Natural Computing with GPUs:

Accelerating Bioinspired Models with CUDA

David Orellana-Martín, Miguel Á. Martínez-del-Amor

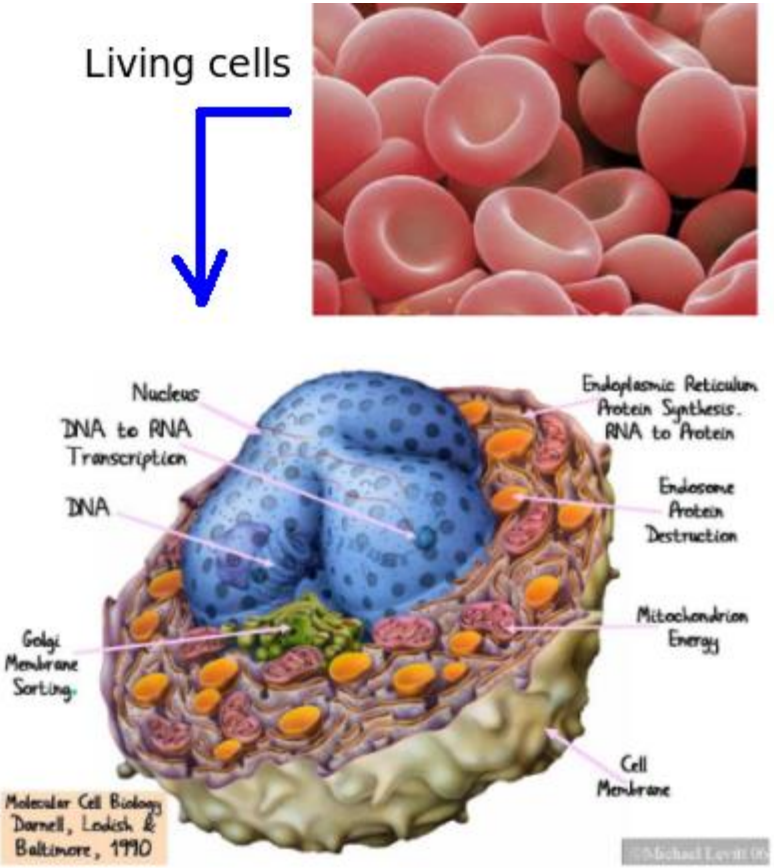
Research Group on Natural Computing

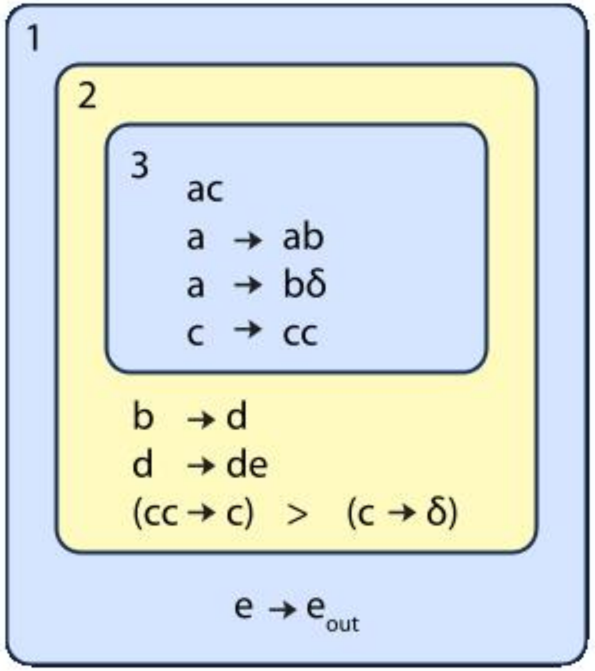
Universidad de Sevilla

Outline

- Introduction to Membrane Computing
- Group trajectory with CUDA
- PMCGPU simulators
- Challenges

