

Improving Performance in Adaptive Fault Tolerance Structure with investigating the effect of the number of replication

Negar Mosharraf¹, Mohammad Reza Khayyambashi²

Department of Computer, Engineering Faculty, University of Isfahan, Isfahan, Iran^{1,2}

mosharraf@eng.ui.ac.ir¹, m.r.khayyambashi@eng.ui.ac.ir²

Abstract

Regarding the wide use of distributed systems in various areas, having a system with fault tolerance ability would be an import characteristic. And in designing the real time distributed systems, this seems to be more considerable. With regard using some middleware like CORBA in designing such systems, and in order to increase their compatibility, speed, performance, to simplify the network programs and other characteristics there is no supporting program to have distributed real time system and fault tolerance at the same time. In fact, adaptive means taking into account the properties of both structures so that the requirements of these two structures are met during performance and this is usually created by a trade off between specifications of both real time and fault tolerance systems. In this study, the FT-CORBA structure as a structure used for supporting fault tolerance programs as well as relative important parameters including replication style and number of replica, which play further role in improved performance and make it adaptive to real time distributed system have been reviewed. Studying these specifications a structure adaptive to real time systems with higher performance than FT-CORBA structure have been made and, finally, the implementation of the said structure and determination of the number of replica and the objects replication style as well as the significance of related parameters have been investigated.

Keyword: Middleware, Fault Tolerant, Trade-offs, Replication, Real time

Introduction

The Middleware such as CORBA(Common Object Request Broker Architecture) has made possible the supporting of a fault tolerance structure and real time distributed system through providing FT-CORBA and RT-CORBA structures while programs capable of having the specification of real time and fault tolerance at the same time are not supported therein[1]. In fact, being real time and reliability are regarded as property system level, which are not simultaneously combined and adapted on distributed system because they often impose conflicting requirements on the distributed system [1,2,3].

The specification of a real time system is so that it should contain bounded request

processing time, predictable and specified task deadline on system [1, 3, 4]. The behavior of a real time system is based on actual time, frequency of client invocation, the relative priorities of the various invocations, the worst case execution times of invocations at the server, availability and allocation of recourses for the application's execution time must be known ahead of run time. And with this information, the real time infrastructure then computes a schedule ahead of run time [1,3,5]. In fact, predictability is considered as a major specification in real time system, while in a system with reliability and fault tolerance, a program should be recovered and be continued in the presence of unanticipated events such as faults [6]. Therefore, there is a basic difference between the specifications of two systems and

the fault tolerance and being real time and when both the specifications are given for the said system there should be a Trade off generated between the sensitive specifications of both the structures so that the real time system will correctly continue its activity and, at the same time, it will be able to discover and recover faults whenever a fault is occurred [1,3,7].

Many programs requiring fault tolerant have a complex and large number of objects and a fault tolerance domain is defined for managing these structures which allows the said structure to grow in network with desired size as shown in Fig. 1 too. A fault tolerance domain includes one or more host(s) and one or more object group(s) and, in total, a host can be a part of several fault tolerance domains managed by Replication Manager. Certainty on the secured access to objects in a fault tolerance domain and non-interfered access to objects in different domains is achieved by FT-CORBA structures the function of which will be discussed below.

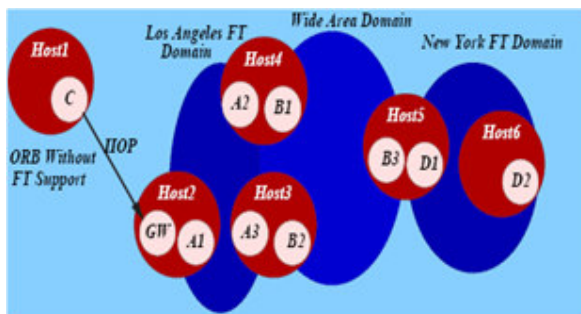


Figure1. Fault tolerance Domain in distributed system

What is often done for the generating of fault tolerance structures is the replication of objects on distributed system so that the object replication will be used when a fault is occurred in current object and the recovery on system would be provided through generating a replication of object and, of course, this is accomplished in such a condition that the

consistency of old and current object is maintained.

Thus, when a fault tolerance structure is designed the determination of replication style and the number of replicas on distributed system are regarded as the main specifications of the structure, especially when the distributed system is real time.

