Two case studies of Monte Carlo simulation on GPU

Junqi Yin

Material Theory Group
Oak Ridge National Lab

Titan Summit, 2011

Outline

Monte Carlo simulation

What is Monte Carlo simulation?

- A class of methods to solve problems by repeated random sampling

Random number generator on GPU

- Linear congruential RNG
- CURAND library
- Mersenne Twister with dynamically generated parameters

Applications of MC

- Physics(statistical sampling)
- Mathematics(numerical integrations)
- Finance(option pricing)
- . . .

Desktop GPUs



	GTX 285	Tesla C1060	Tesla C2050
Processor elements	240	240	448
Peak performance (GFLOPS single)	1062	933	1030
Peak performance (GFLOPS double)	88.5	77	515
Memory Bandwidth (GB/s)	159	102	144
Memory size (GB)	1	4	3
Cost	$330	$1300	$2500

•*2008-2009* •*2010*

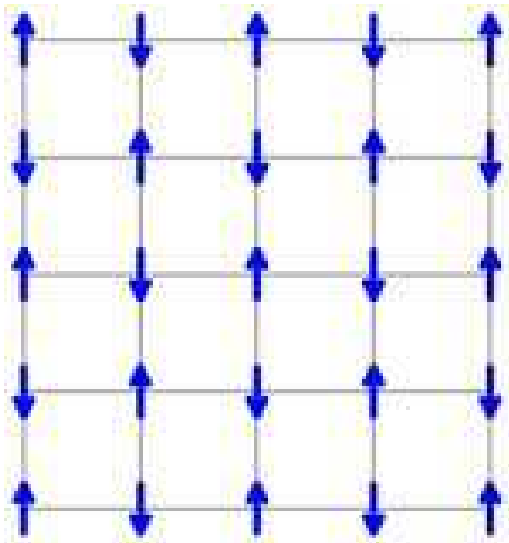
Discrete energy lattice model

Ising Model

$$E = J \sum_{<i,j>} \sigma_i \sigma_j + H \sum_i \sigma_i \qquad \sigma_i = \pm 1$$

fruit fly for statistical physics

- describe magnetic phase transition
- study critical phenomena
- ...



Ising square lattice

Parallelism in space

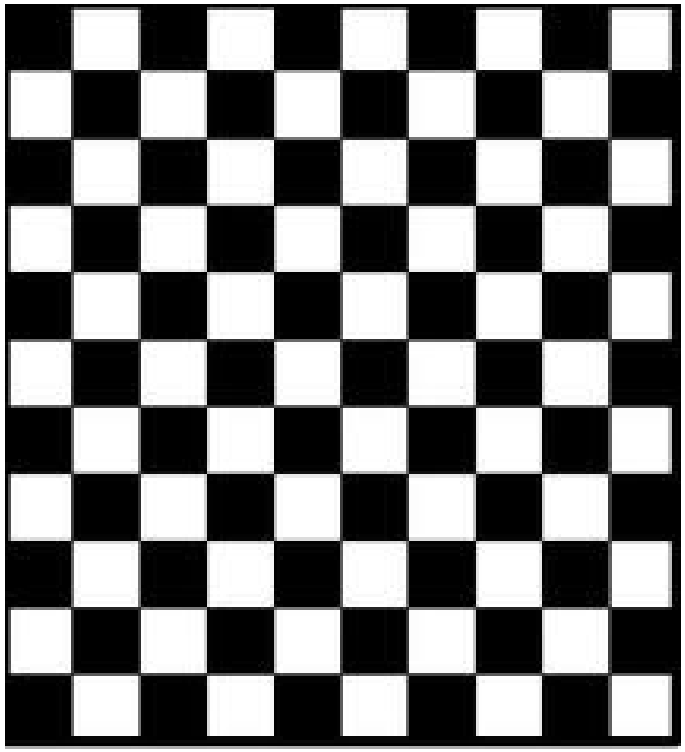
Generate new configuration: Metropolis single spin flip

$$W(\sigma \leftrightarrow -\sigma) = min[1, exp(-\frac{\delta E}{k_B T})]$$

checkerboard update

- step 1: update all spins in sublattice black
- step 2: update all spins in sublattice white

If the next-nearest neighbor interaction is included, 4 sublattices are needed.