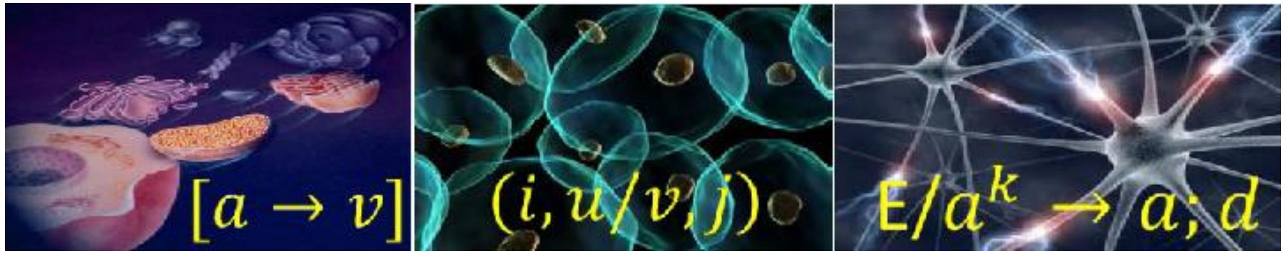
Membrane Computing

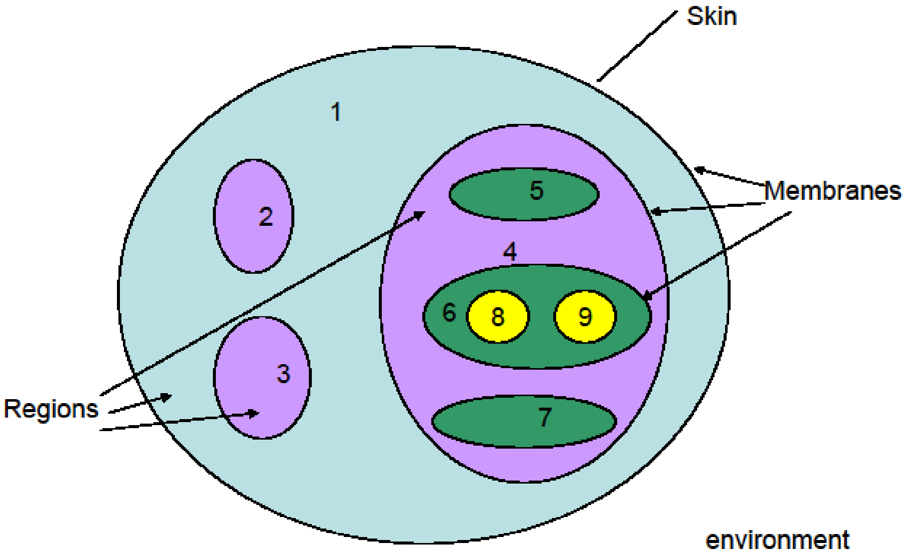




P system

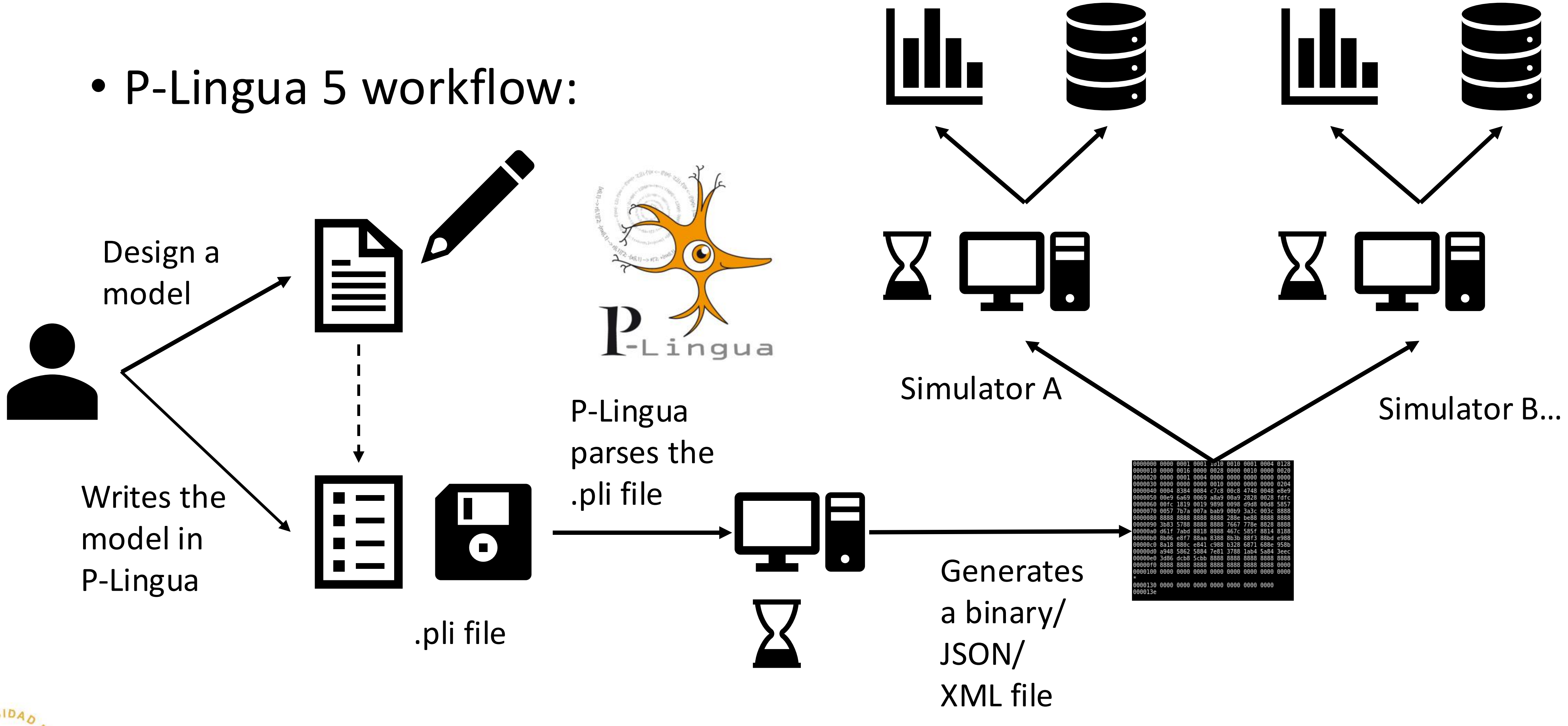
Membrane Computing





Membrane Computing

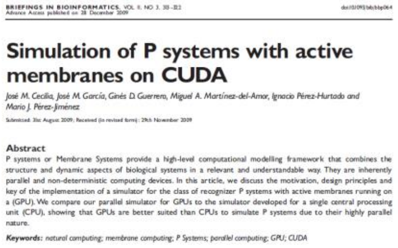
- P-Lingua 5 workflow:



Group trajectory with CUDA

(**2009**)

3xTesla C1060


BRIEFINGS IN BIOINFORMATICS. VOL 11. NO 3. 313–322
Advance Access published on 28 December 2009
doi:10.1093/bib/bbp064

Simulation of P systems with active membranes on CUDA

José M. Cecilia, José M. García, Ginés D. Guerrero, Miguel A. Martínez-del-Amor, Ignacio Pérez-Hurtado and Mario J. Pérez-Jiménez

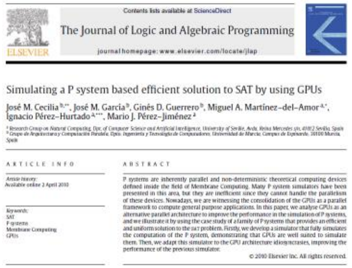
Submitted: 31st August 2009; Received (in revised form): 29th November 2009

Abstract
P systems or Membrane Systems provide a high-level computational modelling framework that combines the structure and dynamic aspects of biological systems in a relevant and understandable way. They are inherently parallel and non-deterministic computing devices. In this article, we discuss the motivation, design principles and key of the implementation of a simulator for the class of recognizer P systems with active membranes running on a (GPU). We compare our parallel simulator for GPUs to the simulator developed for a single central processing unit (CPU), showing that GPUs are better suited than CPUs to simulate P systems due to their highly parallel nature.

