Asynchronously Parallelised Percolation on Distributed Machines

N.R. Moloney[1], and G. Pruessner[2]

[1] Blackett Laboratory, Imperial College, Prince Consort Road, London SW7 2BW, United Kingdom

[2] Huxley Building, Imperial College, 180 Queen's Gate, London SW7 2BZ, United Kingdom

Abstract

We describe a novel method for simulating percolation on distributed machines. We build up very large lattices by combining small patches that are delivered asynchronously by slave nodes. These patches are entirely described by the configuration of their borders. Our results include crossing probabilities, the number and sizes of spanning clusters, and cluster size distributions – for aspect ratios covering six orders of magnitude. We relate our work to results in conformal field theory and a recent conjecture regarding the PDF of the order parameter.

Introduction

Percolation remains a very active field of research, both numerically and analytically. Large scale computer simulations are constrained by CPU power and memory. The Hoshen Kopelman algorithm [4] together with Nakanishi label recycling [5] relaxes the latter constraint by requiring memory of order $\mathcal{O}(L^{d-1})$ - rather than $\mathcal{O}(L^d)$ - where L is the system size and d is the dimension. However, histogram data (cluster size distributions, largest cluster distributions, etc.) still require $\mathcal{O}(L^d)$ memory, and large scale simulations often require parallelisation in order to produce results within a reasonable time.

We have implemented a parallelised algorithm which, in principle, **relaxes all constraints on networks, CPU power, and the memory of 'slave' nodes**. We now describe the algorithm for $d = 2$ and present some results.

Description

A lattice is decomposed into squares which are independently realised by slave nodes. Once a node finishes realising its lattice using the Hoshen Kopelman algorithm, it sends a specially prepared version of the lattice border configuration (rather than the whole lattice) to the master node. The master node asynchronously receives lattices and glues them together as patches to form a superlattice:

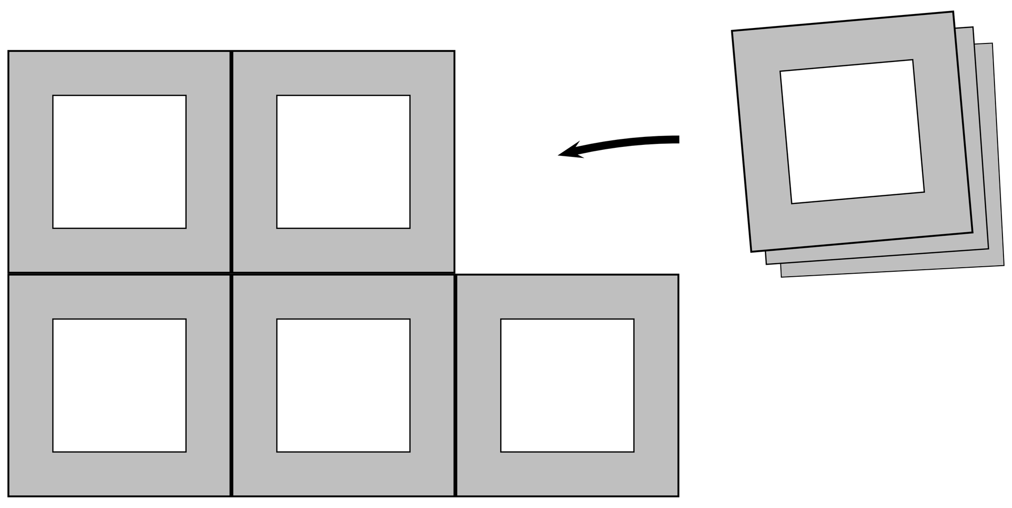


Fig. 1. A superlattice is constructed by combining the borders of individual patch lattices. These patch lattices will generally be rotated, mirrored and permuted.

The master node's responsibilities include the following:

1. Receiving and combining lattice boundaries (see Fig. 1).
2. Accounting for interlattice clusters (see Fig. 2).
3. Combining histograms.
4. Checking for percolating clusters, applying boundary conditions, etc.

The key to this algorithm is the representation of the lattice borders by the slave nodes. We encode all information about size and connectivity explicitly in the border sites. This is essentially a form of Nakanishi label recycling where border sites are considered active and bulk sites not connected to border sites are considered inactive. The slave node scans the border of the lattice it generates, and writes the size of a cluster (as a negative number) in one of the cluster's sites. All other sites in this cluster point to this root site. Clusters in the bulk have their sizes recorded in a histogram.

