Organization of Simulation Experiments for Information Servicing Investigation in Distributed Resource-Sharing Environment

Radi Romansky, Iva Nikolova

Abstract: *The objective of this paper is to present the application of simulation techniques for organization of experiments for performance investigation of the distributed information servicing. The paper considers the simulation framework, which allows estimating the effectiveness of different approaches for request routing for resource discovering in the context of decentralized, dynamical resource-sharing environments. The modelling gives us an approximate picture of expected searching performance. To evaluate the performance of different resource-discovering techniques, we conduct simulation experiments, using in Any Logic simulation environment and some results are presented and discussed.*

Key words: *Information servicing**,** Distributed environments, Resource-sharing networks, Distributed Resource discovery*

1. INTRODUCTION

In recent years, considerable attention has been given by research community to the design of resource-sharing networks and technologies. Some of the keys enabling technologies are presented in [1-3]. Opportunistic sharing of network-connected resources is a low cost method for obtaining access to large-scale collection of resources. This sharing may involve not only file exchange, but also direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies [4, 5].

Resource distribution describes the degree to which a system has been spread across a network, and how dynamic resources are within it. In order to access resources, it must possible to first to locate them. Therefore, an essential service in any resource-sharing environment is the resource discovery. Discovery is supported via search capability: given a description of the resource desired, resource discovering mechanism return locations of resources that match the description. These descriptions can have various forms. Search objects can be different kind resources: storage spaces, files, and services, applications and so on. Their discovery in the conditions of dynamically changing resource-sharing environments can be complicated because of the: possibly dynamics of some resource characteristics within a system as accessibility (for instance, the location of a service may have to be determined through a directory service and refreshed with each successive access), loading, sharing policy and etc.; randomicity (users don't know where the objects are, whether they exists, or which technique is the most suitable for locating them); usage patterns (not having real user request logs) and etc. All these create significant difficulties for traditional centralized and hierarchical resource discovery services and therefore some new mechanisms must exist to find resources in ongoing manner.

The way in which the resource discovering mechanisms are assigned to the resource-sharing environment to find appropriate resource that match to the user descriptions and the number of nodes that are to be searched have a significant influence on both response time of the system and the efficiency exhibited by the search algorithms. The performance of any resource discovery mechanism [6,7] in dynamic distributed environment is influenced by four environment parameters: (1) *Resource information distribution*: some nodes provide information on a large number of resources, whereas others just on few; some resources are common, while others are rare or even unique. (2) *Resource information dynamism*: new nodes can bring new types of resources (new data, new applications, on-line instruments and etc.); some resource attributes are highly variable, while others can be considered static. (3) *Request popularity distribution*: the popularity of users' request for resources varies [8, 9]. (4) *Nodes participation*: the natural tendency of dynamically changed environments is that they evolve over a time with joining / leaving server nodes along their resources.

Many research works in the area of distributed search services show, that in dynamic environments it is more efficient to forward request [6-10]. In connection to this, the paper addresses the organization of modeling studies for performance investigation of resource locating techniques. The object of our current research interest is dynamical, decentralized information and computational space, which to integrate the resources that are shared by a group of users for support of research and educational process within the intranet network. The modeling presents an approximate picture of expected performance of applying different resource-discovering techniques, based on request routing and helps to better understand the tradeoffs between communication costs and performance. The effectiveness of the investigated algorithms is assessed by means of simulation experiments in which four mechanisms were compared. These simulation studies are conducted by means of simulation models, implemented in *AnyLogic* simulation environment. Some results, obtained during the simulation experiments are presented and discussed.

2. SIMULATION FRAMEWORK AND EXPERIMENTAL SETUP

The investigated distributed resource-sharing space is formed by the network integration of the n nodes. The system architecture is based on the decentralized model of P2P content-sharing networks [11]. It has a Server-Client structure where there is no centralized servers exists. Figure 1 and Figure 2 show the architecture of the resource-sharing network and a simplified block diagram of the simulation framework for the study, implemented on the base of our earlier proposed conceptual model [12]. The simulator models: a set of servers' nodes (a collection of distributed nodes that store and provide access to local resource information); a set of resources, published by nodes (there may be multiple resources with identical descriptions); a set of distinct user requests; a set of request-routing algorithms.

