Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (11): 2014-2020, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2014.2020

Fault Tolerant Architecture for Telecom Wireless CORBA

[1]Zhenpeng Xu, [1]Weiwei Li and [2]Yu liu
[1]Jiangsu Automation Research Institute, Lianyungang, Jiangsu, 222061, China
[2]Department of Computer, Weifang University, Weifang, Shandong, 261061, China

Abstract: In order for non-mobile ORB to interoperate with CORBA objects and clients running on a mobile terminal, OMG have specified Wireless Access and Terminal Mobility of CORBA. In the common core of the CORBA specification, Fault Tolerance has been specified. But it is intended for the wired networks. This study proposes a fault tolerant architecture for the Telecom wireless CORBA based on replication and checkpoint of objects. The storage available at Access Bridge is employed to log messages and entity states of objects on behalf of mobile terminals. The logging and recovery infrastructures are designed on each Access Bridge, to implement the fault tolerant for Telecom wireless CORBA. The Logging Mechanism records the message in a log, from which the Recovery Mechanism can retrieve the message during recovery. The performance analysis shows that the proposed fault tolerant architecture ensures a low loss of computing incurred by the fault of the server object. The proposed fault tolerance architecture is a graceful extension of the original wired Fault Tolerant CORBA and is able to cooperate with the published CORBA specifications seamlessly.

Key words: Wireless CORBA, fault tolerance, object replication, checkpoint

INTRODUCTION

To achieve the distributed computing system, Object Management Group (OMG) has specified Common Object Request Broker Architecture (CORBA), which is one of the most popular middlewares nowadays. CORBA provides portability, location transparency and interoperability of applications across heterogeneous platforms (OMG, 2004). Mobile computing system is an extension of traditional distributed computing systems. In order to interoperate with CORBA objects and clients running on mobile terminals, OMG has published Telecom Wireless CORBA specification (OMG, 2005).

In order to meet the requirements of fault tolerance in CORBA, OMG has specified Fault Tolerant CORBA specification based on the entity redundancy (OMG, 2004). This specification provides fault tolerance through the replication of objects and logging recovery mechanisms. Three replication styles are employed for fault tolerance of CORBA. That is cold passive, warm passive and active replications. Logging and checkpointing mechanisms record messages and entity states in logs. Compared to server/process replication, this architecture allows greater flexibility in configuration management of the number of replicas and of their assignment to different hosts. Replicated objects can invoke the methods of other replicated objects without regard to the physical location of those objects. Support for redundancy in time is provided by allowing clients to make repeated requests on the server, using the same or alternative transport paths. But all these are just suitable for the wired networks.

Compared to the wired distributed systems, the wireless link in the mobile systems is more fragile and the mobile terminal is much less reliable due to the certain new characteristics such as dynamic mobility, low wireless bandwidth, disconnection, limited battery life, limited memory and limited stable storage(Elnozahy *et al*., 2002; Yang *et al*., 2006; Gupta *et al*., 2008). Therefore, it is more desirable for the Telecom Wireless CORBA to be equipped with fault tolerant mechanisms to reduce the loss of computation.

In this study, we specifically address a fault tolerant mechanism for the Telecom Wireless CORBA based on logging, replication and checkpoint of objects, which extends the original wired Fault Tolerant CORBA. The storage available at the access bridge is employed as the stable storage to log messages and entity states of objects on behalf of mobile terminals. In particular, we are interested in how to tolerate mobile host disconnection, mobile host crash and access bridge crash. Compared to checkpoint of server/processes, this mechanism based on replication and checkpoint of objects allows greater flexibility in configuration management of the number of replicas and their assignment to different hosts. The graceful fault tolerance architecture extends the original wired Fault Tolerant CORBA and can seamlessly cooperate with the common CORBA specifications. That

Corresponding Author: Zhenpeng Xu, Jiangsu Automation Research Institute, Lianyungang, Jiangsu, 222061, China

is it does not require modifications to the existing non-mobile ORB and modifications to the existing wired Fault Tolerant CORBA specifications.