Keywords: *natural computing; membrane computing; P Systems; parallel computing; GPU; CUDA*

(**2009**)
Simulating P systems with active membranes

(**2010**)
Simulating P systems with active membranes solving SAT


Contents lists available at ScienceDirect
The Journal of Logic and Algebraic Programming
journal homepage: www.elsevier.com/locate/jlap

Simulating a P system based efficient solution to SAT by using GPUs

José M. Cecilia[b,*], José M. García[b], Ginés D. Guerrero[b], Miguel A. Martínez-del-Amor[a,*], Ignacio Pérez-Hurtado[a,**], Mario J. Pérez-Jiménez[a]
[a] Research Group on Natural Computing, Dpt. of Computer Science and Artificial Intelligence, University of Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
[b] Grupo de Arquitectura y Computación Paralela, Dpto. Ingeniería y Tecnología de Computadores, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain

ARTICLE INFO
Article history:
Available online 2 April 2010
Keywords:
SAT
P systems
Membrane Computing
GPUs

ABSTRACT
P systems are inherently parallel and non-deterministic theoretical computing devices defined inside the field of Membrane Computing. Many P system simulators have been presented in this area, but they are inefficient since they cannot handle the parallelism of these devices. Nowadays, we are witnessing the consolidation of the GPUs as a parallel framework to compute general purpose applications. In this paper, we analyse GPUs as an alternative parallel architecture to improve the performance in the simulation of P systems, and we illustrate it by using the case study of a family of P systems that provides an efficient and uniform solution to the SAT problem. Firstly, we develop a simulator that fully simulates the computation of the P system, demonstrating that GPUs are well suited to simulate them. Then, we adapt this simulator to the GPU architecture idiosyncrasies, improving the performance of the previous simulator.

Matrix Representation of Spiking Neural P Systems

Xiangxiang Zeng[1], Henry Adorna[2], Miguel Ángel Martínez-del-Amor[3], Linqiang Pan[1,*], and Mario J. Pérez-Jiménez[3]

[1] Image Processing and Intelligent Control Key Laboratory of Education Ministry
Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan 430074, Hubei, China
xzeng@foxmail.com,
lqpan@mail.hust.edu.cn
[2] Department of Computer Science
(Algorithms and Complexity)
University of the Philippines
Diliman 1101 Quezon City, Philippines
hnadorna@dcs.upd.edu.ph
[3] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
{mdelamor,marper}@us.es

Abstract. Spiking neural P systems (SN P systems, for short) are a class of distributed parallel computing devices inspired from the way neurons communicate by means of spikes. In this work, a discrete structure representation of SN P systems with extended rules and without delay is proposed. Specifically, matrices are used to represent SN P systems. In order to represent the computations of SN P systems by matrices, configuration vectors are defined to monitor the number of spikes in each neuron at any given configuration; transition net gain vectors are also introduced to quantify the total amount of spikes consumed and produced after the chosen rules are applied. Nondeterminism of the systems is assured by a set of spiking transition vectors that could be used at any given time during the computation. With such matrix representation, it is quite convenient to determine the next configuration from a given configuration, since it involves only multiplication and addition of matrices after deciding the spiking transition vector.

(**2010**)

Matrix representation of spiking neural P systems

(China-Philippines)

(**2011-12**)
Simulating spiking neural P systems (Philippines)

Group trajectory with CUDA

(2011)
GeForce
GTX550Ti

(2012)
GeForce
GTX780Ti

(2012)
Simulating
Population
Dynamics
P systems

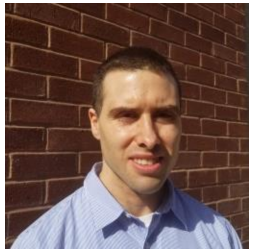
(2013)
PMCGPU
project

(2013)
Miguel Á.
Martínez
del Amor
PhD Thesis

Group trajectory with CUDA

(2014)
NVIDIA CUDA Research Center. K40 Donation

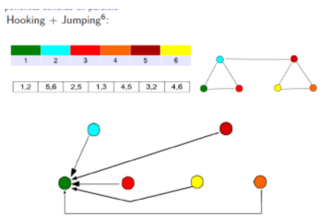
NVIDIA
GPU
RESEARCH CENTER



(2014)
Manuel García-Quismondo
PhD Thesis

(2015)
NVIDIA CUDA Research Center

NVIDIA
GPU
RESEARCH CENTER



(2018)
micro-DCBA for PDP system simulation

(2018)
NVIDIA DLI university ambassador

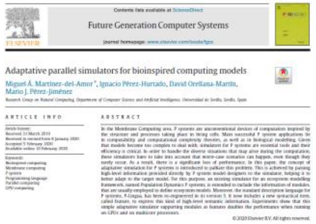
NVIDIA
DEEP LEARNING INSTITUTE
UNIVERSITY AMBASSADOR

Group trajectory with CUDA

(2019)
2xRTX2080

(2019)
Daniel
Campora
PhD Thesis

(2020)
Adaptative
simulation
of PDP
systems


Contents lists available at ScienceDirect
Future Generation Computer Systems
journal homepage: www.elsevier.com/locate/fgcs

Adaptative parallel simulators for bioinspired computing models

Miguel Á. Martínez-del-Amor *, Ignacio Pérez-Hurtado, David Orellana-Martín, Mario J. Pérez-Jiménez

