International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2017, pp .193-198
e-ISSN:2278-621X

DISTRIBUTED TRANSACTION PROCESSING MODEL IN MOBILE COMPUTING

C.G.Thomas[1], Rukaya Banu[2] & Prakshith Bhat[3]

Abstract- Advancements in networking and distributed processing are enabling the emergence of new types of distributed processing environments. The environment for accessing and processing information is rapidly changing from stationary to mobile and location independent. Mobile users are more likely to face with more disconnection because of the properties of the mobile environment. Consequently, transaction processing and efficient update techniques for mobile and disconnected operations have been very popular. Because the traditional techniques do not function properly in a disconnected distributed environment, new mechanisms are to be developed for the management of mobile transaction processing.This paper thus focuses on a distributed transaction model that resolves issues of mobile computing environment such as disconnection transaction processing, distributed execution, mobility and transaction properties by using a new technique; we have referred this model as Distributed Transaction Processing Model and the technique.
Keywords- Transaction Models, Distributed Transaction, Mobile Computing, Transaction Processing, Mobility

1. INTRODUCTION

Mobile transaction is a transaction performed with at least one mobile host in its execution; Progress in web transactionsare innovated and furthermore need for portable gadgets made another paradigm, called cloud computing, where users carrying portable devices have the opportunity to access the information and services regardless of their physical location and movement behavior. Portable clients would face problems with the separation, due to the properties of the portable surroundings. Remote engineering organization gives clients the capacity to with-hold their association. Those database which performs operation could gather together a unit from claiming doing guidelines may be called transaction. Those portable registering standard introduces new specialized foul issues in the region for database frameworks. Toward utilizing database applications, portable clients ought further strengthening bring the capacity will both inquiry Also redesign public, private, Furthermore corporate databases. However, techniques for traditional distributed database management have been based on the assumption that the location of end connections among hosts in the distributed system does not change. On the other hand, over cloud computing, these presumptions need aid never again be substantial and also versatility of hosts makes another sort of area that migrates as hosts move. In this paper we discuss the fact that it needs a test for analysts to distinguishwhat'simportant in all the productive transaction preparing or furthermore upgrade systems for portable and disengaged operations.

2. MOBILE TRANSACTIONS

The portable transaction may be situated about similarly free (component) transactions,which might be an opportunity, to be inserted done whatever approach for other versatile transactions. An component transaction can be further worsened under an additional part transactions and consequently versatile transactions harden backing a optional level for nesting.

Transaction Properties:

Atomicity: -It must be treated as atomic unit, and all of its operations are executed or none. It should be cleareither before the execution of the transaction or after the execution failure or success.

Consistency: -The Data should remain consistent after any transaction. It should not have any effect on the data.

Isolation:- Here, no transaction will affect the presence of any other transaction. A transaction in process and notyet committed must remain remote from any other transaction.

Durability:-That means transaction should be durable enough to hold up the updates even if system fails orrestarts.

3. MOBILE TRANSACTION ISSUES

3.1 Mobile Database

[1] Asst. Professor, Department of Information Technology, AIMIT, St. Aloysius College (Autonomous), Mangaluru, Karnataka, India
[2] MCA V SEM, Department of Information Technology, AIMIT, St. Aloysius College (Autonomous), Mangaluru, Karnataka, India
[3] MCA V SEM, Department of Information Technology, AIMIT, St. Aloysius College (Autonomous), Mangaluru, Karnataka, India

Currently, those portable transaction may be created on the highest priority on right now existing database framework. In this environment, the database resides, replicated and dispersed on the settled hosts in wired environments.

3.2 Service Handoff

When an MH moves under another region, another BH will be allocated to this MH. Data regarding current transaction state is spared and also exchanged starting with old BH with following BH. This operation sometimes is unnecessary because not all the time MH requires assistant.

3.3 Scheduling

Execution time of mobile transaction is variable. Portable transaction might effortlessly miss its obliged due date because of its versatility and more portability. It will be not pertinent done portable transaction if a forgetting due date transaction will be generally aborted.

