Percolation-Based Replica Discovery in Peer-to-Peer Grid Infrastructures

Francesco Palmieri

Federico II University of Napoli,
Via Cinthia, 5, Complesso Universitario di Monte S. Angelo, 80126 Napoli, Italy
francesco.palmieri@unina.it

Abstract. Peer-to-peer Grids are collaborative distributed computing/data processing systems, characterized by large scale, heterogeneity, lack of central control, unreliable components and frequent dynamic changes in both topology and configuration. In such systems, it is desirable to maintain and make widely accessible timely and up-to-date information about shared resources available to the active participants. Accordingly we introduce a scalable searching framework for locating and retrieving dataset replica information in random unstructured peer-to-peer Grids built on the Internet, based on a widely known uniform caching and searching algorithm. Such algorithm is based on bond percolation, a mathematical phase transition model well suited for random walk searches in random power law networks, which automatically shields low connectivity nodes from traffic and reduces total traffic to scale sub-linearly with network size. The proposed schema is able to find the requested information reliably end efficiently, even if every node in the network starts with a unique different set of contents as a shared resources.

Keywords: P2P Grids, content search, percolation.

1 Introduction

Grid is an exciting buzzword in the computing world today, mainly in the scientific area. It is usually defined as the exploitation of a varied set of networked computing resources, including large or small computers, storage/file servers and special purpose devices. The emerging Internet based peer-to-peer (P2P) Grid infrastructures, which are based on a "flat" organization allowing seamless discovery, access to, and interactions among resources and services, have complex and highly dynamic computational and interaction behaviors resulting in significant development and management challenges. P2P Grid infrastructures have the potential of a disruptive technology since they can aggregate enormous storage and processing resources while minimizing the overall costs and greatly reducing or avoiding at all the need for centralized servers. A P2P assembly of general purpose nodes connected through the Internet can evolve better from small configurations to larger ones, ensuring almost unlimited scalability features. In such totally distributed architecture, we increasingly face the problem of providing to the running applications fast and reliable access to large data volumes, often stored into datasets geographically distributed across the network. As a direct

T. Doulamis et al.: (Eds.): GridNets 2009, LNICST 25, pp. 45–56, 2010.

consequence, the concept of replication, that is the distribution of multiple copies of a data source of interest across multiple grid nodes, to be processed locally when needed, has been adopted by grid community to increase data availability and maximize job throughput. Replication starts with an initial data discovery operation aiming to detect all the copies of the required dataset already existing on the Grid, given its logical file name. In traditional grids a centralized catalogue is searched in order to find all the locations where the requested dataset is available, obtaining a list of all the already available replicas. A dataset itself can consist of several physical files but the end-user normally only knows the dataset concept. Unfortunately, the peer to peer paradigm by definition, excludes any form of centralized structure, requiring resource management and control to be completely decentralized, hence no traditional catalog service o centralized search facility can be implemented in P2P grids. In such systems, it is however desirable to maintain and make widely accessible timely and up-to-date information about active participants such as services offered and replicated dataset resources available. It is not obvious how to enable powerful discovery query support and collective collaborative functionality that operate on such a distributed and unstructured organization as a whole, rather than on a given part of it. Further, it is not obvious how to allow for the propagation of search results that are fresh, allowing time-sensitive dynamic content. To cope with the above challenges, we introduce a scalable searching framework for locating and retrieving shared replica information in random unstructured peer-to-peer Grids built on transport networks, such as the Internet, characterized by Power-Law, scale-free network structure and heavy-tailed degree distributions. Such framework is based on a known searching and local uniform caching algorithm based on bond percolation, a mathematical phase transition model well suited for random walk searches in random power law networks, which automatically shields low connectivity nodes from traffic and reduces total traffic to scale sub-linearly with network size. Our focus here is on the adaptability of the search network, dynamically accommodating changes in the data and query distribution, when nodes and data are continually joining and leaving the P2P system. Unlike other P2P information retrieval solutions, there is no need to assume that multiple copies of the shared information made available must be present on the Grid; the proposed schema is able to find the requested information reliably end efficiently, even if every node in the network starts with a unique different set of or objects.

