

Timeliness Multi-Agent Coordination Technology in Autonomous Decentralized Database Systems

Carlos PEREZ-LEGUIZAMO^{†a)} and Kinji MORI^{††b)}, Members

SUMMARY The turn of the century is witnessing radical changes in the way information services are spreading due to the progress of IT and the constantly increase in the number of users of the WWW. Therefore, the business market is changing its strategy for a modern online business environment. Autonomous Decentralized Database System (ADDS), based on autonomous coordinating subsystems, has been proposed as a system architecture in order to meet the innovative e-business requirements for consistency and high response among distributed database systems. Autonomy and decentralization of subsystems help achieving high response time in highly competitive situation and autonomous Mobile Agent based coordination has been proposed to achieve flexibility in a highly dynamic environment. In this paper, it is analyzed the case in which the system size increases; and a multi agent coordination, the same number of mobile agents and sites coexist in the system, is proposed for achieving the timeliness property. The response time in the system is conformed by those transactions that require coordination and those that can be satisfied immediately. In accordance, the distribution of the data in the system for coordination is a medullar issue for the improvement of the response time. A trade-off exists between these two kind of transactions depending on the coordination of the Mobile Agents, the capacity of allocating data among the sites, and as well as the distribution of the data and user requests in the system. In this sense, since the system requires high response time, a data allocation technology in which each mobile agent autonomously determine its own capacity for adjusting data among the sites is proposed. Thus, the system will adapt itself to the dynamic environment. The effectiveness of the proposed architecture and technologies are evaluated by simulation.

key words: *autonomy, decentralized, database systems, multi agents, coordination, timeliness*

1. Introduction

In the recent years, the society has suffered several changes in its ways and methods of consuming. Nowadays, the diversity and the customization of products and services have provoked that the consumer needs be so diverse and frequently in change. As a result, the cooperation among multiple companies and the formation of strategic alliances, such as the Supply Chain Management (SCM), for providing better and quicker services to the consumers have increased [1].

A new system architecture, Autonomous Decentralized Database System (ADDS), has been proposed in order to satisfy the enhanced requirements of current online e-business applications. This system architecture is in-

spired by Autonomous Decentralized Systems [2], where a number of autonomous subsystems are loosely connected through a Data Field (DF). The subsystems are completely autonomous and coordinate among each other through the data field using a content code communication protocol that is address free communication method and depends only on the contents of the data. This system has been shown to have properties of real time, fault-tolerance and online expansion in high-assurance systems [3]. Similarly, the Autonomous Decentralized Database System [4] is based on the concept that each site in the integrated system is completely autonomous. The attribute Allowable Volume (AV) [5] has been defined to provide decentralization and thus to achieve real time updates on each site. A Mobile Agent (MA), an autonomous entity, is devised for coordination by the continuous adjustment of AV among the sites [6].

A problem in this system is as the number of sites increases, some user requests cannot be satisfied quickly since they need for the coordination to complete their requests. As a result, the timeliness of the system cannot be achieved. To solve this problem, a multiple-agent-based coordination, the same number of Mobile Agents and sites coexist in the system, is proposed. However, the response time in the system is conformed by those transactions that require coordination and those that can be satisfied immediately. Moreover, the distribution of the data in the system for coordination is a medullar issue for the improvement of the response time. In order to have an optimal response time, a study of the factors that need to be considered for an allocation data technology in order to get an optimal response time in a highly dynamic environment is discussed. In this sense, since the system requires high response time, a data allocation technology in which each Mobile Agent autonomously determine its own capacity for adjusting data among the sites is proposed. Thus, the system will adapt itself to the dynamic environment.

2. Requirements

2.1 Application Requirements

In business environments, many application systems are expanded side by side, maintained and installed, or multiple systems with different requirement levels are integrated into one system according to the social and economical situations of each company.

The customers demand services should be available at

Manuscript received April 22, 2010.

Manuscript revised August 25, 2010.

[†]The author is with the Technological Research, Servers and Security Group at Banco de Mexico, Mexico City, 06059, Mexico.