checkerboard decomposition

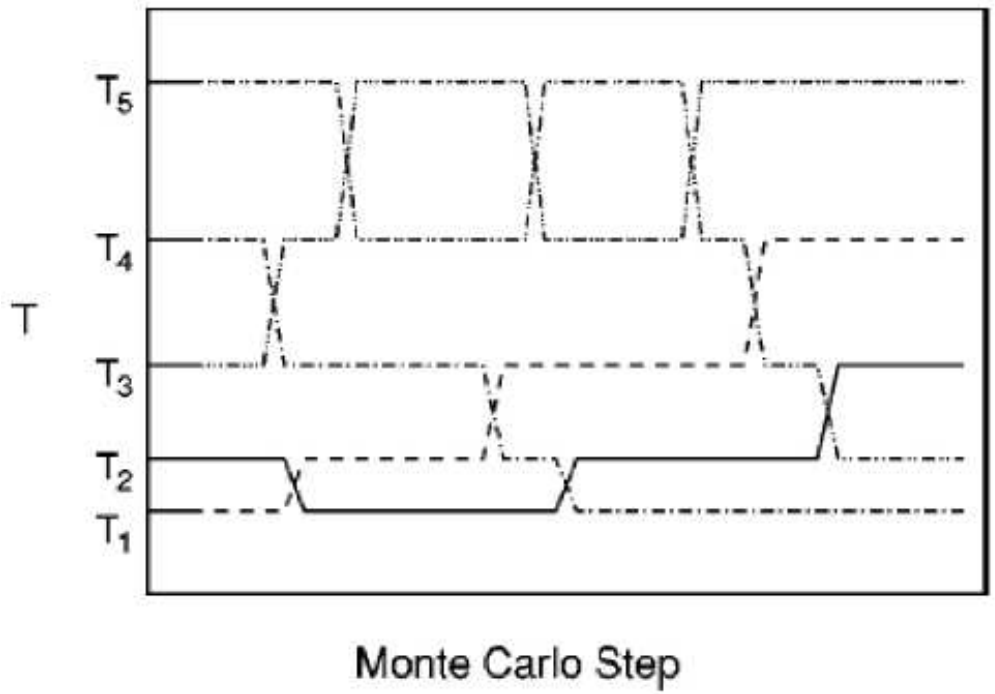
Parallelism in algorithm

Parallel Tempering (replica exchange)

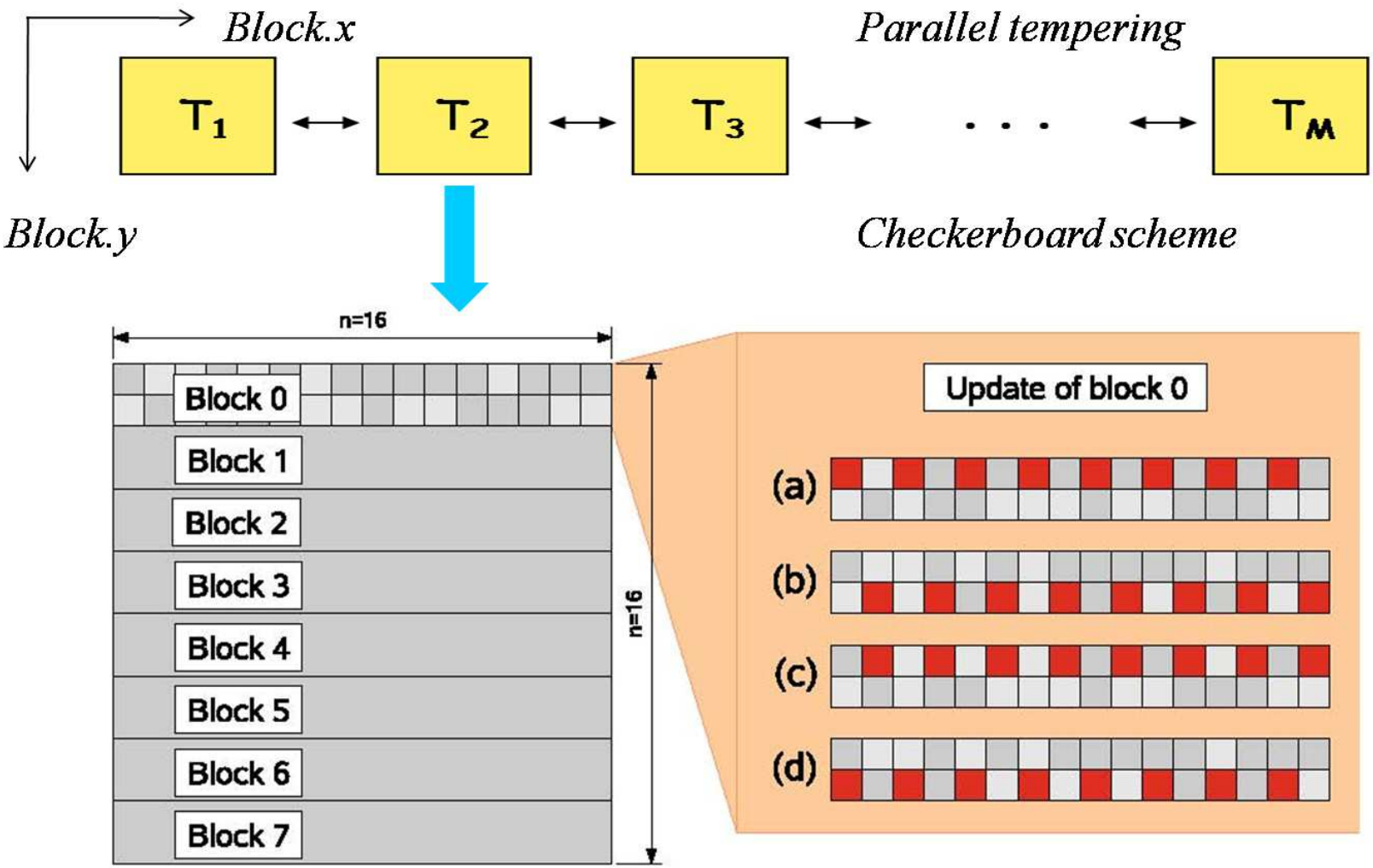
$$W(\beta_m \leftrightarrow \beta_n) = min[1, exp(-(\beta_n - \beta_m)(E_m - E_n))]$$

