Journal of Statistical Physics, Vol. 43, Nos. 5/6, 1986

Lattice Gauge Theory with a Fast Highly Parallel Computer

N. H. Christ[1]

Results for the temperature of the color deconfinement phase transition in pure $SU(3)$ lattice gauge theory are described. These were obtained on a specially built 16-node, 256 Megaflop computer using the Metropolis algorithm. The architecture, performance, and expansion plans for this machine are also discussed.

KEY WORDS: Parallel computer architecture; lattice gauge theory; quantum chromodynamics; deconfinement phase transition.

1. INTRODUCTION

The subject of this paper is well-suited for a conference honoring Dr. Metropolis: I will outline the architecture and describe the performance of a very economical parallel computer being built by a group of theoretical physicists at Columbia to study lattice gauge theory.[1] I will present the numerical results that we have collected over the last four months studying the color-deconfinement phase transition in lattice QCD using the Metropolis algorithm.

Recognizing the fine talks on lattice gauge theory that have come earlier, it is perhaps not necessary to review lattice QCD or to discuss in detail the tremendous computer requirements of these calculations. Suffice it to say that a standard 10-hit Metropolis update of a single $SU(3)$ link variable requires approximately 5000 floating point operations, that a 32^4 lattice contains 4 million link variables, and that a typical calculation requires 10^4–10^5 updates of the entire lattice for a single choice of parameters. Including fermion loops in the computation requires at least an

This work has been supported in part by the Department of Energy and the Intel Corporation.

[1] Department of Physics, Columbia University, New York, New York, 10027.

0022-4715/86/0600-1061$05.00/0

additional factor of 10. Thus, such a calculation requires more than a year on a 1-gigaflop machine operating at 100% efficiency! Each serious lattice *QCD* research group should have its own dedicated CRAY2 or 4-processor XMP at a minimum.

Fortunately, the structure of the lattice gauge theory calculation allows the use of much more economical computer architectures. The homogeneous, local, and statistical nature of the calculations make a mesh interconnected array of pipe-lined floating point processors an ideal architecture. We have built a CRAY1 equivalent, 16-processor machine for $100,000 and are more than half way through the construction of a CRAY2 equivalent 64-node machine that will have cost less than $400,000 when it is completed. Two of us began this project almost three years ago (A. Terrano and myself). Our group has now grown to include six physics graduate students and junior faculty.

2. ARCHITECTURE

Let me begin by describing our design. Since the reliability of our off-the-shelf TTL hardware limits us to approximately 1000 processors and we expect to work on lattices of linear dimension on the order of 10 to 30, the minimum dimension for our mesh is 2. We choose the minimum as the most economical. We then divide the four-dimensional space-time lattice into blocks of two-dimensional hyperplanes storing each block in the memory of a single node. We interconnect the memories with processors so that each pair of nearest-neighbor memories is joined by at least one processor. Figure 1 shows our interconnection scheme.

The elements making up a single node are shown in Fig. 2. The combination of the 80286/80287, 32 kbyte of no-wait-state parity-checked memory and the Multibus interface is equivalent to a very fast personal computer. What makes our machine interesting is the addition of the floating point vector processor unit connected to two 64 kbyte banks of 45 ns static memory.

The vector processor is designed in the simplest fashion permitted by our lattice gauge theory application. It consists of a floating point multiplier (constructed from a TRW 16-bit integer multiplier), a floating point adder, and associated registers, as shown in Fig. 3. We use a 22-bit arithmetic word with a 16-bit significant and 6-bit exponent. Since the memory and data busses use a 16-bit word, a real number is stored in one and one-half words and a complex number in three words. The input and output registers in Fig. 3 are configured to transform between these two formats.

