

POSITIVE EFFECTS OF PROACTIVE COMMUNICATION IN MAS

Félix Ramos

CINVESTAV GDL Lopez Mateos Sur 590 Guadalajara Jalisco México

Abstract: This paper presents an adaptive communication in MAS. To achieve its objectives agents must send and receive information. Most used mechanism of communication in MAS is broadcast but this is quite expensive and prohibited in some sort of applications mainly because the very poor quality of the service. In this paper we describe through a real problem the positive effects of proactive communication in Multi-Agent systems. Even if in this paper we show an example, results can be extended to different problems where negotiation is important.

Key words: Proactive communication, Multi-agent System, Load Sharing, QoS.

1. INTRODUCTION

This paper presents an adaptive communication control in a MAS distributed in geographically. To achieve its objectives agents must send and receive information, a problem is how to make efficient the communication, that is make communication only when necessary and just to whom must be communicated. Today most used mechanism of communication in MAS is broadcast, because it is useful in different phases of a team work [5], for instance the discovery of services and the negotiation process. But broadcast is quite expensive and prohibited in some sort of critic applications because the very poor quality of the service and security problems.

Nowadays several systems use services provided in an ad hoc fashion at fixed locations. Then one first approach is communicating well defined

from the beginning. This approach is not well suitable for dynamic systems like those of Internet where more services are available day after day. Another approach is to establish communication among agents taking into account the distance among services. In general statistic methods have no good experimental results as shown in other studies for assignment of request in Internet [2] because they ignore the network's path dynamic conditions. There are dynamic strategies for assignment of request in internet is taken into account the maximum of parameters like network properties distance replica and load of services [1,3, 4, 5, 6, 7].

The objective of our work is to study how proactive communication can improve the communications cost and improve quality of service. Our solution includes an Agent's knowledge model; this model offers the possibilities specify the relevant metrics to be used as quality of the service and reduce communication cost. In this paper we show using an example of load sharing how proactive communication is useful to reduce communication cost and improve the quality of service. The main contributions of our method of Dynamic Request Placement approach using proactive communication among agents are:

- The ability of agents to adapt to the dynamic behavior of Internet and user preferences; this allows agents to adapt the response time estimation to the QoS requirement of user's requests
- The experimentation in a realistic scenario where actual Internet conditions are considered as high variability, self similarity, hourly effects and servers load.

1.1 Quality of Service

The International Organization for Standards (ISO) [17] defines QoS as a concept for specifying how good the offered networking services are [9]. Generally, QoS parameters are performance measures and the set of parameters for a chosen service determines what will be measured as QoS. We concentrate on the QoS perceived by users specified in terms of:

- Network Performance, describing the requirements that the network services must guarantee, it might be expressed as end-to-end delay,
- Service Performance: characterized for the requirements that the Web service must guarantee expressed as the Web service response time.

2. SYSTEM ARCHITECTURE

In this section is devoted to describe the system's architecture, the agent structure, the knowledge model used and the request placement techniques under examination used among this our proactive communication.

2.1 System Architecture

We consider as in Internet a set of servers each with his own identity and whose service is replicated through common redundancy mechanisms as mirroring or caching. Figure 1 depicts the model of architecture we proposed in [8] showing how a set of Service Agents provides a number of services with a defined QoS.

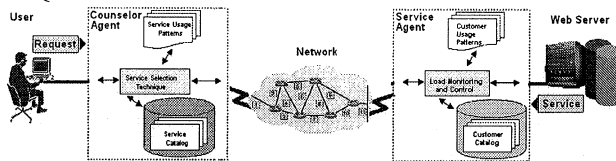


Fig. 1. System's Architecture.

