A Multi-Grid Cluster Labelling Scheme

John Apostolakis
Paul Coddington
Enzo Marinari

CRPC-TR91155
June, 1991

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

SCCS-103
CRPC-TR91155

A MULTI-GRID CLUSTER LABELING SCHEME

John Apostolakis, Paul Coddington
and Enzo Marinari[a]

Physics Department,
Syracuse University,
Syracuse, N.Y. 13244, U.S.A.

June 6, 1991

Abstract

We introduce a simple multi-scale algorithm for connected component labeling on parallel computers, which we apply to the problem of labeling clusters in spin and percolation models. We show that it is only logarithmically slowed down in the critical limit of bond percolation and the Ising model. We also discuss, in light of the proposed Teraflop computers optimized for lattice gauge theories and other lattice problems, the minimum requirements for simple computer switchboard architectures for which one can efficiently implement multi-scale algorithms to fight critical slowing down.

[a]: Permanent Address: Dipartimento di Fisica, Università di Roma *Tor Vergata*, Via E. Carnevale, 00173 Roma, Italy.

Our understanding of critical phenomena, connected to the divergence of some typical length scale of a system, has benefited greatly from large scale numerical simulations. Large computer power and memory allow us to probe the theories deep into the critical region, allowing a new and better understanding of renormalization group and scaling ideas. Critical slowing down is the typical drawback of a large scale simulation close to criticality: if the correlation length ξ of the system is becoming large, a local dynamics will have a correlation time (the number of iterations needed to generate a statistically independent configuration) $\tau \sim \xi^z$, where the dynamical critical exponent $z \geq 2$ [1, 2].

The proposal of Swendsen and Wang [3] to combat critical slowing down exploits the Fortuin-Kasteleyn representation [4] of the Ising spin model [5]. The partition function can be rewritten as a sum over percolation clusters (or, according to the very illuminating language of ref. [6], as a joint spin-link model). Using this equivalence we can implement non-local changes of the system by updating whole clusters of spins at a time. Although in many cases critical slowing down is not completely eliminated [7], the method is generally effective because the maximum cluster size diverges where the correlation length diverges, so we can make very large non-local changes at the critical point.

The dominant computational aspect of this algorithm is the identification of the clusters. This is the classic problem of connected component labeling [8, 9]: on a d-dimensional lattice of $V = L^d$ sites, we have links between neighboring sites which can be *on* (a connection) or *off* (no connection), and our goal is to end up with the same label on all connected sites, with different labels for all disconnected clusters.

Let us define a computational exponent y for this problem, so that the time to label the connected components scales asymptotically as L^{d+y}. For a Monte Carlo simulation, this cost is compounded by the number of iterations needed to generate an independent configuration, which scales as L^z at the critical point [1, 2]. On a serial machine, labeling algorithms are known which are guaranteed to scale not worse than $V \log V$, so that $y = 0$ (see refs. [8, 10, 11, 12] for a discussion of sequential labeling algorithms). This means that the overall computational cost of a Monte Carlo cluster updating algorithm at the critical point will scale as L^{d+z}. For a general labeling algorithm, the overall cost will also include a factor L^y for the component labeling. Hence if for some reason we cannot use a labeling algorithm for

which y is zero, the advantages of cluster update algorithms over traditional local algorithms may be offset by the computational complexity of labeling the clusters.

This could be the case if our computer were parallel rather than sequential, in which case the problem of component labeling is far more complicated [10]. Here the information regarding the connectivity of a given physical part of the lattice is only contained in a single processor, and sending information far away (on the lattice, and hence also on the computer) can take a long time. Our aim is to find a parallel labeling algorithm with no computational slowing down (i.e. $y = 0$). Because of our main interests, which we will clarify in the following, we will consider here the case where the parallel computer is a Single Instruction Multiple Data (SIMD) machine.

In order to handle critical slowing down, a parallel computer will need some physical switching capabilities allowing fast and effective communications between regions that are far away in the physical space of the numerical simulation. It does not seem possible, deep in the critical regime, to completely avoid all forms of non-local data transmission. But since elaborate patterns of switching networks require complex hardware and increase the cost and the potential failure rate of the system, we are interested in finding algorithms which run on machines with simple communication networks, and have good asymptotic scaling behavior.

Our effort is thus also aimed towards a better comprehension of the features that will be required in the next generation of computers optimized for lattice QCD. On machines with a speed of the order of a Teraflop (see refs. [13, 14, 15, 16] for information on the different projects, some of which are already at a very advanced stage), lattice QCD will be studied at very large correlation lengths, for both the gauge interactions and meson masses. It seems that the use of some non-local algorithms will be, at this point, quite essential: effective multi-grid methods will be probably found for inverting the quark propagators, and cluster algorithms could play an important role in speeding up the pure gauge dynamics. Many of these machines will have some kind of SIMD architecture, and the topology of the switching network will be very important. For a dedicated machine, keeping the hardware as simple as possible is a crucial goal, but on the other hand it should be capable of supporting all the relevant algorithms.