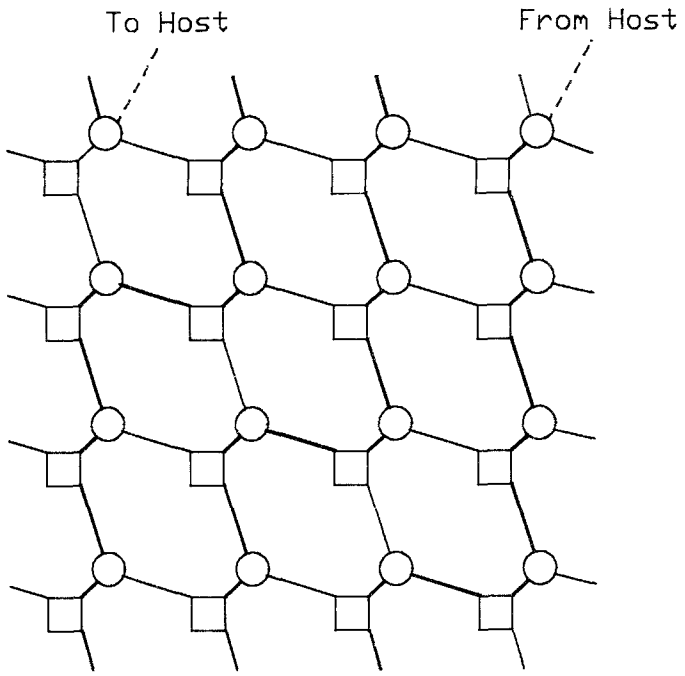

Fig. 1. The interconnection scheme for our 16-node machine. The squares represent independent memory elements while the circles stand for processors. The paired memories and processors are located on the same board. The heavy lines indicate a one-dimensional path determined in software which is used to load and unload the machine from the host computer.

The control of the vector processor, detailed in Fig. 4, is a central element of our design. On the lowest level, the vector processor is microcode-controlled, with the microcode stored as 4096, 56-bit words. However, this microcode memory is addressed by the most primative sequencer: a counter which simply steps sequentially through a series of addresses beginning with an initial address loaded by the 80286 and continuing until a microcode "stop" bit becomes true. Any branching or looping is done by the microprocessor which must restart the vector processor—an operation that we have worked to make very efficient.

The eight base address registers that locate the arguments in memory for the vector processor are divided into two groups of four: while one group is being used by the vector processor, the second group is available to be reloaded by the 80286. The three remaining operations necessary to restart the vector processor have been compressed into a single 80286 move instruction. During that move operation data is loaded into the latch, which configures the data paths, and the microcode address counter started

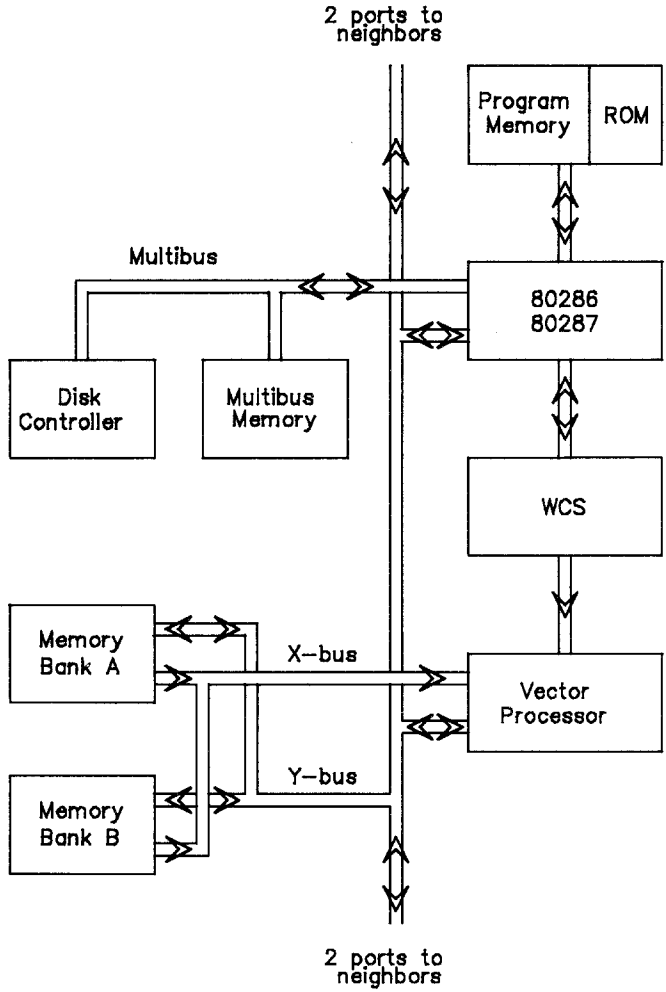

Fig. 2. The elements making up a single node of the computer.

