A HIERARCHICAL ARCHITECTURE WITH PARALLEL COMUNICATION FOR IMPLEMENTING P SYSTEMS

Ginés Bravo, Luis Fernández , Fernando Arroyo, Juan A. Frutos

Abstract: Membrane systems are computational equivalent to Turing machines. However, its distributed and massively parallel nature obtain polynomial solutions opposite to traditional non-polynomial ones.

Nowadays, developed investigation for implementing membrane systems has not yet reached the massively parallel character of this computational model. Better published approaches have achieved a distributed architecture denominated "partially parallel evolution with partially parallel communication" where several membranes are allocated at each processor, proxys are used to communicate with membranes allocated at different processors and a policy of access control to the communications is mandatory. With these approaches, it is obtained processors parallelism in the application of evolution rules and in the internal communication among membranes allocated inside each processor. Even though, external communications share a common communication line, needed for the communication among membranes arranged in different processors, are sequential.

In this work, we present a new hierarchical architecture that reaches external communication parallelism among processors and substantially increases parallelization in the application of evolution rules and internal communications. Consequently, necessary time for each evolution step is reduced. With all of that, this new distributed hierarchical architecture is near to the massively parallel character required by the model.

Keywords: Architecture, hierarchy, P systems

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

Possibilities offered by Natural Computation and, specifically P-Systems, for solving NP-problems, have made researchers concentrate their work towards HW and SW implementations of this new computational model. Transition P systems were introduced by [Păun, 1998]. They were inspired by "basic features of biological membranes". One membrane defines a region where there are a series of chemical components (multisets) that are able to go through chemical reactions (evolution rules) to produce other elements. Inside the region delimited by a membrane can be placed other membranes defiining a complex hierarchical structure that can be represented as a tree. Generated products by Chemical reactions can remain in the same region or can go to another region crossing a membrane. As a result of a reaction, one membrane can be dissolved (its chemical elements are transferred to the container membrane) or can be inhibited (the membrane becomes impermeable and not let objects to pass through).

Membrane systems are dynamics because chemical reactions produce elements that go through membranes to travel to other regions and produce new reactions. This dynamic behaviour is possible to be sequenced in a series of evolution steps between one system configuration to another. These system configurations are determined by the membrane structure and multisets present inside membranes. In the formal Transition P systems model can be distinguished two phases in each evolution step: rules application and communication. In application rules phase, rules of a membrane are applied in parallel to the membrane multiset inside of it. Once application rules phase is finished, then it begins communication phase, where those generated multisets travel through membranes towards their destination in case it is another region. These systems carry out computations through transitions between two consecutive configurations, what turn them into a computational model with the same capabilities as Turing machines.

Power of this model lies in the fact that the evolution process is massively parallel in application rules phases as well as in communication phase. The challenge for researchers is to achieve hardware and/or software implementations of P systems respecting the massively parallelism in both phases. The goal of this work is to design a new hierarchical communication architecture that approaches the best possible way to the inherent characteristics of P systems: application and communication massively parallel.

This paper is structured in the following way: in the first place, the related works are enumerated analyzing the proposed architectures, next a communication hierarchical architecture model is introduced stating detailed analysis of the model. Afterward a comparative analysis with other architectures is presented and finally the conclusions obtained are presented.

Related Works

In [Syropoulos, 2003] and [Ciobanu, 2004] distributed P systems implementations are presented. They use respectively, the Java Remote Method Invocation (RMI) and the Message Passing Interface (MPI) over a PC cluster's Ethernet network. These authors don't make a detailed analysis about importance of time spent in communication phase respect total time of P system evolution, although Ciobanu declares that "the response time of the program has been acceptable. There are however executions that could take a rather long time due to unexpected network congestion" [Ciobanu, 2004].

In reply to this problem, [Tejedor, 2007] presents an analysis of an architecture named "partially parallel evolution with partially parallel communication". This architecture is based on the following pillars:

a. Membranes distribution. At each processor, *K* membranes are allocated that will evolve, at worst, sequentially. Where,

$$K = \frac{M}{P} \quad , K \geq 1 \tag{1}$$

and *M* is the total number of membranes of the P system and *P* is the number of processors of the distributed architecture. The physical interconnection of processors is made through a shared communication line. In this scenario, there are two sorts of communications,

- internal communications that are the ones that occur between membranes allocated at the same processor, and whose communication times is negligible because they are carried out using shared memory techniques.
- external communications that are those that occur between different processors because the membranes that needs to communicate are in different processors.