When two patches are combined by the master it is possible that the common border allows an interlattice cluster to extend between the patches. The master node recognises this by a site-to-site comparison of the border. To produce the correct statistics for interlattice clusters, the original cluster sizes must be deleted from the histogram and replaced by their sum.

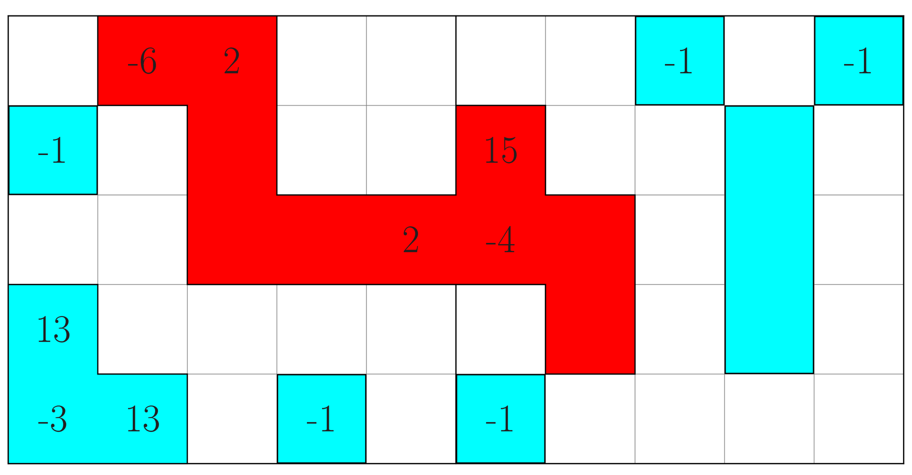


Fig. 2. Two lattices have been glued together to form a superlattice. A cluster in the middle crosses the border separating the original lattices.

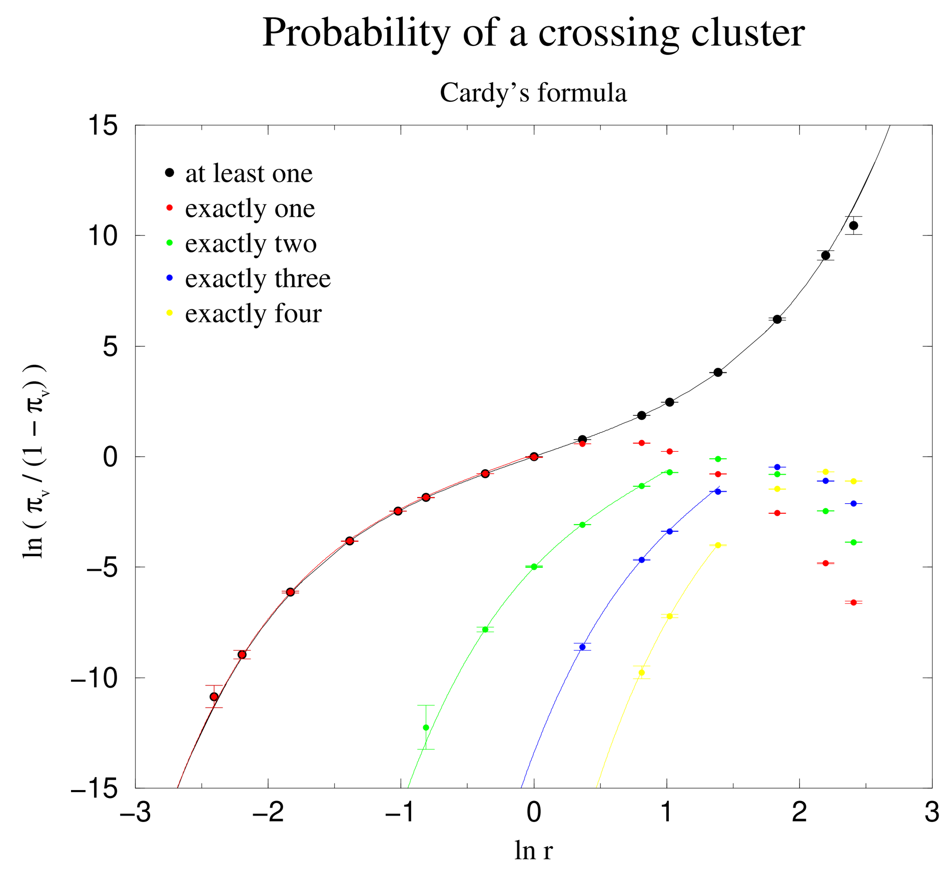
In the above case the master node would therefore perform:

$$\begin{aligned}\texttt{hist}[6] &= -1\\ \texttt{hist}[4] &= -1\\ \texttt{hist}[6+4] &= +1\end{aligned}$$

Since histograms are superimposable, they may be stored locally within the slave node until the entire simulation has finished. Boundary conditions may be applied and percolating clusters identified by scanning the perimeter of the superlattice and checking for common labels. Patches may be combined so as to produce superlattices of different aspect ratios. To improve statistics we can rotate, mirror and permute a set of patches randomly.