^{††}The author is with the Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

a) E-mail: cperez@banxico.org.mx

b) E-mail: mori@cs.titech.ac.jp

DOI: 10.1587/transinf.E94.D.27

any time and the response they get from the system should be rapid. Those companies, who cannot cope with such characteristics in their provision of services, have to face the loss of opportunity since the customers can change easily the provider with one click if their demands are not well satisfied.

On the other hand, from the point of view of the companies, it is required that the system should be continuous available since in a competitive marketplace, the time plays an important role for getting profitable benefits. Thus, those companies who require reducing their stocks because of the storing cost, will try to do it sooner since a delay in the adjustment of the stock could represent a waste of money. The longer time they take to suit the products the more money they have to pay.

2.2 System Requirements

The system needs for the integration of DB systems which can cope with the application needs mentioned above are summarized in Table 1. For a distributed database system in an environment like the Internet, in order to satisfy the heterogeneous needs, such as the reduction of the opportunity-loss and the stock-loss cost; and to realize the adaptability of the system, timeliness, fault tolerance and flexibility are needed.

2.3 Related Work

Some architectures and technologies have been proposed in order to integrate heterogeneous databases [8], [9], however, since these ones are based on tight-coupled and strongly consistency management, they present some problems such as low response time, low flexibility and null consideration of heterogeneous needs.

On the contrary, [10],[11] have developed non-blocking protocols for efficient data consistency management. They exploited the notion of tokens to enable high-volume transaction processing for distributed resource allocation applications. The effectiveness of token partitioning, however, relies on token redistribution techniques that allow dynamic migration of tokens to the servers where they are needed. In the overall, previous related works have the disadvantage to be centralized, because they assume that some sub-entities are able to grasp the total system, or because there exist master-slave relationships between the sub-entities. In rapidly changing environments, it is alleged that adaptability to changes in the system can only be obtained by assuring the autonomy of the sub-entities constituting the

system.

3. Autonomous Decentralized Database Systems

3.1 Concept and Platform

Considering the application and system requirements in a dynamic and unpredictable environment Autonomous Decentralized Database System (ADDS) has been proposed in order to satisfy the requirements and to cope with highly dynamic environment [4]. In ADDS the operational database systems of autonomous business sites with heterogeneous needs can be integrated into a common environment without violating each others characteristics. The ADDS system architecture is shown in Fig. 1.

The system has the following characteristics:

- The system is composed of autonomous sites that can update locally its database.
- Each site can coordinate with other sites through autonomous mobile agents using content code communication protocol which addresses free communication that depends only on the contents of the data.
- The sites are connected each other forming a logical data field.

3.2 Loosely-Consistency Management

The architecture of this system is based on the autonomy that each site has for updating without any communication with some other sites while keeping the consistency of the whole system. In ADDS, each site holds Allowable Volume (AV) which defines the maximum permissible volume with in which each site can have local updates autonomously. Considering a typical example of online Supply Chain System, as shown in Fig. 2, where autonomous sites of makers and retailers are integrated. In Fig. 2, Product A ,which exists 100 in the whole system, is distributed respectively 50, 20, 30 in the Site0, Site1 and Site2. if there is some request of a product (defined by AV) on one site and this requested amount is less than the AV on this site (like Site2), the sites can immediately update this request without communicating with any other site and the consistency of the distributed database is also preserved [5]. Hence the real time in updates is achieved. On the other hand, if the requested amount

Table 1 Application and system requirements.

Application needs	System needs
Heterogeneous needs	Heterogeneity,
Dynamic environment and preferences	Flexibility
One-click response	Timeliness
Continuous service	Fault Tolerance

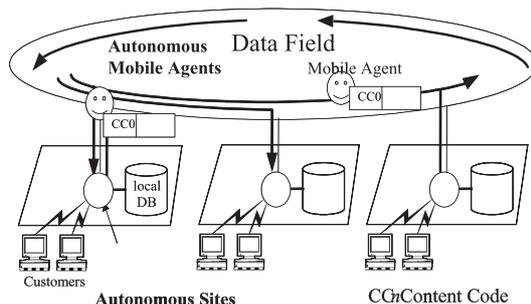


Fig. 1 ADDS architecture.