3.4 Caching

Caching of data can improve presentation and simplify disconnected operation. Much research has been performed in the zone of caching. Caching issues are difficult by the use of Location dependent data (LDD).

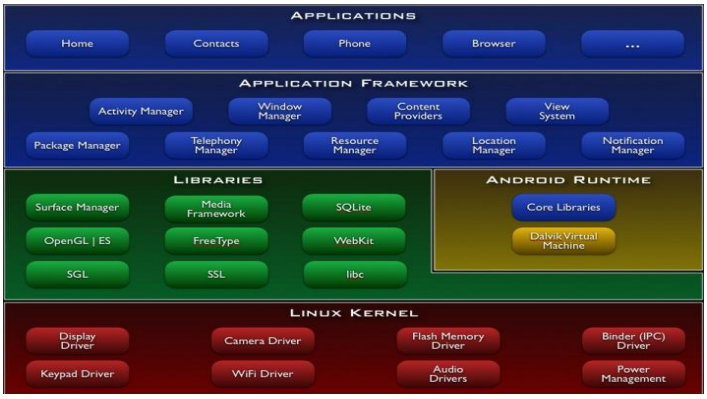
4. OBJECTIVES & SCOPE

We are working to create an Android application to solve the network issue on Mobile Computing. The environment for accessing and processing information is rapidly changing from stationary to mobile and location independent. Transaction processing and efficient update techniques for mobile and disconnected operations. All these concerns are also put forward in the objective of the application and the scope of this paper is to elaborate on the measures to be taken and the reasons justifying the usage of such technique.

5. ANDROID OVERVIEW

Android is an open source and Linux-based Operating System for mobile devices such as Smartphone's and tablet computers. Android offers a combined approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. It was established by the Open Handset Alliance, led by Google, and also other companies.

6. ANDROID ARCHITECTURE



- Android applications: This is the highest layer in the Android stack & is covered of applications that are built-in or any other third party applications. Applications that we develop are also installed in this layer. Applications like: - Camera, Clock, Calculator, Contacts, Calendar, and Media Player etc.
- Application framework**:** This layer is built using Java and delivers high level services and APIs that are provided by the applications. The key Android framework include: Activity Manager, Content Providers, Resource Manager, Location Manager, Notifications Manager, View System, and Telephony Manager.
- Native libraries layer: The Android runtime is contained mainly of the core libraries and the Dalvik Virtual Machine. It is responsible for delivering support for the core features. It converts the byte codes having .class extensions that generated by the Java compiler into the Dalvik executable files that have .dex extensions. The core features of Linux include multi-threading, process and device management, and memory management.
- Linux Kernel: It is the lowest most layer in the Android architecture. It provides support for memory management, process management, and device management. It contains a list of device drivers that provide the communication for

an Android device with other outer devices. The device driver is software that provides a software interface to the hardware devices.

7. TRANSACTION PROCESSING MODELS

We reviewsome existingselected transaction models that have the quality of being able to performwith anefficientmannertosupportmobile transaction management. Thefollowing transaction modelsisdefinesthemobileenvironmentbasedonthe traditional transaction models.

7.1 Report andCo-transactionmodel

Thismodelisproposed byP.K.Chrysanthis and grounded asacontextofspecificmultidatabase system (MDBS).This model isconsidered as a collection of subtransactioneither nestedor open nested transaction model. Nestedtransaction is a parenttransactionmakeschildtransactionsupports more the quality of beingadaptable than atomic transaction.It doesn't sharetheresultbetweenparent and child transaction while transactions are executed.Itallowshierarchy oftransactionnestinglevels and obeysthe bottom-upapproachbytherooti.e. whenachild transactionsuccessfullyexecuted,the object changedby it can be easily obtained toits parenttransactions.Theconsequence of objectmade lasting in a database only when the parent transaction (root) successfully executed. Thismodel arranges the mobiletransaction into following four types:

Atomic transactions- It is related with substantial events like Begin, Commit, and Abort havingthe normalaborts & commit properties.

