Models such as the Ising and Potts systems lend themselves well to simulating the phase transitions that commonly arise in materials science. A particularly interesting variation is when the material being modelled has lattice defects, dislocations or broken bonds and the material experiences a Griffiths phase. The damaged Potts system consists of a set of multi-valued spins on a lattice, where each site is nominally connected to its nearest neighbouring sites, but with some probability of damage that determines whether individual links are present. We simulate the damaged Potts system on large two dimensional (square) and three dimensional (cubic) lattices, using General Purpose Graphical Processing Units (GPGPU) which are well suited to the intrinsic data parallelism of such models. We employ an unusual bit-packing approach to minimise memory usage and optimise cache performance. We report on computational performance, scalability and some physical measurements used in locating the damage-shifted critical properties.

Keywords: Potts model; GPU; bit-packing; phase transition; Griffiths phase

BibTeX reference:

```bibtex
@INPROCEEDINGS{CSTN-135,
  author = {K. A. Hawick and M. G. B. Johnson},
  title = {Bit-Packed Damaged Lattice Potts Model Simulations with CUDA and GPUs},
  booktitle = {Proc. Int. Conf. on Modelling, Simulation and Identification (MSI 2011)},
  year = {2011},
  pages = {371-378},
  address = {Pittsburgh, USA},
  month = {7-9 November},
  publisher = {IASTED},
  doi = {10.2316/P.2011.755-056},
  institution = {Computer Science, Massey University, Albany, Auckland, New Zealand},
  keywords = {Potts model; GPU; bit-packing; phase transition; Griffiths phase},
  owner = {kahawick},
  timestamp = {2011.08.23}
}
```

This is an early preprint of a Technical Note that may have been published elsewhere. Please cite using the information provided. Comments or queries to:

Prof Ken Hawick, Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand.
Complete List available at: http://www.massey.ac.nz/~kahawick/cstn
Bit-Packed Damaged Lattice Potts Model Simulations with CUDA and GPUs

Ken A. Hawick and Mitchell G.B. Johnson
Computer Science, Institute for Information and Mathematical Sciences, Massey University, North Shore 102-904, Auckland, New Zealand
Email: {k.a.hawick, m.johnson}@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181
August 2011

ABSTRACT

Models such as the Ising and Potts systems lend themselves well to simulating the phase transitions that commonly arise in materials science. A particularly interesting variation is when the material being modelled has lattice defects, dislocations or broken bonds and the material experiences a Griffiths phase. The damaged Potts system consists of a set of multi-valued spins on a lattice, where each site is nominally connected to its nearest neighbouring sites, but with some probability of damage that determines whether individual links are present. The damaged Potts system on large two dimensional (square) and three dimensional (cubic) lattices is simulated using General Purpose Graphical Processing Units (GPGPU) which are well suited to the intrinsic data parallelism of such models. An unusual bit-packing approach is employed to minimise memory usage and optimise cache performance. A discussion is given on computational performance, scalability and some physical measurements used in locating the damage-shifted critical properties.

KEY WORDS
Potts; GPU; bit-packing; phase transitions, Griffiths phase.

1 Introduction

New magnetic materials such as manganites [1–4] exhibit exciting properties that are leading to new technological devices such as dense storage media for next generation hard disk drives. Understanding the properties of such materials continues to be a challenge and often computer simulation and modelling is the only practical means of investigation of complex microscopic phenomena. Numerical experiments based on a simulation model can form the basis for making macroscopic measurements which can be compared with measurements made of macroscopic properties of real materials [5, 6].

The Griffiths phase [7–11] is a regime exhibited by random spin systems and there is considerable recent research reporting the presence of this phenomena in real experiments with manganite materials. Comparisons can be made with random magnetic spin models [12] and in particular much work has been done with the diluted or bond-damaged [13] Ising model of a magnet [14] including important theoretical studies [15] and comparisons with disordered dielectric materials [16]. The dilute Ising model has formed the basis for a very large body of research on phase transitions and critical phenomena
Reported research has covered the dynamics of dilute magnets [19, 20] and possible scaling laws [21, 22] explaining the relaxation behaviour [23] as well as residual persistence effects [24].

The Potts model extends the Ising model by allowing an arbitrary number of spin states – the Ising model only supports two – and which has a range of other applications including quantum gravity [25] as well as practical magnetism in materials.