ARTICLE INFO

ABSTRACT

(2021)
3xRTX3090

(2021)
NVIDIA
Hardware
grant.
A100


World Scientific

Sparse Spiking Neural-Like Membrane Systems on Graphics Processing Units

(2024)
Sparse
simulation
of Spiking
Neural P
systems

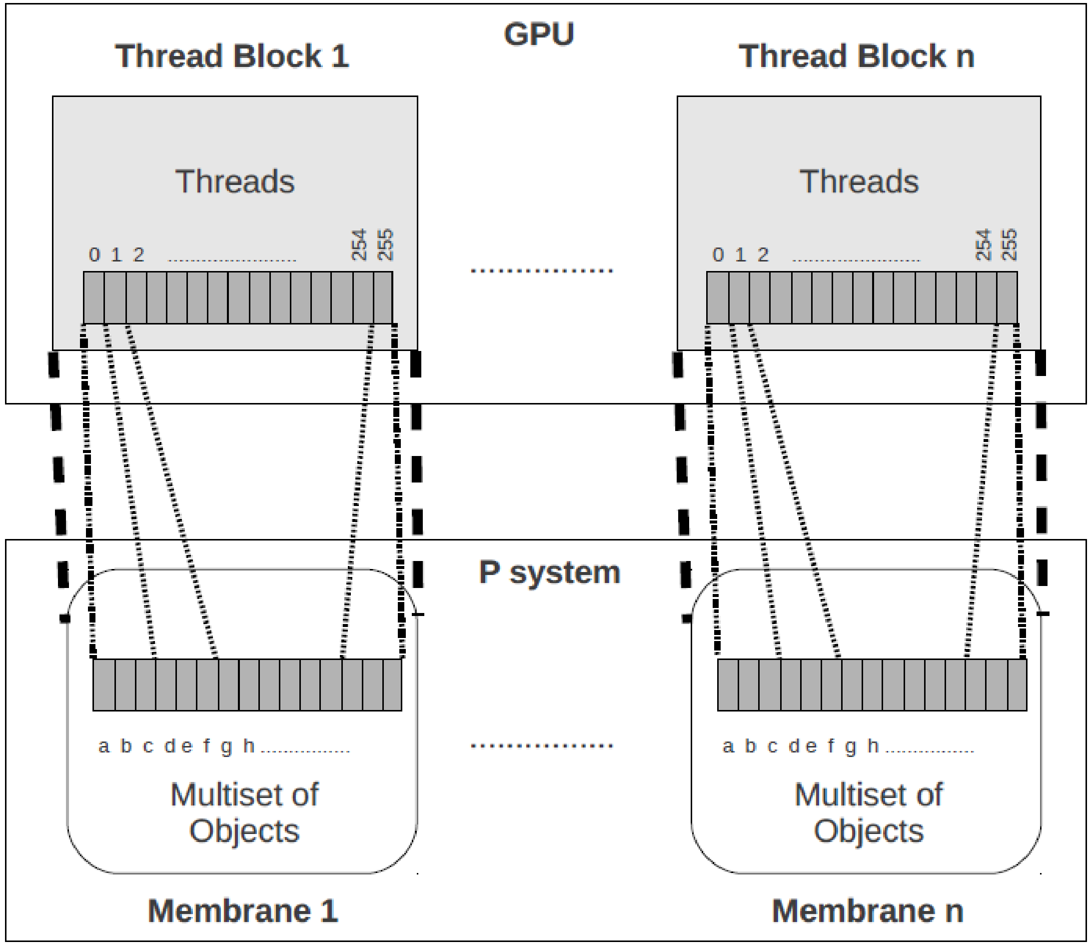
RGNC

NVIDIA University Ambassador

- Miguel Ángel Martínez-del-Amor, nominated in 2018
- Certificated instructor for:
  - Fundamentals of Accelerated Computing with Modern CUDA ++
  - Fundamentals of Deep Learning
  - Fundamentals of Accelerated Data Science
- Interested in:
  - Simulation of artificial neural networks, with emphasis on efficiency (sparsity & quantization)
  - Transferring concepts to spiking neural P systems and other natural computing models