The benefit obtained is that the number of the external communications decreases.

b. Proxy for processor. Membranes that are in different processors do not communicate directly. They do by the means of proxys hosted at their respective processor. Proxys are used to communicate among processors. A proxy assumes communications among membranes of one processor towards the proxy of another one. In the same way, when information from other proxys is receive, it is redistributed to the membranes of the processor.

The benefit of using proxys in the communication among membranes instead of direct communication occurs because the communication protocols penalize the transmission of small packets due to protocol overhead. So, communicate N messages of L length is slower than one message of (S * L) length.

c. Tree topology of processors. The benefit obtained by using a tree topology in the processors interconnection is that the total number of external communications is minimized due to proxys only communicate with their direct ancestor and direct descendants. This way, total number of external communications is 2(P-1).

d. Token passing in the communications. In order to avoid collision and network congestion, it has been established and order in the communication. The idea is not to have more than one proxy trying to transmit at the same time.

The analysis of this distributed architecture leads to the following conclusions:

- This solution avoids communication collisions and reduces the number and length of the external communications.
- In this model, minimum time for an evolution step (T_{min}) is determined by the formula:

$$T_{\min} = 2\sqrt{2\ M\ T_{apl}\ T_{com}} - 2T_{com} \tag{2}$$

where, T_{apl} is the maximum time used by the slowest membrane in applying its rules, and T_{com} is the maximum time used by the slowest membrane for communication

- The number of processors (P_{opt}) that leads to the minimum time is:

$$P_{opt} = \sqrt{\frac{T_{apl}\, M}{2 T_{com}}} \tag{3}$$

Hierarchical Architecture

Previous model parallelize over P_{opt} processors the application of rules and the internal communications among membranes in the same processor. On the other hand, external communications, necessaries for the communication among membranes allocated at the same processor, are sequential. For that reason, we propose a variation that permits to parallelize, up to a certain degree, external communications among nodes. This way, time of an evolution step is reduced drastically and it will tend towards the massively parallel character of a P system.

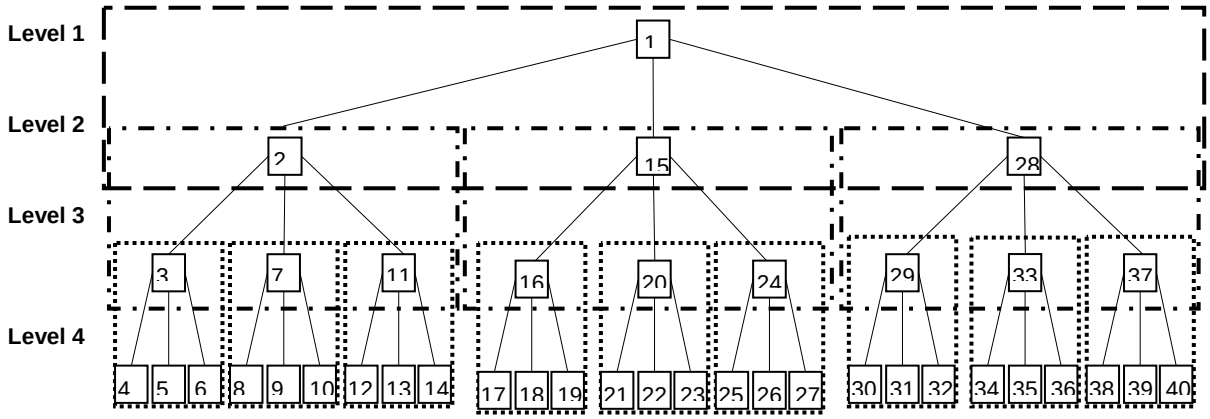

Figure 1. Hierarchical Architecture of 4 levels and amplitude equal to 3.