Creating fault tolerance structures through FT-CORBA is fundamentally static, i.e. the specifications of fault tolerance structure cannot be altered during the execution time and the object replication style cannot be changed on execution time. Utilizing a mechanism for dynamic changes in the determination of the object replication style and the number of replicas can be considered as a great advantage and, as a consequent, it causes that in case of programs with bounded time whenever a fault is occurred the changes as per system current situation are deemed necessary.

Therefore, with considering the features as stated above, the determination of the number of replicas and objects replication style in a real time distributed system have been studied and its effect on improved performance of adaptive fault tolerance structures in respect of restrictions having been described as a guideline in improved performance and reliability.

FT-CORBA

In FT-CORBA infrastructure 4 main specifications are noticed such as the object replica, management and distributed of replica, storage and recovery of object and replicas consistency. FT-CORBA architecture has three basic modules: Login & Recovery Service (LRS), Replication Management Service (RMS) and Fault Management Service

(FMS) which have been shown in Fig. 2 separately [1, 8]. Login & Recovery Service is located underneath the ORB and each application object inherits checkpointable interface to allow its state to be retrieved and assigned for purposes of recovery. In case of failure or rising any fault in this section, the recovery action will come out and in any situation get_state() function will reserve the existence state and if it is necessary to recovery, set_state() function active another replica object. The Replication Management Service which consists of 3 service of which Property Manager service is one of most important of this structure which distinguishes the main properties of fault tolerance structure by receiving specifications of a real time system and then a the decision about properties of fault tolerance can made. Main determinable characteristics are as follows: replication style, minimum number of replica, consistency style, membership style check, point interval and etc [1,8]. From Generic Factory Service which is used to create a new object, in case of detecting any fault a message from Fault Manager will be gotten which is responsible for the create and delete of an object group and with relation to the Object Group Manager.

Fault Manager consists of 3 service, Fault Detector which frequently checks if any fault occurs by sending is_alive() message in the program and can detect the fault on one or a few objects in one process or a few processes in one host and will inform to Fault Notifier Service if any fault is detected [1,8]. The Fault Notifier Service receives fault reports from Fault Detector and Fault Analyzer. According to submitted report by Fault Notifier the Fault Analyzer can make the necessary decision for predicting all kind of faults and sending them to the Fault Notifier. As described above, in

Replication Manager Service there is a major Service of fault tolerance structure named Property Manager in which the features associated with determination of Number of replicas, Replication style, Membership style, Consistency style and Check point are specified and we can consider the appropriate decisions for fault tolerance structure through using specifications as sent from real time system and adjustments consistent with current states of real time system can made, for example, on the basis of what response time our real time system needs the replication style in such a way that Fault Tolerance structure will be able to respond the requests in shortest time will be considered. Therefore, with considering the features having been taken into account in FT-CORBA structure this structure can used for adjustment and consistency of real time according to given specifications and restrictions.

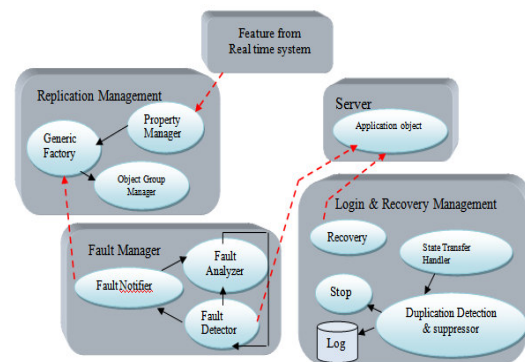


Figure2. FT-CORBA infrastructure

Replication styles and determining the number of replicas

One major feature in FT-CORBA structure and in Replication Manager Service is the determination of replication style in fault tolerance structure. Two common approaches for maintaining replicas are Active and Passive replication. In Active Replication, as shown in Fig. 3 [9], all objects do any

requested operation and, in fact, when a fault is occurred on an object the remainder objects will be ready to respond the said requests. Therefore, in this method all objects have similar states and their compatibility is established on system at any moment. This style results in the execution of a command by several objects and their responses to request. Active replication may lead to duplicate operations because every replica of an actively replicated object sends the invocation or response so to avoid having duplicate operations, it is necessary to have the duplication invocation suppressed [9]. For passive replication only one object designated the primary, processes (invokes) the operations and the remaining passive replicas known as backup, are preloaded into memory, and are synchronized periodically with the primary replica, and then the reply would be given to the invocation sender. In this style if the existing primary replica fails, one of the backup replicas is chosen to be the new primary and, as such must be restored to the state that the old primary possessed. The state of primary must be continuously or periodically captured and stored so that it is available for recovery if the primary replica fails. In the case of passive replication under normal operation, the cost of checkpointing the primary replica's state to a log must be considered if the state of the object is large. This checkpointing could become quite expensive but instead, the passive replication can lead to decreasing using network bandwidth and requires only one replica to be operational, and thus conserves processing power. Thus, the passive replication has the advantage that it consumes processing power less, i.e., it does not require the operation to be performed by each of the replicas [3]. When the recovery time and response time is limited,