2 Related Work

Several papers have analyzed search strategies for unstructured decentralized P2P infrastructures. Some of these strategies have been also used to implement search facilities in P2P grids [1]. Content-based search techniques include content-mapping networks [2][3][4]. In such schemes, when a peer joins a network it is assigned a responsibility to index a "zone" of the advertisement space in such a way that the union of the indices of all the peers covers the whole advertisement space. The paper in [5] explores alternatives (expanding rings and random walks) to the classical flooding search strategies whereas [6] exploits the theory of random graphs to prove properties of a generalization of the search that combines flooding and random walks. The work in [7] focuses on random walks and introduces a number of local search strategies that utilize high degree nodes in power-law graphs to reduce search time. The work in [8] quantifies the effectiveness of random walks for searching and construction of unstructured P2P networks. It also compares flooding and random walk by simulations on different network topologies. On the other side, [9] introduces a scalable searching approach for locating contents in random networks with heavy-tailed degree distributions. The analysis of the size of the giant connected component of a random graph with heavy tailed degree distributions under bond percolation is the heart of their results on which also our scheme is based.

3 P2P Grid Infrastructures

The traditional early production grid architectures are based on a service-oriented computing model with a super-local resource management and scheduling strategy. In detail, the overall control logic is based on a certain number of centralized managers that are the only entities with a complete view of the resources available on the whole Grid or on their own local management domain. On the other side, the P2P model, that has achieved wide prominence in the context of multimedia file exchange, allows the distributed computing concept to reach out to harness the outer edges of the Internet and consequently will involve scales that were previously unimaginable. The client/server architecture does not exist in a peer-to-peer system. Instead, peer nodes act as both clients and servers - their roles are determined by the characteristics of the tasks and the status of the system. Conceptually, these new computing infrastructures are characterized by decentralized control, heterogeneity and extreme dynamism of their environment. Participants frequently join, leave and act on a best effort basis. Predictable, timely, consistent and reliable global state maintenance is infeasible. The information to be aggregated and integrated may be outdated, inconsistent, or not available at all. Failure, misbehavior, security restrictions and continuous change are the norm rather than the exception. Deployment of P2P grids is entirely user driven, obviating the need for any dedicated management of these systems. Peers expose the resources that they are willing to share (i.e. a dataset) and each resource may be replicated several times, a process that is totally decentralized and over which the original peer that advertised the resource has little or no control at all. Peers can form groups with fluid group memberships. Today, the greatest enabling factor for peer-to-peer Grid architectures is the widespread availability of high-end desktop PC or Workstation always connected to the Internet that at the state of the art offer a computational capacity of 4-6 GFlops, that is expected to become in the order of 100 GFlops within the same time frame. Such a great processing power that makes it possible to execute extremely demanding applications is largely unused (at least for the most part of the day). This opens up a very interesting window for resource sharing, also sustained by the current trend of growth of the bandwidth availability on the average and high-end Internet connections making ubiquitous Internet-based peer-to-peer Grid computing one of the most valid options available in the computing arena.

4 A Scalable Search Model in P2P Grids

The fact that today most computers in peer-to-peer grids are interconnected through the Internet give us the opportunity, in formulating our replica discovery paradigm, to exploit some of the Internet characteristics, a task that can greatly benefit from physical modeling approaches. Several empirical studies on the topology of the Internet showed that the connectivity of its nodes exhibits power law attributes and scale-free behavior in degree distribution [10]. In other words, let $P(k)$ such distribution, that is the probability that an arbitrary node be connected to exactly k other nodes:

$$P(k) = ck^{-\lambda}, \quad k \geq m \tag{1}$$

with an exponent $2 < \lambda < 3$; where c is a normalization factor and m is the minimal connectivity (usually taken to be $m = 1$). Here, we can also evidence that

$$\sum_{k=2}^{k_{\max}} P(k) = 1 \tag{2}$$

where k_{max} is the maximum number of neighbors any node may have. Many naturally occurring networks (social contacts, ecological nets of predator-prey, etc.,) exhibit such degree distribution, and since several features are shared by peer-to-peer grid computing systems and these complex networks, much can be gained through integrative and comparative approaches, allowing cross-fertilization between those two important areas. Power–law networks can also be characterized by two fundamental properties [10][11]:

– a small number of links are connected by numerous nodes, while a large number of links are connected by a few nodes;
– the number of hop–counts between nodes is reduced (small–world property).

