A New Distributed Computing Environment based on Mobile Agents for Massively Parallel Applications

FZ. Benchara, M. Youssfi, O. Bouattane, H.Ouajji, M.Bensalah

***Abstract*—**In this paper, we propose a new distributed environment for High Performance Computing (HPC) based on mobile agents. It allows us to perform parallel programs execution as distributed one over a flexible grid constituted by a cooperative mobile agent team works. The distributed program to be performed is encapsulated on team leader agent which deploys its team workers as Agent Virtual Processing Unit (AVPU). Each AVPU is asked to perform its assigned tasks and provides the computational results which make the data and team works tasks management difficult for the team leader agent and that influence the performance computing. In this work we focused on the implementation of the Mobile Provider Agent (MPA) in order to manage the distribution of data and instructions and to ensure a load balancing model. It grants also some interesting mechanisms to manage the others computing challenges thanks to the mobile agents several skills.

***Keywords*—**Parallel and Distributed computing, Distributed environment, Mobile agents, Image Processing.

I. Introduction

NOWADAYS , everyone need to get information, results and achieve tasks in real time. So it is possible by the use of the computer science technologies which make the complex tasks easy in order to perform these. For example running an application of weather predictions which is based on a big number of data and complex simulations using just one or two processors can be a hard task for the machine and sometime impossible to achieve the results. So we need to introduce cooperation amongst processing power of different machines in order to overcome these problems.

The parallel computing concept is widely used in order to overcome these challenges with its flexible and extensible architectures. Many fast parallel machines are created in order to be flexible with the applications needs but they presented others challenges according to their high cost and to their limitation on the test and validation of new parallel algorithms. So the creation and the use of the Parallel Virtual Machine (PVM) [1] are considered as a suitable solution for these needs. This PVM machine is constituted over a grid computing using a set of heterogeneous machines connected with each other by the middleware. In [2], the authors proposed a virtual machine using mesh connected computer MCC which becomes mesh with multiple broadcast in [3] and polymorphic torus in [4] and a reconfigurable mesh computer RMC with integrated network for each processing element in [5]-[6] and recently by the use of GPUs and FPGAs in [7]-[8]-[9]. It is also the aim of our laboratory where we have designed and developed a parallel and distributed virtual machine based on our work in [10]-[11]-[12]-[13] which are improved in [14] by assigning a set of distributed VPEs (Virtual Processing Element) objects for each processing element in the grid. We can say that by introducing the concept of the grid and especially the middleware, the parallel computing is converged to the parallel and distributed one where the computing performance depends on the quality and the performance of the middleware. The question now is how to achieve a high performance computing. For the load balancing problem some algorithms have been designed for distributed systems [15]-[16]-[17]-[18]-[19]. To move the loads in a distributed system the authors have used in [20] a mobile agent, which migrate loads from overloaded nodes to the lightly loaded ones and considering that all the grid nodes are homogeneous.

In this context, related to all these previous works we are focused on the use of the middleware which is based on the mobile agents. It is considered as a new grateful computer science technology which is used in [21] in order to propose a new model for automatic construction of business processes based on multi agent systems. And also in [22] in order to improve the management, the flexibility and the reusability of grid like parallel computing architecture and in [23] in order to improve the time efficiency of a medical reasoning system. So thanks to the several interesting mobile agents skills, we design and implement a parallel and distributed environment composed by the middleware which assigns and orchestrates a set of mobile agents as AVPUs (Agent Virtual Processing Element) for each physical processor in heterogeneous parallel and distributed grid computing. And which implements some interesting mechanisms for load balancing, fault tolerance, and

FZ. Benchara is a Phd Candidate from ENSET, Hassan II University of Casablanca, Mohammedia, Morocco (e-mail: benchara.fatemazahra@gmail.com).

M. Youssfi, is a Phd in ENSET, Hassan II University of Casablanca, Mohammedia, Morocco (e-mail: med@youssfi.net).

O. Bouattane, is a Phd in ENSET, Hassan II University of Casablanca, Mohammedia, Morocco (e-mail: o.bouattane@gmail.com).

