A Multi Agent Approach for the Construction of a Peer-to-Peer Information System in Grids

Agostino FORESTIERO, Carlo MASTROIANNI and Giandomenico SPEZZANO
ICAR-CNR 87036 Rende (CS), Italy
{forestiero,mastroianni,spezzano}@icar.cnr.it

Abstract. A Grid information system should rely upon two basic features: the replication and dissemination of information about Grid services and resources, and an intelligent distribution of such information among Grid hosts. This paper examines an approach based on ant-based systems to replicate and map Grid services information on Grid hosts according to a given semantic classification of such services. Information is disseminated by agents (ants), which traverse the Grid by exploiting the P2P interconnections among Grid hosts. An entropy index is used to evaluate the performance of the proposed Ant-based Replication and MApping protocol (ARMAP), and control the dissemination of resource information. This approach enables the use of a semi-informed search algorithm which can drive query messages towards a cluster of peers having information about resources belonging to the requested class. A simulation analysis has been performed to evaluate the performance of the ARMAP protocol.

Keywords. Grid, P2P, information system, multi agent system, ant system, spatial entropy, resource mapping.

Introduction

The information system of a Grid framework 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 Grid systems, e.g. in the Web Services Resource Framework (WSRF) [20]), the information system is handled through a centralized or hierarchical approach. Nowadays the Grid community agrees that in large and highly heterogeneous Grids it is more efficient to devise scalable Grid information services based on a peer-to-peer (P2P) approach [9, 17].

In this paper, it is assumed that the resources offered by a service-oriented Grid are Grid services which are deployed and published by Grid Virtual Organizations; for example, within the WSRF framework, published Grid services are Web services having enriched functionalities such as state management. Furthermore, it is assumed that such Grid services can be semantically classified according to their features: a *class of resources* is defined as a set of Grid services matching specified properties. Generally, a query is not generally issued to search a single resource, but to collect information about resources belonging to a given class [4, 12]. After receiving a number of responses, a user or client can choose the resource which is more appropriate for their purposes.

This paper proposes a novel approach for the construction and management of a Grid information system which allows for an efficient search of resources. It is assumed that an underlying P2P infrastructure interconnects Grid nodes and can be used to explore the Grid. The proposed approach exploits the features of (i) epidemic mechanisms tailored to the dissemination of information in distributed systems [13] and (ii) self adaptive systems in which some sort of "swarm intelligence" emerges from the behaviour of a high number of agents which interact with the environment [3].

The rationale of using a replication approach is the following: even if a Grid service is provided by a particular Grid host, a number of information documents describing this service should be distributed on the Grid in order to facilitate discovery operations. An information document can be composed of a description of the service (i.e. a set of parameter/value pairs which specify the main characteristics of the service) and an URL reference to the WSDL interface of the service. For the sake of simplicity, in the following an information document describing a Grid service or resource will be referred to as a *Grid resource* or simply as a *resource*.

This paper proposes to disseminate information in a controlled way, with the purpose of maximizing the benefit of the replication mechanism and facilitate the discovery operations. Replicas are spatially sorted (or "mapped") on the Grid so that resources belonging to the same class are placed in nearby Grid hosts. The mapping of resources is managed through a multi agent approach, based on the model that was introduced in [6] to emulate the behavior of ants which cluster and map items within their environment. This paper proposes a variant of that model, in which items (in our case the Grid resources) are both *replicated* and *mapped*. A number of agents traverse the Grid via the underlying P2P interconnections and copy or move resources from one host to another, by means of *pick* and *drop* random functions. In particular, each agent is tailored to *pick* resources of a given class from a region in which such resources are scarcely present, and *drop* them in a region where they are already being accumulated.

An entropy function is defined for two main purposes: (i) to evaluate the effectiveness of the replication and mapping protocol for different values of network and protocol parameters; (ii) to choose the modality of mapping, between the *copy* modality and the *move* modality. In a first phase, the *copy* modality is used to generate an adequate number of resource replicas on the network. With this modality, when executing a pick operation, an agent does not remove resources from the current host: it generates a copy of the resources belonging to a given class, and takes such resources until it will leave them in another host. However, the copy modality cannot be maintained for a long time, since eventually every host would have a huge number of resources, thus weakening the efficacy of resource mapping. Accordingly, after a proper interval of time, the protocol should switch to the *move* modality: when an agent picks some resources, they are actually *removed* from the current host, thus preventing an excessive proliferation of replicas.

A *semi-informed* discovery protocol can efficiently exploit this form of resource mapping: if a number of resources of the same class are accumulated in a restricted region of Grid hosts, queries for such resources can be driven towards that region, in order to maximize the number of useful responses. A discovery operation can be performed in two phases. In the first phase, a query is forwarded through a *blind* mechanism. In the second phase, whenever a query gets close enough to a Grid region specialized in the needed class of resources, the search becomes *informed*: the query is driven towards the specialized Grid region and will easily discover a high number of useful resources.