the active style should be used to, because all of the replicas of an actively replicated object perform every operation and even if one of the replicas fails, the other operational replicas can continue the processing and perform the operation [10].

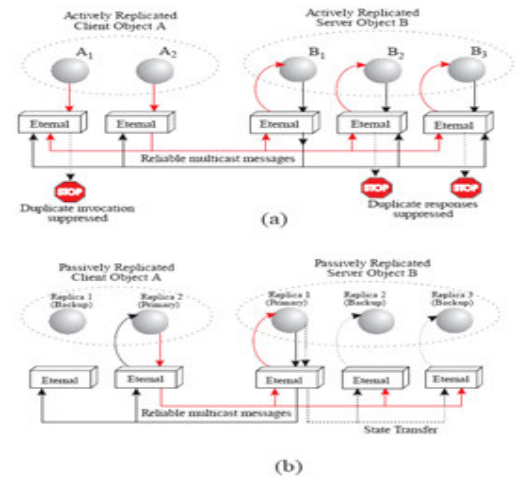


Figure3. Active replication (a), Passive replication (b)

Combination of the aforementioned two styles led to another style so-called Semi-active [6]. In this style, all replicated objects are arranged in a ordered list and they will be in connection with the primary object and any sent message will be dispatched to both the primary object and the first object of this list and it will be executed and, upon the execution of the said command, the remainder objects may be updated to new state. In this way, there is always a replication of objects in consistency with the main object and in case of any failure and fault on the primary object; the first object as appeared on replicated objects list will be chosen as the main object. In conclusion, fault recovery time became faster and this will decrease the sending of messages in network as compared to Active method. As seen, a set of advantages of Active Replication and Passive Replication methods will exist in Semi-active method [6]. With taking into

account the advantages and disadvantages of the above-mentioned methods and in view of the restrictions and parameters of real time system it is necessary to choose an appropriate object replication method for fault tolerance structure. As a result of time limitations and in view of the sensitivity of real time systems the Active method is usually selected for objects replication because in such systems the time to respond the sent request will be considered as an important and basic property [6].

Fault tolerance structures are made in a static manner by FT-CORBA. Therefore, in order to upgrade the performance of FT-CORBA structure an investigation of objects replication style can made and the number of replicas according to objects' current state on the structure and, in this way, to create the best method and the least replication of objects. AFT-CCM is one of the structures having been involved in the adjustment of objects replication style in accordance with programs being executed on FT-CORBA.

In AFT-CCM structure, considering the number of faults occurred during program execution the objects replication style is set dynamically and, finally, led to the adjustment of one major specification of fault tolerance structure in an optimal manner and based on system current state. In this method for which Fig. 4 of machine has been shown in the state related to the determination of objects replication style [11], if only one error is observed in 15 seconds the Passive method will be selected as objects replication style in fault tolerance structure and if two errors occurred in the network in 15 seconds Passive method will be replaced by Semi-active method, and, if there is still increased number of faults and 2 other faults occurred in 15 seconds the Active method will be preferred as objects replication style and these changes

will be similarly continued during program execution.

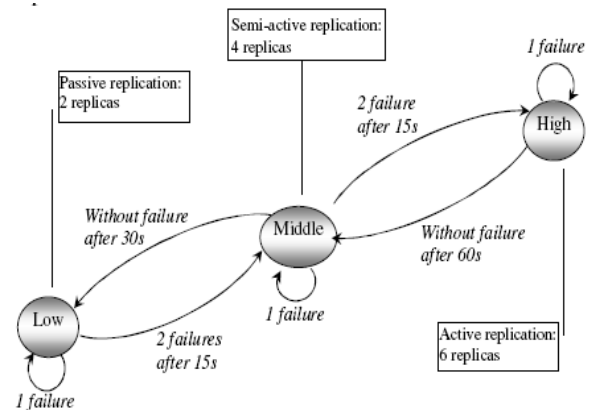


Figure4. State machine of the QoS manager in AFT-CCM structure

In view of the said structure, on the basis of number of faults, the given replication style is selected and, as a consequent, it will lead in upgrading of performance and improved function of FT-CORBA structure on distributed real time systems. This action as illustrated in Fig. 5 is a result of AFT-CCM method showing improved function of fault tolerance structure as compared to previous structures [3, 11].