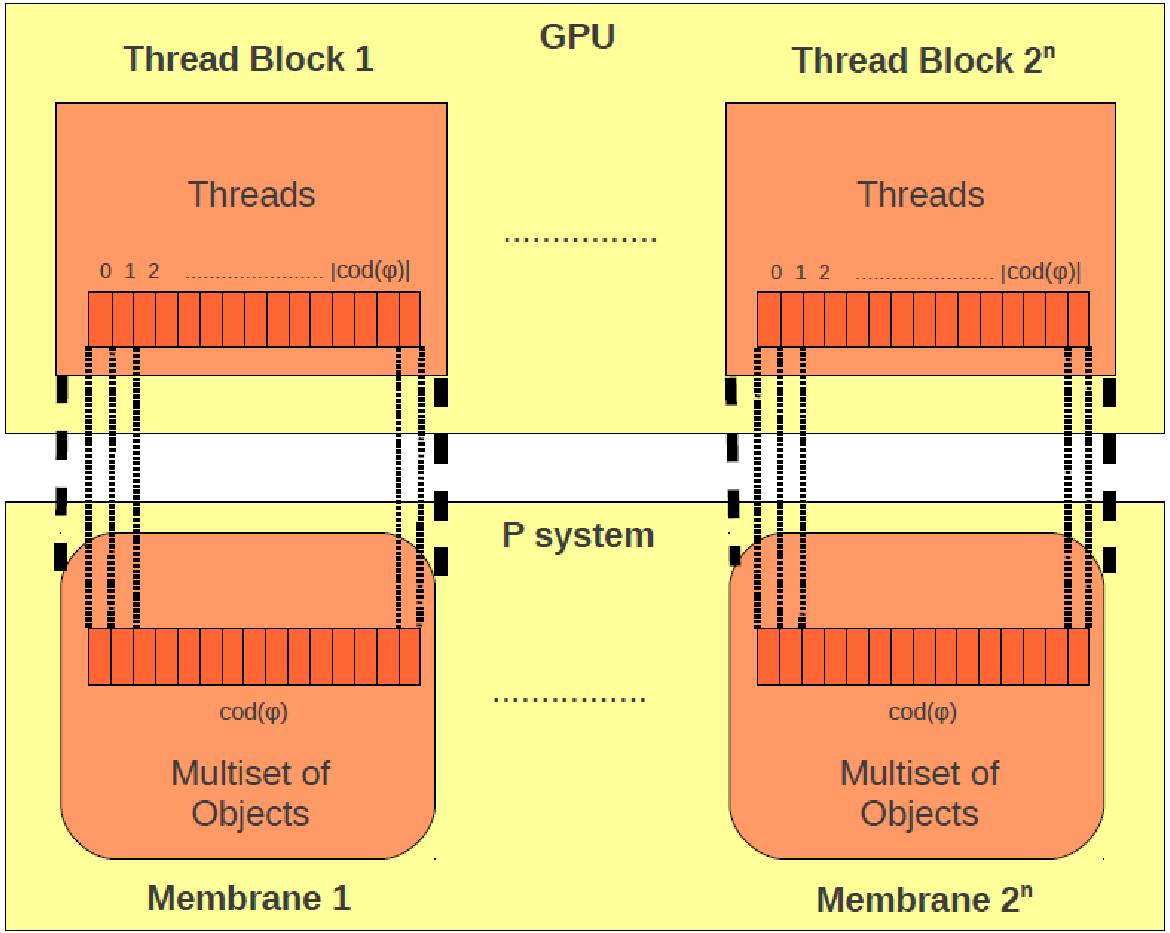
Parallelism schema of P systems

- P systems with active membranes



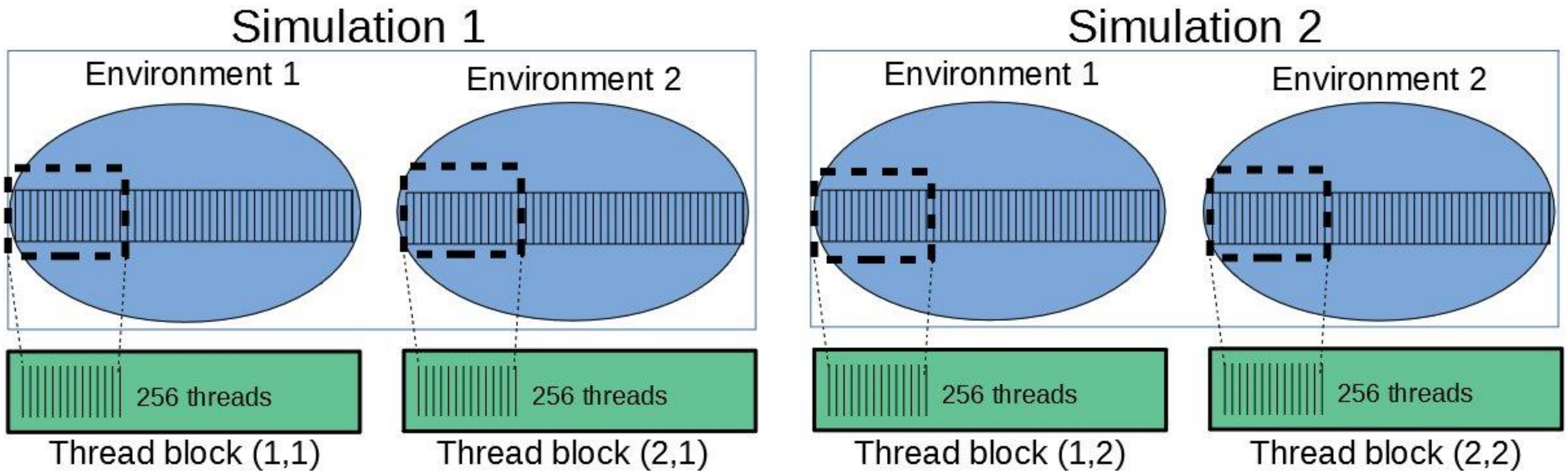
Parallelism schema of P systems

- Solution to SAT by P systems with active membranes