The new architecture consists of having at its distribute the processors in a hierarchical way, specifically, in a balanced tree of *N* levels depth and *A* processors in amplitude. For instance, figure 1 shows a balanced tree of *N* = 4 and *A* = 3.

For example of figure 1, when every node have applied its rules in parallel, external communications are carried out sequentially in each one of the 9 subtrees arranged between levels 3 and 4; hence, at every instant, as many external communications are carried out as subtrees exist between levels 3 and 4. Subsequently, external communications in each one of the three subtrees arranged between levels 2 and 3 are carried out sequentially; hence, at every instant, as many external communications are carried out as subtrees exist between levels 2 and 3. And finally, external communications in the subtree arranged between levels 1 and 2 are carried out sequentially.

From a logical point of view, each subtree requires a particular physical network to reach the parallelism of its external communications. This way, the processors of intermediate subtrees need 2 communication interfaces, one for the network of the subtree which is root, and another one for network of the subtree which is a leaf. On the other hand, only one interface is required for the processors in the extreme levels 1 and N because they are part of just one subtree. On the other hand, from a physical point of view, the number of logical networks can be reduced to one using Ethernet switches because they permit the separation of collision domains.

Figure 2 chronogram shows the parallelism in application times and in the external communications of the previous example.

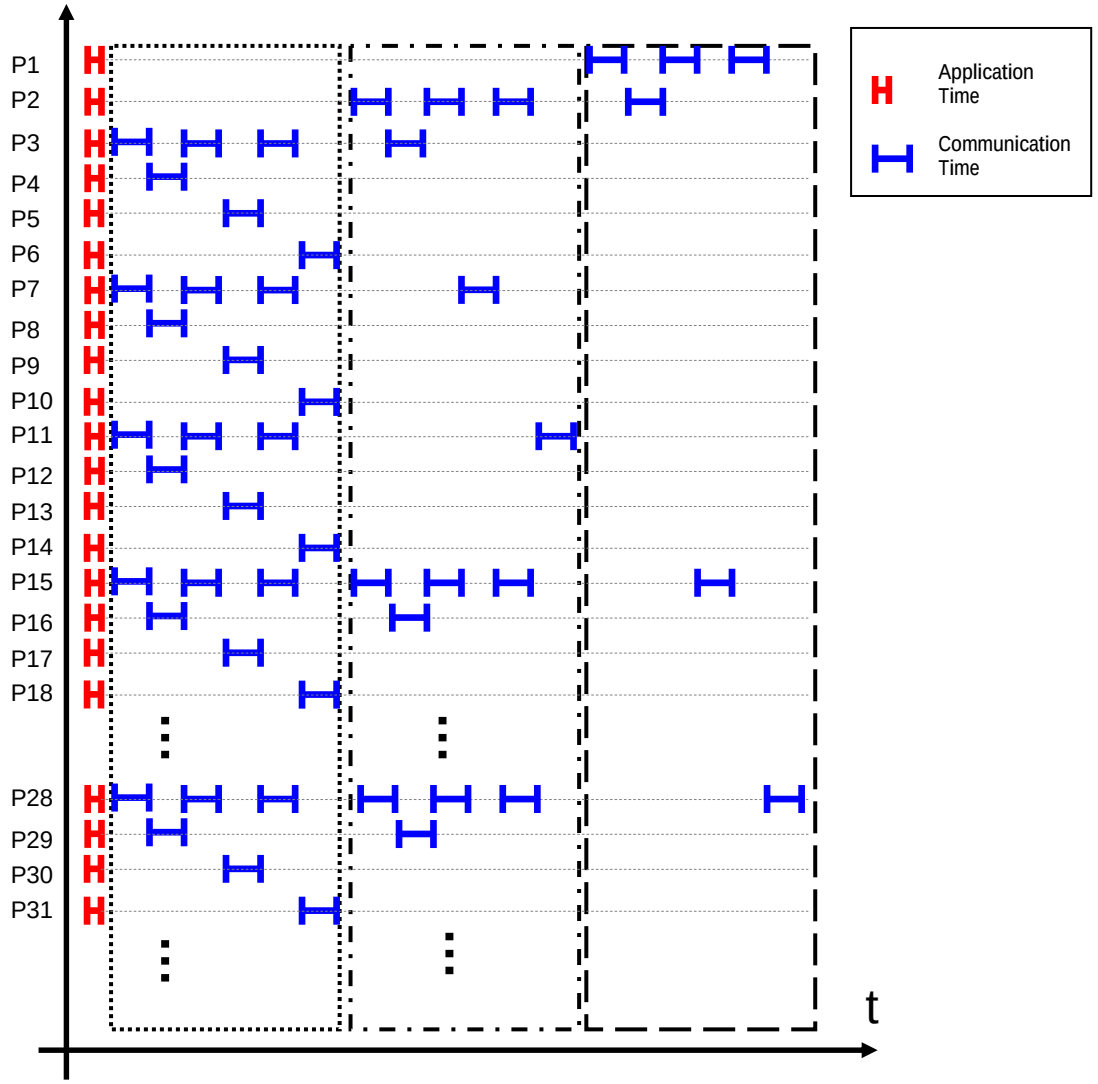

Figure 2. Chronogram of the Hierarchical Architecture of 4 levels and amplitude equal to 3.

Considering the hierarchical distribution of processors, the pillars of this model are:

a. Membranes distribution as in [Tejedor, 2007].
b. Proxy for processor as in [Tejedor, 2007].
c. Balanced tree topology of processors. Benefit obtained from this interconnection topology among processors is that the number of total external communications is minimized because proxys only exchange information with their direct descendants so, total number of external communications is *2(P-1)*, where

$$P = \frac{A^N - 1}{A - 1} \tag{4}$$

