International Journal of Instrumentation Control and Automation

Volume 1 | Issue 2 Article 6

July 2011

Review of Some Transaction Models used in Mobile Databases

Prasanta Kumar Panda
Dept of CSE, NMIT, Bhubaneswar, PrasantakumarPanda@gmail.com

Sujata Swain
Dept of MCA, RIT, Rourkela, SujataSwain@gmail.com

P. K. Pattnaik
KIST,Bhubaneswar, P.K.Pattnaik@gmail.com

Recommended Citation
Panda, Prasanta Kumar; Swain, Sujata; and Pattnaik, P. K. (2011) "Review of Some Transaction Models used in Mobile Databases," *International Journal of Instrumentation Control and Automation*: Vol. 1 : Iss. 2 , Article 6.
DOI: 10.47893/IJICA.2011.1019
Available at: https://www.interscience.in/ijica/vol1/iss2/6

Review of Some Transaction Models used in Mobile Databases

Prasanta Kumar Panda
Lecturer
Dept of CSE
NMIT , Bhubaneswar

Sujata Swain
Lecturer
Dept of MCA
RIT, Rourkela

P. K. Pattnaik
Professor
Dept of CSE
KIST,Bhubaneswar

Abstract:

Mobile computing is presently experiencing a period of unprecedented growth with the convergence of communication and computing capabilities of mobile phones and personal digital assistant. However, mobile computing presents many inherent problems that lead to poor network connectivity. To overcome poor connectivity and reduce cost, mobile clients are forced to operate in disconnected and partially connected modes. One of the main goals of mobile data access is to reach the ubiquity inherent to the mobile systems: to access information regardless of time and place. Due to mobile systems restrictions such as, for instance, limited memory and narrow bandwidth, it is only natural that researchers expend efforts to soothe such issues. This work approaches the issues regarding the cache management in mobile databases, with emphasis in techniques to reduce cache faults while the mobile device is either connected, or with a narrow bandwidth, or disconnected at all. Thus, it is expected improve data availability while a disconnection. Here in the paper, we try to describe various mobile transaction models, focusing on versatile data sharing mechanisms in volatile mobile environments.

Key Words: Transaction, compensatable, Isolation

I. Introduction

The database which performs operation can be collected together to create a unit of carrying out instructions is called transaction. A transaction is nothing but a legitimate implementation of database operation [4]. A transaction transformed the database or related information from one coherent uniform state to another state. A transaction starts from creating a coherent state of database [2]. Then the database operation is started executing, after completion of operation a new logical state is connected. Each transaction is completed in successful manner and saved in the database [8]. If any fault is found during successfully completion, then at the beginning state re-connection performed by terminating the previous connection. This operation doesn't affect other transaction state, it is concurrently executed. In transaction either read, write or both operation can be performed. A transaction must end with validness and check weather the successfully completed or not. In mobile environment hosts are endlessly moved from one location to another location. So mobile transactions obtain or retrieve information from a storage device either in connected or disconnected mode [14]. As the location changes continuously by mobile host so bandwidth is not fixed in wireless network. If bandwidth is highly available then information can easily accessed which is called strong connection mode and if bandwidth is low then it is weak connection mode [21]. Depending upon the bandwidth mobile transaction can switches from strong to weak connection mode, so changing to lower state very common in wireless network.

Here we review some existing selected transaction models that have the quality of being able to perform with an efficient manner to support mobile transaction management.

A. Report and Co-transaction model

This model is proposed by P.K.Chrysan this and grounded as a context of specific multi database system (MDBS).This model considered as a collection of sub transaction either nested or open nested transaction model [17]. Nested transaction is a parent transaction makes child transaction supports more the quality of being adaptable than atomic transaction. It doesn't share the result between parent and child transaction while transactions are executed [21]. It allows hierarchy of transaction nesting levels and obeys the bottom-up approach by the root i.e. when a child transaction successfully executed, the object changed by it can be easily obtained to its parent transactions. The consequence of object made lasting in a database only when the parent transaction (root) successfully executed [10]. This model arranges the mobile transaction into following four types:

- Atomic transactions

It is related with substantial events like Begin, Commit, and Abort having the normal aborts & commit properties.