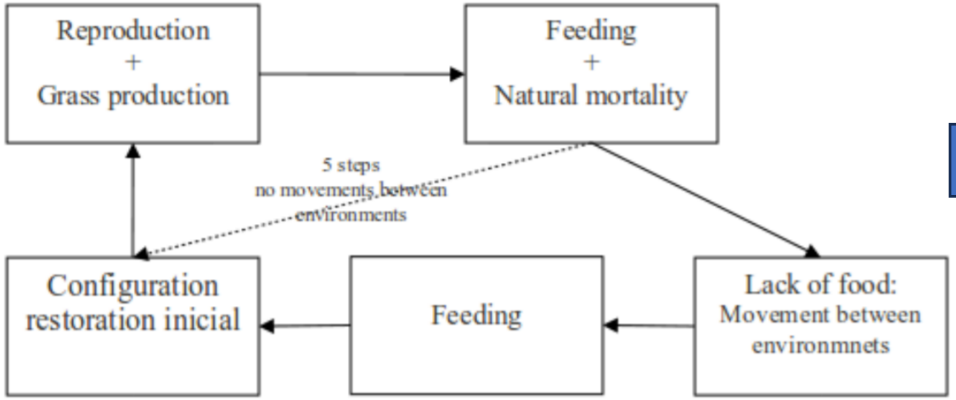
Parallelism schema of P systems

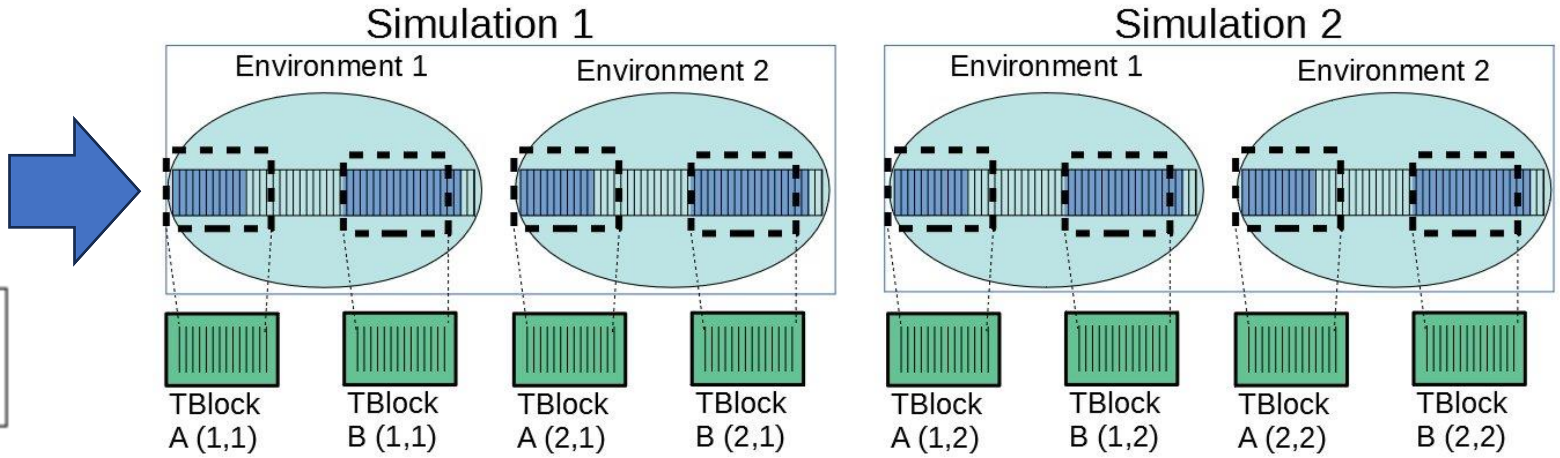
- Population Dynamic P (PDP) systems



Parallelism schema of P systems

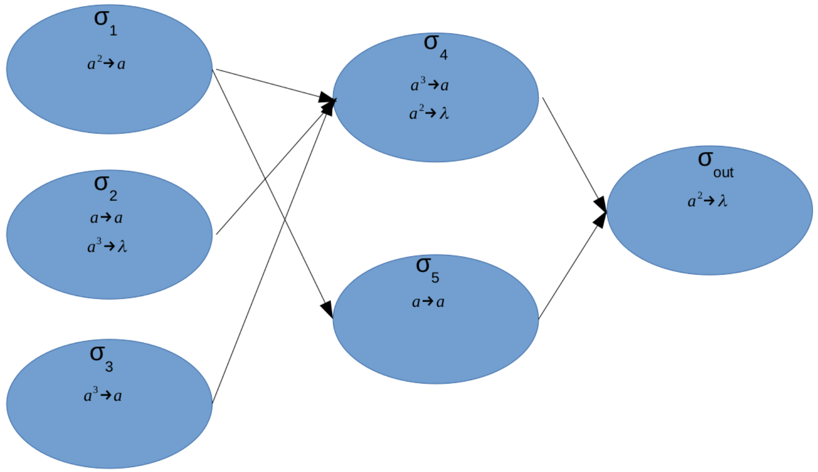
- Population Dynamic P (PDP) systems (Adaptative)

