral symmetry (both anomalous and non-anomalous) of the domain wall method may be particularly important for an accurate study of the QCD phase transition.

As for objective 2, the strange quark content of the plasma is of vital interest, since excessive strange quark production could be a signal of plasma formation. The strange quark content is determined by measurement of the contribution of strange quarks to the energy density of the plasma, so emerges from a study of the equation of state.

The determination of the equation of state is very costly, since it involves computing the difference between the energy density at a nonzero and zero temperature. The loss of significance in the subtraction makes the computational effort scale in the lattice spacing as a^{-4} relative to the effort for mapping out the phase diagram. Thus algorithmic improvements that suppress lattice artifacts and permit simulation on a coarser lattice are of critical importance. Recent studies on smaller lattices have shown dramatic changes resulting from algorithmic improvement in the Wilson fermion scheme [14, 15]. We expect similar improvements in the domain wall approach.

The quark number susceptibilities, which are related to event-by-event fluctuations in heavy ion collions through the fluctuation–dissipation theorem, provide a clear signal for the onset of the transition. That of the strange quark may prove to be a particularly useful tool for the experimental identification of the plasma.

Real time excitations of the plasma (objective 4) are very poorly understood, but have a direct relevance, for example, for the measurement of the dilepton emission spectrum. Maximum entropy methods, now being explored by a number of groups, offer the promise of extracting real-time spectral information from finite-temperature lattice Green's functions [16].

Based on our extensive experience in the study of QCD thermodynamics at zero baryon density using simpler actions, this program will be a multi-year project, which will require long-term use of the proposed multi-Tflops facilities.

A.3 Structure and Interactions of Hadrons

The third major scientific goal of our collaboration is to achieve a quantitative, predictive understanding of the structure and interactions of hadrons.

The internal structure of the nucleon is a defining problem for hadron physics just as the hydrogen atom is for atomic physics. Indeed, the DOE Strategic Plan specifically highlights the goal of developing a quantitative understanding of how quarks and gluons provide the binding and spin of the nucleon based on QCD. Major experimental efforts at Bates, Jefferson Lab, SLAC, Fermilab, the HERMES experiment at DESY, and the EMC, SMC, and NMC experiments at CERN provide rich and precise measurements of the quark and gluon structure of the nucleon, and proposed experiments such as the RHIC spin program promise to reveal even greater detail. With recent advances in lattice field theory, it is now possible to calculate this nucleon structure directly from QCD, so that multi-Terascale lattice calculations are now an essential tool to obtain the full physics potential of major accelerators and detectors.

A wealth of experimental observables can be calculated on the lattice. Electromagnetic form factors measured in elastic electron scattering characterize the distribution of charge and magnetization arising from all the quarks in the nucleon. Parity violating electron scattering experiments exploit the fact that the neutral weak current couples to a different linear combination of up, down, and strange quarks than the electromagnetic current to measure the strange quark contributions to electric and magnetic form factors. Recent experiments at Bates and Jlab [17] have now measured

these form factors, and future experiments are expected to determine them with high precision [18].

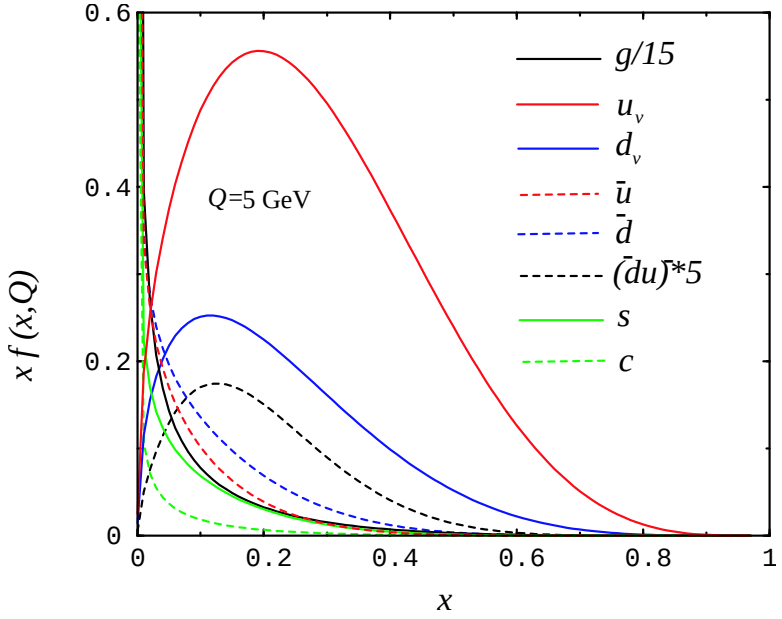

Figure 13: High energy scattering experiments have measured the distribution of quarks and gluons in the proton. The measured distributions $f(x,Q)$ as a function of momentum fraction x for gluons, g, up and down valence quarks, u_v and d_v, up and down sea quarks, $\bar{u}$ and$\bar{d}$, and strange and charmed quarks, s and c are shown in the figure for measurements at momentum transfer Q = 5 GeV. Multi-teraflops facilities will enable calculation of the moments of these distributions from first principles, providing a fundamental understanding of the structure of the proton.