There are a number of useful quantities that can be measured from a simulated Potts or Ising system including the critical exponents that govern how properties such as the magnetization or heat capacity of the system change with temperature [26–29]. Of particular interest to us in this present work is how the critical temperature itself changes as the magnetic system is damaged. Changing the fraction of bonds present is akin to simulating the faults and defects and polycrystalline structure of real magnetic materials. These can be characterised by a damage parameter that specifies how many of the spin-spin interaction bonds are actually present.

Work has been reported in the literature concerning how damage is formed and spreads in an Ising system [30–35] and it is of great interest to study this in diluted Potts models [36, 37] also. Some work has been reported on the Potts model on graphs [38] and chains [39] and on two-dimensional systems [40, 41], but not in damaged arbitrary-Q state and higher dimensional Potts systems.

It is necessary however to be able to simulate large model system sizes with in excess of $10^6$ spins in two dimensions or even $10^9$ spins in three dimensions. As we discuss in this paper, it is also necessary to be able to run simulations for a relatively large number of steps. This is firstly to properly thermally equilibrate systems so that they are truly representative of the simulated anneal temperature. Secondly, to make high quality measurements on simulated models the run-time statistics need to be collected over long time sequences of uncorrelated model sample configurations.

Data parallel computing has long been a readily applicable technique to speed up Ising and Potts model simulations. Various approaches can be used to distribute the spin-updates across independent processing elements in a manner than does not affect the fundamental properties of the model. Graphical Processing Units (GPUs) are a power modern platform implementing the data parallel ideas that used to require dedicated supercomputers in the 1980s and 1990s.

GPUs have been successfully used for simulating Ising and Potts models on regular undamaged lattices [42, 43]. Work is also reported on simulating the Ising model on irregular graph structures such as small world networks [44]. To our knowledge however no group has reported attempts to simulate damaged lattice Potts or Ising systems using GPUs.

To make use of GPUs for work of this nature it is necessary to use compact memory representations [45] of the spin states but also of the bonds that are local to each spin. In this paper we employ a bit-packing approach to simulate the Potts model on various models of GPU using NVidia’s Compute Unified Device Architecture (CUDA) programming language [46].

In Section 2 we describe how the bond-diluted Potts model is formulated and implemented using GPUs and data parallel programming. We present selected performance results for various GPU models in Section 3 and discuss their implications for large scale magnetic materials simulations in Section 4. We offer some conclusions and areas for further work in Section 5.

## 2 Method

The Potts model is a generalisation of the well-known Ising model of a magnet. Each spin takes on one of possible $Q$ values and the spin sites are coupled to their nearest neighbours so that for a ferromagnetic coupling, it is energetically favourable for the spin values to align. A numerical experiment typically involves initialising the spin system completely randomly then applying a thermal annealing algorithm to drive the model system to an equilibrium for the particular annealing temperature. For systems like the Ising and Potts models there is a critical temperature $T_c$, above which the system remains random with no magnetic moment, but below which the system has aligned clusters of spins and exhibits a definite magnetic moment. These spin models can be simulated as described below but to accurately determine their properties requires simulation of quite large model systems and quite long simulation times are required to ensure the models are properly equilibrated and genuinely representative of the anneal temperature, before measurements are made on them.

In the work we report here, we apply an additional complication – that the lattice of couplings between spins is damaged - so that some bonds are missing. This adds to the computational work in carrying out numerical experiments that must scan in temperature as well as damage probability. Fortunately data parallel computing methods are accessible by the use of Graphical Processing Units (GPUs) to speed up the simulations.

We set up our $d$-dimensional model on a periodic mesh of $N = L^d$ spins. For simplicity in the work reported here we use the same lattice length $L$ in all dimensions $(x, y, z, ... )$. The Hamiltonian (in dimensionless units) of
and takes values $M \in [0, 1]$. Specifying $M$ for an arbitrary number $Q$ of states is somewhat less intuitive to define, but a commonly used definition is:

$$M = \frac{1}{N} \left( \frac{Q \cdot M_{\text{max}} - N}{Q - 1} \right)$$

(4)

where $M_{\text{max}}$ is the number of spins that have the most populous value of $Q$. This again gives a good measure of how well aligned the system is, with value $M \equiv 1$ for a fully ordered system with all spins having the same value, and $M \equiv 0$ for a system with equal numbers of all $Q$ states.