TELECOM WIRELESS CORBA ARCHITECTURE

The overall architecture in Telecom Wireless CORBA is depicted in Fig. 1.

Home Domain for a given terminal is the domain that hosts Home Location Agent (HLA). HLA keeps track of Access Bridge (AB) that a mobile terminal is currently associated with(OMG, 2005).

AB is the network side end-point of the GIOP tunnel. The AB acts as a relay between the server and client. In addition, AB may support the handoff. AB may also provide notifications of terminal mobility events.

Terminal Bridge (TB) is the terminal side end-point of the GIOP tunnel. TB may also provide a mobility event channel that delivers notifications related to handoffs and connectivity losses.

A Visited Domain is a domain that hosts one or more ABs through which it provides ORB access to some mobile terminals. It also contains some Static Hosts (SHs). All communications in the visited domain are via wired links.

A Terminal Domain consists of a mobile terminal device that hosts an ORB and a TB through which the objects on the mobile terminal can communicate with objects in other networks. The mobile terminal device can move around while maintaining network connections by a wireless interface.

Each AB covers a geographical area within which it can communicate with mobile terminals directly, plotted as dashed circle in Fig. 1. When a mobile terminal moves across the border of the geographical area, a handoff occurs between the new AB and the old AB. a graceful handoff behavior is defined so that the TB can seamlessly transfer the GIOP tunnel from the current AB to a new one. If the mobile terminal can support simultaneous transport connectivity to two AB, then the TB creates a new tunnel to a new AB before shutting down the tunnel to the previous AB. Generally, a handoff consists of three distinct phases: the information gathering phase, the decision phase and the execution phase. To focus on the fault tolerant architecture, the detailed handoff implementation is ignored in this study.

All hosts communicate with each other by messages only. The General Inter-ORB Protocol (GIOP) tunnel is the communication channel to transmit GIOP messages between TB and AB, through which GIOP Tunnel Protocol (GTP) messages are transmitted. GTP is an abstract, transport-independent protocol. It defines message formats for establishing, releasing and reestablishing the tunnel as well as for transmitting and forwarding GIOP messages. No messages can be exchanged

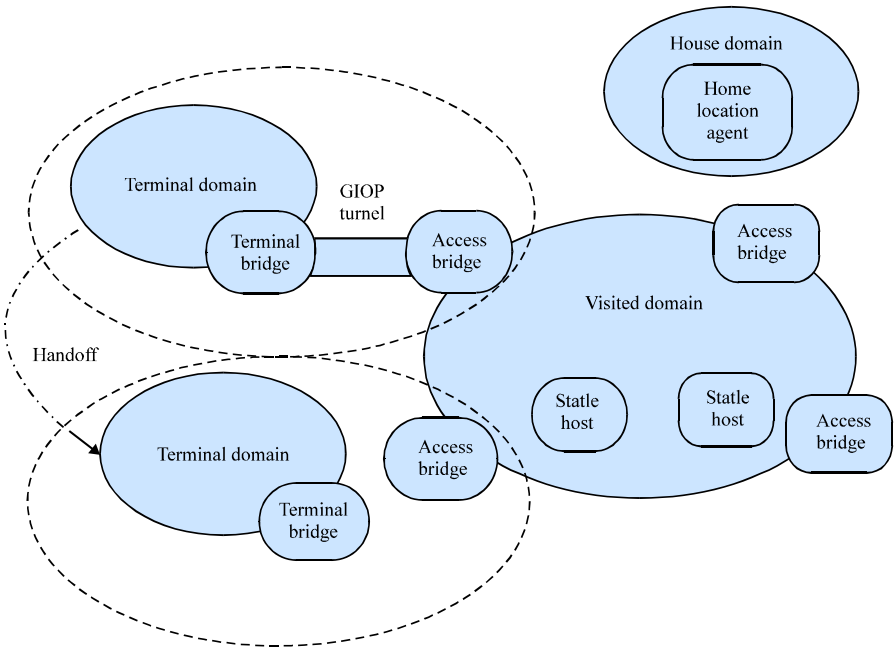

Fig. 1: Architecture for wireless CORBA

among Tbs directly. All messages to and from a mobile terminal are relayed by its currently associated AB.

