

Simulazione di un reticolo di Ising bidimensionale in CUDA

Stefano Mandelli

14 Luglio 2016

L'Hamiltoniana di Ising che identifica il modello studiato è

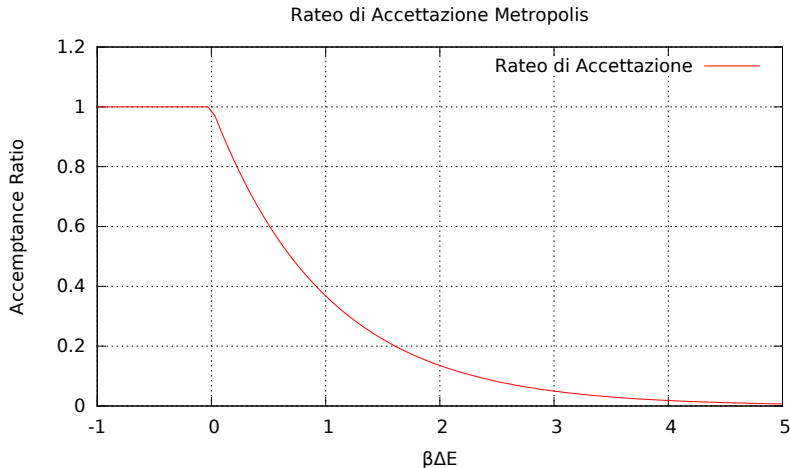
$$H = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i \quad (1)$$

le cui grandezze fisiche di interesse sono la magnetizzazione media e il calore specifico per spin definiti come

$$m = \frac{1}{N} \left\langle \sum_i s_i \right\rangle, \quad c_v = \frac{\beta^2 k_B^2}{N} \left(\langle E \rangle^2 - \langle E^2 \rangle \right) \quad (2)$$

Algoritmo di Metropolis

L'algoritmo con cui vengono proposti gli update degli spin è quello di metropolis la cui probabilità di accettazione è data da



Implementazione del codice su GPU

Ottimizzazione su memoria shared del reticolo attraverso aggiornamento su doppia scacchiera

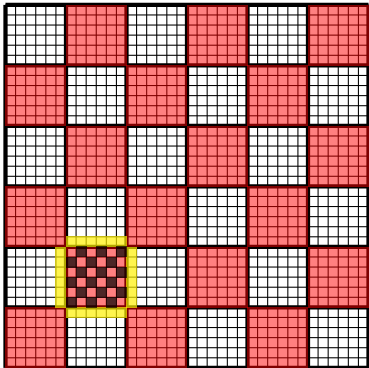
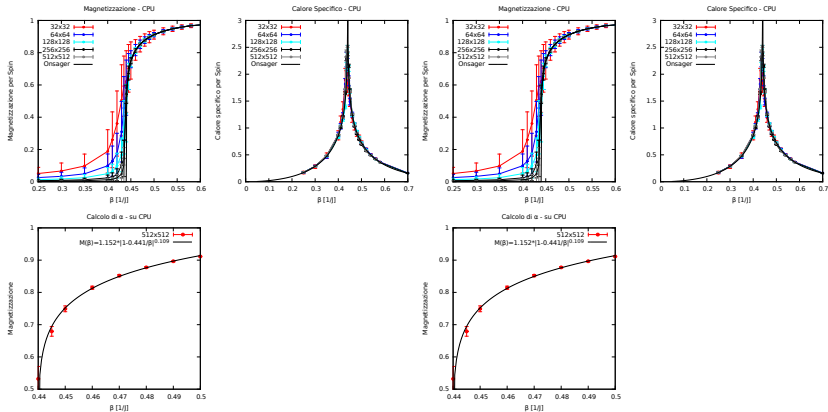


Figura: Le zone rosse più le aree gialle rappresentano i blocchi di memoria *shared*. Ogni blocco viene aggiornato a scacchiera. Le condizioni di raccordo blocco-blocco sono identificate dai siti reticolari colorati in giallo

Confronto Modello Fisico

Confronto CPU - GPU dei grafici di magnetizzazione e calore specifico con la soluzione di Onsager al limite termodinamico