This paper shows that the proposed protocol, namely the ARMAP protocol (Ant-based Replication and MApping Protocol), can be effectively used to build a Grid information system in which resources are properly replicated while keeping the overall entropy and the network load as low as possible.

The remainder of the paper is organized as follows. Section 1 introduces the ARMAP protocol, and discusses the random functions that drive the behaviour of mobile agents. Section 2 analyzes the performance of the proposed protocol by means of an event-driven simulation framework built upon the Swarm simulation environment [15]. Section 3 discusses related work and compares our approach to other ones proposed in the last years. Section 4 concludes the paper.

1 A Multi-Agent Protocol for Mapping Resources on the Grid

In this section the ARMAP protocol is defined and discussed. The aim of this protocol is to disseminate Grid resources and spatially map them on the Grid according to their semantic classification, in order to gather a consistent number of resources of the same class in a restricted region of the Grid. It is assumed that the resources have been previously classified into a number of classes Nc, according to their semantics and functionalities (see [4] and [12]).

The ARMAP protocol exploits the random movements and operations of a number of mobile agents that travel the Grid using the P2P interconnections. This approach is inspired by ant-based systems [3, 5, 6], in which swarm intelligence emerges from the collective behaviour of very simple mobile agents (ants), and a complex overall objective is achieved.

In ARMAP, each mobile agent can pick a number of resources on a Grid host, carry such resources while moving form host to host, and deposit them on another Grid host. Initially, it is assumed that each agent is “class-specific”, i.e. it manages the resources of only one class. This assumption will be released later. The basic features of the ARMAP protocol (agent movements and pick and drop operations) are described in Section 1.1. Section 1.2 introduces the spatial entropy function used to evaluate the effectiveness of ARMAP and discusses a decentralized approach, based on ants’ pheromone, that is used by a single agent to evaluate the correct time at which it should switch the protocol modality from *copy* to *move*. Section 1.3 discusses the role of the ARMAP protocol in the design of a Grid information system.

1.1 ARMAP basic operations

Agent Movement

Each agent travels over the Grid through the P2P interconnections among Grid hosts. For the sake of simplicity, the ARMAP protocol has been analyzed in a P2P network in which peers are arranged in a grid-like topology, as in the Swarm simulator [15]: each peer is connected to 8 neighbour peers, including horizontal, vertical and diagonal neighbours. At random times, each agent makes a random number of hops along the P2P network (the maximum number of hops Hmax is a protocol parameter), executes the agent's algorithm specified by the ARMAP protocol, and possibly performs a *pick* or *drop* operation.

Pick operation

Once an agent specialized in a class Ci gets to a Grid host, if it is currently unloaded (i.e. it is not taking resources of class Ci), it must decide whether or not to pick the resources of class Ci that are managed by the current host. The probability of picking the resources of class Ci is defined through by a *pick* random function; to favor the spatial mapping of resources, such probability must be inversely proportional to the number of resources of class Ci that are currently located in the local region of the Grid.

More precisely, the Ppick random function, defined in formula (1), is the product of two factors, which take into account, respectively, the relative accumulation of resources of a given class (with respect to the other classes), and their absolute accumulation (with respect to the initial number of resources of that class).

$$(1)\quad P_{pick} = \left(\frac{k1}{k1 + fr}\right)^2 \cdot \left(\frac{(fa)^2}{k2 + (fa)^2}\right)^2$$

The fr fraction can assume values comprised between 0 and 1 and is computed as the number of resources of class Ci accumulated in the peers located in the *visibility region* divided by the overall number of resources (of all classes) 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. the peers located within the *visibility radius*). Here it is assumed that the visibility radius is 1, so that the *visibility region* is composed of 9 hosts, the current one included.

The fa fraction, which also can assume values comprised between 0 and 1, is computed as the number of resources of class Ci that are *owned* by the hosts located in the visibility region (i.e. resources that are directly published by such hosts) divided by the number of resources of the same class that are currently *maintained* by such hosts (i.e. including resources, published by other hosts, that have been previously deposited within the visibility region by mobile agents). Note that Ppick is directly proportional to the fraction fa, which in turns is inversely proportional to the extent to which the hosts within the visibility region have accumulated resources of class Ci so far. k1 and k2 are threshold constants which are both set to 0.1.

If the ARMAP protocol works in the *copy* modality, when an agent picks some resources of class Ci, it leaves a copy of them in the current host; otherwise, if the *move* modality is assumed, such resources are removed from the current host. In the latter case, the current host will only maintain the resources of class Ci that it directly owns, but it loses all the information about the resources of that class that are owned by other hosts.

