Transaction Commit Protocol in Mobile Environment: A Survey

Muzammil Hasan

Department of Computer Science & Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur(UP), India

muzammil.mmmec@gmail.com

Abstract-- Mobile computing [2] is one of the most growing fields in recent days; a large number of researches are going on in the field of mobile computing for atomic commit protocol [1] for mobile transaction [3] in mobile environment. Various protocol are being proposed by various researchers for committing mobile transaction, most of them are based on the Two Phase Commit (2pc) protocol proposed by Jim Gray which ensures basic ACID (Atomicity, Consistency, Isolation, Durability) property of transaction. In this paper we surveys the various proposed protocol and try to find out at which extent they serve their purpose and how they deal with limitation of mobile network such as limited resources, less computational power of device, frequent disconnection of mobile device, unreliable wireless link and mobility of mobile device.

Keywords--Cell, Handoff, Transaction Fragment, Mobile Computing, Committing Protocol (ex. 1PC, 2PC)

I. INTRODUCTION

The concept of transaction has already proven to be very important in the field of committing protocol in mobile environment. When a transaction is take place in mobile environment it must satisfy basic properties of transaction that are called ACID properties, Where A stands for atomicity which means process all of transaction or none of it, C stands for consistency which means when a transaction is executed it includes numbers of system data on all system must be in same state, I stands for isolation which means while transaction are executing they do not interact with other transaction act as they are independent, D stands for durability which means effects of a completed transaction must be persistent in nature. When a commit protocol is designed then it must be kept in mind that all basic properties of transaction must be satisfy.

In mobile environment transaction are distributed in nature they access data at several sites, In distributed environment each site has a local transaction manager responsible for maintaining a log for recovery purpose, participating in coordinating the execution of transaction executing at that site and each site has a transaction coordinator responsible for: starting the execution of transaction that originates at the site, distribute transaction fragment for execution at appropriate sites, coordinating the termination of transaction that originates at the site.

Commit protocols are used to ensure atomicity across sites when a transaction which executes at different sites must either be committed at all the sites or aborted at all the sites, commit protocol not accept a transaction executed at one site and aborted at another. First protocol that was proposed for committing transaction in distributed environment is known as One Phase Commit (1PC) protocol proposed by Jim Gray[Gray 78] but it had no of drawbacks which were then solved but two phase commit protocol. Two Phase commit protocol(2pc) is now most commonly used protocol in distributed environment for committing transaction. Most of the protocols that are develop for committing transaction in mobile environment are based in 2PC for example :Mobile 2PC(M2PC).

The rest of paper is organized as follows. Section 2 describes mobile system architecture and section 2.1 present limitation of mobile system architecture. Section 3 describe various commit protocol exist in mobile architecture. Section 3.1 describes A Two Phase Commit Protocol for Mobile Environment (M-2PC) [8]. Section 3.2 describes Unilateral Commit Protocol for Mobile and Disconnected Computing [4]. Section 3.3 describes the Integrated Commit Protocol for Mobile Network[5] and Section 3.4 describes about Timeout based Mobile Transaction Commitment Protocol[6] and section 3.5 describe about modified reliable timeout based commit protocol[11].
Section 4 present a discussion about protocol studied in section 3. In section 5 we present conclusion of paper.

II. MOBILE SYSTEM ARCHITECTURE

A mobile architecture is made of two main parts: fixed part of network and wireless part of network they contain mainly two types of entities Mobile Host (MH) and Fixed Host (FH). MH are those which kept changing their location and fixed host are those they do not change their location. MH are connected to fixed part of network via a special FH called Base Station (BS) or Mobile Support Station (MSS). Each BS covered a geographical area called cell. BS communicates with MH via wireless medium and it communicates to fixed part of network via wired medium. A MH can communicate directly with BS of cell in which he moving currently.

When during a communication one MH moves from one cell to another cell then this process is called handoff. Handoffs may be of two types: soft handoff or hard handoff.

2.1. Limitation of Mobile Environment

Mobile environment is suffered from following limitation:

- Range and Bandwidth
- Frequently disconnection of MH
- Unsecured network
- Location management of MH
- Power consumption of MH

III. COMMIT PROTOCOLS IN MOBILE ENVIRONMENT

Various Protocols are presented by various researchers for committing transaction in mobile environment.

Here we are surveys some very famous protocol among them. A Two Phase Commit Protocol for mobile Environment (M-2PC) Unilateral Commit Protocol for Mobile and disconnected computing (UCM) it try to find out solution of the problem which arise when 2PC is implemented in mobile and disconnected computing. Next we discuss Integrated Commit Protocol for mobile Network (ICP) and later we discuss Timeout based Mobile Transaction Commit Protocol (TCOT).