at an address given by the lower address bits of destination of the move operation. Less than 1 μs passes between the end of one vector processor operation and the start of the next.

The interprocessor connection shown in Figs. 1 and 2 requires some further explanation. As these diagrams suggest, we simply connect the data busses between neighboring processors. One of the two banks of data memory (that feeding the X bus in Fig. 2) is always providing data to the local vector processor. The second data bank is connected by the Y bus

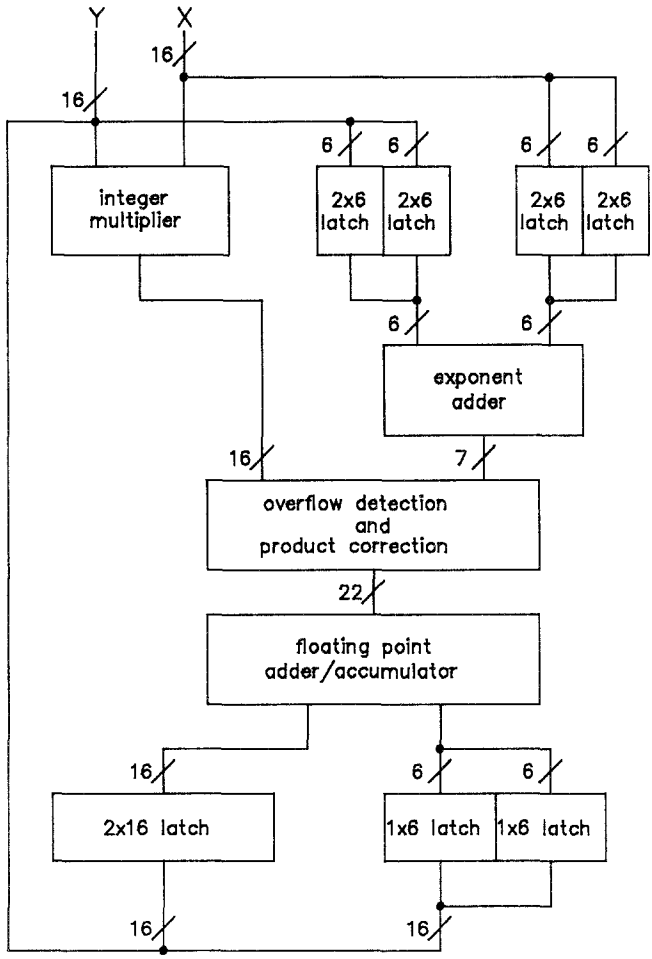

Fig. 3. A block diagram of the data handling components of the vector processor.

either to the local vector processor or to the vector processor in the $-x$ or $-y$ directions. During such an interboard operation (either reading or writing) the local vector processor is joined to the corresponding bank of the neighboring node in the $+x$ or $+y$ direction. The addresses for both memory banks are provided by the local vector processor and these off-board transfers require lock-step, synchronous operation for all communicating boards.

