Parallel and Distributed Programming for Data/Computation Intensive Applications

Bilal Jan

Department of Control and Computer Engineering
Politecnico di Torino

Supervisors: Bartolomeo Montrucchio (DAUIN)
Carlo Ragusa (DENERG)

February 27, 2015

Outlines

Introduction

Parallel Sorting of Large Data Sets

Fast Fourier Transforms on GPU

Accelerating Micromagnetic Simulations on GPUs

Conclusion

GPU Computing

- An assembly of hundreds and thousands of PUs
- SIMD Processing: Single Instruction on Multiple Data streams simultaneously
- Well suited for highly parallel numeric applications
- Best Price/Performance ratio
- Programming Tools
  - CUDA (NVIDIA Proprietary)
  - OpenCL (Open Standard and Heterogeneous)

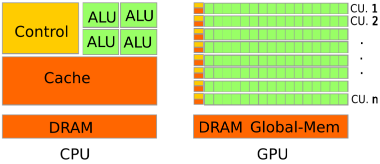


Figure: CPU vs GPU core ratio

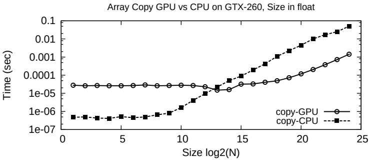


Figure: Performance Comparison CPU vs GPU

Parallel Sorting 1/3: Min-Max Butterfly Sort

- Finds minimum and maximum in data
- For data of size $\boldsymbol{N}$
  - Total Stages are $log_2\boldsymbol{N}$
  - Complexity in terms of butterflies (comparators) is $(N/2)log_2N$
- All stages are executed sequentially
- Butterflies inside any stage $\boldsymbol{S}_i$ are executed in parallel
- After complete run of the Algorithm minimum and maximum values in data are placed at x(0) and x(N-1)

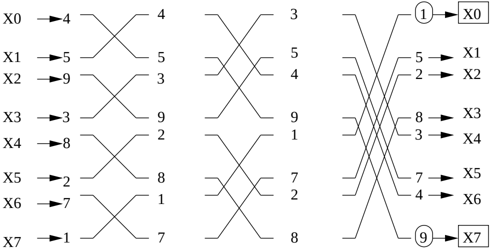


Fig **: 1** 8x8 Min-Max Butterfly

Parallel Sorting 2/3: Full Butterfly Sorting

- Complete Sorting
- For data of size ***N***
  - Total Stages are $log_2 N + \sum_{\mathbf{r}=1}^{log_2 N - 1}(\mathbf{r})$
  - Complexity in terms of butterflies (comparators) is $(N/2)xT.Stages$
- All stages are executed sequentially
- Butterflies inside any stage $\boldsymbol{S}_i$ are executed in parallel

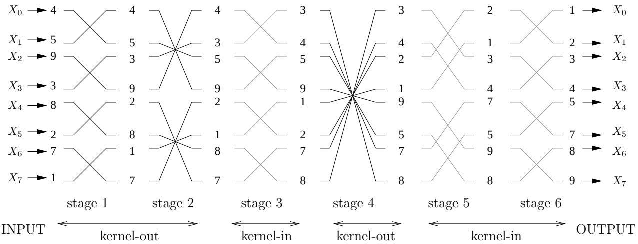
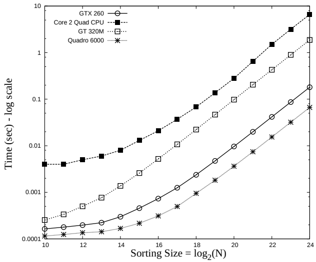


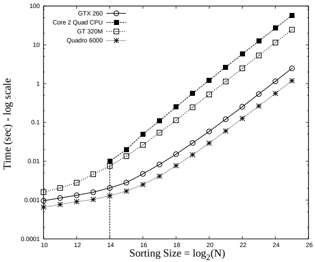
Fig **:** Size 8 Full Butterfly Sorting.

[2]Published in *"IASTED-Parallel and Distributed Computing and Networks 2013"*

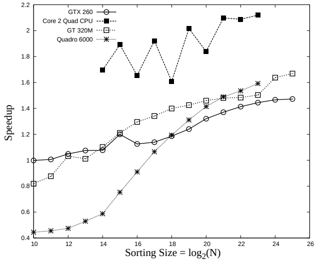
Parallel Sorting 3/3: Results Butterfly Network Sort



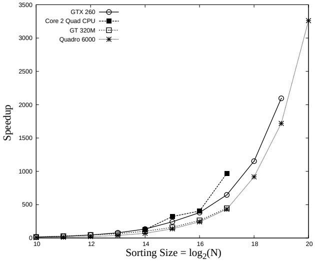
Sorting time of Min-Max Butterfly sort for various input size and devices types.