The ideas which make cluster algorithms effective are shared by multigrid methods (see for example the Alan Sokal lecture notes [17], and the fact

that the same lower bound for the dynamical critical exponent z applies both to cluster schemes and to the mixed cluster multi-grid schemes [18]). The main problem is including what we know about the large distance behavior of the system in the updating schemes.

In this note we propose a regular, completely synchronous, multi-scale algorithm for cluster labeling. We show that it does not undergo any power-law computational slowing down (i.e. $y = 0$) in the two cases of the 2-d bond percolation and Ising models at their critical points. The algorithm is effective on a general SIMD machine with some very basic non-local connections. In the following we will assume that the machine allows very fast communication between sites which are a distance of 2^m sites away in any direction of the physical lattice. These are the only non-local connections we need in order to build an algorithm which is not affected by power-law slowing down. Note that these are the same connections that are required in order to efficiently implement a Fast Fourier Transform algorithm. Such connections would be provided, for example, by a machine with a hypercube topology. We think that the results we present here strongly support the need for some kind of effective non-local communication for the next generation of Teraflop computers.

Our method has some similarities with the one proposed by Brower, Tamayo and York [19], in that it is a SIMD, multi-grid style algorithm, however it is much simpler, and seems to have better scaling properties. We will report in a separate paper [20] on some variations and improvements to this algorithm, and the performance of these different methods, along with a more detailed comparison to the method of Brower *et al.*

The simplest local component labeling algorithm on a massively parallel computer is based on label propagation [10, 19]. We start with a different label on each site, and with a list of first neighbor connections which indicate whether a given pair of sites is connected or not (these will be Boolean variables in the following: *off* means no connection is present and *on* means that there is a connection). Each site then looks to its left (and then above, right and below), and if it is connected to this neighbor, then it takes the neighboring label, if it is less then its own label. Eventually an equilibrium situation is reached which gives a correct labeling of the clusters, with the label of a cluster being the minimum initial label of all the sites in that cluster. This algorithm suffers from computational slowing down, and in the cases of interest (spin and percolation models at the critical point) its

performance degrades very fast with increasing volume, with $y \approx 1$. This is because clusters at a second order phase transition are dominated by a large cluster whose diameter is of order L, and for any local labeling algorithm, the minimum label has to diffuse across this large cluster.

Our method is based on two main ideas. The first is to use a *multi-scale* approach in propagating cluster labels. Boolean connections at a distance 2^m, for $m = 1, ..., l-1$ (where the lattice size $L \equiv 2^l$), are built in the x and the y direction (in the $d = 2$ case) by a logical AND of connections at level $m-1$. For example, the distance 2 connection between sites i and $i + 2\hat{x}$ is set *on* if sites i and $i + \hat{x}$ and sites $i + \hat{x}$ and $i + 2\hat{x}$ are both connected. This is done in all directions of the lattice, for all levels up to distance $L/2$. The algorithm works for a lattice of any dimensionality.

The second idea is that inter-site connections can be *improved.* This means that a connection (between two sites at a generic distance M) which was originally *off* can be declared to be *on* if at any time during the labeling process the two sites are found to have the same label, in which case we know that they must belong to the same connected cluster. Clearly a connection between two sites at a distance M can in this way be *on* even if the direct path between the two sites contains sites belonging to different connected components: the existence of an *on* connection between two sites only implies that a connected path joining the two sites exists. Using connection improvement greatly reduces the number of iterations needed to converge to the final values of the labels.

Thus, during one multi-scale label updating cycle each site will look in turn at each of its $2d$ neighbors at all levels m of the multi-scale connections. It will update its label when necessary and *update its connection*, by merging the level $m-1$ connections, and also using connection improvement. A full cycle of the algorithm sweeps all l connection levels, and a single full cycle solves the trivial case where all connections are *on*.

Clearly at the beginning of the dynamical procedure the long distance connections are all *off*, and due to the fractal structure of the connections, it will take many iterations before a significant number of long distance connections are improved. It is thus very useful to tailor the number of multi-grid levels as a function of the cycle number [20]: a lower depth is useful at the beginning, while using longer distance connections is more useful towards the end of the procedure. Here we just show results for the simplest case, where the depth is constant. What eventually happens is that some of the appropri-

ate long distance connections are set *on* since the sites have been recognized to belong to the same cluster, and they become fast long distance communication channels (displaying the labels using color graphics shows dramatic changes due to merging of large precursor sub-clusters as these channels are opened up).