Anche l'ottimizzazione di BLOCKL è fondamentale

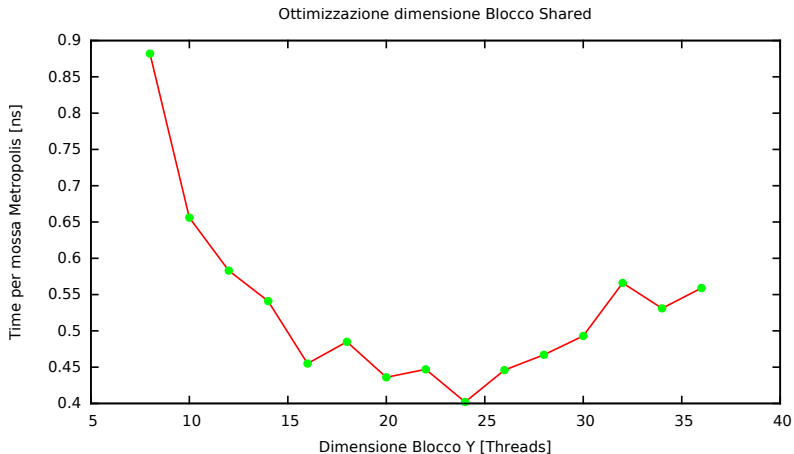


Figura: Il valore di BLOCKL che ottimizza al meglio le performance è 24

Ottimizzazione PNRG

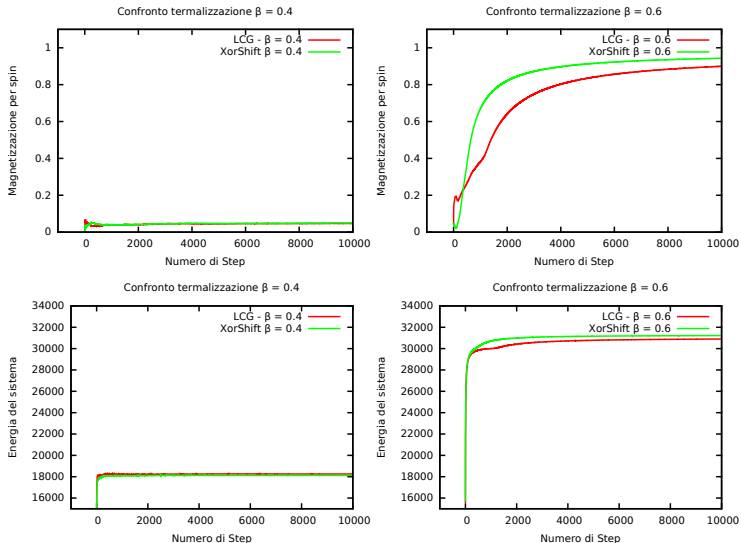


Figura: Grafici calcolati per un reticolo 128×128

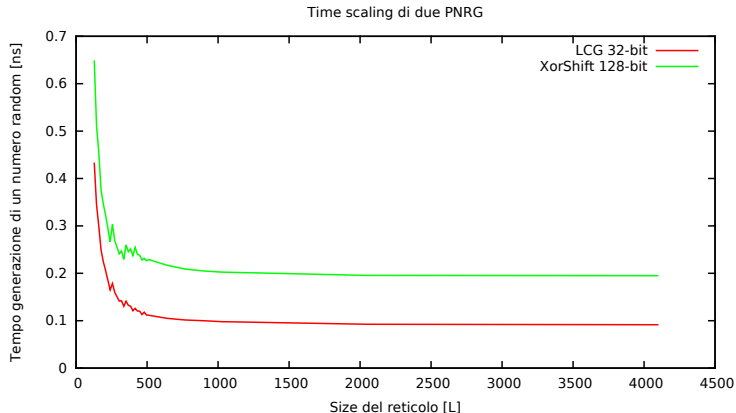


Figura: Tempo di generazione di un numero pseudorandom in funzione alla taglia del reticolo

Confronto sui tempi di simulazione

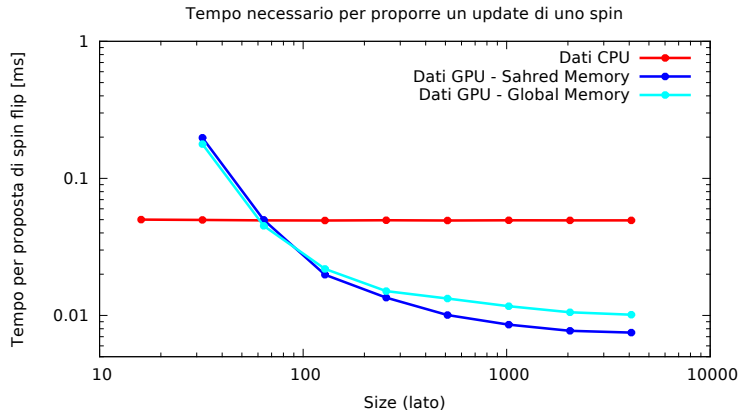


Figura: Tempo di proposta di update di uno spin, scalato per la taglia del reticolo