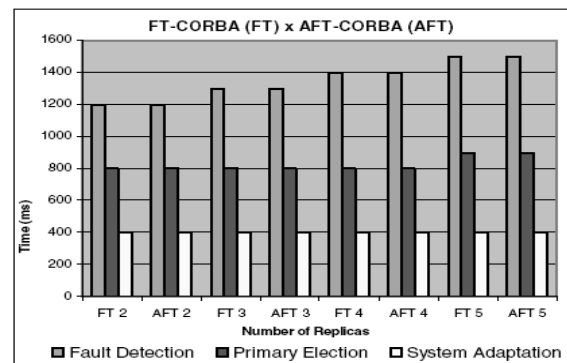


Figure5. compared result in FT-CORBA and AFT-CCM

Adaptive Fault Tolerance Structure

According to the mentioned specification of the distributed real time system, and if the system also needs the creation of fault

tolerance structure, the adaptive fault tolerance structure on real time system should be able to create so that to adjust the fault tolerance structure based on real time system specifications and through creating tradeoff between specifications of both real time and fault tolerance systems to make adaptive fault tolerance structure performance improved. Number of replicas is a parameter of importance in these situations, in addition to objects replication style, and of effectiveness in structure performance. If the number of replicas is selected properly this will result in both well execution of fault tolerance structure and increasing of response time associated with real time system in an acceptable level, therefore, adaptive fault tolerance structure will be placed on real time system so that it will be able to cause specifications related to fault tolerance based on real time system specifications and, in fact, it will lead in a trade off between the specifications of real time and fault tolerance enabling us to improve the performance and function of fault tolerance structure under such conditions.

Considering to the structure shown in Fig.2, one can use real time specifications to regulate fault tolerance structure and to enter those specifications to the Property Manager unit as input parameters. If active replication style is considered to define the number of object replications, then according to the limitation of the response time of each invocation, one can dynamically increase the number of object replications until fault tolerance conditions are met and also not to violate the time limitations of the real time system.

During the system execution, if some invocation is sent to any of the distributed objects so that its response time is not beyond the maximum response time, then it is considered as a fault in the structure and

therefore, the underlying object will be recovered and the current object will be eliminated from the structure and a new object with the current state is added to the system to be replaced to the faulty object. Hence, the number of replications will be always optimum, in order to minimize the increase of response time in the real time system due to the creation of fault tolerance structure.

According to the definition of replications style and the number of object replication in an adaptive fault tolerance structure, these specifications are considered as the regulated specifications based on the real time system, and will be added to the FT-CORBA structure. Therefore an Adaptive FT-CORBA structure will be created based on the real time and it causes the other parts of FT-CORBA which are in communication with Replication Management to perform more efficiently.

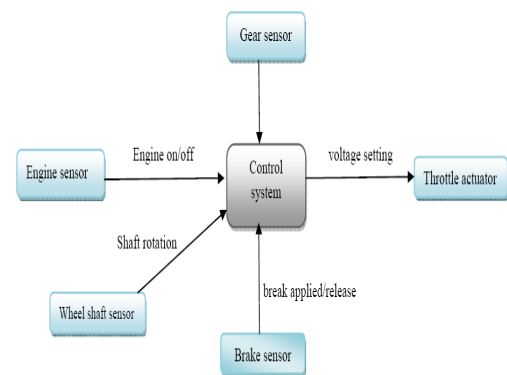


Figure6. All part of the cruise control system

In order to evaluate the results of the mentioned fault tolerance structure, the Cruise control system has been considered as the distributed real time system. Cruise Control system is a real time system which is used to regulate engine speed of the automobile and since the response time and creation of fault are important in such systems, this system have been considered in order to test the fault

tolerant structure. In Fig 6, all parts of the cruise control system are shown. Determining the replication style and the number of replicas in a distributed real time system, these specifications will be considered as adjusted specifications based on real time system and entered FT-CORBA structure. In FT-CORBA structure, a unit named Property Manager exists in Replication Manager which takes the adjustment of main specifications and parameters on the basis of distributed real time system entering fault tolerance structure and, as a result, adaptive FT-CORBA structure is created according to real time system. This causes other parts of FT-CORBA structure be in connection with Replication Manager to perform their activity efficiently.