An object service of a mobile terminal may become unavailable due to mobile host disconnection, mobile host crash and or access bridge crash. As mentioned before, physical damage becomes more probable to a mobile terminal. It is limited to lower processing power, lower memory resources and lower power supply.

RELATED WORKS

In mobile computing system, present fault tolerant mechanisms are mainly based on traditional checkpoint and message logging techniques. In such schemes, the host periodically saves its state at a stable storage. Thus, upon a fault of the host, execution can be restarted from the last saved checkpoint (Brzezinski *et al*., 2006).

It was indicated earlier in (Li and Wang, 2005; Men *et al*., 2006) that a mobile terminal's disk storage cannot be considered stable. Thus, most mechanisms utilize the storage available at the local mobile support station or AB, as the stable storage. There are many proposed process's checkpoint and rollback recovery schemes for mobile computing system (Yang *et al*., 2006; Gupta *et al*., 2008; Brzezinski *et al*., 2006; Li and Wang, 2005; Men *et al*., 2006; Park *et al*., 2002; Zhang *et al*., 2008). However, all the proposed fault tolerant schemes are not suitable for the Telecom Wireless CORBA Architecture which is based on remote object service and these schemes can not seamlessly cooperate with the CORBA specifications (Chen and Lyu, 2003).

In order to meet the requirements of fault tolerance for CORBA, OMG has specified Fault Tolerant CORBA specification based on the entity redundancy and logging mechanism. This specification provides fault tolerance through the replication of objects and logging and recovery mechanisms. Three replication styles are employed in fault tolerance of CORBA, cold-passive, warm-passive and active replications. Logging mechanism records entity states and messages in logs. This architecture allows greater flexibility in configuration management of the number of replicas and of their assignment to different hosts, compared to server or process replication. Replicated objects can invoke the methods of other replicated objects without regard to the physical location of those objects. Support for redundancy in time is provided by allowing clients to make repeated requests on the server, using the same or alternative transport paths. But all these are just intended for the wired networks and powerful static hosts. As a mobile terminal is not suitable to act as stable storage for logging mechanism, it is inadequate that the AB just acts as a relay between the server and client.

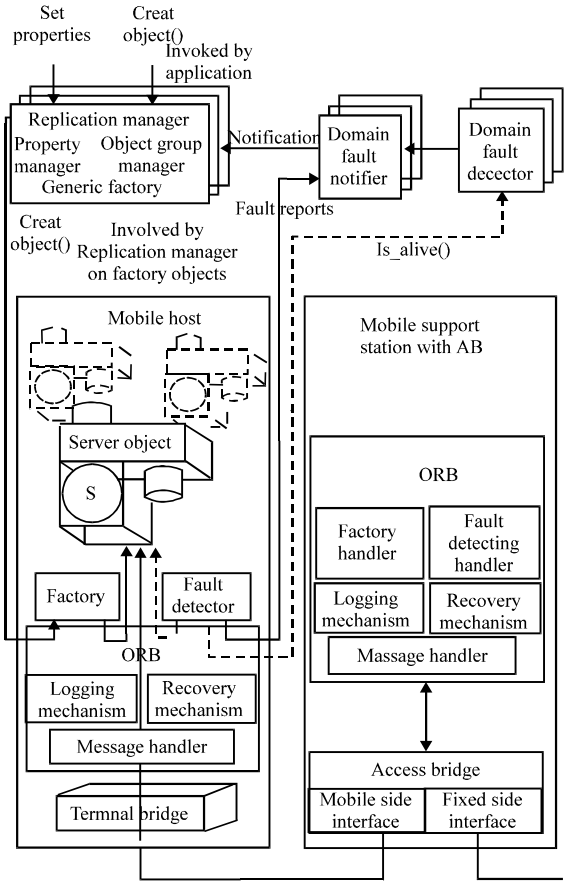

Fig. 2: Fault tolerant architecture

FAULT TOLERANCE FOR TELECOM WIRELESS CORBA

Fault tolerance architecture: Figure 2 presents the architectural overview of proposed fault tolerant Telecom Wireless CORBA mechanism, which is based on replication and checkpoint of objects.

