On GPU-Oriented P systems

M.A. Martínez-del-Amor, D. Orellana-Martín, A. Riscos-Núñez, M.J. Pérez-Jiménez

Dept. of Computer Science and Artificial Intelligence
Universidad de Sevilla
Seville, Spain
*mdelamor@us.es, dorellana@us.es, ariscosn@us.es, marper@*us.es

INVITED TALK EXTENDED ABSTRACT

P systems are massively parallel computing devices studied under Membrane Computing, and that are inspired by the structure and functioning of living cells [8]. Although many types and variants have been defined, the main common ingredients are a compartmentalized structure given by membranes, and multiset of objects within each region that evolve by a pre-defined set of rules. Simulating P systems is of huge importance for developing validation and verification tools [9]. In order to accelerate these simulations, it is possible to leverage High Performance Computing technologies to handle their massively parallel nature. Implementing P system parallelism is a hard task on some platforms, but we have shown that GPUs present a high-level of parallelism that can be employed successfully for this task [4][5]. In this short abstract, we discuss the need of defining P systems with ingredients that ease the design of parallel simulators on GPUs.

A. Nature of P system parallelism

Membrane system devices are inherently parallel in the sense that objects within membranes evolve according to the defined rules in parallel, and this holds to all membranes simultaneously too [8]. It is possible also to transfer objects between membranes, and the membrane structure can evolve. However, the following key aspects are inherent to P systems:

- Synchronization: a P system computation consists of transitions steps governed by a global clock. The state of a P system at a given moment is called configuration.
- Maximal parallelism: objects that can evolve in a transition step must do it. That is, after applying all rules, there are no remaining applicable rules.
- Non-determinism: the computation of a P system is a tree where there might be many computation paths. It is possible that different multisets of rules can be applied to a configuration.

Many P system models have been defined in the literature, by building up models with different ingredients: electrical charges associated to membranes, promoters, proteins, symport/antiport rules, division rules, dissolution rules, cooperation, etc. Moreover, three main flavors have been introduced: cell-like P systems (where the membrane structure is a tree), tissue-like P systems (a directed graph), and spiking-neural-like P systems (a directed graph).

B. GPU simulators for P systems

In previous work, we have shown that the simulation of P systems is bounded by memory and memory bandwidth [5][7]. Indeed, selecting and executing rules requires not much computation, but instead memory storage and transfers. This issue has constrained the performance of simulators. In order to get some acceleration, the bio-inspired parallelism of P systems can be leveraged. Today, we can take advantage of high performance computing platforms, so that we can map the massive parallelism of membrane systems into parallel architectures. In this concern, GPUs [4] have been shown to be a good platform, for the following reasons [5]:

(1) the shared memory system helps to synchronize the simulation;

(2) the double parallelism present on a GPU can be used to directly map the double parallelism of P systems;

(3) the fast memory system of GPUs reduces the main bottleneck of simulating a P system.

Table I summarizes some simulators so far implemented on GPUs. The first column indicates the codename given to the project. The second column shows the P system model that is simulated, and the coverage (G for generic, so for the whole type, and S for specific, so for just one family for a certain problem). The third column shows the peak speedup achieved in the experiments (T for stressing tests, R for real examples). The last column says the GPU employed.

TABLE I. DEVELOPED P SYSTEM SIMULATORS ON GPUS

Simulator Codename	P system model and coverage	Peak speedup	GPU tested
PCUDA [1]	(G) Active membranes	7x (T) 1.67x (R)	C1060
PCUDASAT [3]	(S) Active membranes	63x (R)	C1060
TSPCUDASAT [6]	(S) Tissue w/ cell division	10x (R)	C1060
ABCDGPU [7]	(G) Population Dynamics	18.1x (T) 5x (R)	K40
ENPS-GPU	(G) Enzimatic Numerical	10x (T)	GTX460M
CuSNP [1]	(G) Spiking Neural	50x (R)	GTX750

G= Generic, S=Specific, T=Stress testing, R=Real examples.

C. Performance characterization for the GPU

When analyzing the development and design of P system simulators on the GPU, it is possible to identify the key features that can dramatically affect the performance [5]:

- Object density: if it is not possible to previously estimate an upper bound for the number of different objects that might appear in a membrane, the design has to allocate space for the whole set of possible objects (alphabet). Normally, this leads to sparse arrays, and so, many idle threads.
- Rule intensity: related to the object density, not all defined rules are applied at every transition steps, so threads need time to explore which ones are applicable.
- Rule competition: when rules can have several objects in the left-hand side, it might happen, that a set of rules compete for a same object. In this situation, specific and very elaborated algorithms, such as DCBA, have to be defined and spend much time to decide which rules use the competing objects.
- Membrane synchronization: when assigning GPU thread blocks to membranes, and the membranes exchange objects at every step, the synchronization must be done after every transition. If we know beforehand that there is not much object exchanging, it would be possible to process membranes in parallel without much synchronization.

