WChord: a Hybrid and Bio-Inspired Architecture to Peer to Peer Networks

Thiago Q. de Oliveira, Tiago C. Pessoa, André R. Cardoso, Joaquim Celestino Júnior
Laboratório de Redes de Comunicação e Segurança (Larces)
Universidade Estadual do Ceará (UECE)
Fortaleza-CE, Brazil
{*thiagoq, carneiro, andrec, celestino*}*@larces.uece.br*

***Abstract*—Given the high demand for Peer to Peer applications (P2P), it becomes essential to develop an architecture to operate this type of network to use efficiently the bandwidth and minimize the existing overhead for the maintenance of its topology. Among the existing architectures for P2P networks, the Chord is notable for possessing a powerful search system, but on the other hand it has a high overhead during its topology update. This work presents a hybrid bio-inspired architecture for P2P networks, called WChord which adds concepts of Chord and the dominance hierarchy of wasps P. dominulus.**

***Keywords*-component: P2P networks; Chord; Peer to Peer network architecture; Wasps; Peer to Peer protocol**

I. Introduction

Studies have shown that P2P systems have been impacting on bandwidth consumption of the internet, often substantially surpassing its own web application traffic [1].

P2P networks application are very popular on the Internet. Examples of this type of application are MSN Messenger, Yahoo Messenger, GTalk, and file sharing applications such as Emule, Kazaa, and Bit Torrent.

Some applications that previously worked using the client-server architecture were attracted by the benefits offered by P2P networks. Benefits may include eliminating a single point of failure, data replication and the possibility of interaction with other users. Among these appliactions, it is possible to mention some distributed databases [2] and distance education applications [3].

In addition, new P2P applications arise in a mass quantities and the number of individuals using these applications increase daily. Today, for example, an increase in the number of online games that use P2P networks as their basic architecture, can be easily spotted [4].

P2P networks are very dynamic. Within them there are nodes going in and out all the time, in general the stream is very high. Researches showed that 80% of nodes stay in this type of network for less than an hour [5]. For P2P systems to maintain their topologies updated, data exchanges must be carried out frequently, causing a considerable increase in data traffic.

Any improvements in the way P2P networks organize themselves is important, because these applications are the most used by internet users and are those which have caused the greatest impact on the bandwidth consumption.

All these factors have made many surveys be developed in this area in an attempt to reduce bandwidth consumption, reduce the delay to ensure communication between peers as close to real-time as possible, allow download of a same resource from multiple sources and distribute network traffic.

Many architectures focused on P2P networks were developed as the CAN [6], Chord [7], Pastry [8] and Tapestry [9]. The Chord has a powerful search system.

In this work, the Chord was changed in order to obtain the best suited model to the dynamism inherent in P2P networks. Henceforth minimizing the amount of messages exchanged to maintain stable network and therefore reducing bandwidth consumption and peer processing for topology maintenance. This change was based on dominance hierarchy of wasps P. dominulus, resulting in a hybrid bio-inspired P2P architecture called WChord (Wasps-Chord).

Organization of my findings is as follows. First some related works are shown. Sections three presents the main features of the Chord. Followed by a more complete description of the inspired wasps model. Section five presents the proposed architecture of the Wchord. Next the result of a simulation which compares the two architectures is presented. Finally, the conclusion and the importance of future work is presented.

II. Related Works

In Garces [10], two layers are proposed to take advantage of the heterogeneity of peers, based on network lag and the communication between these groups. The routing is first performed with super-peers. After finding the appropriate super-peer, the search is performed on the group for which that super peer is responsible. This way, the authors show that the search for a resource is in the order of complexity O (logM), where M is the number of super-peers. This model does not take into account important metrics to define super-peers as bandwidth, timelive nor processing capacity.

In Coral [11], the authors use a different mechanism. The focus is on reducing the lag time lookup in mechanisms instead of trying to reduce the number of hops. To do this, the authors divide the ring of the Chord into three distinct

978-1-4577-1124-4/11/$26.00

rings, called clusters. The level-2 consists of peers which are 30msec from one to the other, while the level-1 consists of level-2 with a 100msec band. Finally, the level-0 represents the original chord ring. The idea is to group the peers so that the lookup between them is quite fast. However, this architecture is quite complex, both by grouping into the appropriate sections as by the fact that the communication between peers also depends on the actual network flow at the moment.