As shown in Fig. 2, besides AB is employed as the agent to implement the existing wired Fault Tolerant CORBA speciation on behalf of mobile terminals; it equips with the logging and recovery infrastructure, which is the main extension of the original fault tolerant CORBA.

At the top of the figure, several components of Fault Tolerance Infrastructure are shown, Replication Manager, Fault Notifier and Fault Detector, all of which are implemented as CORBA objects. Logically, there is a single instance of Replication Manager and Fault Notifier

in each fault tolerance domain, but physically, they are replicated to protect against faults, just as are the application objects. Replication Manager inherits Property Manager, Object Group Manager and Generic Factory interfaces. The left of the figure describes a mobile host holding three client or server objects and the right of the figure shows an AB equipped with the mobile side interface and fixed side interface. A typical system will contain many such client and server objects.

Figure 2 also describes Factory and Fault Detector objects that may be present on each mobile terminal and are specific to that mobile terminal. These host-specific objects are not replicated, unlike the service objects shown at the top of the figure, which are replicated objects. Message Handler, Logging Mechanism and Recovery Mechanism are present on each host. These are not CORBA objects but, rather, are a part of the ORB, or are located between the ORB and the operating system. Factory Handler, Fault Detecting Handler, Message Handler, Logging Mechanism and Recovery Mechanism are also present on each AB. Similarly, these are not CORBA objects but, rather, are a part of the ORB, or are located between the ORB and the operating system.

Replication management: To keep consistent with the existing wired Fault Tolerant CORBA and Telecom Wireless CORBA specifications, the presented Fault Tolerance for Wireless CORBA architecture supports the corresponding replication interface.

If a replication operation of an object on a mobile host is invoked on Replication Manager, the request is relayed to the related destination through AB currently associated with. A replicated object is to be created by the related mobile host using create_object() operation of the local Generic Factory interface.

The Membership style maintained in Property Manager defines whether the membership of an object group is infrastructure-controlled or application-controlled. Through, create_member(), add_member() and remove_member() operations of the Object Group Manager interface, the Replication Manager can exercise control over the addition and removal and location, of members of an object group. The application can also exercise control over the addition and removal and location, of members of an object group in application-controlled style.

AB is employed as the agent to implement the existing wired Fault Tolerant CORBA specification on behalf of mobile terminals. For Replication Manager and other hosts in the system, it seems that the new created object at a mobile terminal belongs to the AB with which the mobile terminal is currently associated.

Similar with the traditional wired Fault Tolerant CORBA specifications, each individual replica on mobile terminal has its own object reference, the object group as a whole has its interoperable object group reference, which is created by Replication Manager. Because of the object group abstraction, the client objects are not aware that the server objects are replicated (client transparency to replication) and are not aware of faults in the server replicas or of the recovery of server replicas when a fault has occurred (client transparency to faults).

To cooperate with the existing wired Fault Tolerant CORBA specification, three replication styles are supported in the proposed fault tolerant architecture, ACTIVE, WARM_PASSIVE and COLD_PASSIVE replication styles.

The ACTIVE replication style requires that all of the members of an object group execute each method invocation on the object group independently but in the same order, so that they maintain exactly the same state and, in the event of a fault in one member, that the application can continue with results from another member without waiting for fault detection and recovery. Even though each of the members of the object group generates each request and each reply, the message handling mechanism detects and suppresses duplicate requests and replies and delivers a single request or reply to the destination object.

