Evolving a Surrogate Model of Transaction Management for Mobile Cloud

Ravimaran Shanmugam, M. A. Maluk Mohamed, and S. Nazreen

Abstract—**Mobile cloud computing with handheld devices had resulted in reducing the infrastructure and administrative costs and creating more efficient, flexible and collaborative computing models for business growth. However, Transaction within a mobile cloud faces many restrictions. Building a reliable and fault tolerant transactional data management system for managing large volume of data is one of the major challenges in mobile cloud. Multiple researches are in progress to define the new models of success in mobile cloud transaction. In this paper, we propose Surrogate Object Model of Transaction Management for mobile cloud. Each mobile node in the cloud sends transaction request to its surrogate object located in its respective mobile support station. Surrogate object should check its data cache for transaction execution with a given reliability instead of sending request to database server such that the network lifetime is improved. Reliability and fault tolerance are achieved by moving transaction request to surrogate object and network life time is maximized by moving the surrogate objects to less loaded base stations. We show via simulations that the proposed approach achieve the better reliability and fault tolerance and also maximize the network lifetime compared to existing approaches.**

Index Terms—**Data Cache, Mobile Cloud, Surrogate Object.**

I. Introduction

The popularity of cloud computing and its potential to transform service delivery has led a growing number of businesses to implement cloud computing for their transactional process, while many others are actively considering it. In the last couple of years, there is a very big paradigm shift in business models, with mobile becoming an important component in cloud computing. As per the recent survey, the size of the mobile cloud is increased to $45 billion and transactions process performed over the cloud is to $25 billion by 2016. This new paradigm poses several constraints in traditional transaction management system. There are different forms of risks in storing sensitive transactional information such as customer data, account details and credit card numbers on an un-trusted host. Thus the existing transaction models for cloud computing are not well suited for mobile cloud due to the differences in cost and performance. Due to latency problem, most of the existing transaction processing models are unacceptable for most of the mobile cloud applications. The different research initiatives are going on to revise mobile transaction management to be adapted to the growing mobile cloud computing paradigm. Here, we propose a Surrogate Object Model of Transaction Management for efficient support to the mobile transaction management over the mobile cloud with minimum latency, computing, cost and energy resources. With surrogate object [1][2][3], the latency issues associated with running transaction processing application over the mobile cloud is solved by maintaining a cache of data stored in the mobile host and reduce the wireless data transfers. It also solves the failure and location management problem associated with the transaction processing due to mobility of hosts. The proposed transaction processing system in the mobile cloud is more affordable for a lot more businesses and for different end customers. The remaining of this paper is organized as follows: In section 2, we have surveyed some related work which highlights several observations that motivated our work. In section 3, we introduce our proposed surrogate object model. In section 4, we simulate the proposed model and evaluate its performance. Conclusion and future work are drawn in section 5.

Manuscript received April 20, 2012; revised May 15, 2012.

The authors are with the Department of Computer Science and Engg., M.A.M. College of Engineering, Anna University Tiruchirappalli, India (e-mail: ssg_ravimaran@mamce.org, ssg_malukmd@mamce.org and ssg_nazreen@mamce.org)

II. Related Work

There are very minimum number of current efforts to integrate mobile devices into the cloud for data and transaction management. Most of the existing efforts focus on considering data management on traditional relational database in traditional cloud environments, but not focused for mobile clouds. Most of these efforts are highly supporting the data management on the cloud, but do not support for better serializable transaction and data management over the mobile cloud.

Jing Zhao and Xiangmei Hu [4] studies the performance of query processing on structured data and proposed MapReduce techniques. In this technique, a user query is divided into subqueries . With the help of replicas in cloud, each sub query is mapped to k+1 subqueries and each one has to wait in the queue of the slave where the query data is stored. The technique also adopts two different scheduling strategies to dispatch the subquery and introduce the pipeline strategy to reduce the client's long waiting time.

Adrian Daniel Popescu [5] proposed an adaptive software model which can effortlessly switch between MapReduce and parallel database in order to efficiently execute queries regardless of their response times. He presented a transparent frame work for switching between the two architectures based on an intuitive cost model, which computes the expected execution time in presence of failures and achieves lowest query execution time. In [6], several benchmarks for measuring the relative performance of different cloud based data management systems have been proposed which supports further research and development of cloud-based data management systems.