Note that the ARMAP protocol assumes that each host is informed about the resources that are maintained by the hosts located within the visibility region. This assumption is not restrictive, since it is only required that a host periodically sends to the adjacent hosts a message containing information about the resources that it is currently maintaining. A soft state mechanism can be used to manage the possible disconnection of neighbors: if information about the resources maintained by a neighbor host is not refreshed after a proper amount of time, it must be deleted.

Drop operation

Whenever an agent specialized in a class `Ci` gets to a new Grid host, it must decide whether or not to drop the resources of class `Ci`, in the case that it is carrying any of them. As opposed to the pick operation, the dropping probability should be directly proportional to the relative and absolute accumulation of resources of class `Ci` in the visibility region. The `Pdrop` function is shown below.

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

In (2), the threshold constants `k3` and `k4` are set to 0.3 and 0.1, respectively.

A first set of experiments (not reported here) was performed to evaluate the impact of the threshold constants on ARMAP performance and to choose a proper set of values for such parameters. However, this setting is not very critical, since it was noted that a different set of values can only affect the rapidity of the mapping procedure, but has no remarkable effect on the qualitative behavior of ARMAP.

A high-level description of the ARMAP algorithm executed by each agent is given in Fig. 1: the different behaviour of an agent with the copy and move modalities can be noted.

```
// Na = number of agents: each one is specialized in a class of
        resources
// Hmax = max number of P2P hops that an agent can perform between two
//       successive operations
// mod = ARMAP modality (copy or move)
For each agent a (specialized in class Ci) do forever {
  Compute integer number h between 1 and Hmax;
  a makes h P2P hops;
  if (a is unloaded) {
    compute Ppick;
    draw random real number r between 0 and 1;
    if (r<=Ppick) then {
      pick resources of class Ci from current host;
      if (mod == move)
        remove resources of class Ci from current host;
    }
  }
  else {
    compute Pdrop;
    draw random real number r between 0 and 1;
    if (r<=Pdrop) then
      drop resources of class Ci into current host;
  }
}
```

Fig. 1. High-level description of the ARMAP algorithm

To tune the number of agents Na, it is supposed that each peer, at the time it joins the Grid, generates an agent with a given probability Pgen. For example, Pgen is set to 0.5 if it is wanted Na to be approximately equal to half the number of peers (Np/2).

So far, only class-specific agents were considered: with this assumption, each peer casually selects the class of resources in which the generated agent will be specialized. However, to improve performance, it is also possible to generate *generic* agents, which are able to pick and drop resources belonging to all the resource classes or a subset of them. In such a case, the algorithm shown in Fig.1 is slightly modified: the agent computes the pick and drop random functions separately for each class it can manage. This way an agent may pick the resources of class Ci from a Grid host, and drop the resources of another class Cj into the same host. The performance increase obtained by using generic agents will be shown in Section 2.2.

1.2 Entropy function and pheromone mechanism

An entropy function is defined to evaluate the effectiveness of the ARMAP protocol. For each peer p, the entropy Ep gives an estimation of the extent to which the visibility region centred in p has accumulated resources belonging to one class, thus giving a contribute in the sorting process. As shown in formula (3), the overall spatial entropy E is defined as the average of the entropy values Ep computed at all Grid hosts. In (3), fr(i) is the fraction fr, as defined in Section 1.1, evaluated for the class of resources Ci. Note that Ep has been normalized, so that its value is comprised between 0 and 1.

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

Simulation runs were executed to evaluate the correct time at which the modality switch (from *copy* to *move*) should be performed in order to minimize the entropy function. The assumption here is that each agent knows the value of the entropy function at every instant of time. Results are given in Section 2.

However, in the real world an agent has only a local view of the environment and cannot determine its behaviour on the basis of global system parameters such as the overall system entropy. Therefore, a method is introduced with the purpose of enabling a single agent to perform the modality switch only on the basis of local information. Such a method is based on the observation that an increase of the overall entropy value corresponds to a significant decrease of the mean level of activity of agents, i.e. of the frequency of pick and drop operations that are performed by agents. Indeed a low agent activity is a clue that a high degree of spatial sorting among resources has already been achieved.

Accordingly, a single agent can evaluate its own level of activity by using a pheromone mechanism, as suggested in [7]. In particular, at given time intervals, each agent counts up the number of successful and unsuccessful pick and drop operations (a pick or drop operation attempt is considered *successful* when the operation actually takes place – i.e. when the random number extracted is lower than the operation probability function, see Fig.1). At the end of each time interval, the agent makes a deposit into its pheromone base, by adding a pheromone amount equal to the fraction of unsuccessful operations with respect to the total number of operation attempts. An evaporation mechanism is also used to give a higher weigh to the recent behaviour of the agent.

The pheromone level at the end of the i-th time interval, Φi, is computed with formula (4).