The final element of the computer is the "central controller" whose interconnection with the individual nodes is represented in Fig. 5. This is a

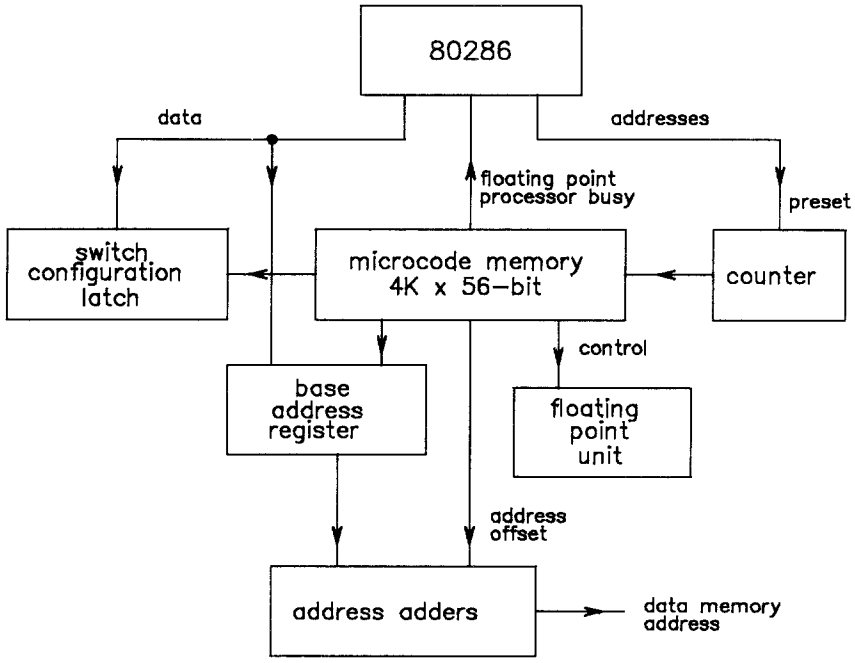

Fig. 4. The control circuitry for the vector processor.

single board which generates a 12-MHz clock signal for the 80286 and a 8-MHz clock signal for the vector processor. It is the only element of the machine connected to the host computer (a VAX11/780). Under instruction from the host it will interrupt or reset the array of processors. Conversely it will relay a finished or error signal from the array to the host. Finally it will resynchronize the array after receiving a request to that effect from all nodes.

3. PROGRAMMING

The machine is programmed on three levels. At the bottom one has the microcode driving the vector processor. These microcode programs are treated as subroutines typically performing a few hundred floating point operations. For example, the lattice gauge theory calculations require microcoded routines which multiply a pair of $SU(3)$ matrices, calculate the exponential of a real number or multiply a pair of $SU(3)$ matrices, and add the product to a third matrix. These programs are created by a microcode assembler which runs on the VAX.

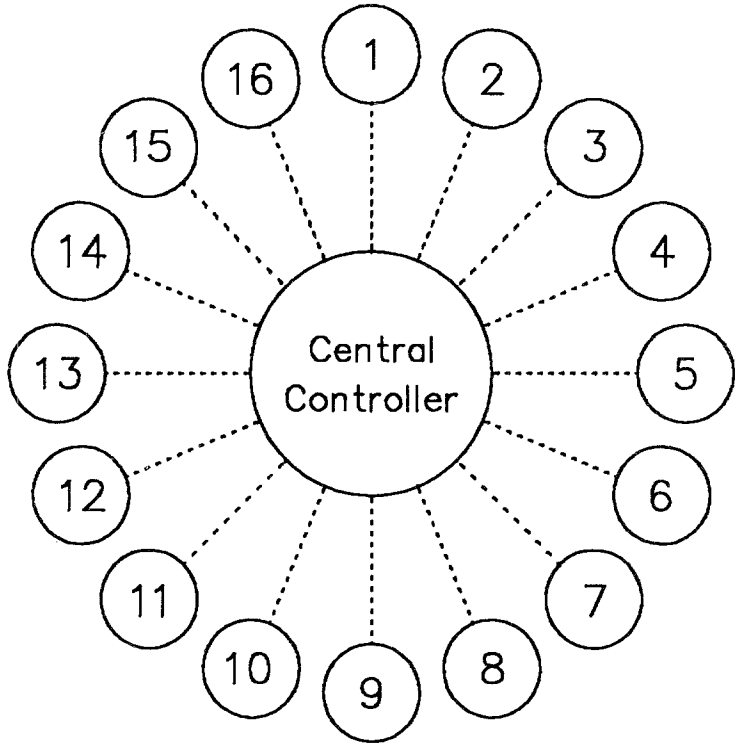

Fig. 5. The "radial" connection of each node to the central controller. Each dotted line represents a ribbon cable carrying eight signals.