Advantage

- Overcome energy barriers
- Fast approach to equilibrium
- More independent samples



Overall implementation



Code

Main function

```
 dim3 grid(nT,L/2);
 dim3 block(L/2);
 for(MC steps) {
// Monte Carlo Sweeps
     Kernel<<<grid,block>>>(d_Spin,d_random,1);
     Kernel<<<grid,block>>>(d_Spin,d_random,2);
     Kernel<<<grid,block>>>(d_Spin,d_random,3);
     Kernel<<<grid,block>>>(d_Spin,d_random,4);

     ......
}
```

Code

Kernel function

```
 __global__ void Kernel(int* Spin, int* State, int sublattice) {
 int tid = threadIdx.x, bidx = blockIdx.x, bidy = blockIdx.y;
 __shared__ int r[2*L/2];
// Load state for random number generator
 r[tid] = State[tid + L/2*bidy + L/2*L/2*bidx];
 r[tid+L/2] = State[tid + L/2*bidy + L/2*L/2*bidx + nT*L/2*L/2];
switch(sublattice) {
    case(1):
    site = 2*tid + bidx*L + 2*L*nT*bidy;
    ......
    case(2):
    site = 2*tid + bidx*L + 2*L*nT*bidy + 1;
    ......
    case(3):
    site = 2*tid + bidx*L + 2*L*nT*bidy + 1 + L*nT;
    ......
    case(4):
    site = 2*tid + bidx*L + 2*L*nT*bidy + L*nT;
    ......
}
```