Deep inelastic scattering of electrons, muons, and neutrinos measures structure functions characterizing the light cone quark density, quark spin density, and gluon density as a function of momentum fraction, and the moments of these distributions can be calculated on the lattice. Several decades of intense experimental study of the nucleon using high energy electromagnetic and hadronic probes is succinctly summarized by the experimental quark and gluon distributions shown in Figure 13 as a function of the momentum fraction x. A particularly important example is the lowest moment of the spin density, which measures the fraction of the nucleon spin carried by the spin of quarks. Indeed, the only way to fully resolve the so-called "spin crisis" which arose when experiments showed that only about 20% of the spin of the nucleon originates from quark spins is to calculate in lattice QCD how the total spin is divided between quark and gluon spin and orbital angular momentum. Moments of generalized parton distributions [20], which for example enable experimental separation of the fraction of spin carried by quarks and by gluons, can also be calculated on the lattice and are a focus of experimental interest in at the 12 GeV upgrade at JLab and at the proposed electron-ion collider.

Prototype calculations with limited computational resources have already established the methodology to calculate the nucleon form factor [21], the axial charge [22], the contributions of strange quarks in the nucleon [23], and moments of the quark density, spin, and transversity distributions in both quenched [24] and full [25] QCD. These calculations also clearly show the need for Multi-

teraflops computational resources by highlighting the major role the long-distance pion cloud plays in hadron structure.

Physically, one knows that the long-distance pion cloud contributes substantially to the nucleon magnetic moment, axial charge, and moments of structure functions. However, accurate calculation of the the pion cloud is computationally very expensive for a combination of reasons. One must calculate at very light quark mass in order to build up the quark-antiquark excitations producing light pions, thereby encountering the algorithmic slow-down associated with light masses. One must use a very large box corresponding the the large Compton wavelength $\frac{1}{m_\pi}$ associated with a light pion, thereby bearing the cost of large lattice volumes. Finally, one must use dynamical quarks to include all the physical processes contributing to the pion cloud, precluding the economy of the quenched approximation. Recently, chiral extrapolations [26] were performed for unquenched calculations of moments of structure functions [25] showing that lattice measurements of the order of 5% accuracy are needed down to a ratio $\frac{m_\pi}{m_\rho} = 0.3$ or $m_\pi = 230 MeV$ in a box of linear dimension $4.3\,fm$ for reliable extrapolation. The number of floating point operations per independent full QCD configuration, N in Teraflops-years, for hybrid Monte Carlo calculations with conventional Wilson quarks is [27]:

$$N \simeq .038 \left[\frac{L}{4}\right]^{4.55} \left[\frac{.08}{a}\right]^{7.25} \left[\frac{.3}{\frac{m_\pi}{m_\rho}}\right]^{2.7}$$

where L is the spatial lattice dimension in fm, the time dimension is twice the spatial dimension, and a is the lattice spacing in fm. Using of the order of 400 independent configurations for adequate statistical accuracy, a calculation with a lattice spacing $a = 0.1\,fm$ and $m_\pi/m_\rho = 0.3$ requires 8 Teraflops-years, *ie*, dedicated use of a computer that sustains 8 Teraflops on QCD for one year. Thus, the physics of the pion cloud necessitates computational facilities sustaining of the order of 10 Tflops for a quantitative understanding of hadron structure.

Spectroscopy is the classic tool for discovering the relevant degrees of freedom of a physical system and the forces between them. One of the fascinating features of QCD is that it offers the possibility of a richer range of hadronic states than has yet been observed experimentally, so that precise lattice calculations can play a pivotal role in helping guide experimental searches. Lattice calculations will study the number and structure of hadronic excited states, as well their transition form factors. The presence or absence of hadrons with exotic quantum numbers, the nature of glueballs, and the overlap between model trial functions and exact hadron states will provide insight into the role of flux tubes, dibaryons, and the inner workings of QCD. Again, exploratory calculations of the lowest negative parity N^* state [28], a comprehensive calculation of the glueball spectrum [29], calculation of the masses of Hybrid mesons [30], and a study of the existence of the H particle [31] show that the requisite methodology is ready. Precision multi-Terascale calculations will provide crucial insight into current and future hadron spectroscopy.

Currently, there is no fundamental understanding of the very foundation of nuclear physics, the nucleon-nucleon interaction. Significant insight into the role of gluon exchange, quark exchange, meson exchange, and the origin of short range repulsion will be obtained by lattice calculations of the adiabatic potential between heavy-light systems, [32, 33] that is, mesons or baryons containing a single heavy quark in addition to other light quarks or antiquarks.

In addition to calculating observables to compare with experiment, lattice calculations are invalu-

able to obtain insight into fundamental aspects of QCD. Current lattice techniques can study the role of instantons [34] and their associated zero modes [35] in chiral symmetry breaking, the role of center vortices [36] and magnetic monopoles [37] in confinement, and calculation of the parameters entering chiral perturbation theory. The lattice also allows theorists to answer interesting theoretical questions inaccessible to experiment, such as how the properties of QCD change with the number of colors, quark flavors, or quark masses.

Major research projects that would be carried out in the areas of hadron structure and interactions in the early years of this project include the following:

