A Decentralized Ant-Inspired Approach for Resource Management and Discovery in Grids

Agostino Forestiero, Carlo Mastroianni, Giandomenico Spezzano
ICAR-CNR 87036 Rende (CS), Italy
{forestiero,mastroianni,spezzano}@icar.cnr.it

Abstract. This paper examines a decentralized and self-organizing approach inspired by ant behavior for building an information system in which metadata related to Grid resources is disseminated and logically organized. Each agent/ant, by analyzing its own past activity, copies and moves resource metadata among Grid hosts and contributes to collect resources belonging to the same class in a restricted region of the Grid, so decreasing the system entropy. A semi-informed resource discovery protocol exploits the ants' work: asynchronous query messages issued by clients are driven towards "representative peers" which maintain information about a large number of resources having the required characteristics. Agents control their activities, and query messages travel the network, according to self-organizing mechanisms based, respectively, on sematectonic and marker-based stigmergy, with no information about the global system state. Simulation analysis suggests that the combined use of the proposed resource mapping protocol (ARMAP) and resource discovery protocol (ARDIP) is profitable: as resources are progressively reorganized by the ARMAP process, users are able to find more and more results in a smaller amount of time.

Keywords: agents, autonomous systems, grid, peer-to-peer, resource discovery, resource replication, swarm intelligence.

1 Introduction

A Grid information system provides resource discovery and browsing services which are invoked by Grid clients when they need to use hardware or software resources matching given criteria and characteristics. In currently deployed Grids, for example in the Web Services Resource Framework (WSRF [24]), resources are available in the form of Grid services, i.e. Web services having enriched functionalities such as state and lifecycle management.

In most Grid/cluster systems (for example in Condor submission systems), users do not generally need to find a particular resource, but a number of resources belonging to a given *class*, so that they can subsequently select the resource which is the best suited for their application [6, 16]. A resource class can be seen as a set of resources satisfying a given set of syntactic and semantic constraints on the values of resource metadata parameters, for example of *WSRF Resource Properties*. Resource matchmaking can be realized by means of XML queries operated on XML metadata documents describing Grid services. A survey on the management of metadata for describing and discovering Grid resources and services can be found in [15].

Currently used Grid framework, for example those based on the Globus Toolkit, often adopt centralized or hierarchical information systems, which are not suitable for very large Grids for their intrinsic poor scalability features. The design of decentralized information systems is therefore urgent. An agent-based protocol (i.e. ARMAP, *Ant-based Replication and MApping Protocol*) was proposed in [9] for building and managing a decentralized Grid information system. ARMAP exploits the features of (i) epidemic mechanisms tailored to the dissemination of information in distributed systems [17, 5] and (ii) self adaptive systems in which a sort of "swarm intelligence" emerges from the work of a large number of agents which interact with the environment [3].

ARMAP disseminates metadata documents describing Grid resources in a controlled way, by spatially mapping such documents according to the semantic classification of resources. A metadata document is composed of a syntactical description of the service (i.e. a WSDL document) and/or an ontology description of service capabilities. By mapping metadata documents on Grid hosts it is possible to achieve a *logical reorganization* of resources. For the sake of simplicity, in the following an information document describing a Grid resource will be simply referred to as a *resource*. When distinction is important, the term *logical resource* or *resource descriptor* will be used, while the real resource described by the metadata document will be named *physical resource*.

With ARMAP, resources are replicated and mapped on the Grid so that resources belonging to the same class are placed in nearby Grid hosts. This is managed through a multi-agent approach that emulates the behavior of ants which cluster and map items within their environment [8]. It is here assumed that an underlying peer-to-peer (P2P) infrastructure interconnects Grid nodes and can be used by agents to explore the Grid [11]. ARMAP agents traverse the Grid and copy or move resources from one host to another, by means of appropriate *pick* and *drop* random functions.

A semi-informed discovery protocol (namely ARDIP, *Ant-based Resource DIscovery Protocol*) is proposed to exploit the logical resource organization achieved by ARMAP. The rationale is the following: if a large number of resources of a specific class are accumulated in a restricted region of the Grid, it becomes convenient to drive query messages (issued by peers to search for resources of that class) towards that region, in order to maximize the number of discovered resources and minimize the response time. An ARDIP discovery operation is performed in two phases. In the first phase, the random walk technique is adopted [14]: a number of asynchronous query messages are issued by the requesting peer and will travel the Grid in parallel through a *blind* mechanism. In the second phase, whenever a query message gets close enough to a Grid region which is collecting the needed class of resources, the search becomes *informed*: the query message will be driven towards this Grid region – more specifically, towards a *representative peer* that is selected within this region - and will easily discover a large number of useful resources.

The semi-informed ARDIP protocol aims at combining the benefits of both blind and informed resource discovery approaches which are currently used in P2P networks [22]. In fact, a pure blind approach (e.g. using flooding or random walk techniques) is very simple and scalable but has limited performance and can cause an excessive network load, whereas a pure informed approach (e.g. based on routing indices [6] or adaptive probabilistic search [21]) generally requires a very structured resource organization which is impractical in a large, heterogeneous and dynamic Grid.

Simulation analysis shows that ARMAP and ARDIP protocols can be profitably used together. In particular, as the resource mapping performed by ARMAP proceeds, a single query can find more and more useful resources while response times and traffic load decrease. Moreover, results allow for implicitly comparing the proposed approach with the classical random walk technique [14]; indeed if no reorganization of resources is performed, the ARDIP protocol can never enter the *informed* phase, so it becomes equivalent to the random walk paradigm.

An interesting and distinguishing feature of the approach proposed in this paper is that both ARMAP agents and ARDIP query messages control their activities and travel the network only on the basis of local information, without any knowledge of the global Grid state. In particular, a pheromone-based decentralized mechanism is used by ARMAP agents to choose their modality (*copy* or *move*) and by ARDIP query messages to choose their search policy (*blind* or *informed*). The self-organizing and decentralized nature of the involved algorithms, along with the analysis of performance results obtained with variable Grid sizes, seem to suggest that the proposed approach is scalable and can be adopted in a Grid framework regardless of the Grid size.

The remainder of the paper is organized as follows. Section 2 illustrates the ant-based approach for the management and discovery of resources in Grids. In Section 2.1, the key points of the ARMAP protocol are summarized (more details can be found in [9]) while Section 2.2 describes the mechanism used by ARMAP to deal with the dynamic nature of the Grid environment. Section 2.3 introduces the new ARDIP protocol and focuses on the three modules which compose ARDIP: (i) the module for the selection of representative peers, (ii) the semi-informed search mechanism and (iii) the pheromone mechanism used by query messages to exchange information about the presence of representative peers. Section 3 focuses on the discussion of simulation results, which demonstrate the effectiveness of the ARDIP resource discovery protocol in a Grid environment, if it is used in conjunction with the ARMAP resource mapping protocol. In particular, Section 3.1 introduces the main system and protocol parameters, Section 3.2 illustrates the performance results related to the ARMAP protocol, and Section 3.3 focuses on the performance of the ARDIP protocol. The paper discusses related work in Section 4 and concludes in Section 5.

2 Ant-Based Protocols for the Management and Discovery of Resources

2.1 The ARMAP protocol

The aim of the ARMAP protocol (*Ant-based Replication and MApping Protocol*) is to achieve a logical organization of Grid resources by spatially mapping them on the Grid according to their semantic classification. It is assumed that resources have been previously classified into a number of classes `Nc`, according to their semantics and functionalities (see [6] and [16]). ARMAP exploits the random movements and operations of a number of mobile agents that travel the Grid using P2P interconnections. This approach is inspired by ant systems [3, 7, 8], in which swarm intelligence emerges from the activity of a large number of very simple mobile agents (ants), and a complex overall objective is achieved.

The ARMAP protocol has been analyzed in a P2P network in which peers are arranged in a 2-dimension toroidal space, and each peer is connected to at most 8 neighbor peers. Such a mesh topology was chosen to obtain an intuitive and immediate graphical representation of system evolution, which helps to understand the involved dynamics. However, it is also assumed that peers can leave and re-join the network, as discussed in Section 2.2; therefore at a specific time a peer is actually connected to a random number of active peers (at most 8), so relaxing the mesh assumption.

The ARMAP process is driven by the *pick* and *drop* probability functions [9], as briefly discussed in the following.

Pick operation

Once an agent specialized in a given class gets to a Grid host, if it is currently unloaded, it must decide whether or not to pick the resource descriptors of that class which are managed by that host. In order to achieve the replication and mapping functionalities, a *pick* random function `Ppick` is defined with the intention that the probability of picking the resource descriptors of a given class decreases as the local region of the Grid accumulates such descriptors. This way resource mapping is further facilitated. The `Ppick` random function, defined in formula (1), is the product of two factors, which take into account, respectively, the relative accumulation of resource descriptors of a given class (with respect to the other classes), and their absolute accumulation (with respect to the initial number of resource descriptors of that class).

$$(1)\ P_{pick} = \left(\frac{k_1}{k_1 + f_r}\right)^2 \cdot \left(\frac{(f_a)^2}{k_2 + (f_a)^2}\right)^2$$

In particular, the fr fraction is computed as the number of resource descriptors of the class of interest, accumulated in the peers located in the *visibility region*, divided by the overall number of resource descriptors that are accumulated in the same region. The visibility region includes all the peers that are reachable from the current peer with a given number of hops (i.e. within the *visibility radius*). Here it is assumed that the visibility radius is equal to one, so that the visibility region is composed of at most 9 hosts, the current one included. The fa fraction is computed as the number of physical resources owned by the hosts located in the visibility region out of the number of resource (both logical and physical) that are maintained by such hosts, including the descriptors deposited by the agents. The inverse of fa gives an estimation of the extent to which such hosts have accumulated resource descriptors of the class of interest. k1 and k2 are non-negative constants which are both set to 0.1 [3]. If the *copy* modality is used, the agent, when executing a *pick* operation, leaves the logical resources on the current host, generates a replica of them, and carries the replicas until it will drop them in another host. Conversely, with the *move* modality, as an agent picks the resources, it removes them from the current host, thus preventing an excessive proliferation of replicas.