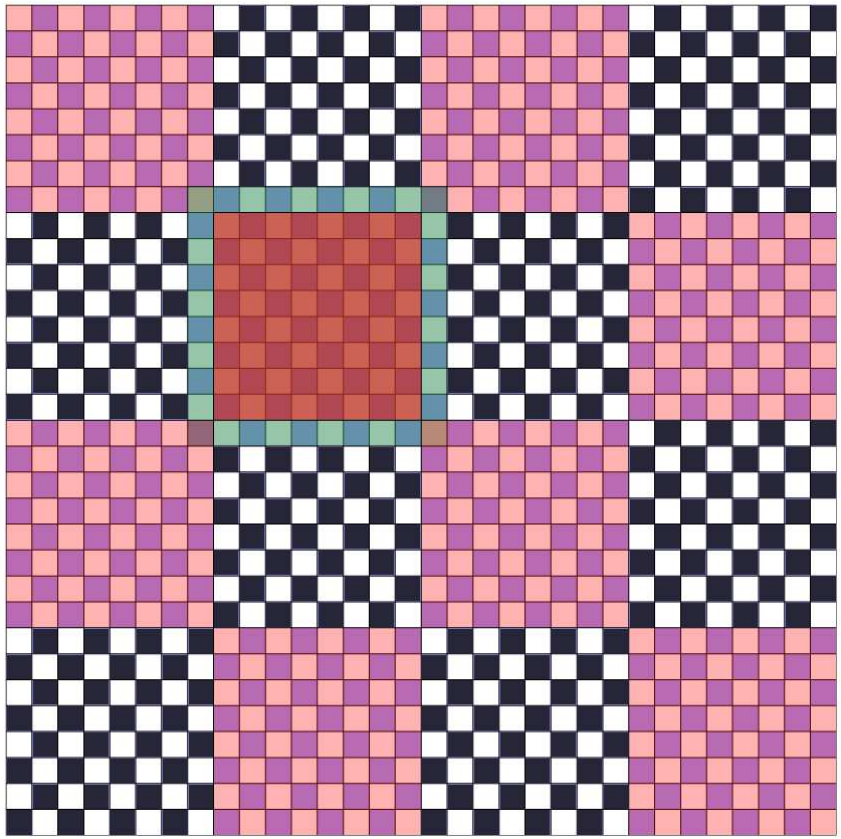
Optimization

Optimize memory transfers

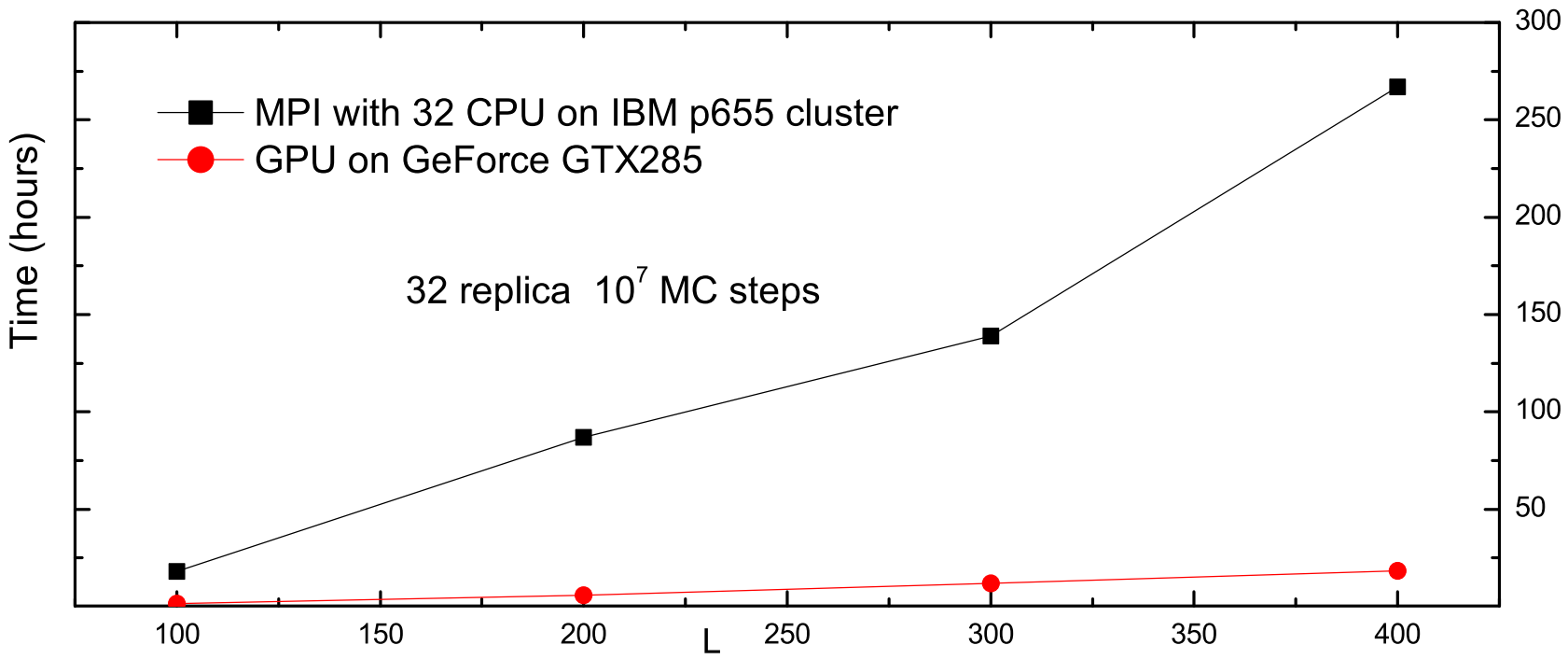
- measure E, M, etc on device
- swap $\{\sigma\}_T$ on device
- array($\{\sigma\}$) for coalesced memory access
- lookup table on constant memory
- use shared memory as a cache

Maximize arithmetic intensity

- more computation per memory access(28 SPFP per load for Fermi)



Performance



J. Yin and D. P. Landau, Phys. Rev. E**80** 051117 (2009)

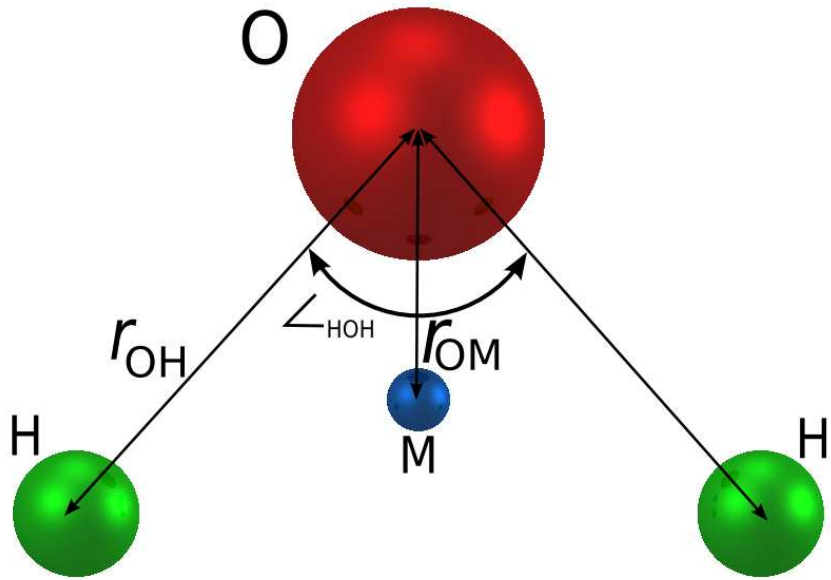
Continuous energy off-lattice model

Water Model

$$\epsilon_{mn} = \sum_{i}^{m} \sum_{j}^{n} \frac{q_i q_j e^2}{r_{ij}} + \frac{A}{r_{OO}^{12}} - \frac{C}{r_{OO}^{6}}$$

r_{OH}(Å)	0.9572
$\angle HOH$(deg)	104.52
r_{OM}(Å)	0.15
q_H	0.52
q_M	-1.04
$A \times 10^{-3}$(kcal Å^{12}/mol)	600
C(kcal Å^{12}/mol)	610