H. Ouajji, is a Phd in ENSET, Hassan II University of Casablanca, Mohammedia, Morocco (e-mail: ouajji@enset-media.ac.ma).

M. Bensalah, is a Phd in FSR, University Mohammed V AGDAL Rabat, Morocco (e-mail: m.bensalah@fsr.ac.ma).

to reduce the communication cost in order to have a control about all the parallel and distributed computing challenges and ensure a high performance computing. This paper is organized as follows:

- We will describe the proposed model for parallel and distributed computing, its main components which are: the mobile agents team leaders and team workers and the Mobile Provider Agent in section 2.
- The section 3 is focused on presenting several mechanisms used by the mobile provider agent in order to perform a load balancing middleware and a high performance parallel and distributed computing.
- Some interesting results performed by implementing the c-means and the fuzzy c-means algorithm in this model will be presented in the section 4.

II. Distributed Computing Environment Architecture

A. Distributed Computing Environment Model

Distributed Computing Environment is a new scalable and robustness model for performing a high performance execution of the parallel programs in a distributed system. It constitutes a parallel and distributed grid computing which is flexible with different topologies: 2D Mesh, 3D Mesh… and with different architectures: SIMD (Single Instruction Multiple Data), SPMD (Single Program Multiple Data), MIMD (Multiple Instruction Multiple Data), MPMD (Multiple Program Multiple Data…It is based on cooperative mobile agent team work as (AVPUs) in order to perform parallel and distributed tasks. In Fig.1 for example in order to perform a distributed segmentation of big data segmentation we need to implement the well known segmentation algorithm the c-means which is performed using SPMD architecture. Each (AVPU) receives the distributed program and data from its team leader (AVPU) and achieves tasks and sends the results to its team leader in order to perform the image segmentation.

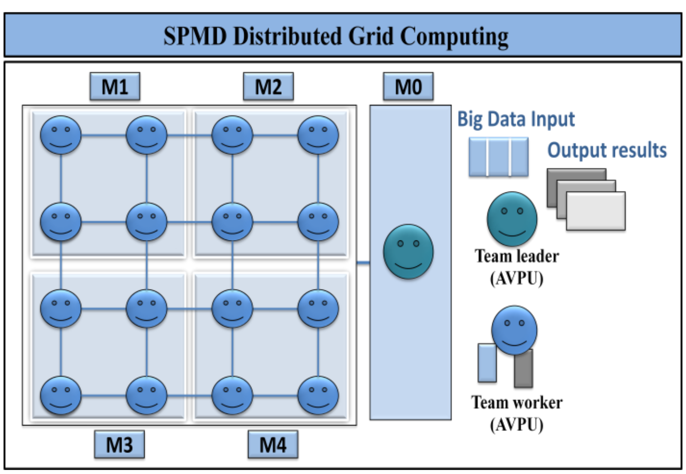

Fig. 1 2D Mesh Grid Computing for Distributed image

B. Model Main Components Overview

This distributed computing environment in Fig.2, is based on the power of the middleware and the mobile agents. The HPC of the parallel programs is performed in this environment by cooperative main components which are created in these different environment states.

1. Launching State

The middleware creates (*Node Agent Container*) for each involved machine in the distributed computing and connects each of it in order to constitute the grid computing. The first machine responsible for launching this environment is named (*Node Host Agent node Container*).

2. Deployment State

When the parallel program is deployed, the middleware deploys (*team leader AVPU*) for each node which encapsulate tasks and create their (*AVPU team workers*). We can have one or two team leader AVPU in the same node according to the number of the parallel programs deployed in this environment. Also each AVPU are autonomous and can decide to replicate its self in order to ensure a fault tolerance environment.

3. Running State

When the parallel program is running, the team leader AVPU send the tasks and data to the Mobile Provider Agent (*MPA*) in order to manage and provide them to the team workers AVPUs by ensuring the load balancing of tasks execution in the grid computing. At the end, the Agent Provider sends the results to the team leader agents in order to perform the final results and return it to the MPA in order to be broadcasted for different nodes in the grid.