III. CHORD

The Chord [7], classified as a system for structured P2P networks that uses the concept of DHT (Distributed Hash Table), has a circular space (ring) where the nodes and the keys are placed in growing positions from the ring. Every node has an identifier (Node_ID), which is determined through the hash function SHA-1 using the peer IP address, while peer resources, represented by a key (Key_ID), and are determined by SHA-1 hash function using the resource name. A key is mapped to the first Node_ID that is greater than or equal to it (the key is mapped to the next peer clockwise). It means the first peer to have an id greater than the value of Key_ID stores data from that resource.

This way, each peer is responsible for a set of N keys, and if the peer leaves the network, the keys are automatically assigned to the next clockwise direction peer ring. The goal is that the table with the available resources on the network is distributed in a balanced way by peers that compose it.

In addition, the peers in the chord are unaware all the ring peers in existence, only a small amount. Peers store the other peers location in a table called finger table.

The location of the network resources will always be stored by the peer whose ID is greater than the resource ID. So, for consistency, when a peer joins the network, some resources will be mapped by this peer, because it will be the peer with ID directly above the ID of the resources. Similarly, when a peer leaves the network, resources that were being mapped by this peer will be stored at the next peer ring in the sequence.

To find a resource on the network, the SHA-1 hash function with the resource name is utilized. This allows the location of the peer responsible for storing the given resource and through the table finger_table the search will find the peer that stores the data for that resource.

However, by having ring topology, all the peers must be inserted in the ring, as the flow of entering and exiting peers on the network increases, an event known as chunk, the ring topology has to adapt according to the condition at that time and redistribute the DHT table. This process can lead to inefficiencies in the system, since processing and messaging exchanges are expended to update topology.

To keep the ring consistent, all the nodes maintain a list of R successors. If the immediate node successor in question does not respond, the peer may contact the next successor from that list. Whereas the probability of a peer fails is P and this probability is the same for all peers, then the probability for all of these peers to fail simultaneously is P^R. If all R successors of a peer fail, there will be a break in the ring.

Although the probability of a simultaneous successors failure is theoretically low, in very dynamic networks, the possibility of a disruption can still be fairly high. This fact is considered a vulnerability in the Chord system.

IV. DOMINANCE HIERARCHY OF WASPS P. DOMINULUS

Bio-inspired models have been explored by scientists to make systems and platforms increasingly independent of an administrator. It also allows for the ability to adapt to changes in the environment in which they live [12] [13]. In order to solve that, mathematical models based in behavior of biological systems have been built and implemented in the computing world [14] [15]. This work is based on Bio-inspired model of dominance hierarchy of Wasps P. Dominulus [16] [17] [18] [19] which is described below.

The wasps from the species P. dominulus have linear dominance hierarchy with their females, where the alpha female α dominates all the others in the nest. The beta female β dominates all the others except the alpha. And thus the hierarchy continues until the last female which is dominated by all the others. This hierarchy is stabilized through the fighting that occurs in pairs and decrease in intensity, from violent fights to the simple recognition of submission [16] [17].

The probability of two females fighting is directly proportional to the time they are moving in the nest. Thus, the higher the position of a member in the nest, the greater the probability it interacts with another member, because the dominant females are responsible for tasks such as laying eggs and nest building. Females in lower positions tend to perform external tasks, such as food gathering, and are therefore less likely to be involved in fights.

It is usually a dominant female which takes the initiative to start a fight, as their combat wins activate the endocrine system, differentiating itself from other wasps, enhancing its domineering status. Therefore, the wasp which wins a fight tends to win more matches, while those which lose fights are weakened and tend to continue losing further fights.

As a result of creating this hierarchy, there has been the emergence of an efficient labor division, as well as increased reproduction, within the nest. The model utilized to describe this behavior is based on the principle of positive feedback: a wasp which wins a fight tends to win other fights.

Each individual i belonging to nest n is characterized by a force F_i that influences its ability to win fights. F_i on a wasp reflects its endocrine activity and its ovarian development. The F_i parameter proportional to the amount of times i wins a fight. Whenever i wins a fight, the F_i parameter increases and whenever i loses a fight, the F_i decreases.

Unlike in nature, in the monte carlo model[18], each individual i belonging to the nest n starts the simulation with the same strength and dominance index, which is the wins ratio over the total of challenges with which it participated.