parameter for water model TIP4P



General ensemble sampling

Metropolis Method

$$Z = \int e^{-E[\mathbf{x}]/k_{\mathrm{B}}T} d\mathbf{x}$$

Wand-Landau Method

$$Z = \int g(E) e^{-E/k_{\mathrm{B}}T} dE$$

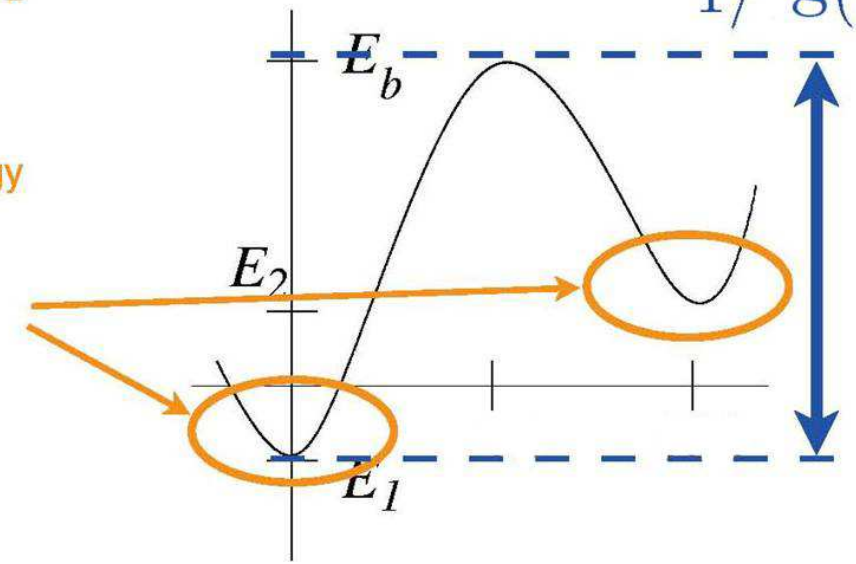
Sample configuration space with probability

$e^{-E[\mathbf{x}]/k_{\mathrm{B}}T}$

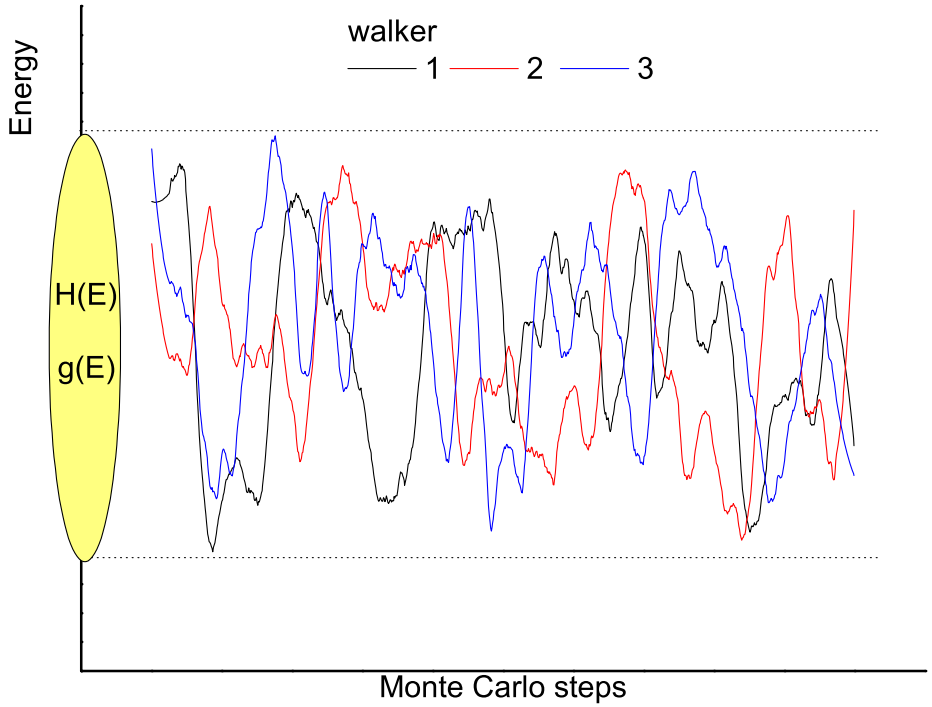
Samples mainly regions around energy minima