- Calculation of the strange quark contribution to the nucleon's electromagnetic form factors. These results will elucidate the strange quark content of the nucleon and complement fundamental parity-violating electron scattering experiments.
- Precision calculation of nucleon form factors with sufficiently light quark masses to include the physics of the pion cloud. This analysis based on QCD is needed, for example, to understand the recent precise measurements of the ratio of the proton's electric and magnetic form factors and to understand measurements of the neutron's electric form factor.
- Calculation of moments of nucleon parton distributions and nonleading twist operators in order to understand the quark-gluon structure of the nucleon. The results will provide fundamental understanding for HERMES and RHIC-spin high-energy experiments, and for experiments at JLab using 6–12 GeV electron beams.
- Calculation of leading light-cone quark-distribution amplitudes of the nucleon. These amplitudes determine the normalization of perturbative contributions to form factors and large momentum-transfer Compton scattering.
- Calculation of the spectrum of lowest lying N^* resonances for several spins in order to determine the lattice QCD spectrum in a mass region where states are predicted to exist in the quark model, but none have been observed experimentally.
- Calculation of N^* and Δ transition form factors. These calculations would be provide essential information about how any deformation of the nucleon is manifested in transition form factors and complement the new experimental N^* program.
- Calculation of the Born-Oppenheimer potential between two baryons, each of which contains one heavy quark and *two* light quarks, would address the central problem of establishing the link between QCD and the nucleon-nucleon interaction. The use of one heavy quark in each baryon allows for a clean definition of the relative coordinate.
- Exlporatory studies of chiral phase transitions of staggered fermions at finite temperature and chemical potential using the meron cluster algorithm in order to develop a practical algorithm to study quantum field theory at finite baryon density. Exploration of the quantum link D-theory formulation of QCD with cluster algorithms.

  A more detailed description of proposed research in hadron structure and interactions by members of this collaboration may be found on the web [5]

[5] `ftp://www-ctp.mit.edu/pub/negele/LatProp/`

References

[1] R. Brower *et al.* (Lattice QCD Executive Committee), "Computational Infrastructre for Lattice Gauge Theory", April, 4 2002, http://www.lqcd.org/strategic-plan-04-04.pdf.

[2] S. Hashimoto *et al.* [JLQCD Collaboration], arXiv:hep-lat/0209091; C. Bernard, *et al.*, arXiv:hep-lat/0209086.

[3] N. Yamada, arXiv:hep-lat/0210035.

[4] A. S. Kronfeld and S. M. Ryan, arXiv:hep-ph/0206058.

[5] A. S. Kronfeld,

[6] Details of the CLEO-c proposal and program are available at `http://www.lns.cornell.edu/public/CLEO/spoke/CLEOc/index.html`.

[7] V. Lubicz, Nucl. Phys. Proc. Suppl. **94**, 116 (2001) [hep-lat/0012003].

[8] G.P. Lepage, Nucl. Phys. (Proc. Suppl.) **60A**, 267 (1998); Phys. Rev. D **59**, 074501 (1999); K. Orginos, R.L. Sugar and D. Toussaint, Phys. Rev. D **60**, 054503 (1999); Nucl. Phys. (Proc. Suppl.) **83-84**, 878, 2000.

[9] Z. Fodor and S.D. Katz, Phys. Lett. **B534**, 87 (2002); C.R. Allton, S. Ejiri, S.J. Hands, O. Kaczmarek, F. Karsch, E. Laermann, Ch. Schmidt, and L. Scorzato, hep-lat/0204010; Ph. de Forcrand and O. Philipsen, hep-lat/0205016.

[10] F.R. Brown *et al.*, Phys. Rev. Lett. **65**, 2491 (1990).

[11] Y. Iwasaki *et al.*, Phys. Rev. **D54**, 7010 (1996).

[12] The MILC collaboration: C. Bernard *et al.*, C. Bernard *et al.* (The MILC collaboration), Phys. Rev. D **61**, (2000) 111502.

[13] P. Vranas, Lattice 2000 Review, hep-lat/0011066

[14] The MILC Collaboration: C. Bernard, *et al.*, Phys. Rev. **D56**, 5584 (1997); R.G. Edwards and U.M. Heller, Phys. Lett. **B462**, 132 (1999).

[15] Y. Iwasaki *et al.*, Phys. Rev. Lett. **78**, 179 (1997); Nucl. Phys. B (Proc. Suppl.), **63**, 397 (1998).

[16] M. Jarrell and J.E. Gubernatis, Phys. Repts. **269**, 135 (1996); M. Asakawa, T. Hatsuda, and Y. Nakahara, hep-lat/0011040.

[17] B. A. Mueller *et. al.* Phys. Rev. Lett. **78** (1997) 3824.

[18] TJNAF experiments E91-010, E91-017, and E91-004.

[19] H. L. Lai *et al.*, hep-ph/9706502.

[20] X. Ji, Phys. Rev. D55 7114 (1997): J. Phys. 24, 1181 (1998).

[21] S. Capitani *et al.*, Nucl. Phys. Proc. Suppl. **73**, 294 (1999)

[22] S. Güsken *et al.* Phys. Rev. D **59**, 114502 (1999).

[23] S. J. Dong, K. F. Liu, and A. G. Williams, Phys. Rev. **D58** 074504 (1998).

[24] M. Göckeler *et al.*, Phys Rev. D53, 2317 (1996).

[25] D. Dolgov *et al.* [LHPC collaboration], lattice QCD," arXiv:hep-lat/0201021.
Hackett-Jones

[26] W. Detmold, W. Melnitchouk, J. W. Negele, D. B. Renner and A. W. Thomas, Phys. Rev. Lett. **87**, 172001 (2001) [arXiv:hep-lat/0103006].

[27] T. Lippert, S. Gusken, and K.Schilling, Nucl. Phys. B (Proc. Suppl.) **83**, 182 (2000).

[28] D. G. Richards, hep-lat/0011025, to appear in Nucl. Phys. B, (Proc. Suppl.) (2000).

[29] C.J. Morningstar and M. Peardon, Phys. Rev. D 60, 34509 (1999).

[30] K.J. Juge, J. Kuti, and C.J. Morningstar, Phys. Rev. Lett. 82, 4400 (1999).