Drop operation

Whenever an agent specialized in a given class gets to a new Grid host, it must decide whether or not to drop the resource descriptors of that class, in the case that it is carrying any of them. As opposed to the *pick* operation, the dropping probability is directly proportional to the relative and absolute accumulation of resource descriptors of the class of interest in the visibility region. The Pdrop function is shown in formula (2); the threshold constants k3 and k4 are set to 0.3 and 0.1, respectively.

$$(2)\ P_{drop} = \left(\frac{f_r}{k_3 + f_r}\right)^2 \cdot \left(\frac{k_4}{k_4 + (f_a)^2}\right)^2$$

System entropy

A system entropy function, defined in [9] and reported in formula (3), evaluates the effectiveness of the ARMAP protocol in the spatial mapping of resources. For each peer `p`, the entropy `Ep` gives an estimation of the extent to which the visibility region centered in `p` has accumulated resource replicas belonging to a single class. This function, inspired by the Shannon entropy formula, is constructed upon the value of `fr(i)`, defined as the fraction of resources of class `Ci` within the visibility region. The `Ep` formula is normalized, so that possible values are comprised between 0 (if all the resources in the visibility region belong to just one class) and 1 (if the resources are equally distributed among the different classes). The overall spatial entropy `E` of the network is defined as the average of the entropy values `Ep` computed at all Grid hosts.

$$(3)\quad \mathrm{Ep} = \frac{\sum_{\mathrm{i=1..Nc}} \mathrm{fr(i)} \cdot \log_2 \frac{1}{\mathrm{fr(i)}}}{\log_2 \mathrm{Nc}}, \quad \mathrm{E} = \frac{\sum_{\mathrm{p}\varepsilon\mathrm{Grid}} \mathrm{Ep}}{\mathrm{Np}}$$

The overall entropy can be minimized if each agent exploits both the ARMAP modalities, i.e. *copy* and *move*. In a first phase, the *copy* modality is used to generate an adequate number of resource replicas on the network. However, the *copy* modality cannot be maintained for a long time, since eventually every host would have a huge number of resources of all classes, thus weakening the efficacy of resource mapping. Conversely, if the modality switch is performed when the entropy begins to increase, then the agents can continue their mapping work and the entropy decreases to a much lower value. Therefore, each agent at a certain time must switch from the *copy* to the *move* modality. An approach that allows for determining the correct time at which this modality switch must be performed is described in [9]. This approach is based on the observation that a decreasing trend of agents' activity (i.e. of the frequency of *pick* and *drop* operations that they perform) is an indication that the overall entropy is beginning to increase. This led to the definition of a decentralized and self-organizing mechanism, inspired by ants' pheromone [7], through which each agent can tune its modality by itself, analyzing its own past activity. It must be remarked that the use of a decentralized mechanism is a key feature and is indeed a requisite for any resource management algorithm designed for a distributed environment like a Grid. In particular, it would not be possible to guarantee the scalability of the ARMAP protocol if agents had to acquire a global knowledge of the system state in order to properly tune their behavior.

2.2 Managing a dynamic Grid with ARMAP

In a dynamic Grid, peers can, more or less frequently, go down and connect again. As a consequence of this dynamic nature, two different and opposite issues must be tackled. The first is related to the management of new resources provided by new or reconnected hosts: if all the agents switch to the *move* modality, it becomes impossible to replicate and disseminate information about the new resources; hence agents cannot live forever, and must be gradually replaced by new agents that set off in the *copy* modality. At the same time, the system must deal with obsolete resources, i.e. with resources provided by peers that have left the system, that are no longer exploitable.

To tackle these two issues, it is necessary to manage the lifecycle and the gradual turnover of agents, and control the overall number of agents that travel the Grid. The proposed solution is to correlate the lifecycle of agents to the lifecycle of peers. An assumption is made that the average amount of time for which a peer remains connected to the Grid is known and equal to `PlifeTime`[*]. When joining the Grid, each peer generates a number of agents given by a discrete Gamma stochastic function, with average `Ngen`, and sets the life time of these new agents to `PlifeTime`. This setting assures that (i) the relation between the number of peers and the number of agents is maintained over the time (more specifically, the overall number of agents is approximately equal to `Ngen` times the number of active peers) and (ii) a proper turnover of agents is achieved, which allows for the permanent dissemination of logical resources, since new agents start with the *copy* modality. Furthermore a peer, when leaving the Grid, loses information about the logical resources that it has collected during its connection time. This solves the problem of managing obsolete resources: indeed the replicas related to an obsolete resource are gradually eliminated by the Grid, as the peers that are storing such replicas leave the system and the agents that are carrying them expire.

[*] In this paper we assume for simplicity that the average connection time is the same for all the peers. However, in a real system it would only be required that each peer knows *its own* average connection time. This estimation can approximately be done on the basis of peer’s past activity.

2.3 The ARDIP protocol

The ARDIP (*Ant-based Resource DIscovery Protocol*) protocol is used by clients to discover Grid resources having specific characteristics, i.e. belonging to a given class. The objective of ARDIP is to drive a query message towards a region of the Grid in which the needed class of resources is being accumulated. Because ARDIP fully exploits the replication and spatial mapping of resources achieved by ARMAP, the two protocols should be used together: as ARMAP agents perform the logical reorganization of resources and build the Grid information system, the number of useful resources that can be discovered by a query message increases and at the same time the query response time decreases, as shown in Section 3.3. The ARDIP protocol is based upon three modules: (a) a module for the identification of *representative peers* which work as attractors for query messages; (b) a module which defines the semi-informed search algorithm; (c) a *stigmergy* mechanism that allows query messages to take advantage of the positive outcome of previous queries. These modules are described in the following.

Identification of representative peers

As resources of a given class are accumulated in a Grid region, the peer that, within this region, collects the maximum number of resources belonging to that class is elected as a *representative peer* for that class. The objective of a search operation is to let a query message get to a representative peer in an amount of time as low as possible, since such a peer, as well as its neighbors, certainly manages a large number of useful resources. Each peer periodically verifies if it can elect itself as a representative peer for a class of resources. The ARDIP protocol assumes that a peer is a representative peer of a given class if the following condition is satisfied:

- the peer maintains a number of *logical* resources of that class that exceeds Fg times the mean number of *physical* resources belonging to the same class; hereafter, Fg is referred to as *gathering factor*.

Moreover, to limit the number of representative peers in the same region, each representative peer periodically checks if other representative peers are present in its neighborhood, specifically within the *comparison radius* Rc: two neighbor representative peers must compare the number of resources that they maintain, and the "loser" will be downgraded to a simple peer.

Semi-informed search

When a client or user needs to discover resources belonging to a given class, a number of asynchronous query messages are issued by ARDIP. The semi-informed search algorithm includes a *blind* search phase and an *informed* search phase. For the blind search phase, the random walk paradigm [14] is used: query messages travel the Grid through P2P interconnections following a random path. The network load is limited by means of a caching mechanism that avoids the formation of cycles and the use of a time-to-live parameter `TTL`: the `TTL` value is equivalent to the maximum number of hops that can be performed by a query message before being discarded. The blind search procedure is switched to informed as soon as one of the query messages approaches a representative peer of the class of interest, i.e. when such a message is delivered to a peer which knows the existence of a representative peer in the neighborhood and knows a route to it (see the description of the stigmergy module below). During the informed search phase, the query is driven towards the representative peer, and the `TTL` parameter is ignored so that the query cannot be discarded until it actually reaches the representative peer. The semi-informed walk of a query message ends in one of the two following cases: (i) when the `TTL` is decremented to 0 during the blind phase; (ii) when the query reaches a representative peer. In both cases a queryHit message is created, and all the resources of the class of interest, which are found in the current peer, are put in this message. The queryHit follows the same path back to the requesting peer and, along the way, collects all the resources of the class of interest that are managed by the peers through which it passes.

Stigmergy mechanism

The French entomologist Grassé coined the term "stigmergy" in the 1950's [10] to describe a broad class of multi-agent coordination mechanisms that rely on information exchange through a shared environment. For example, in ant colonies, each ant adjusts its activity according to the state of the environment, which in turn can be modified by the activity of other ants. An ant that finds a food source leaves a pheromone along its way back to the nest, and such a pheromone will signal to other ants the presence of food. The term is formed from the Greek words *stigma* “sign” and *ergon* “action”, and captures the notion that an agent’s actions leave signs in the environment, signs that it and other agents sense and that determine their subsequent actions. Two varieties of stigmergy are distinguished [4]. Such a distinction is whether the signs consist of special markers that agents deposit in the environment (“marker-based stigmergy”) or whether agents base their actions on the current state of the environment (“sematectonic stigmergy”).