We simulated the process of submitting and servicing a set of user requests in a network of N_{Nodes} server nodes. All nodes are empowered as both - a client (may request services from others) and a resource server (may offer services to others). The nodes also have possibilities to act as routers, forwarding requests they cannot service.

The network infrastructure allows a multiple requests to be serviced simultaneously by node. Particular nodes communicate in between via sending and receiving messages, marking the request type and the received answer. Each node maintains two types of information: a) about a number of resources it contributes and b) about other nodes in the system.

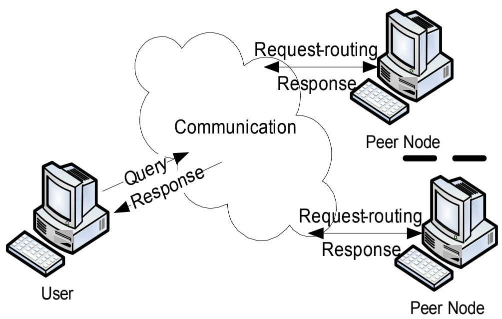

Fig.1. Network Architecture

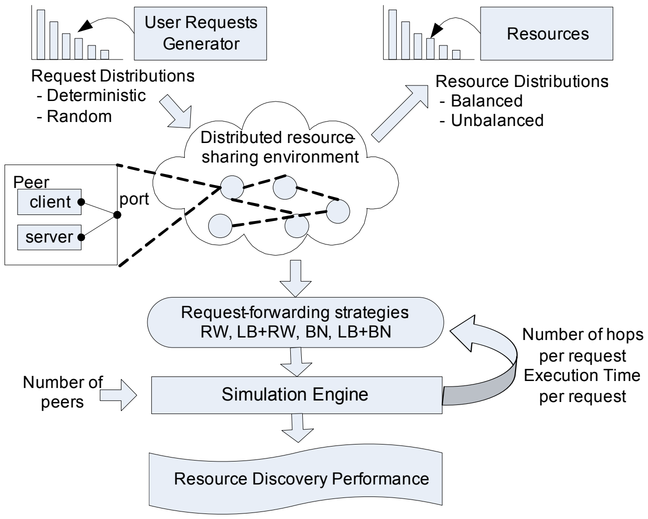

Fig.2. Simulation Framework

For the workload model, it assumes that every participating node provides search functionality on N_{res} resources (is assumes that the average number of resources per node remains constant with the increase in the network size). Each resource $r_j \in R$, $0 \le i \le N_{res}$,, matches a randomly chosen set of queries $q_i \subset Q$ of *D* different types. The query kind is chosen randomly according to uniform distribution $p(x) = 1/(b-a) : a \le x \le b; 0 - otherwise$. In our experiments we randomly chose a fixed percentage of nodes, respectively, were repeated to compare various resource discovery strategies.

The number of queries N_q in a batch arriving is chosen randomly accordingly a geometric distribution $p(x) = p(1-p)^{x-1}, x = 1,2...$. In this context, we experimented with two resource distributions: a balanced distribution (all nodes providing the same number of resources), and unbalanced one (a small part of nodes have large number of resources, while the others one providing only one). The tradeoff between the amount of information about neighbors and search performance generate a large set of alternatives – from random forwarding (no information about resources provided by other participating nodes) to one-hop forwarding (when nodes know exactly which node has the requested resource). It considers a set of simple resource discovery mechanisms varying the request-processing component. This component is based on forwarding. It is assumed simple requests. The requests may be processes locally or remote – refers to the propagation rules. It assumes that users send their requests to some known (typically local) node. A node that has a matching resource responds to the requester; otherwise, it decrements *TTL* and forward it (if TTL>0) to some other node until its *TTL* expires or matching resources are found. Requests are dropped when received by a node with no other neighbors or when *TTL=0*.