Results

We present some preliminary results for superlattices of different aspect ratios with constant area 3000^2.



The black points plot the probability of finding at least one vertical crossing cluster, π_v, for different aspect ratios, where r = width/height. The curved black line is given by Cardy's formula [2],

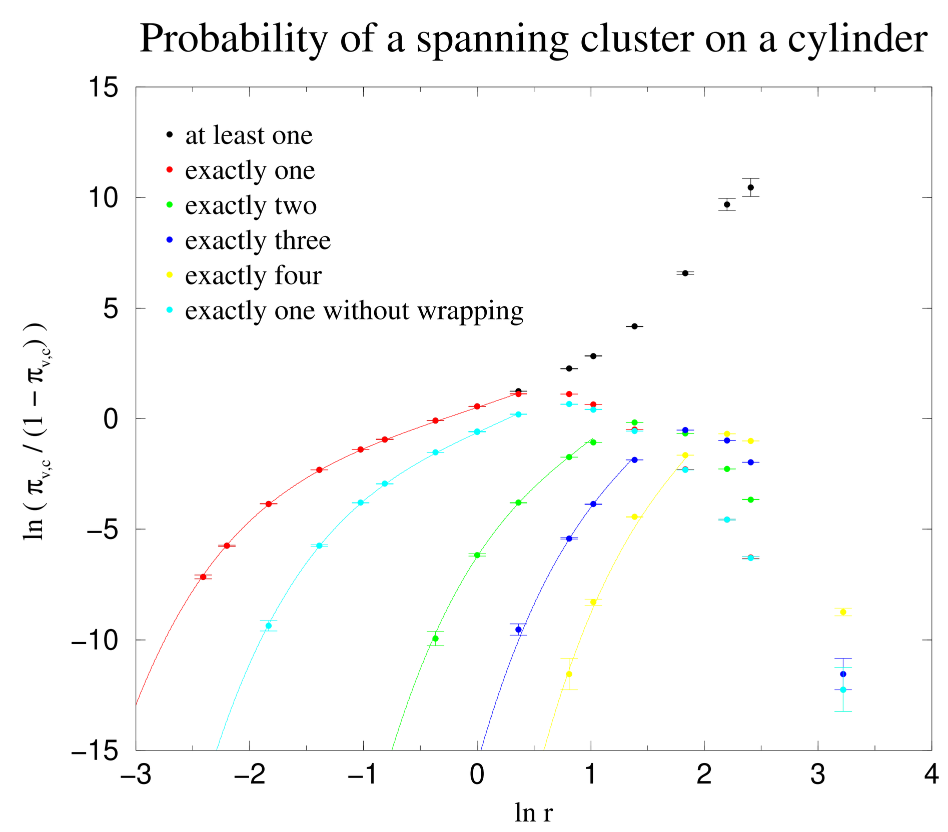
$$\pi_v = \frac{3\Gamma(\frac{2}{3})}{\Gamma(\frac{1}{3})^2}\eta^{\frac{1}{3}}\,{}_2F_1\left(\frac{1}{3},\frac{2}{3};\frac{4}{3};\eta\right)$$

$$\eta = \left(\frac{1-k}{1+k}\right)^2 \quad \text{where } r = \frac{K(1-k^2)}{2K(k^2)}$$

and $K(u)$ is the complete elliptic integral of the first kind. The coloured points plot the probability of finding exactly 1, 2, 3 or 4 vertical crossing clusters for different aspect ratios. The curved coloured lines are conjectured by Cardy [3] to be valid for small r, and are given by,

$$\lim_{L\to\infty} \pi_v(n, \frac{1}{r}) \sim e^{-\frac{2}{3}\pi n(n-\frac{1}{2})\frac{1}{r}}$$

for an exact number n of spanning clusters.