When studying the results of the given fault tolerance structure, Cruise control system has been taken as a distributed real time system for making fault tolerance structure. Cruise control system is a real time system which is, indeed, used for the adjustment of speed in machines engine and because of the importance of response time and fault occurred in systems it is considered as given system for testing the function of fault tolerance structure and in Fig. 6 all parts associated with cruise control system are shown. So said, it is contained of 6 parts; namely: control system, break sensor, engine sensor, wheel shaft sensor, throttle accelerator and gear sensor. The function of cruise control system is such that the user reaches the desired speed in system through the execution of accelerator command and in any speed user will be able execute the given function at the same speed to keep the system fixed and, if necessary, to decrease the system speed by execution of break function. Each object in the real time system has its own response time and Any operation for execution can be performed

whether through active replication or passive replication. Active replication is so that any operation is sent to all replicated objects in distributed fault tolerance structure and when the runtime of each command exceeds the given response time it will be regarded as fault and the sent response will not be acceptable. The objects replication style can also be considered as passive in fault tolerance structure and, then, to review the execution of real time system and finally to select the appropriate replication style according to obtained results as described below.

In a study on the function of distributed real time system and according to descriptions given about the method of fault tolerance structure execution on real time system, the performance and function of given programs in different situations will be reviewed. Diagram as illustrated in Fig. 7 shows a state in which the servers have been considered as central and the objects are replicated on a system; it is observed that when the number of replicas are taken as passive the response time will be longer than active replication and the number of faults on structure will be higher than a state in which active replication is chosen as objects replication style.

Therefore, it seems that selection of active replication is more suited to real time system in which the response time for requests is important. Also, in a review on the number of objects replication in Fig. 7 if the diagram associated with active replication is taken into consideration. In this state it is seen that when the number of replicas reaches over 9 the average response time will be strongly enhanced and the number of faults on structure will be increased too. Therefore, whenever it is desired that the given real time system to have appropriate response time in addition to fault tolerance capability, the

objects should be replicated with at most 9 replicas. Thus, in property manager of FT-CORBA structure this quantity as input of real time system will be adjusted and the performance of fault tolerance structure will be controlled, so as not to increase the response time of real time system in a not desired manner as well as not to increase the number of replicas in a not required manner, When the execution of program is in such a manner that objects are replicated on different computers. In this state as the same as what observed in previous state, as shown in Fig. 8, the situation of active replication is better than passive replication in terms of response time to requests and the number of faults occurred on objects. If active replication is considered for reviewing of this state 11 replicas can generated up to for each object according to

obtained diagram and to keep the number of faults in an acceptable level, also, the comparison of the both diagrams 6 and 7 shows that, in view of response time as obtained by objects when the objects replication is considered less than 9 replicas, it is better that objects are replicated on a system so that the response of objects will be received in a shorter time. In Figs. 7 and 8 the effect of increased objects replications on the response time have been reviewed, adjustment and improved fault tolerance structure function. Figs. 9 and 10 deal with a state in which the objects replication have been established fixed and optimal and with taking into account the obtained results the objects replication style has been considered active and the clients' role have been studied and their increase in system's response time.