Four request propagation strategies, based on randomness and or historical information about the result set size of prior queries are simulated and analyzed: (RW) *Random walk*: the node to which a request is forwarded is chosen randomly. No extra information is stored on nodes; (LB+RW) *Learning-based* + *Random Walk*: nodes learn from experience by recording the request answered by other nodes. A request is forwarded to the peer that answered similar request previously. If no relevant experience exists, the request is forwarded to a randomly chosen node; (BN) *Best neighbor*: the number of answers received from each node is recorded (without recording the type of request answered). A request is forwarded to the node that answered the largest number of requests. First requests are always sent to random nodes; (LB+BN) *Leaning-based plus best neighbor*: this strategy is identical with the learning-based strategy except that, when no relevant experience exists, the request is forwarded to the best neighbor instead of to a random node. The parameters used in experiments are: *Total Simulation time=5000 ms*; *Number of nodes* N_{Nodes} *- 10 ÷200*; *Number of resources* $N_{res,}$*=20*; Number of queries N_q-. p=0.2 –probability of success on any given attempt associated with a batch of x=5 size.

3. EXPERIMENTAL STUDIES AND SIMULATION RESULTS

All described above algorithms for request forwarding have been systematically simulated to obtain their performance. For each request-routing algorithms (fixing the intensity of request generation) performance measuring have been obtained using balanced and unbalanced distribution of resources. We are interested to quantify the cost of resource location, measuring the average response time in environments with different resource-sharing characteristics. Some of the results are illustrated by figures bellow.

♦ *Effect of resource and request information distribution (Fig. 3)*: Figures a-d show the effect of resource distribution for balanced and unbalanced resource sharing. As expected, the LB+RW is the best performing in both cases with average number of hops per request tend to 1, i.e. hit the resource location. Similar are the results obtained using LB+BN. Key of the performance of these strategies is the fact that they take advantage of similarity in requests by using a possibly large cache. The least efficient in case of balanced resource-sharing are BN (that is because of the small number of different resources per node), followed by RW, nevertheless it has the advantage that no additional storage space is required on nodes to record history. In the case of unbalanced resource distribution the least efficient is RW.