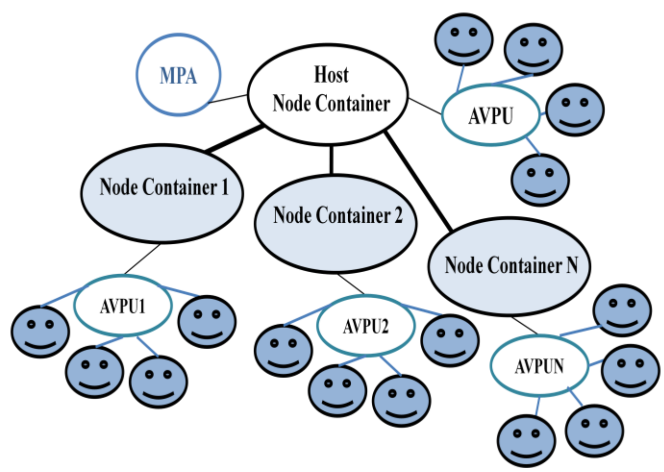

Fig. 2 Distributed Computing Environment main components overview

III. From Parallel to Distributed Computing

A. A Fast Distributed Computing Middleware

As this environment is constituted over heterogeneous machine with different degrees of performance we need some additional component which is the Mobile Provider Agent

(MPA) in our model as presented in Fig.3 in order to achieve our main goal. The Mobile Provider Agent is responsible for managing the pool of tasks, data, and results. It is an intelligent mediator between the team leader AVPU and the team workers AVPU. This MPA have knowledge of the number of team workers and their nodes performance. It manages a set of distributed pools by introducing the priority of the execution and the agent AID (Agent Identifier) for each tasks and data in these pools. So the agents can easily follow their data and tasks when they move to the MPA container. The MPA have also the ability to decide according to the parallel program architecture when to send the tasks and data and when to keep the agents to move to the pools. So with the implementation of the MPA in our model we have a control about the load balancing problem, and we reduce at the same time the communication cost. Also MPA is autonomous, it can decide to move and to clone itself and resume its work in order to ensure a fault tolerance environment.

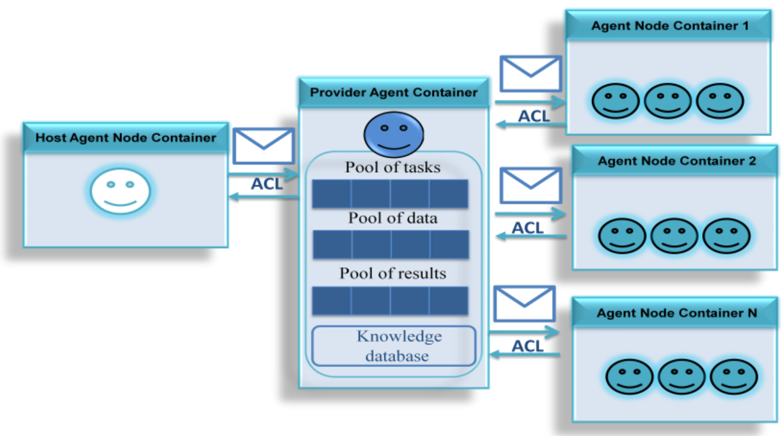

Fig. 3 Distributed Middleware Mechanisms for Computing Management

B. Cooperative Mobile Agent Model Implementation

In Fig. 4, we describe a scenario about the interaction between the different model main components in order to perform the execution of the parallel programs. This model is implemented using the JADE framework [24].