A useful approach for calculating the critical temperature of a simulated Potts system uses the fourth-order Binder cumulant method [47, 48]. This cumulant is defined at different network sizes $N$ as:

$$U_N = 1 - \frac{(M^4)_N}{3(M^2)_N}$$

(5)

and when plotted against temperature $T$, forms an $S$-shaped curve that becomes sharper with larger simulated system sizes.

The cumulant curves plotted for different $N$ values coincide and intersect at the critical temperature and thus provides us a means of extrapolating simulated behaviour of finite systems to the thermodynamic limit. It is useful to simulate three different system sizes, and by homing in on the region of intersection, simple least-squares fitting yields not only a critical Temperature value, but also an estimate of the uncertainty in it.

### 2.1 Bit-Packing

Simulating a large $q$-state Potts system requires a large amount of memory. Using a naive method of storing the spin value in a single integer for a $8192^2$ sized 2D system the memory used will be 262 megabytes excluding memory needed for random number generation. This is exacerbated in a damaged lattice system where each link must be tracked, to accomplish this we use an unsigned short (that occupies two bytes) for each site to store the spin value in a single integer for a $8192^2$ sized 2D system point.

The dynamics of the model is then to select spin sites and attempt to flip their spin to a different value and accept the change probabilistically using the Boltzmann distribution weighting factor $\exp(-\Delta \mathcal{H})$ determined from the change in energy Hamiltonian that would potentially occur from the flip.

The magnetisation $M$ of the whole system is a measure of how aligned the spins are. The magnetisation of the $Q = 2$ state Ising model is:

$$M = \frac{1}{N} \sum_{i=0}^{N-1} \sigma_i$$

(3)
There are three types of memory available to the program-
er using CUDA, the first is Global Memory which is the most abundant memory available ranging from 800MByte on the GTX260 to 6GByte on the Tesla C2070 devices. This memory is the only type accessible by the CPU host and is also the slowest, although through careful use of coalescing techniques it is possible to load 32, 64 or 128 reads at a time. Excessive direct use of global memory imposes a harsh penalty on the programmer if non-sequential reads and writes are performed in the kernel as is common in the Potts model. Shared Memory is accessible from all threads within a block and can be used efficiently to share information with neighboring threads reducing global read/writes. Texture Memory is a cached area of global memory and is highly efficient at accessing spatially close data in 1,2 or 3D. Constant Memory is another cached piece of global memory but can be accessed by multiple threads at one time because it is guaranteed to be constant thus avoiding any race conditions and bank conflicts. We utilize Global memory to store the random numbers and the lattice site information that includes the spin and neighbor links, constant memory is used to store the lookup table for the flip probabilities which is computed on the host according to the temperature parameter and is stored in a look-up table.

Shown in Table 2 are the global Memory sizes of modern GPUs that we have used to test our simulation on. They provide enough memory that we can store large systems efficiently on device memory instead of swapping sections in and out from host memory as may be required for simulating really large systems in many dimensions.

The Random number generator used on this simulation is the Ran2 generator presented in [49]. This has been proven to be a reliable source of random numbers and has passed all of the “die hard” tests for random numbers [50]. We assign an integer in global memory for each random number instance for each thread to access based on its thread ID.

### 2.3 Monte Carlo Update

Fixed dimension simulations of Ising and Potts systems are relatively easy to visualize and implement. In CUDA the most common method of storing an array of spins is to condense the multi-dimensional array into a more GPU friendly 1D and accessed using the formula \( k = y \times \text{width} + x \) for 2D and \( k = z \times \text{width} \times \text{height} + y \times \text{height} + x \) for 3D. This is still the best method for variable d-dimensions but is complicated by the unknown number of dimensions to iterate through. We use bit logic methods to perform \( k\)-indexing [45] to address sites in the lattice stored in a 1D array as shown in Algorithm 2.