The second characteristic promotes faster propagation of information between nodes, which is a great advantage in distributed search, by optimizing the performance of the query/response traffic. However, because of the first characteristic, if several spoke nodes propagate some information at the same time or at almost the same time, the involved messages concentrate at the hub node. That is, the nodes with the very highest connectivity are subject to most of the search/query traffic. It is also likely that this tendency will increase as the number of nodes in the network increases because the number of links connected to the hub–node in turn increases. Finding a provably scalable method to search for unique content (such as the location of a replicated file) on unstructured power law networks is still an open problem. Several solutions, based on various random-walk strategies, have been proposed, but most of these reduce query traffic cost only by a constant factor. However, Power law networks are known to be an ideal platform for efficient percolation search [9], that can be a very attractive approach to ensure the needed scalability to the overall replica search/discovery system because of its relations between the probabilistic and algebraic topological properties of the Internet-based P2P organizations. According to such approach, it is always possible to overlay a scalable global-search system on this naturally scale-free graph of social contacts to enable peers to exchange their replication data efficiently. Furthermore, the query traffic cost using percolation search scales sub-linearly as a function of the system size [9]. When performing percolation search on a power–law network, we can distinguish several types of participating nodes working at different scales. Some nodes have a small number of neighbors, and thus are required to process a small number of queries passing through them. On the other end, there are nodes with large numbers of neighbors that will do a lot of work. Such search scheme implicitly makes use of high degree nodes to both cache information and respond to as much queries as possible. Thus, to achieve the best performance, one should enforce a basic form of hierarchy in the P2P Grid topological organization – that is: only high performance and huge capacity nodes have to be highly connected.

5 The Percolation Paradigm

Percolation theory [12][13] is a field of mathematics and statistical physics that provides models of phase transition phenomena that are observed in nature. Let us consider the following question, originally due to Broadbent and Hammersley, to introduce percolation theory [14]. Water is poured on one side of a large (ideally infinite) porous stone. What is the probability that the water finds a path to the opposite side? By modeling the stone as a square grid (see fig. 1 below) in which each edge can be open and hence traversed by water with probability p, and closed otherwise, independently of all other edges, one can show that for $p > 0.5$ water percolates trough the stone with probability one. One can then ask at what rate the water percolates and how it depends on. In other words, how rich in disjoint paths is the connected component of open edges? Bond percolation removes each edge in the grid with probability $1 - p$ (each edge is kept with probability p), where p is the percolation probability. The surviving edges and nodes form the percolated network.

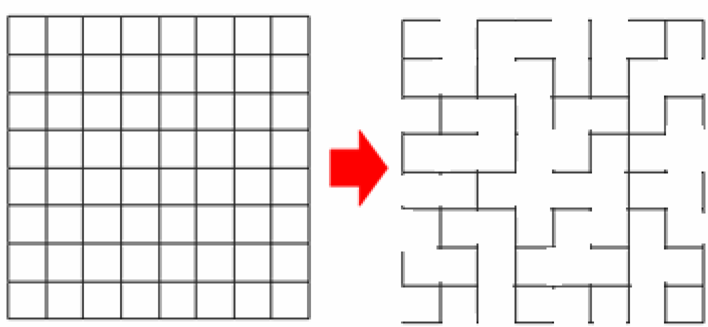

Fig. 1. Percolation on a square lattice

We can construct a mapping such that the open grid edges of a percolation model correspond to the presence of an active peering connection between the grid nodes, and the open percolating paths represent the resulting search tree on the P2P grid infrastructure that can be used to convey the requested replica location information. The percolation threshold p_c is the lowest percolation probability in which the expected size of the largest connected component goes to infinity for an infinite-size graph. If the percolation probability is lower than the percolation threshold, then the percolated network consists only of small-size connected components and lacks a giant connected component. Otherwise, if $p > p_c$ then a giant connected component emerges in the percolated network. Note that the percolation threshold is extremely small for any PL network. As a result, the size of the percolated network core is extremely small, since it is proportional to p_c. When percolation occurs above p_c, high-degree nodes are always reached during the process and become part of the giant connected component. Thus, in an unstructured P2P network search, when dataset replica information is cached in random high-degree nodes and every query originating from a random initiator node reaches and restarts from a random high-degree node, percolation can reliably find any specific dataset with an high probability (near to 1).

5.1 The Percolation Search Algorithm

Accordingly we adapted to our problem the random-walk search algorithm suggested in [9] where each node scans the contents of all its neighbors. The random walk search idea is simple: for each query, a random walker starts from the initiator and asks the nodes on the way for the requested information until it finds a match. If there are enough replicas of the needed dataset on the network, each query would be successfully answered after a few steps. In this schema, a random walker starts from a requester node to resolve a query. At each step of the walk, it scans the neighbors of the node it visits. For a power-law graph, the search quickly (with probability approaching one) converges towards high-degree nodes. Percolation search properly consists of three building blocks: *content implantation*, *query implantation*, and *bond percolation*.