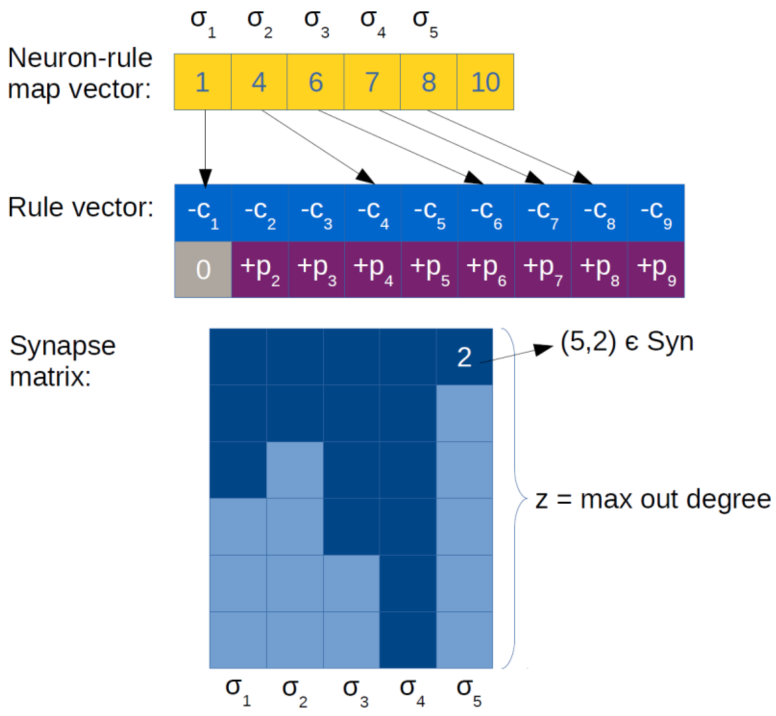




Parallelism schema of P systems

- Sparse representation of Spiking Neural P systems



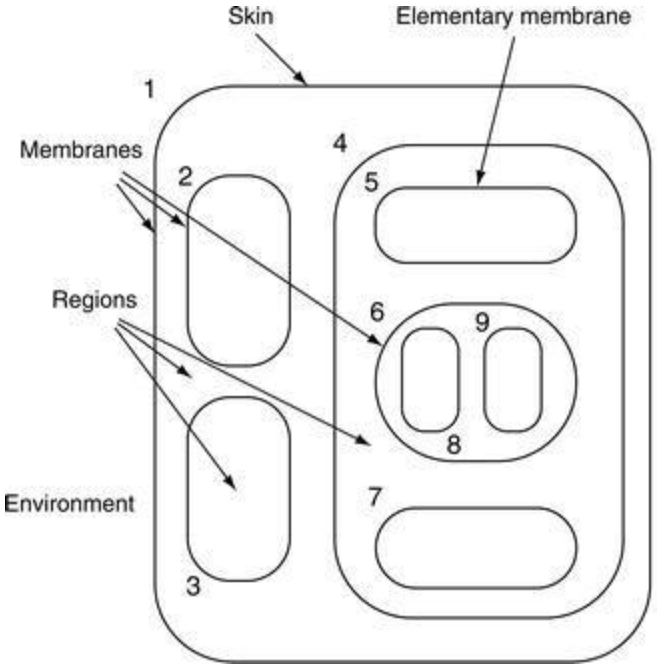


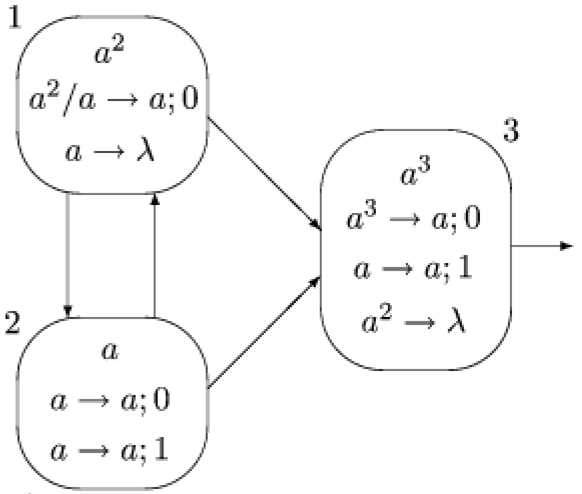
Summary of results

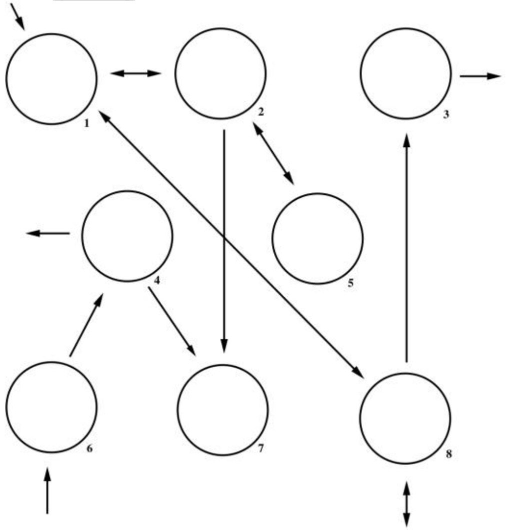
Simulator Codename	P system model and coverage	Peak speedup	GPU tested
PCUDA	(G) Active membranes	7x (T) 1.67x (R)	C1060
PCUDASAT	(S) Active membranes	63x (R)	C1060
TSPCUDASAT	(S) Tissue w/ cell division	10x (R)	C1060
ABCDGPU	(G) Population Dynamics	18.1x (T) 5x (R)	K40
ENPS-GPU	(G) Enzimatic Numerical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

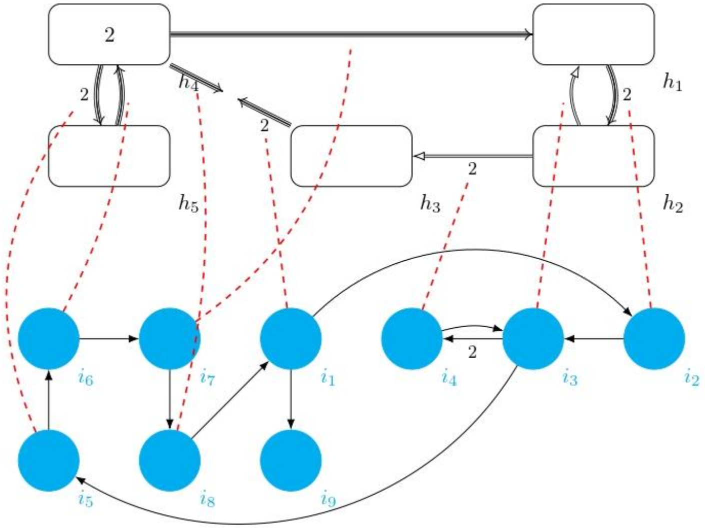
G= Generic, S=Specific, T=Stress testing, R=Real examples.