Each agent in the system is an independent self-contained, concurrently executing thread of control encapsulating a perceived system state. Agents communicate with its environment and with other agents via a common communication protocol. The Internet is the environment where agents interoperate to improve QoS in the case presented in this paper reducing response times perceived by users and load balancing Service Agents. The description of the community of agents is:

1) *Counselor Agent*: As depicted in Figure 1, a counselor agent is constituted of an interface to get the request and a service solver. Each Counselor Agent implements a Service Discovery mechanism, a communication manager to be able to communicate with other agent using common models and communication protocols. The basic implementation of a counselor Agent uses a MPI (Message Passing Interface), a user interface to intercept requests from users and deliver received responses, a Service Directory to resolve service requests, a Knowledge Model to update the QoS service repository. Finally at the core of the counselor agent it's the Request Placement Strategy to make dynamic placement decisions to a selected server depending on the selected technique.

2) *Service Agent*: A Service Agent is located on each Web Server and is responsible for controlling access client requests and service load levels by means of monitoring QoS. It's also responsible for appropriately market QoS negotiation protocols avoiding a suboptimal load balance due to the Counselor Agents behavior trying to optimize its individual QoS obtained from popular Web services. Figure 1 illustrates the Service Agent

component structure inside we have the QoS information Scheduler who is responsible for the dissemination of QoS information to Counselor Agents in order to enforce load balancing. As part of the Service Agent component structure there's also a QoS passive monitor. Finally, the main component of the service Agent is the Service Manager who incorporates all the necessary components to control access to services, load balancing and forwarding of responses to Counselor Agents.

2.2 Service Discovery Mechanism

When users request a service, the Counselor Agent checks its knowledge base (i.e. service directory) to determine whether or not it is aware of the existence of the re-quested service. If, based on its knowledge, the Counselor Agent knows about the requested service then it will contact the corresponding Service Agent directly. But if the Counselor agent can't determine the existence of the requested service, it will start a Service Discovery mechanism to obtain the currently available services and update its knowledge base. The Service Discovery mechanism is implemented based on two popular spanning tree techniques:

- MST. Multiple Spanning Tree, and
- NPR. Non Predetermined Root methods described in [9].

2.3 Knowledge Model

The Knowledge Model objective is to maintain and estimate of services QoS in the Service Directory in order to improve whole QoS of the system. The Knowledge Model is based on statistical information of the use of Web services. For each service it stores, as a time series, the identity of the Service Agent, the QoS, and time validity for the perceived QoS. This last parameter enables Counselor Agents to consider old information when it becomes obsolete. An example of the knowledge base of a Counselor is displayed in figure 2.

The background to store QoS as a time series is bounded to the well known daily effect on Web services [9], bandwidth and latencies depends on time of day and day of week. This means that each Counselor Agent stores its QoS parameters as a time series, which is sampled at different hours, and deduces a function representing the QoS by interpolation dealing with large skew, long tails and high variability. Figure 2 shows such QoS as a function of day and time of the day. Whenever a new response is received for an allocated request, the QoS is stored at specific time H of day D, this new value is used to update the Counselor Agent knowledge as follows considering the memory of the agent.

$$q_{x(D, H)} = \mu * (q_x - 1_{(D, H)}) + (1 - \mu)q \quad (1)$$

Where μ is comprised between 0 and 1, and represents the Agent capability to keep a reference of past measured QoS. Additionally when service information has not been updated before some delay V , it is considered obsolete. In that case, a Counselor Agent must start a new learning phase based on relevant knowledge.

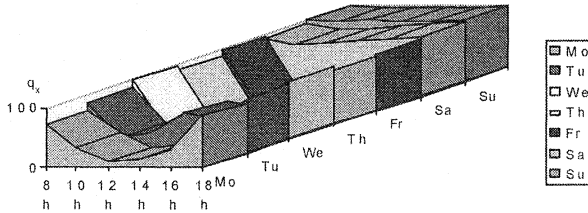


Fig. 2. QoS as a time series function.