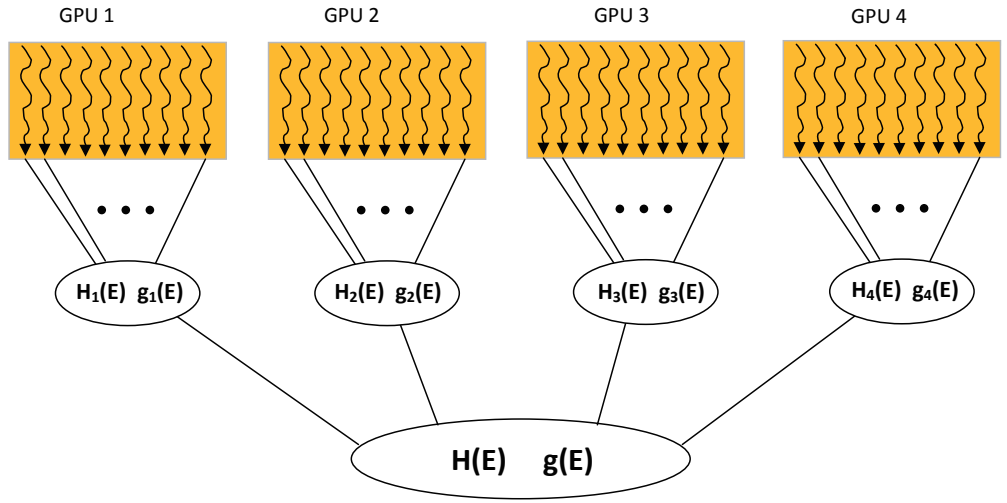
$1/\ g(E[\mathbf{x}])$

Samples all energies equally -



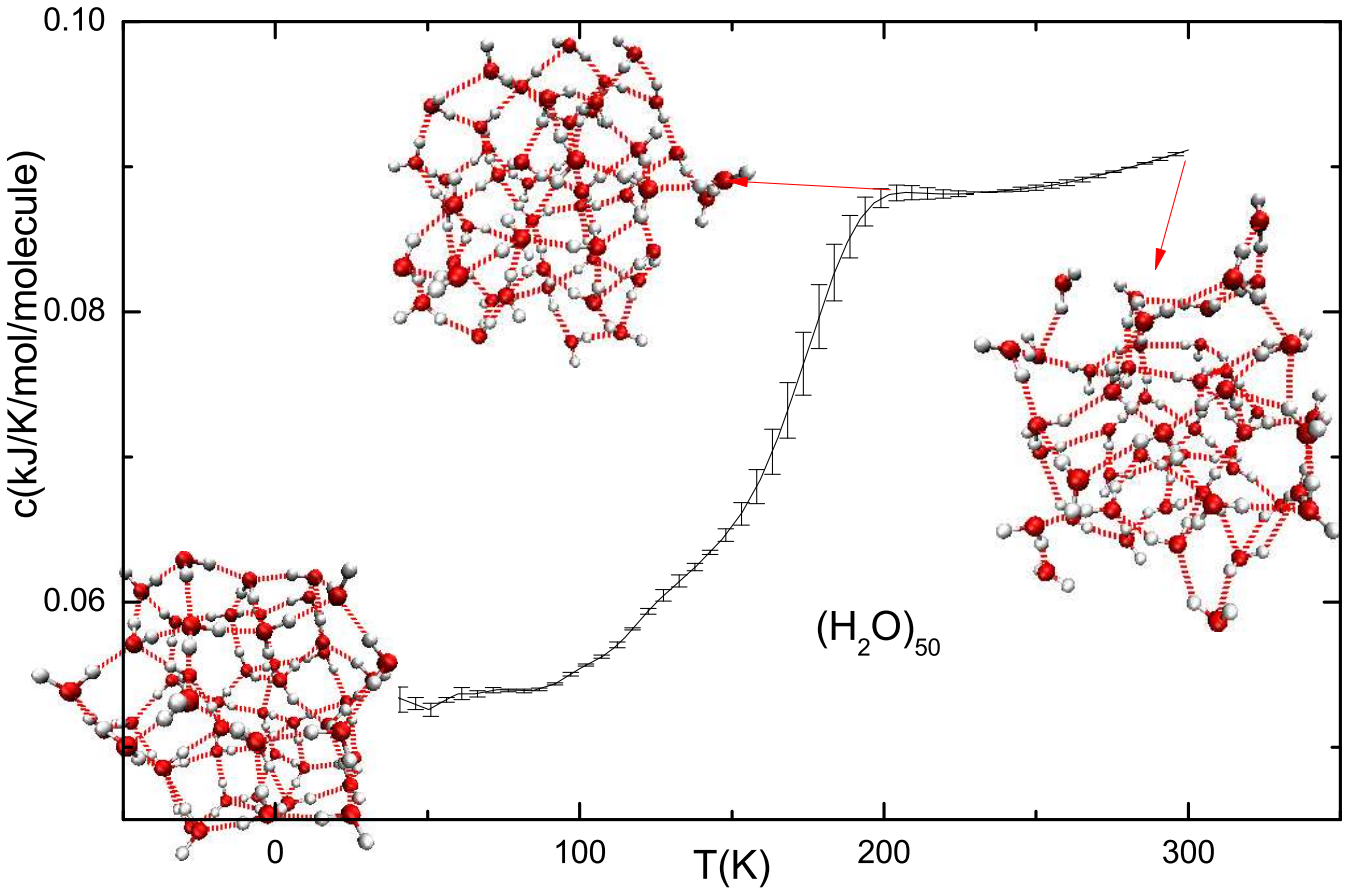
Implementation





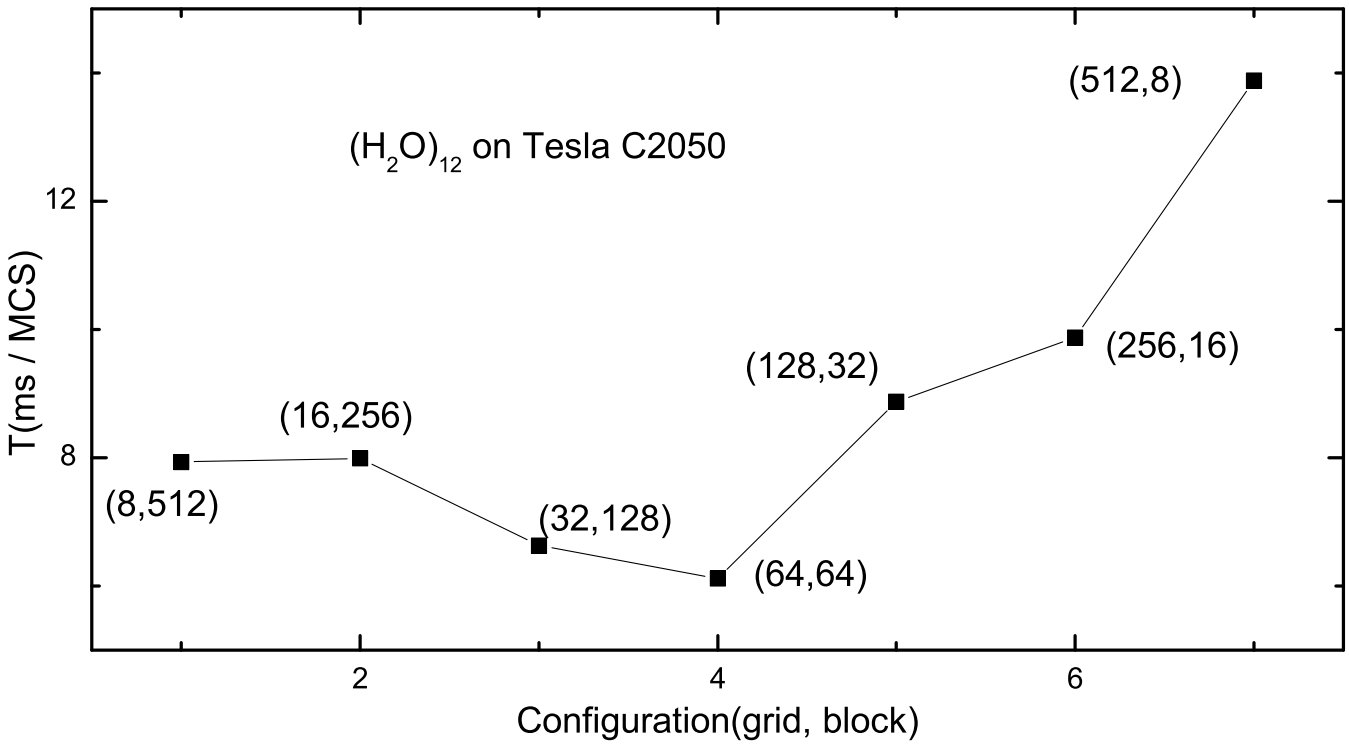
- "gather" instead of "scatter"

Results