Sorting time of full Butterfly Sort for various input in the power of 2 and different devices types



Speedup of full Butterfly sort vs Bitonic sort.



Speedup of full Butterfly sort against less parallel Odd-Even sort.

Fourier Transformation on GPU

- DFT converts a time domain signal into frequency domain.
- High computation complexity $\mathcal{O}(n^2)$
- FFT are fast methods for computing the DFT.
- FFT complexity $\mathcal{O}(n \log n)$.
- The parallel structure of Cooley-Tukey FFT is well suited for GPU architecture

$$X(k) = \sum_{n=0}^{N-1} x(n)\, \mathbb{W}_N^{nk}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbb{W}^0 & \mathbb{W}^0 & \cdots & \mathbb{W}^0 \\ \mathbb{W}^0 & \mathbb{W}^1 & \cdots & \mathbb{W}^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{W}^0 & \mathbb{W}^{N-1} & \cdots & \mathbb{W}^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

$$X[K] = \sum_{n=0}^{\frac{N}{2}-1} x[2n]\mathbb{W}_{N/2}^{nk} + \mathbb{W}_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]\mathbb{W}_{N/2}^{nk}$$

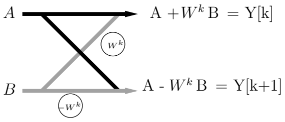
$$X[K] = A[k] + \mathbb{W}_N^k \cdot B[k]$$

The Cooley-Tukey Algorithm for FFT

Twiddle Properties

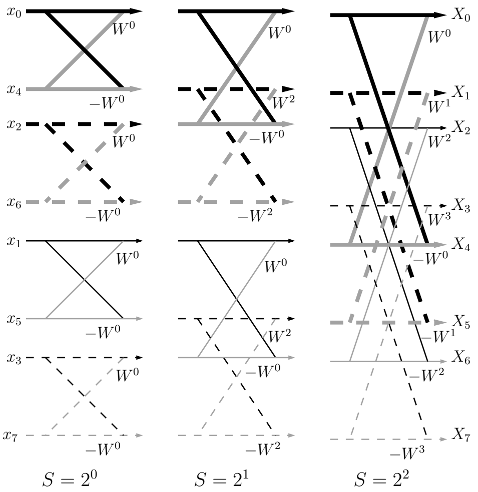
- Symmetry
  $\mathbb{W}_N^{k+\frac{N}{2}} = -\mathbb{W}_N^k$
- Periodicity
  $\mathbb{W}_N^{k+N} = \mathbb{W}_N^k$
- Recursion
  $\mathbb{W}_N^2 = -\mathbb{W}_{N/2}$

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} A \\ B \end{bmatrix}$$



2 Point DFT

Cooley-Tukey DIT-FFT, Radix-2, N=8



S=Memory Load/Store Stride

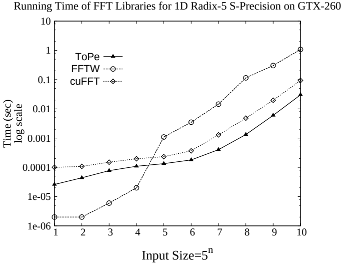
ToPe: An OpenCL based multidimensional FFT Library

- Almost Arbitrary length transform size
- Complex-to-Complex Transform type
- Multi-Radix ($N = r^n$, where $r = 2 - 8, 10, 15, 16$)
- Algorithms (Cooley-Tukey DIT, Modified Cooley-Tukey, Mixed Radix FFT)
- Dimension supported up to 3D
- Precision (single and double)
- Auto tuning for multiple GPU with Static Load Balancing (GPU+Thread Level)
- Open Source (`http://code.google.com/p/tope-fft`)