Non-compostable transactions- Itisnotlinked withcompensating transaction. It canexecuteat anytimeandthe parents of these transactions have the responsibility to commitandabort.

Reporting transactions: A report can be regarded as a delegation of state between transactions. The reporting transaction not assigning all its results to its parent transactions .It only has one receiver at any time during execution. The updating is completed permanently if receiving parent transaction is successfully executed but if receiver parent transactions unsuccessfully terminate then corresponding reporting transaction abort.

Co-transactions: These transactions executed like co-procedures executed. When one transaction is executed then control passes from current transaction to another transaction during sharing the results. At a time either both transaction successfully executed or failed.

7.2 Kangaroo transactionmodel

ThismodelisproposedbyDunham; This Transactions incorporatetheproperty thatthetransactions inamobileenvironment jumpfromonebase stationtoanotherasthemobile unitmoves.Themodelcapturesthismovementbehavior andthedata behaviorreflecting theaccesstodatalocatedindatabasesthroughoutthe staticnetwork.Thistransaction modeldevelopsand grows based onabstractideaof global andsplit transactioninmulti database environment.

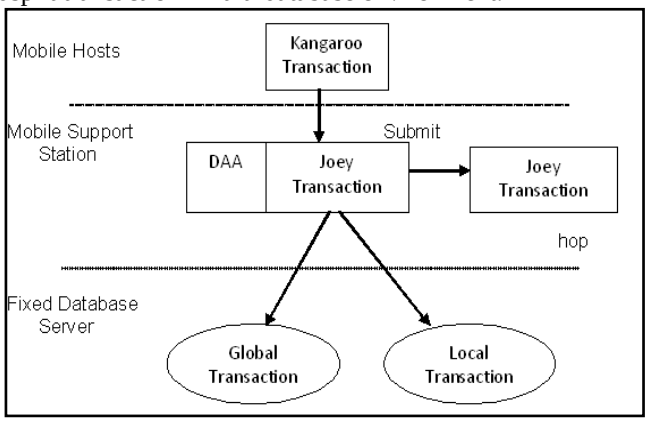


Figure1-Kangarootransactionmodel

7.3 Two-Tier transactionmodel

This model is proposed by Gary and also called as Base Tentative model. This model is grounded on a data replication scheme. For each object, there is a master data copy and various replicated copy. Transactions are arranged in two categories: Base and Tentative. Base transaction function on the master copy whereas tentative transaction retrieves the replicated copy. When the mobile host is abrupt, tentative transactions modify the replicated data copy.When the mobile host reconnects, tentative transactions are converted to Base transactions that are re-executed on the master copy. Tentative transactions topically commit on the replicated copies and the dedicated result is produced for visible to other tentative transactions.

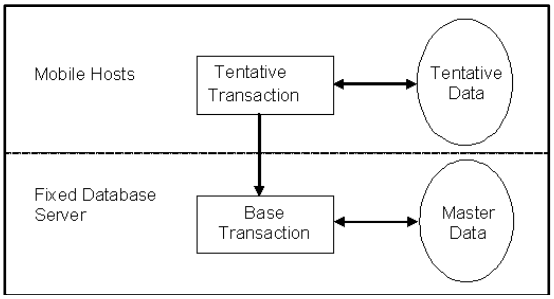


Figure 2- Two-tiertransactionmodel

7.4 Multidatabase transactionmodel

Themobilehostcanplaymanyrolesinadistributed database environment. It maysimply submitoperationstobeexecutedonaserveroranagentatthefixednetwork. A structure for versatile figuring in an agreeable multi-database handling condition and a worldwide exchanges director office is additionally presented. Eachmobileclientisassumedtosubmitatransactiontoacoordinating agent.Once thetransactionhasbeensubmitted,thecoordinating agentschedules and coordinates its executiononbehalfofthemobileclient.Mobileunitsmayvoluntarilydisconnectfrom thenetworkpriortohavingany associatedtransactionscompleted. They pointed a design that gives undeniable exchange administration structure to the clients and application programs so they will have the capacity to get to information over different locales straightforwardly, implementingtheconceptextensibility tosupportvariousdatabasesystemsinthe framework so that thecomponentscancooperatewitharelationaloranobject- oriented databasesystem incorporating the concept of mobile computing through the use of mobile workstations into themodel.