The approach proposed in this paper uses both types of stigmergy. Indeed sematectonic stigmergy is used by ARMAP agents to choose their operation modality (*copy* or *move*), as mentioned in Section 2.1. Conversely marker-based stigmergy is exploited by the ARDIP protocol: when a query accidentally gets to a representative peer for the first time, the returning queryHit will deposit an amount of pheromone in the peers that it encounters as it retreats from the representative peer. In this work, it is assumed the pheromone is deposited only in the first two peers of the queryHit path. A pheromone base is maintained for each resource class, and information is given about the right direction (i.e. the next neighbor peer) to get to the representative peer of the desired class. Accordingly, when a query gets to a peer along its *blind* search, it checks the amount of pheromone which has been deposited in that peer for the resource class of interest; if the pheromone exceeds a threshold `Tf`, it means that a representative peer is close, so the search becomes *informed* and the query is driven towards the representative peer. Since the set of representative peers can change as the ARMAP clustering process proceeds, an evaporation mechanism is defined to assure that the pheromone deposited on a peer does not drive queryHits to ex-representative peers. The pheromone level at each peer is computed every time interval of 5 minutes, which is equal to the mean time interval between the issues of two successive query messages from a peer (see Table 1 in Section 3.1). The amount of pheromone Φ_i, after the i-th time interval, is given by formula (4).

(4) $\Phi_i = E \cdot \Phi_{i-1} + \varphi_i$

The evaporation rate `Ev` is set to 0.9; φ_i is equal to 1 if a pheromone deposit has been made in the last time interval by at least one agent, otherwise it is equal to 0. The pheromone level can assume values comprised between 0 and 10: the superior limit can be obtained by equalizing Φi to Φ_{i-1} and setting φ_i to 1 in formula (4). The threshold `Tf` is set to 2.

With these parameters, it is assured that the threshold is exceeded as soon as a few number of queryHits deposit their pheromone in different time intervals, while the algorithm is more conservative when it has to recognize that a representative peer has been “downgraded”: indeed, up to 15 time intervals are necessary to let such a level assume a value lower than the threshold.

It can also happen, though quite rarely, that a peer collects pheromone deposited by two (or more) different queryHits which are related to the same resource class but come from two (or more) different representative peers. To which neighbor should a query for that resource class be forwarded? In such cases the peer maintains a different pheromone quantity for each neighbor. The query is forwarded to the neighbor peer associated to the higher amount of pheromone, provided that it succeeds the threshold. It corresponds to sending the query to the “oldest” representative peer, which intuitively is most likely the representative peer that has collected the largest number of resources so far.

3 Performance Analysis of ARMAP and ARDIP Protocols

3.1 Description of the environment

A wide set of simulation runs were performed by exploiting a Java event-based object-oriented simulator, implemented by the paper’s authors, that has already been used for other research issues [16]. In the simulator, both peers and agents are associated to simulation objects which communicate among them through *events*. Such events are ordered on the basis of their expiration time, i.e. the time at which they have to be delivered to destination objects. When an event is received by an object, the latter (e.g. a peer) reacts according to its automaton, and may send other events to other objects (e.g. a peer forwards an ARDIP query message to neighbor peers). Furthermore, the simulator was extended in order to integrate and exploit a set of libraries offered by the SWARM environment [19]: graphical libraries which allowed for monitoring the behavior of the ARMAP and ARDIP protocols, and libraries for setting the network topology. For each simulation session, 10 simulation runs were performed and, by tuning the length of each run, all performance values were computed with a pre-determined statistical relevance, i.e. with at least 0.95 probability that the statistical error was below 5%.

Table 1 reports the main parameters used in the simulation analysis, categorized in network parameters, ARMAP parameters and ARDIP parameters. The number of peers `Np` (or Grid size) is set to 2500, and each peer is connected to a maximum of 8 neighbor peers, as discussed in Section 2. Grid resources are classified into `Nc` different classes, with `Nc` set to 5 in this work. The number of physical Grid resources owned and published by a single peer is determined through a Gamma stochastic function having an average value equal to 15 (see [11]). Such resources are uniformly distributed among the `Nc` classes.

Table 1. Network, ARMAP and ARDIP parameters

Grid Parameter	**Symbol**	**Value**
Grid size (number of peer)	**`Np`**	2500
Maximum number of neighbor peers	*not used*	8
Mean number of resources published by a peer	*not used*	15
Number of resource classes	**`Nc`**	5
ARMAP Parameter	**Symbol**	**Value**
Number of ARMAP agents	**`Na`**	**`Np`**/2
Mean life time of a peer	**`Plifetime`**	100,000 s
Mean time interval between two successive movements of an ARMAP agent	*not used*	60 s
Maximum number of hops for each ARMAP agent's movement	**`Hmax`**	3
Visibility radius	**`Rv`**	1
ARDIP Parameter	**Symbol**	**Value**
Mean query generation frequency	*not used*	1/300 (1/s)
Number of query messages generated when issuing a query	**`Nqm`**	2-8
Time to live	**`TTL`**	3-7
Gathering factor for the identification of representative peers	**`Fg`**	24-48
Comparison radius for the identification of representative peers	**`Rc`**	2
Mean message processing time	*not used*	50 ms
Mean link delay	*not used*	50 ms

ARMAP parameters are defined in the second section of Table 1; their impact was analyzed in [9]. Each peer, whenever connecting to the Grid, spawns a random number of ARMAP agents; this number is generated by a Gamma random function the average of which is equal to Pgen. In this work, unless otherwise stated, Pgen is set to 0.5, so that the number of agents Na is approximately equal to half the number of peers Np. The average connection time of a peer (PlifeTime) is set to 100,000 s. At random times, specifically every 60 s on average, each agent moves in the P2P network (each movement is composed of a number of hops not greater than Hmax, which is set to 3), and possibly performs a pick or drop operation, as described in Section 2.1. The visibility radius Rv, defined in Section 2.1, is set to 1.

ARDIP parameters are defined in the last section of Table 1. When a peer issues a query for a class of resources (queries are uniformly distributed among the Nc classes), a number Nqm of asynchronous query messages, with Nqm set to values ranging from 2 to 8 in different simulation sessions, are forwarded to neighbor peers. They follow different directions and move in parallel through the Grid according to the semi-informed mechanism described in Section 2.3. The TTL parameter was varied from 3 to 7. Table 1 also specifies the values of the gathering factor Fg and of the comparison radius Rc, used to elect the representative peers. Finally, the mean amount of time for processing a query or queryHit message and the mean link delay between two neighbor peers were both assumed to be 50 ms on average, with Gamma statistical distributions.

3.2 Performance of the ARMAP protocol

A graphical description of the mapping process performed by ARMAP is shown in Figure 1: the 50x50 grid represents the simulated Grid, and each cell represents a peer. Note that the network is wrapped, i.e. the nodes on the left (upper) edge are “adjacent” to the nodes on the right (lower) edge. The values of system parameters are set as specified in Section 3.1, except for the number of resource classes which is set to 3 in order to facilitate the graphical illustration of the process. Circle, square and cross symbols correspond to class C_1, C_2 and C_3, respectively. Each peer is marked with a circle (square, cross) if in the corresponding peer the number of resources of class C_1 (C_2, C_3) exceeds the number of resources belonging to the other 2 classes. Furthermore, the thickness of the symbol is proportional to the number of resources of the most numerous class. To depict the gradual accumulation of resources, four snapshots of the network are shown. Such snapshots are taken at time 0 (when the ARMAP protocol is started), and after 25,000, 50,000 and 100,000 seconds, respectively. It can be observed that resources belonging to the same class are collected in nearby Grid hosts, and specialized regions emerge as the ARMAP process goes on.