[31] J. W. Negele, A. Pochinsky, and B. Scarlet, Nucl. Phys. B (Proc. Suppl.) 73, 225 (1999).

[32] D. G. Richards, D. K. Sinclair, and D. Sivers, Phys. Rev. **D42** 3191 (1990).

[33] C. Michael, and P. Pennanen, Phys. Rev. **D60** 054012 (1999).

[34] J.W. Negele, in Unfolding the Matter of Nuclei, Course CXXXVIII, Varenna, Italy, July (1998).

[35] T. L. Ivanenko and J. W. Negele, Nucl. Phys. B (Proc. Suppl.) **63** 504 (1998).

[36] L. Del Debbio, M. Faber, J. Greensite, S. Olejnik, Phys. Rev. D55, 2298 (1997).

[37] A.S. Kronfeld, M.L. Laursen, G. Schierholz, U.J. Wiese, Phys. Lett. B198, 516 (1987).

B QCDOC Architecture

The QCDOC architecture is a natural evolution of that used in the QCDSP machines. Individual processing nodes are PowerPC-based and interconnected in a 6-dimension mesh with the topology of a torus. A second, Ethernet-based network provides booting and diagnostic capability as well as more general I/O. The entire computer will be packaged in a style that provides good temperature control, a small footprint and easy accessibility. Central to this design is the IBM Blue Logic technology which makes possible the high-density, low-power combination of an industry-standard RISC processor with 64-bit floating point, embedded DRAM, 500 MHz communications and the wide array of pre-designed functions needed to assemble the complete, functioning unit.

Node architecture. Each node is made of a single applications specific integrated circuit (ASIC) chip and an industry standard DIMM memory. The ASIC contains a 440 PowerPC processor core with a 64-bit floating point unit. Part of IBM's embedded PowerPC offering, this is a highly functional 32-bit processor with a 32 Kbyte, prefetching instruction cache, and a 32 Kbyte data cache with flexible cache control. The processor includes memory management with a 64-entry translation-lookaside-buffer which supports variable page sizes from 1 Kbyte to 256 Mbyte. This RISC PowerPC core is Book E compliant with dynamic branch prediction and a dual issue pipeline.

In addition to the 440 processor core, this single ASIC chip contains 4 MBytes of on-chip memory, referred to as "embedded DRAM" or EDRAM. This is sufficient to hold the code and data for a standard lattice QCD calculation and provides a large bandwidth to the processor, up to 8 GBytes/sec. (In fact, for most QCD kernels, the entire code will easily fit in the 32 Kbyte instruction cache.) In addition, this ASIC contains the DMA capability needed to move data automatically between EDRAM and external memory, the circuity to support internode communication and an Ethernet controller for the boot-diagnostic-I/O network described below.

The single DIMM memory card, which will be part of each node, will be 64-bit-wide, 166 MHz DDR SDRAM with an additional 8 bits of ECC. The memory size is determined by the particular memory modules acquired, ranging from 32MBytes to 2 GBytes per node. The budget in this proposal targets a 128 Mbyte DIMM card per node for a total machine memory of 0.66 TBytes.

Inter-node Communications. Each processor will have the capability to send data to and receive data from each of its twelve nearest neighbors in six dimensions at a rate of 500 Mbits/sec. This will provide a total off-node bandwidth of 1.5 GByte/sec. Each communication link will have a phase locked receiver and single-bit error detection with automatic resend. Each of these twenty-four communication channels will have its own direct memory access capability allowing autonomous reads/writes from either EDRAM or external SDRAM. Instructions controlling each of these DMA transfers will be stored as 16 sequences of block-strided-moves located in 24 separate, on-chip register blocks. (Note, since two of these six dimensions will be used to partition the machine, only two-thirds of this communications bandwidth or 1 GBytes/sec will be available for a typical QCD calculation.)

Low-latency, global functionality in the form of an automatic "store and forward" capability will be provided to enhance the speed of global sums and broadcasts. While this is a subset of the global operations built into the QCDSP machines, it is well matched to the intrinsic latency of the QCDOC communications network and PowerPC execution speed. This store-and-forward operation will introduce a latency of 120 ns per node in the communications path. A double-precision global sum on a 4K node partition is expected to take $\approx 9\mu$sec.

Given the simple nearest-neighbor connections that make up our six-dimensional torus geometry, a faulty processor will block communications between the remaining nodes on some paths of importance. While we might attempt to develop a partitioning scheme for the machine which keeps a five-dimensional slice available for substitution in the case of such a failure, this is most likely not worth the trouble. Just as with the earlier QCDSP machines, the simple modular nature of the individual nodes allows a faulty processor to be easily replaced. Thus, a node failure will cause the partition containing that node to fail. One will replace the faulty node and restart the calculation from the most recent checkpoint. We expect the booting of the entire machine to require on the order of fifteen minutes. With our present 20,000 nodes of operational QCDSP hardware, such replacements are required infrequently, perhaps on the order of once per week and booting our largest 4608-node, QCDSP machine takes roughly 10-15 minutes.

Booting, diagnostics and I/O. The SCSI booting, diagnostic and I/O network of the QCDSP machines will be replaced by 100 Mbit/s Ethernet. The Ethernet connections of four processors will be joined together with a Fast Ethernet switch whose output will be fed to a higher level switch which includes a Gbit Ethernet link. This Ethernet tree will be used in broadcast mode to provide boot code to the processors, will allow individual processors to be interrogated for diagnostic purposes and permit easy connection to industry standard RAID disks, providing a large aggregate I/O bandwidth. A fully-functional, IBM RISCWatch debugger will be provided, allowing a multi-node, window-per-processor, source-code-based graphical debugging interface.