We implement a classical Metropolis update method known as Red- Black checker board update [49] which divides the lattice up into two independent sets of red and

**Algorithm 1** Pseudo-code for the allocation of bits for an \( D \) dimensions and \( Q \) states Potts model. OR and SHIFT are bitwise operations

```plaintext
declare Q_MASK ← \((Q + (Q \mod 2))/2\)
for \( k ← 0 \) to \( N \) do
    site[k] ← site[k] OR Q_MASK
for \( d ← 0 \) to \( D * 2 \) do
    declare random ← RandomInteger(0 - 1)
    site[k] ← site[k] OR (random SHIFT LEFT
    (d + Q_MASK))
end for
end for
```
**Algorithm 2** Pseudo-code for decomposing a $D$ dimensional index $\text{index} [x][y]...$ into a single $k$ index where the length for each dimension $d$ is denoted by $L[d]$

```
declare $k$ index
for $d \leftarrow 0$ to $D$
  $k$ index $\leftarrow k$ index + ($\text{index} [d] \times L[d]$)
end for
```

Figure 4: The arrangement of bits for a single site of an 8 state, 3 dimensional Potts model with 1 broken neighbor link.

black with $N/2$ sites each. We then iterate through each colour updating every site in the colour set. This avoids race conditions and sweep artifacts by ensuring that all neighbors for any cell in the red set are located in the black set. This algorithm is much better suited to the SIMT architecture of GPUs than the common serial implementation of picking random sites and updating them because there is no overhead of generating random positions and we also avoid any conflicts with race conditions trying to update the same site by assigning each thread a unique one. Although the Red-Black update is very efficient, the global memory access constraint of loading 32, 64 or 128 bytes at a time is not well suited to the checker board pattern and wastes a lot of time loading unneeded memory. We solve this by crinkling [45] the data as shown in Figure 4. This crinkling is done independently of the update kernel and is performed once at initialization and again at the end of the simulation. This crinkling method allows all of the bytes of the memory load to be utilized for the current update, thus reducing the total number of memory loads.

### 3 Results

Figure 5 shows how the critical temperature varies when parameter $p$ is lowered to introduce damage into the $d = 2, Q = 2$ model (Ising case) in two dimensions. These data are preliminary, being based upon small system sized runs. The method described was applied using three different sized model simulations and recording the intercept of the three independent Binder cumulant plots. As can be seen, introducing damage lowers the critical temperature almost linearly. Near the bond-percolation threshold of $p = 0.5$ the system is no longer represented by a single connected cluster of spin sites and the phase transitional behaviour disappears. This is similar to the Griffiths phase behaviour whereby the magnetic properties of real materials no longer adhere to Curie’s law and there is effectively a transition in the transition. We plan to investigate this phenomena in greater detail for $d = 2, 3, 4; Q = 2, 3, 4, 5,...$ and also perhaps for different geometric lattices such as face-centred cubic and body-centred cubic.

Figure 6 plots the log of the system size vs the log of the total time in two - dimensions. As the System size
Table 2: Table showing NVidia CUDA capable GPU’s that were used for testing with results shown in Section 3. We restrict the average spin-flip range to $L = 1024 - 8192$ to ensure all GPU cores are being fully utilized. Note - The GTX260 used in these tests is the black edition that has 216 CUDA cores and is no longer listed on the NVidia web site.

<table>
<thead>
<tr>
<th>GPU</th>
<th>CUDA cores</th>
<th>Memory Size (GB)</th>
<th>Spin-Flip Time (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTX 580</td>
<td>512</td>
<td>1.5</td>
<td>0.334 +- 0.001</td>
</tr>
<tr>
<td>GTX 480</td>
<td>480</td>
<td>1.5</td>
<td>0.395 +- 0.001</td>
</tr>
<tr>
<td>GTX 260</td>
<td>216</td>
<td>0.8</td>
<td>1.174 +- 0.002</td>
</tr>
<tr>
<td>GTX 295</td>
<td>240</td>
<td>0.8</td>
<td>1.654 +- 0.002</td>
</tr>
<tr>
<td>Tesla C2070</td>
<td>448</td>
<td>6</td>
<td>0.541 +- 0.002</td>
</tr>
<tr>
<td>Tesla C2050</td>
<td>448</td>
<td>3</td>
<td>0.542 +- 0.002</td>
</tr>
</tbody>
</table>

Figure 7: Graph of $L$ vs Time taken per spin flip in 10$^{-9}$ seconds (nano seconds) in two - dimensions when $Q = 3$ and $D = 2$ and no broken bonds. Error bars are shown but are smaller than the symbol sizes.

increases the execution time increases exponentially according to $N^2$. The curve in the lower end of the graph are due to the maximum number of cores on the GPU not being fully saturated which is then shown by the flattening out beginning in the range (log $L$) 2.5 in the GTX200 series GPUs and (log $L$) for the GTX400, GTX500 and Tesla series. The times for GTX 260 and GTX295 (using only a single GPU) are the same even though the 295 has 240 CUDA Cores, all the other cards are the GF100 (Fermi) chip-set and have similar performance with a predictable descending order from the newer 580 through to the Tesla cards.