The next layer of programing directs the microprocessor's control of the vector processor. This code must execute very efficiently to keep pace with the vector processor and is written in assembly language. A well-designed application program will use microcode routines of at least a hundred lines. This will allow the 10 μs required by an 80286 assembly language program to concurrently prepare for the next vector processor operation.

Finally, on the highest level, one has a Fortran or PL/M program which moves data from the disks or Multibus memory and calls the more primative subroutines.

4. PERFORMANCE

Figure 6 shows the 16-node machine. Each board has its own four-slot Multibus card cage, each with two $\frac{1}{2}$ Mbyte commercial memory boards. Each fourth card cage also contains a disk controller connected to a

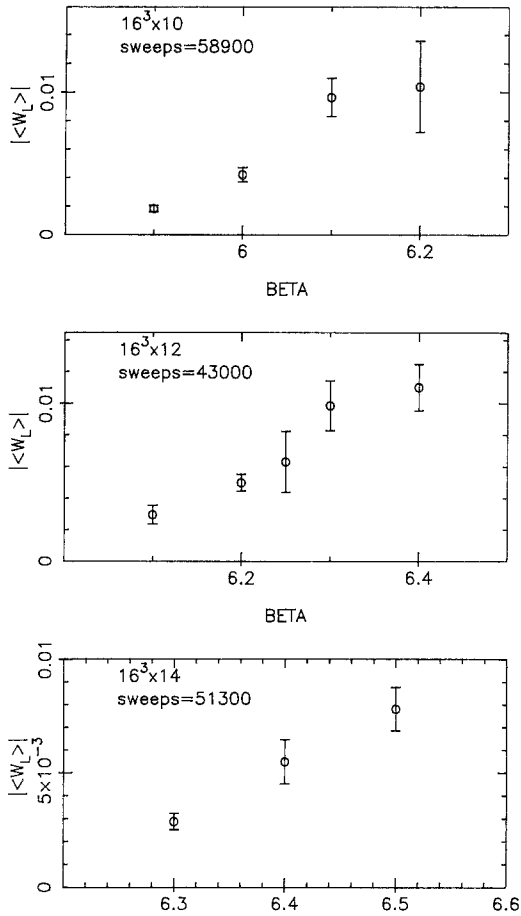

Fig. 6. The magnitude of the expectation value of the Wilson line operator, averaged over blocks of 100 Monte Carlo sweeps, as a function of β for lattices with a spacial volume of 16^3 and 10, 12, and 14 sites in the time direction.

500 Mbyte Winchester disk. Thus the 16-node machine has 16 Mbytes of Multibus memory and 2 Gbytes of disk storage. The individual boards measure 12×18 in and cost under \$3000 to fabricate.

The performance of the machine can be best described by the execution time required by various lattice gauge theory programs. For example, a $SU(3)$ matrix multiplication can be performed by a microcode program of 145 lines, which implies a computational rate of 11 Mflop/s. The sequence of many such multiplications and accumulations needed to compute the total action of the system under study is not as efficient and runs at approximately 7 Mflop/s. The evaluation of an exponential is a less

vectozized operation that is performed to machine accuracy in 10 μs by the vector processor. A complete Metropolis program that carries out a full statistical simulation (generation of random numbers, computation of exponentials, and, in general, considerable scalar operation) is usually characterized by the time required to update a single 3×3 complex matrix. This update time is 20 ms for a VAX11/780, 2 ms for a CDC 7600, and 80 μs for a Cray-1. For our 16-node machine the corresponding time is 180 μs. We have made changes to a single board, which when made to all 16, will halve this time.