- Non-compensatable transactions

It is not linked with compensating transaction. It can execute at any time and the parents of these transactions have the responsibility to commit and abort [6].

- 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 [15]. 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.

B. Kangaroo transaction model

This model is proposed by Dunham and made to perform to represent the movement behaviour and data behaviour of transaction when a mobile host changing the position from one mobile cell to another in static network. It is named so because in mobile environment hop transaction move one base station to another [19]. This transaction model develops and grows based on abstract idea of global and split transaction in multi database environment. In this model Data Access Agent (DAA) at each base station used for accessing local and global databases. DAA accepts transactions express to need from a mobile user, and forwards the request to the corresponding database servers [5]. These transactions will be committed on servers. DAA acts as a Mobile Transaction Manager and data access coordinator.

For each transaction request, DAA produce a Kangaroo transaction and make the first set of Local transaction and global transaction. This set of Local and global transaction is called a Joey transaction. The execution of a Kangaroo sub-transaction in each mobile cell is supported by a Joey transaction that control in the scope of the mobile support station[5]. The Joey transaction performs to require of a proxy transaction to approve of the execution of the sub transaction of the Kangaroo transaction in the mobile cell. To accomplishing a task of the Joey transaction is maintained by the Data Access Agents that perform the mobile transaction manager at the mobile support station.

A Kangaroo transaction has a unique identification number composed of the base station number and unique sequence number within that base station [13]. When the mobile unit changes location from one to another, the control of the Kangaroo transaction changes to a new DAA at another base station. The DAA at the new base station produce a new Joey transaction [16].

a. Clustering model

This model is proposed by Pitoura and accepts a fully distributed system and considered as an open nested transaction model. This model is grounded on collection of related to meaning or nearly placed data together to form a cluster.

Clusters can be characterized statically or dynamically [12]. Each cluster composed of reciprocally consistent data. The level of consistency may changes calculating on the accessibility of network bandwidth among clusters [11]. This will provide applications with the potentially changes to suit to the currently available bandwidth, allowing for the user with data of variable level of quality.

Transaction from a mobile host composed of a set of weak and strict transactions grounded on the consistency requirement [18]. Weak transaction consists only weak read and weak write operations which can access only within the clusters. Strict transaction consists only strict read and strict write operation which can access all the clusters [19]. Weak transactions are committed in within clusters then committed among the clusters.

b. Isolation –only model

This model is proposed by Satyanarayan and used in Coda file system. Coda is a distributed file system by using file hoarding and concurrency control for mobile clients which provides disconnected operations [11]. Isolation only transaction covers read/write conflicts only within the service but it can only take value or importance in write/write conflicts. Here transactions are chronological succession of file accessing operations. Like Clustering, transactions are arranged in two categories:

i) First class which doesn't hold any separate section file accesses

ii) Second class which are carried out under disconnection.

First class transaction perform to act without delay after being executed, whereas Second class on one occasion goes to a pending state and waits for validation. When reconnection becomes possible second class transactions are made legally valid according to the wanted consistency criteria [14]. If validation is successful, results are integrated and committed otherwise transactions entering the resolution state.

A procurator asserts the log information throughout disconnection and informs the information on reconnection. The system hold highly assign to priority files in the cache using hoarding techniques. At a regular time interval procurator assures the priority of files and consistency with the server [11]. Altered files are again brought to the cache in order to check the consistency of transactions.

C. Two-Tier transaction model

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. Like Clustering and Isolation only transaction, 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 [23]. 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 [20].

D. Multi database transaction model

This model is grounded on a framework to adopt as a belief on transaction submission form mobile hosts in a multi database environment. Call for messages from a mobile host to its coordinating site is dealt asynchronously allowing for the mobile host to unplug it [15]. The coordinating node carry out the messages on behalf of the mobile unit and it is possible to query the position of the global transaction from mobile hosts. In the aimed Message and Queuing Facility (MQF), for each mobile work station there exists a message queue and a transaction queue [11]. Called for, reference and information type messages such as, called for connection/reconnection, reference for connection/reconnection mobile workstation, expect message queue position can be used. To deal the transactions presented, a simple global transaction queuing mechanism is suggested [17]. This comes near in time is grounded on the finite state machine concept. Set of possible state and transition can be distinctly determined between the beginning and ending state of the global transaction. For the execution of this mechanism five transaction sub-queues are used. These are input queue, allocate queue, active queue, suspend queue, output queue [7]. These are used to manage global transactions/sub transactions presented to local site by the mobile workstations.