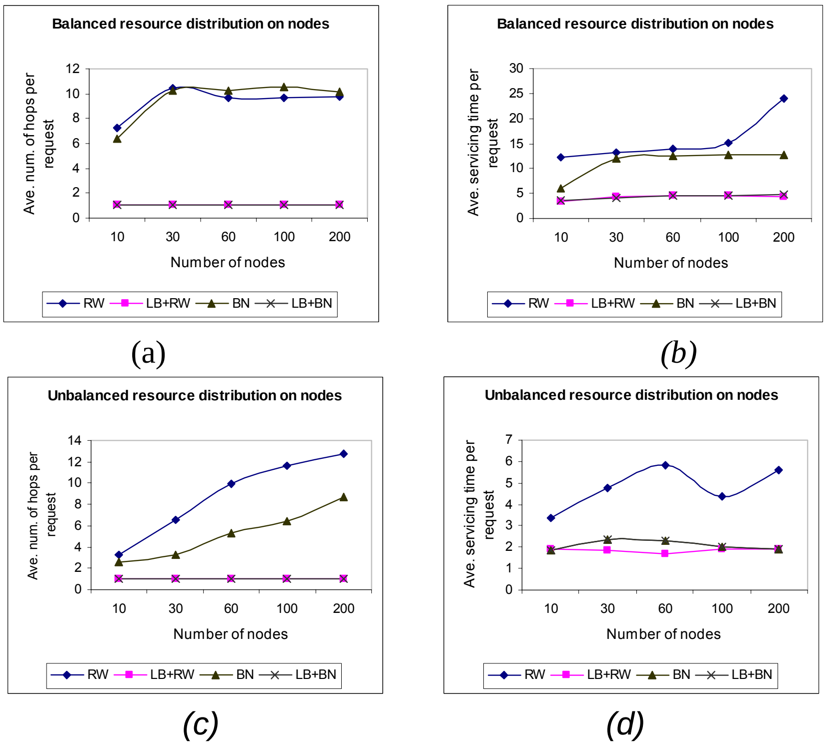

Fig. 3. Effect of resource and request information distribution

♦ *Effect of resource distribution and node participation (Fig. 4):* Figures a-d show the effect of node participation and resource information distribution on the percent of local processed requests and the lost ones. Figures compare the results among the four resource discovery techniques. It is observed that when the number of nodes is small (n=10) and the resource frequency is low (there is a request for resources that do not exist) the approximately 30% of requests are dropped (Fig. 4a). With increasing the number of nodes and the resource frequency this percent tends to 0. For unbalanced mode, when the request frequency per node varies (one service / several services) are: 15% / 6% (RW), approximately 0% (LB+RW, LB+BN), 8% / 1% (BN) (see Fig. 4(c)). The number of locally serviced request per node for balanced mode is constant (approximately 10%), because of the number of resources per node is constant – 10 (Fig. 4b).