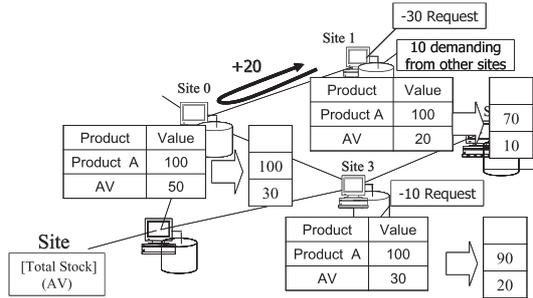


Fig. 2 Allowable volume.

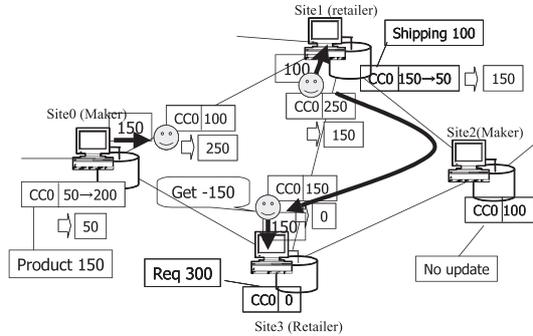


Fig. 3 Background coordination technology.

is more than the local AV available at his site (like Site1), the site is required to coordinate and arrange the lacking amount of AV from other sites (like Site0).

3.3 Background Coordination

The background coordination deals with the adjustment of AV among the sites [6]. As previously, we will take the on-line Supply Chain System as an application of our research throughout the paper, where autonomous sites of makers and retailers are integrated to form an ADDS. The consumers' needs at each site change dynamically and independently, and therefore the amount of AV at each site is required to be optimized. In this process some sites may run into frequent shortage of volume and some other sites might have surplus volume. Thus, an adjustment of AV among the sites is required. In ADDS, the mobile agent (MA) that is an autonomous entity carrying AV, is in charge of continuously adjusting the AV among the sites, see Fig. 3. For example, Site0 (Maker) produces goods 150, MA which is in Site0 receives 150 and MA moves to Site1. Next, since Site1 had the 100 request before MA moves to Site1, MA gives 100 to Site1 and MA moves to Site2. Next, since Site2 didn't have update before MA moves to Site2, MA passes without negotiating with Site1 and MA moves to Site3. Next, since Site3 had the 300 request before MA moves to Site3, MA gives 150 to Site3. Thus, while MA visits all sites, MA negotiates with each site.

3.3.1 Autonomous MA Circulation Mechanism

In ADDS, the MA moves among all the sites in order to maintain the consistency of the system, thus forming a logical dual-ring network. Each site in the system is furnished with a Site Information Table (SIT) that records the information of the neighboring sites directly connected to it (previous and next). The details of this technology can be found in [4].

3.3.2 Autonomous Fault Detection and Recovery

The decentralized structure of the ADDS offers high reliability and in case of failure of any site in the system the rest of the system keeps on its normal operation without stopping. ADDS is an architecture that can cope with a certain level of faults by itself since it consists of autonomous subsystems, i.e., sites and a mobile agent. However, in case of the MA's failure by a communication fault or software bug, a recovery process is initiated in order to recover not only the MA but also the AV that the MA carried in the moment of the failure. The details of this technology can be found in [7].

4. System Size and the Response Time

As a large number of business combines are getting online and are also using this infrastructure for mutual coordination, as a result while the users demand a high response time due to extensive competition on one hand, the system is getting of gigantic size on the other hand and thus affecting the response time.

While ADDS provides a robust solution to the changes in the dynamic environment and the infrastructure, the response time poses some restrictions on the size of the system.

This paper examines the critical relation between the user's response time and the number of sites in a typical integrated system and proposes a dynamic solution for the coordination of system.