For the COLD_PASSIVE or WARM_PASSIVE replication styles, only a single member, the primary member, executes the methods that have been invoked on the object group. The object group contains additional backup members for recovery after a fault.

For the COLD_PASSIVE replication style, the state of the primary is extracted from a log and loaded into a backup member when needed for recovery. For the WARM_PASSIVE replication style, the state of the primary member is loaded into one or more backup members periodically in normal failure-free execution.

Message logging and object state checkpointing: AB is on the border between wireless and wired network. In Telecom Wireless CORBA architecture, all messages to and from a mobile terminal are traversed through ABs. Every message holds a local copy in AB. It does not need to send an extra copy of each message elsewhere for logging purpose to tolerate mobile host crash. Therefore, the storage available at AB is employed as the stable storage to log messages and entity states of objects on behalf of mobile terminals.

Typically, the information recorded in the log consists of request and reply messages. The message log

is preserved the order in which messages were received by the members of the object group, so that they can be replayed in the correct order during recovery.

Each GIOP message is passed to Logging Mechanism and Recovery Mechanism of AB and recorded into the reliable storage, automatically and invisibly to the application. Logging Mechanism records the message in a log, from which Recovery Mechanism can retrieve the message during recovery. Each GIOP message may be also passed to the local Logging Mechanism and Recovery Mechanism of the mobile terminal for logging in order to speed up the failover process in case the fault.

The Checkpointable interface inherited from the object of the mobile terminal, which provides get_state() or get_update() operations, to enable the local Logging Mechanism to record its state. Object states and updates achieved through get_state() and get_update() operations in the Checkpointable interface. States and updates is positioned logically in the message sequence at the point at which they were requested by get_state() or get_update() request message, even though the state or update may be contained in a reply message that is sent at a later time. A complete state consists of get_state() request message and the reply to that request.

In the view of the global Logging Mechanism and other host, it seems that the new checkpoint of an object at a mobile terminal belongs to the AB with which the mobile terminal is currently associated.

The message flow of checkpoint procedure in FT- Telecom Wireless CORBA is shown in Fig. 3.

According to the value of the object group checkpoint interval obtained from Property Manager, get_state() or get_update() is invoked periodically by the Logging Mechanism to deliver checkpoint request to the AB with which the mobile terminal is currently associated.

Upon the receipt of the checkpoint request from the Logging Mechanism, The get_state() or get_update() request will be relayed to the destination mobile host by the AB.

Upon the receipt of the get_state() or get_update () invoke from the local AB, The local Logging Mechanism invokes the get_state() or get_update() operation on mobile host to obtain object's state by the Checkpointable interface inherited. After the application object returns the state. The state information is transferred to the local associated AB through the wireless link and stored by the AB's Logging Mechanism.

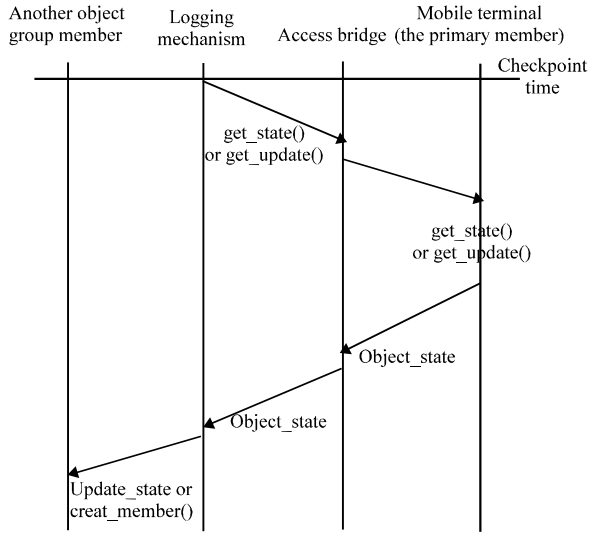

Fig. 3: Message flow of check pointing procedure

The state of the backup is typically updated using the most recent state plus the following updates. For the WARM_PASSIVE replication Style, after the Logging Mechanism obtaining the latest object state needed by get_state() or get_update() operation on the primary member, the backup members of the object is updated in order to speed up the failover process in case the primary fails. Furthermore, the state of the primary member may be used to create a new member of the object group by create_member() operation as shown in Fig. 3.

