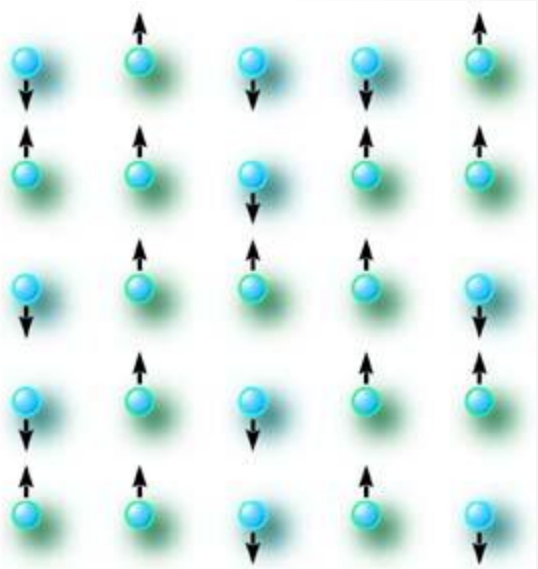
Monte Carlo simulation of Ising Model with Parallel Tempering

Cade Thomasson
Louisiana State University

Ising Model

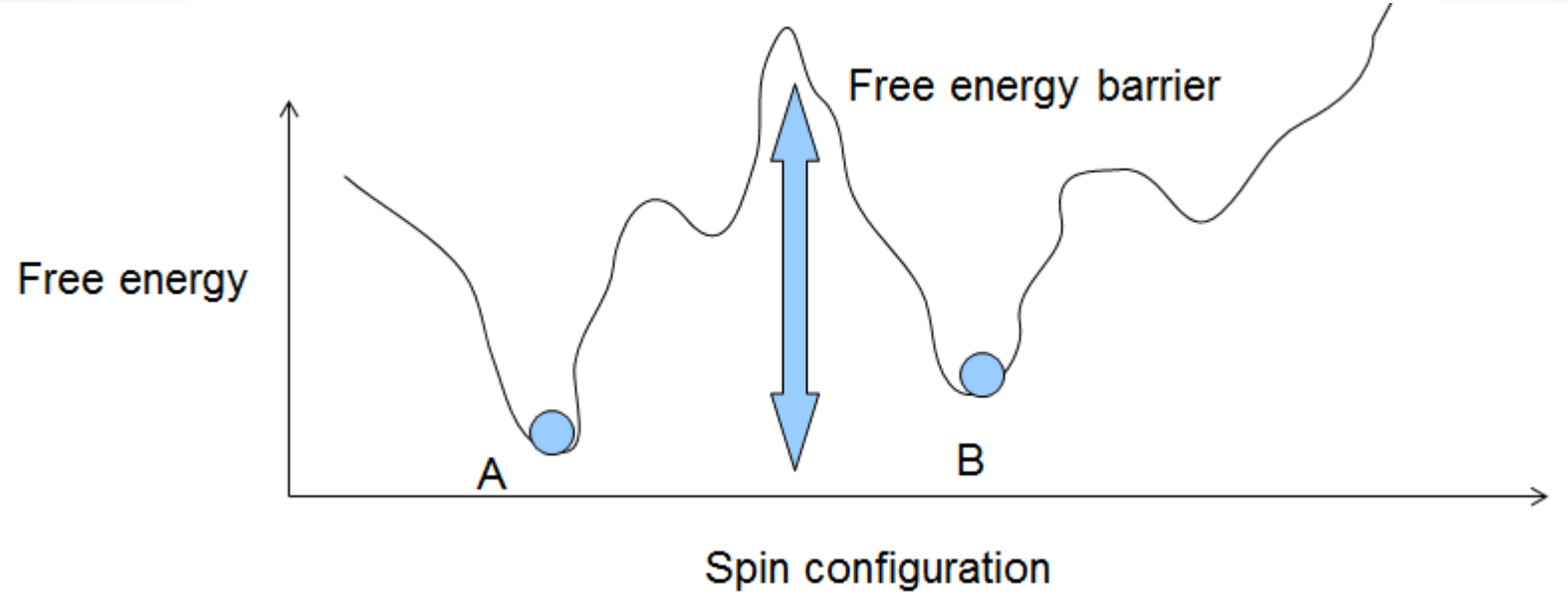
- The ising model is used to describe magnetic systems
- The model represents spins in a lattice with each spin oriented either up or down. These spins are coupled by nearest neighbor coupling, with a coupling constant, *J*.
- We can get a Hamiltonian for this model: $H_i = -J\sum_{j_{nn}} s_i s_j$



Metropolis Algorithm

- The Metropolis algorithm is a Markov chain Monte Carlo method.
- Useful for random samplings of a complex distribution, using an equation to calculate the probability of accepting a change/move.
- The only problem with this is if the calculated probability is so low that the chances a change/move will be accepted is very low.
- Parallel tempering is used to fix this problem

- When we use the metropolis algorithm on the ising model we propose flipping a spin in the configuration from spin up to spin down or vice versa.
- Our probability equation: $\frac{\exp(-\beta E_B)}{\exp(-\beta E_A)}$



Parallel tempering

- For that free energy barrier, parallel tempering is used.
- Other simulations are running at higher temperatures, we can propose a change of configurations with one of these simulations
- The probability of accepting this is given by the equation: $\frac{\exp(-\beta_B E_A)\exp(-\beta_A E_B)}{\exp(-\beta_B E_B)\exp(-\beta_A E_A)}$

The code

- The code has been parallelized on the gpu.
- The major bottleneck is memory access.
- We are using multi-spin coding to do multiple Ising spin calculations in one integery, allowing us to fit 14 simulations on one multiprocessor.
- Random number's are fetched from a pre-generated table stored in global memory.
- Improvements can be made using things like global memory to texture memory pre-fetching or making sure that the register and shared memory is being used as much as possible.