Figure 7 plots $L$ vs Time per spin flip in nano seconds for two dimensions. As in Figure 6 the curve on the lower $L$ values indicate the maximum number of cores are not being utilized until approximately $L = 1024$ where the times flatten out and are consistent for all higher length scales. Again we see the GTX200 series GPUs performed worse than the other cards, up to a factor of two vs the 580. We average the spin-flip time for each GPU in Table 2.

4 Discussion

A Highly Parallel N-dimensional, Q-state damaged Potts simulation using CUDA and NVidia GPGPU processors has been implemented. The speed of GPUs versus serial CPU implementations has been extensively discussed for both Ising [44, 45] and Potts models in [43] thus we have only compared GPU implementations. Direct comparison of our N-dimensional, damaged kernel execution times to others such as [43] is difficult because the overhead of checking each neighbor link and computing k-index for the arbitrary dimensions is significant. However we have found that our fastest average spin-flip time is on the GTX580 with 0.334 nano seconds and the second fastest is on the GTX480 with 0.395ns this comparable to the range of times presented in [43] at 0.54ns - 0.147ns. We have found our code to yield an almost identical speed per spin-flip independent of the number of dimensions, lattice length, number $Q$ of states and any number of broken bonds.

The time taken function for a full update of the Potts system is approximately linear in the number of spins providing $N$ is big enough and shows that the new architecture of the the GTX480, GTX580 and the Tesla cards produce a significant improvement over the older cards. This is attributable to the new memory formats including a L1 and L2 cache, larger multiprocessors and more cores. The GTX260 and the GTX 290 have the same performance and this is due to the higher clock speed of the 260 compensating for its smaller number of processors. The difference between the two generations of GPU are not as significant as has been seen in different types of simulation and we suspect that this is due to the highly efficient memory management with crinkling of the checkerboard update pattern. Our data for spin-flip times show little variation once the system size is big enough so that we are fully utilizing all cores. This is in contrast to the results presented in [43] however this may be due to the small size of systems tested.
In Figure 5 we have shown some preliminary results for the critical temperature shift as the percentage of broken links are increased, this is out of the scope of this paper but will be considered for future work. It is nevertheless representative of the quantitative results that can be obtained from this sort of simulation work.

The random number method we have implemented is memory intensive and could be replaced by another method such as the Quantis random number generator described in [50]. It is important for this sort of work that the generated deviates are of good quality as well as fast. Otherwise, they could bias the Monte Carlo probabilities and give an incorrect value for the phase transitional temperature shift.

5 Conclusion

A description has been presented showing how the damaged Ising and Potts models can be simulated at high performance using a bit packing strategy on various models of Graphical processing Unit(GPU) using bit packing. The bit packing allows us to make good use of the cache available on modern GPUs and thus keep as much of the simulation data “in cache” as possible. This speeds up further the gains made by using data-parallel computing techniques such as crinkle-mapping to keep the GPU cores as busy as possible.

The simulation model is shown to be correct and have discussed some of the implications for simulating models of this type. We have presented some preliminary data on the shifts in critical temperature and anticipate being able to systematically investigate the damaged Potts model - and other models of its class, over a range of dimensions, numbers of spin states and lattice geometries using our custom simulation code apparatus.

There appears to be interesting scope to further study the transitional behavior and phenomena like the Griffiths phase in simulated models. An area for further study is to employ different dynamical schemes in addition to the Metropolis dynamics we have employed here. Glauber and cluster dynamics are also possible and there is scope to further investigate the changes in the correlation times and temporal scaling exponents using this approach.

References

[17] Prudnikov, V.V., Prudnikov, P.V., Vakilov, A.N., Krinitsyn, A.S.: Computer simulation of the criti-


[27] Zinn-Justin, J.: Analysis of ising model critical exponents from high temperature series expansion. Le Journal de Physique 40 (1979) 969–975


