A Hybrid Method for Solving Tridiagonal Systems on GPU

UC DAVIS UNIVERSITY OF CALIFORNIA

Yao Zhang
Andrew A. Davidson
John D. Owens

Jonathan Cohen

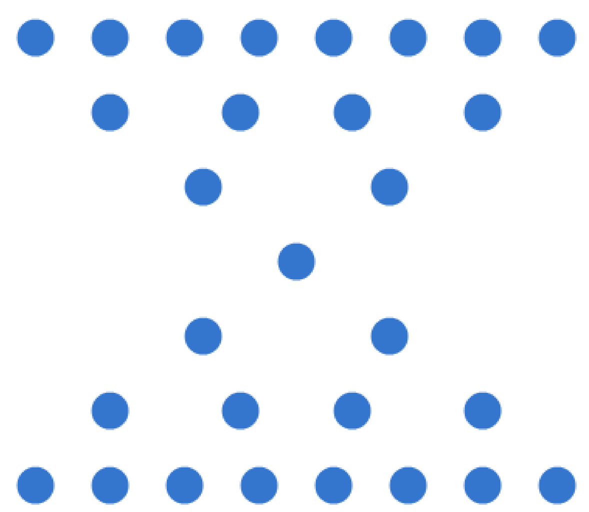
NVIDIA

The Hybrid Method

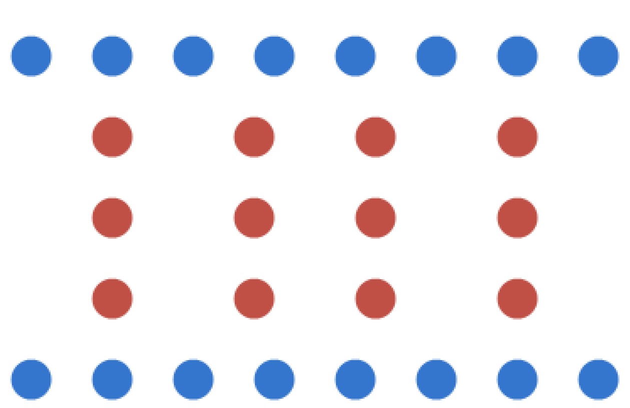
A comparison between the CR, PCR and hybrid algorithms in terms of algorithmic steps and work per step for solving an 8-equation system. A dot stands for a unit of work (an equation), and a row of dots stands for an algorithmic step. We assume the length of vector arithmetic unit is 4 in this example.

The Thomas Algorithm
2*8 = 16 steps.
The vector unit has a utilization rate of 25% through all steps.
Less work per step, more steps.

Parallel Cyclic Reduction (PCR)
log2(8) = 3 steps.
The vector unit is fully utilized across all steps.
More work per step, fewer steps.



Cyclic Reduction (CR)
2log2(8) = 6 steps.
The vector unit is partially idle during the middle three steps.
Less work per step, more steps.



CR-PCR
2 + log2(4) = 4 steps.
The vector unit is fully utilized across all steps.
Less work per step than PCR, fewer steps than CR.

PCR-Thomas
2 + 2*2 = 6 steps.
The vector unit is fully utilized across all steps.
Less work per step than PCR, fewer steps than the Thomas algorithm.

Problem Statement

$$\begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & c_{n-1} \\ & & & & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ \vdots \\ d_n \end{pmatrix}$$

Algorithm	Algorithmic steps	Work per step
The Thomas Algorithm	$2n$	1
Cyclic Reduction	$2\log_2 n$	$(1, \frac{n}{2})$
Parallel Cyclic Reduction	$\log_2 n$	n

Numerous Applications
Shallow Water Simulation
Depth-of-fields Blurs
Numerical Ocean Models
Spectral Poisson Solvers
Cubic Spline Approximation
Semi-coarsening for Multi-grid Solvers
Alternating Direction Implicit (ADI) Method
Pre-conditioners for Iterative Linear Solvers