The initial probability of an individual i dominating an individual j belonging to the same nest, is called $Q_{ij}+$, is $\frac{1}{2}$. $Q_{ij}+$ and is defined by the following expression [18]:

$$Q_{ij}+ = \frac{1}{1 + e^{-n*(Fi-Fj)}} \tag{1}$$

The probability of the individual i loosing to an individual j $Q_{ij}-$, is expressed by $1 - (Q_{ij}+)$. At the beginning the forces of all hive members are equal, so $Q_{ij}-$ and $Q_{ij}+$ also are.

In nature it is known that the strongest individuals tend to interact more than weaker ones. The probability of interaction of a individual i with another individual, belonging to the same nest, Y_i, is defined by the following expression [18]:

$$Y_i = \frac{i}{1 + e^{\frac{-F_i}{\theta}}} \tag{2}$$

In the previous expression, θ is a constant that takes the value of 100 to the probability that an individual's interaction with another individual in the same nest is approximately $\frac{1}{2}$ in the first hundred simulation interactions (fighting). As in nature, the strongest the individuals i and j are, the more likely it will be the interaction between them.

V. The WChord proposal

As already mentioned, the Chord, despite having a very promising design, which promotes load balancing and has an efficient search engine, there are some negative aspects.

One of these aspects is the possibility of chord ring rupture, mainly in networks where the peers flow entering and exiting is high. In addition, a considerable amount of messages are exchanged for topology maintenance. This messaging leads to a higher processing by peers and a higher bandwidth consumption. To minimize these problems, the WChord was proposed and is described below.

The WChord has a different design from the Chord. Just as there is a different social hierarchy within the wasps, the building of the peers hierarchy is also different. This hierarchy determines the peers that will be part of the ring. Other peers will bind to the peers of the ring, as well as the topology of hybrid P2P networks.

This restriction of the peers that will be part of the ring from the hierarchy between them, allows significant reduction in the overhead for updating topology and the possibility of breaking the ring.

A. General characteristics of the WChord

The WChord maintains some features present in the Chord. In the Chord all nodes have a unique ID generated by the hash function SHA-1 as a parameter for the node IP address. This function returns an integer that identifies the node on the network. Another important function is to generate a resource ID, passing as a parameter of the resource name.

In the proposed model, peers which have characteristics of the wasp bio-inspired model, fight amongst themselves to determine which peers will be part of the ring. Those having greater strength will be part of the ring while others will connect to any peer in the ring as a hybrid in the P2P's network architecture. These hybrids are commonly known as peer/super-peers.

B. Model description

Using the bio-inspired model described in the previous section, the hierarchy is based on the strength of the individual. In WChord the peers (wasps) strength will be calculated according to processing power, bandwidth, and timelive (amont of time that the peer is actively on the network).

Since the goal is for the network to suffer as little as possible with input and output nodes in the ring, timelive is the most impotant factor. Timelive decreases the risk of rupture and peers which are active for a long time, generally, tend to stay longer [5].

Secondly, the bandwidth is considered the most relevant because the peers that will be part of the ring will be mini-servers of the peers connected to it.

Such a force, in this first model, is measured from the expression below:

$$F = \frac{aP + bB + cT}{a + b + c} \tag{3}$$

In the previous expression, P represents the processing capacity, B is the bandwidth, T is the timelive and a, b and c are constant where $a < b < c$. Thus, the most important criteria have greater importance in the result of the force. It is possible to adjust the constants to define the relevance of the priorities.

Peers are identified by their ID and strength, so it is crucial to determine which peers will be part of the ring. Therefore, comparisons ("fighting") would be made among the peers to determine which of them will be part of the ring. As in nature, the wasps ("peers") of greatest strength will have a higher chance of winning.

Such matches are held between the peers connected to a super-peer periodically. The super-peer stores the ranking of peers connected to it after the fighting.

The WChord inherits from Chord the stabilisation protocol, the method of verification of predecessors, the routing table checking and the list of sucessors.

C. Resources mapping and location

In the Chord, the peer resource names are passed to a SHA-1 hash function that returns an integer. The peer that owns the Node_ID equal to or immediately above this ID will keep the resource location.

In the proposed model, the super-peers will be responsible for generating the SHA-1 hash function for the resources of peers connected to it. The location of these resources will then be saved by the peer ring with and appropriate ID, by following the same model as the Chord. Remember that only those peers which are part of the ring are able to store the location of resources.

Peer resources are mapped by their constituent super-peer to a peer of the ring. The super-peer will send an identifier, coupled to the resources name, for later location.

