Simulating Polymer Systems on GPU

Wolfhard Janke

Institut für Theoretische Physik, Universität Leipzig, Germany

IMPRS School 2012: *GPU Computing – Methods and Applications in the Natural Sciences*

Wroclaw, Poland, 29 October – 02 November 2012

Bead-Spring Homopolymer Model

Lennard-Jones potential ($\epsilon = 1$, $\sigma = 2^{-1/6} r_0$, $r_0 = 0.7$):

$$E_{\rm LJ}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]$$

Truncated at $r_c = 2.5\sigma$:

$$E_{\rm LJ}^{\rm mod}(r_{ij}) = E_{\rm LJ}(\min(r_{ij}, r_c)) - E_{\rm LJ}(r_c)$$

Finitely extensible nonlinear elastic (FENE) anharmonic potential ($R = 0.3$, $K = 40$):

$$E_{\rm FENE}(r_{ii+1}) = -\frac{K}{2} R^2 \log \left[1 - \left(\frac{r_{ii+1} - r_0}{R} \right)^2 \right]$$

Total energy of a conformation $\mathcal{C} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$:

$$E(\mathcal{C}) = \sum_{i<j}^{N} E_{\rm LJ}^{\rm mod}(r_{ij}) + \sum_{i}^{N-1} E_{\rm FENE}(r_{ii+1})$$

Simulation Method

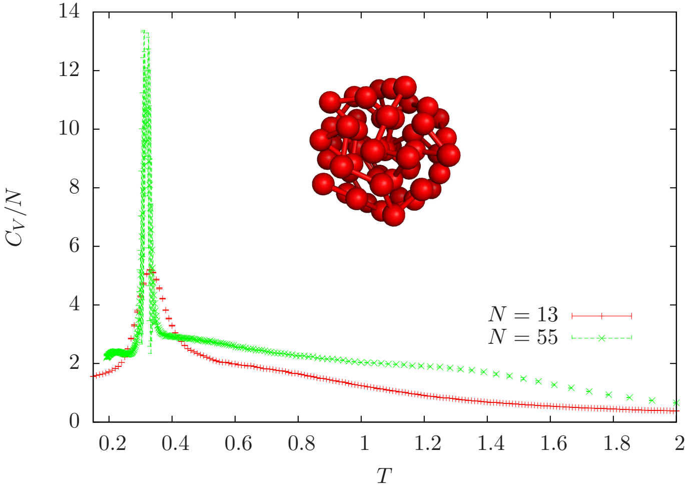
Parallel tempering (replica exchange) Monte Carlo with n_r replicas:
Exchange conformations of neighboring replicas i and $i+1$ with probability

$$p = \min\left(1, \exp\left[(E_i - E_{i+1})(\beta_i - \beta_{i+1})\right]\right)$$

J. Gross, WJ, M. Bachmann, Comp. Phys. Comm. **182** (2011) 1638; Physics Procedia **15** (2011) 29.

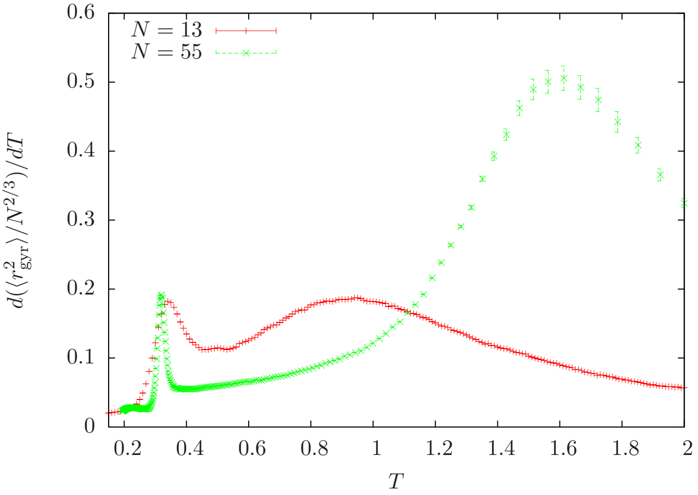
Specific Heat

Specific heat of an elastic polymer chain with lengths $N = 13$ and 55. The inset shows the icosahedral structure of the 55mer at low T.



Radius of Gyration

Temperature derivative of mean squared radius of gyration.



Specification of used Hardware

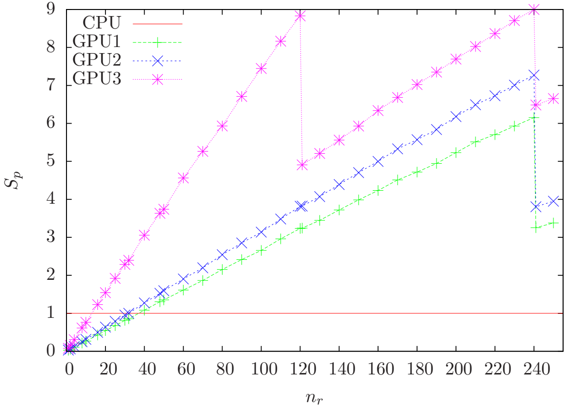
Table: Specifications of the used hardware.