3.1. A Two Phase Commit Protocol for Mobile Environment (M-2PC)

In this Paper[8] author present a protocol which named M-2PC which preserves the basic principle of 2PC and try to minimize the impact of unreliable wireless communication. In M2PC at the time of execution of transaction three important roles are played by entities taken place in execution of transaction. Transaction Initiator which is the MH initiating transaction, participant which are the processing entities of the execution of transaction and coordinator which is responsible for consistent termination of transaction.

In this paper author define two cases for execution of transaction mobile environment: first case is when client is mobile and servers are fixed. In first case client (MH) initiates the transaction and send log to coordinator which is H-BS (Base Station in which Client is registered) after sending request client can disconnect to H-BS now H-BS act as coordinator but client has to informed about its current location when it goes into new cell. Now coordinator send prepare message to all participants then all participant send vote message to coordinator then coordinator send decision message to all participant then all participant send ack message to coordinator finally decision send to client then client send ack to coordinator and then coordinator release all acquired resources.

In second case both client and servers are mobile. In this case client sent commit and log to coordinator then coordinator send prepare to message all participant then all participant send vote (yes or no) and log to participant agent(participant BS) then participant agent send vote to coordinator then coordinator send decision to participant agent and then participant agent send ack to coordinator and decision to participant then ack message is sent to participant agent by participant and then participant agent release all acquired resources then final decision is sent to client then client send ack to coordinator after receiving ack coordinator release all acquired resources.
If there are $n$ nodes are taken place in the execution of transaction then numbers of message transfer in M2PC is $4n-1$.

### 3.2. Unilateral Commit Protocol for Mobile and Disconnected Computing

When 2PC is implemented in mobile environment it suffers from following problem: offline processing, lightweight capabilities of mobile devices, moving devices. In this paper[4] author try to solve out those problem and represent a unilateral commit protocol for mobile environment having following properties : support offline execution and disconnection during commit, decrease the cost of wireless communication by reducing the no of message transfer over wireless medium but for achieving this properties some assumption are applied on the way participants manage transaction. UCM eliminates the voting phase of 2PC by applying some assumption on the participant.

The author of UCM assumes that five type of component takes part in execution and termination of transaction namely the Application that ask for the execution of sequence of operation, LogAgent that logs each operation before execution, the Participants that execute these operations, the coordinator that guide termination protocol and the PAgents (one for each Participant) that represent the participant in termination protocol and play an active role during recovery. In UCM Coordinator is located in fixed network while the other components can be potentially hosted by mobile this is done for reducing wireless message cost.

In this paper author compare UCM with 2PC implemented in mobile environment on the basis of number of message transfer, message round and latency (the latency of an atomic commit protocol is defined as time spent between submission of commit request to application until the reception of decision by the participants) .let $n$ denotes the no of participant and $\Delta$ time delay to deliver a message.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Message Rounds</th>
<th>Latency</th>
<th>Message Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCM</td>
<td>1</td>
<td>$\Delta$</td>
<td>2n</td>
</tr>
<tr>
<td>2PC</td>
<td>2</td>
<td>$3\Delta$</td>
<td>4n</td>
</tr>
</tbody>
</table>

As define above UCM apply some assumption at participants how to manage transaction if participant fails to apply those assumption then it does not able to satisfy basic properties of transaction.

UCM is not fault tolerant in nature once coordinator is failed they does not define any way to choose a coordinator.

### 3.3. Integrated Commit Protocol for Mobile Network

In this paper [5] author present a protocol which contains the feature of 2PC of sending less no of message, non blocking behavior of 3PC and it also uses the Paxos Consensus [7] for achieving its non blocking behavior. In this protocol a cluster (group of nodes) which are more stable as compare to other nodes and also having a single hop distance select as group of coordinator and one of them select as a main Coordinator. In ICP Initiator send request to databases, databases send their decision to their Coordinators, then all Coordinator send their votes to main Coordinator, then main Coordinator send his decision to Coordinators then all Coordinator send acknowledgement to main Coordinator then finally main coordinator send their final decision to Coordinator then coordinator send their decision to their databases and databases finally send result to initiator.

If in any case main coordinator fails or network partition is occur then it uses Paxos Consensus to select a new main coordinator .If there are total $n$ numbers of database and $f$ main coordinators in the network then total numbers of message will be $(3n+4f+1)$.Where 4f are the numbers of message transfer between coordinator and main coordinator and remaining message are equivalent to numbers of message transfer in 2PC protocol.

It is not necessary in mobile network that nodes select in ICP for group of coordinators are situated at single hop distance then in this case if network partition is occur or main Coordinator fails then message cost will be increase and no of messages also increased .There is also one main problem in ICP it only deal with non compensatable Flat Transaction.