In WChord, the DHT resource location table proposal has the following fields: resource ID, resource name, IP address of the host that owns the resource or a super-peer which has that feature, the IP address of the peer connected to super-peer which actually has the resource. Thus, in relation to the Chord, the field for the peer connected to the ring that holds the resource is added.

While the order of complexity of a search in Chord is $O(\log N)$, where N is the number of peers present on the ring. In WChord, the order of complexity is $O(\log M)$ where $M < N$ is the number of super-peers, since the search will be routed between the super-peers. This way, the WChord has a search engine more efficient than the Chord, decreasing the lookup message flow.

D. Input and output of Peers

In this proposal, the first peer (wasp) to join the ring will be responsible for forming a group (nest). As long as other peers are entering the network, fights will be carried out to see which of them will be part of the ring and which ones will be the α-dominant (strongest individual in the nest). The one which is not considered able to be part of the ring will bind to any peer of the ring to join the network.

It's necessary to define some criteria for the amount of peers that will be part of the ring of WChord. It's possible to use peers that are exceedingly strong to increase the average of peers or to use a determined percentage of peers as part of the WChord.

According to [5] 80% of peers remain in the network for less than an hour. Therefore,the first time the WChord was used operators chose to use 20% of peers to create ring of WChord.

Obviously the number of peers on the network can vary greatly, being quite complex to ensure that only 20% of these peers are super-peers. Doing this at timed intervals, which can be configured according to how dynamic peers are fighting amongst themselves, 20% of peers will be chosen to be super-peers.

Such fighting could cause network architecture to change a lot, where a peer could become a super-peer and vice versa. This possibility could cause the network to have a large overhead, to redistribute the location of resources and for the formation of topology. So it is very important to choose appropriate constants associated to strength. Using formula 3 the constant c is defined exceeding b and a. Therefore we can minimize the probability of changing this topology, since the super-peers shall primarily be peers with more time on the network.

Other peers will use the hashing mechanism about the IP address to determine which peer to bind. This will link with the peer in the ring that an ID equal to or greater than it, a principal similar to the one used to map network resources.

The obstacle that must be overcome is joining too many peers to the ring, which in turn could risk rupture of the ring. Therefore it is extremely important to keep the ring smaller and more stable.

On egress, if it is a common peer, its output being voluntary or abrupt will not affect the system, since only the super-peer should be updated. The super-peer will run a method in the background to check which peers connected to it are still active. If a super-peer network wishes to leave voluntarily or involuntarily, peers connected to that super-peer will be attached to the next ring peer. Another possible approach would be to promote greater strength peer to super-peer, however, at this time, we chose the first option.

At the beginning of a new cycle, there is less than 20% of nodes as super-peers. The super-peers will check which peers could complete the ring, using the ranking they store after the fighting.

Regarding resource mappings in WChord, if a peer in the ring fails or leaves the network, the mappings from that peer will be mapped to its successor. In the case of a peer joining the ring, the mappings will be redistributed using the same method as described in the Chord.

The WChord aims to join the best networks based on DHT and hybrid networks. The first because they promote load balancing, and the latter because they are popular on the internet and have proven to be robust and scalable.

VI. Analysis and Simulations

The simulation environment chosen to compare the Chord with WChord was PeerSim [20]. The version used in this work was the 1.0.4.

PeerSim is an open source Simulator for P2P networks developed in Java and features high scalability (up to 1 million nodes). With this silumator, it is possible to make event-driven simulations and simulation oriented cycles.

Several implementations of protocols have been developed for this simulation environment, including: Chord, BitTorrent, Pastry, T-Man.

A. Results

Figure 1 presents the results of the simulation of a network that contains fifty thousand nodes and size of list of successors of each peer ring equal to 15. In that network, nodes are added and removed obeying exponential distribution, as was done in chord [7]. Figure 1 shows the number of failures in the network using the Chord and the WChord along the simulation. As in WChord not all peers will be part of the ring, the number of failures is much less than the number of failures in the Chord.

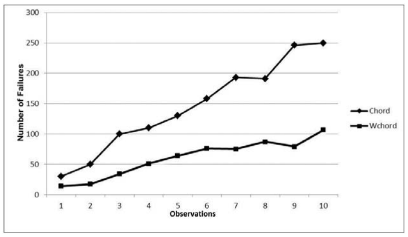

Figure 1. Evolution of the number of failures in Chord and WChord during the simulation

In Chord each peer network belongs to the ring, while in the WChord, the ring is composed of about twenty per cent of the total value of peers, which is recalculated periodically. As the amount of nodes belonging to the ring is smaller in WChord, the probability of failure on the ring of WChord is also smaller.