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

The evaporation rate `E` is set to 0.9, and φi is the fraction of unsuccessful operations performed in the last time interval. As soon as the pheromone level exceeds a given threshold `Tf`, the agent can conclude that the frequency of pick and drop operations has remarkably reduced, and switches its protocol modality from *copy* to *move*. The value of `Tf` is set by observing the global system behaviour, as explained in Section 2.2. Note that the choice of updating the pheromone level every time interval, instead of every single operation, has been made to fuse multiple observations into a single variable, so giving a higher strength to agents' decisions. The length of the time interval must be long enough to collect a significant number of observations and short enough to give the right weight to pheromone evaporation; a value of 2000 sec was found to be a good compromise.

1.3 Role of the ARMAP protocol for the design of P2P information systems in Grids

The ARMAP protocol can be a significant step towards the design and construction of a P2P-based information system in a Grid environment. However, to better understand its role, it is necessary to discuss how ARMAP can be related to the overall information system design process, which could be composed of the following three components:

1. classification or clustering of Grid resources;
2. replication and mapping of resources with the ARMAP protocol;
3. discovery service.

The first component allows users to identify the features and functionalities of the resources they need (i.e. a particular resource class). Classification of resources can be performed with different techniques, as discussed in the Section 3.

The ARMAP protocol assumes that resources have already been classified.

The third component, the discovery service, assumes that the resources have been re-organized through the ARMAP protocol, or at least that ARMAP is working while discovery requests are being forwarded. The use of ARMAP permits to handle resource discovery by combining the flexible and scalable features of a *blind* approach with the efficiency and fastness of an *informed* approach. A possible discovery protocol that takes full advantage of the ARMAP work is briefly described in the following. A query message first travels the Grid network with a blind/random mechanism; however, the search procedure is turned into an informed one as soon as the query approaches a low-entropy region, i.e. a region which has gathered resources belonging to one particular class. In each low-entropy region a peer - for example the peer that collects the maximum number of resources belonging to the class in which such a region is specialized - is elected as a *representative peer*. It is possible to devise a procedure that allows representative peers to exchange information with each other and allows for the construction of an overlay network connecting representative peers. This way, a query can be routed towards the nearest representative peer, regardless of the class of resources in which it is specialized. Then, this representative peer could re-route the query to another representative peer which is specialized in the class of resources under interest, or simply send an informative message to the requesting peer. When a query finally gets to the proper representative peer, it will easily find a high number of useful results.

This paper is focused on the performance of the ARMAP protocol; the resource discovery protocol described above will be evaluated in future work.

2 Simulation Analysis

In this section we introduce and discuss the parameters and performance indices used to evaluate the ARMAP protocol, than we report and discuss some relevant simulation results which demonstrate the protocol effectiveness in a Grid environment.

2.1 Simulation Parameters and Performance Indices

Simulation runs were performed on a square toroidal grid of peers, by exploiting the software architecture and the visual facilities offered by the Swarm environment [15]. When an agent moves to a destination peer, it performs the algorithm shown in Fig. 1, possibly picks and/or drops a number of resources, and finally moves to another peer. Table 1 and Table 2 report, respectively, the simulation parameters and the performance indices used in our analysis. The number of peers Np (or Grid size) was varied from 225 (a 15x15 grid) to 10000 (a 100x100 grid). Each peer generates one or more agents with a given probability: by modulating this probability, the overall number of agents Na was varied from $Np/4$ to $2Np$. Both class-specific and generic agents were considered in the simulations.

The number of resources (Grid services) owned and provided by a single peer is determined through a gamma stochastic function having an average value equal to 15 (see [9]). Grid resources are assumed to be classified in a number of classes varying from 3 to 7; the class to which each resource belongs is selected by the simulator with a uniform random function. The average time between two successive agent movements is set to 60 sec. To move towards a remote host, an agent exploits the P2P interconnections between Grid hosts. The number of P2P hops that are performed within a single agent movement is also a random function. The maximum number of hops, referred to as $Hmax$, is varied from 1 to $D/2$, where D is equal to the square root of Np. The visibility radius, defined in Section 1.1, is set to 1.

The overall entropy E, defined in Section 1.2, is used to estimate the effectiveness of the ARMAP protocol in the reorganization of the resources. The $Nrep$ index is defined as the mean number of replicas that are generated for each resource: note that new replicas are only generated when the ARMAP protocol works with the *copy* modality. Fop is the frequency of operations (pick and drop) that are performed by each agent; this index gives estimation of agents' activeness and system stability, since such operations are less frequent when a low level of entropy has already been achieved. Finally, the traffic load L is defined as the number of hops per second that are performed by all the active agents. This index is used to evaluate the communication costs produced by ARMAP.

