



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 both at the critical point in two dimensions. The data lines are a linear fit to the data. The number of configurations ranges from 100 to 1000 for values of L .

- [13] N. Avicco *et al.*, *A 100 Gigafllops Parallel Computer*, Roma *La Sapienza* preprint 733 (Roma, Italy, April 1990).
- [14] S. Aoki *et al.*, *Proposal for a Lattice Gauge Theory Terafllops Computer*, 1991 (unpublished).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 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**, 103 (1990).
- [20] I. Apostolakis, P. Coddington and E. Marinari, in preprint.
- [21] D. Hillis, *The Connection Machine*, (MIT Press, Cambridge, MA, 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.

- [4] J. 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.).
- [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.), 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] F. Baillie and P. D. Coddington, *Cluster Identification Algorithms - Spin Models - Sequential and Parallel*, to be published in *Concurrent Practice and Experience*.
- [11] R. G. Edwards, X. - J. Li and A. D. Sokal, *Sequential and Vertex Algorithms for Computing the Connected Components of an Undirected Graph*, in preparation.
- [12] E. Marinari and C. Rovelli, in preparation.

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

We believe that algorithms of this kind, together with a simple but effective communication network, will be useful in future studies of critical phenomena and of lattice models in the continuum limit.

Acknowledgements

This work was done using Connection Machines at the National 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 Parallel Computation with NSF cooperative agreement N00014-90-1-0959 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, in Nucl. Phys. B (Proc. Suppl.).
- [3] R. H. Swendsen and J. - S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

at long distance connections are set *on* since the sites to belong to the same cluster, and they become fast long communication channels (displaying the labels using color changes due to merging of large precursor sub-clusters opened up).

We have implemented our code on the Connection Machine, a typical massively parallel SIMD computer mapping of the hypercubic connections to the physical structure of the some specific *power of 2* communications, which are as fast as the speed of local communications. These are just the local communications which we need for the efficient implementation.

In order to test the algorithm we have analyzed two typically relevant cases: the bond percolation model (site probability p), and the Ising model, both in two dimensions. The behavior of the labeling algorithm is not relevant for us, we are really interested in is the *average* time to label per configurations which occur in the simulation of these models. The greatest at the critical point, which has been studied in the percolation model, and the Curie temperature for the Ising model. We obtained our data by averaging over a large number of different site connections, taken from configurations at the two models, in order to get statistically significant results. We obtain the scaling behavior of our algorithm.

In Fig. 1 we show the average number of iterations needed as a function of L for the case of bond percolation (circles) and Ising model (squares). The logarithmic slowing down is not seen, any sign of power-law behavior (so $y = 0$), or of a L^2 term in the logarithm. Since each iteration of the algorithm is a cycle of $2L$ steps (each step taking the same amount of time), the computational complexity is $O(L^2)$. This is a $(1/L)^2$ term to the overall slowing down of a spin model cluster.

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

Our algorithm is very general, and can be applied to a

performance degrades very fast with increasing volume, because clusters at a second order phase transition are cluster whose diameter is of order L , and for any local the minimum label has to diffuse across this large cluster.

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

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

Thus, during one multi-scale label updating cycle each site turns at each of its $2d$ neighbors at all levels m of the multi-scale. It will update its label when necessary and update its connections at the level $m-1$ connections, and also using connections at level m . A cycle of the algorithms weeps all l connection levels, and solves the trivial case where all connections are *on*.

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

that the same lower bound for the dynamical critical exponent to cluster schemes and to the mixed cluster. The multi-grid main problem is including what we know about the large d of the system in the updating schemes.

In this note we propose a regular, completely synchronous algorithm for cluster labeling. We show that it does not law computational slowing down (i.e. $\gamma = 0$) in the two cases percolation and Ising models at their critical points. It runs on a general SIMD machine with some very basic non-local communication between sites which is always in one direction of the physical lattice. These are the only non-local connections in order to build an algorithm which is not affected by d slowing down. Note that these are the same connections that are efficiently implemented in a Fast Fourier Transform algorithm. We think that the results we present here strongly support a kind of effective non-local communication for the next generation of computers.

Our method has some similarities with the one proposed by Tamayo and York [1] that it is a SIMD, multi-grid style algorithm however it is much simpler, and seems to have better scaling. We will report in a separate paper [2] on extensions and improvements to this algorithm, and the performance of these different algorithms. For a more detailed comparison to the method of Brower *et al.*

The simplest local component labeling algorithm on a computer is based on label propagation [3] where each site has a different label on each site, and with a list of first neighbor connections. Whether a given pair of sites is connected or not (the variables in the following: *off* means no connection is present, *on* that there is a connection). Each site then looks to its right and below), and if it is connected to this neighboring site with a lower label, if it is less than its own label. Eventually a situation is reached which gives a correct labeling of clusters. The label of a cluster being the minimum initial label of any site in the cluster. This algorithm suffers from computational slowing down in the cases of interest (spin and percolation models at transition points).

Our understanding of critical phenomena, connected to some typical length scale of a system, has benefited from large scale numerical simulations. Large computer power and the ability to probe the theories deep into the critical region, allowing for a better understanding of renormalization group and scaling ideas, is the typical drawback of a large scale simulation. If the correlation length ξ of the system is becoming large, the simulation will have a correlation time (the number of iterations needed to get statistically independent configurations) that diverges as the critical exponent z increases [1].

The proposal of Swendsen and Wang [2] for critical slowing down exploits the Fortuin-Kasteleij representation of the Ising model [5]. The partition function can be rewritten as a sum over percolation clusters (or, according to the very illuminating analogy of the Ising model). Using this equivalence we can implement non-local updates by updating whole clusters of spins at a time. Although critical slowing down is not completely eliminated, it is significantly reduced and effective because the maximum cluster size diverges when the correlation length diverges, so we can make very large non-local changes at each point.

The dominant computational aspect of this algorithm is the labeling of the clusters. This is the classic problem of connected components in a d -dimensional lattice of W links between neighboring sites which can be *on* (a connection) or *off* (no connection). Our goal is to end up with the same label on all connected sites. Labels for all disconnected clusters.

Let us define a computational exponent y for this problem as the time to label the connected component ξ^y of a system. In a Monte Carlo simulation, this cost is compounded by the time needed to generate an independent configuration, which is ξ^z for a critical system. On a serial machine, labeling algorithms which are guaranteed to scale as ξ^y are not available [6, 7]. For a discussion of sequential labeling algorithms, see [8, 9]. This means that the overall computational cost of a Monte Carlo simulation at the critical point is ξ^{y+z} . For a labeling algorithm, the overall cost will be ξ^y if $y < z$ compared to a labeling algorithm. Hence if for some reason we cannot use a labeling

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 switch-board architectures for which one can efficiently implement multi-scale algorithms to fight critical slowing down.

^(a): Permanent Address: Dipartimento di Fisica, Università
Vergata, Via E. Carnevale, 00173 Roma, Italy.