For more results on water models, refer to J. Yin and D.P. Landau, J. Chem. Phys 134, 074501(2011)

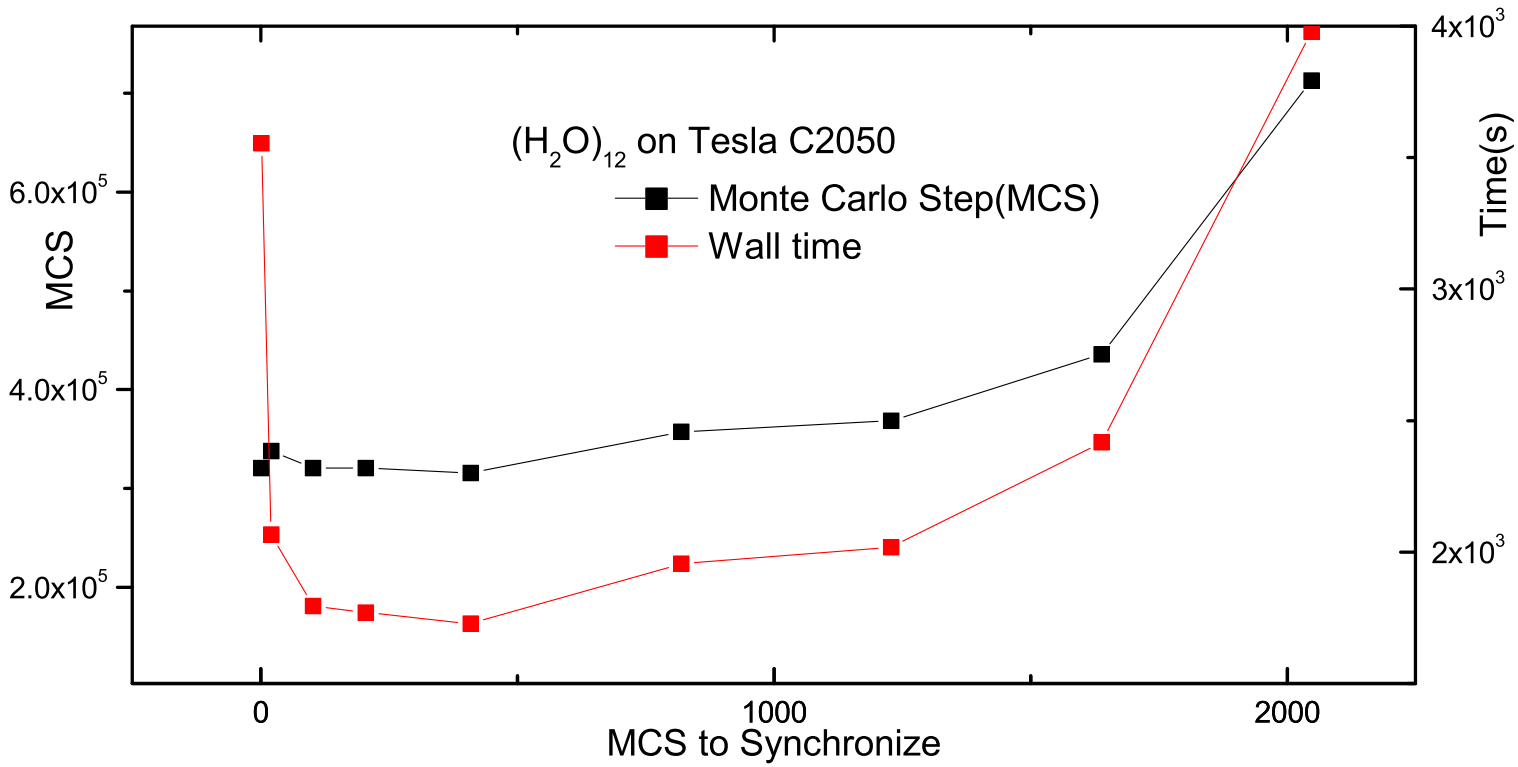
Tuning



Maximize processor occupancy

- Optimize execution configuration (*Occupancy calculator*)