d. Token passing in the communication. A sequential order of communication is established for each processor in the same subtree; this way, there can not be more than proxy trying to transmit at the same time in the same subtree which is in. But, sequential external communications of a subtree are carried out in parallel with the ones of any other subtree of the same level. Last, established order for different levels is bottom-upm, i. e., no subtree of a given level begins its communications until every subtree of lower levels have finished.

This communication policy avoids collisions and network congestion, but additionally permits to be parallelized the 2(P-1) external communications so, the longest external communication sequence in each evolution step will be:

$$2(A-1)(N-1) \tag{5}$$

Hence, in this hierarchical architecture *K* membranes have been located in each processor. At the worst, the application of the rules in each one of these membranes will be made sequentially in each processor. Therefore, the required time to carry out the application of the rules of *M* membranes will be:

$$K\,T_{apl} \tag{6}$$

From (1), (4) and (6) the required time to carry out the application of the rules of *M* membranes will be:

$$M\,\frac{(A-1)}{A^N-1}T_{apl} \tag{7}$$

On the other hand, from (5) it is obtained the required time to carry out the communication among processors of the architecture:

$$2(A-1)(N-1)T_{com} \tag{8}$$

Therefore, from (7) and (8) the required time to perform a complete evolution step will be:

$$T = M\,\frac{(A-1)}{A^N-1}T_{apl} + 2T_{com}(N-1)(A-1) \tag{9}$$

Once the required time to perform an evolution step is known, we can determine the number of levels (L_{opt}) and the amplitude (A_{opt}) of the architecture in order to minimize this time:

$$A_{opt} = 2 \tag{10}$$

$$L_{opt} = \frac{\ln\left(\sqrt{T_{apl}\,\frac{M}{T_{com}}\ln(2)}\cdot\sqrt{T_{apl}\,\frac{M}{T_{com}}\ln(2)+8}+T_{apl}\,\frac{M}{T_{com}}\ln(2)+4\right)}{\ln(2)}-2 \tag{11}$$

From (9) and (10) the minimum time required to perform an evolution step is:

$$T_{\min} = \frac{M}{2^{L_{opt}}-1}T_{apl} + 2T_{com}(L_{opt}-1) \tag{12}$$

And, from (4) and (10) the number of processors necessary to run the P system minimizing the necessary time to carry out an evolution step will be:

$$P_{opt} = 2^{L_{opt}}-1 \tag{13}$$

Comparative Analysis

In this section, we present an empirical analysis comparing proposed architectures in [Tejedor, 2007] with the hierarchical architecture proposed here.

