Calculating Energy Momentum Tensor Correlators on GPU

Chuan Miao

Final Colloquium Int. Graduate School GRK 881
12-14 Sep. 2012, Bielefeld

Outline

- Motivation
- Advantages of GPU computing
- Energy momentum tensor correlators
- Conclusion and outlook

Motivation

QCD energy momentum tensor

$$\Theta_{\mu\nu} = T_{\mu\nu} + \frac{1}{4}\delta_{\mu\nu}\theta$$

- θ is trace anomaly due to breaking of scale invariance
- $T_{\mu\nu}$ is the traceless part of the EM tensor

Correlators $\langle T_{\mu\nu}(x) T_{\rho\sigma}(y)\rangle$ and $\langle\theta(x)\theta(y)\rangle$ @ $T = 0$

- Glueball states contribution (2^{++} and 0^{++} states)
- Multiple level algorithm used to improve the signal
- Anisotropic lattices are used ➔ isotropic lattices

Morningstar, Peardon (1999)
Meyer, Teper (2005)
Chen et al. (2006)

Motivation

Transport properties of QGP @ $T \neq 0$

- Shear and bulk viscosities from spectral function — Meyer (2008)

$$G(\tau, T) = \int_0^\infty d\tau\, e^{i\omega\tau} \rho(\omega, T) K(\omega, T, \tau)$$

$$\eta(T) = \pi \lim_{\omega \to 0} \frac{1}{\omega} \rho^{13,13}(\omega, T)$$

  – On anisotropic lattices, 2 renormalization factor needed, making determine the shape of correlator harder

➔Isotropic lattices, cutoff effects, high statistics

GPU is a good tool

NVIDIA TESLA C2070

Peak performance:

1.03 Tflops (single precision)

0.5 Tflops (double precision)

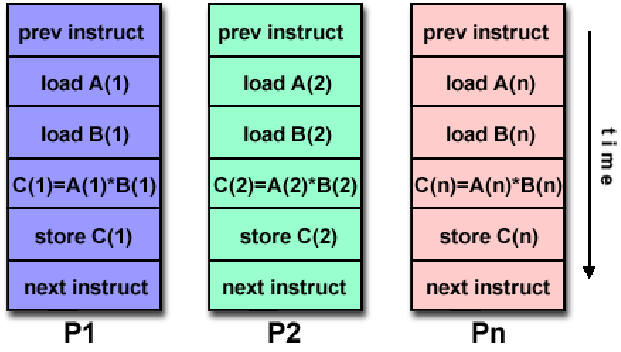
6 Gigabytes memory

GPU

NVIDIA Tesla C2070

- Why GPU is powerful?
  - Many stream processors (a few hundred)
    - Peak performance ~ 1 Tflops (single precision)
    - SPs are simpler than CPU cores
      - Small registers, limited shared memory
      - Has lower frequency ➔ cheaper and greener
  - SIMD
    - Processing data in parallel
    - Creating many threads, smart hiding memory latency

➔ easy programming



CUDA programming model

- Kernels (functions running on SPs)
  - Should be light weight
  - Avoid branches
  - Optimizing for memory access
  - But everything is in C language (eg. no need of SSE)

- Suitable for simulating pure gauge theory
  - Heatbath or overrelaxation algorithm is lightweight
  - Local interaction, transparent memory layout

Pseudo Random Number Generators

- CUDA provides random number generator library curand.
  - Its default RNG does not suitable for Monte Carlo
  - More advanced RNG from curand does not compile

- Tiny Mersenne Twister (Saito and Matsumoto, 2011)
  - Very long period 2^127-1 and good quality

Discretization of EM Tensor

Scalar channel, same form as gauge action

$$\theta = \frac{\beta(g)}{2g} F^a_{\rho\sigma} F^a_{\rho\sigma}$$

Plaquette discretization of trace anomaly

$$\theta(t) = -\frac{d g_0^{-2}}{d \ln a} \sum_x \sum_{\mu\nu} \mathrm{Re}\,\mathrm{tr}\{1 - P_{\mu\nu}(t, \boldsymbol{x})\}$$

Discretization of EM Tensor

Tensor channel

$$T_{\mu\nu} = \frac{1}{4}\delta_{\mu\nu}F^a_{\rho\sigma}F^a_{\rho\sigma} - F^a_{\mu\alpha}F^a_{\nu\alpha}$$

- Due to cubic symmetry, correlator of T_{ij} depend only on two functions
  $\langle T_{ij}(x)T_{kl}(y)\rangle = A(x-y)(\delta_{ik}+\delta_{jl}) + B(x-y)\delta_{ij}\delta_{kl}$
  ➔ $\langle T_{ij}(x)T_{ij}(y)\rangle = \frac{1}{4}\langle(T_{ii} - Tjj)(x)(T_{ii} - Tjj)(y)\rangle$