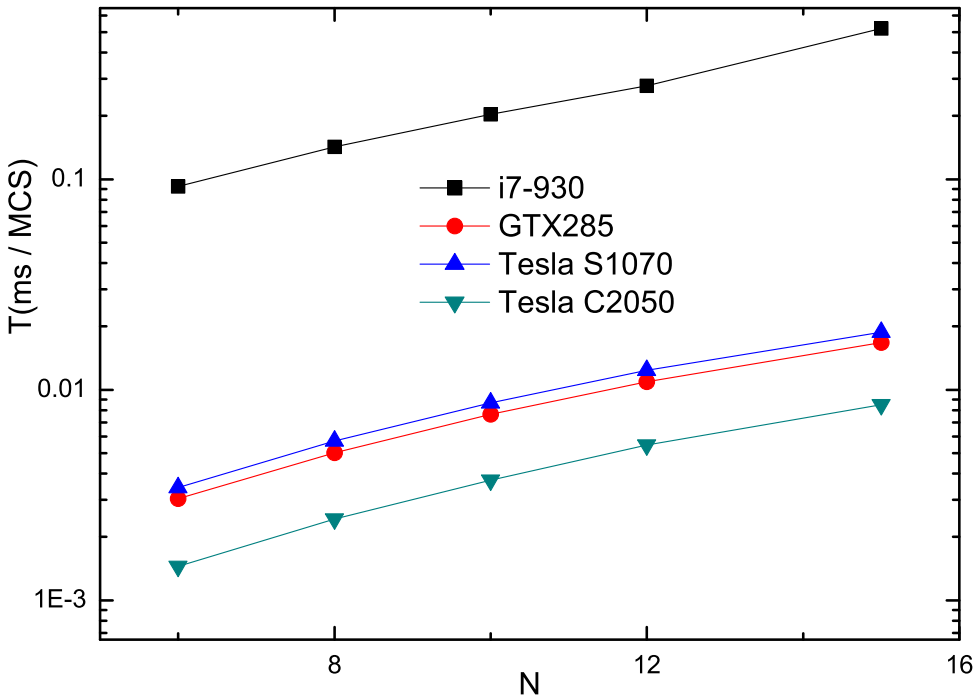
Tuning

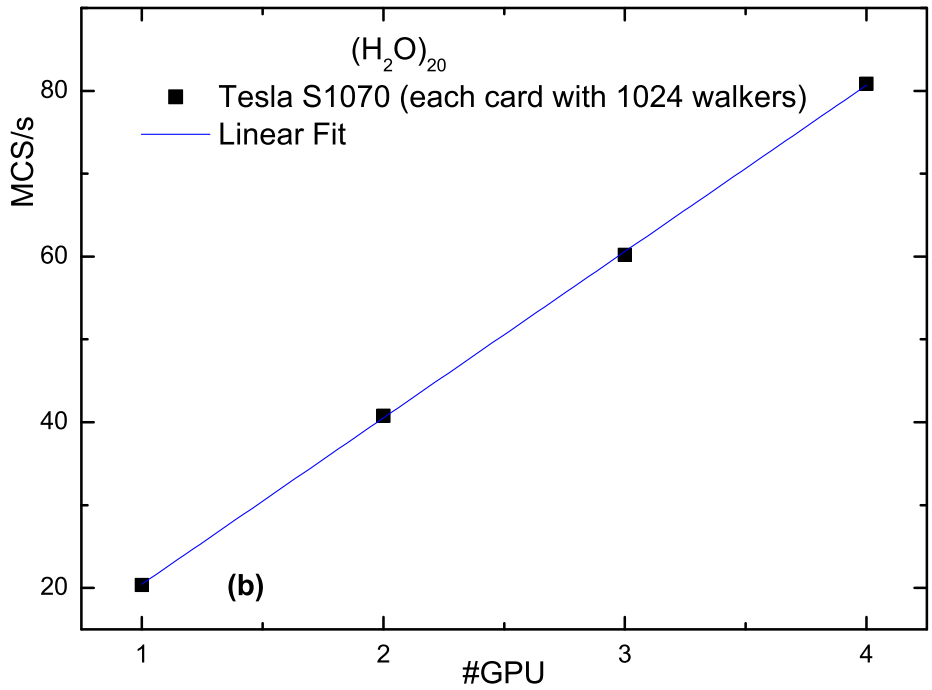


Balance

- Tradeoff between sync overhead and convergence

Performance





Summary

Ising models

- The implementation can be applied to variant lattice spin models, such as Heisenberg model, spin glass, etc.
- $50\times \sim 100\times$ speedup can be expected.

Water models

- Algorithms involving many replicas can straightforwardly be extended to multi-GPUs.
- Wang-Landau method is a good candidate for sampling complex systems on GPU.