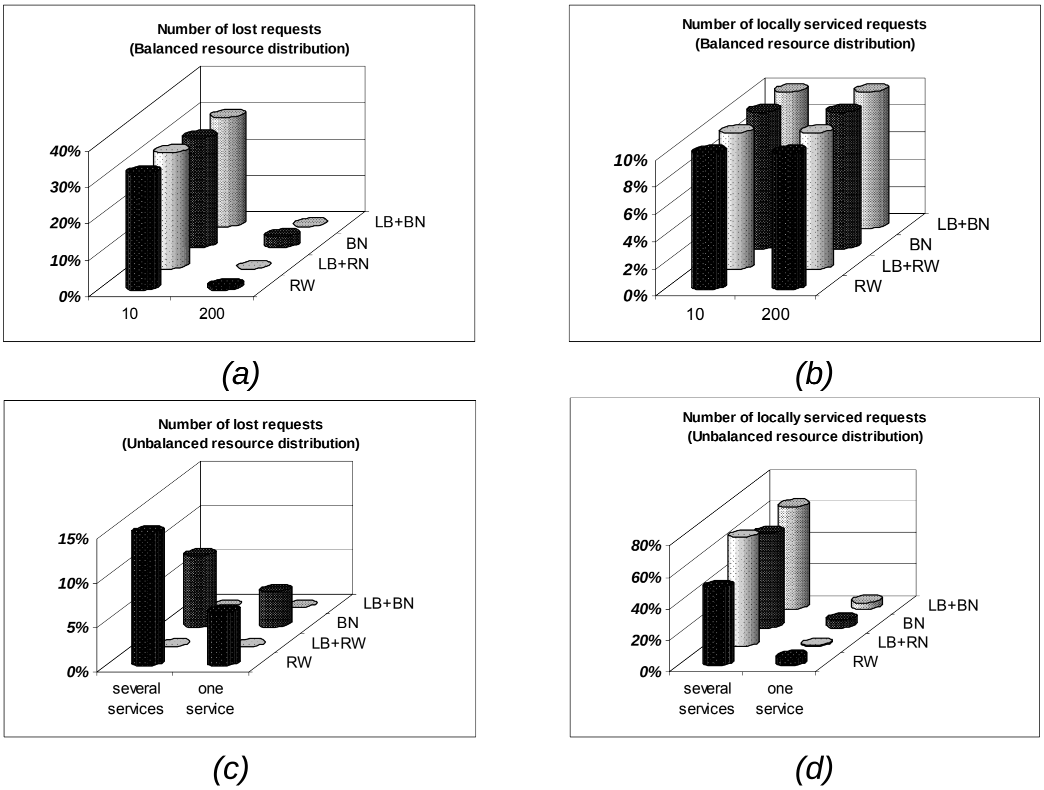

***Fig. 4*.** Effect of resource distribution and node participation

CONCLUSIONS

The paper deals with investigation of decentralized approach for resource discovery in distributed resource-sharing environment, in which participating nodes maintain and publish information about possibly large set of resources. The effectiveness of the investigated algorithms for resource discovery, based on request forwarding was assessed by means of simulation experiments in which four mechanisms were compared. In the comparison, we considered a set of different factors in modeling the behavior of these strategies to the resource discovery in distributed environment that rely on decentralization and the heterogeneity of shared resources. These results could be used to advocate one strategy over another if the cost is decisive factor. In this context, we have observed that for all network sizes in our experiments, if the cost is of importance, the learning-based is the most successful, because it is independent from the resource distribution. Otherwise:

at the balanced resources distribution random walk could be used; at the unbalanced – best neighbor. Furthermore, these estimates show another important thing, namely that with increasing the number of nodes (for example in large-scale networks) more sophisticated mechanisms with a request-propagation component are needed for efficient resource location.

REFERENCES

[1] Oram A., Peer-to-peer: Harnessing the Power of Disruptive Technologies, O'Reilly 2001, ISBN 0-596-00110-X
[2] Foster, I., Kesselman, C. & Tuecke, S."The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of High Performance Computing Applications, 15 (3), p 200-222, 2001.
[3] Casanova H, "Distributed Computing Research Issues in Grid Computing", ACM SIGAct News, Vol. 33, No. 3, 2002, pp. 50 – 70
[4] C. Shahabi, F. Banaei-Kashani "Decentralized Resource Management for a Distributed Continuous Media Server, IEEE Transactions on Parallel and Distributed Systems, Vol. 13, #6, June 2002.
[5] K. Czajkowski, S. Fitzerald, I. Foster, and C. Kesselman. "Grid information services for distributed resource sharing". In 10th IEEE International Symposium on High Performance Distributed Computing, 2001, San Francisco CA, IEEE Press.
[6] Kutten, S., and Peleg, D. "Deterministic distributed resource discovery". In 19th ACM SIGACT/SIGHOPS Symposium on Principle of Distributed Computing (Portland, Oregon, 2000)
[7] Rowstron, A. I. T., and Druschel, P. Pastry: Scalable, decentralized object location, and routing for large-scale P2P systems. In Middleware (2001), p. 329-350.
[8] K. Spripanidkulchai. "The popularity of Gnutella queries and its implication on scalability", Proc. O'Reilly Peer-to-peer and Web Services Conference, 2001
[9] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S. (1999) "Web caching and Zipf-like distributions: Evidence and implication. In InfoCom (NewYork, 1999), IEEE Press.
[10] Hoschek, W "A Comparison of Peer-to-Peer Query Response Modes"., Int'l. Conf. on Parallel and Distributed Computing and Systems (PDCS 2002), Cambridge, USA, November 2002.
[11] Peter, B., Tim, W., Bart, D., & Piet, D., "A Comparison of Peer-to-Peer Architectures", Broadband Communication Networks Group (IBCN), Department of Information Technology (ITEC), Ghent University, Belgium, 1-2, 2002.
[12] Romansky, R., Nikolova I. "Conceptual model for investigation of the information servicing in a distributed informational-learning environment", In Proc. of CompSysTech'2004, Rousse, Bulgaria 17-18 June, 2004

ABOUT THE AUTHORS

Associate Professor Radi Romansky, Ph.D., Department of Computer Systems, Technical University-Sofia, Phone: +359 2 965-32-95, E-mail: rrom@tu-sofia.bg.

Assistant Professor Iva Nikolova, Department of Computer Systems, Technical University-Sofia, Phone: +359 2 965-26-80, E-mail: inni@tu-sofia.bg.