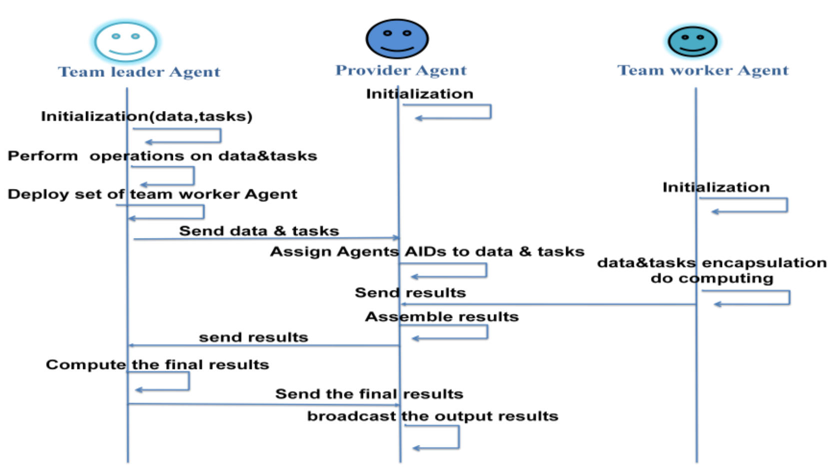

Fig. 4 Sequence diagram for cooperative and distributed computing model

IV. APPLICATIONS

The main idea in this section is to present the performance of this model by performing the execution of two parallel programs for medical image segmentation at the same time. Each program is assigned to team leader agent in order to be executed over SPMD architecture in this environment.

A. Distributed C-means Implementation

The parallel c-means algorithm as defined in [25] is implemented in this environment as a distributed one in order to perform the image segmentation. It is performed using the corresponding following steps described in Fig.5.

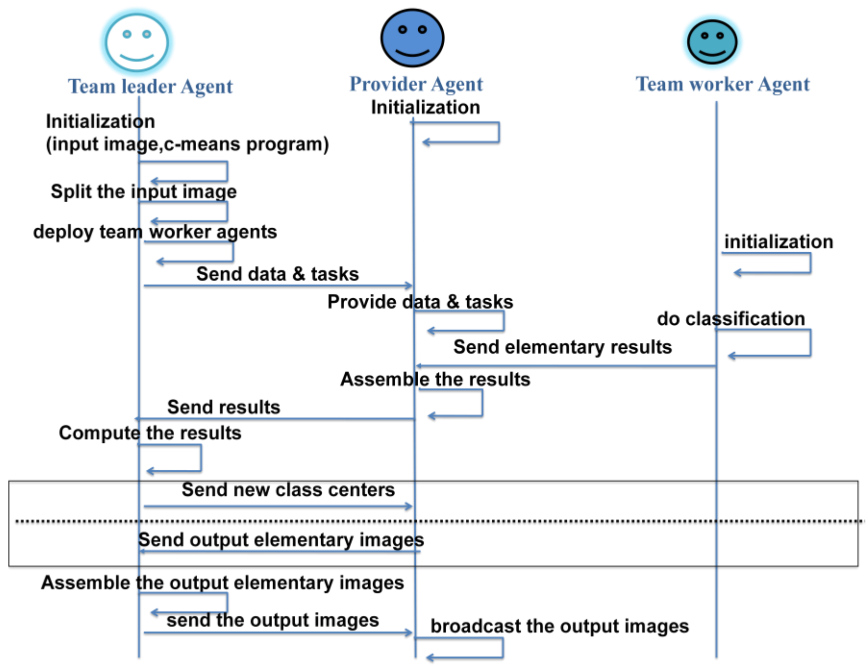

Fig. 5 Sequence diagram for cooperative and distributed computing model

B. Distributed Fuzzy C-means Implementation

The well known clustering algorithm the Fuzzy c-means (FCM) which is proposed by Dunn [26] and extended by Bezdek [27] is also implemented in this environment as a distributed program and which is encapsulated in a team leader agent. This later cooperates with its team works in order to perform the big data image segmentation.

C. Experimental Results

In order to present the effectiveness features of the implementation of both programs in different team leader agents in our model we choose an MRI medical image with different class centers initialization which is segmented into 5 output images in Fig.6a-g in order to detect the abnormal region in this MRI cerebral image. In Table I, Table II, and Fig.7 we see clearly the dynamic convergence of c-means and fuzzy c-means algorithms running respectively at the same time in this model.