If Ru represents the user demand at any particular instant on a particular site in the system, AV_i is the amount of product volume on this site and AV_{all} is the total AV in the system then on the arrival of a user request on site i there exist two cases:

- i. $AV_i \geq Ru$

In this case site i has enough AV to satisfy the user's demand immediately. Therefore the response time to the user T_w is just the local processing time T_{CAV} on this site, i.e.,

$$T_{w1} = T_{CAV}$$

where T_{w1} represents the response time of those transaction that do not require coordination.

ii. $AV_i < Ru$

When the site does not have the enough AV to satisfy the user's demand the site collects the lacking amount of AV from the mobile agent carrying extra AV from other sites in the system. If T_{MA} is the time to complete the MA one cycle among the sites, the average time for the MA to reach the demanding site can safely be stated as $T_{MA}/2$. If the probability that MA is already carrying the required amount of lacking MA when it arrives the demanding site is p_1 then the user's response time in this case T_{w_2} can be written as follows

$$\begin{aligned}
T_{w_2} &= T_{CAV} + \\
& p_1 \cdot \frac{T_{MA}}{2} + \\
& (1 - p_1) \cdot p_1 \cdot \left(\frac{T_{MA}}{2} + T_{MA} \right) + \\
& (1 - p_1)^2 \cdot p_1 \cdot \left(\frac{T_{MA}}{2} + 2T_{MA} \right) + \\
& (1 - p_1)^3 \cdot p_1 \cdot \left(\frac{T_{MA}}{2} + 3T_{MA} \right) + \\
& \vdots \\
& = T_{CAV} + \sum_{k=0}^{\infty} (1 - p_1)^k \cdot p_1 \cdot \left(\frac{1}{2} + k \right) \cdot T_{MA}
\end{aligned}$$

where T_{w_2} represents the response time of those transaction that require coordination.

Moreover, if the probability that $AV_i \geq Ru$ is represented as p_2 then the user's average response time T_w is given by

$$\begin{aligned}
T_w &= p_2 \cdot T_{w_1} + (1 - p_2) \cdot T_{w_2} \\
&= T_{CAV} + \\
& (1 - p_2) \sum_{k=0}^{\infty} (1 - p_1)^k \cdot p_1 \cdot \left(\frac{1}{2} + k \right) \cdot T_{MA} \quad (1)
\end{aligned}$$

As clear from the above relation, the user's average response time depends on the mobile agent's circulation time T_{MA} that in turn depends on the total number of sites in the integrated system. Therefore, the scale of the system highly affects the response time of the total system.

As mentioned in Sect.3.3, the coordination mobile agent (MA) is a program with data, whose size is usually hundreds of kilobytes. In the Internet, although the bandwidth is increasing, it still takes several hundreds of milliseconds to transfer data from one site to another. Moreover, since the agent does some process according to the current status of the sites, it also takes time. Thus, in an autonomous decentralized database system, as the number of sites increasing, it takes the MA long time to circulate around all the sites.

5. Autonomous Multi-Agent Data Allocation Technology

In the proposed technology, the adjustment of the AV is carried on not only by a single but for multiple and autonomous

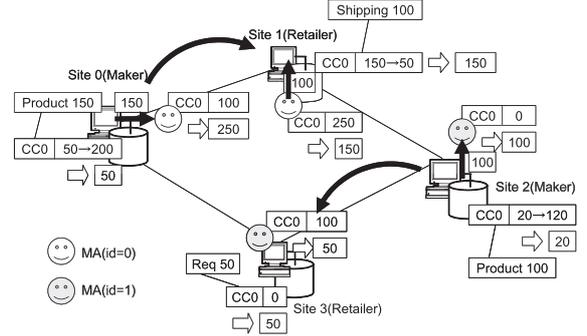


Fig. 4 Multi-agent coordination.

mobile agents. Each mobile agent in the system has the functionality of coordinating AV with the sites by allocation and negotiation. When multiple mobile agents coexist in the system, the time that a site has to wait in order to negotiate AV with a MA reduces and consequently the probability to satisfies the site needs, demand or supply of AV, increases. For example, as shown in Fig. 4, in the system, exist two mobile agents which are doing the coordination simultaneously among the 4 sites according to the basic strategy explained in Sect. 3.3. In this sense, the response time may improve since the sites should not wait only for one mobile agent for the coordination.

For this proposal, the number of mobile agents that coexist in the system is equal to the number of sites. We consider this assumption since even the number of mobile agents may have some influence in the response time, determining the optimal number of them is a problem that requires to know the whole information of the system which is not suitable for a very dynamic and unpredictable environments. Moreover, if in the system exists a large number of mobile agents, congestion in the sites occurs and the response time increases. Thus, the site is responsible for the management of its own mobile agent. When it joins to the system, the site will generate a mobile agent with the same ID that is unique for the whole system and, will let it to coordinate AV with the other sites. On the other hand, when the site leaves the system, it is responsible for killing its own mobile agent.