E. Pro-motion transaction model

This model is proposed by G.D.Walborn, P.K. Chrysanthis and grounded on nested transaction model [1]. The Pro-motion model specially emphasize on supporting disunited transaction processing grounded on the client-server architecture. Mobile transactions are conceived as long and nested transactions where top level transaction is executed at fixed hosts, and sub transactions are accomplished at mobile hosts[7]. The accomplished task of sub-transactions at mobile host is confirmed by the concept of compact objects

<table>
<tr><td colspan="2">Methods common to all objects</td><td colspan="2">Type specific method</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>

Compact as object

Compacts are brought in as the introductory unit caching and control. Object Semantics is applied in the structure of compacts to better liberty and to increase concurrency [3]. A compact collected together required information to deal it. Pro-motion conceive the total mobile system as one highly large long-lived transaction accomplished on the server. Resources required to make compacts are found by this transaction through usual database operations. Compact structure is the duty of the compact manager at the data base server. Compacts are handled by the compact agent which is like to cache management daemon in coda file system, covered disconnection and handled storage on a mobile host [4]. Marked by dissimilarity Coda daemon, the compact agent behave as a transaction manager for transactions carried out on the mobile host, which successively responsible from concurrency control, logging, recovery. This model affirms disconnected transaction processing thru the back of compact object. When the mobile host is disunited from the fixed database, the sub-transactions are burst and carried out at the mobile host.

Disconnected transaction processing is a dominant transaction processing in Pro-motion [12]. So Pro-motion model demands high content mobile resources at the mobile hosts.

F. Toggle transaction model

This model is proposed by Dirckze and Grunewald and alike multi database transaction model. In this model a Mobile Multi database system is determined as a assembling of set and mobile databases [16]. Mobile Multi database management system is the software which occupies on a determined network and operates several database systems. A global transaction is determined as comprising of a set of operations, each of which is a legal operation consented by some service interface. Any subset of operations of a global transaction that right to enter the same site may be executed and will figure logical unit called a site-transaction. Site-transactions are executed below the assurance of the respective DBMS [10]. As mobile users change location to a new location of another Mobile Support Station (MSS), operations of a global transaction may be presented from different MSSs. Such transactions are referred to as migrating transactions.

In the model global transaction manager is planned to comprise of two layers: Global Coordinator layer and Site Manager layer. Global Coordinator layer comprise of Global Transaction Coordinators (GTCs) in each MSS and supervises including every performance and migration of global transactions [14]. The Site Manager layer comprise of Site Transaction Managers (STMs) in taking part database sites and handles the execution of vital or non-vital site-transactions. Each global transaction is determined to have a data structure that hold the current execution position of that transaction, and comply the user in migration from MSS to MSS. In this model, said that, concurrency is fixed as all site-transactions that fulfill at each site are pulled to engagement with each other [18]. The artificial engagement bring to existence by the algorithm will be terminate by working semantic information of site transactions. Each service interface will involve to furnish engagement information on all operations consented by that site [11]. This information will be applied to generate engagement between site-transactions.

a.Comparisons of transaction models

Model	Atomicity	Consistency	Isolation	Durability	Execute in	Transaction type	Operation mode
Report & Co-transaction model	Yes	Yes	Yes	yes	Mobile unit or fixed network	Global transactions and sub transactions	Connected,
Kangaroo transaction model	May be	NO	NO	NO	fixed network	Coordination and transaction execution	Movement in connected mode
Clustering model	NO	NO	NO	NO	Mobile unit or fixed network	Strict & Weak transaction	Connected, weak connected, disconnected
Isolation only transaction model	No	No	No	Yes	Mobile unit or fixed network	Validation and resolution of second class transactions	Connected, disconnected
Two-tier transaction model	No	No	No	No	Mobile unit or fixed network	Base transaction	Connected, disconnected
Multi database transaction model	No	NO	NO	NO	Mobile Unit or fixed network	Coordination and execution of multitrans-actions	Movement in connected, disconnected mode
Promotion model	Yes	Yes	Yes	Yes	Mobile unit or fixed network	compact construction, commit of locally committed transactions	Connected, disconnected
Toggle transaction model	Yes	Yes	Yes	Yes	Mobile unit or fixed network	site-transaction, migrating transaction	connected

CONCLUSION:

Mobile transaction originates and ends at same site. The implication of the movement of such transaction is that classical atoicity, concurrency and recovery solutions must be revisited to capture the whole behavior. As an effort in the direction we analyzed variety of transaction models and compared them in order to reveal the similarities and dissimilarities.