Content implantation (fig. 2-a) means that every peer node in a network of n nodes announces its content through a short random walk of size $O(\log n)$ starting from itself. Only its own contents are duplicated (cached) on any node visited along the way. Clearly, highly connected nodes will develop larger caches. In fact the cache sizes obey exactly the same distribution as the degrees. The total number of contents is hence $O(n \log n)$ and the average cache size is $O(\log n)$.

A Query implantation (fig. 2-b) is executed each time a peer node issues a query. This means that the query will be ”implanted” on a small number of peers through a random walk of size $O(\log n)$ starting from the requester. Content and query implantation ensure that both content (in our case a replica location information) and queries are known by at least one high-degree node, since a random walk in a power-law network gravitates towards high-degree nodes because they have a higher number of incoming links. The search is finally executed in the bond percolation step (fig. 2-c). Each peer which has an implanted query will perform a probabilistic broadcast, in which it sends the query to each of its neighbors with probability p. Probability p is set to such a value (usually a constant multiple of the percolation threshold p_c) so that a query starting from a high-degree node will be broadcast to the so-called giant component of the network (which all high-degree nodes are part of with a high probability). Clearly, if a query reaches a node which has already received the same query from another neighbor, the query is not implanted, thus avoiding loops in the query path. Since content implantation ensures that if each resource is known by at least one high-degree node, it will be found with a probability very close to 1. The information resulting from a successful search process reaches backwards the request originator node through the same path by which the query message arrived at the hit node (fig. 2-a). Every step in the algorithm is totally local and truly uniform. High degree nodes will be naturally “distilled” from on the power law network.

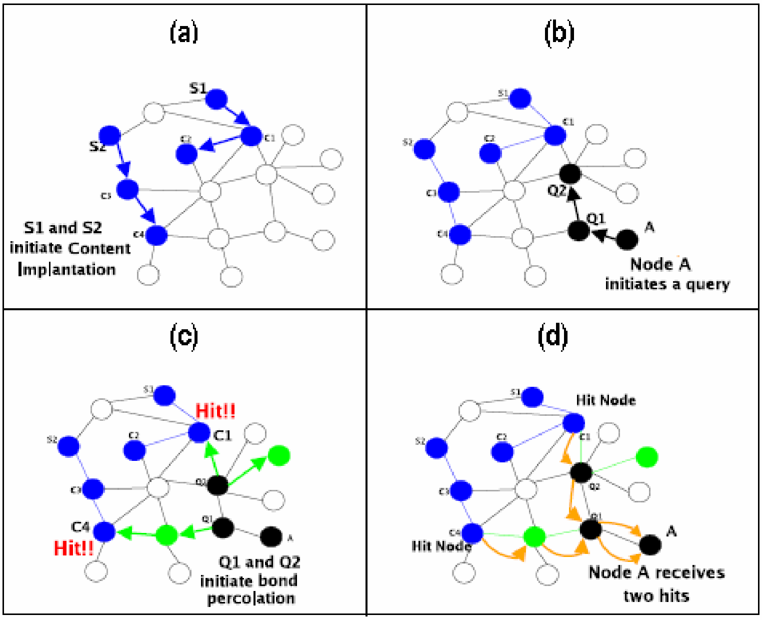

Fig. 2. Percolation search example

The search networks generated by the above algorithm form random connected graphs, where nodes are connected to few random neighbors, and have a relatively small distance between each other. The average number of hops between nodes is approximately equal to $log(n)/log(k)$, where n is the number of nodes and k is the average node degree. The existence of a small network diameter gives only a lower limit to the number of hops between two peers. The fact that the network has a power law distribution of the edges, even performing a random walk from node to node, will result in significant reduction in the number of nodes visited. This is because a random walk will tend to select high degree nodes. However, specifically choosing high degree nodes to traverse first, improves search further. Such search networks can be essentially controlled by two parameters: the exponent λ of the power–law, and the maximum degree k_{max}. Another important parameter greatly conditioning the efficiency of the whole search process is the percolation threshold p_c. In detail, in power-law networks of a finite size, the percolation threshold approaches 0 and can be calculated [15] from a degree distribution, as:

$$p_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{3}$$

Here, k stands for the degree of a node, and the notation $\langle \ldots \rangle$ means the average over the degree distribution. Finally, since any content in the network can be found with probability one in time O(log n), while generating only $O(n \times \frac{2 \log k_{\max}}{k_{\max}})$ traffic per query, and since in random PL networks $k_{max} = c \times n$ then the overall traffic scales sub-linearly with the network size as $O(\log^2 n)$ per query [16].