Recovery upon the Fault:Based on strong replica consistency, the recovery is managed by Recovery Mechanism in the presented fault tolerant architecture.

If Fault Detector suspects that the primary member is faulty, Replication Manager, at its discretion, restarts the current failure primary member or promotes a backup member to become the new primary member.

For a backup member of an object group with the COLD_PASSIVE replication style that is being promoted to the primary member, Recovery Mechanism must apply the entire complete log to the member. In the following recovering process, the new primary member requires to invoke set_state() operation after getting the latest state of the fault object from the entire complete log and apply messages from the log to the member to bring that member to the correct current state, so that it can start to process messages normally.

Upon the receipt of set_state() request and the entire complete log, the local Recovery Mechanism of the mobile terminal invokes set_state() operation and apply all the message logged to the member.

For a backup member of an object group with the WARM_PASSIVE replication style that is being promoted to primary member, the member has already received states and updates during normal operation. Recovery Mechanism applies to the member, only messages in the complete log that follow the most recent state or update applied to the member during normal operation. In the following recovering process, the new primary member only requires to apply messages from the log to the member to bring that member to the correct current state, so that it can start to process messages normally.

For a backup member of an object group with the ACTIVE replication style that is being promoted to primary member, Recovery Mechanism need not apply any log or update to the new primary member, since all the members maintain exactly the same state and, in the event of a fault in one member, that the application can continue with results from the new primary member without waiting for the recovery.

PERFORMANCE ANALYSIS

A simple fault tolerant Telecom Wireless CORBA instance based on the proposed architecture is designed in Java language. Ten objects are implemented to cooperate to complete a complicated matrix multiplication. Each object group maintains three replications member for fault tolerance. The sequence of the discrete failure event in experienced during in the computing is generated according to an exponential distribution with rate γ. Each server object takes checkpoints with the fixed interval T = 100s.

Let β denote the ratio of the cost of transferring a state to the cost of executing a method invocation. The comparison of the total execution time of the program with the proposed fault tolerant mechanism and without any fault tolerant mechanism is illustrated in Fig. 4. With the increasing of the failure rate, the total execution time of program without any fault tolerant consideration increases significantly. The reason is that all the mobile terminal's objects are required to restart from the beginning of the initial state upon a fault of any object without fault tolerant consideration in Telecom Wireless CORBA.

However, the total execution time under the proposed fault tolerant architecture increases slowly with the increasing of the failure rate. The reason is that the object's state is just required to roll back to the latest local checkpoint upon a fault in the proposed fault tolerant mechanism. The proposed fault tolerant architecture

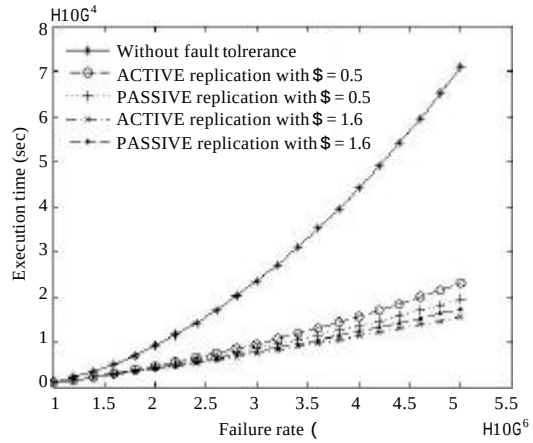

Fig. 4: Execution time in different failure rate

ensures a low loss of computing incurred by the fault of the server object. It is observed that the execution time is proportional to the failure rate. Therefore, the effect of the proposed fault tolerant architecture is more important in Telecom wireless CORBA environment with a high failure rate.