Plaquette discretization

$$T_{11} - T_{22} = \text{Re Tr}\,\{P_{10} + P_{13} - P_{20} - P_{23}\}$$

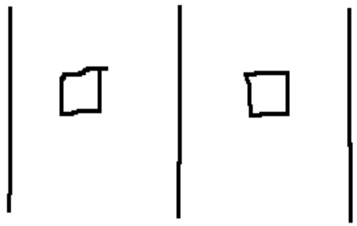
with add. renormalization factor $Z_l = 1 - \frac{1}{2}g^2(c_\sigma - c_\tau)$

2-level Algorithm

$$\langle\theta(t_1)\theta(t_2)\rangle = \frac{1}{N_{bc}}\sum_{bc}\langle\theta(t_1)\rangle_{bc}\langle\theta(t_2)\rangle_{bc}$$

One measurement includes

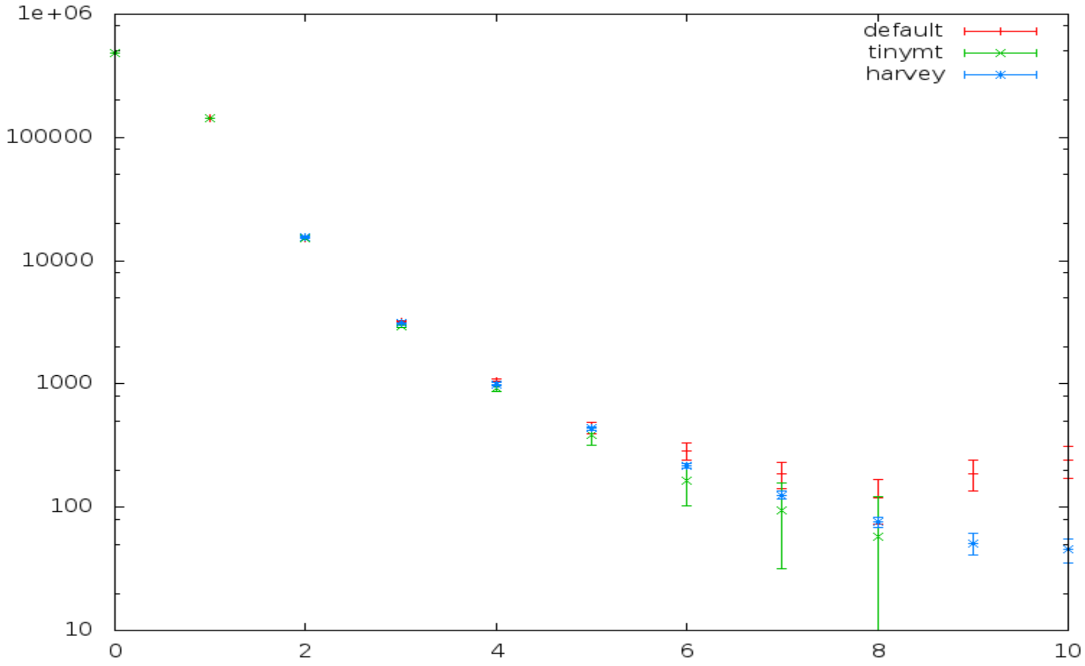
- Fix boundary, update and measure inside slab for n round
- Update globally



- Works only for pure gauge theory and local observables
- Reduce errors dramatically
- For EM tensors, number of hits inside slab does not seem matter