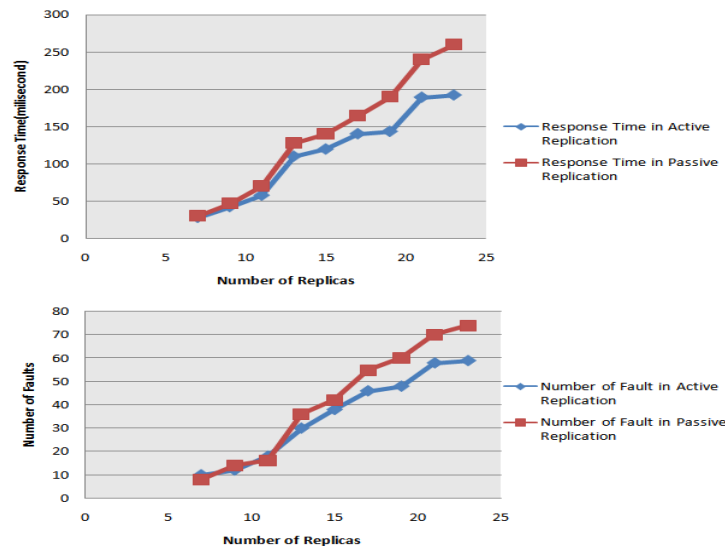


Figure7 .Objects Replication on central servers

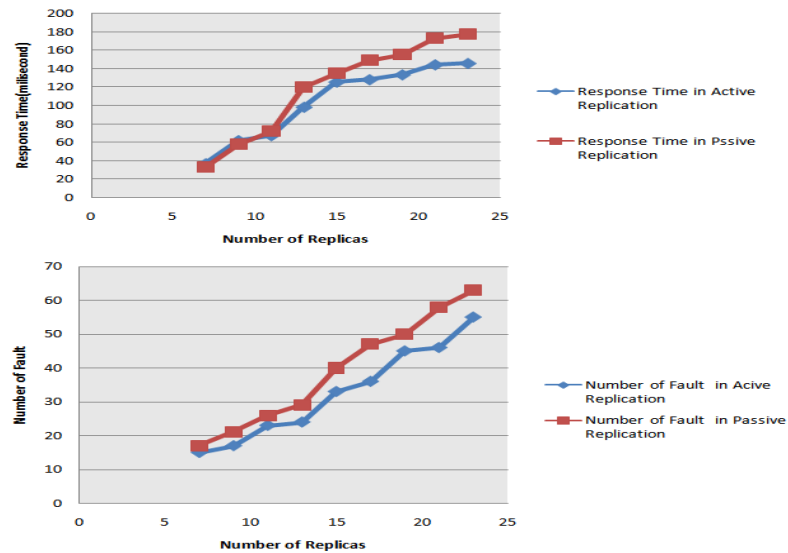


Figure8 .Objects Replication on several computers

Fig. 9 has been considered when the servers are central and the number of replicas is 8. It is observed that when the clients are increased the response times will increase not appropriately and the number of faults are increased on the structure as well and at most up to 8 clients work on system central servers as appropriate and acceptable in respect of response time, but when the objects are replicated on several servers and the number

of clients have been increased, at most up to 15 clients will be responsible at the same time as shown in Fig. 10. The comparison of Figs. 9 and 10 shows that when the replication of objects are fixed and optimal and if the clients have been increased it will be better that the objects replication to be on distributed system rather than be arranged as objects central so that making it possible to receive the response of objects in a shorter time.