D. Ingredients for GP systems

Bearing in mind the performance characterization discussed above, it is possible to analyze which P system ingredients can help to the design of simulators. In this sense, the aim of a recent accepted project (MABICAP) is to define a new P system model that contains features that ease the simulation on GPU. We named these systems GPU-oriented P systems (GP systems for short).

The first ingredient to be identified was the electrical charges [6]. By implementing two specific solutions to SAT problem, using a cell-like and a tissue-like approach, we found that having charges associated to membranes helps to codify information, and therefore, save on object definitions. The less objects are defined, the lower object density gets.

Another key feature we found recently is that having rules with minimal production (i.e. up to one object in the right-hand side) can help to define an upper bound in situations where there is limited membrane communication: the maximum number different objects to appear in a membrane is less or equal to the size of the input multiset.

Other ingredients are to be explored, such as minimal cooperation (up to two objects in the left-hand side of rules), asynchronous computation (membranes evolve independently), and minimal parallelism (not all objects must evolve). However, the main restriction to impose here is that the constructed model must be powerful enough in the theoretical meaning. Finally, we plan to define flavors of GP systems that can be employed for applications such as real ecosystem modelling.

Keywords - P systems; Simulation; Parallelism; GPU;

ACKNOWLEDGMENT

The authors acknowledge the support from the research project TIN2017-89842-P, cofinanced by “Ministerio de Economía, Industria y Competitividad” of Spain, through the “Agencia Estatal de Investigación”, and by FEDER funds of the European Union.

BIOGRAPHIES

M.Á. MARTÍNEZ-DEL-AMOR received his Ph.D. degree in 2013 from the University of Seville, Seville, Spain. Today, he is an assistant professor at the same university, and a member of the Research Group on Natural Computing.

D. ORELLANA-MARTÍN received his master’s degree in 2016 from the University of Seville. Today, he is developing his Ph.D. at the Research Group on Natural Computing at the University of Seville.

A. RISCOS-NÚÑEZ received his Ph.D. degree from the University of Seville in 2004. He is currently a Professor at the University of Seville, and a member of the Research Group on Natural Computing.

M.J. PÉREZ-JIMÉNEZ received his Ph.D. degree from the University of Seville in 1992. Currently, he is a numerary member of the Academia Europaea (The Academy of Europe), a full Professor at the University of Seville, and head of the Research Group on Natural Computing.

REFERENCES

[1] J.P. Carandang, J.M.B. Villaflores, F.G.C. Cabarle, H.N. Adorna, M.Á. Martínez-del-Amor. “CuSNP: Spiking Neural P Systems Simulators in CUDA”. *Romanian Journal of Information Science and Technology*, vol. 20, 1, pp. 57-70, 2017.

[2] J.M. Cecilia, J.M. García, G.D. Guerrero, M.A. Martínez-del-Amor, I. Pérez-Hurtado, M.J. Pérez-Jiménez. “Simulation of P systems with active membranes on CUDA”. *Briefings in Bioinformatics*, vol. 11, 3, pp. 313-322, 2010.

[3] J.M. Cecilia, J.M. García, G.D. Guerrero, M.A. Martínez-del-Amor, I. Pérez-Hurtado, M.J. Pérez-Jiménez. “Simulating a P system based efficient solution to SAT by using GPUs”. *Journal of Logic and Algebraic Programming*, vol. 79, pp. 317-325, 2010.

[4] D. Kirk, W. Hwu. PROGRAMMING MASSIVELY PARALLEL PROCESSORS: A HANDS ON APPROACH. Morgan Kauffman, 2010.

[5] M.A. Martínez-del-Amor, M. García-Quismondo, L.F. Macías-Ramos, L. Valencia-Cabrera, A. Riscos-Núñez, M.J. Pérez-Jiménez. “Simulating P systems on GPU devices: a survey”. *Fundamenta Informaticae*, vol. 136, 3, pp. 269-284, 2015.

[6] M.A. Martínez-del-Amor, J. Pérez-Carrasco, M.J. Pérez-Jiménez. “Characterizing the parallel simulation of P systems on the GPU”. *International Journal of Unconventional Computing*, vol. 9, 5-6, pp. 405-424, 2013.

[7] M.A. Martínez-del-Amor, I. Pérez-Hurtado, A. Gastalver-Rubio, A.C. Elster, M.J. Pérez-Jiménez. “Population Dynamics P systems on CUDA”. *Lecture Notes in Bioinformatics*, vol. 7605, pp. 247-266, 2012.

[8] Gh. Paun. “Computing with membranes”, *Journal of Computer and System Sciences*, vol. 61, 1, 108-143, 2000.

[9] L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, M.J. Pérez-Jiménez. “From Super-cells to Robotic Swarms: Two Decades of Evolution in the Simulation of P Systems”. *The International Membrane Computing Society Bulletin* (IMCS Bulletin), Number 4, December 2017, pp. 65-88.