Table 1. Simulation parameters

Parameter	Value
Grid size (number of peer), **Np**	225 to 10000
Number of agents (class-specific or generic), **Na**	Np/4 to 2Np
Mean number of resources provided by a peer	15
Number of classes of resources, **Nc**	3 to 7
Mean amount of time between two successive movements of an agent	60 s
Maximum number of hops, **Hmax**	1 to D/2
Visibility radius, **Rv**	1

Table 2. Performance indices

Performance Index	Definition
Mean Spatial Entropy, **E**	Defined in Section 1.2
Mean number of replicas, **Nrep**	Mean number of replicas of a generic resource (included the original copy) which have been disseminated in the Grid
Mean frequency of operations, **Fop**	Mean number of successful operations - pick or drop - that are performed by a single agent per unit time (operations/s)
Traffic load, **L**	Mean number of hops that are performed by all the agents of the Grid per unit time (hops/s)

2.2 Simulation Results

Simulation results shown in this section are relative to simulations performed for a network with 2500 hosts that provide resources belonging to 5 different classes. In particular, results shown in Figures 2-6 are obtained by setting the number of agents Na to Np/2 (i.e. each peer generates an agent with probability 0.5), and the maximum number of hops Hmax to 3. The impact of these two parameters is discussed later. All agents are supposed to be generic, i.e. they can pick and drop resources belonging to every class.

Performance measures are reported versus time to illustrate the effect of the ARMAP protocol in the reorganization and mapping of resources. Figure 2 shows that the exclusive use of the copy modality is not effective: the system entropy decreases very quickly in a first phase, but increases again when the agents create an excessive number of replicas (Figure 3). The reason of this behaviour is the following: if agents continue to create new replicas, eventually all peers will possess all the resources, thus completely undoing the mapping operation. The curve labeled as “copy/move” in Figure 2 is obtained by switching the protocol modality from *copy* to *move* when it is observed that the entropy function increases two times in succession (entropy is calculated every 2000 sec). The effect of the modality switch is that the system entropy not only ceases to increase but decreases to much lower values. This is due to the action of agents that do not generate further replicas, as shown in Figure 3, but continue their work in creating low-entropy regions specialized in particular classes of resources.

Note that in these it is assumed that a *global* view of the system – the value of the overall system entropy - is known by all the agents. In the following this approach will be referred to as the *global* one.

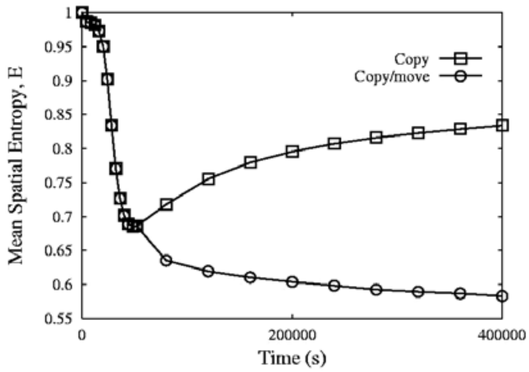

Fig. 2. Mean spatial entropy vs. time; comparison between ARMAP used with the copy modality and ARMAP with the copy/move modality switch.

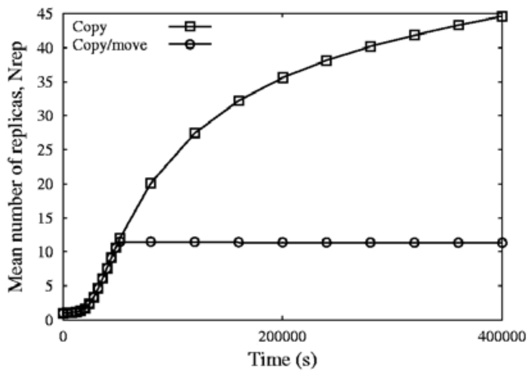

Fig. 3. Mean number of replicas vs. time; comparison between ARMAP used with the copy modality and ARMAP with the copy/move modality switch.

Figure 2 and 4 show that, with the *copy* modality, at the same time as the system entropy begins to increase, the frequency of operations of a single agent begins to decrease. This experimental result can be exploited to define a *local* approach that allows agents to perform the modality switch on their own. To this aim, the pheromone mechanism explained in Section 1.2 is used, but it is necessary to set a proper value of the pheromone threshold `Tf` beyond which an agent must perform the modality switch. This value was set with the following method: by running simulations with the *global* approach, we calculated the mean pheromone value of a generic agent at the time at which the system entropy begins to increase. This value was used as the pheromone threshold of an agent in the *local* approach. Figure 5 compares the curves of system entropy obtained with the *global* and *local* approaches. It is confirmed that the local approach is effective, since it approximates the global one very strictly, by creating a slightly higher number of replicas, as depicted in Figure 6.

Another interesting result of simulations (not shown here) is that the behavior of ARMAP does depend neither on the Grid size, nor on the mean number of resources owned by a generic peer. This assures that each agent can ignore such global parameters and confirms the effectiveness of the local approach based on the pheromone mechanism. The value of the pheromone threshold Tf only depends on the number of agents per peer, i.e. the ratio Na/Np; this parameter can be easily known by the agents since it is equal to the probability that a peer generates and forwards an agent.