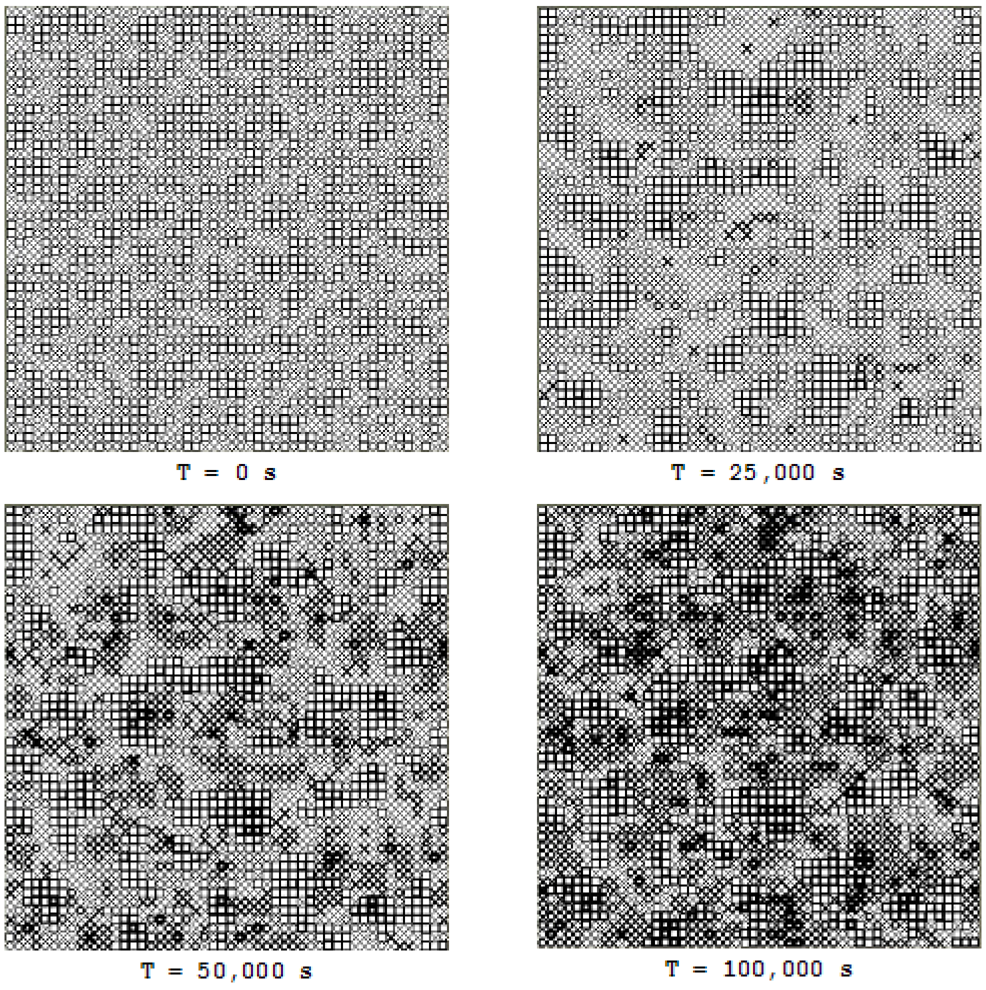

Fig. 1. Gradual mapping of logical resources in a network with 2,500 peers and 3 resource classes. Each peer contains a symbol (circle, square or cross) that corresponds to the most numerous class of resources contained in such a peer. The symbol thickness is proportional to the number of resources of the dominant class.

The performance of the ARMAP protocol was also proved by evaluating a set of performance indices which are reported in Table 2. The overall entropy `E`, defined in Section 2.1, is the most important index, because the primary objective of ARMAP is the logical reorganization of resources, which corresponds to decreasing the system entropy. The `Nrpp` index is defined as the average number of resources that are managed by a Grid peer at a given time. `Fop` is the frequency of operations (pick and drop) that are performed by each agent; this index gives an estimation of both agents' activeness and system stability, since operations are less frequent when a low level of entropy has already been achieved. Two indices are defined to evaluate the overhead related to the use of mobile agents: the traffic load `L`, defined as the number of hops per second that are globally performed by all the agents, and the mean agent memory `Mag`, defined as the average amount of data which is carried by an ARMAP agent, under the assumption that a resource metadata document requires 1 Kbytes of data.

Table 2. Performance indices for the ARMAP protocol

Performance Index	**Definition**
Mean Spatial Entropy, **`E`**	Average of the entropy values computed at all Grid hosts, see Section 2.1
Mean number of resources per peer, **`Nrpp`**	Mean number of resources that are maintained by a Grid host
Mean frequency of operations, **`Fop`**	Mean number of pick or drop operations that are performed by a single agent per unit time (operations/s)
Traffic load, **`L`**	Average number of hops performed by ARMAP agents per unit time (hops/s)
Mean agent memory, **`Mag`**	Average amount of data that is stored and carried by an ARMAP agent (Kbytes)

Figure 2 gives an overview on the work of the ARMAP protocol, on the assumption that the mapping operations are started at time 0. The trends of the first three performance indices listed in Table 2 are depicted in this figure: proper multiplication/reduction factors, as detailed in the figure legend, are used in order to depict such performance indices within the same chart. It can be noted that both the entropy value and the mean number of resources vary with a very fast rate in the first phase (in which most agents work with the *copy* modality) and with a slower rate in the second phase (in which most agents work with the *move* modality). As a result of the copy/move modality switch, the entropy is a decreasing monotonic function. Conversely, in [9] it was shown that without this switch, the entropy would decrease to a certain extent, than it would increase again because of an excessive proliferation of resource replicas. The operation frequency initially increases, than it begins to decrease when the mapping process reaches an advanced stage, since the pick and drop functions are activated with lower and lower probabilities. The trend shown in Figure 2 is the value averaged for all the active agents, but it was found that every single agent experiences a similar trend starting from the time at which it is generated by a peer that joins to the Grid. A self-organization approach based on ants' pheromone enables each agent to perform the modality switch (from *copy* to *move*) by itself, only on the basis of local information. More specifically, each agent maintains a pheromone base which increases as the pick and drop operations that it performs become more and more infrequent. When a proper pheromone threshold is reached, the agent switches to the *move* modality, thus preventing an excessive proliferation of replicas. Further details about this self-organizing mechanism can be found in [9].

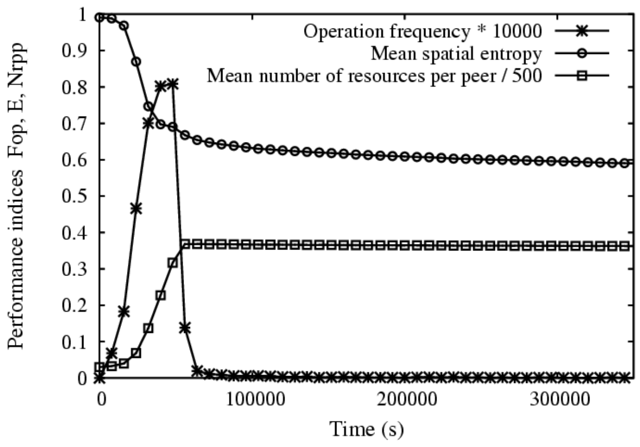

Fig. 2. Mean spatial entropy, mean number of resources per peer and operation frequency vs. time; the copy/move modality switch is activated by each agent according to a pheromone mechanism.

To evaluate the scalability characteristics of the ARMAP protocol, performance indices were computed for different values of the Grid size, i.e. of the number of peers Np. Figures 3 and 4 are related, respectively, to the overall entropy and the average number of resources per peer. It can be seen that both indices are hardly affected by the Grid size: the curves obtained for different numbers of peers are almost identical and the differences among them are smaller than, or comparable with, the statistical confidence interval. This suggests that a single peer does not need to have knowledge about the size of the network to regulate its behavior, which of course is a strong requisite for a decentralized system.

Figure 5 reports the trend of the overall system entropy obtained with different numbers of agents (i.e. with different values of Pgen). It is observed that an increase in the number of agents makes the system entropy decrease faster. Moreover, a higher activity of agents also causes an increase in the traffic load (Figure 6). When observing the average amount of data stored by an agent (Figure 7), it is interesting to note that a lower number of agents does not notably increase the maximum value of this index, but does increase the duration of the time interval in which agents must carry a significant amount of data.

Therefore, a correct setting of the ratio Na/Np should take into account the trend of these performance indices and in general should depend on system features and requirements, for example on the system capacity of sustaining a high traffic load. However, Figure 5 suggests that this setting can increase the rapidity of the first phase of the ARMAP process, but it does not have a significant impact on the value of system entropy that is reached in a stable situation. Hence, the stable reorganization of resources is anyhow guaranteed.

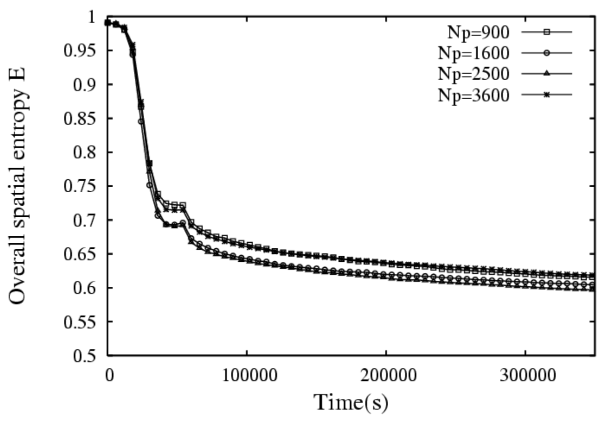

Fig. 3. Overall spatial entropy vs. time, with different values of the network size Np.

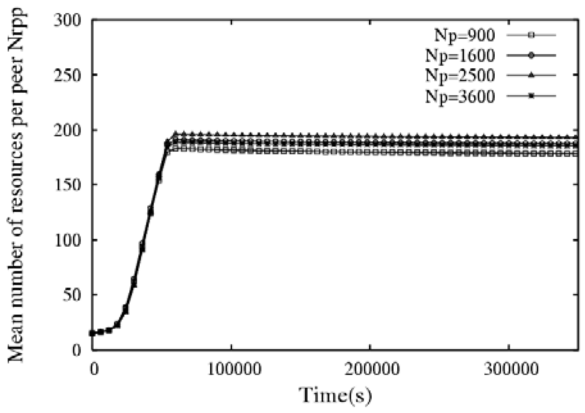

Fig. 4. Mean number of resources per peer vs. time, with different values of the network size Np.

