

An Agent Based Semi-informed Protocol for Resource Discovery 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 resources, and an intelligent logical distribution of such information among Grid hosts. This paper examines an approach based on multi agent systems to build an information systems in which metadata related to Grid resources is disseminated and logically organized according to a semantic classification of resources. Agents 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 agents' work: query messages issued by clients are driven towards "representative peers" which maintain information about a large number of resources having the required characteristics. Simulation analysis proves that the combined use of the resource mapping protocol (ARMAP) and the resource discovery protocol (ARDIP) allows users to find many useful results in a small amount of time.

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 belonging to a given *class*, i.e. matching given criteria and characteristics.

An agent-based protocol (i.e. ARMAP, *Ant-based Replication and Mapping Protocol*) was proposed in [5] to spatially sort (or "map") resources according to their semantic classification [7]. ARMAP exploits the features of (i) epidemic mechanisms tailored to the dissemination of information in distributed systems [8] and (ii) self adaptive systems in which "swarm intelligence" emerges from the behavior of a high number of agents which interact with the environment [1]. By mapping metadata documents on Grid hosts, a *logical reorganization* of resources is achieved.

In this paper, 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 is convenient to drive search requests (issued by hosts 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 a *blind* mechanism, specifically the random walks technique [6], is

adopted: a number of query messages are issued by the requesting host and travel the Grid through the peer-to-peer (P2P) interconnections among Grid hosts. In the second phase, whenever a query gets close enough to a Grid region which is collecting the needed class of resources, the search becomes *informed*: the query is driven towards this Grid region and will easily discover a large number of useful resources. Simulation analysis shows that the ARMAP and the ARDIP protocol, if used together, allow for achieving a very high effectiveness in discovery operations.

The semi-informed ARDIP protocol aims to combine the benefits of both blind and informed resource discovery approaches which are currently used in P2P networks [11]. In fact, a pure blind approach (e.g. using flooding or random walks 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 [2] or adaptive probabilistic search [10]) generally requires a very structured resource organization which is impractical in a large, heterogeneous and dynamic Grid.

The remainder of the paper is organized as follows. Section 2 summarizes the key points of the ARMAP protocol and describes the ARDIP protocol. Section 3 analyzes the performance of the ARDIP protocol. Section 4 concludes the paper.

2 A Multi-agent Protocol for Resource Discovery on Grids

This Section is organized as follows. In Section 2.1, the key points of the ARMAP protocol are briefly summarized (more details can be found in [5]). In Section 2.2, the new ARDIP protocol is introduced.

2.1 ARMAP Basics

The aim of the ARMAP protocol is to achieve a logical organization of Grid resources by spatially sorting them on the Grid according to their semantic classification. It is assumed that the resources have been previously classified into a number of classes \mathcal{N}_c , according to their semantics and functionalities (see [7]). In the following, an information document describing a Grid resource will be referred to as a *logical resource*, or simply *resource*. When distinction is important, the actual resources will be named *physical resources*. 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 [1, 3], in which swarm intelligence emerges from the collective behavior of very simple mobile agents (ants).

Once an agent gets to a Grid host (or *peer*), for each resource class c_i , it must decide whether or not to *pick* the resources of class c_i that are managed by that host, unless the agent already carries some resources of that class. In order to achieve the replication and mapping of resources, a pick random function P_{pick} is defined with the intention that the probability of picking the resources of a given class decreases as the local region of the Grid accumulates such resources. The ARMAP protocol can work in either the *copy* modality or the *move* modality; with the copy modality, an agent that picks some (logical) resources of class c_i will leave a copy of them in the current host; with the move modality, such resources are removed from the current host. A self-organization approach based on ants' pheromone [4] enables each agent

to perform the modality switch (from the copy to the move modality) only on the basis of local information. Analogously, whenever an agent gets to a new Grid host, it must decide whether or not to *drop* the resources of class C_i , if it is carrying any of them. As opposed to the pick probability, the dropping probability is directly proportional to the relative accumulation of resources of class C_i in the local region. A spatial entropy function was defined in [5] to evaluate the effectiveness of the ARMAP protocol in the spatial ordering of resources. Figure 1 gives a graphical description of the mapping process performed by ARMAP. The values of system parameters are set as specified in Section 3, except for the number of resource classes which is set to 3 in order to facilitate the graphical illustration of the process.