5. COLOR DECONFINING PHASE TRANSITION

Finally, I would like to describe the calculation that we are presently doing with the machine. Recall that the lattice version of non-Abelian gauge theory approximates space-time by a four-dimensional mesh of points with lattice spacing a. The gauge variables are associated with the links joining neighboring sites: an $SU(3)$ matrix U_l is introduced for each link l, so that an observable $O(U)$ can be computed from the path intergral

$$\langle O \rangle = \prod_l \int d[U_l] \exp\left\{\frac{2}{g_0^2} \sum_p \operatorname{re}\,\operatorname{tr}\left[\prod_{l \in p} U_l\right]\right\} \Big/ \prod_l \int d[U_l] \times \exp\left\{\frac{2}{g_0^2} \operatorname{re}\,\operatorname{tr}\left[\prod_{l \in p} I_l\right]\right\} \tag{1}$$

Here the sum in the exponent is over all elementary plaquettes p in the lattice.

Of course, the lattice used in an actual evaluation of (1) necessarily corresponds to a finite space-time volume. When performed in such a finite volume with periodic boundary conditions in the time direction, the Euclidean path integral in (1) is actually a calculation of

$$\frac{\operatorname{tr}\{e^{-H/kT} O\}}{\operatorname{tr}\{e^{-H/kT}\}} \tag{2}$$

where the temperature T is related to the number of sites N_t in the time dimension of the lattice by $kTa = 1/N_t$.

The quantity that we have been studying is the temperature of the color deconfinement phase transition T_c. Since the lattice spacing a is the only dimensioned parameter in the calculation, T_c has the form

$$T_c = t(\beta)/a \tag{3}$$

where the conventional parameter $\beta = 6/g_0^2$. We can then adjust β until the critical temperature T_c equals the temperature of the lattice $T = 1/N_t a$. If we call that value β_c, then

$$1/N_t = t(\beta_c) \tag{4}$$

and varying N_t allows us to determine $t(\beta)$.

This function is of special interest because it is predicted by continuum perturbation theory

$$t(\beta_c) = c\left(\frac{1}{g_0^2}\right)^{51/121} e^{-(24\pi^2/33g_0^2)}[1 + O(g_0^2)] \tag{5}$$

up to the constant c. Comparison of $t(\beta)$ with (5) is an important test that the lattice spacing a has been chosen sufficiently small.

The critical value of β can be recognized by measuring the expectation value of the Wilson line operator

$$W_L(U) = \mathrm{tr}\left\{\prod_{l \in L} U_l\right\} \tag{6}$$

This operator is constructed as the trace of the product of the link matrices lying along a line L in the time direction. It is the large mass limit of the propagator of a single quark fixed at the spacial position of the line L. We can increase our statistics by averaging over all possible lines. For $T < T_c$, the theory is confining and such a isolated quark has an infinite energy, so the expectation value of W_L should vanish. For $T > T_c$, the massive quark has a finite self-energy E_q. Thus we expect

$$\langle W_L(U) \rangle = 0 \qquad T < T_c \tag{7a}$$

$$\langle W_L(U) \rangle = \alpha e^{-E_q/kT} \qquad T > T_c \tag{7b}$$

Unfortunately, this hypothesis has two difficulties. Even if $\langle W_L(U) \rangle \neq 0$ for $T > T_c$, it is still very small since the self-energy of a massive point charge is linearly divergent in the limit $a \to 0$. Second, for a finite volume with periodic boundary conditions, we must worry about the inconsistency of Gauss's law if only a single charge is present in the volume. This is realized by the presence of a zero mode, the integration over which makes the expectation value in (10) vanish for any value of T.

For the non-Abelian theory, this zero mode is the symmetry $U_l \to zU_l$ for all time-like links l in a particular time plane. Here z is one of the cube roots of unity. Since this Z_3 symmetry transformation multiplies $\langle W_L(U) \rangle$ by z, integration over these three transforms of many gauge-field con-

figuration makes $\langle W_L(U) \rangle$ vanish. As the volume becomes infinite such global changes of phase become impossible. Thus for a finite volume calculation we must limit the length of Monte Carlo time over which we average to avoid these jumps in Z_3 phase. This in turn limits the precision with which (7a) can be measured.