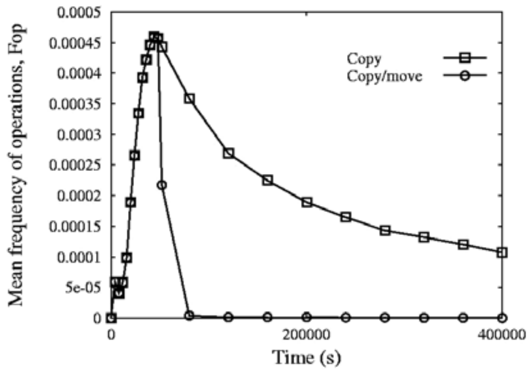

Fig. 4. Mean frequency of operations vs. time; comparison between ARMAP used with the copy modality and ARMAP with the copy/move modality switch.

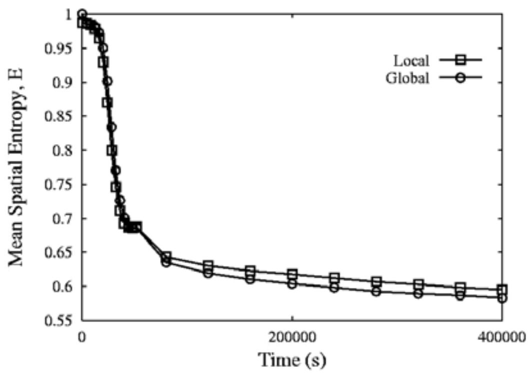

Fig. 5. Mean spatial entropy vs. time; comparison between global and local approaches.

Table 3 shows, for different values of the ratio Na/Np, the values of the time at which the switch modality should be performed (calculated with the *global* approach), and the mean pheromone level of a generic agent at that time. Such a pheromone level, as explained above, is the threshold at which agents must switch to the move modality when the *local* approach is used. It can be observed that, as the number of agents increases, the time at which such agents must stop creating new replicas decreases, since the overall activity of agents is higher. As a consequence, the threshold pheromone level decreases as well: a single agent will reach the pheromone threshold after a shorter interval of time.

Figure 7 reports the trend of the overall system entropy obtained with different numbers of agents: the local approach is used and the pheromone thresholds are set to the corresponding values shown in Table 3. An increase in the number of agents makes the system entropy decrease faster and reach lower values. However, a higher activity of agents also causes an increase in the traffic load (Figure 8). 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.

Figure 9 compares the trend of the overall system entropy obtained with generic and specialized (class-specific) agents: in both cases the number of agents is set to Np/2. The performance increase achieved with the use of generic agents is confirmed, since they permit to obtain much lower values of the system entropy.

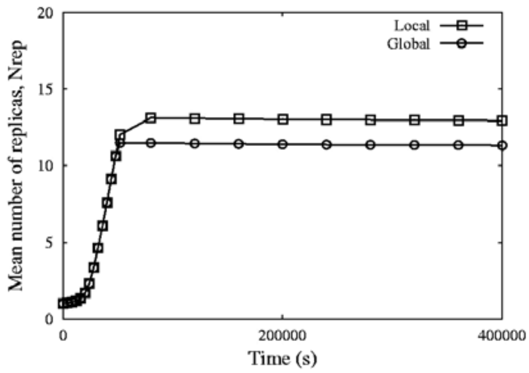

Fig. 6. Mean number of replicas vs. time; comparison between global and local approaches.

Table 3. Modality switch time and pheromone level at switch time with different numbers of agents.

Number of agents Na	Modality switch time (s)	Pheromone level at switch time
Np/4	116,000	9.44
Np/2	52,000	8.94
Np	28,000	7.41
2Np	18,000	5.89

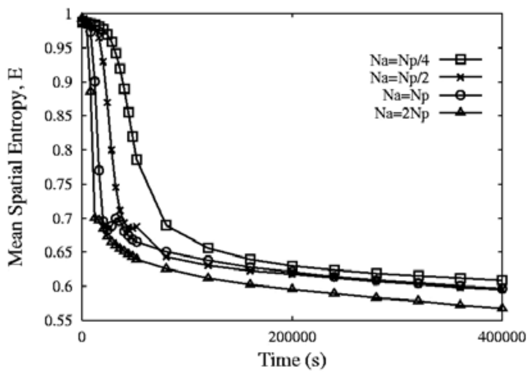

Fig. 7. Mean spatial entropy vs. time, with different numbers of agents.