2.4 Dynamic Server Selection Strategies

In this section we describe the techniques used to demonstrate how proactive communication is useful to improve the quality of service. The problem of finding a request placement technique that improves system QoS perceived by the system users has been extensively investigated analytically, empirical, and via simulation [4, 10, 11]. We can classify the request placement techniques in the literature into static, statistical and dynamic. Static techniques have proved to be useful to well parameterized architectures so are not useful for the sort of systems we are interested in. On the other hand, dynamic placement strategies use small probes as key in the negotiation process to detect the current network and service conditions in order to make a decision based on the current perceived.

We focused in showing by simulation that proactive communication is the key in the dynamic techniques to improve QoS because it is a factor for adapting automatically changes in necessities of clients and servers. The contribution of this study is the comparison in performance of three algorithms concurrently getting different services from different servers hosting a Web page instead of getting all the required services from a Web Page in the same server. The difference among the algorithms is that they use different knowledge levels and negotiation mechanisms. First technique rather does not use knowledge we name this random strategy. The second use a simple mechanism like free market and third one use knowledge of both clients and servers to create a proactive communication.

2.5 Random Technique

The first strategy, we refer as RND, lies under the basic idea that Counselor Agents take advantage of the Service Discovery mechanism and the resulting Service Directory uniformly assigning the requests to servers randomly. A first guess is that this technique can overload some servers.

2.6 Market-based Technique (MKT)

The MKT is the second strategy we use to compare and show how proactive communication improves the QoS. This technique is based on the contract net negotiation protocol. Using contract net in our example, for every Web service request Counselor Agents starts a QoS negotiation process [12]. Three steps distinguish such process: first a request-for-bidding is launched to all Service Agents proposing the service. Second, Service Agent replies are evaluated; these replies are also used by the Counselor Agent to refresh its knowledge on the QoS of the service requested. Finally, the contract is awarded to the Service Agent comprising the best QoS. Obviously this strategy is general and simple, but quite expensive. Therefore this technique is modified to consider only a subset of N Service Agents whose historical information does not reflect a poor QoS. Consequently the modified version of MTK reduces the communication cost for large number of candidate Service Agents.

2.7 Bilateral Technique (BIL)

The third technique tries to solve some drawbacks of last one by means of improving the quality of information a Counselor Agent has. The proactive communication among Counselor Agents and Service Agents allows have fresh information to take best decisions. The proactive communication behaves next way: more a client request a specific service, there exist more strong relations among the corresponding Counselor and Service agents. Also memory and learning skills are part of the mechanism necessitated for in our proactive communication. The basic algorithm is composed of two parts one of Counselor Agent the second of Service Agent both described next:

On the reception of a service request from users, if the Counselor Agent has no actual information about the QoS of Service Agents, it sends a request-for-bidding to select the Service Agent that has the best QoS from responses, the Counselor agent learns about the QoS of responsive Service Agents. Otherwise, the Counselor Agent uses its knowledge and allocates the request on the Service agent with the best QoS.

The Service Agents behave as follows: When a Counselor Agent receives a request-for-bidding; it makes its offer and stores the sender identity. From that moment it will send QoS offers to register Counselor Agents according to three criteria: the date of the last request for bid, the date of its last request allocation, and the number of requests that the Counselor Agent has awarded to it. In this way is implemented the proactive communication, that is, a Service Agent will offer its service to frequent clients, while the others will be less often addressed. This proactive communication, ensure that Service Agents provide Counselor Agents with accurate knowledge of the QoS of services it frequently re-request, and this, without perturbing the whole system with useless offers.

3. SYSTEM AND WORKLOAD MODEL

In this section we present the model of workload we use to make experiments with the strategies described before. To analyze the performance of dynamic request placement techniques over wired networks, we work at the logical IP-network level. The model then consists of realistic IP hosts and routers, abstracted LANs, and wide-area links, but ignores the details of link-level transmission beyond gross characterizations of the bandwidth and transmission delays on long-distance. The simulation model for the three Dynamic Request Placement techniques introduced in section 2 has been implemented through an extension we have made to the SIDE [13] package, designed for describe network configurations and specify distributed programs organized into event-driven threads.