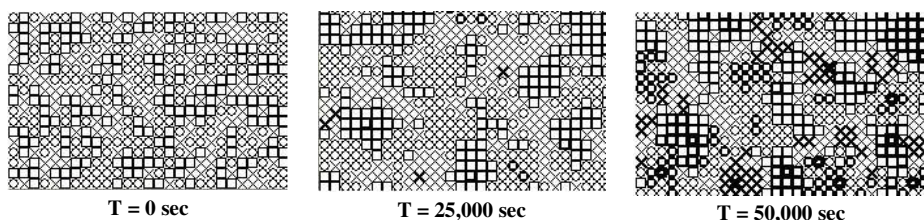


Fig. 1. Gradual mapping of logical resources in a Grid with 3 resource classes. Each peer contains a symbol (circle, square or cross) that corresponds to the most numerous class of resources contained in the peer. The symbol thickness is proportional to the number of resources belonging to the dominant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

2.2 The ARDIP Protocol

The ARDIP (*Ant-based Resource Discovery Protocol*) protocol is used by clients to discover Grid resources 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 accumulating. Because ARDIP fully exploits the replication and spatial sorting 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, it is more and more likely that ARDIP queries can find a remarkable number of useful resources in a small amount of time.

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 allow query messages to take advantage of the positive outcome of previous search requests. These modules are described in the following.

Identification of Representative Peers. As a class of resources C_i is accumulated in a Grid region, the peer that, within this region, collects the maximum number of resources belonging to the class C_i is elected as a *representative peer* for this class. The objective of a search operation is to let a query message get to a representative peer, since such a peer, as well as its neighbors, certainly manages a large number of useful resources. The ARDIP protocol assumes that a peer p is a representative peer of class C_i if at least one of the two following conditions are verified: (a) the peer p maintains

a number of logical resources of class C_i that exceeds f_1 times the mean number of physical resources belonging to class C_i which are offered by a generic peer; (b) the peer p maintains a number of logical resources of class C_i that exceeds f_2 times (with $f_2 < f_1$) the number of logical resources of the same class maintained by its neighbors.

Condition (a) is satisfied by a peer that holds a very high number of logical resources; in general it can be satisfied only when the clustering process of ARMAP is already in an advanced stage. Conversely, condition (b) can be satisfied when clustering is still in progress. 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, within the *comparison radius* R_c : two neighbor representative peers must compare the number of resources they maintain, and the “looser” will be downgraded to a simple peer.

Semi-informed Search. When a user needs to discover resources belonging to a given class C_i , a number of 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 walks paradigm is used: the query messages travel the Grid through the P2P interconnections by following a random path. The network load is limited with the use of a `TTL` parameter, which 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 an informed one as soon as one of the issued query messages approaches a representative peer of class C_i , i.e. when such a message is delivered to a peer which knows the existence of a representative peer 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. Therefore, the semi-informed walk of a query message ends in one of two 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 class C_i , 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 class C_i that are managed by the peers through which it passes.

Stigmergy Mechanism. The stigmergy mechanism is a mechanism, often observed in biological systems, through which elementary entities exploit the environment to communicate with each other. For example, in ant colonies, 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 the food source. The ARDIP protocol exploits a similar mechanism: 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 paper, it is assumed the pheromone is deposited only in the first two peers of the `queryHit` path.

When a query gets to a peer along its blind search, it checks the amount of pheromone which has been deposited there; if the pheromone exceeds a threshold τ_f , it means that a representative peer is close, so the search becomes informed. An evapo-

ration mechanism assures 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. The amount of pheromone Φ_i , computed after the i -th time interval, is given by formula (1).

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

The evaporation rate E is set to 0.9; φ 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 threshold T_E is set to 2.

3 Simulation Analysis

In this section, we discuss some relevant 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, the present section introduces the main system and protocol parameters, while Section 3.1 focuses on the performance of the ARDIP protocol.

A wide set of simulation runs were performed by exploiting the libraries and visual facilities offered by Swarm [9], a software package for multi-agent simulation of complex systems developed at the Santa Fe Institute. Table 1 reports the network, ARMAP and ARDIP parameters used in the simulation analysis.