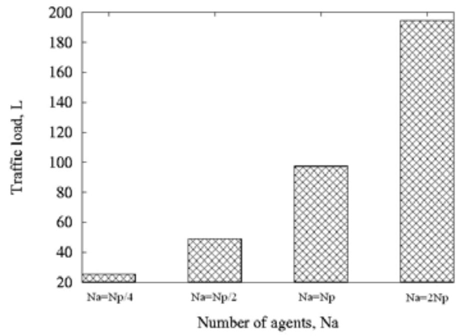

Fig. 8. Mean traffic load (hops/s), with different numbers of agents.

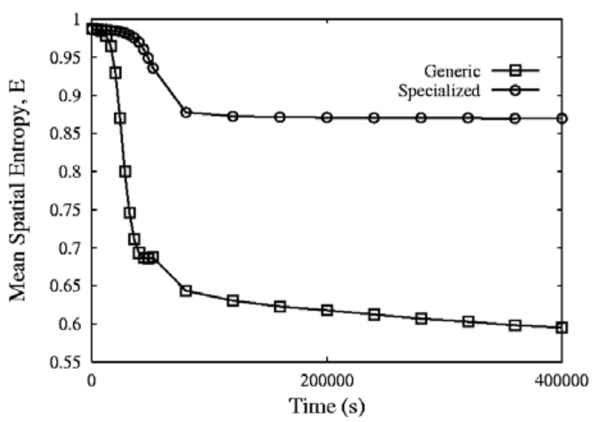

Fig. 9. Mean spatial entropy vs. time. Comparison between generic and specialized agents.

Finally, the effect of the parameter Hmax is analyzed in Figure 10-11. Here, the number of (generic) agents Na is set to Np/2. An increase of Hmax speeds up the sorting of resources (Figure 10), since an agent can move resource between more distant peers, but causes a significant increase in the traffic load (Figure 11). Again, a correct setting of Hmax should take into consideration the network requirements. We chose to set Hmax to 3, because this is the minimum value that permits to move an agent between two visibility regions that are completely disjoint, given that the visibility radius is set to 1.

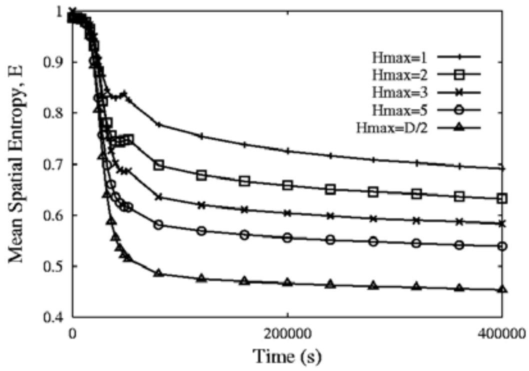

Fig. 10. Mean spatial entropy vs. time, with different values of Hmax.

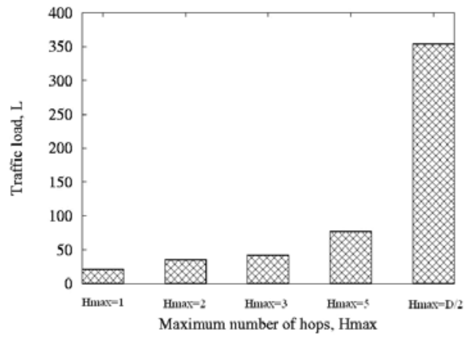

Fig. 11. Mean traffic load (hops/s), with different values of Hmax.

3 Related Work

Since Grid hosts provide a large set of distributed and heterogeneous resources, an efficient Grid information service is a pillar component of a Grid. 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 [12, 21] 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 with a global overlay planning. In Grids, users do not usually search for single resources (e.g. MP3 or MPEG files), but for software or hardware resources that match an extensible set of resource descriptions. Accordingly, while structured protocols, e.g. based on highly structured overlays and Distributed Hash Tables (e.g. Chord [14]), 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 as blind or informed [18]. In a blind search (e.g. using flooding or random walks [11]), nodes hold no information that relates to resource locations, while in informed methods (e.g. routing indices [4] and adaptive probabilistic search [19]), there exists a centralized or distributed information service that drives the search for the requested resources. As discussed in Section 1.3, the approach 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 introduced in this paper is based on 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 [16]. Research in MASs is concerned with the study, behaviour, and construction of a collection of autonomous agents that interact with each other and the environment. Ant-based algorithms are a subclass of agent systems which aim to solve very complex problems by imitating the behaviour 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. Reference [5] introduces an approach based on ant behaviour and genetic algorithms to search resources on a P2P network. 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 [5] the approach is tailored to improve search routes with a given distribution of resources in the network, this paper proposes to reorganize and replicate the 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.

Our protocol is a variant of the basic sorting algorithm proposed in [6]. However, the latter assumes that only one item can be placed in a cell of a toroidal grid, and items can only be moved form one cell to another. Conversely, the ARMAP protocol assumes that a cell (i.e. a Grid host) can store several items (i.e. Grid resources) and an agent can create many replicas of the same item.