Furthermore, as shown in Fig. 4, ACTIVE replication is useful when the cost of transferring a state is larger than the cost of executing a method invocation, or when the time available for recovery after a fault is tightly constrained. Typical examples include enterprise electronic trading applications and safety-critical applications, such as hospital patient monitoring. PASSIVE replication is useful when the cost of executing a method invocation is larger than the cost of transferring a state and the time for recovery after a fault is not constrained. Typical examples include enterprise inventory, logistics applications and hospital record keeping.

CONCLUSION

In this study, a fault tolerant architecture for the Telecom wireless CORBA based on replication and checkpoint of objects is proposed. The storage available at the AB is employed as the stable storage to log messages and entity states of objects on behalf of mobile terminals. The logging and recovery infrastructures are designed on each AB, to implement the fault tolerant for Telecom wireless CORBA. Each GIOP message is passed to the Logging and Recovery Mechanisms, automatically and invisibly to the application. The Logging Mechanism records the message in a log, from which the Recovery Mechanism can retrieve the message during recovery.

The details of how to tolerate mobile host crash are introduced. The final performance comparison shows the proposed ensures a low loss of computing incurred by the fault of the server object. The proposed fault tolerance architecture is a graceful extension of the original wired Fault Tolerant CORBA and is able to cooperate with the published CORBA specifications seamlessly. It is therefore particularly attractive for Telecom wireless CORBA environment, in which the mobile terminal is more fragile and is much less reliable.

ACKNOWLEDGEMENT

The work is supported by Natural Science Foundation of Jiangsu (Project No. BK2012237).

REFERENCES

OMG, 2004. The common object request broker: Architecture and specification. Version 3.0.3-Editorial changes formal/04-03-01, Object Management Group, March 2004.

OMG, 2005. Wireless access and terminal mobility in CORBA. Version 1.2., Object Management Group. http://www.omg.org/spec/WATM/1.2/.

Elnozahy, E.N., L. Alvisi, Y.M. Wang and D.B. Johnson, 2002. A survey of rollback-recovery protocols in message-passing systems. ACM Comput. Surv., 34: 375-408.

Yang, J.M., D.F. Zhang and W.W. Li, 2006. A robust and efficient rollback recovery implementation scheme. Acta Electron. Sin., 34: 237-240.

Gupta, S.K., R.K. Chauhan and P. Kumar, 2008. Backward error recovery protocols in distributed mobile systems: A survey. J. Theor. Applied Inform. Technol., 4: 337-347.

Brzezinski, J., A. Kobusinska and M. Szychowiak, 2006. Checkpointing and rollback-recovery protocol for mobile systems with MW session guarantee. In: Proceedings of the 20th International Parallel and Distributed Processing Symposium, April 25-29, 2006, Rhodes Island, Greece, pp: 8-15.

Li, G.H. and H.Y. Wang, 2005. A novel min-process checkpointing scheme for mobile computing systems. J. Syst. Archit., 51: 45-61.

Men, C., N. Wan and Y. Zhao, 2006. Using computing checkpoints implement efficient coordinated check pointing. Chin. J. Electron., 15: 193-196.

Park, T., N. Woo and H.Y. Yeom, 2002. An efficient optimistic message logging scheme for recoverable mobile computing systems. IEEE Trans. Mobile Comput., 1: 265-277.

Zhang, Z., D. Zuo, Y. Ci and X. Yang, 2008. A rollback recovery algorithm based on causal message logging in mobile environment. J. Comput. Res. Dev., 45: 348-357.

Chen, X. and M.R. Lyu, 2003. Message logging and recovery in wireless CORBA using access bridge. In: Proceedings of the 6th International Symposium on Autonomous Decentralized Systems, April 9-11, 2003, Pisa, Italy, pp: 107-114.