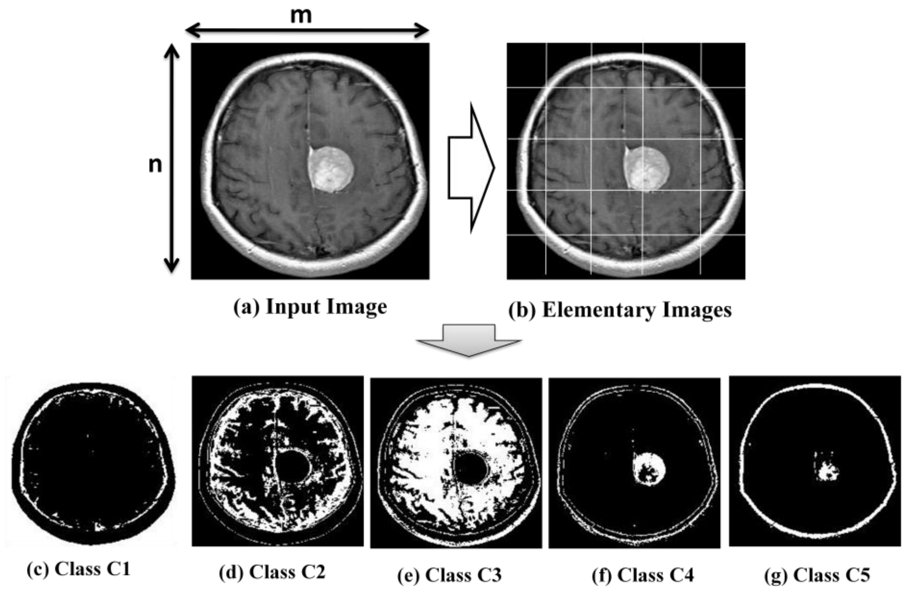

Fig.6 Output Segmented images results

TABLE I
DIFFERENT STATES OF THE DISTRIBUTED C-MEANS (DCM) ALGORITHM STARTING FROM CLASS CENTERS
(C1, C2, C3, C4, C5) = (1 , 49, 140, 240, 249)

Iteration	Value Of Each Class Center					Absolute Value Of The Error
	C_1	C_2	C_3	C_4	C_5	$\lvert J_n-J_{n-1}\rvert$
1	1	49	140	240	249	819074
2	1,385	74,547	116,739	219,488	251,929	385030,8088
3	3,082	77,002	111,825	201,473	249,149	39840,1231
4	3,790	77,838	110,026	190,175	244,954	10689,31769
5	3,790	76,132	107,894	183,859	242,084	6942,411183
6	3,790	75,195	106,710	179,022	240,019	3326,263246
7	3,790	73,316	105,261	176,592	239,212	2312,307919
8	3,343	71,708	104,525	174,495	238,277	3503,283554
9	3,082	71,343	104,403	172,825	237,385	2116,30221
10	3,082	69,498	103,368	171,858	236,932	1186,783677
11	2,876	68,258	102,866	171,3827	236,932	1700,633533
12	2,688	67,959	102,866	171,3827	236,932	1314,60772
13	2,688	67,959	102,866	171,3827	236,932	0

TABLE II
DIFFERENT STATES OF THE DISTRIBUTED FUZZY C-MEANS (DFCM) ALGORITHM STARTING FROM CLASS CENTERS
(C1, C2, C3, C4, C5) = (49.5, 50.5, 140.5, 240.2, 249.5)

Iteration	Value Of Each Class Center					Absolute Value Of The Error
	C_1	C_2	C_3	C_4	C_5	$\lvert J_n-J_{n-1}\rvert$
1	49,5	50,5	140,5	240,2	249,5	566746,6435
2	38,958	42,937	121,596	222,116	243,412	64496,77352
3	20,897	34,294	108,416	206,890	244,013	89667,11461
4	4,829	38,072	101,909	194,924	244,460	108307,9287
5	1,083	48,186	99,567	186,355	243,646	52428,48332
6	1,024	53,258	99,344	180,582	242,467	6553,849896