The increment in the number of mobile agents in the coordination process have some collateral effects in terms of response time when considering the type of transaction. In the system, two different kind of transactions are identified, those that do not require for any coordination since they can be satisfied immediately, there is enough AV in the system to satisfy the request, and those which necessarily needs of the coordination process, the site has to wait for a mobile agent. In this sense, the distribution of the AV in the system for coordination and the way that it is done, the amount of AV allocated and negotiated among the sites and the mobile agents, is an important factor to considered and control in order to be close to an optimal response time.

5.1 Analysis of the Response Time and the AV and Users Requests' Distribution for Multiple Mobile Agents

For the analysis of the response time and the AV distribution, formula 1 has to be rewritten as following. When considering a fixed system size, the number of times of negotiations that the MAs need in order to complete the user request N_{MA} can be written as

$$N_{MA} = (1 - p_2) \sum_{k=0}^{\infty} (1 - p_1)^k \cdot p_1 \cdot \left(\frac{1}{2} + k\right) \quad (2)$$

From this relation, we see that p_2 , the probability of getting enough volume for user's demand immediately, depends on the total allowable volume AV_{all} in the system and increases with increase in AV_{all} . p_1 , the probability of the MAs having enough volume to satisfy a certain site's demand at a particular instant, however is much more complex in nature and also depends on the number of sites with heterogeneous needs, the location of the sites and the number of mobile agents besides the AV_{all} .

When in the system there exists a high distribution of AV and users' requests uniformly distributed among all the sites, the majority of the transactions can be satisfied immediately without few coordination and consequently p_2 increases and p_1 decreases. Since we are considering that in the system coexist several MAs, the total p_2 and p_1 are determined by the amount of AV that each MA carries during the coordination.

On the other hand, when the AV and user's requests allocation are completely separated, there would exist several transactions that need of the coordination in order to be satisfied but at the same time there will be several sites which would like to release AV. In this sense p_2 may decrease basically because of the position of the sites in the logical ring, while p_1 increases since eventually all the AV released from some sites will be allocated. A similar analysis than the previous paragraph should also be done for multiple MAs.

Summarizing, we have N_{MA} that is a decreasing function of both p_1 and p_2 whose behavior depends on two factors, the AV and user requests' distribution and the AV that the MA's carries during the coordination. Under this observation and since our objective is to optimize the user's response time and we cannot change the AV and user requests' distribution, we proposed to limit the capacity of the AV that the MAs can carry during the coordination.

5.2 Relation Between the Response Time and the Capacity of AV in the MAs

Figure 5 shows the trade-off that exists between the response time when the system is highly uniform distributed (skew 0), the AV and users' request are indistinctly located among all the sites, and concentrated (skew 1), the AV is in half of the sites and the users' requests are located in the other half, when the MAs' capacity changes. For this simulation the total AV in the system is constant and as soon as it is consumed

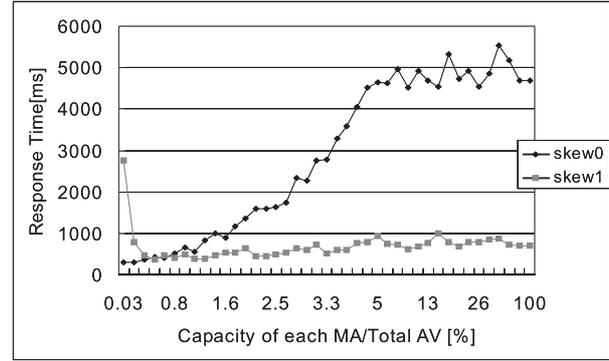


Fig. 5 MAs' capacity vs response time.

by the users, the system regenerates the same quantity and locate it randomly among the sites.

Thus, if the MAs' carries just a few amount of AV during the coordination process, the response time of the system is high when highly uniform distributed but low when concentrated. As discussed in the previous section, this is mainly due to the fact of the distribution of AV and users' requests and the balance in the AV that each MA has during the coordination. Thus, in a highly uniform distributed situation, it is convenient that the MAs' carry a small amount of AV since most of the AV_{all} is located in the sites and these ones are receiving and satisfying users' requests. While in concentrated environment, most of the sites that demands AV has to wait for the coordination. Thus, if the MA can carry few AV these sites have to wait for longer time.