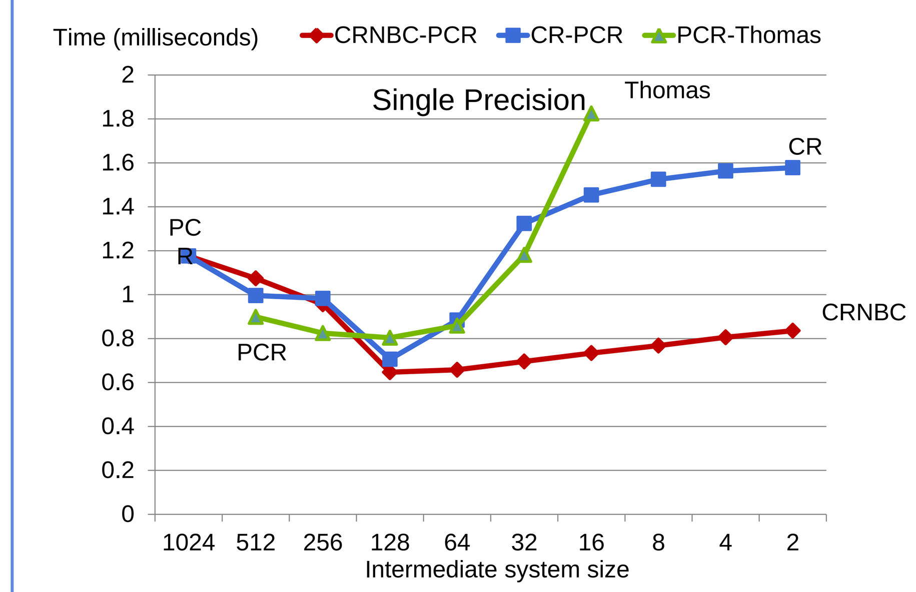
Reference

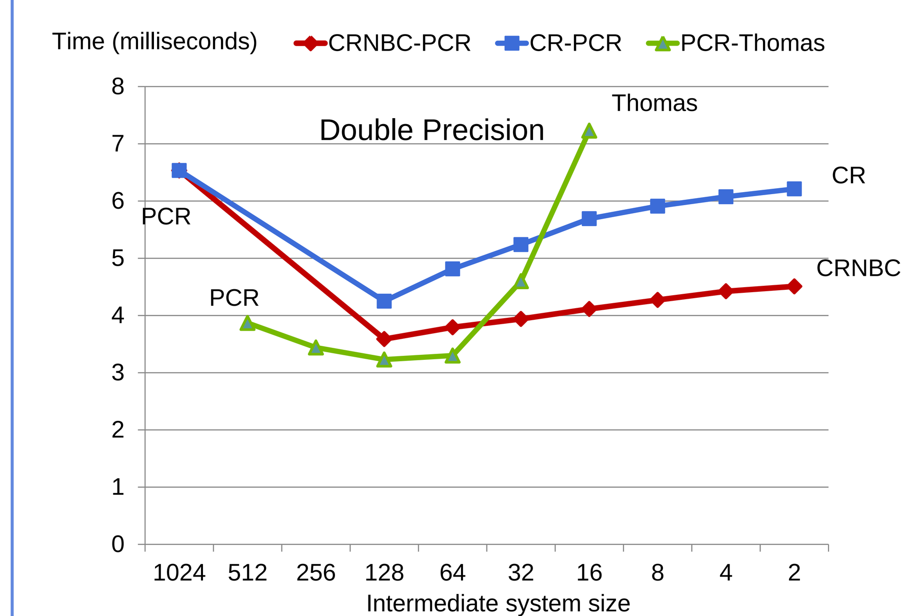
[1] Yao Zhang, Jonathan Cohen, Andrew A. Davidson, John D. Owens. A Hybrid Method for Solving Tridiagonal Systems on the GPU. In GPU Computing Gems, 2010. In press.

[2] Oles Shishkovtsov and Ashu Rege. DX11 effects in Metro 2033, March 2010.