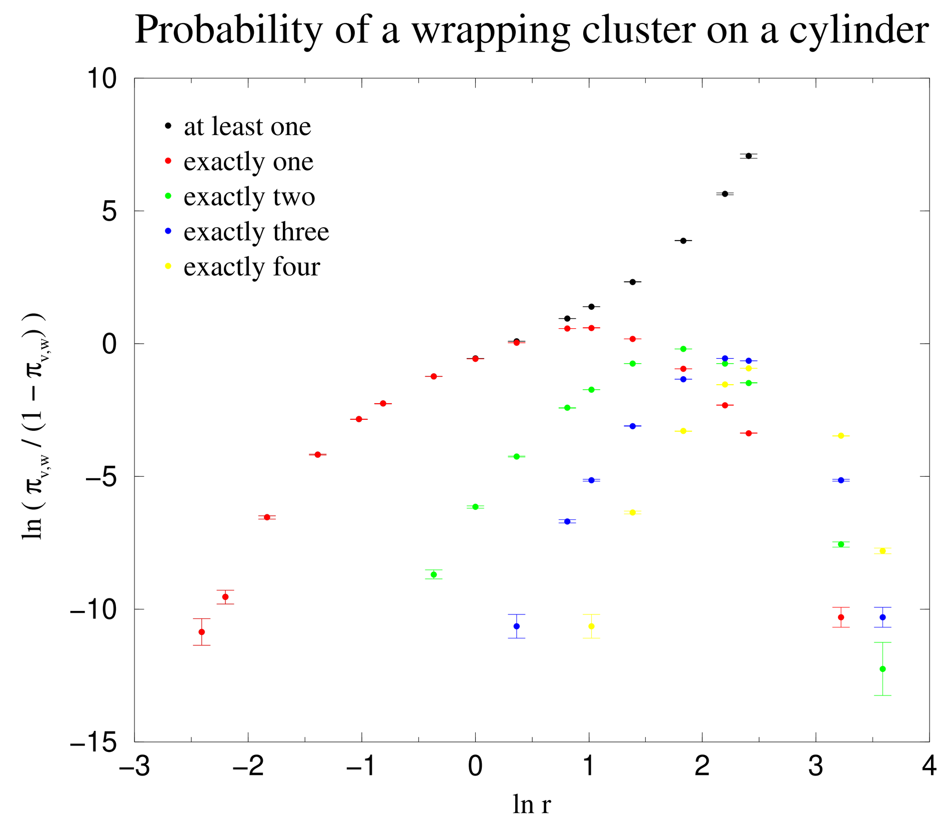
The black points plot the probability of finding at least one vertical crossing cluster on a cylinder, $\pi_{v,c}$, for different aspect ratios, with r defined as above. Periodic boundary conditions are applied horizontally. The coloured points plot the probability of finding exactly 1, 2, 3 or 4 vertical crossing clusters for different aspect ratios. The curved coloured lines are conjectured by Cardy [3] to be valid for small r, and are given by,

$$\lim_{L\to\infty} \pi_{v,c}(n, \frac{1}{r}) \sim \begin{cases} e^{-(\frac{5\pi}{24})\frac{1}{r}}, & \text{for } n = 1;\\ e^{-\frac{2}{3}\pi(n^2-\frac{1}{4})\frac{1}{r}}, & \text{for } n \geq 2.\end{cases}$$