We have implemented our code on the Connection Machine, which is a typical massively parallel SIMD computer [21]. Here the mapping of the hypercubic connections to the physical structure of the lattice does provide some specific *power of* 2 communications, which are are executed at half the speed of local communications. These are just the type of fast non-local communications which we need for the efficient implementation of our algorithm.

In order to test the algorithm we have analyzed two typical and physically relevant cases: the bond percolation model (sites are connected with probability p), and the Ising model, both in two dimensions. The worst case behavior of the labeling algorithm is not relevant for these problems - what we are really interested in is the *average* time to label physically realistic configurations which occur in the simulation of these models. This time is greatest at the critical points of the models [19], which are at $p = \frac{1}{2}$ for the percolation model, and the Curie temperature for the Ising model. We have obtained our data by averaging over a large number of different realizations of the site connections, taken from configurations at the critical points of the two models, in order to get statistically significant results from which we can obtain the scaling behavior of our algorithm.

In Fig. 1 we show the average number of iterations needed for labeling as a function of $\log_2 L$ for the case of bond percolation (circles) and the Ising model (squares). The logarithmic slowing down is very clear. We do not see any sign of power-law behavior (so $y = 0$), or of a higher power of the logarithm. Since each iteration of the algorithm involves a multi-grid cycle of $\log_2 L$ steps (each step taking the same amount of time), the total computational complexity goes as $L(\log L)^2$. Hence this algorithm adds only a $(\log L)^2$ term to the overall slowing down of a spin model cluster algorithm.

We have experimented with many variations and optimizations of the algorithm (see ref. [20]). These can greatly increase the performance of the algorithm, and make a substantial difference in a realistic simulation using large lattices.

Our algorithm is very general, and can be applied to any of the cluster

algorithms which have been shown to reduce critical slowing down in many different systems (for reviews of cluster algorithms, see refs. [1, 2, 22, 23]). The algorithm could also be used in other applications of component labeling, such as image analysis [9]. As an example, the issue of component labeling is relevant to the next generation of high energy physics experiments, since on-line reconstruction of traces will demand fast labeling algorithms. This problem is being studied, for example, in relation to the design of the Superconducting Supercollider (SSC) detectors [24].

We believe that algorithms of this kind, together with SIMD computers with a simple but effective communication network, will be crucial tools in future studies of critical phenomena and of lattice field theories in the continuum limit.

Acknowledgements

This work was done using Connection Machines at the Northeast Parallel Architecture Center at Syracuse University, Sandia National Laboratory, Rice University, and the Advanced Computing Laboratory at Los Alamos National Laboratory. Work supported in part by the Center for Research on Parallel Computation with NSF cooperative agreement No. CCR-8890615, and a grant from the IBM Corporation.

References

[1] A. D. Sokal, in *Computer Simulation Studies in Condensed Matter Physics: Recent Developments*, eds. D. P. Landau *et al.* (Springer-Verlag, Berlin-Heidelberg, 1988).

[2] A. D. Sokal, *How to Beat Critical Slowing Down - 1990 Update*, in Proc. of Conference 'Lattice 90', Tallahassee, October 1990, to be published in Nucl. Phys. B (Proc. Suppl.).

[3] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

[4] C. M. Fortuin and P. W. Kasteleyn, Physica **57**, 536 (1972).

[5] See for example B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model,* (Harvard University Press, Cambridge, Mass., 1973).

[6] R. G. Edwards and A. D. Sokal, Phys. Rev. D **38**, 2009 (1988).

[7] X.-J. Li and A. D. Sokal, Phys. Rev. Lett. **63**, 827 (1989).

[8] E. M. Reingold, J. Nievergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice* (Prentice-Hall, Englewood Cliffs, N.J., 1977); E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, (Computer Science Press, Rockville, Maryland, 1978).

[9] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, (Academic Press, New York, 1982).

[10] C. F. Baillie and P. D. Coddington, *Cluster Identification Algorithms for Spin Models - Sequential and Parallel,* to be published in Concurrency: Practice and Experience.

[11] R. G. Edwards, X.-J. Li and A. D. Sokal, *Sequential and Vectorized Algorithms for Computing the Connected Components of an Undirected Graph*, in preparation.

[12] E. Marinari and C. Rovelli, in preparation.

[13] N. Avico *et al.*, *A 100 Gigaflops Parallel Computer,* Roma *La Sapienza* preprint 733 (Roma, Italy, April 1990).

[14] S. Aoki *et al.*, *Proposal for a Lattice Gauge Theory Teraflops Computer,* 1991 (unpublished).