Mechanical design. As in the QCDSP machines, we plan to exploit the homogeneity of this style of massively parallel machine to achieve a high degree of mechanical modularity. The individual processors will be mounted two per daughter card, one being impractical given the 5 inch width of the DDR SDRAM cards. We will mount 32 such daughter cards on a mother board and then 8 mother boards in a rather large crate with a single backplane. These crates will be cooled by vertical air flow passing through a water-cooled radiator below each crate. Cable connections will be provided on the backplane for the off-node communications of each motherboard.

From the perspective of communications, each motherboard will appear as a 3-dimensional cube with each of its six faces containing 32 nodes. Thus, each such face will require 64 differential pairs of communication links which will be passed directly through motherboard VHDM-HSD(Molex) or ZPack-HS-Zd(AMP) edge connectors, directly through the backplane to mating cable connectors on the reverse side. With six cable sets per motherboard, any desired topology for the complete machine can be accommodated when the machine is cabled up.

B.1 Overview of ASIC components

An effective overview of the QCDOC ASIC is provided by Figure 14. To a large extent, this ASIC is created from already existent IBM macros that are simply interconnected in the specified way to create the larger unit. Special to our design, in addition to our particular choice of components, is the serial communications unit (SCU), the prefetching EDRAM controller (PEC) and the DMA controller permitting direct transfers between external and embedded DRAM. The various components of this diagram are briefly described below.

440 Core. A standard PowerPC Core with attached floating point unit described below. In addition to the usual PLB interface, we have designed a special on-chip-memory controller, implemented following a OCM2 strategy, to efficiently use the EDRAM.

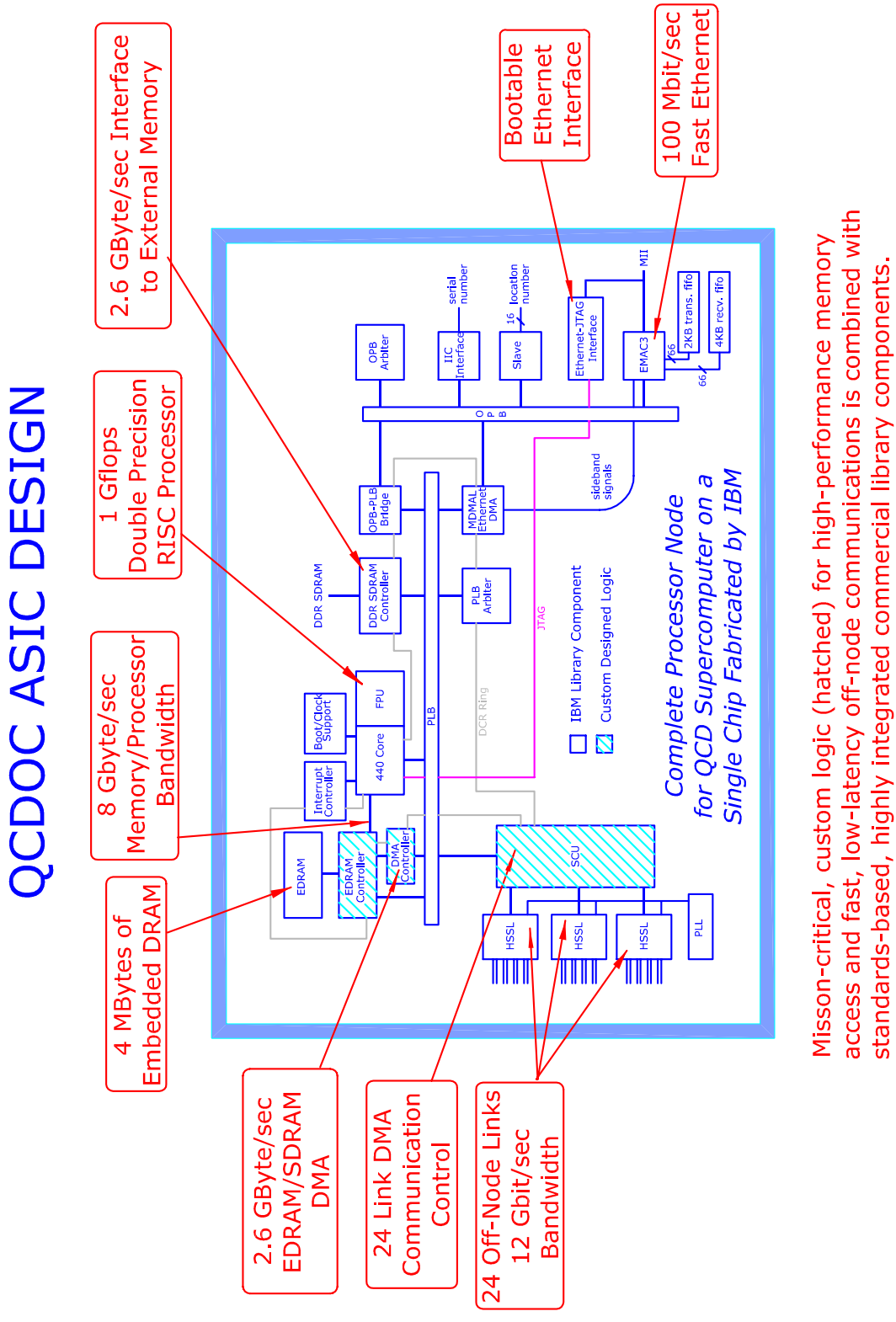

Figure 14: Block diagram providing an outline of the QCDOC ASIC design

Boot/Clock support. This can be assembled out of IBM-provided components. It creates the 500 MHz clock and sequences the power-up of the chip as the voltages appear.

