GPU enhanced spin dynamics with Trotter-Suzuki evolutions

Axel D. Dente*, Pablo R. Zangara and Horacio M. Pastawski

Instituto de Física Enrique Gaviola (IFEG) – CONICET, Argentina

Facultad de Matemática, Astronomía y Física (FaMAF) - Universidad Nacional de Córdoba, Argentina

Objective:

Accelerate Quantum Spin Dynamics simulations.

Introduction:

Quantum simulations are used in Physics and Chemistry.

Efficient simulations of spin systems are relevant for ensemble quantum computation as well as in NMR experiments.

The Problem: **Required computational resources increase exponentially with the size of the system.**

Here we present a GPU implementation for the Trotter-Suzuki algorithm which has been previously implemented on CPUs.

We compare the wall times for two GPU programs written in **Cuda FORTRAN** and **Cuda C**.

Trotter-Suzuki Approximation

$$\mathbf{U}(t) = e^{-\mathrm{i}tH} = e^{-\mathrm{i}t(H_1+\ldots+H_K)}$$

$$= \lim_{m\to\infty} \left(\prod_{k=1}^{K} e^{-\frac{\mathrm{i}tH_k}{m}} \right)^m$$

$$\tilde{\mathbf{U}}_\mathbf{1}(t) = e^{-\mathrm{i}tH} = e^{-\frac{\mathrm{i}tH_1}{m}} \ldots e^{-\frac{\mathrm{i}tH_K}{m}}$$

The total evolution is separated in a set of small partial evolutions

High orders Approximations

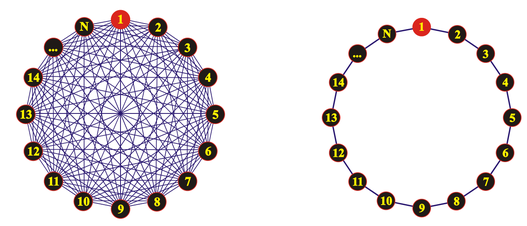
$$\tilde{\mathbf{U}}_\mathbf{2}(t) = \tilde{\mathbf{U}}_\mathbf{1}^\dagger(-t/2)\tilde{\mathbf{U}}_\mathbf{1}(t/2)$$

$$\tilde{\mathbf{U}}_\mathbf{4}(t) = \tilde{\mathbf{U}}_\mathbf{2}(at)\tilde{\mathbf{U}}_\mathbf{2}(at)\tilde{\mathbf{U}}_\mathbf{2}((1-4a)t) \times \tilde{\mathbf{U}}_\mathbf{2}(at)\tilde{\mathbf{U}}_\mathbf{2}(at)$$

The 4th order can be evaluated from 1st order evolutions.

With Trotter-Suzuki we can solve General Spin Problems without any specific assumption

$$H = \sum_{j=1}^{N} \sum_{\alpha=x,y,z} h_j^\alpha \, S_j^\alpha + \sum_{j,k=1}^{N} \sum_{\alpha=x,y,z} J_{j,k}^\alpha S_j^\alpha S_k^\alpha .$$



Single-spin evolution operator

$$\exp\left(-it\left[\sum_{j=1}^{N}\sum_{\alpha=x,y,z} h_j^\alpha \, S_j^\alpha\right]\right) = \prod_{j=1}^{N} \exp\left(-it \sum_{\alpha=x,y,z} h_j^\alpha \, S_j^\alpha\right)$$

The evolution determined by the jth factor rotates the spin j about the vector $\mathbf{h}_j = (h_j^x, h_j^y, h_j^z)$.

State Representation

↓↑↓↑↑↓↓ → 0101100 (in binary notation)
44 (in decimal notation)

Parallelization of Rotations

↓↓↓↓ , ↓↓↓↑ , ↓↓↑↓ , ↓↓↑↑ , ↑↓↓↓ , ↑↓↓↑ , ↑↓↑↓ , ↑↓↑↑ , ↓↑↓↓ , ↓↑↓↑ , ↓↑↑↓ , ↓↑↑↑ , ↑↑↓↓ , ↑↑↓↑ , ↑↑↑↓ and ↑↑↑↑ .

Single individual rotations are parallelized.