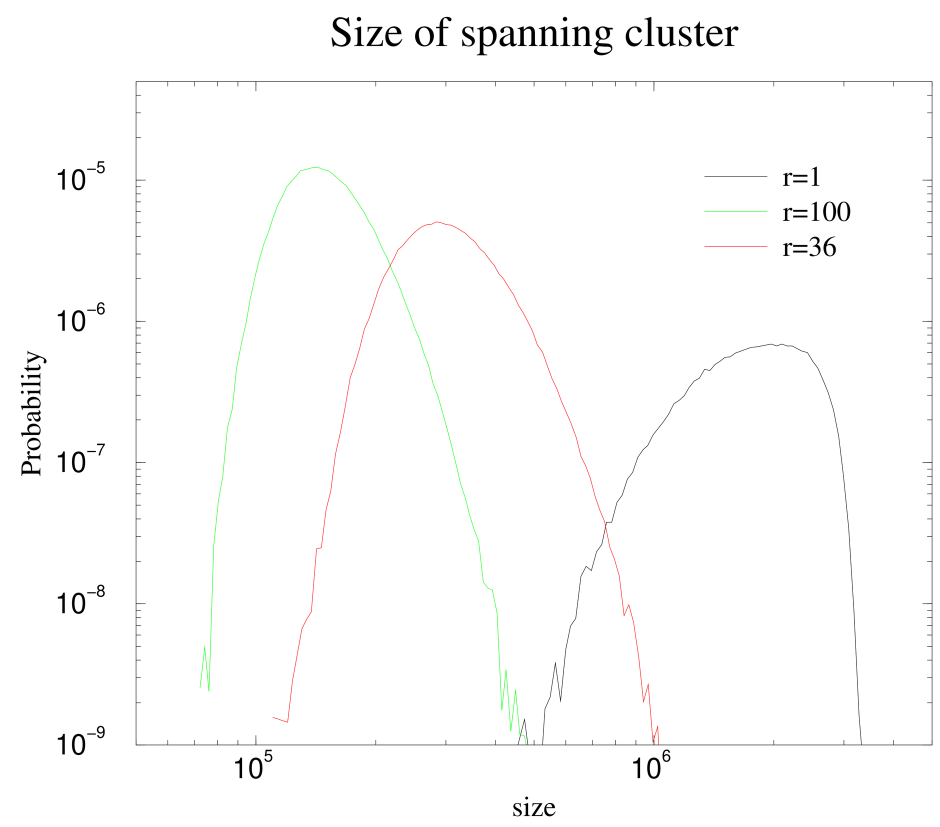
and

$$\lim_{L\to\infty} \pi_{v,c}(n, \frac{1}{r}) \sim e^{-\frac{2}{3}\pi(1-\frac{1}{4})\frac{1}{r}},$$

for having a single spanning cluster without wrapping.



The black points plot the probability of finding at least one vertical wrapping cluster on a cylinder, $\pi_{v,w}$, for different aspect ratios, with r defined as above. Periodic boundary conditions are applied horizontally. The coloured points plot the probability of finding exactly 1, 2, 3 or 4 vertical wrapping clusters for different aspect ratios.



The histograms plot the size of spanning clusters [1] for three different aspect ratios, $r = 1, 36, 100$, with r defined as above.

Discussion

If a master node is too slow in combining patches it can only serve a small number of slave nodes - another master node should share the burden. If a master node is too quick in combining patches the rate of arrival of patches should be increased. There is therefore an optimal master to slave node ratio where the master combines patches as fast as they arrive. If this ratio is unity then each slave has its own master and the whole scheme reduces to a way of recycling a single lattice to produce results for different aspect ratios. We note that a node scales as $\mathcal{O}(L^d)$ in time for producing a patch lattice, while a master node scales as $\mathcal{O}(L^{d-1})$ in time for gluing lattices together. In this way it is possible to tune the above ratio by adjusting the size of the patch lattice.

Conclusion

The method we have implemented relaxes the typical constraints on simulating percolation and even works for slow computers served by a slow network. Systems of enormous size can be simulated within a reasonable time.

The algorithm makes use of the following key features

1. The asynchronous nature of superlattice construction - it does not matter when or whence a patch lattice arrives.
2. Two-dimensional decomposition by using a compact representation of a patch lattice border.
3. The superimposability of histograms across patches.

Although this poster considers site-percolation, the concept is equally applicable to bond-percolation. Our method is particularly well suited for studying spanning and wrapping clusters for different aspect ratios.

The machines we used to produce our preliminary results belong to the undergraduate teaching sections and are entirely idle outside working hours. Hitherto they ran screen savers during the night. They now perform our algorithm instead, and we expect to comfortably beat the world record for the largest percolation simulation, set by a Cray T3E [6] less than a year ago.

References

[1] S.T. Bramwell, K. Christensen, J.-Y. Fortin, P.C.W. Holdsworth, H.J. Jensen, S. Lise, J.M. López, M. Nicodemi, J.-F. Pinton, and M. Sellitto, *Universal fluctuations in correlated systems*, Phys. Rev. Lett. **84** (2000), 17.

[2] J.L. Cardy, *Critical percolation in finite geometries*, J. Phys. A **25** (1992), L201–L206.

[3] ______, *The number of incipient spanning clusters in two-dimensional percolation*, J. Phys. A **31** (1998), L105–L110.

[4] J. Hoshen and R. Kopelman, *Percolation and cluster distribution. I. cluster multiple labeling technique and critical concentration algorithm*, Phys. Rev. B **14** (1976), 3438–3445.

[5] H. Nakanishi and H.E. Stanley, *Scaling studies of percolation phenomena in systems of dimensionality two to seven: Cluster numbers*, Phys. Rev. B **22** (1980), 2466–2488.

[6] D. Tiggemann, *Simulation of percolation on massively-parallel computers*, Int. J. Mod. Phys. C **12** (2001), 871, preprint available from cond-mat/0106354.