7.5 Pro-motiontransactionmodel

Thismodelisproposedby G.D.Walborn, P.K. Chrysanthis and grounded on nestedtransaction model .PRO-MOTION issaidtobeamobiletransactionprocessing system which supports disconnected transactionprocessinginamobileclient-serverenvironment. It aimtomigrateexistingdatabaseapplicationsandsupportingthedevelopment ofnewdatabaseapplicationsinvolvingmobileandwirelessdataaccess. Nestedsplittransactionsare anexample of opennesting,whichrelaxes thetop-levelatomicity restrictionofclosednestedtransactionswhereanopennested transactionallowsitspartialresultstobeobservedoutside the transaction.PRO-MOTIONconsiderstheentiremobilesub-systemas oneextremely large,long-livedtransactionwhichexecutesattheserverwitha sub-transactionexecuting ateachMH.EachoftheseMHsub-transactions,inturn, is the root ofanothernested-split transaction. It is statedthat,bymakingtheresultsofa transactionvisibleassoonastransactionbeginstocommitatthe MH,itcanprovide additional transactions to progress even though the data items involved have been modifiedbyanactive(i.e.non-committed)transaction.Inthisway,localvisibility and localcommitment canreducetheblockingoftransactions duringdisconnection and minimizetheprobabilityof cascadingaborts. Theaccomplishedtask of sub- transactionsat mobilehost is confirmedby the conceptof compact objects.

<table>
<tr><td colspan="2">Methods commontoall objects</td><td colspan="2">Type specific methods</td></tr>
<tr><td>Obligation</td><td>Data</td><td colspan="2">Consistency rules</td></tr>
<tr><td colspan="4">State Information</td></tr>
</table>

Figure 3: Compactsasobjects

7.6 Moflex transactionmodel (MTM):

TheMoflextransactionmodelisanextension ofthe Flextransactionmodelto supportmobiletransactions. The Moflex display is based over multi-database frameworks and in view of the ideas of split-join exchanges. Themobiletransactionmanagers make use ofthetwo- phasecommitprotocol to coordinatethecommitment ofthe Moflextransaction.TheMoflextransactioncommitswhen itssub-transactionsthataremanaged byMTM have reached one oftheacceptablegoalstates, otherwise itisaborted. A compensable sub-exchange is privately dedicated, and the outcomes are made unmistakable to different exchanges.For non-compensable sub-transactions,the lastmobiletransaction manger, whichcorrespondstotheendlocation ofthe mobile host,plays therole as the committingcoordinator.

Table1: StrengthsandWeaknessesofexistingmobiletransactionmodels

S.No.	Existing Models	Strength	Weakness
1.	Kangaroo transactionmodel	i.Supports Mobility. ii.SupportsDistributedExecution.	i. Itdoesnotguaranteeserializability. ii. Themobilehosts always maycommunicatewithmobilesupportstations. iii. Doesn't
2.	ReportingandCo-transactionmodel	i. ItexhibitsTransactionProperties. ii. SupportsDistributedExecution.	i. Doesn't mention the state of mobile hosts and doesn'tsupport mobility. ii. Delegators and delegatee transactions connectivity are required, doesn't support disconnections.
3.	Pro-motion transactionmodel	i. Durability is ensured through the support of compact objects this model supports disconnections.	i. No explicitly disconnection of mobility. ii. High –capacity resources are required in mobile host to support disconnections. iii. TheDTP is not supported in this model.
4.	Two-tiertransactionmodel	i.It supports disconnected transaction processing. ii. When disconnection occurs, tentative transactions are locally carried out based on replicated versions of data objects.	i. Two-tier transaction model does not support the mobility of transactions. ii. Among Transaction Properties, Isolation is not achieved iii. Tentative transactions locally commit at the mobile host on replicated copies, and the committed results are made visible to other
5.	Multi-databasetransaction model	i. Supports database systems. ii. Helps in Distributed Execution.	i. If the transactions is not submitted to coordinating agents then there is no execution in the database. ii. Mobile host always tend to communicate