References

1. ***Y. J. Al-Houmaily and P. K. Chrysanthis: 1-2PC: the one-two phase atomic Commit protocol, ACM Symposium on Applied Computing (SAC),***
2. ***Barbara: Mobile Computing and Databases - A Survey, IEEE Transactions on Knowledge and Data Engineering (TKDE),***
3. ***S. Chang and D. Curtis: An Approach to Disconnected Operation in an Object-Oriented Database, International Conference on Mobile Data Management (MDM).***
4. ***P. K. Chrysanthis: Transaction Processing in Mobile Computing Environment, IEEE Workshop on Advances in Parallel and Distributed Systems.***
5. ***E. Y. M. Chan, V. C. S. Lee and K.-W. Lam: Using Separate Processing for Read-Only Transactions in Mobile Environment, International Conference on Mobile Data Management (MDM).***
6. ***R. A. Dirckze and L. Gruenwald: A pre-serialization transaction management technique for mobile multidatabases, Mobile Networks and Applications (MONET).***
7. ***J. Holliday, D. Agrawal and A. E. Abbadi: Disconnection Modes for Mobile Databases, Wireless Networks.***
8. ***S. K. Madria and B. K. Bhargava: A Transaction Model to Improve Data Availability in Mobile Computing, Distributed and Parallel Databases.***
9. ***S. K. Madria and B. K. Bhargava: A Transaction Model for Mobile Computing, International Database Engineering and Application Symposium (IDEAS).***
10. ***V. K. Murthy: Seamless Mobile Transaction Processing: Models, Protocols And Software Tools, International Conference on Parallel and Distributed Systems (ICPADS).***
11. ***K. Ramamritham and P. Chrysanthis: Advances in Concurrency Control andTransaction Processing, IEEE Computer Society Press***
12. ***R.A.Dirckze, L.Gruenwald, "A Toggle Transaction Management Technique for Mobile Multidatabases", Proceedings of the CIKM.***
13. ***M.H.Dunham, A.Helal, S.Balakrishnan, "A Mobile Transaction Model that Captures Both the Data and Movement Behavior", ACM-Baltzer Journal on Mobile Networks and Applications (MONET).***
14. ***S.K.Madria, B.Bhargava, "A Transaction Model to Improve Data Availability in Mobile Computing", conditionally accepted to Distributed and Parallel Databases,An International Journal, Kluwer Publishers.***
15. ***G.D.Walborn, P.K.Chrysanthis, "Transaction Processing in PRO-MOTION" , Proceeding of the 1999 ACM Symposium on Applied Computing, SAC .***
16. ***Bobineau, P. Pucheral, and M. Abdallah. A Unilateral Commit Protocol forMobile and Disconnected Computing. In Int. Conf. Parallel and Distributed Computing Systems (PDCS).***
17. ***M. J. Carey and M. Livny. Conflict Detection Tradeoffs for Replicate Data. ACM Transactions on Database Systems (TODS).***
18. ***P. K. Chrysanthis and K. Ramamritham. Synthesis of Extended Transaction Models Using ACTA. ACM Transactions on Database Systems (TODS).***
19. ***K. Elmagarmid. Database Transaction Models for Advanced Applications.***
20. ***Q. Lu and M. Satynarayanan. Improving Data Consistency in Mobile Computing Using Isolation-Only Transactions. In IEEE HotOS Topics Workshop.***
21. ***K. Ramamritham and P. K. Chrysanthis. Advances in Concurrency Control and Transaction Processing. IEEE Computer Society Press.***
22. ***K. Ku and Y. Kim. Moflex Transaction Model for Mobile Heterogeneous Multidatabase Systems. In IEEE Workshop on Research Issues in Data Engineering.***