The other factor is that in WChord, in general, most peers that exit the system do not belong to the ring, while in the Chord all peers which exist were present in the ring.

Figure 2 shows the number of messages exchanged by the stabilization protocol between peers on the network using the Chord and the WChord for the topology maintenance. As in WChord not all peers will be part of the ring, the number of messages to maintain consistent the ring is much less than the number of messages exchanged in the Chord.

It is possible to observe that at the beginning of the simulation there are many stabilizing messages that will decrease over time until the situation is nearly stabilized. This is due the formation phase of the ring, there are constant updates in table finger_table, resulting in many messages to update topology.

In Figure 3, a simulation is carried out to verify how the size of a list of successors may impact on the number of failures. Using the list of successors with size 15, 30, 45 and 60, representing, respectively, $\log_2 x$, $2\log_2 x$, $3\log_2 x$, where x is the number of nodes.

Figure 3 shows that the increase in a list of successors of a node has little influence on the number of network failures. Such differences are only meaningful when increasing the list into more expressive proportions.

However, increasing the list of successors, directly impacts on the number of stabilization protocol, as shown in

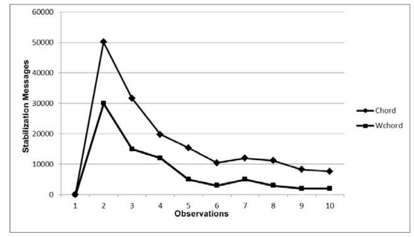

Figure 2. Evolution of number of stabilizing messages the in Chord and WChord

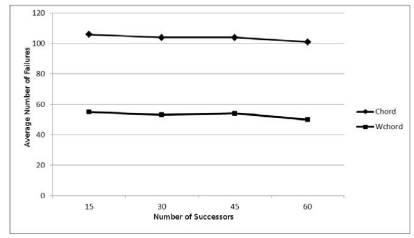

Figure 3. Average number of failures x size of list of successors.

Figure 4.

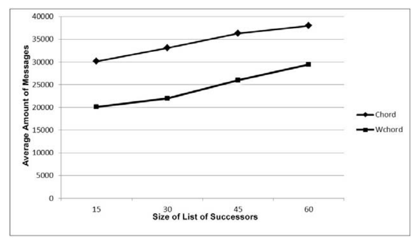

Figure 4. Average amount of stabilization protocol x size of list of successors.

This is due to the fact that in bigger lists of successors more messages must be exchanged in order to keep the list consistent. Therefore, there isn't an advantage to increase the size of successors list of the form indiscriminate, since this directly affects the consumption of bandwidth and processing by the nodes.

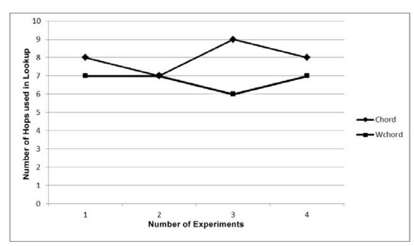

Figure 5. Average number of hops used in lookup engine x number of experiments

Finally, Figure 5 shows the average number of lookup messages to the location of a resource in 4 different experiments. It is observed that, in this case, the chord and wchord have similar results, despite the fact that the wchord has fewer nodes in the ring.

VII. Conclusion and Future works

This work describes a new architecture for P2P networks based on a bio-inspired model, proving its viability in relation to the Chord model. This infrastructure explores the advantages offered by hybrid P2P networks and P2P networks based on DHT. It also explains an existing bio-inspired model of self-organization in the hierarchy of wasps P. Dominulus.

A mathematical analysis of the proposed architecture will be held and published, as well as greater experimentation to attest to its benefits.

In the next version that will be implemented, we will include the functionality of allowing the WChord to perform searches for similar results and not just equal to the key name, functionality called semantic search.

A more detailed study in search of a better criterion to define how many peers will be part of the ring of WChord will be held.

It is also possible to evaluate the possibility of reducing the frequency in which the stabilization protocol runs on WChord, since they are more stable peers selected to be part of the ring.

As can be seen in the test results, the proposed architecture, WChord, brings benefits in relation to the Chord to provide an environment more tolerable to failures, minimizing the ring rupture risk, with less bandwidth consumption and consequently less overhead. On the other hand, the super-peers have a greater burden to maintain active connections to the peers connected to it.