5.2 Implementation Issues

While the design of the above search algorithm is based on theoretical concepts, the final formulation is straightforward and hence very easy to implement through dedicated search agents. The multi-agent technology has features well fitting for distributed communication, and is particularly robust for the interaction and negotiation tasks within P2P organizations. In a distributed agent framework, we conceptualize a dynamic community of agents, where multiple agents contribute services to the community by cooperating like individuals in a social organization. The appeal of such architectures depends on the ability of populations of agents to organize themselves and adapt dynamically to changing circumstances without top-down control from a central control logic. At first, each grid node that needs to export a dataset replica activates a specific content implantation agent that periodically starts a short random walk throughout its neighborhood and replicates on the traversed nodes the involved dataset content. Such an agent lives for the entire lifetime of each active replica. Clearly the data replication will be performed on a node along the short walk only if it is really needed (the information is not already present) and feasible (there is available space and the replica operation is locally authorized). When a job requests a specific object resource in the grid through a specific web service interface, a job-related search agent will be created. This agent will be in charge of finding candidate dataset replicas through the proposed percolation-based interactions, lives until the associated search task executes and will be dissolved when it is finished and the replica results are sent to the requesting node and cached on the intermediate nodes along the search tree. Clearly, an aging timeout is associated to each cached replica so that when such a timeout expires and the information has not been refreshed by another content implantation or during the backward query/response process, the entry is no longer valid and hence deleted. It should be noted that a search agent might be involved in several query implantation activities during the same percolation search process. Since cooperation, negotiation, and competition are natural activities common in multi-agent systems the above content and query implantation functionalities and the following percolation search process are naturally implemented by using the agent oriented approach. The percolation search network needs to be overlaid on top of the peer-to-peer interactions between the grid nodes built during both the content implantation and query processing activities and the involved agents cooperate by implementing the above search steps by interacting through an existing P2P grid communication paradigm. Communication between neighbor peers can be enabled by existing P2P interaction facilities such as JXTA [17] (from juxtaposition) or P2PS [18] (Peer-to-Peer Simplified) protocols. JXTA provides features such as dynamic discovery while allowing peers to communicate across NAT, DHCP, and firewall boundaries. It is independent of transport protocols and can be implemented on top of TCP/IP, HTTP, TLS, and many other protocols. P2PS is a lightweight infrastructure for developing P2P style applications whose architecture is inspired to and provides a subset of functionality of JXTA. In both the solutions a peer is any node that supports the specific interaction protocols and could be any digital device. Peers publish the existence of a dataset resource through an advertisement, which is simply an XML document describing the resource. Interactions between agent peers are self-attenuating, with interactions dying out after a certain number of hops. These attenuations in tandem with traces of the peers, which the interactions have passed through, eliminate the continuous echoing problem that results from loops in peer connectivity. In such environment, the specific agents described before will be implemented at the P2P Grid middleware level on each local grid node involved in the above percolation-based replica discovery and management framework. These agents provide a high-level representation of the corresponding search or replica location notification capability. Hence, they also characterize the available dataset resources as information service providers in a wider grid environment. Agents can be structured within the proposed architecture according to a simple layered model (see fig. 3).