Figure 3 shows the number of processors of both architectures to reach their respective optimum times for an evolution step. As it can be seen, hierarchical architecture have a bigger number of processors than previous work. Also, the growing slope becomes steeper as the number of membranes of the P system is growing. This way, hierarchical architecture reaches a better parallelism degree in proportion to a bigger number of processors in the architecture. This fact increases the parallel application of evolution rules and the internal communication among membranes allocated at the same processor.

Consequently, the bigger parallelization degree of our architecture and external communications parallelization between subtrees of same level obtains smaller minimum times per evolution step. Figure 4 shows resulting times for both architectures as the number of membranes of the P system grow up.

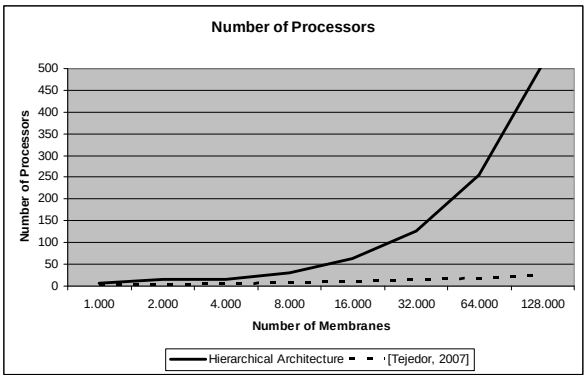

Figure 3. Number of processors to reach optimum times per evolution step among membranes in both architectures.

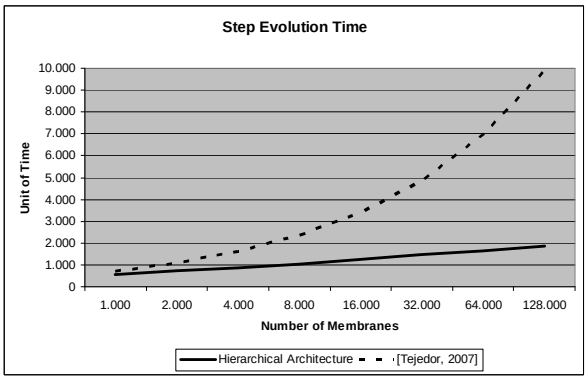

Figure 4. Optimum times per evolution step in both architectures.

Conclusions

In this paper a hierarchical architecture of communications to implement P system has been introduced. This architecture is based on the location of several membranes at the same processor, the use of proxys for communicating processors placed in a balanced tree topology and token passing in the communication.

This solution, just like previous architectures, avoids communication collisions, reduces the number and length of the external communications, but permits for the first time the parallelization of external communications and increases drastically the application rules and internal communications parallelization degree. All this, allows us to obtain a better step evolution time than any other suggested architectures and is closer to the massively parallelism character inherent to the membranes computer model.

Bibliography

[Păun, 1998] Gh.Păun. Computing with membranes. Journal of Computer and System Sciences, 61 (2000), and Turku Center for Computer Science-TUCS Report No 208, 1998.

[Tejedor, 2007] A. Tejedor, L. Fernandez, F. Arroyo, G. Bravo, An architecture for attacking the bottleneck communication in P Systems. In: M. Sugisaka, H. Tanaka (eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics, Jan 25-27, 2007, Beppu, Oita, Japan, 500-505.

[Ciobanu, 2004] G.Ciobanu, W.Guo. P Systems Running on a Cluster of Computers. Workshop on Membrane Computing (Gh. Păun, G. Rozenberg, A. Salomaa Eds.), LNCS 2933, Springer, 123-139, 2004.

[Syropoulos, 2003] A. Syropoulos, E.G. Mamatas, P.C. Allilomes, K.T. Sotiriades, A distributed simulation of P systems, A. Alhazov, C. Martin-Vide and Gh. Păun (Editors): Preproceedings of the Workshop on Membrane Computing; Tarragona, July 17-22 2003, 455-460.

Authors' Information

Ginés Bravo García – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: gines@eui.upm.es

Luis Fernández Muñoz – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Fernando Arroyo Montoro – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: farroyo@eui.upm.es

Juan Alberto Frutos Velasco – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: jafrutos@eui.upm.es