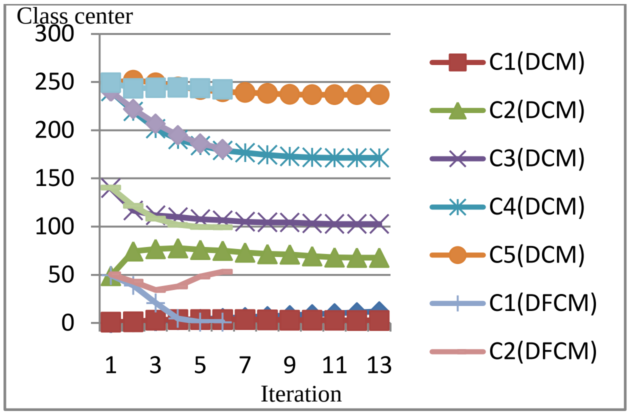

Fig. 7 Dynamic convergence of *(DCM)* with initial class centers *(c1, c2, c3, c4, c5) = (1, 49, 140,240,249)* and *(DFCM)* with *(c1, c2, c3, c4, c5) = (49.5, 50.5, 140.5, 240.2, 249.5)*

V. CONCLUSION

In this paper, we have presented a distributed environment for high performance parallel and distributed computing. It is based on a cooperative mobile agent grid computing flexible for different parallel architectures which is constituted over a distributed system. It is based on an innovated idea which is the use of Mobile Agents as virtual processing unit AVPUs. Each mobile agent associates its skills: autonomy, and mobility, and adaptability, and communication ability using ACL messages (Agent Communication Language) in order to provide the processing power needs for performing the parallel programs execution and at the same time have a control about all the parallel computing challenges. In this environment we distinguish the Mobile Provider Agent (MPA) which manages and orchestrates the computing between the different team works and grants a load balancing, and a fault tolerance environment. And it implements also some interesting mechanisms for reducing the communication cost in the grid computing in order to perform a high performance computing.

REFERENCES

[1] H.El-Rewini, M.Abd-El-Barr, “Advanced Computer Architecture and Parallel Processing”. Wiley, 2005.

[2] R. Miller. et al, “Geometric algorithms for digitized pic-tures on a mesh connected computer,” IEEE Transactions on PAMI, Vol. 7, No. 2,pp.216-228,1985.

[3] V. K. Prasanna and D. I. Reisis, “Image computation on meshes with multiple broadcast,” IEEE Transactions on PAMI, Vol. 11, No. 11, pp.1194-1201,1989.

[4] H. LI, M. Maresca, “Polymorphic torus network,” IEEE Transaction on Computer, Vol. C-38, No. 9, pp.1345– 1351,1989.

[5] T. Hayachi, K. Nakano, and S. Olariu, “An O ((log log n)2) time algorithm to compute the convex hull of sorted points on re-configurable meshes,” IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 12,pp.1167- 1179,1998.

[6] R. Miller, V. K. Prasanna-Kummar, D. I. Reisis, and Q. F. Stout, “Parallel computation on re-configurable meshes,” IEEE Transactions on Computer, Vol. 42, No. 6,pp.678-692,1993.

[7] Yue Zhao, Francis C.M. Lau, "Implementation of Decoders for LDPC Block Codes and LDPC Convolutional Codes Based on GPUs," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp.663-672, 2014.

[8] Jing Wu, Joseph JaJa, Elias Balaras, "An Optimized FFT-Based Direct Poisson Solver on CUDA GPUs," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp.550-559, 2014.

[9] Abid Rafique, George A. Constantinides, Nachiket Kapre, "Communication Optimization of Iterative Sparse Matrix-V ector Multiply on GPUs and FPGAs," IEEE Transactions on Parallel and Distributed Systems, 28 Feb. 2014. IEEE computer Society Digital Library. IEEE Computer Society.