	reference CPU	GPU1	GPU2	GPU3
name	Xeon E5620	Tesla C1060	GTX285	GTX480
# processors	1	30	30	15
# cores per processor	4 (1 used)	8	8	32
RAM	16384MB	4096MB	1024MB	1536MB
clock speed	2.4GHz	1.3GHz	1.48GHz	1.4GHz
max. threads per block	-	512	512	1024
shared memory size	-	16kB	16KB	48kB
registers per block	-	16384	16384	32768

Using CUDA on the GPUs
Run times for 55 monomers

Naive Parallel Tempering Implementation

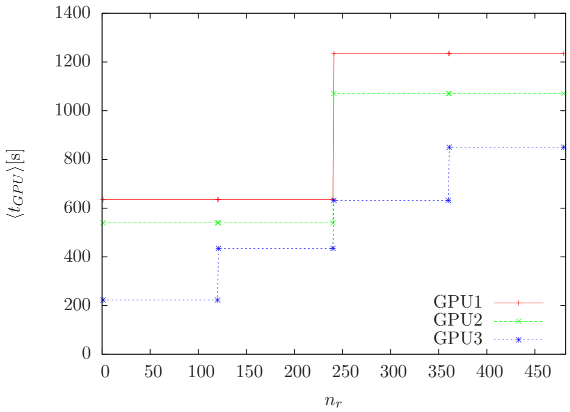
Each replica assigned to one thread block containing only one thread
("embarrassingly parallel")



GPU1,2: 30 procs × 8 threads = 240; GPU3: 15 procs × 8 threads = 120

Naive Parallel Tempering Implementation

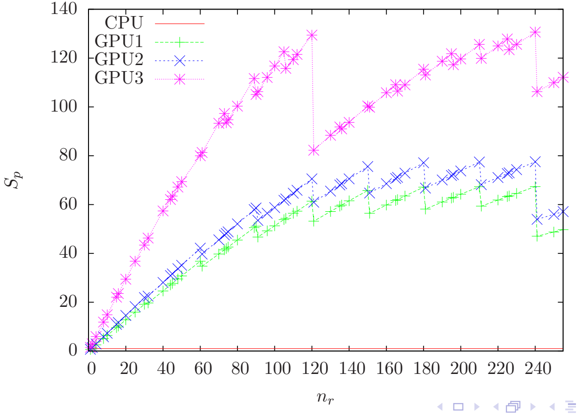
Number of replicas n_r = number of thread blocks



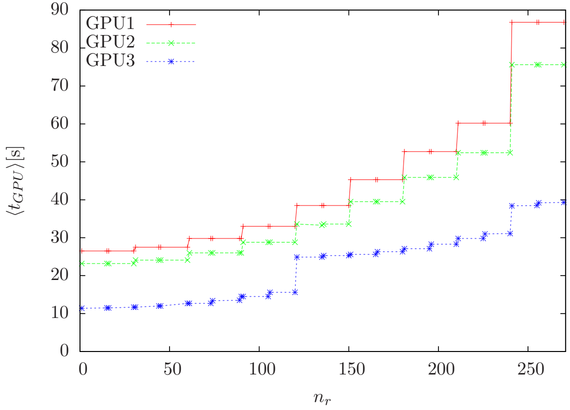
GPU1,2: $240/8 = 30 =$ all multiprocessors (SMs) are busy
GPU3: $120/8 = 15 =$ all multiprocessors (SMs) are busy

Improved Parallel Tempering Implementation

1. Parallel calculation of energy: 64 threads in thread block = 2 warps for one replica
2. Using shared memory for storing coordinates of monomers (all 64 threads have fast memory access)
3. Using optimized CUDA math library calls



Improved Parallel Tempering Implementation



GPU1,2: Steps in units of 30 = number of SMs; GPU3: Steps in units of 15

Speedup: GPU vs CPU

Table: Overview of maximum achieved speed-ups – $\max(S_p(n_r))$ – for the two different GPU implementations, compared to the single-core CPU implementation.

	naive	improved
GPU1	6.1$\times$	68$\times$
GPU2	7.2$\times$	78$\times$
GPU3	9$\times$	130$\times$

Summary and Outlook

- High speedup needs many parallel-tempering replica ($\rightarrow$ multi-histogram reweighting, multiplexing parallel tempering)
- Many cores CPUs
- My personal view

Acknowledgements

Collaborations with

- Michael Bachmann (now Univ. of Georgia, Athens, USA)
- Jonathan Gross (now Univ. of Georgia, Athens, USA)

Supported by

- DFG Sonderforschungsbereich SFB/TRR 102 (project B04)
- Sächsische DFG Forschergruppe FOR877 under Grant No. Ja 483/29-1
- Graduate School of Excellence GSC 185 “BuildMoNa”
- Deutsch-Französische Hochschule (DFH-UFA) Grant No. CDFA-02-07
- DFG Project JA 483/24-3
- Alexander von Humboldt Foundation

Advertisement

WORKSHOP CompPhys12:

13th International NTZ–Workshop on
Recent Developments in Computational Physics

29 November – 01 December 2012, ITP, Universität Leipzig
www.physik.uni-leipzig.de/˜janke/CompPhys12

See you there ?