### 3.4. Timeout based Mobile Transaction Commitment Protocol

In this paper [6] author present a one phase commit protocol with timeout constraints. It is named as TCOT. In TCOT author describe two type of timeout values: Execution Timeout ($E_t$) define time in which a node of commit set completes the execution (commit or abort), Shipping Timeout ($S_t$) define time to ship data from H-MU(MU from which transaction initiates) to Database.
In TCOT a BS (Base Station) (may be H-BS or Current BS) select as coordinator in which H-MU (from which transaction initiates) is registered. A transaction is initiated from the H-MU and request is sent to coordinator then coordinator assigns transaction fragment to each participant with timeout constraints. If during the execution of transaction H-MU shift to another cell (Handoff) then it registered to new BS then current BS is act as a coordinator and then new coordinator (Current BS) informs member of commit set about new coordinator through wired channel.

TCOT is non blocking in behavior due to use of its timeout constraints because it does not wait for infinite time for a decision from MU. In TCOT each MU assigns a Et for execute its assigned transaction fragment which is stores at coordinator. If a MU is not able to execute transaction in its assign timeout values then it sends request to increase its timeout value then coordinator assign him new Et and also updates his new Et in its update cache. If Coordinators does not receive commit message from all databases in commit set then it sends global abort message to those databases who committed their transaction fragment. TCOT also deals with fragment compensation if transaction cannot be committed as whole.

As shown in paper TCOT is compared to modified 2PC (M2PC) and its perform better then M2PC [8] in terms commit time, throughput and message cost but it does not deal with failure of coordinator. If coordinator fails then it stop working until coordinator is recovered.

3.5. Modified Reliable Timeout based Commit Protocol

The Modified Reliable Timeout Based Commit Protocol is a one-phase commit protocol, supports disconnections and mobility. It eliminates the voting phase of 2PC during which the coordinator verifies that the participants can guarantee ACID properties and is an extension of RTBC Protocol [9][10], which includes the features of UCM[4], TCOT[6].

In MRTBCP transaction is initiates and fragment at the MH, the et0 fragment is executed at MH and remaining fragments are transferred to part-FH’s. Once the Part-FH’s receives their respective fragments, they compute and send Et to the MH, after receiving all Et’s, the MH calculates maximum Time (Tm=Max (Et0,Et1,…….EtN)) required to execute the transaction at MH & Part-FH’s. Let Et being an upper bound of the execution time, i.e. just long enough to allow a fragment to successfully finish its entire execution on participant site.

While executing at Part-FH, if time expires before they acknowledge MH, then MH decides to abort and informs global abort decision to all. The Part-FH can unilaterally abort the transaction. A global commit is decided by the MH if it receives an acknowledgement from all the participants before time expiry. Once all the acknowledgments are received by the participants, it issues a commit request to CO. The coordinator force-writes and delegates the commit messages to participants at wired network and waits for an acknowledgement. After receiving all acknowledgments the coordinator informs the MH, about the decision.

The transaction’s commit and acknowledgment messages are continuously logged at Log Agent to ensure atomicity. If the transaction reaches validation phase then the global decision is commit. If any problem arises, the global transaction is immediately aborted. The maintenance of DB’s and log autonomously at each participant insures the proper recovery in case of its failure. At this point, ACID properties are guaranteed by the participants for all the local transaction branches.

IV. DISCUSSION

M2PC is based on 2PC protocol, it allows disconnection of mobile nodes during execution of transaction but it behave as a blocking protocol because it has to wait for acknowledgment from all participant to execute a transaction and total no of message transfer in M2PC is one less then message transfer in 2PC. ICP is a protocol which comprise the feature of 2PC and 3PC with consensus protocol. In ICP 3n+4f+1 message is transferred(n messages initiator to databases, n messages votes from databases to coordinator, n messages do commit from coordinators to databases, 3f message due to use of 3PC, f messages from coordinators to main coordinator, 1 message from main coordinator to initiator) ICP is the best protocol in term of failure of coordinators due to its fault tolerant nature.

MRTBCP is one phase commit protocol which is extension of RTBCP and it includes the features of UCM and TCOT. MRTBCP takes less commit time in compare of TCOT because it does not uses Shipping Timeout. But MRTBCP is complex in nature because log agent is maintained at each participant site. TCOT is non blocking protocol in nature because it apply timeout constraints. TCOT is best in term of commit time as compare to other commit protocol for mobile environment.
UCM is single phase commit protocol which is based on 1PC approach. In UCM no of message transfer is 2n where n is the numbers of mobile nodes. UCM is best in term of wireless messages. M2PC, ICP and UCM is based on legacy protocol used for transaction commit (1PC and 2PC) approach while TCOT use new timeout approach.

V. CONCLUSION

In this paper we address transaction commitment problem in mobile environment. It describes challenges face in committing transaction in mobile environment and surveys paper dealing with transaction commit in mobile environment. Most of the protocol discussed above in this paper are based on 2PC and ensure ACID properties of transaction. This study represents first step of our ongoing research that help us to design a commitment protocol for mobile environment and help to satisfies as many requirements as possible.

REFERENCES