Two-spin terms

$$H^z = \sum_{j,k=1}^{N} J_{j,k}^z S_j^z S_k^z$$

The evolution of this kind of interaction only involve phase corrections (which can be easily evaluated)

We pre-transform the "x" and "y" terms into "z" Hamiltonians

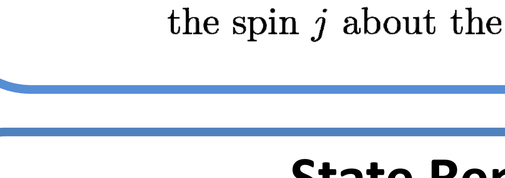
$$e^{-\mathrm{i}tH^x} = \bar{Y}Ye^{-\mathrm{i}tH^x}\bar{Y}Y = \bar{Y}\exp\left(-it\left[\sum_{j,k=1}^{N} J_{j,k}^x S_j^z S_k^z\right]\right)Y$$

$$e^{-\mathrm{i}tH^y} = X\bar{X}e^{-\mathrm{i}tH^y}X\bar{X} = X\exp\left(-it\left[\sum_{j,k=1}^{N} J_{j,k}^y S_j^z S_k^z\right]\right)\bar{X}$$

Single-Spin Rotations ← $X = \prod_{j=1}^{N} X_j$; $Y = \prod_{j=1}^{N} Y_j$

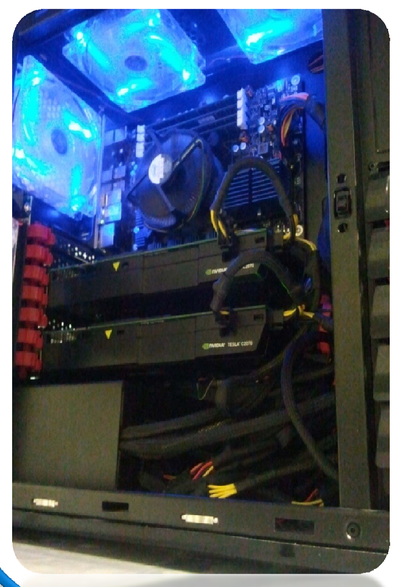
Rotation $X90$ → Phases in Z → Rotation X-90

RESULTS



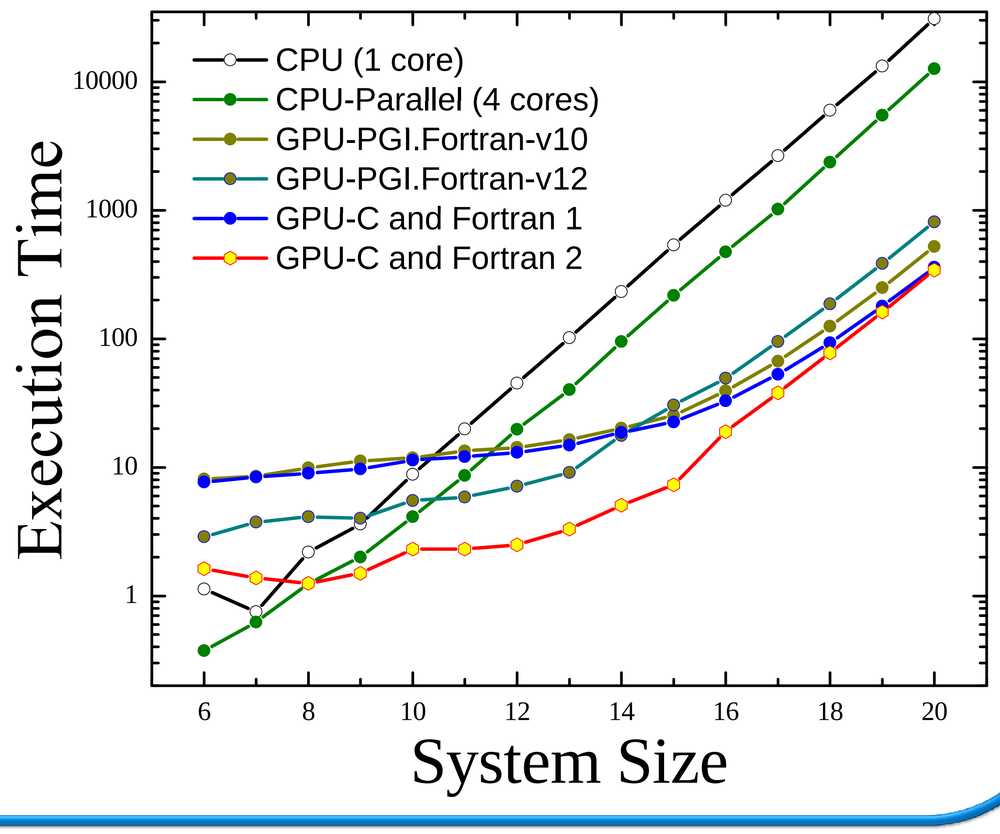
Compilers

We use INTEL compilers for the CPU version and PGI-FORTRAN and NVCC for the GPUs versions



Resources

Intel DX58SO motherboard with Intel Core i7-950@3.07GHz, 16GB DDR3 1066MHz, holding twoTesla C2070 boards (448 cores, 6GB GDDR5 ECC RAM each), donated by NVIDIA.



CONCLUSION

The comparison shows that the GPU implementation with the nvcc compiler acquired the best performance for this problem.

*Email: dente@famaf.unc.edu.ar

www.lanais.famaf.unc.edu.ar & www.famaf.unc.edu.ar/grupos/GPGPU/