On the other hand, when the MA carries a large amount of AV, the response time has a considerable improvement in a concentrated environment since the demanding sites can access to the free AV quicker. However, in case of a highly uniform distributed environment, at the beginning most of the sites release their initial AV concentrating it mainly in few MAs. This unbalance in the AV carrying by the MAs produce the sites that already have released its AV cannot response quick to the new users' requests having to wait for the next MA that has AV to negotiate.

Under this changing and unpredictable environment, the capacity of the MA for carrying AV plays an important role for the assurance of the system. The MAs has to consider the factors already explained in order to optimize the response time when the distribution of AV and users' requests changes. Another important factor that we can deduce from this graph is the role that the total AV in the system plays. Thus, for different initial conditions of this, the crossing point of the trade-off also changes and consequently the system should adapt to this kind of ever evolving situations.

5.3 The Principles of the Proposal

The basic principle in which is held this proposal is that the autonomy of the system should be kept. In an ADDS, neither the sites nor the MAs can grasp the total information of

the system. In this sense, we propose that the MAs determine its capacity for carrying AV and adapt itself to the current situation of the system by an estimation of some values that it can collect during the adjustment of AV among the sites. Since the environment changes dynamically at any time, the real values cannot be detected but the MAs can follow the tendency of the system.

(1) Function of Heterogeneity

In ADDS different sites with different needs have to coexist without violating each other. Moreover, it is necessary to consider also that the environment is always changing. As seen in the previous section, the needs of the users or the sites will be reflected in the status of the sites. Thus, we consider that in the system exists two kind of sites: those which are requesting AV and those which are providing AV. Based on this relation, we define the Function of Heterogeneity in the system in terms of the dispersion of the AV among the sites and the relation of requesting and providing AV sites. In this context, the AV becomes the Data and the dispersion becomes the Heterogeneity.

The MAs autonomously evaluate the function of heterogeneity that determines the normalized standard deviation *hetero* that exists between the AV requesting and providing sites. This function is defined by considering V_k as the difference between its demand value (D_k) and the Allowable Volume (AV_k) at site k and the total number of sites α . The function of heterogeneity is defined as follows:

$$hetero = \frac{1}{\alpha} \cdot \sum_{k=1}^{\alpha} \frac{(V_k - V_{k-1})^2}{V_k^2 + V_{k-1}^2} \quad (3)$$

The function *hetero* gives a value within a range from 0 to 1 that corresponds to all the requested patterns that may exist in the system during the time that the MA does the evaluation.

5.4 Heterogeneity Data Allocation Technique

During the continuous adjustment of AV among the sites, the MAs will calculate the necessary values, referred in 5.3, in order to determine its maximum capacity for carrying AV (M). The MAs will take any site in the system as a host-site by keeping the address of its original site and in one round the calculation of its new capacity will be done. We make this assumption since the more frequently a MA can determine its capacity, the better coordination in terms of timing and accuracy. The MAs will adapt itself to the changing situations in the system by modifying its capacity in the host-site taking in consideration how the distribution of the data and users requests changes. The change in the capacity of each MA is determined by considering the *hetero* value and a roughly estimation of the AV in the sites. Thus, the MA continuously calculate its new capacity as follows:

$$M = hetero \times \frac{SiteSurpAV + MASurpAV}{NumberSites}$$

Where *SiteSurpAV* corresponds to the sum of the AV that

the sites that want to provide to the system AV. When the MA arrive into a site, it checks the $Request_{AV}$ value, if this value is negative then it means the site wants to provide AV and consider it for the calculation.

$$SiteSurpAV = \sum_{k=1, Request_{AV_k} < 0}^{NumberSites} -Request_{AV_k}$$

On the other hand, the *MASurpAV* is the current AV in all the MA's. Since this value is not possible to know it, we utilize the *OldAVCapacity* as an estimation value.

$$MASurpAV = OldAVCapacity \times NumberSites$$

6. Evaluation Technique

We evaluate our proposal with the purpose of measuring the improvement in the response time of the proposed technology compared to the conventional solutions, one MA coordination and multiple agent coordination (without Data allocation). We propose to evaluate this technology in terms of the average response time of the whole system during the adjustment of AV as well as the total satisfaction ratio.