[10] M. Youssfi, O. Bouattane and M. O. Bensalah “A Massively Parallel Re-Configurable Mesh Computer Emulator: Design, Modeling and Realization” J. Software Engineering & Applications,pp.11-26,2010

[11] Bouattane, B. Cherradi, M. Youssfi and M.O. Bensalah “Parallel c-means algorithm for image segmentation on a reconfigurable mesh computer”, ELSEVIER. Parallel computing, 37, pp.230-243, 2011.

[12] M. Youssfi, O. Bouattane, and M.O. Bensalah “ On the Object Modelling of the Massively Parallel Architecture Computers” Proceedings of the IASTED Inter. Conf. Software engineering, Innsbruck, AUSTRIA,pp.71-78,2010.

[13] M. Youssfi, O. Bouattane, and M.O. Bensalah “Parallelization of the local image processing operators. Application on the emulating framework of the reconfigurable mesh connected computer”. Proceeding of scientists meeting in Information Technology and Communication JOSTIC. Rabat, pp.81-83, 2008.

[14] Mohamed Youssfi, Omar Bouattane, Fatéma Zahra Benchara , Mohammed Ouadi Bensalah “A Fast Middleware For Massively Parallel And Distributed Computing”, Inter. Journal of Research in computer and Communication Technology, Vol 3, Issue 4, pp.429-435,2014

[15] D.R. Karger and M. Ruhl, "Simple Efficient Load-Balancing Algorithms for Peer-to-Peer Systems," Theory of Computing Systems, vol. 39, pp.787-804, 2006.

[16] Ioannis Konstantinou, Dimitrios Tsoumakos, Nectarios Koziris, "Fast and Cost-Effective Online Load-Balancing in Distributed Range-Queriable Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 8, pp.1350-1364, 2011.

[17] Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, Yu-Chang Chao, "Load Rebalancing for Distributed File Systems in Clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5,pp.951-962, 2013.

[18] Hung-Chang Hsiao, Che-Wei Chang: "A Symmetric Load Balancing Algorithm with Performance Guarantees for Distributed Hash Tables". IEEE Trans. Computers vol.62, no. 4, pp.662-675, 2013.

[19] G.Cybenko, Load balancing for distributed memory multiprocessors. Journal of Parallel and Distributed Computing, 7, pp.279-301, 1989.

[20] Liu, J., Jin, X. and Wang, Y. "Agent-Based Load Balancing on Homogeneous Mini-grids: Macroscopic Modeling and Characterization", IEEE Transactions on Parallel and Distributed Systems, pp.586-594, 2005.

[21] José A. García Coria, José A. Castellanos-Garzón, Juan M. Corchado, “Intelligent business processes composition based on multi-agent systems”, ELSEVIER. Expert Systems with Applications, vol. 41, Issue 4, Part 1, pp.1189-1205, 2014.

[22] David Sánchez, David Isern, Ángel Rodríguez-Rozas, Antonio Moreno, “Agent-based platform to support the execution of parallel tasks”, ELSEVIER. Expert Systems with Applications, vol. 38, Issue 6, pp.6644-6656, 2011.

[23] Alejandro Rodríguez-González , Javier Torres-Niño, Gandhi Hernández-Chan, Enrique Jiménez-Domingo, Jose Maria Alvarez-Rodríguez , “Using agents to parallelize a medical reasoning system based on ontologies and description logics as an application case”, ELSEVIER, Expert Systems with Applications, vol. 39 , pp.13085-13092, 2012.

[24] F. L. Bellifemine, G. Caire, and D. Greenwood, “Developing Multi-Agent Systems with JADE”,Wiley, 2007.

[25] O.Bouattane, B. Cherradi, M. Youssfi and M.O. Bensalah “Parallel cmeans algorithm for image segmentation on a reconfigurable mesh computer”, ELSEVIER. Parallel computing, vol. 37,pp.230-243,2011

[26] Dunn J.C., “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters”. Journal of Cybernetics, vol.3, Issue 3, pp.32-57, 1973.

[27] J.C. Bezdek, “Pattern Recognition with Fuzzy Objective Function Algorithms”, Plenum Press, New York, 1981.