The problem of driving the behaviour of a single agent, which should autonomously be able to take actions without having an overall view of the system, has been discussed in [7]. 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. In this paper, a similar approach is used to drive the behaviour of an agent, in particular to evaluate the correct time at which it should switch from the copy to the move modality.

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. Reference [8] proposes to disseminate information selectively to groups of users with common interests, so that data is sent only to where it is wanted. In this paper, instead of classifying users, it is proposed to exploit a given classification of resources: resources are replicated and disseminated with the purpose of creating low-entropy Grid regions that are specialized in specific resource classes. The so obtained information system allows for the definition and usage of a semi-informed search method, as explained in Section 1.3.

The ARMAP protocol assumes that a classification of resources has already been performed. This assumption is common in similar works: in [4], 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 [10].

4 Conclusions

This paper introduces an approach based on multi agent systems to manage resources and construct an efficient information system in Grids. The presented ARMAP protocol accomplishes two main purposes: it replicates resource information on Grid hosts, and gathers information related to similar resources in restricted regions of the Grid. The work is performed by ant-like agents who travel over the network by exploiting P2P connections. A simulation study allowed for a deep analysis of the ARMAP protocol for different values of network and protocol parameters. The controlled dissemination of information assured by ARMAP guarantees a high availability of resources and favours an efficient management of the information system, since different clusters of peers can become specialized in different resource classes. Furthermore, with the use of ARMAP it is possible to devise a semi-informed discovery protocol that maximises the success of a query request by routing it towards a cluster of peers which are specialized in the resource class specified in the query. Further work is currently focused on the performance analysis of such a discovery protocol.

Acknowledgements

This work has been partially supported by the Italian MIUR FIRB Grid.it project RBNE01KNFP on High Performance Grid Platforms and Tools and by the project KMS-Plus funded by the Italian Ministry of Productive Activities.

References

[1] Andrzejak, A., Xu, Z.: Scalable, efficient range queries for grid information services, Proceedings of the Second IEEE International Conference on Peer-to-Peer Computing, Linkping University, Sweden (2002).

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

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

[4] Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: 22 nd International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria (2002), pp.23-33.

[5] Dasgupta, P.: Intelligent Agent Enabled P2P Search Using Ant Algorithms, Proceedings of the 8th International Conference on Artificial Intelligence, Las Vegas, NV, 2004, pp. 751-757.

[6] Deneubourg, J. L., Goss, S., Franks, S., Sendova-Franks, A., Detrain, C., Chrétien, L.: 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, February 1991, Paris, France, pp. 356-365.

[7] Van Dyke Parunak, H., Brueckner, S. A., Matthews, R., Sauter, J., Pheromone Learning for Self-Organizing Agents, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 35, no. 3, May 2005.

[8] Iamnitchi, A., Foster, I., Interest-Aware Information Dissemination in Small-World Communities, IEEE International Symposium on High Performance Distributed Computing (HPDC 2005), Research Triangle Park, NC, July 2005.

[9] Iamnitchi, A., Foster, I., Weglarz, J., Nabrzyski, J., Schopf, J., Stroinski, M.: A Peer-to-Peer Approach to Resource Location in Grid Environments, eds. Grid Resource Management, Kluwer Publishing (2003)

[10] Kronfol, A. Z.: FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine, PhD thesis at Princeton Univerity, May 2002.

[11] Lv, C., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks, ACM, Sigmetrics (2002)

[12] Mastroianni, C., Talia, D., Verta, O., A Super-Peer Model for Resource Discovery Services in Large-Scale Grids, Future Generation Computer Systems, Elsevier Science, Elsevier Science, Vol. 21, No. 8 (October 2005), pp. 1235-1456.

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

[14] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications, Proc. of ACM SIGCOMM, San Diego, CA, USA (2001)

[15] The SwarmWiki environment, Center for the Study of Complex Systems, the University of Michigan, http://www.swarm.org/wiki.

[16] Sycara, K.: Multiagent systems, Artificial Inteligence Magazine, vol. 19, no. 2, pp. 79–92, 1998.

[17] Talia, D., Trunfio, P.: Towards a Synergy between P2P and Grids, IEEE Internet Computing 7(4) (2003) 94-96

[18] Tsoumakos, D., Roussopoulos, N.: A Comparison of Peer-to-Peer Search Methods. Proc. of the Sixth International Workshop on the Web and Databases (WebDB), San Diego, CA, 2003, pp.61-66.

[19] Tsoumakos, D., Roussopoulos, N.: Adaptive probabilistic search for peer-to-peer networks. In: Third International Conference on Peer-to-Peer Computing (P2P'03), Linkoping, Sweden (2003), pp. 102-110

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

[21] Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network, 19th Int'l Conf. on Data Engineering, IEEE Computer Society Press, Los Alamitos, CA, USA (2003)