[15] M. Fischler *et al.*, Nucl. Phys. **B** (Proc. Suppl.) **17**, 263 (1990).

[16] Y. Iwasaki *et al.*, Nucl. Phys. **B** (Proc. Suppl.) **17**, 259 (1990).

[17] A. D. Sokal, *Multi-Grid Monte Carlo for Lattice Field Theories,* lectures given at the Winter College on "Multilevel Techniques in Computational Physics", ICTP, Trieste, January 1991.

[18] X.-J. Li and A. D. Sokal, to be published.

[19] R. C. Brower, P. Tamayo and B. York, J. Stat. Phys. **63**, 73 (1991).

[20] J. Apostolakis, P. Coddington and E. Marinari, in preparation.

[21] D. Hillis, *The Connection Machine*, (MIT Press, Cambridge, Mass., 1985).

[22] U. Wolff, Nucl. Phys. **B** (Proc. Suppl.) **17**, 93 (1990).

[23] J.-S. Wang and R. H. Swendsen, Physica A **167**, 565 (1990).

[24] R. Rusack, private communication.

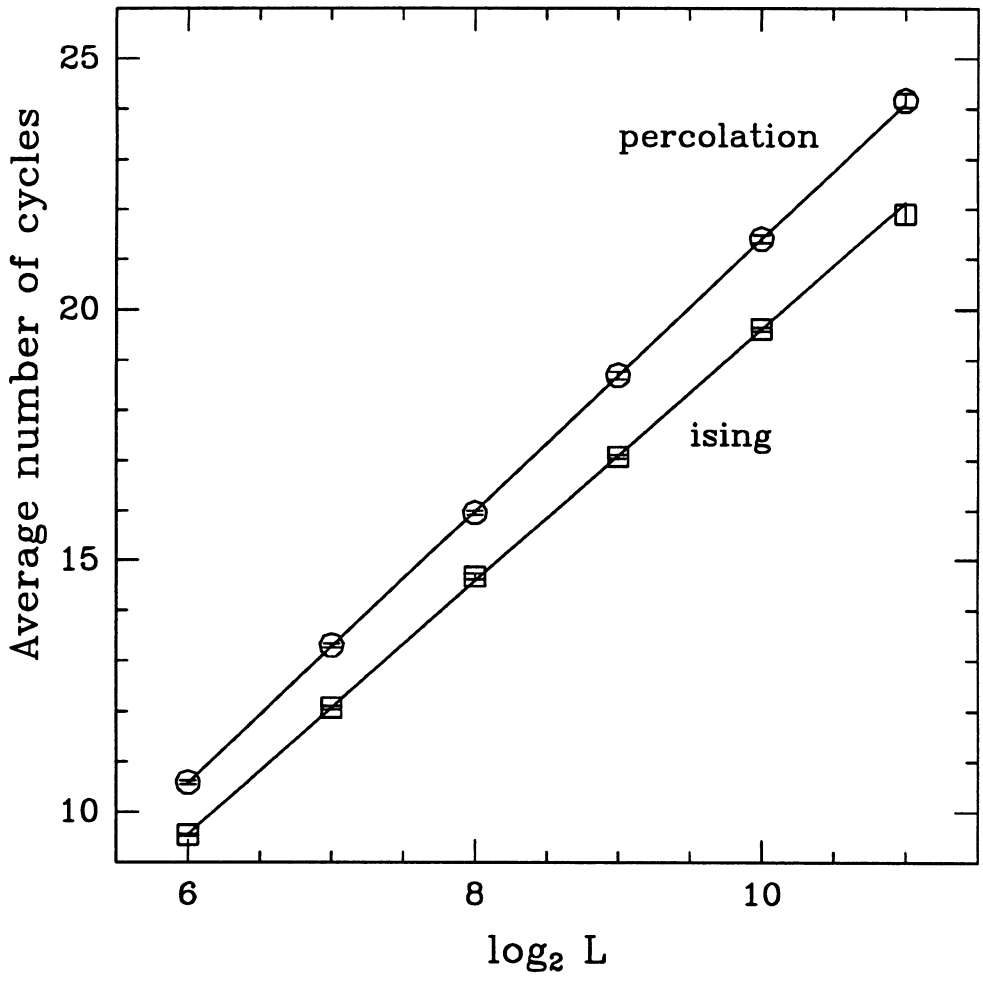

Figure 1. Number of multi-grid cycles needed to converge as a function of $\log_2 L$ for the case of bond percolation (circles) and the Ising model (squares), both at the critical point in two dimensions. The lines are χ^2 fits to the points. The number of configurations ranges from 100 to 1000 for the different values of L.