Table 1. Environment and protocol parameters

Parameter	Value
Grid size (number of peer), N_p	2500
Maximum number of neighbor peers of a Grid peer	8
Mean number of resources published by a peer	15
Number of resource classes, N_c	5
Number of ARMAP agents, N_a	$N_p/2$
Mean amount of time between two successive movements of an ARMAP agent	60 s
Maximum number of hops for each ARMAP agent's movement, H_{max}	3
Number of query messages issued by the requesting peer	4
Time to live, TTL	3-7
Factor $\mathbf{f1}$, for the identification of representative peers	5
Factor $\mathbf{f2}$, for the identification of representative peers	2.5
Comparison radius, R_c	2
Mean generation frequency with which a Grid peer issues query messages	1/300 (1/s)
Mean message elaboration time at a Grid peer	100 ms

3.1 Performance of the ARDIP Protocol

The performance of the ARDIP protocol was analyzed by evaluating the performance indices defined and explained in Table 2. Figure 2(1) depicts the value of the N_{rep} index evaluated at different times while the ARMAP mapping process proceeds. This figure confirms that representative peers are selected almost exclusively with

condition (a) in the first phase, but thereafter the weight of condition (b) becomes predominant, as discussed in Section 2.2.

The index F_{sq} is essential to evaluate how many search requests are actually delivered to a representative peer. Figure 2(2) proves the valuable effect caused by the combined use of ARMAP and ARDIP protocols. In fact, after a very small amount of time, the logical reorganization of resources produces a significant increase in F_{sq} . Moreover, as the TTL value increases, F_{sq} increases as well, since a search request extends the blind search phase and has more chances to get to a representative peer.

Table 2. Performance indices

Performance Index	Definition
Number of representative peers, N_{rep}	Mean number of representative peers of all classes that are selected by ARDIP to attract query messages (see Section 2.2)
Fraction of striking queries, F_{sq}	Fraction of queries that are actually driven towards a representative peer
Mean number of results, N_{res} , $N_{res}(rep)$, $N_{res}(norep)$	Mean number of resources that a node discovers after its query (computed for all the requests, for the requests that are actually delivered to a representative peer, for the requests that are not delivered to a representative peer)
Response times, Tr , $Tr(rep)$, $Tr(norep)$	Mean amount of time (s) that elapses between the generation of a query and the reception of a corresponding queryHit (computed for all the requests, for the requests that are actually delivered to a representative peer, for the requests that are not delivered to a representative peer)

The most important performance measure is N_{res} , the mean number of results that are discovered after a query request. 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 [12]. The trend of N_{res} is depicted in Figure 3(1), which shows that the mean number of results is larger and larger as resources are being organized by ARMAP. Furthermore, even when the probability of reaching a representative peer begins to become stable (Figure 2(2)), the number of results continues to increase, meaning that representative peers are able to collect more and more resources of the class in which they are specialized. It is also noted that the queries which successfully get to a representative peer can discover considerably more results than the rest of the queries, and such a difference increases with the value of the TTL parameter.

The semi-informed discovery protocol not only increases the number of results, but also allows users to discover them in a shorter amount of time. Figure 3(2) shows that the response time decreases as the ARMAP work proceeds, and also that the response time is notably smaller if the query reaches a representative peer; in this case, in fact, the discovery operation is stopped even if the TTL value is still greater than 0, and a queryHit is immediately issued (see Section 2.2). However, this performance improvement is achieved only when the TTL value is sufficiently high.

Finally, performance results not reported in this paper show that the logical reorganization of resources, and the use of the ARDIP protocol, allows for decreasing the traffic load experienced by a single peer. Indeed, when a query messages is driven towards a representative peers, on average it has to make a lower number of hops with respect to a completely blind search.

