Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

GPU-BASED CALCULATION OF TRAJECTORY SIMILARITIES

Stefan Rybacki, Tobias Helms, Lars Moldenhauer, Adelinde Uhrmacher

Institute of Computer Science
University of Rostock
Albert-Einstein-Straße 22
18059 Rostock, GERMANY

ABSTRACT

Graphics Processing Units (GPUs) are more and more used for general purpose calculations. In the area of modeling and simulation, GPU's calculations are typically associated with executing specific simulation models. Besides this application, we propose the use of GPUs not only in the context of model execution but also to analyze simulation results, e.g., to compute the similarity of simulation trajectories. We present initial evaluation results from using the GPU for such applications and discuss opportunities, challenges, and pitfalls. We conclude with further possibilities as well as some future directions for leveraging the GPU for different analyzing tasks in the field of modeling and simulation.

1 INTRODUCTION

Besides visualizing data, Graphics Processing Units (GPUs) are increasingly employed for general purpose computations. However, in the area of modeling and simulation, their application has been focused on executing simulation models, either completely or parts of it in combination with the CPU (Rybacki et al. 2009, Vouzis and Sahinidis 2011, Dematt and Prandi 2010, Richmond et al. 2010). Due to the peculiarities of the GPU, which favors single instructions and multiple data (SIMD) computations, discrete stepwise simulation approaches, e.g., cellular automata, lend themselves for this approach and it has been shown that the execution performance can increase significantly by using the GPU, whereas in the context of stochastic discrete event simulation, GPU calculations are still comparatively rare. In contrast to using GPUs only for model execution, we propose to exploit their power additionally to analyze simulation results. For instance, a full factorial parameter scan of a simulation model easily leads to thousands of time series, i.e., simulation trajectories. Interactive techniques exist to explore those trajectories, e.g., techniques which cluster trajectories according to their similarity enabling the user to determine interesting parameter combinations and sensitivities (Luboschik et al. 2014). However, to use such techniques the similarity between trajectories needs to be calculated beforehand. Due to huge numbers of similarities to be computed, this procedure can be time-consuming. The use of complex similarity measures, which is often needed (Luboschik et al. 2014), even aggravates the problem. Fortunately, the calculation of similarities between trajectories can easily be encoded as a SIMD computation. On the one hand, there is the similarity calculation (the single instruction) between two trajectories which on the other hand is carried out for multiple trajectory pairs (the multiple data).

2 RESULTS

Different similarity measures exist, e.g., the mean squared error (MSE), the Hausdorff distance, the Fréchet distance, and dynamic time warping (DTW). Some of them are more demanding calculation-wise than others and some are more suitable to be implemented in a GPU-based manner. As a straightforward approach, we adapted the MSE and developed a GPU-based implementation of this measure. However, we already consider the impact of different measures by using a plug-in based design which allows an easy exchange of different measures and algorithms (implementations). We started our work based on the similarity measure MSE and used the framework APARAPI (Frost 2014) to implement the algorithm. APARAPI is a framework which allows to develop GPU programs in Java. The MSE is computed as follows:

$$mse = \frac{1}{n}\sum_{i=1}^{n}(A_i - B_i)^2.$$

For instance, each squared difference can be computed in parallel on the GPU. For this, an instance of APARAPI's class `Kernel` must be created overriding its run method with the code to run on the GPU (squared distance). The run method is then executed in parallel simply by executing the kernel instance with aparapi.

By using this approach we achieved, particularly with many trajectories, a speed up of roughly one order of magnitude. Nevertheless, we faced also some challenges and problems, mainly caused by either the data bandwidth, the calculation of the averaged differences (due to its semi-parallel nature), or operating system specific configurations, e.g., the timeout detection and recovery for graphic cards. Future work includes the evaluation of other more complex similarity measures, e.g., DTW (Sart et al. 2010), also referring to the induced speed up on the GPU. Additionally, we will analyze the potential of the GPU within other areas of simulation result analysis, e.g., identifying

arbitrary behavior patterns in or over trajectories (Chang et al. 2012), like increasing, decreasing, oscillating. These patterns shall be specified using a domain specific language, which can be automatically interpreted and translated into code to be executed (if possible) on the GPU. Eventually, this approach shall be integrated into the statistical model checking of the modeling and simulation framework JAMES II (Peng et al. 2014).

REFERENCES

Chang, K.-W., B. Deka, W.-M. W. Hwu, and D. Roth. 2012. "Efficient Pattern-Based Time Series Classification on GPU". In *ICDM*.

Dematt, L., and D. Prandi. 2010. "GPU computing for systems biology". *Briefings in Bioinformatics* 11 (3): 323–333.

Frost, G 2014. "Aparapi in amd developer website". http://developer.amd.com/tools/heterogeneous-computing/aparapi/.

Luboschik, M., S. Rybacki, F. Haack, and H.-J. Schulz. 2014. "Supporting the integrated visual analysis of input parameters and simulation trajectories". *Computers & Graphics* 39 (0): 37 – 47.

Peng, D., R. Ewald, and A. M. Uhrmacher. 2014. "Towards Semantic Model Composition via Experiments". In *Proceedings of the 2014 Workshop on Principles of Advanced and Distributed Simulation*, 151–162.

Richmond, P., D. Walker, S. Coakley, and D. Romano. 2010. "High performance cellular level agent-based simulation with FLAME for the GPU". *Briefings in Bioinformatics* 11 (3): 334–347.

Rybacki, S., J. Himmelspach, and A. M. Uhrmacher. 2009. "Experiments with single core, multi core, and GPU based computation of cellular automata". In *First International Conference on Advances in System Simulation*, 62–67. Piscataway, New Jersey: The Institute of Electrical and Electronics Engineers, Inc.

Sart, D., A. Mueen, W. Najjar, E. Keogh, and V. Niennattrakul. 2010, Dec. "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs". In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, 1001–1006.

Vouzis, P. D., and N. V. Sahinidis. 2011. "GPU-BLAST: using graphics processors to accelerate protein sequence alignment". *Bioinformatics* 27 (2): 182–188.