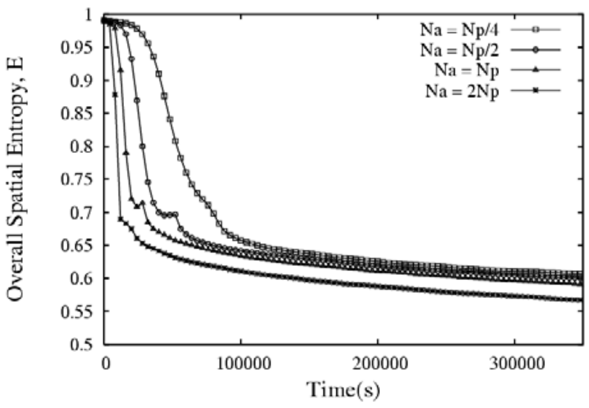

Fig. 5. Mean spatial entropy vs. time, with different numbers of ARMAP agents.

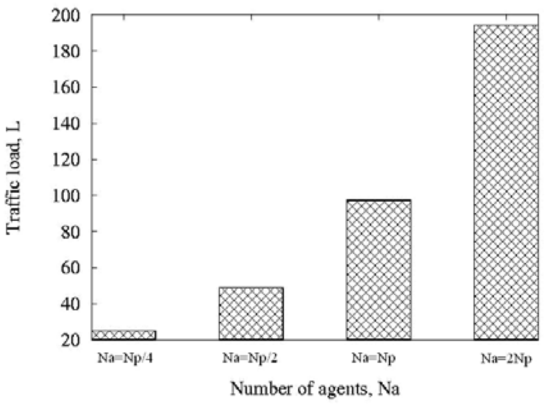

Fig. 6. Mean traffic load generated by ARMAP agents (hops/s), with different numbers of agents.

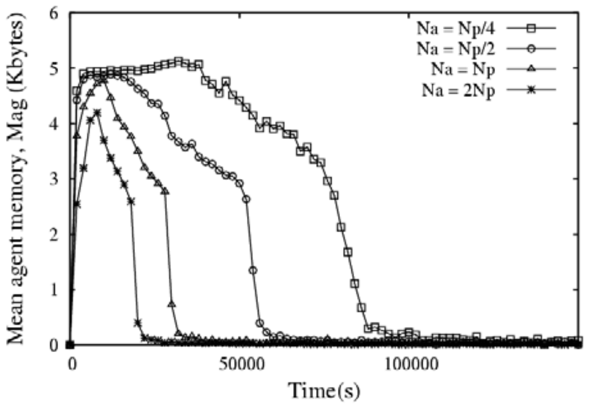

Fig. 7. Mean amount of data carried by an ARMAP agent (Kbytes) vs. time, with different numbers of agents.

3.3 Performance of the ARDIP protocol

The performance of the ARDIP protocol was analyzed by evaluating the performance indices defined in Table 3. In the following such indices are plotted w.r.t. time, to evaluate the expected outcome of a discovery request at different stages of the ARMAP process. It is worth highlighting that results obtained at time 0 are achieved without exploiting the ARDIP *informed* phase, because no representative peers have been yet selected. Therefore it is possible to compare the ARDIP protocol with the classical random walk technique, since in the ARDIP *blind* phase the random walk paradigm is adopted.

The `Nrep` index is the number of representative peers of all classes that are selected by ARDIP with the mechanism described in Section 2.3. Figure 8 depicts the trend of this index as the ARMAP mapping process proceeds. Curves obtained with different values of the *gathering factor* `Fg` are compared: since `Fg` is a threshold used to elect the representative peers, obviously its increase causes a decrease in the number of representative peers. The gathering factor is hereafter set to 36. With this value, `Nrep` converges to a value slightly higher than 200, corresponding to about 40 representative peers (out of 2500 peers) per class. A snapshot of the state of the network was taken 300,000 seconds after the start of the ARMAP process, because at that time the system has reached a stable condition, as can be observed in previous figures. Figure 9 shows the representative peers for two of the five resource classes, and confirms that representative peers are located in the core of the respective accumulation regions and that nearby peers are able to drive query messages to a representative peer. The presence of a significant number of representative peers of the same class guarantees a high probability of reaching at least one of those in a limited amount of time.

The index `Fsq` is defined as the fraction of queries for which at least one of the `Nqm` issued query messages enters the informed phase and gets to a representative peer. In the following, a query message that reaches a representative peer, as well as the related discovery request, are referred to as *striking (query) message* and *striking query*, respectively. Since the ARDIP discovery protocol aims at driving query messages towards representative peers, the index `Fsq` is essential to evaluate the extent to which the replication and organization of resources performed by ARMAP helps to increase the effectiveness of ARDIP.

Table 3. Performance indices for the ARDIP protocol

Performance Index	**Definition**
Number of representative peers, **`Nrep`**	Number of representative peers of all classes that are selected by ARDIP to attract query messages
Fraction of striking queries, **`Fsq`**	Fraction of discovery requests that are successfully driven towards a representative peer
Mean number of results, **`Nres`**, **`Nres(stk)`**, **`Nres(no-stk)`**	Mean number of resources that a node discovers after its query (computed for all the queries and separately for striking and non-striking queries)
Average response times, **`Tavg`**, **`Tavg(stk)`**, **`Tavg(no-stk)`**	Mean amount of time that elapses between the generation of a query and the reception of a corresponding queryHit (computed for all the queries and separately for striking and non-striking queries)
Response times of the first queryHit, **`Tfst`**, **`Tfst(stk)`**, **`Tfst(no-stk)`**	Mean amount of time that elapses between the generation of a query and the reception of the *first* corresponding queryHit (computed for all the queries and separately for striking and non-striking queries)
Query traffic load, **`Lq`**, **`Lq(rep)`**	Mean frequency of query messages (messages/s) that are received by a peer (calculated for all the peers and for the representative peers only)

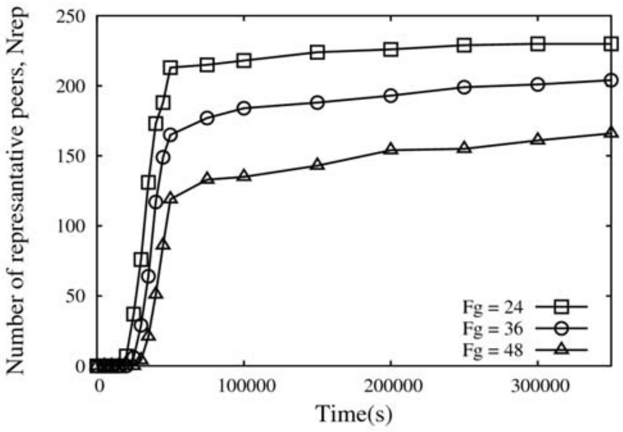

Fig. 8. Selection of representative peers as the ARMAP mapping process proceeds. The number of representative peers is reported for different values of the gathering factor Fg.

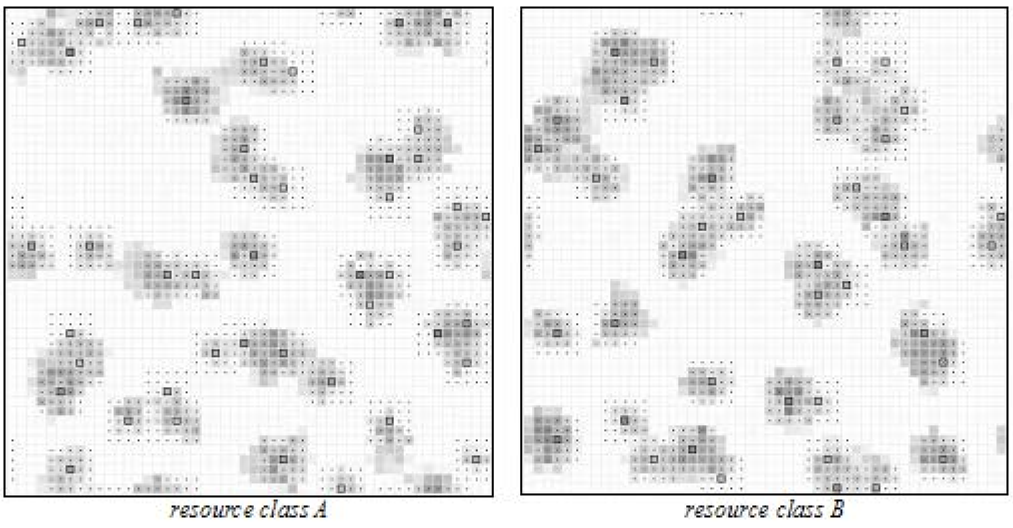

Fig. 9. Representative peers, belonging to two classes (out of five), selected by ARDIP 300,000 seconds after the start of the ARMAP process, with a gathering factor Fg set to 36. In the two figures, the squares are the representative peers respectively belonging to classes A and B, the dots are the peers that drive query messages to a nearby representative peer, and the number of resources of the respective classes is represented with a grey scale.