3.1 Web traffic model

Although no solid, abstract description of Internet traffic exists. We use the best, of the salient characteristics of such traffic. The traffic that drives our experiments is based on the "Behavioral model" of web browsing developed by H.K. Choi [14]. Choi's model is an application-level description of the critical elements that characterize how HTTP protocols are used. A motivation for choosing Web-like traffic to drive these experiments was the assumption that properly generated traffic would exhibit highly variable and bursty demands on Web services.

3.2 System Model

A model in SIDE [16] consists of a number of source files. The basic unit of execution is called a process and it looks like a specification of a finite

state machine. A process always runs in the context of some station, which conceptually represents a logical component of the controlled/ modeled system. In our simulation each Client is modeled as a process at some station owning a copy of the presented Web Traffic Model to generate Web requests, such requests are forwarded by an interface to the Counselor Agent process and waits for the Web-request to be retrieved. Counselor Agents process implements one of the three techniques to decide where to send the received requests, allocated requests are sent to Service Agents through the network interface. The network connections have different structures to represent available network bandwidths present in the Internet links and also have different distribution of transmission delays, finally the network topology is defined by a connectivity function. Service Agent process run in the context of Web server stations and receive requests that accept or reject them depending on service load and in the case of Bilateral implementation it sends its QoS offers to clients accordingly to the mechanism presented in section 2. Likewise the process model of Web servers is an abstraction of actions that occur at the session level layer of the HTTP protocol. We model a Web server as a set of resource queues. Service times are based upon two types of re-requested objects: static and dynamic objects. Service time for static objects is proportional to its size, and dynamic objects require extra time drawn from a Hyper Geometric Distribution representing objects that requires to be processed at Web server before the response is sent to the client.

4. EXPERIMENTAL SETUP

In this section we describe the scenario within which experiments of the three strategies have been carried out. We use SIDE as a platform to simulate the described strategies. Every simulation experiment for a given configuration has consisted of 1200 measurement sessions. Connection failures are set to occur in 0.5% to 0.7% of the measurements, and are independent of both client and time-of-day. Read failure rates are typically even smaller, ranging from 0.05% to 0.2%. The values for the Web traffic model that drives client Web-request generation are shown in Table 1.

TABLE I
WEB TRAFFIC MODEL PARAMETER STATISTICS

<i>Parameter</i>	Mean	S.D.	Distribution
Request Size	360.4	106.5	Lognormal
Obj. Main	10710	25032	Lognormal
Size In-line	7758	126168	Lognormal
Parsing time	0.13	0.187	Gamma
In-line objects	5.55	1.14	Gamma

clients. Five services were chosen because it is a large enough number to make the task of correct service selection hard enough for the Agents to learn service values within the time frame of the experiment.

TABLE II
PARAMETERS CONSIDERED IN THE WEB-SERVICE MODEL

<i>Parameter</i>	Type	Mean	Frequency
Service	High	0.7	0.01
	Med	0.1	0.14
Time	Low	0.0	0.85
		.001	

5. RESULTS

In order to evaluate the advantages of proactive communication, we compare the bilateral technique that uses proactive communication against the MKT and Random strategies. Obviously the MKT technique use memory about QoS of Services so in some way it is mix strategy. We consider four metrics to evaluate the effectiveness for each technique we measure:

- The median of user response times experienced by users to complete a Web-request for 500K size (large Web-requests).
- The CDF (cumulative probability function) of user response times.
- The overhead of each technique given by its communication cost derived from QoS information collection and dissemination and technique negotiation mechanisms.