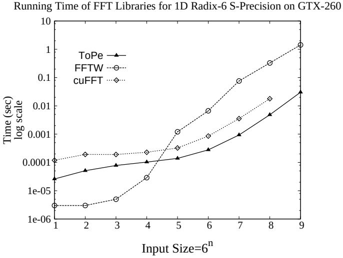
Speedup over FFTW $\sim 50\times$

Speedup over cuFFT $\sim 5\times$

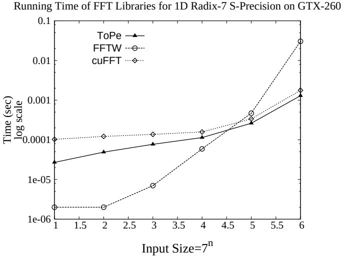
Results 1/2: ToPe FFT



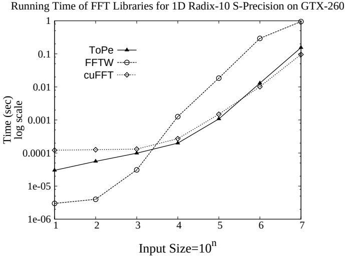
Radix-5



Radix-6

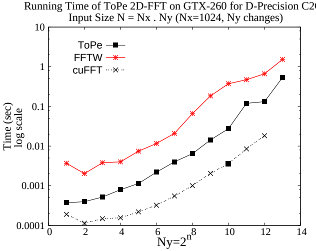


Radix7

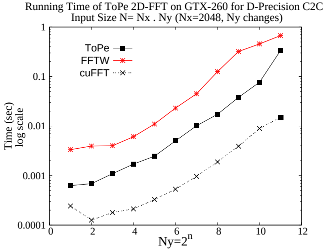


Radix10

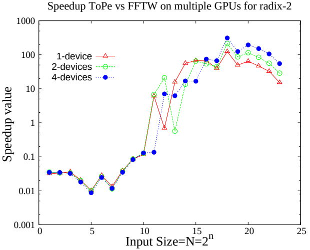
Results 2/2: ToPe FFT



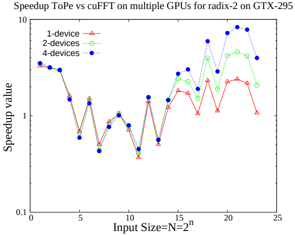
2D FFT



2D FFT



1D Speedup against FFTW on Multiple GPUs



1D Speedup against cuFFT on Multiple GPUs

Micromagnetics 1/3:Accelerating Magnetostatic Field Computation using GPUs

The equation for Magnetostatic field **H** at a cell position is

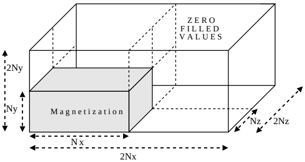
$$\mathbf{H}(\mathbf{r}) = -\sum_{\mathbf{r}'}^{n} \mathbf{N}\left(\mathbf{r}-\mathbf{r}'\right)\cdot \mathbf{m}\left(\mathbf{r}'\right)$$

N is 3 ×3 geometric tensor and **m** is magnetization

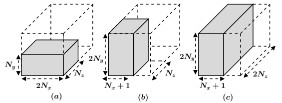
$$\widetilde{\mathbf{H}} = -\widetilde{\mathbf{N}}\cdot\widetilde{\mathbf{M}}$$

$$\widetilde{\mathbf{M}}\left(k_x', k_y', k_z'\right) = \sum_{r_z'=0}^{2N_z-1}\sum_{r_y'=0}^{2N_y-1}\sum_{r_x'=0}^{2N_x-1} \mathbf{m}\left(r_x', r_y', r_z'\right) \exp\left[\frac{-2\pi \mathrm{j} r_x' k_x'}{2N_x}\right] \exp\left[\frac{-2\pi \mathrm{j} r_y' k_y'}{2N_y}\right] \exp\left[\frac{-2\pi \mathrm{j} r_z' k_z'}{2N_z}\right]$$

$$\begin{aligned}
\tilde{\mathbf{H}}_{x(i,j,k)} &= \tilde{\mathbf{N}}_{xx'(i,j,k)}\tilde{\mathbf{M}}_{x(i,j,k)} + \tilde{\mathbf{N}}_{xy'(i,j,k)}\tilde{\mathbf{M}}_{y(i,j,k)} + \tilde{\mathbf{N}}_{xz'(i,j,k)}\tilde{\mathbf{M}}_{z(i,j,k)}\\
\tilde{\mathbf{H}}_{y(i,j,k)} &= \tilde{\mathbf{N}}_{yx'(i,j,k)}\tilde{\mathbf{M}}_{x(i,j,k)} + \tilde{\mathbf{N}}_{yy'(i,j,k)}\tilde{\mathbf{M}}_{y(i,j,k)} + \tilde{\mathbf{N}}_{yz'(i,j,k)}\tilde{\mathbf{M}}_{z(i,j,k)}\\
\tilde{\mathbf{H}}_{z(i,j,k)} &= \tilde{\mathbf{N}}_{zx'(i,j,k)}\tilde{\mathbf{M}}_{x(i,j,k)} + \tilde{\mathbf{N}}_{zy'(i,j,k)}\tilde{\mathbf{M}}_{y(i,j,k)} + \tilde{\mathbf{N}}_{zz'(i,j,k)}\tilde{\mathbf{M}}_{z(i,j,k)}
\end{aligned}$$