Figures 10 and 11 confirm the valuable effect caused by the combined use of ARMAP and ARDIP protocols. In fact, after a very small amount of time, the work of ARMAP produces a significant increase in Fsq. Results in Figure 10 are obtained by setting the parameter Nqm to 4 and varying the TTL parameter from 3 to 7, whereas in Figure 11 TTL is set to 5 and Nqm ranges from 2 to 8. Figure 10 shows that Fsq increases with the value of TTL, since a discovery request can extend the blind search phase and has more chances to get to a representative peer. An analogous effect is achieved by increasing the Nqm parameter, since representative peers are searched by a larger number of query messages in parallel.

The most important performance measure is obviously Nres, the mean number of results that are discovered by a query – to be precise by all the query messages generated by a single discovery request - issued to search resources belonging to a specific class. Indeed, it is generally argued that the *satisfaction of the query* depends on the number of discovered resources returned to the user that issued the query: for example, in [26] a resource discovery operation is considered *satisfactory* only if the number of results exceeds a given threshold. Figure 12 shows the number of results that are collected on average by striking queries, by non-striking queries, and by all the queries, for two different values of TTL and Nqm set to 4. Many interesting conclusions can be drawn by observing this figure. The most obvious one is that the number of results increases with the TTL value, since a larger fraction of the network can be explored; however response times and traffic increase as well, as shown later. More important, the mean number of results increases as resources are being organized by ARMAP, meaning that specialized regions are able to offer a significant number of resources. Moreover, it is noted that striking queries can discover considerably more results than non-striking queries, and such a difference increases with time, mostly because the progressive clustering of resources reduces the number of results that can be collected by non-striking queries (which soon become very rare, as Figure 11 shows). Similar considerations raise from the observation of Figure 13, in which the TTL value is set to 7 and the number of query messages ranges from 2 to 8. The number of results increases with the Nqm parameter, but the traffic load increases as well, as will be shown in the following.

The use of the ARMAP/ARDIP protocols not only increases the number of results, but also allows users to discover them in a shorter amount of time. Figure 14 and 15 show the average response times experienced when varying the TTL value and the parameter Nqm, respectively. From Figure 14, it appears that a higher TTL causes an increase in the response time, due to the fact that the queryHit messages have to travel a longer path. Whereas the response time of non-striking queries is almost constant over time, since all the query messages exploit the entire value of the `TTL` parameter, the reorganization of resources produces a significant decreases of average response times for striking queries and, since the relative weight of striking queries increases, for generic queries as well. In fact, when a query message gets to a representative peer, the discovery operation is stopped even if the `TTL` value is still greater than 0, and a queryHit is immediately issued. Therefore the presence of a striking message decreases the average response time. This can be also observed in Figure 15: the average response time of a striking query decreases with the value of `Nqm` because the relative impact of the response times experienced by striking messages is higher for a lower number of query messages.

Figures 16 and 17 depict the values of the response times corresponding to the first received queryHit, obtained by varying the `TTL` value and the parameter `Nqm`, respectively. These response times are significantly lower than average response times (Figures 14-15), due to two phenomena. The first is related to the statistical nature of message processing time (the minimum processing time is lower than the average), and has impact on all the queries; the second, which has impact only on striking queries, comes from the fact that the first queryHit most likely corresponds to a striking query message, which generally performs a lower number of hops. It is also interesting to note, from Figure 17, that the time of the first result decreases as the `Nqm` parameter increases, which is different to what happens on average response times. The reason is that a larger number of query messages allows for a more massive exploration of the Grid and hence a faster discovery of representative peers.

The query traffic load, defined in Table 3, is depicted in Figures 18 and 19. It increases with the `TTL` value and with the number of query messages `Nqm`. Since an increase in these two parameters also allows for achieving a larger number of results, it is necessary to reach a trade-off that takes into account the expected number of results and the processing load that a Grid node can undergo. However, a very interesting consideration is that over time the logical reorganization of resources and the use of the ARDIP protocol allow for decreasing the query traffic load experienced by a single peer. Indeed, when a query messages is driven towards a representative peers, on average it performs a lower number of hops than that experienced with a blind search. A further benefit is that a representative peer experiences an even lower query load than a generic peer. In fact, when a query is issued by a peer which is adjacent to a representative one, the route to the representative peer is most likely known in advance, thanks to the pheromone mechanism described in Section 2.3, and the blind search phase can be entirely skipped. In such a case, only one query message is generated by the requesting peer instead of Nqm (in fact all the Nqm queries would follow the same informed path that leads to the adjacent representative peer and would collect the same results), thus reducing the query traffic load at the nearby representative peer.

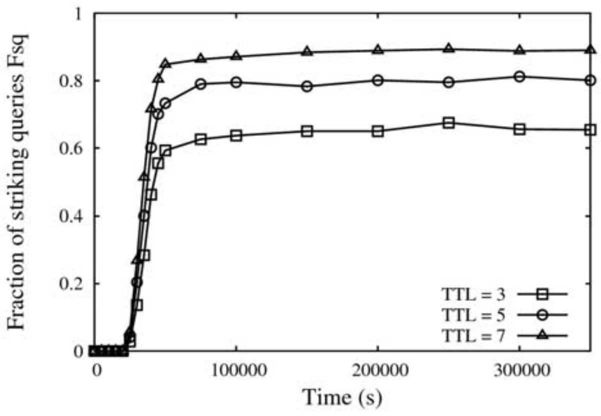

Fig. 10. Use of ARMAP representative peers: fraction of discovery requests that are successfully driven to a representative peer, for different values of TTL and Nqm set to 4.

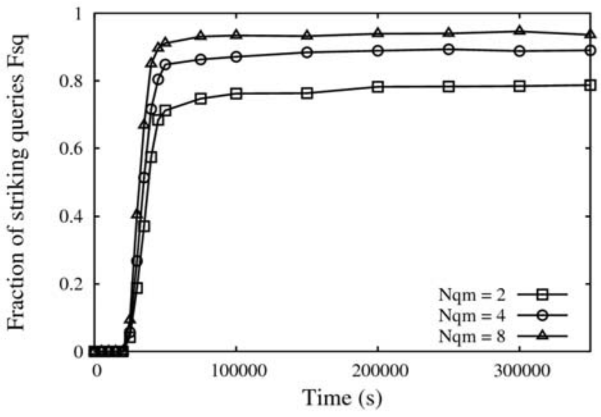

Fig. 11. Use of ARMAP representative peers: fraction of discovery requests that are successfully driven to a representative peer, for different values of Nqm and TTL set to 7.

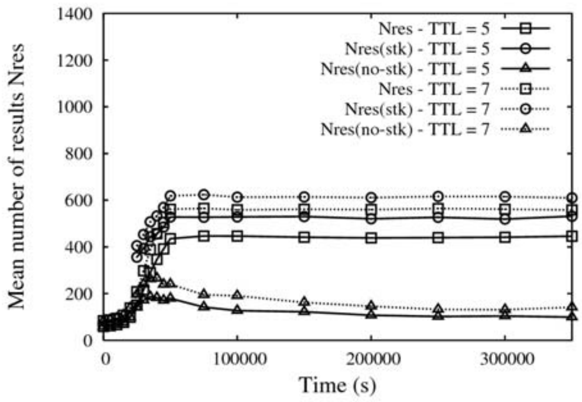

Fig. 12. Mean number of results. Performance values, for different values of TTL and Nqm set to 4, are reported for all the queries, and separately for striking and non striking queries.

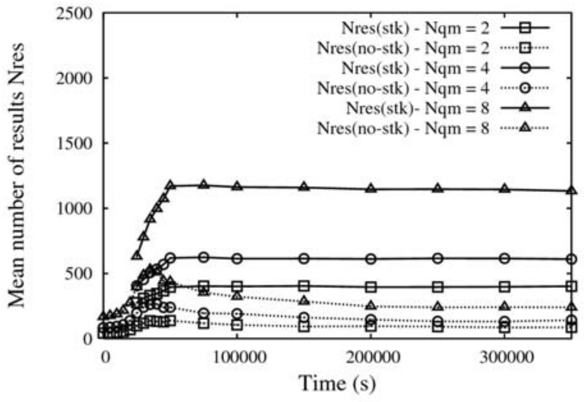

Fig. 13. Mean number of results. Performance values, for different values of Nqm and TTL set to 7, are reported separately for striking and non striking queries.

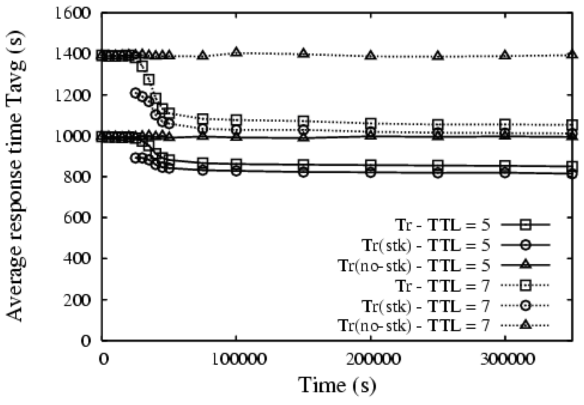

Fig. 14. Average response time. Performance values, for different values of TTL and Nqm set to 4, are reported for all the queries and separately for striking and non-striking queries.