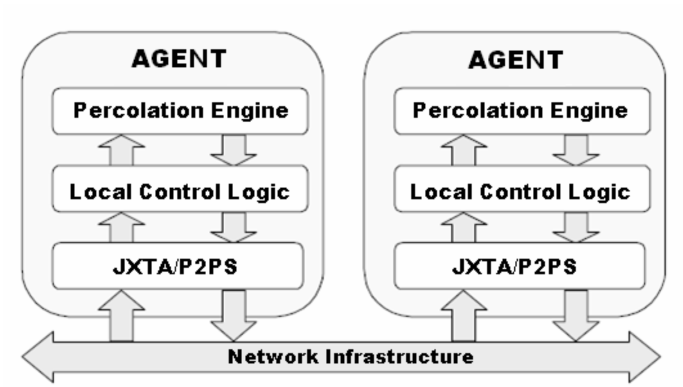

Fig. 3. The generic agent layered structure

Here we can distinguish a bottommost communication layer, implemented through the above JXTA or P2PS facilities, realizing all the Agent-to-Agent interactions and peer communication mechanisms. An intermediate local control layer is needed for all the authorization and management tasks to be autonomously performed at the individual node level (local resources management, authorization and access-control policies enforcing, data replication etc.). Finally, a “percolation engine” at the uppermost layer realizes all the main specialized agent functionalities, starting from implementing the random walks for content implantation to replica information caching and performing percolation search through “probabilistic broadcast” query implantation activities and backward result propagation.

6 Functional Evaluation

In order to evaluate our model through simulation, we generated a random network with a power law link distribution by using the Pajek [19] environment. The generation process is based on generalized Barabasi-Albert construction model [20], which presumes that every vertex has at least some baseline probability of gaining an edge, to generate edges by mixture of preferential attachment and uniform attachment. For generating condition, we set the total node number to 30000, λ=2 and a maximum degree of 6. We worked with a TTL value varying from 15 to 25. Such parameter can be thought as an upper bound of hops on the query implantation. In order to evaluate the results, we analyzed under varying percolation probabilities the behavior of the following three metrics expressed as percentage values: the success rate in finding an object, the number of edges and the number of nodes traversed throughout the search process. We first evaluated our replica search algorithm with a single copy of a content randomly located on the network (Fig. 4).

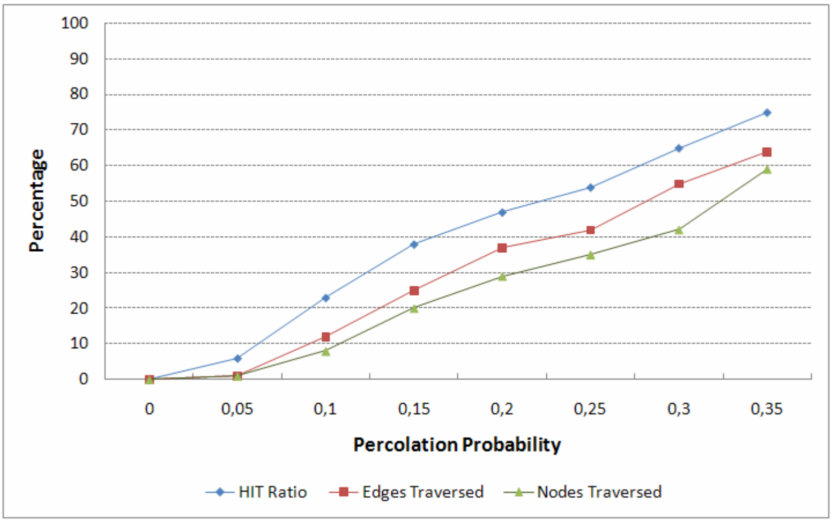

Fig. 4. Simulation results with 1 replica, TTL=15

Here no content implantation step is performed by the owner node and no information is cached throughout the network during the search process. We can observe that the percolation search mechanism is however effective but it really converges only when percolation probability approaches to 0.4 and an unacceptably high number of nodes and edges are traversed to reach the needed content. In fact the hit ratio and the traversed edges/nodes trends grow together almost linearly with the percolation probability. Next we also considered another relevant issue: what would be the improvement in performance if multiple high degree nodes in the network had the same content so that the above percolation paradigm may be really effective. Accordingly as a part of the percolation search algorithm, we execute both the initial content implantation and the following caching steps that make sure that any subsequent query step would find any content with probability approaching one. Figure 5 below shows the case where 10 replicas of any content are randomly spread in the network. We can note that, in presence of a sufficient number of replicated contents, also a slight increment in percolation probability improves the hit rate exponentially until it rapidly reaches 100%.