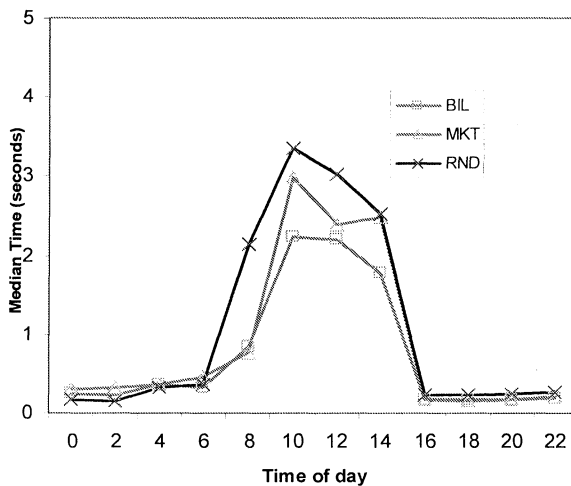


Fig. 4. Time of Day Effects.

We found very positive the inclusion of proactive communication in this problem of load sharing. In Fig. 4, we report the responsiveness of the three techniques, for large 500K Web-requests, we see that BIL outperform MKT and RND for bigger Web-request, due to the feedback mechanism of BIL technique and the amortization request-bidding over baggers requested Web pages. Differences between request placement techniques are far pronounced during busy daytime hours than for quieter periods (see Fig. 4), reflecting the network load on the techniques, and the periodicity of hourly effects. MKT technique always perform worse than BIL and better than RND, with minor exceptions curves do not cross, implying that as QoS perceived changes clients need not to use a different technique to achieve optimal performance.

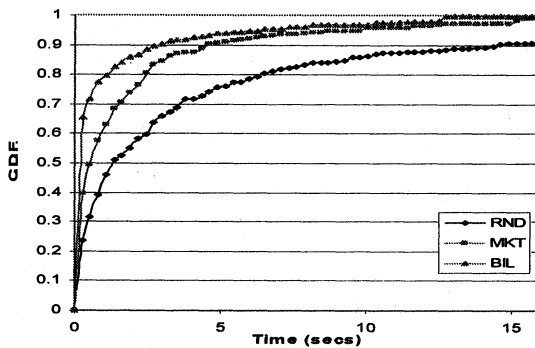


Fig. 5. Median Response Time Cumulative Probability Function.

We compare in Fig. 5, the performance of the three techniques using the total response time of allocated requests and the overhead induced by negotiation techniques implemented by such techniques in Fig. 6; we can see that selecting with RND technique has the lowest performance although it does not produce any overhead as compared with MKT whose performance is higher because its QoS actualization mechanism, but it incurs in high overhead too, then the most cost-effective strategy is BIL whose performance is high and overhead remains relative low in comparison with RND and MKT.

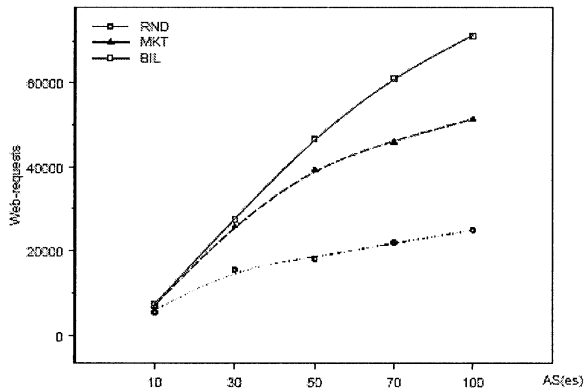


Fig. 6. Request Placement Technique overhead.

The result of the comparison in a total qualitative agreement goes in the sense that the relationship among the improvement of QoS perceived by users of the contrasted curves is preserved. In the quantitative side, the figures presented in this paper show slower response times that those conducted before, the reason being the workload model described in [16] on which experiments were conducted are based on a Non-Homogenous-Poisson Process, showing low variability and uniform response times.

6. CONCLUSION AND FUTURE WORK

The experimental results demonstrate the positive role of proactive communication in the load sharing problem. Positive obtained results are due to proactive communication allows keeping updated the information that agents keep about most common counterparts (services).