Preliminary

$< \theta\theta >_c$



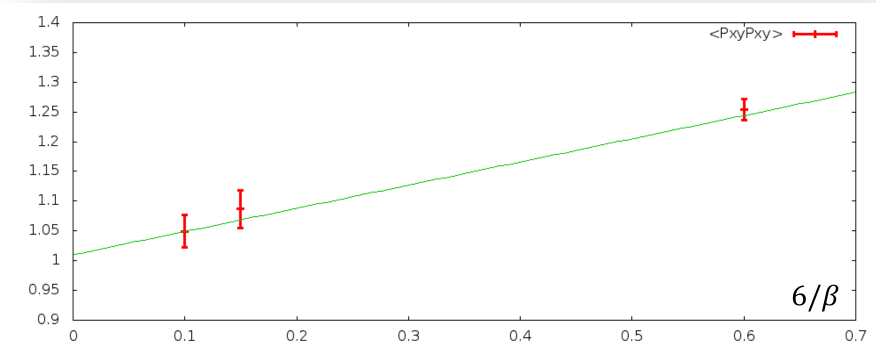
$\beta = 6.20$

$V = 20^4$

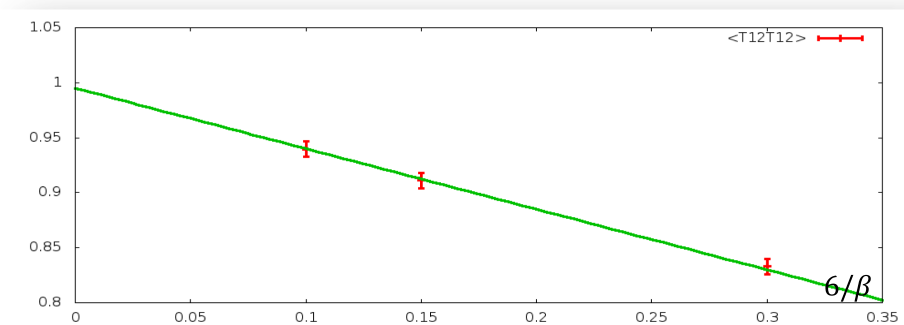
- Falls off quickly, two level algorithm helps
- Pseudo RNG tinymt

Tests @ weak coupling

$\langle P_{xy}\, P_{xy} \rangle$



$\langle (T_{11} - T_{12})(T_{11} - T_{22}) \rangle$



- Correlators @ t=1 on $N_\tau = 6$ lattices with $\beta = 10,20,40,60$ normalized to leading order lattice pert. values

Performance

- Heatbath per link takes ~ 30 ns on NVIDIA 2070
- Compared with 3.4 μs on Pentium 4 1.4 GHz with SSE (Luscher 2001)
- Performance comparable
  P4 ~ 8~10 Gflops vs. NVIDIA C2070 1Tflops

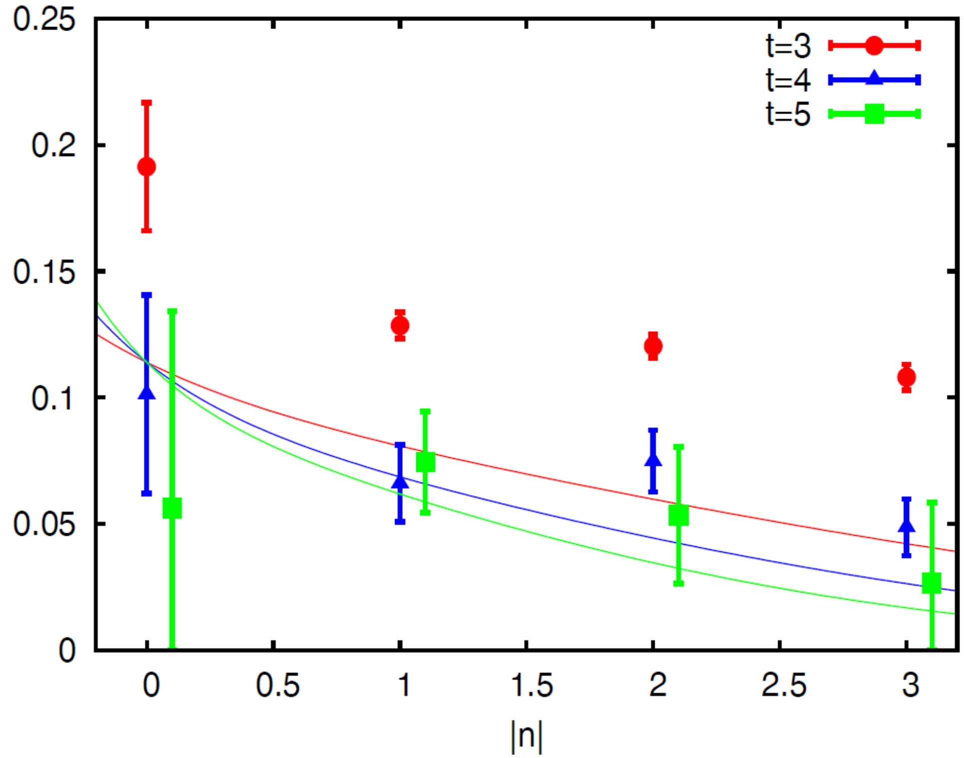
But GPU is much affordable and greener

Compare with perturbation

- $G_{12}(t, p = 0) \propto \frac{1}{t^5}$

- Correlator project to finite momentum with smaller errors
- Errors in lattice data increase with t
- Cutoff effects at small t

$$t^5 \int \langle T_{12}(0) T_{12}(t, \boldsymbol{x}) \rangle_c e^{i\boldsymbol{p}\cdot\boldsymbol{x}} d^3\boldsymbol{x}$$



$$\boldsymbol{p} = \frac{2\pi}{L}\boldsymbol{n}$$

Conclusion

- GPU is cheaper and greener
- and efficient with pure gauge simulations
- Easy programming, optimization for memory
- Need special care about pseudo random number generator

- Cutoff effects, to compare with lattice pert. Calculation
- Start with $T = 0$ with isotropic lattices $24^4, 32^4$ and 40^4 to extrapolate to continuum limit

Backup slides

$$48^3 \times 16$$

$$\frac{a_\sigma}{a_\tau} = 2$$