DMA controller. Logic that we will provide, allowing automatic transfers over the processor local bus (PLB) between external SDRAM and EDRAM. This should use the full 128-bit width of the PLB and exploit the caching capability the DDR SDRAM controller provides.

DMA Ethernet controller. A MCMAL unit specifically configured to load and unload the Ethernet controller. This is an IBM-supplied macro.

EDRAM. This embedded 4 Mbyte DRAM provides code and data storage on-chip. In the most demanding problems, we expect to hold the entire local data and code in this on-chip location.

Ethernet controller. This is a standard EMAC3 unit that is available as an IBM macro. It has a standard interface to the OPB and the Ethernet DMA controller. We do not at present plan to implement a PHY layer on chip. Instead, this controller will drive a standard MII interface to an off-chip PHY controller.

Bootable Ethernet Interface. This second Ethernet connection permits hard-wired Ethernet control of the JTAG interface to the 440. This provides complete control of the processor to the host computer through the Ethernet interface, allowing processor reset and bootcode loading directly to the 440 instruction cache. In addition, it supports the IBM RISCWatch debugging tool. This Ethernet/JTAG interface has been developed at IBM and an FPGA hardware version is currently under test at Columbia.

FPU. A 64-bit, IEEE floating point unit that is being designed by IBM as a co-processor to the 440 core. The unit contains 32, 64-bit floating registers and, as is described below, performs $SU(3)$ matrix times 2-spinor arithmetic at 84% efficiency. Its physical and software integration with the 440 core is automatic and seamless.

HSSL. The physical units managing the 500 MHz serial communication in our 6-dimensional network. After an appropriate training period, the input 500 MHz serial data is byte aligned and provided to the serial communications unit at 62.5 MHz. Similarly, input 8-bit data is serialized and clocked out at eight times the frequency. This is a high-performance, IBM-provided macro that we will use without modification. Each HSSL controls four independent serial inputs and outputs.

Interrupt controller. A group of three copies of a standard IBM macro which will process and combine the interrupts generated by the components of this ASIC and additional external interrupts generated elsewhere in the machine.

Location number slave. This unit will be connected to wires strapped in such a way as to uniquely determine the location of the ASIC within the larger machine. This will include location on the motherboard, location of the motherboard within the crate and of the crate within the larger machine. This location information will be used to construct a MAC address that can be used at boot-up to locate the node.

OPB. The standard IBM on-chip peripheral bus. It is included here in order to remove inessential loads from the more time-critical PLB and to provide the standard interface required by the Ethernet controller.

OPB-PLB bridge. A standard IBM macro connecting the OPB and PLB. The only version of this bridge that we will need appears as a slave on the PLB and a master on the OPB.

OPB arbiter. A standard IBM macro necessary to manage the control and arbitration signals on the OPB.

PLB arbiter. A standard IBM macro necessary to manage the control and arbitration signals on the PLB.

PLB. This is the main on-chip bus. It is 128-bits wide, will run at 166 MHz and contains two somewhat independent sub-busses. Note there is a limitation of 8 masters total on the PLB and the 440 necessarily uses 3, leaving 5. We use only four: the SCU read, SCU write, the EDRAM DMA and the MCMAL DMA required for the Ethernet controller.

PLL. An IBM-supplied hard macro required by the HSSL units. Its location and wiring pattern must carefully follow IBM specifications so as to meet the requirements of the HSSLs.

PEC. This memory controller provides an independent, buffered interface for the EDRAM to the PLB, the DMA controller and the 440 core. The later connection is provided through a dedicated 500 MHZ, 128-bit PLB bus. Prefetching and buffering are included in the PEC, allowing two independent streams of sequential data to be efficiently read from EDRAM from each of these three ports. In particular, this unit provides an 8 GByte/sec bandwidth between EDRAM and the 440's data cache. The PEC also provides standard double-bit error checking and single-bit correction for the EDRAM.

SCU. Next to the EDRAM controller, this serial communications unit requires the most design effort. It contains 12 semi-autonomous units that will assemble the incoming bytes from the HSSL units into 64-bit words which will then be fed, under DMA control, to their destination in memory. A further 12 units will parse 64-bit words fetched under DMA control from memory and provide them as a sequence of bytes to the sending components of the HSSL. Sending and receiving units will each have 3-word deep buffers and will synchronize by exchanging ACK packets. A ninth byte will be sent with each transmitted word which will identify the type of packet to follow as well as two bits of parity. Bad data will be retransmitted. Finally, the SCU will provide the store-and-forward function that supports low-latency global sums and broadcasts.

B.2 Prefetching EDRAM Controller (PEC)

We now discuss this important custom module in greater detail. This unit, within the processor ASIC, will allow high-bandwidth, low-latency communication between the data cache of the 440 processor core and the 4 MBytes of embedded DRAM. This component will allow us to fully exploit the very significant advantages available from the large on-chip memory. Without the limitations of between chip drivers and pinout constraints, we can easily implement 1024-bit wide data busses allowing very high memory bandwidth. However, we must solve the significant technical challenges of providing such fast memory access with the lowest possible latency and in a fashion consistent with the general functionality required by the PowerPC architecture. The PEC unit which we are designing has the following features.