A magnetized body with non-periodic ***m****. The size of the magnetic body is doubled along each axis to remove aliasing effects and to make a periodic signal in Fourier space.*



(a)Firslty Ny Nz transforms are carried out along the x-axis. (b) In the second stage (N x + 1) Nz transforms are carried out along the y-axis. The +1 due to conjugate property of FFT's.(c) Finally, (N x + 1) 2Ny transforms along the z-axis.

[2] Portion of this work has been published in "10th European Conference on Magnetic Sensors and Actuators, Vienna, Austria"

Micromagnetics 2/3: Flowchart of our GPU Field Solver

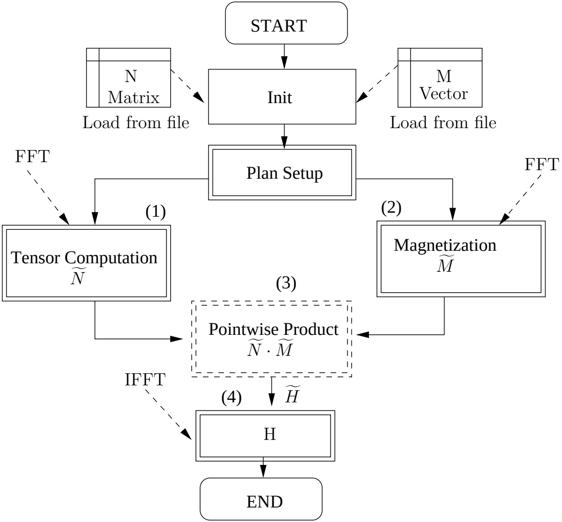


Figure: Design Overview of GPU Magnetostatic Field solver. The double line rectangles show processes performed in parallel by the GPU. The dotted line rectangles show the dot product performed by parallel threads on CPU.

Micromagnetics 3/3: Results and Comparisons

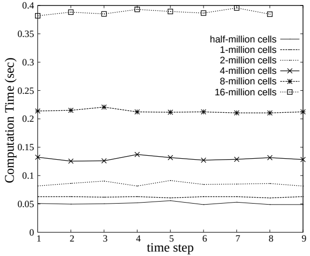


Fig:Total Demag field time GPU-GTX 260 S-Precision with MTT and point-wise multiplication time on CPU

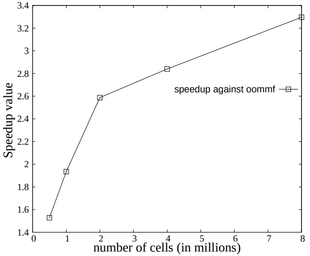


Fig: Total Demag Field Computation speedup against OOMMF

Accuracy

Computation Cells	ϵ_{mean}	ϵ_{max}	η_{mean}
1 Million	$7.59e-11$	$3.52e-10$	$1.01e-14$
2 Million	$4.04e-11$	$1.68e-10$	$5.33e-15$
4 Million	$7.56e-11$	$3.65e-10$	$9.96e-15$
8 Million	$7.91e-11$	$3.38e-10$	$1.04e-14$

Table: *Validation of our simulation computing* $\mathbf{H}$ *against* μ*-mag Standard problem 4 with an S-state initial magnetization. Here* $\mathbf{H}'$ *is OOMMF computed. Here,* $\epsilon = \left|\mathbf{H} - \mathbf{H}'\right|_2$, *and* $\eta = \epsilon / \left|\mathbf{H}'\right|$

[3] This work has been accepted for publication in Elsevier "Computers & Electrical Engineering" Journal

Time Stepping Technique 1/2: Runge Kutta like scheme for the Integration of LLG

- Numerical solution for Landau-Lifshitz equation of motion for magnetization
- The fundamental LLG equation:

$$\frac{\partial \mathbf{m}}{\partial t} = -\mathbf{m} \times \left(\mathbf{h}_{\text{eff}} + \alpha \left(\mathbf{m} \times \mathbf{h}_{\text{eff}}\right)\right)$$

- Discretization using mid-point rule:

$$\mathbf{m}^{k+1}(i) - \mathbf{m}^{k}(i) = \tau/2 \left[\mathbf{m}^{k+1}(i) + \mathbf{m}^{k}(i)\right] \times \mathcal{H}$$

- Time-stepping scheme based on the properties of mid-point rule and Runge-Kutta method.
- Properties of magnetization dynamics are preserved in all steps.
- At each time-step a 3×3 linear system of equations is solved instead of $3N \times 3N$ in the previous cases
- It reduces the computations of the $\mathbf{h}_{\text{eff}}$ per time-step.

