Swept rule domain decomposition for 1D PDEs on heterogeneous compute clusters

Daniel J. Magee[1] and Kyle E. Niemeyer[2]

[1, 2]School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University

mageed@oregonstate.edu and kyle.niemeyer@oregonstate.edu

Introduction & Motivation

Accurate, fast solutions to Partial Differential Equations are essential to simulating physical phenomena with ever increasing fidelity. This greater resolution provides insight that drives technological development, but requires extensive computational resources. Our project applies a new technique (the swept rule) for apportioning these resources (domain decomposition) in PDE solutions to heterogeneous computing clusters.

Recently, high-performance computing clusters, such as ONRL's Titan, have moved hybrid processing paradigm by incorporating accelerating processors like GPUs. The features of GPU architecture originally designed for high resolution and frequency displays, hundreds to thousands of simple, parallel cores, a transparent memory hierarchy, and generous on-chip memory capacity, make it a natural tool for scientific computing.

Figure 1: GPU used in this study: Tesla K40c

Figure 2: CPU used in this study: Intel Xeon 2630-E5

Performance results

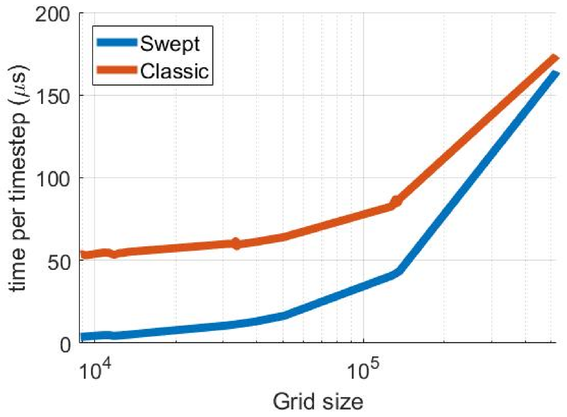


Figure 3: Best execution times per timestep for the heat equation using swept and classic decompisition

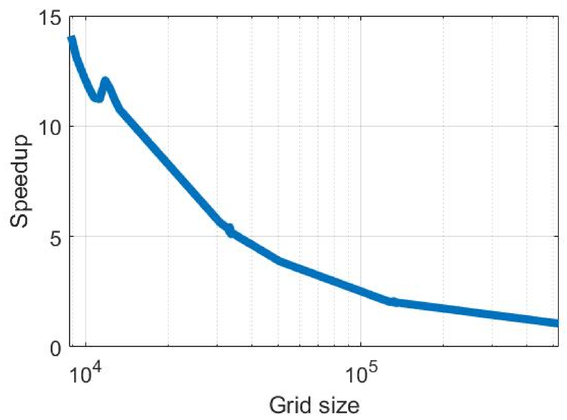


Figure 4: Speedup, t_c/t_s, of swept rule compared to classic decomposition.

These results are taken from an experiment which varies the node width, GPU affinity (the ratio of the grid partition size on the GPU to a single CPU process), and total grid size using 8 MPI processes, one for each CPU core, one of which controls and communicates with the GPU. Figures 3 and 4 above compare the best times for the respective procedures at each grid size.

Conclusions

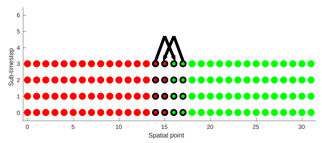
The results of this MPI+CUDA (GPU+CPU) implementation match previously observed trends in the GPU only implementation of the swept rule applied to the Heat equation:

- The performance benefits of the swept rule diminish with increasing grid size. As the cost of communication and kernel launch, which occur every sub-timestep in classic decomposition, become less significant as more computation is performed, the positive effect of the swept rule diminishes.
- The heat equation benefits from swept rule decomposition. This is because it permits a simple numerical scheme without path divergence that requires little memory overhead. Exploring more complicated equations that performed worse in the previous study, such as the Euler equations for inviscid flow, will provide a more general picture of performance benefit.
- The node width has a noticeable but minor impact on the performance of the swept rule. In general node widths of $64 - 512$ produce the best results. Additionally, GPU affinity did not substantially affect the results.

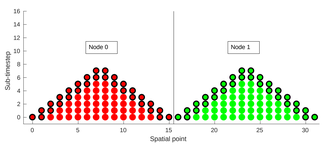
These results are limited by the environment where they were obtained, a workstation with one CPU and one GPU. Distributed, heterogeneous environments incur substantially greater communication costs and diminish computation load per processor which should bolster the swept rule's comparative efficacy.

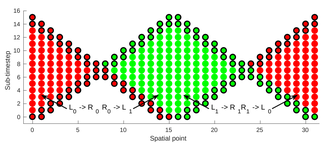
Method

Classic Decomposition



Swept Decomposition





As compute clusters become larger, more processors need to communicate to exchange results to continue a solution. The obvious answer to this challenge is to simply partition the spatial domain and pass the values at the edge of each node (partition) to the neighboring node. We refer to this decomposition method as Classic.

Since each sub-timestep, a step in a numerical method using a three-point spatial stencil, is dependent on the results of a previous sub-timestep, each process in a classic decomposition needs to communicate values at each edge spatial point to its neighbors after each sub-timestep. While the amount of data transferred in each communication is small, the latency cost of each communication event ($100\mu s - .5\mu s$) becomes significant over the many millions of timesteps needed to evaluate a meaningful equation with stability and precision.

The swept rule seeks to mitigate this cost by exploiting the domain of dependence. As shown in the figures to the left, the swept rule marches forward in time computing all grid points for which all the dependent values are known. When there are no grid points with known stencils, it passes the values needed to continue to a neighboring node. In this way, it communicates twice while advancing across a number of sub-timesteps equal to the node width.

The test case for the performance results presented here is the heat equation in one dimension,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \, . \tag{1}$$

Using a first-order, explicit, finite-difference approximation, forward differencing in time and central in space, we can solve for T at the new timestep,

$$T_i^{m+1} = \mathrm{Fo}(T_{i+1}^m + T_{i-1}^m) + (1 - 2\mathrm{Fo})T_i^m \, , \tag{2}$$

where i is the spatial node index and m is the time index corresponding to time t^m.

Works Cited

1. DJ Magee & KE Niemeyer. `HeterogeneousSwept1D`. GitHub: `https://github.com/Niemeyer-Research-Group/HeterogeneousSwept1D`. v1.0 (2017)
2. M Alhubail & Q Wang. "The Swept Rule for Breaking the Latency Barrier in Time Advancing PDEs." (2017) *Journal of Computational Physics*, in press. `doi:10.1016/j.jcp.2015.11.026`
3. DJ Magee & KE Niemeyer. "Accelerating solutions of PDEs with GPU-based swept time-space decomposition" (2017) In preparation, available via `arXiv:1705.03162`.

Working Repository:

Acknowledgements

This material is based upon work supported by NASA under award No. NNX15AU66A under the technical monitoring of Drs. Eric Nielsen and Mujeeb Malik. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40c GPU used for this research.

Further Information

Niemeyer Research Group (OSU)

`https://niemeyer-research-group.github.io`

Daniel's Site

`https://www.danieljmagee.com`