Core Connect Bus. The PEC interface will interface directly to the processor's the Core Connect bus. This bus runs at a frequency of 1:1 with the core. It is 128 bits wide with pipelining. The PEC interface will support all of these features. The aggregate bandwidth of the Core Connect bus is 6.4GB/s (worst case). Access to EDRAM is memory mapped. All PowerPC data transfer modes will be supported.

PLB Slave. The PEC also has a PLB slave interface which allows for reads and writes by any PLB master. Thus, the EDRAM is memory mapped as a PLB slave. Full, 128-bit transfers are supported at 166MHz. This PLB interface will permit the transfer of data to and from external SDRAM. It will also provide the instruction-fetch path for the 440 processor.

Error Checking and Correcting (ECC). The EDRAM is ECC encoded. The ECC is mapped into sectors of 64 bits which avoids the complication of read-modify-write for writes that maintain 64 bit boundaries. Writes not aligned to 64-bit boundaries are handled with read-modify-writes.

Data Prefetch. Data will always be fetched in 1024-bit lines (flines). If the data from any fline is accessed by either the CCB or the PLB, the sequential fline is fetched and put into a prefetch register. Each interface, the PLB read, the DMA read, and the CCB data read, has 2 sets of these registers allowing for ping-ponging between different areas of memory without invalidating prefetched data.

Performance. The peak performance (worst case) of the EDRAM is 16.8 GByte/s. The peak performance of the CCB bus is 6.4 GByte/s with a minimum latency of 4 processor cycles. The peak performance of the PLB bus is 2.1 GByte/s with a minimum latency of 6 processor cycles.

B.3 Serial Communications Unit (SCU)

Figure 15 provides an overview of the functions and organization of the serial communications unit. This figure shows only one of the twelve `SEND-RECEIVE` units. The stubs required by the other eleven units are shown on the pass-through module, arbiter and the PLB interface.

The `REC` unit buffers the bytes provided by the HSSL and assembles them into full 64-bit words after interpreting and stripping away various control and parity bits. It can store up to three 64-bit words so that initially the sending unit can transmit 3 words before an acknowledgment is received. If the received word is to be stored in memory, it is transferred to the `RECEIVE REGISTER` where 32 such words can be accumulated, allowing efficient transfer to the page-oriented EDRAM or DDR SDRAM. Since such transfers may come from up to twelve incoming receive ports, these final burst transfers must be sequenced by the arbiter.

The pattern of data writing or reading is controlled by a separate DMA engine for each of the twelve incoming and twelve outgoing directions. The instructions for each DMA engine will be stored in a dedicated on-chip register array. They will be specified by a chain of block-strided-move commands with enough space in each of the 16 SRAMs to permit chains of up to 16 such moves. The register array will be addressed by the communication start-up instruction allowing frequently used patterns to be kept resident in the register array and reused.

This same DMA control will store the incoming data and extract the data to be sent out, placing it in corresponding `SEND REGISTER`s. From there it is moved to the `SND_BUF` from which it is parsed into bytes, the appropriate control and parity bits added and the resulting 9 bytes sent out to the HSSL.

The pass-through unit provides low-latency global operations. A sequence of words coming in on one of the twelve input wires can be re-routed out to any combination of the output wires with minimal latency. This supports an efficient global broadcast operation as well as a low latency global sum. In addition, the first double- or quad-word to be transmitted during such an operation