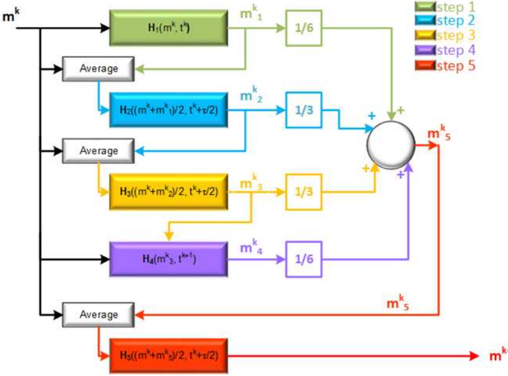
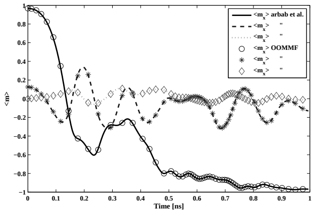
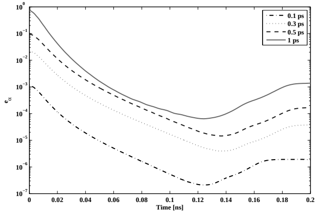


Fig: Flow of different steps.

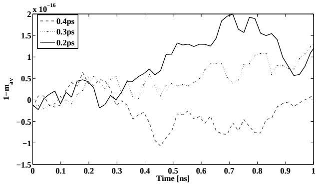
Time Stepping Technique 2/2: Results Runge Kutta like scheme ...



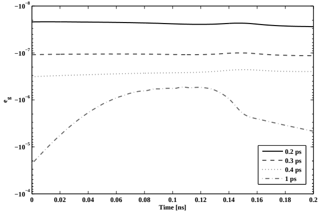
*Comparison with OOMMF using muMAG Standard Problem 4. Plot of spatially average magnetization <**m** > versus time with constant applied field at an angle of* 190° *off x-axis and time-step of 1.25 ps.*



Plot of the relative error $\mathrm{e}_\alpha = (\tilde{\alpha} - \alpha)/\alpha$ *a as a function of time for a 5 nm cell size and various time-steps*



Plot of 1-mav as a function of time for various time-steps, confirming the conservation of magnetization.



Plot of the relative energey for conservative case $\alpha = 0$ *as a function of time at various time-steps.The energy is preserved for all time steps.*

[1] This work has been published in "Journal of Applied Physics"

Conclusion

- Developed and implemented new sorting algorithms
- Developed generic FFT library on GPUs
- GPU accelerated Magnetostatic Field Solver
- Designed and developed new time integration method for LLG equation
- Designed new load balancing scheme on multiple GPUs

Publications

Proceedings

- (2013) B. Jan, B. Montrucchio, C. Ragusa, F.G. KHAN, O.U. KHAN, “Parallel Buttterfly Sorting Algorithmn”. Proceeding, IASTED-Parallel and Distributed Computing and Networks PDCN-2013 Innsbruck, Austria.
DOI:10.2316/P.2013.795-026.
- (2014) O. Khan, B. Jan, C. Ragusa, A. Rahim, F. Khan, B. Montrucchio, “Optimization of a Mult-Dimensional FFT Library for Accelerating Magnetostatic Field Computations”. In: 10th European Conference on Magnetic Sensors and Actuators, Vienna, Austria, July 6-9, 2014. p.250, ISBN 9783854650218.

Journals

- (2012) B. Jan, B. Montrucchio, C. Ragusa, F. Khan, O. Khan, “Fast parallel sorting algorithms on GPUs”. In International Journal of Distributed and Parallel Systems, vol 3,pp.107-118. ISSN 2229-3957, DOI:10.5121/ijdps.2012.3609.
- (2014) A. Rahim, C. Ragusa, B. Jan, O. Khan, “ A mixed mid-point Runge-Kutta like scheme for the integration of LLG”, in Journal of Applied Physics, vol. 115 n.17, 17D101, ISSN 0021-8979.
- (2015) F.Khan, B. Jan, C.Ragusa, B. Montrucchio and O. Khan, "An Optimized Magnetostatic Field Solver on GPU using OpenCL", in Elsevier journal on "Computers and Electrical Engineering".

Thank You for Your Attention.