Also we think that random selection technique would be sufficient when the primary goal is load balancing and when the clients do not have QoS constraints. On the other hand the Bilateral dynamic technique we have developed, would be useful in an environment in which QoS-aware clients that have different requirements accessing servers that display significant variability in their response times. Although our proactive based technique was mainly developed to improve QoS perceived by Web service's users, we are sure that results obtained for proactive communication can be generalized to different problems where negotiation is an important aspect in the solution of the problem.

REFERENCES

- [1] J. Guyton and M. Schwartz. Locating nearby copies of replicated internet servers. In Proceeding of ACM SIGCOMM' 95, 1995.
- [2] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee, Application-layer anycasting: a server selection architecture and use in a replicated web service, IEEE/ACM Transactions on Net-working, vol. 8, no. 4, pp. 455–466, Aug. 2000.
- [3] M. Crovella and R. Carter. Dynamic server selection in the internet. In Proceeding of IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems, 1995.
- [4] Crovella and R. Carter, Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks, In Proceedings of IEEE INFOCOM, 1997.
- [5] M. Rabinovich, I. Rabinovich, R. Rajaraman, and A. Aggarwal. A Dynamic Object Replication and Migration Protocol for an Internet Hosting Service. In Proc. 19th Int'l Conf. on Dis-tributed Computing Systems, pp. 101–113, Austin, TX, June 1999. 18 IEEE.
- [6] S. Krishnamurthy, W. H. Sanders, M. Curier, Performance Evaluation of a QoS-Aware Framework for Providing Tunable Consistency and Timeliness, Proc. of The 10th IEEE International Workshop on Quality of Service (IWQoS 2002), May 2002.
- [7] S. Krishnamurthy, W. H. Sanders, M. Curier, Performance Evaluation of a Probabilistic Replica Selection Algorithm, Proc. of The Seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002), Jan. 2002.
- [8] F. F. Ramos, "Placement dynamique de requêtes multiagents dans les systèmes d'information globaux", PhD thesis, Université de Technologie de Compiègne, juin 1997.
- [9] M. Bui, F. Butelle, C. Lavault, (2003). A Distributed Algorithm for Constructing a Minimum Diameter Spanning Tree. Journal of Parallel and Distributed Computing.
- [10] S.G. Dykes, K. A. Robbins, C.L. Jeffery, "An Empirical Evaluation of Client-Side Server Selection Algorithms", in IEEE INFOCOM 3:1361-1370, 2000.
- [11] M. H. Balter, M.E. Crovella, C.D. Murta, "On Choosing a Task Assignment Policy for a Distributed Server System", in Parallel and Distributed Computing 59:204-228, 1999.
- [12] G. R. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver", IEEE Trans. on Computer, 12(29), Dec. 1981, pp.1104-1113
- [13] P. G. Burzynsky, Protocol Design for Local and Metropolitan Area Networks, Prentice Hall, 1996
- [14] H. K. Choi, J. O. Limb, "A Behavioral Model of Web Traffic", Proceedings of the Seventh Annual International Conference on Network Protocols, 1999.
- [15] K. Calvert, M. Doar, and E. W. Zegura, "Modeling internet topology, " IEEE Communications Magazine, June 1997. [10] C. Jin, Q.Chen, S. Jamin, "Inet: Internet Topology Genera-tor," University of Michigan Technical Report, CSE-TR-433-00, September 2000.
- [16] Felix F. Ramos, Liming Chen and Marc Bui. A Comparison of Two Dynamic Request Placement Strategies in Large Distributed Information Systems. In International Workshop on Database and Expert Systems Applications of DEXA 1999 IEEE Computers Society. Florida, on September 1999.
- [17] ISO, "Quality of Service Framework", ISO/IEC JTC1/SC21/WG1 N9680, International Standards Organization, UK, 1995.