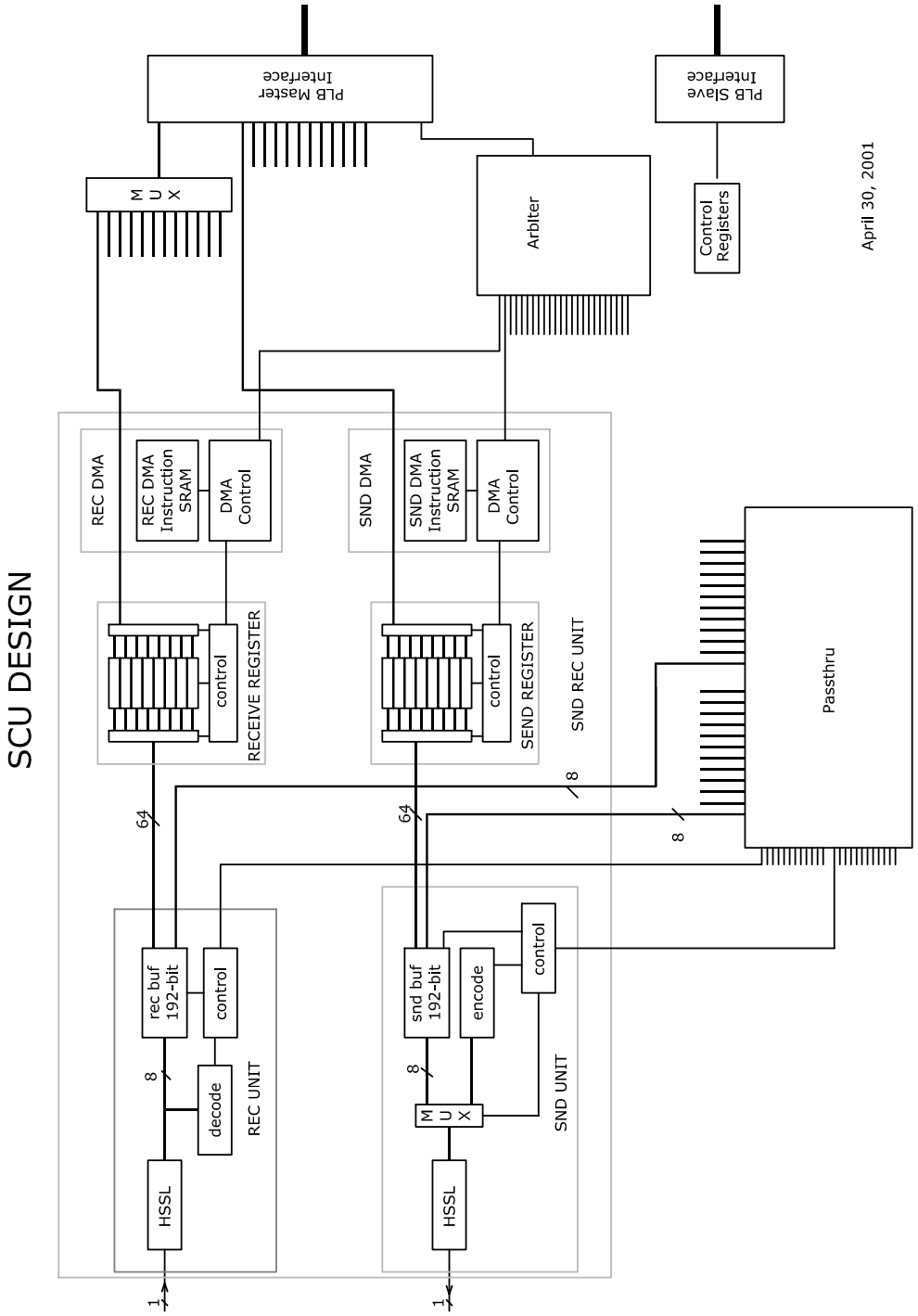

Figure 15: Block diagram for the serial control unit.

can be written directly to this unit by the 440 so that a broadcast or global sum can be started without the added latency of a DMA fetch from memory.

Two more capabilities will be included in the SCU. The first is the provision of a second type of data communication in which a double-word, written directly by the 440, is automatically sent to the corresponding neighbor and an interrupt generated for the receiving processor. These "supervisor" packets, identified by the ninth, control byte sent with the tranmitted data, can be used to efficiently implement a message passing communications protocol such as MPI. The second is the transmission of "interrupt" packets which carry eight independent interrupt bits and can be restricted to circulate within a specific partition. This permits complete isolation of code execution between independent partitions of a single machine.

C Senior Personnel

In this appendix we list the senior personnel who are participating in this project. They comprise nearly all of the senior lattice gauge theorists in the United States, as well as senior computer scientists and engineers who have agreed to join in the effort.

Claude Bernard	Washington University
Tanmoy Bhattacharya	Los Alamos National Laboratory
Richard Brower	Boston University
Thomas Blum	Brookhaven National Laboraoty
Matthias Burkardt	New Mexico State University
Shailesh Chandrasekharan	Duke University
Dong Chen	T.J. Watson Laboratories, IBM
Jie Chen	Thomas Jefferson National Accelerator Facility
Norman Christ	Columbia University
Michael Creutz	Brookhaven National Laboratory
Thomas DeGrand	University of Colorado
Carleton DeTar	University of Utah
Shao-Jing Dong	University of Kentucky
Zhihua Dong	Columbia University
Terrence Draper	University of Kentucky
Patrick Dreher	Massachusetts Institute of Technology
Anthony Duncan	University of Pittsburgh
Robert Edwards	Thomas Jefferson National Accelerator Facility
Estia Eichten	Fermi National Accelerator Laboratory
Aida El-Khadra	University of Illinois, Urbana
Rudolf Fiebig	Florida International University
Alan Gara	T.J. Watson Laboratories, IBM
Steven Gottlieb	Indiana University
Rajan Gupta	Los Alamos National Laboratory
Anna Hasenfratz	University of Colorado
Urs Heller	Florida State University
James Hetrick	University of Pacific
Donald Holmgren	Fermi National Accelerator Laboratory
Xiangdong Ji	University of Maryland
Gregory Kilcup	Ohio State University
Joseph Kiskis	University of California, Davis
John Kogut	University of Illinois, Urbana
Julius Kuti	University of Californa, San Diego
Andreas Kronfeld	Fermi National Accelerator Laboratory
Frank Lee	George Washington University
Peter Lepage	Cornell University
Keh-Fei Liu	University of Kentucky
Paul Mackenzie	Fermi National Accelerator Laboratory
Robert Mawhinney	Columbia University

Celso Mendes	University of Illinois, Urbana
Colin Morningstar	Carnegie Mellon University
John Negele	Massachusetts Institute of Technology
Shigemi Ohta	KEK and Riken BNL Research Center
Robert Pennington	National Center for Supercomputer Applications
Donald Petravick	Fermi National Accelerator Laboratory
Andrew Pochinsky	Massachusetts Institute of Technology
Claudio Rebbi	Boston University
Ronald Rechenmacher	Fermi National Accelerator Laboratory
Daniel Reed	University of Illinois, Urbana
David Richards	Thomas Jefferson National Accelerator Facility
Stephen Sharpe	University of Washington
Junko Shigemitsu	Ohio State University
James Simone	Fermi National Accelerator Laboratory
Donald Sinclair	Argonne National Laboratory
Amarjit Soni	Brookhaven National Laboratory
Robert Sugar	University of California, Santa Barbara
Eric Swanson	University of Pittsburgh
Harry Thacker	University of Virginia
Doug Toussaint	University of Arizona
Steven Wallace	University of Maryland
William Watson, III	Thomas Jefferson National Accelerator Facility
Tilo Wettig	Yale University
Uwe-Jens Wiese	Massachusetts Institute of Technology
Walter Wilcox	Baylor University