The system model used for this simulation was composed by some retailers sites coexisting in a common environment with the purpose of cooperating among them. The sites manage the same kind of products but the characteristics, requirements and needs of each of them are different according to its own context. The sites will receive the requests from the users and if possible try to satisfy locally. Otherwise, it will queue the requests and wait for AV when the MA comes to negotiate. If the request cannot be satisfied during a timeout, then it is cancelled.

This simulation was done under the assumption that the popularity of some sites changes. Thus, the *userDistributionInterval* determines the time in which the system changes the users requests pattern from *uniform* to *skew* and viceversa. For modelling this behavior, a *Zipf* distribution, that suggest the Web use follows, was utilized. The rest of the parameters utilized for the simulation are shown in Table 2. In summary, these parameters were established in order to emulate a highly dynamic environment in which several requesting AV and providing AV sites coexist in the Internet, and there exists the need to allocate the data among them according to each site requirements.

Figure 6 shows the comparison of ADDS in its normal process by a single agent, multi-agent and multi-agent applying the data allocation technology. As it can be seen, the average response time for the system improves when the MAs determine dynamically their capacity according to the system environment. In this sense, the system can achieve timeliness.

Figure 7 shows the improvement in terms of the satisfaction ratio, that is defined as the ratio between the total number of satisfied requests within a certain timeout and the total number of requests, when applying the proposed technology. When the total number of sites increases, the satisfaction ratio of the proposal degrades much slower than the

Table 2 Parameters for simulation environment.

Parameter	Value
NumNodes	2 - 50
BandWith	500 - 2000 [Kbps]
AvDemandUser	1 - 5
UserTimeout	3000 [ms]
RequestArriveRate	100 [ms]
UserDistributionChangeInterval	50000 [ms]
NumMakers	1
ProduceAV	1500
ProduceAVPeriod	50000 [ms]
DbAccessTime	100 [ms]
DemandChangeInterval	5000 [ms]
MaSize	20 [KB]
MaProcessTime	100 [ms]
TotalAV	3000
SimulationTime	200000 [ms]
Epsilon	0.3

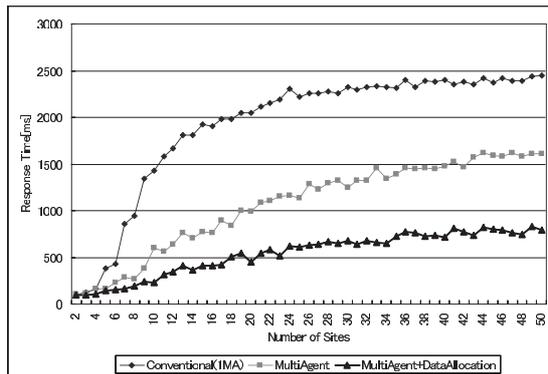


Fig. 6 Response time vs number of sites.

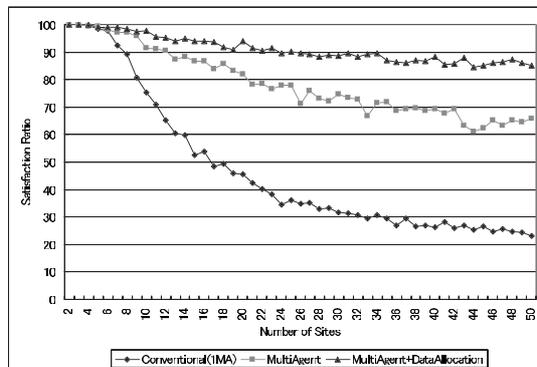


Fig. 7 Satisfaction ratio vs number of sites.

conventional ones. That is because the proposed technology can adapt the system dynamically and the AV allocation can be done in more efficient way.