Current work on artificial intelligence

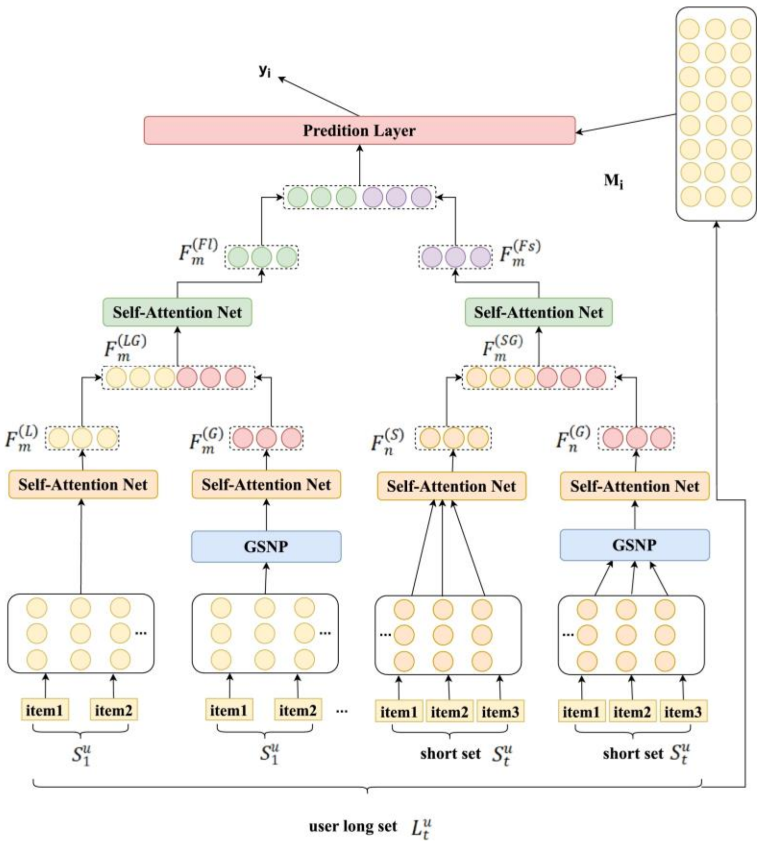


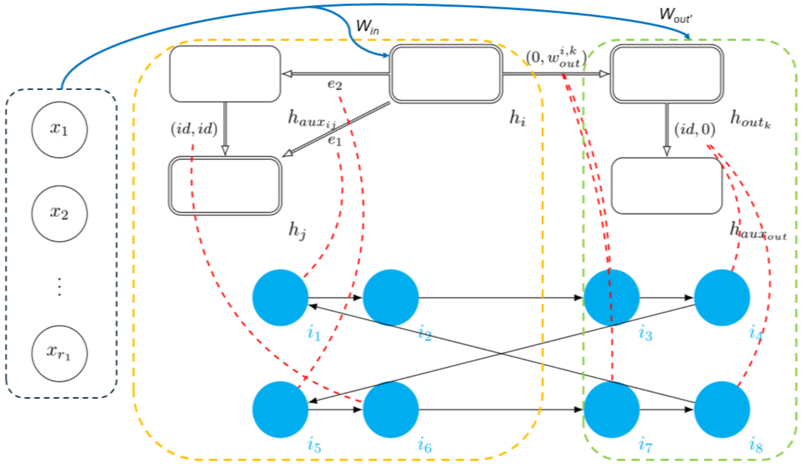


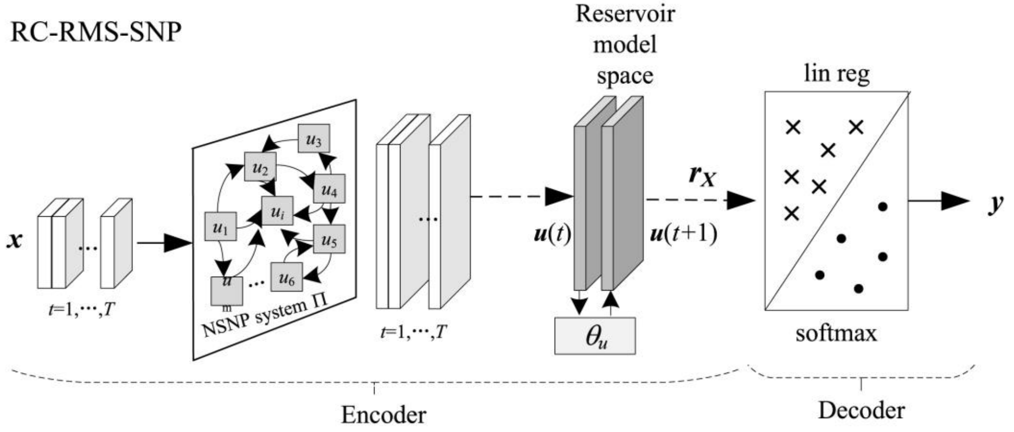




Current work on artificial intelligence

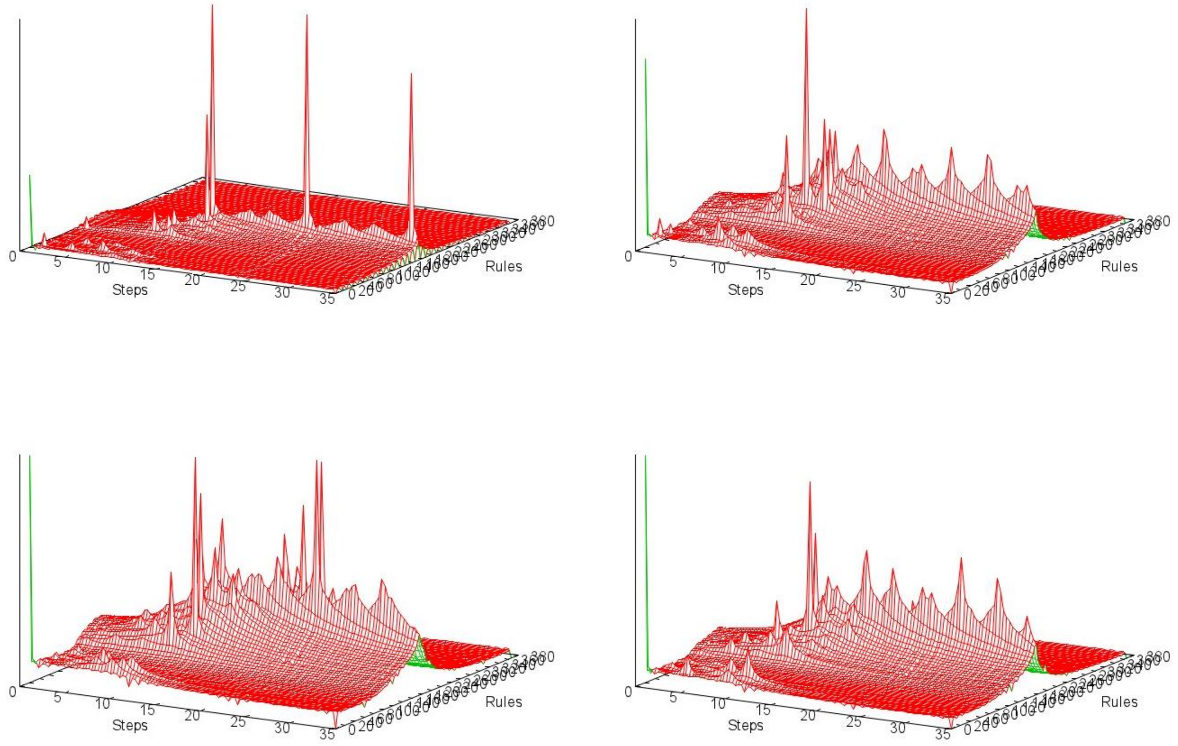


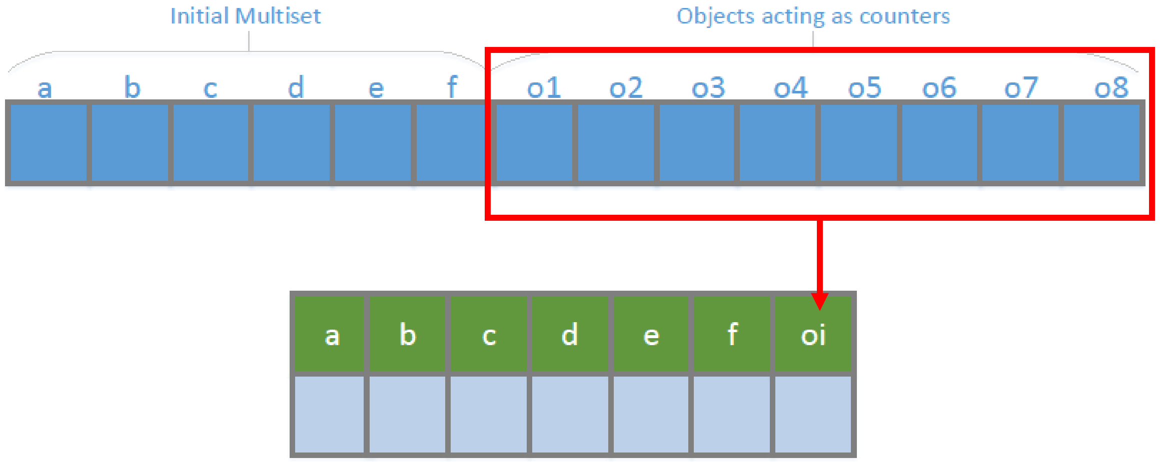




Challenges

- Sparse designs (different objects as counters)





Challenges

- Developing calibrator for model parameters
- Supporting new spiking neural P systems models used for deep learning
- Development from Python
- Multi-GPU models

Thank you very much!