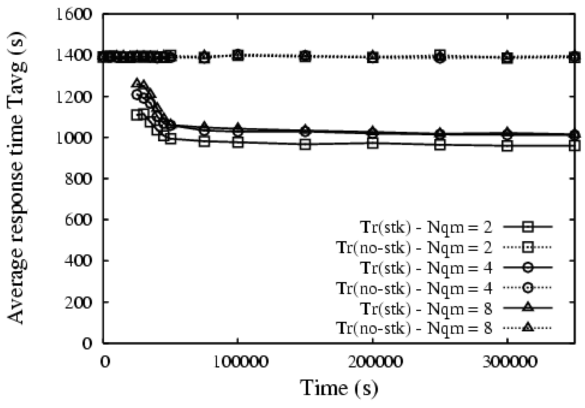

Fig. 15. Average response time. Performance values, calculated for different values of Nqm and TTL set to 7, are reported separately for striking and non striking queries.

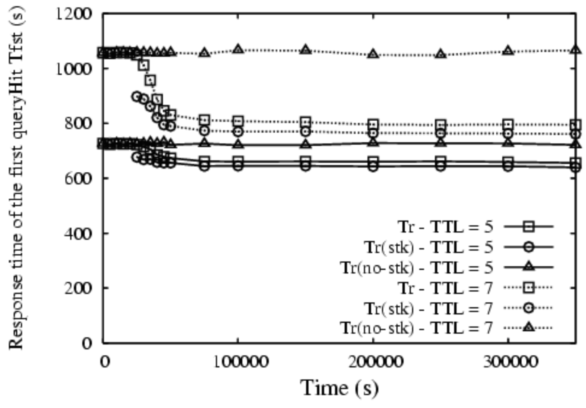

Fig. 16. Response time of the first queryHit. Performance values, for different values of TTL and Nqm set to 4, are reported for all the queries, and for striking and non striking queries.

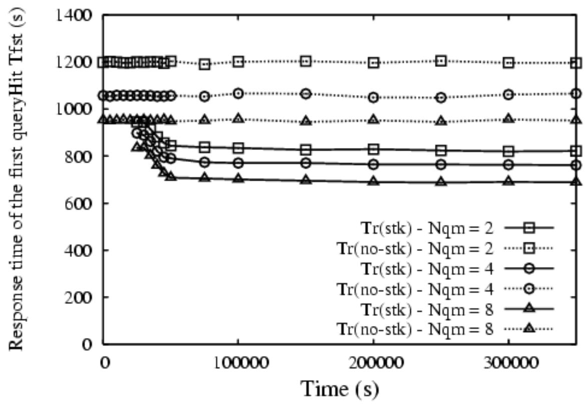

Fig. 17. Response time of the first queryHit. Performance values, for different values of Nqm and TTL set to 7, are reported separately for striking and non striking queries.

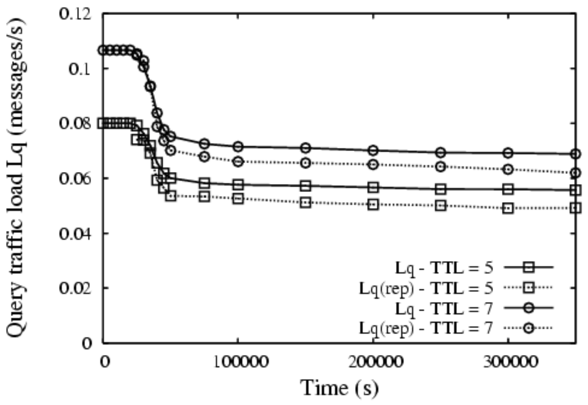

Fig. 18. Query traffic load experienced by generic peers and representative peers. Performance values are calculated for different values of TTL and Nqm set to 4.

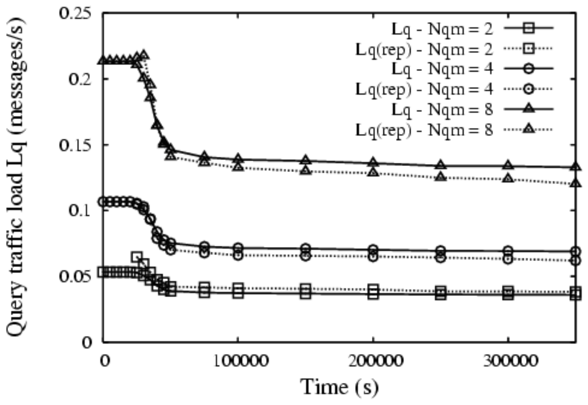

Fig. 19. Query traffic load experienced by generic peers and representative peers. Performance values are calculated for different values of Nqm and TTL set to 7.

4 Related Work

In very large Grids, hosts can provide a large set of distributed and heterogeneous resources. In this kind of environment, the information service is a pillar component because it is essential to assure an efficient management of resources that allows users to discover and use the resources that they need for their applications. Current Grid information services offer centralized or hierarchical information services, but this kind of approach is going to be replaced by a decentralized one, supported by P2P interconnection among Grid hosts.

In the last years, a number of P2P techniques and protocols have been proposed to deploy Grid information services: for example, super-peer networks [16, 25] achieve a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and fault-tolerant features offered by distributed search. P2P search methods can be categorized as *structured* or *unstructured*. The structured approach assumes that hosts and resources are made available on the network by means of a global overlay planning. In Grids, users do not usually search for specific resources (e.g. MP3 or MPEG files), but for software or hardware resources that match an extensible set of features. Accordingly, while structured protocols, based on highly structured overlays and Distributed Hash Tables (e.g. Chord [18]), are usually very efficient in file sharing P2P networks, unstructured or hybrid protocols seem to be preferable in largely heterogeneous Grids. Unstructured search methods can be further classified into *blind* and *informed* ones [22]. If nodes have no information on where the resources are actually located, a search request must be performed through a random exploration of the network, therefore a *blind* search mechanism is adopted, such as the *flooding* or the *random walk* technique [14]. If a centralized or distributed information service maintains information about resource location, it is possible to drive query messages with an *informed* mechanism, such as the *routing indices* mechanism [6] or the *adaptive probabilistic search* [21]. As discussed in Section 2.3, the ARDIP semi-informed discovery protocol presented in this paper aims to combine the flexible and scalable features of a blind approach with the efficiency and rapidity of an informed approach.

The ARMAP protocol, which is the base of the ant-based approach presented in this paper, exploits the features of Multi-Agent Systems (MAS), and in particular of ant-based algorithms. A MAS can be defined as a loosely coupled network of problem solvers (agents) that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver [20]. Research in MASs is concerned with the study, behavior, and construction of a collection of autonomous agents that interact with each other and the environment. Ant-based algorithms are a class of agent systems which aim to solve very complex problems by imitating the behavior of some species of ants [3].

The Anthill system [2] is a framework that supports the design, implementation and evaluation of P2P applications based on multi-agent and evolutionary programming. In Anthill, societies of adaptive agents travel through the network, interacting with nodes and cooperating with other agents in order to solve complex problems. Recently, an approach based on ant behavior and genetic algorithms to search resources on a P2P network has been introduced [7]. A sub-optimal route of query messages is learnt by using positive and negative pheromone feedbacks and a genetic method that combines and improves the routes discovered by different ants. Whereas in [7] the approach is tailored to improve search routes with a given distribution of resources in the network, the ARMAP algorithm logically reorganizes and replicates resources in order to decrease the intrinsic complexity of discovery operations. Instead of directly using ant-based algorithms to search resources, the ARMAP protocol exploits an ant-based replication and mapping algorithm to replicate and reorganize resources according to their categorization.

The problem of driving the behavior of agents, so making them able to take actions autonomously, without having an overall view of the system, is discussed in [23]. There, a decentralized scheme, inspired by insect pheromone, is used to limit the activity of a single agent when it is no more concurring to accomplish the system goal. Information dissemination is a fundamental and frequently occurring problem in large, dynamic, distributed systems, since it consents to lower query response times and increase system reliability. In [5], the authors examine a number of techniques that can improve the effectiveness of blind search by proactively replicating data. In particular, two natural but very different replication strategies are described: *uniform* and *proportional*. The uniform strategy, replicating everything equally, appears naive, whereas the proportional strategy, where more popular items are more replicated, is designed to perform better but fails to do that. Actually, it is shown that any strategy that lies between the two performs better than the two extreme strategies. In [12] it is proposed to disseminate information selectively to groups of users with common interests, so that data is sent only to where it is wanted. In our paper, instead of classifying users, it is proposed to exploit a given classification of resources: resources, or resource metadata documents, are replicated and disseminated with the purpose of creating low-entropy regions of the network that are specialized in a specific class of resources. The so obtained information system allows for the definition and usage of the ARDIP semi-informed discovery protocol.

ARMAP and ARDIP protocols assume that the classification of resources has already been performed. This assumption is common in similar works: in [6], performance of a discovery technique is evaluated by assuming that resources have been previously classified in 4 disjoint classes. Classification can be done by characterizing the resources with a set of parameters that can have discrete or continuous values. Classes can be determined with the use of Hilbert curves that represent the different parameters on one dimension [1]; alternatively, an n-dimension distance metric can be used to determine the similarity among resources [13].

5 Conclusions and Future Work

This paper introduced an approach based on multi agent systems for building an efficient information system in Grids. A number of self-organizing agents travel the network by exploiting P2P interconnections; agents replicate and gather information related to resources having similar characteristics in restricted regions of the Grid. Such a logical reorganization of resources is exploited by a semi-informed resource discovery protocol, namely the ARDIP protocol, which is tailored to route a query message towards a "representative peer" that collects a large number of resources having the desired characteristics.