7. Conclusion

Nowadays, the Internet has grown up to a common business environment for companies where they can provide services to large number of uncertain customers. The preferences of the customers on the Internet are always changing dy-

namically and drastically. The customers require the services to be quick otherwise they can change to another company by only one-click. Autonomous decentralized database system is proposed as an architecture for achieving these needs. In this system, the database is distributed in a number of sites. The sites define those numeric data with the property called allowable volume (AV), with in which they can update their local databases autonomously, so timeliness is achieved. A background coordination mechanism, performed by an autonomous mobile agent, is devised to adapt the system to evolving situation. By adjusting the allowable volume among the sites in advance, the users can be satisfied quickly even their preferences changes.

In case the system size increases, a multi agent coordination, the same number of mobile agents and sites coexist in the system, is proposed for achieving timeliness. The response time in the system is conformed by those transactions than require coordination and those than can be satisfied immediately. Thus, the distribution of the data in the system for coordination is a medullar issue for the improvement of the response time. A trade-off exits between these two kind of transactions depending on the coordination of the mobile agents, the capacity of allocating data among the sites and as well as the distribution of the data and user requests in the system. In this sense, a data allocation technology in which each mobile agent autonomously determine its own capacity for adjusting data among the sites was proposed. Thus, the system will adapt itself to the dynamic environment. The effectiveness of the proposed architecture and technologies were evaluated by simulation.

References

- [1] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Managing Supply Chain*, McGraw Hill, 2003.
- [2] K. Mori, "A proposal of the autonomous decentralized system concept," *IEICE Trans. Electron.*, vol.104, no.12, pp.303-310, Dec. 1984.
- [3] I.L. Yen, R. Paul, and K. Mori, "Toward integrated methods for high-assurance systems," *Computer*, vol.31, no.4, pp.32-34, April 1998.
- [4] C. Perez-Leguizamo, S. Kato, and K. Mori, "Autonomous decentralized database system for assurance in heterogeneous e-business," *Proc. COMPSAC'01*, pp.589-595, Oct. 2001.
- [5] C. Perez-Leguizamo, W. Dake, and K. Mori, "Loosely-consistency management technology in distributed DB systems for assurance," *IEICE Trans. Inf. & Syst.*, vol.E86-D, no.10, pp.2104-2113, Oct. 2003.
- [6] C. Perez-Leguizamo, W. Dake, and K. Mori, "Autonomous mobile-agent-based data allocation technology in distributed DB systems for assurance," *IEICE Trans. Commun.*, vol.E87-B, no.7, pp.1818-1825, July 2004.
- [7] C. Perez-Leguizamo, D. Wang, and K. Mori, "Fault tolerance technology for autonomous decentralized database systems," *Proc. SRDS'03*, pp.329-338, Oct. 2003.
- [8] A. Elmagarmid, M. Rusinkiewics, and A. Sheth, *Management of Heterogeneous and Autonomous Database Systems*, Kindle, 1999.
- [9] P.A. Bernstein and E. Newcomer, *Principles of Transaction Processing*, Morgan Kaufmann, 2009.
- [10] U. Cetintemel, B. Ozden, M.J. Franklin, and A. Silberschatz, "Partitioning vs. replication for token-based commodity distribution," *UMIACS-TR-2000-62*, 2000.

- [11] U. Cetintemel, B. Ozden, M.J. Franklin, and A. Silberschatz, "Design and evaluation of redistribution strategies for wide-area commodity distribution," ICDCS'01, April 2001.



Carlos Perez-Leguizamo received the B.S. in Computer Engineering from the National Autonomous University of Mexico in 1997. In 2000, he joined the Tokyo Institute of Technology, Japan, as a master student under the supervision of professor Kinji Mori. He received M.S. and Ph.D. degrees in Computer Science in 2002 and 2005, respectively. In 2005, he joined the Technological Research, Servers and Security Group at Banco de Mexico. In 2007 he became the research leader of this group. He also joined the Panamericana University as a lecturer since 2007.



Kinji Mori received the B.S., M.S and Ph.D. degrees in the Electrical Engineering from Waseda University, Japan in 1969, 1971, and 1974, respectively. From 1974 to 1997 he was with System Development Lab., Hitachi, Ltd. In 1997 he joined Tokyo Institute of Technology, Tokyo, Japan as a professor. He proposed Autonomous Decentralized Systems (ADS) in 1977. He has more than 300 patents in the world. He is one of the founders of the International Symposium on ADS (ISADS).