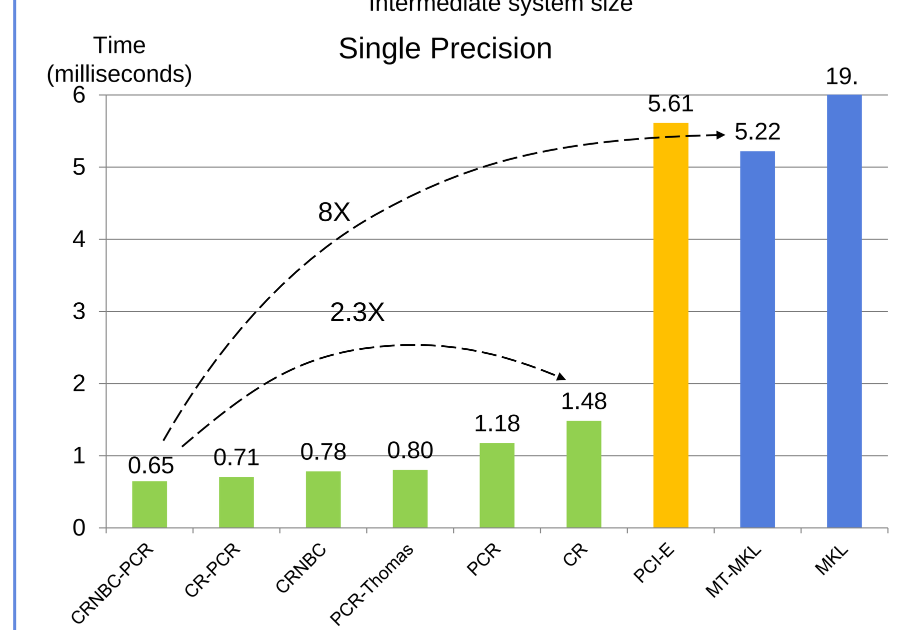
Performance Results

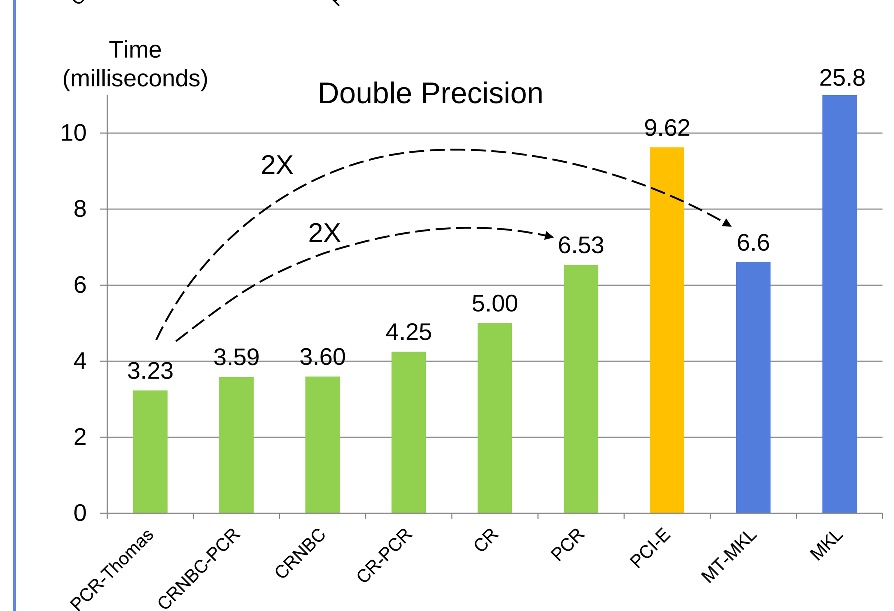
Our test platform uses a 2.8 GHz Intel Core i7 quad-core CPU, a GTX 480 graphics card with 1.5 GB video memory, CUDA 3.1 and the Windows 7 operating system.





Timings for the hybrid solvers with various intermediate system sizes for solving 1024 1024-equation systems. CRNBC is the CR solver optimized for no bank conflicts. We label the algorithm names in the figure. The nearer a switch point is to a labeled algorithm, the more proportion that algorithm takes in the hybrid solver.





Performance comparison between various GPU and CPU solvers for solving 1024 1024-equation systems. PCI-E: CPU-GPU PCI-Express data transfer. CRNBC: a CR solver optimized for no bank conflicts. MKL: a sequential tridiagonal routine from Intel's MKL library. MT-MKL: a multithreaded MKL solver.