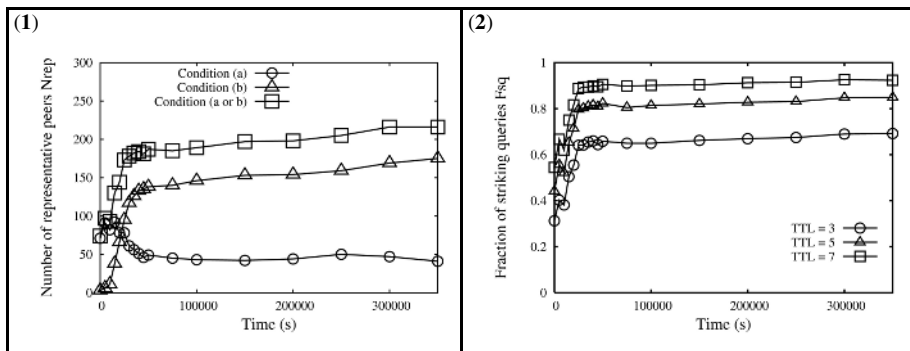


Fig. 2. Selection and actual use of representative peers as the mapping process proceeds. (1): number of representative peers selected with condition *a*, with condition *b*, and overall number of them. (2): fraction of search requests that are successfully driven to a representative peer, for different values of TTL.

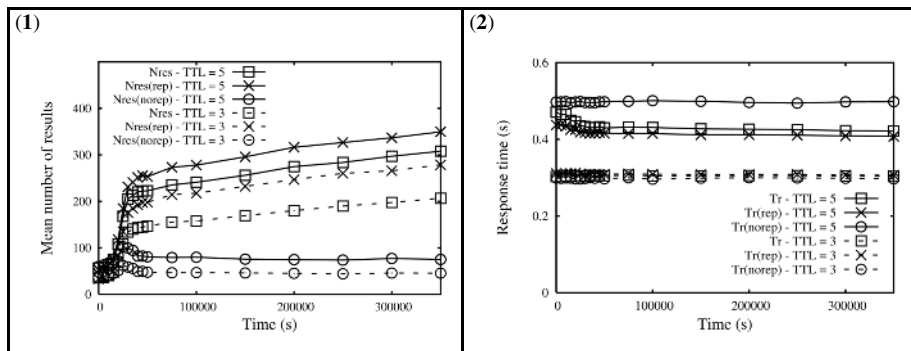


Fig. 3. Performance of search requests as the mapping process proceeds. (1): mean number of results. (2): response time. Values of performance indices, calculated for different values of TTL, are reported for queries that reach a representative peers, for queries that do not reach a representative peers, and for all the queries.

4 Conclusions and Future Work

This paper introduces 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 shows that, as the reorganization of resources proceeds, ARDIP allows users to discover more and more resources in a shorter amount of time, without increasing the traffic load experienced by Grid

hosts. Current work focuses on an enhancement of the ARDIP protocol which fully exploits the features of the *small world* paradigm and on the implementation of ARDIP based on WSRF-compliant Web services.

References

1. 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)
2. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. Proc. of the 2th International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria (2002), pp. 23-33
3. Dasgupta, P.: Intelligent Agent Enabled P2P Search Using Ant Algorithms, Proc. of the 8th International Conference on Artificial Intelligence, Las Vegas, NV (2004), pp. 751-757
4. 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 (2005)
5. Forestiero, A., Mastroianni, C., Spezzano, G.: A Multi Agent Approach for the Construction of a Peer-to-Peer Information System in Grids, Proc. of the 2005 International Conference on Self-Organization and Adaptation of Multi-agent and Grid Systems SOAS 2005, Glasgow, Scotland (2005)
6. Lv, C., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks, *ACM, Sigmetrics* (2002)
7. Mastroianni, C., Talia, D., Verta, O.: A Super-Peer Model for Resource Discovery Services in Large-Scale Grids, *Future Generation Computer Systems*, Elsevier Science, Vol. 21, No. 8 (2005), pp. 1235-1456
8. Petersen, K., Spreitzer, M., Terry, D., Theimer, M., Demers, A.: Flexible Update Propagation for Weakly Consistent Replication, Proc. of the 16th Symposium on Operating System Principles, ACM (1997), pp. 288-301
9. The Swarm environment, Swarm Development Group of Santa Fe University, New Mexico, <http://www.swarm.org>
10. 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
11. 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
12. Yang, B., Garcia-Molina, H.: Efficient search in peer-to-peer networks, Proc. of ICDCS, Wien, Austria (2002)