BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–2

STOCHSIMGPU: Parallel stochastic simulation for the Systems Biology Toolbox 2 for MATLAB

Guido Klingbeil [1,*], Radek Erban [2], Mike Giles [3] and Philip K. Maini [1,4]

[1]Centre for Mathematical Biology, Mathematical Institute, [2]Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, [3] Oxford-Man Institute of Quantitative Finance, and the Mathematical Institute, University of Oxford, Oxford OX1 3LB
[4] Oxford Centre for Integrative Systems Biology, University of Oxford, Oxford OX1 3QU

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT
Motivation: The importance of stochasticity in biological systems is becoming increasingly recognised and the computational cost of biologically realistic stochastic simulations urgently requires development of efficient software. We present a new software tool STOCHSIMGPU which exploits graphics processing units (GPUs) for parallel stochastic simulations of biological/chemical reaction systems and show that significant gains in efficiency can be made. It is integrated into MATLAB and works with the Systems Biology Toolbox 2 (SBTOOLBOX2) for MATLAB.

Results: The GPU-based parallel implementation of the Gillespie stochastic simulation algorithm (SSA), the logarithmic direct method (LDM), and the next reaction method (NRM) is approximately 85 times faster than the sequential implementation of the NRM on a central processing unit (CPU). Using our software does not require any changes to the user's models, since it acts as a direct replacement of the stochastic simulation software of the SBTOOLBOX2.

Availability: The software is open source under the GPL v3 and available at http://people.maths.ox.ac.uk/~klingbeil/STOCHSIMGPU. The website also contains supplementary information.

Contact: klingbeil@maths.ox.ac.uk

1 INTRODUCTION

Decision making in biological systems may depend on single molecular reaction events making it necessary to develop, and carefully investigate, stochastic simulations of such events. A classic example is the pathway bifurcation in λ-phage infected in *E. coli* cells (Arkin *et al.*, 1998). Three exact stochastic simulation algorithms of chemical reaction systems are commonly used in Systems Biology: (i) the stochastic simulation algorithm (SSA) of Gillespie (1977), the efficient and exact reformulations (ii) next reaction method (NRM) of Gibson and Bruck (2000) and (iii) the logarithmic direct method (LDM) of Li and Petzold (2006).

To accurately approximate the statistical distribution of the molecular populations at any given point in time large ensembles of realisations are needed emphasising the need for efficient computation. Unlike existing efficient simulation tools like Lis *et al.* (2009), we use NVIDIA CUDA to transform GPUs of modern PCs into massively parallel co-processors. CUDA is supported by many of NVIDIA's current GPUs and is available in many off-the-shelf computers[1]. We present an implementation of these algorithms which computes ensembles of many realisations approximately 85 times faster on a GPU than on a CPU assuming no specialised knowledge about GPUs by the user.

2 APPROACH

STOCHSIMGPU is a direct replacement of the stochastic simulation implementation provided by the SBTOOLBOX2 for MATLAB by Schmidt and Jirstrand (2006) hiding the technical details and focusing on user-friendliness. It is tightly integrated and directly usable within MATLAB. The user benefits without any changes to their code from the efficient computations on the GPU. The software simulates ensembles of many independent realisations of stochastic simulations of chemical reaction systems in parallel using the three exact algorithms SSA, NRM and LDM. The reaction systems have to be based on the law of mass action. The sampled realisations are used to compute averages and histograms of the molecular populations across the realisations on the GPU.

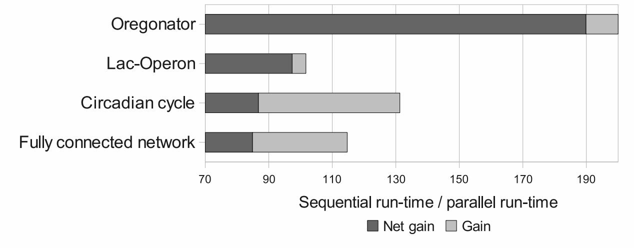
A CUDA enabled GPU consists of a set of streaming multiprocessors (SMs). These contain 8 single precision and one double precision floating point processor cores and a pool of fast on-chip shared memory (Lindholm *et al.*, 2008). This massively parallel design makes GPUs especially well suited for problems where the same set of instructions can be applied to several data sets simultaneously like the parallel stochastic simulation of large realisation ensembles.

STOCHSIMGPU computes in a task parallel approach ensembles of many independent realisations of stochastic simulations. The maximum number of realisations depend on the GPU used and the reaction system simulated. Its features include:

- Three exact simulation algorithms, SSA, NRM, and LDM,
- Integration into MATLAB requiring no special GPU knowledge,

[*] to whom correspondence should be addressed

[1] Beginning with the GeForce 8 series. A list of supported GPUs is available at: http://www.nvidia.com/object/cuda_learn_products.html.