References

[1] Fernandes, S. F. L.; Silvestre, G. S. Rocha-jr J.B. Sadok, D. Kamienski, C. "Análise de Tráfego P2P no Backbone da RNP", Anais do 22º. Simpósio Brasileiro de Redes de Computadores 2004, Gramado/RS, Maio 2004.

[2] Vasconcelos, E. (2009, Sep.) "Um exemplo de banco de dados distribuído".[Online]. Available: http://www.webartigos.com/articles/17334/1/um-exemplo-de-implementao-de-banco-de-dados-distribudo/pagina1.html

[3] Brito, G. A. D.D. Moura. A. M. C. "Sistema ROSA - P2P: uma Arquitetura Distribuída para Integração de Objetos de Aprendizagem", Anais do XVI Simpósio Brasileiro de Educação, 2005.

[4] Kozovits, L. E. Feijó, B. (2005, Dec.) "Arquiteturas para Jogos Massive Multiplayer". [Online]. Available: ftp://139.82.16.194/pub/docs/techreports/03_36_kozovits.pdf

[5] Wilcox-O'Hearn, B. "Experiences deploying a large-scale emergent network". In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, Março 2002.

[6] Ratnasamy, S. Francis, P. Handley, M. Karp, R. Schenker, K. "A Scalable Content Addressable Network", Proc. ACM SIGCOMM, 2001.

[7] Stoica, I.. Morris, R. Karger, D. Kaashoek, M. F. Balakrishnan, H. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", IEEE/ACM Trans. Net., 2001

[8] Rowstron, A. and Druschel, P. "Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-peer Systems", Proc. Middleware, 2001.

[9] Zhao, B. Y. Huang, L. Stribling, J. Rhea, S. C. Joseph, A. D.Kubiatowicz, J. D. "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE JSAC, 2004.

[10] Garces-Erice, L. Ross, K. W. Urvoy-Keller, G and Biersack, E.N. "Hierarchical peer-to-peer look-up services", INFOCOM2003, 2002.

[11] Freedman, M. Mazieres, D. "Sloppy hashing and self-organizing clusters". In 2st International Peer to Peer Systems Workshop, 2003.

[12] Cicirelle, V. A. Smith, S. F. "Wasp Nests for Self Configurable Factories". Canada: AGENTS'01 ACM. Maio/Junho, 2001.

[13] Nouyan, S. Ghizzioli, R., et al. "An InsectABased Algorithm for the Dynamic Task Allocation Problem". IDADIA – Technical Support Series, 2005.

[14] Bonabeau, E. Deneubourg, J. L. Theraulaz, G. "Response Threshold Reinforcements and Divisions of Labour in Insect Societies". Proc Biol Sci. 265(1393): 327–332. 1998.

[15] Bonabeau, E. Deneubourg, J. L. Heraulaz, G. "Self Organization of Hierarchies in Animal Societies: The Case of the Primitively Eusocial Wasps Polistes Dominulus Christ". Journal of Theoretical Biology, Volume 174, Number 3, pp. 313-323(11). 1995

[16] Bonabeau, E. Deneubourg, J. L. Sobkowski, A. Theraulaz, G.(2009, Dec.) "Adaptive Task Allocation Inspired By a Model of Division of Labor in Social Insects". [Online]. Available: http://www.agent.ai/download.php?ctag=download&docID=358

[17] Deneubourg, J. L. Gervet, J. Goss, S. Theraulaz, G. "Task Differentiation in Polistes Wasp Colonies: A Model for Self-AOrganizing Groups of Robots". Proceedings of the first [17] international conference on simulation of adaptive behavior on From animals to animats table of contents. Paris, France. Pages: 346 - 355. 1991.

[18] Bonabeau, E. et al. "Self Organization in Biological Systems". United States of America: Princeton Studies in Cloplexity, 2003. 2º ed.

[19] Chakaravarti, A. J. Baumgartner, G. Lauria, M. "The Organic Grid: Self Organizing Computation on a Peer to Peer Network". IEEE Transactions on Systems, Man and Cybernetics. Vol. 35, nº 3. Maio, 2005.

[20] Jelasity, M. Montresos, A. Jesi, G. P. Voulgaris, S. (2011, Sep.) "The PeerSim Simulator". [Online]. Available: http://peersim.sf.net