Simulation analysis showed that, as the reorganization of resources proceeds, the ARDIP protocol allows users to discover more and more resources in a shorter amount of time, and at the same time to decrease the traffic load experienced by Grid hosts. It must be remarked that performance results are related to a particular choice of parameter values: for example resources are categorized in 5 different classes and most results are computed for a Grid of 2,500 hosts. However, performance evaluation w.r.t several parameters, e.g. Grid size, number of agents, and TTL, suggests that an imperfect choice of parameter values cannot spoil the reorganization and discovery process, but can only make such process faster or slower, and final achievements (for example the decrease in overall entropy and the increase in the number of results) seem to be preserved in any case. Furthermore the self-organizing and decentralized nature of the involved algorithms, along with the analysis of performance results obtained with variable Grid sizes, suggest that the proposed approach is scalable and can be adopted in a Grid framework regardless of the Grid size.

Current work focuses on the implementation of ARDIP for the discovery of WSRF-compliant Web services. Web services can be categorized according to their syntactic and semantic features (e.g. WSDL specification of input and output parameters, pre and post conditions, ontology concepts) and QoS information (e.g. information about service availability, reliability, execution time, price). By tuning ARMAP pick and drop probability functions, agents can favor the dissemination of metadata documents associated to Web services having high QoS rankings and hinder the dissemination of low-QoS services. The goal is to evaluate how this enhancement can improve the QoS features of Web services discovered by ARDIP queries.

Another enhancement under evaluation is the possibility of dynamically adapting the ARMAP algorithm, depending on users' and system requirements. For example, if a larger number of resource replicas is desired, this goal could be fulfilled by delaying the copy/move modality switch. Since this change cannot be communicated instantaneously to all the agents, a decentralized method for gradually informing peers and agents about the parameter updating must be envisaged and evaluated.

Acknowledgements

This work has been supported by the Italian MIUR FIRB project RBNE01KNFP on High Performance Grid Platforms and Tools and by the LOGICA project funded by Regione Calabria (Programma Operativo Regionale – POR, Misura 3.16.B2)

References

[1] A. Andrzejak and Z. Xu, Scalable, efficient range queries for grid information services, Proceedings of the 2nd IEEE International Conference on Peer-to-Peer Computing, Linkping University, Sweden, 2002, pp. 33.

[2] O. Babaoglu, H. Meling, and A. Montresor, Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems, Proceedings of the 22nd International Conference on Distributed Computing Systems, ICDCS '02, Wien, Austria, 2002, pp. 15-22.

[3] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity (1999).

[4] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz and E. Bonabeau, Self-Organization in Biological Systems, Princeton, NJ, Princeton University Press (2001).
[5] E. Cohen and S. Shenker, Replication strategies in unstructured peer-to-peer networks, Proceedings of the ACM SIGCOMM '02 Conference, 2002.
[6] A. Crespo and H. Garcia-Molina, Routing indices for peer-to-peer systems. Proceedings of the 22nd International Conference on Distributed Computing Systems ICDCS'02, Wien, Austria, 2002, pp. 23-33.
[7] P. Dasgupta, Intelligent Agent Enabled P2P Search Using Ant Algorithms, Proceedings of the 8th International Conference on Artificial Intelligence, Las Vegas, NV, 2004, pp. 751-757.
[8] J. L. Deneubourg, S. Goss, S. Franks, A. Sendova-Franks, C. Detrain and L. Chrétien, The dynamics of collective sorting: robot-like ants and ant-like robots, Proceedings of the first international conference on simulation of adaptive behaviour, From animals to animats, Paris, France, 1991, pp. 356-365.
[9] A. Forestiero, C. Mastroianni, G. Spezzano, A Multi Agent Approach for the Construction of a Peer-to-Peer Information System in Grids, Proceedings of the International Conference on Self-Organization and Adaptation of Multi-agent and Grid Systems SOAS' 2005, Glasgow, UK, 2005, pp. 220-236.
[10] P. Grassé, La Reconstruction du nid et les Coordinations Inter-Individuelles chez Bellicositermes Natalensis et Cubitermes sp. La théorie de la Stigmergie: Essai 'interprétation du Comportement des Termites Constructeurs, Insectes Sociaux 6 (1959), pp. 41-84.
[11] A. Iamnitchi, I. Foster, J. Weglarz, J. Nabrzyski, J. Schopf and M. Stroinski, A Peer-to-Peer Approach to Resource Location in Grid Environments, eds. Grid Resource Management, Kluwer Publishing (2003).
[12] A. Iamnitchi and I. Foster, Interest-Aware Information Dissemination in Small-World Communities, Proceedings of the IEEE International Symposium on High Performance Distributed Computing, HPDC 2005, Research Triangle Park, NC, 2005, pp. 167- 175.
[13] A. Z. Kronfol, FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine, PhD thesis at Princeton Univerity, 2002.
[14] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, Search and replication in unstructured peer-to-peer networks, ACM, Sigmetrics (2002), pp. 258 - 259.

[15] C. Mastroianni, D. Talia and P.Trunfio, Metadata for Managing Grid Resources in Data Mining Applications, Journal of Grid Computing, Kluwer Academic Publishers, Vol. 2, No. 1 (2004), pp. 85-102.

[16] C. Mastroianni, D. Talia and O. Verta, A Super-Peer Model for Resource Discovery Services in Large-Scale Grids, Future Generation Computer Systems, Elsevier Science, Vol. 21, No. 8 (2005), pp. 1235-1248.

[17] K. Petersen, M. Spreitzer, D. Terry, M. Theimer and A. Demers, Flexible Update Propagation for Wakly Consistent Replication, Proceedings of the 16th ACM Symposium on Operating System Principles, 1997, pp. 288-301.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, Proceedings of the ACM SIGCOMM Conference, San Diego, CA, USA, 2001.

[19] The SWARM environment, SWARM Development Group of Santa Fe University, New Mexico, http://www.swarm.org.

[20] K. Sycara, Multiagent systems, Artificial Inteligence Magazine, Vol. 19, No. 2 (1998), pp. 79–92.

[21] D. Tsoumakos and N. Roussopoulos, Adaptive probabilistic search for peer-to-peer networks, Proceedings of the 3rd International Conference on Peer-to-Peer Computing P2P'03, Linkoping, Sweden, 2003, pp. 102-110.

[22] D. Tsoumakos and N. Roussopoulos, A Comparison of Peer-to-Peer Search Methods, Proceedings of the 6th International Workshop on the Web and Databases WebDB, San Diego, CA, 2003, pp.61-66.

[23] H. Van Dyke Parunak, S. A. Brueckner, R. Matthews and J. Sauter, Pheromone Learning for Self-Organizing Agents, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 35, No. 3 (2005), pp. 316- 326.

[24] The Web Services Resource Framework, http://www.globus.org/wsrf/.

[25] B. Yang and H. Garcia-Molina, Designing a Super-Peer Network, Proceedings of the 19th International Conference on Data Engineering, IEEE Computer Society Press, Los Alamitos, CA, USA, 2003, pp. 49- 60.

[26] B. Yang and H. Garcia-Molina, Efficient search in peer-to-peer networks, Proceedings of the 22nd International Conference on Distributed Computing Systems ICDCS, Wien, Austria, 2002.

Author Bios

Agostino Forestiero received his Laurea degree in Computer Engineering from the University of Calabria, Cosenza, Italy, in 2002 and his PhD in Computer Engineering from the University of Calabria in 2006. Since 2002 he collaborates with the Institute of High Performance Computing and Networks of the Italian National Research Council (ICAR-CNR) in Cosenza. His research interests include Grid Computing, Peer-to-Peer Networks and Swarm Intelligence.

Carlo Mastroianni is a researcher at the Institute of High Performance Computing and Networks of the Italian National Research Council (ICAR-CNR) in Cosenza, Italy, since 2002. He received his PhD in Computer Engineering from the University of Calabria in 1999. His research interests focus on distributed systems and networks, in particular on Grid Computing, Peer-to-Peer Networks, Content Distribution Networks, Multi Agent Systems. He published more than 50 scientific papers on international journals and conferences. He currently lectures Computer Networks at the University of Calabria. He is serving as a program committee member for many international journals and conferences.

Giandomenico Spezzano is a Research Director at the Institute of High Performance Computing and Networking (ICAR) of the Italian National Research Council (CNR), where he manages the Group of Intelligent Grid and Peer-to-Peer Systems. He is also a Contract Professor at the University of Calabria, Italy, since 1994. Previously, he worked at CRAI (Consortium for Research and Applications of Information Technology), Italy, where he has led various research projects in the distributed and parallel computing area. He received the Laurea degree in Industrial Technologies Engineering from the University of Calabria, Italy, in 1980. He has published two books and about 130 papers in international journals and conference proceedings such as IEEE TEC, IEEE CSE, FGCS, PDCP, Parallel Computing, Concurrency:Practice and Experience. His current research interests cover models and tools for massively parallel architectures, grid computing, peer-to-peer computing, parallel and distributed data mining, parallel genetic programming, cellular automata and swarm intelligence. He is serving as a program committee member for many international conferences and he is member of ACM and IEEE Computer Society.