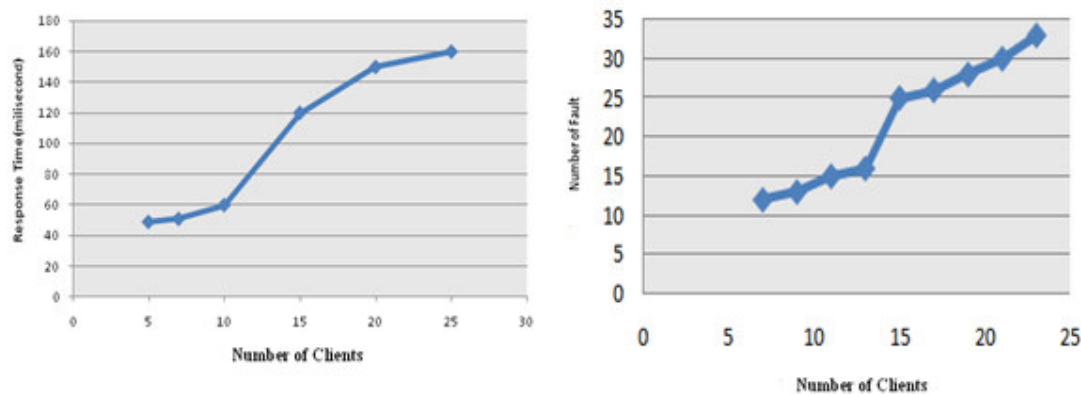


Figure9 .Replication of clients on central servers

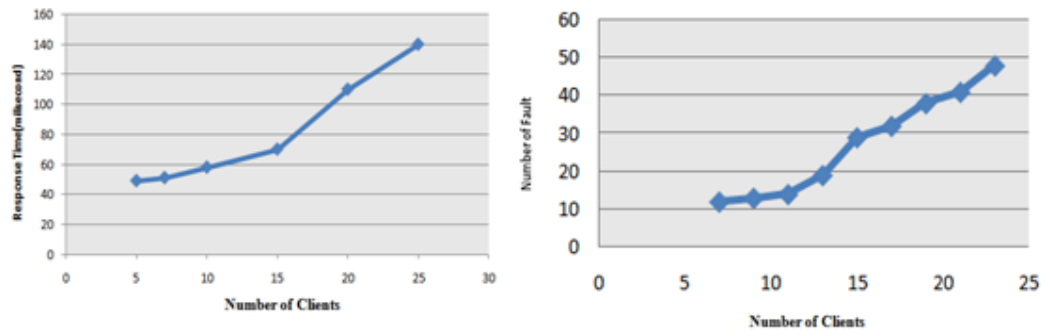


Figure10 .Replication of clients on several computers

Conclusion

With considering the specifications and restrictions of distributed real time systems and since there is need to create fault tolerance structures on these systems, this research has dealt with the review of implementation of adaptive fault tolerance structure for upgrading performance and its adjustability based on real time systems specifications, and, in view of specifications received from real time system the objects replication style and the number of replicas have been studied as effective and sensitive parameters in the function of fault tolerance structure and they were used for FT-CORBA structure. In respect of implementation and the review of results obtained it was observed that this will lead to improved FT-CORBA structure function for distributed real time system and it will include the adjustability of fault tolerance structure on the basis of distributed real time system's specifications.

References

[1]Narasimhan, P., T. A. Dumitras, A. M. Paulos ,S.M.pertet, C.F.Reverte,J.G.Slember and D.Srivastava, "MEAD: support for real-time fault-tolerance CORBA", *Concurrency and Computation: Practice & Experience* Wiley. Volume 17, Issue 12, 2005, PP1527 – 1545.

[2]Balasubramanian, J., S. Tambe, A. Gokhale, D. C.Schmidt, Ch. Lu and Ch. Gill, " FLARe: A Fault tolerance Lightweight Adaptive Real time Middleware for distributed real time and embedded system", *Proceedings of the 4th on Middleware doctoral symposium, Newport Beach, California, Nov26 – 30, article No 17, 2007, PP320-330.*

[3]Narasimhan,P., L.E. Moser, P.M. Melliar-Smith,"Strongly consistent replication and recovery of fault-tolerant CORBA applications", *Computer System Science and Engineering Journal* 17(2), 2002, PP103-114.

[4]Kim, k.H,"Middleware of Real Time object Based Fault Tolerance Distributed computing systems", *Proceedings of the 2001 Pacific Rim International Symposium on Dependable Computing (PRDC2001), 2001, PP3-8.*

[5]Kim, K.H,"ROAFTS: A middleware architecture for real time object oriented adaptive fault tolerance", *The 3rd IEEE International Symposium on High-Assurance Systems Engineering, 1998, PP50-57.*

[6]Sahoo, B., Aser Avinash Ekka, "Backward fault recovery in Real time Distributed system of periodic task with Timing and Precedence

constrain". Proceedings of the International Conference on Emerging Trends in High Performance Architecture, Algorithms & Computing (HiPAAC 2007), 2007, PP 124 - 130.

[7]Felber,P,PriyaNarasimhsn,"Experiences strategies and challenges in building fault tolerance CORBA systems", IEEE Transaction on computer, Volume 53 , Issue 5, 2004, PP 497 – 511.

[8]Object Management Group, Fault Tolerant CORBA, 2004, Chapter 25, CORBA v3.0.3, OMG Document formal/04-03-10 edition. <http://www.omg.org/cgi-bin/doc?formal/01-09-29>

[9]Fraga, J., Frank siqueria, Fabio Favarim, "An Adaptive Fault Tolerance Component

Model", Proc 9th IEEE international object oriented real time dependable system (WORDS'03),01-03 Oct, 2003, PP 179- 187.

[10]Gokhale,A.S, Balachandran Natarajan, Joseph K.Cross, Douglas C.Schmidt,"Towards Real-Time Fault-Tolerant CORBA Middleware", Cluster Computing journal, Volume 7, Issue 4, 2004, PP 331 – 346.

[11]Lung, L.CH, Fabio Favarim ,Giuliana Texeria ,Miguel Correia, "An Infrastructure for adaptive Fault Tolerance on FT-CORBA", Proc 9th IEEE international symposium on object and component oriented Real time Distributed, 24-26 April,16v,2006, PP504-511.