System	Reactions	Species	Grid layout
Oregonator	5	8	79 × 128
Lac-Operon	16	11	157 × 128
Circadian cycle	19	9	79 × 128
Fully connected network	30	6	209 × 96

Fig. 1. Speed-up of the parallel GPU-based implementations compared to sequential implementations on the CPU. The sequential NRM is chosen as a reference, since it delivers significantly better performance than the SSA on a CPU. Gain is the speed-up of the fastest parallel algorithm over the sequential implementation. Net gain denotes the speed-up of the fastest parallel implementation over the NRM on a CPU.

- Histogram computation across all realisations of the stationary process or at any number of user defined time points and at the steady state,
- Computation of the mean of the realisations,
- Sampling of the molecular populations at equidistant time points (or non-equidistant in time whenever the n-th reaction event occurs).

The available on-board memory of the graphics board limits the maximum number of samples of the molecular populations to be stored. The number of samples times the number of species times the number of realisation has to fit into the on-board memory. To maximise performance, the current molecular populations, propensity functions, and the NRM's indexed priority queue are stored in the shared memory. The size of on-chip shared memory limits the size of the reaction system computable[2]. Furthermore, the reaction kinetics is limited to the law of mass action.

3 PERFORMANCE

We compared the speed-up, this is the ratio of sequential run-time on a CPU to parallel run-time on the GPU, of our parallel implementation in two ways. The speed-up (gain) of the parallel over the sequential implementation for each algorithm (SSA, NRM and LDM), as well as the speed-up (net gain) compared to our sequential NRM implementation which we found to be the most efficient sequential algorithm. At a conservative estimate, the parallel stochastic simulation using GPUs is approximately 85 times faster than the sequential implementation on a CPU. Figure 1 shows the speed-up for four example systems of which two are biologically meaningful (Klingbeil *et al.*, 2010). The speed-up shown is the net gain a user can expect when simulating biologically meaningful chemical reaction systems.

4 DISCUSSION

We developed a GPU-based software package for efficient stochastic simulation of homogeneous (well-mixed) chemical systems. Parallel computing on GPUs also has a potential to accelerate more detailed models of intracellular processes. For example, spatially distributed (reaction-diffusion) systems are sometimes modelled using compartment-based approaches Erban and Chapman (2009) which enable the use of the Gillespie SSA to simulate the time evolution of the system. In particular, STOCHSIMGPU is directly applicable to these models. Since STOCHSIMGPU is optimised for non-spatial models, there are limits on the size of the reaction-diffusion system. If the reaction-diffusion system is discretised into many compartments, a different software package should be used Hattne *et al.* (2005).

Requirements: NVIDIA GeForce 8800 GPU or later, NVIDIA CUDA 2.2 toolkit or later, MATLAB 7.7.0 (R2008b) or later and the SBTOOLBOX2 (http://www.sbtoolbox2.org).

ACKNOWLEDGEMENT

GK was supported by the Systems Biology Doctoral Training Centre and the Engineering and Physical Sciences Research Council. This publication was based on work supported in part by Award No KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST). The research leading to these results has received funding from the European Research Council under the *European Community's* Seventh Framework Programme *(FP7/2007-2013)* / ERC *grant agreement* No. 239870. RE would also like to thank Somerville College, University of Oxford for a Fulford Junior Research Fellowship. MG was supported in part by the Oxford-Man Institute of Quantitative Finance, and by the UK Engineering and Physical Sciences Research Council under research grant EP/G00210X/. PKM was partially supported by a Royal Society Wolfson Research Merit Award.

REFERENCES

Arkin, A., Ross, J., and McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, **149**(4), 1633–1648.

Erban, R. and Chapman, S. (2009). Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions. *Physical Biology*, **6**(4), 046001.

Gibson, M. and Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, **104**, 1876–1889.

Gillespie, D. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, **81**(25), 2340–2361.

Hattne, J., Fange, D., and Elf, J. (2005). Stochastic reaction-diffusion simulation with mesord. *Bioinformatics*, **21**(12), 2923–2924.

Klingbeil, G., Erban, R., Giles, M., and Maini, P. K. (2010). Fat vs. thin threading approach on GPUs: application to stochastic simulation of chemical reactions. *IEEE Transactions on Parallel and Distributed Systems*. Submitted.

Li, H. and Petzold, L. (2006). Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. Technical report, Department of Computer Science, University of California Santa Barbara.

Lindholm, E., Nickolls, J., Oberman, S., and Montrym, J. (2008). Nvidia tesla: A unified graphics and computing architecture. *IEEE Computer Society Hot Chips*, (19), 39–45.

Lis, M., Artyomov, M. N., Devadas, S., and Chakraborty, A. K. (2009). Efficient stochastic simulation of reaction-diffusion processes via direct compilation. *Bioinformatics*, **25**(17), 2289–2291.

Schmidt, H. and Jirstrand, M. (2006). Systems Biology Toolbox for MATLAB: a computational platform for research in Systems Biology. *Bioinformatics*, **22**(4), 514–515. http://www.sbtoolbox2.org.

[2] The Tesla architecture provides 16 kB, the Fermi architecture up to 48 kB of shared memory. See supplemental online material for details.