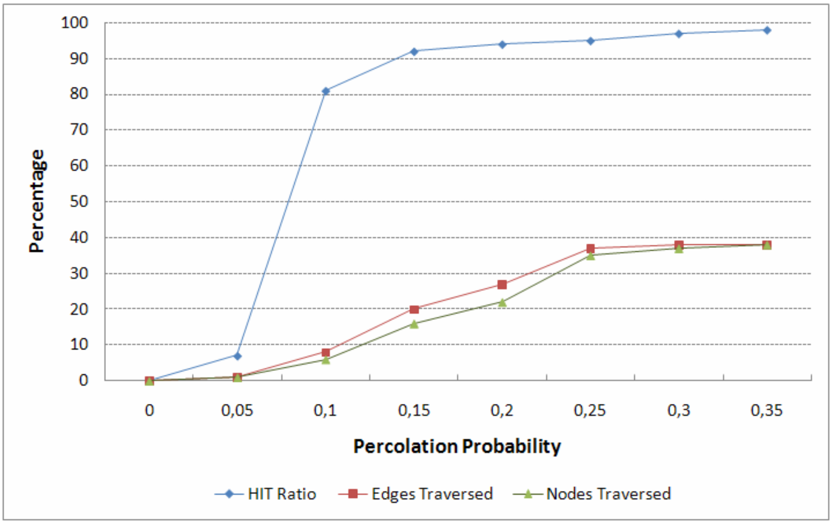

Fig. 5. Simulation results with 10 replicas, TTL=15

7 Conclusion

This work focuses on searching for replicated objects (files, resources, etc.) in fully decentralized P2P grids, i.e., where a (large) set of peers, can exchange information in absence of any central service. Accordingly we propose a scalable search model for such totally unstructured grids based on a known algorithm that uses random-walks and bond percolation on random graphs with heavy-tailed degree distributions to provide access to any content on any node with probability one. We analyzed the validity of our model and examined its dynamics through simulations. We can conclude that our proposal can be effective in both reducing the total amount of queries/checks and ensuring an high success rate, which means we can provide a robust and effective search platform for emerging P2P grid infrastructures.

References

1. Botros, S., Waterhouse, S.: Search in JXTA and other distributed networks. In: P2P Computing conference (2001)
2. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable Network. In: ACM SIGCOM (2001)
3. Dabek, F., Brunskill, E., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I., Balakrishnan, H.: Building peer-to-peer systems with Chord, a Distributed Lookup Service (2001), `http://pdos.lcs.mit.edu/chord`
4. Zhao, B.Y., Kubiatowicz, I., Joseph, A.: Tapestry: An infrastructure for fault-tolerant wide area location and routing", Computer Science Department, UCB Report UCB/CSD-O 1-1141 (2001)
5. Lv, Q., Cohen, E., Li, K., Shenker, S.: Search and Replication in unstructured peer-to-peer network. In: International Conference on Supercomputing (2002)

6. Gkantsidis, C., Mihail, M., Saberi, A.: Hybrid Search Schemes for Unstructured Peer-to-Peer Networks. In: IEEE Infocom 2005. IEEE CS Press, Los Alamitos (2005)
7. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in Power-Law Networks. Physical Review E 64 (2001)
8. Gkantsidis, C., Mihail, M., Saberi, A.: Random Walks in Peer-to-Peer Networks. In: IEEE INFOCOM 2004, Hong Kong, China (2004)
9. Sarshar, N., Roychowdury, V., Boykin, P.O.: Percolation search algorithm, making unstructured p2p networks scalable. In: Fourth IEEE P2P 2004, Zurich, Switzerland. IEEE Computer Society Press, Los Alamitos (2004)
10. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power–law relationships of the Internet topology. In: Proceedings of ACM SIGCOMM 1999, October 1999, pp. 251–262 (1999)
11. Barabasi, A., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
12. Grimmett, G.: Percolation, 2nd edn. Springer, New York (1999)
13. Meester, R., Roy, R.: Continuum Percolation. Cambridge Univ. Press, Cambridge (1996)
14. Broadbent, S.R., Hammersley, J.M.: Percolation processes in Crystals and mazes. Proc. Cambridge Philosoph. Soc. 53, 629–641 (1957)
15. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 167–256 (2003)
16. Sarshar, N., Boykin, P.O., Roychowdury, V.: Scalable percolation search on complex networks. Theoretical Computer Science 355, 48–64 (2006)
17. Sun Microsystems. The JXTA project and peer-to-peer technology (November 2004), `http://www.jxta.org`
18. Wang: P2PS (Peer-to-Peer Simplified. In: 13th Conference - Frontiers of Grid Applications and Technologies, Louisiana State University, pp. 54–59 (2005)
19. Pajek project, `http://vlado.fmf.uni-lj.si/pub/networks/pajek`
20. Albert, R., Barabasi, A.: Topology of Evolving Networks: Local Events and Universality. Phys. Rev. Lett. 85, 5234–5237 (2000)