8. PURPOSE OF THE MOFLEX TRANSACTION MODEL

This model is the extension of flex model which is built upon multiple databasesystems andit is on the basic of split-join concept.The maincharacteristic ofa Moflextransaction is the Subtransactions consists of compensableor non-compensableare initialized inmobilehost and are submitted to mobiletransactionmanager(MTM)thatresidesat mobile support station. The MTM sendsthese sub-transactionstothelocalexecutionmonitor(LEM) at local databasesystemsfor executing. Figure 4presents the architecture of Mobility transaction model.The advance mobility transaction model has the capacity to manage and support mobility, heterogeneity and flexible in defining and executing of mobile transactions. Hence, the Moflex transaction model can be suitable for mobile heterogeneous multi-database systems.

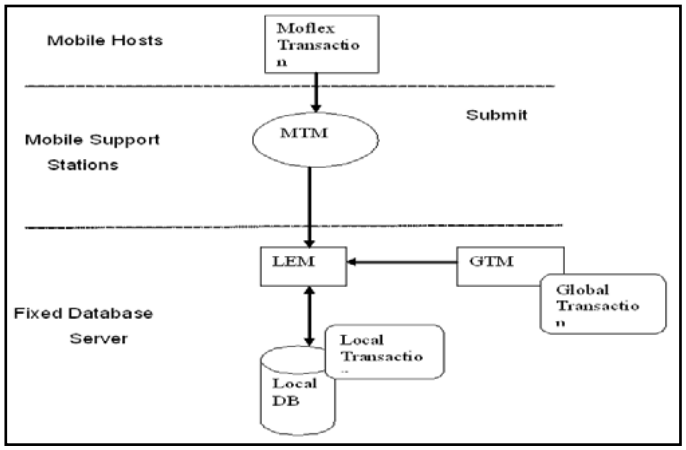


Figure 4-Moflextransactionmodel

9. CONCLUSION

This model supports transactions and ensures ACID properties of distributed transaction system. We have used android platform to implement the transaction in mobility with the help of MTM. In this technique we increase the transaction success probability, this by consequence, raises the performance of the system. When compared to other transaction models which has

the weakness in transaction of data Moflex model reduces the mobility issues when there is change of location to transfer the data.

10. REFERENCES

[1] M.J. Carey and M.Livny Conflict Detection Tradeoffs for Replicate Data ACM Transactions on DatabaseSystems(TODS).

[2] K.Ramamritham and P.K.Chrtsanthis.Advances in Concurrency Control and Transaction Processing.

[3] V.K.Murthy:Sealess Mobile Transaction Prcessing Models,Protocols And Software Tools,International Conference n Parallel and Distribution Systems(ICPADS).

[4] R.A.Dirkze and L.Gruenwald:A prre-serialization transaction management technique for mobile multidatabases.Mobile Networks and Applications(MONET).

[5] Q.Lu and M.Satyanarayana Improving Data Consistency in Mobile Computing Using Isolation-only Transaction. In IEEE HatOS Topics Workshop.

[6] E.Y.M.Chan,V.C.S.Lee and K.-W.Lam:Using Separate Processing fr Read-only Transactions in Mobile Environment,International Conference on Mobile Data Management.

[7] Chang and D.Curit: An Approach to Disconnected Operation in an object-oriented.

[8] R.A.Dirckze and L.Gruenwald: A pre-serialization transaction management technique for mobile multidatabases,Mobile Networks and Applications(MONET).

[9] Y.J.Al-HoumailyandP.K.Chrysanthis:1-2PC:the one-twophaseatomic Commitprotocol, ACM SymposiumonAppliedComputing(SAC) .