Our results to date for a lattice of spacial volume 16^3 are shown in Fig. 6. For each of the three temporal dimensions, the magnitude of the Wilson line is plotted as a function of β. This quantity is obtained from our Monte Carlo calculation by first averaging the trace in (6) over all 16^3 lines in our spacial volume and then averaging over blocks of 100 Monte Carlo sweeps. The magnitudes of the complex numbers obtained from each block are then averaged and the fluctuations among them (including the effects of correlations) used to compute the errors. We interpret the rise in $|\langle W_L(U) \rangle|$ over the narrow range of β shown in Fig. 6 as arising from the deconfining phase transition. Assuming that the critical value of β lies in the region of greatest increase, we deduce that β take values $6.05 \pm .05$, $6.275 \pm .025$, and $6.40 \pm .05$ for lattices with $N_t = 10, 12$, and 14, respectively. (These preliminary conclusions are confirmed and strengthened by a more detailed analysis in Ref. 2.)

These results are compared to values of β obtained earlier[3] on smaller lattices and to the predictions of scaling in Fig. 7. The essentially

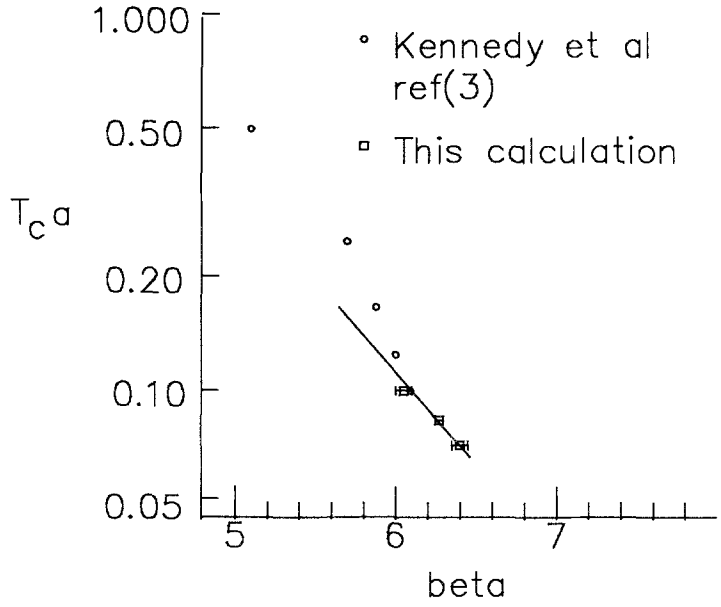

Fig. 7. The color-deconfinement critical temperature as a function of β. The solid line has the slope predicted by the continuum renormalization group eq. (5).

straight line shown in the figure is the prediction of (5) with the multiplicative constant adjusted to make the curve pass through our three points. As can be seen, the region $\beta > 6.05$ shows a different β dependence than that seen for lower β values, a behavior consistent with the predictions of continuum perturbation theory and the renormalization group. Thus we tentatively conclude that for $\beta > 6.05$ the lattice spacing $a = .1/kT_c$ is sufficiently small that the continuum theory is being described.

This is a very positive conclusion. If realistic calculations require a choice of β making physical lengths on the order of 10 lattice spacings, then lattices of perhaps 32 sites on a side will be adequate to make real numerical tests of QCD. Lattices of this size are certainly quite feasible on the 64-node machine mentioned earlier. However, the inclusion of Fermion loops on such a lattice will most likely require the 256-node machine that is the final phase of our project. With some good fortune, this 4 Gflop computer should be operational in $1\frac{1}{2}$ years.

REFERENCES

1. N. Christ and A. Terrano, *IEEE Trans. Comput. C* **33**:344 (1984).
2. N. Christ and A. Terrano, *Phys. Rev. Lett.* **56**:111 (1986).
3. A. Kennedy, J. Kuti, S. Meyer, and P. J. Pendelton, *Phys. Rev. Lett* **54**:87 (1985).