Ooi Beng Chin [7] presented the opportunities and challenges of developing a scalable Cloud data management system. The objective of this scalable data management system is to examine the anatomy of a Cloud data intensive system which provide multi-tenancy architecture, high throughout low latency transactions, and high performance reliable query processing on cloud platform. Raghu Ramakrishnan [8] proposed technical challenges involved in designing hosted, multi-tenanted data management system which focused on efficient cost management,averaging usage peaks and enabling higher-level services. Qiming Chen [9] proposed cycle-based query model for data stream analysis as cloud service for mobile applications. The model allows a SQL query to run cycle by cycle for processing the unbounded stream data chunk by chunk and deployed a new infrastrcture with multi-engines without centralized coordination and physical movement and copying of data. Yogesh Simmhan [10] investigated the building of scientific workflow for datamanagement in the cloud which provide users with insights on the impacts of different implemention approaches on the performance. The paper also investigates how to explore the relative performance of different implementation approaches of the cloud including storage architecture, data models, tradeoffs inconsistency and availabilty.

In paper [11], Xiangu Zhang presented an efficient method to construct multi-dimensional index for Cloud computing system. The method uses the combination of R-tree and KD-tree to organize data records and offer fast query processing and efficient index maintenance. The paper also proposed a cost estimation-based index update strategy that can effectively update the index structure which provides improved query efficiency and better scaling with the size of the data. Bogdan Nicolae [12] presented a novel idea for designing highly scalable distributed storage systems that are optimized for heavy data access concurrency and also proposed different algorithms for data and meta data that enables a high throughput under concurrency. In paper[13], Michael Grossniklaus addressed the new data management requirements for cloud systems especially for object data bases. The full potential of cloud computing data management can only be leveraged by exploiting object database technologies which provide effective platform to model and implement data partitions, while, at the same time, helping to reduce joint processing. The paper [14-15] describes a different works on the cloud platform for distributed database processing and transaction management. As the volume of data processed by the cloud applications increases, the need for building an efficient data management system emerged as a crucial requirement. Due to rapid expansion and massive usage of smart phone, deploying the mobile devices such as smart phone into the cloud platform (say mobile cloud) for large scale distributed transaction data management processing is also an important factor. Due to various limitations with mobile devices and wireless connection, the data and transaction management on the mobile cloud caused a different problem such as delay in transaction execution, failure of transaction, low reliability and no guarantee for completing the transaction execution. In this work we extend the current distributed mobile transaction processing system to cloud environment for providing low latency and less-cost platform to handle high volumes of the transactions. This platform is significantly differentiated from the current transaction processing systems which are in general built separately from the database server and suffer from the overhead of mobility and data access over the different environments. The main contribution of this paper is to support sharing of data and processing of data among different transactions at differnet mobile devices in cloud environment.This is achieved by inroducing the surrogate object over the cloud to act on behalf of each mobile device which can autonomically perform local caching for providing faster, less expensive, safer and more accessible mobile tranaction system data access. This model can handle high volumes of transactions in mobile cloud with enhanced responsiveness and without frequent physical data moving and copying.

III. Surrogate Object Model and Assumptions

We consider an architecture with surrogate objects, mobile nodes [handheld devices], static nodes, and a static mobile support stations. Surrogate objects are autonomous software entities that have the capacity to adapt to changing environmental conditions in the mobile cloud and to reallocate themselves among the active mobile nodes to carry out their transaction requests. The model carries out the transaction execution with the support of surrogate object and deals the mobility of transactions across the mobile nodes and yields better latency and enhanced mobility support. Fig. 1 depicts the model considered for our proposed work.

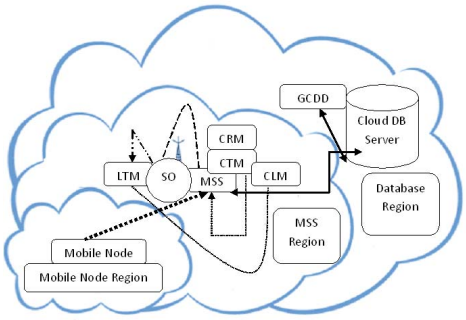

Fig 1. Cloud transaction model using surrogate object

The following are the assumptions in our model.

1) Mobile nodes are periodically sending the transaction request to their mobile support station via wireless communication.
2) Each Mobile support station (MSS) has a group of surrogate objects for their respective mobile nodes.
3) All the mobile nodes have movement capabilities and are less powerful than static nodes in terms of battery energy, communication, computation.
4) Mobile support station received the transaction request from the individual mobile nodes and forward the request to the respective surrogate object if exist.
5) The mobile support station is static throughout the transaction process.
6) The mobile nodes, surrogate objects, static nodes and

mobile support stations are considered to be the part of cloud.

7) The surrogate object can cache mobile host specific data and reduce the response times for many transactions and solve the latency issues.
8) The surrogate object can support disconnected operations of the MHs by buffering client requests or using the cached data to handle them.
9) The surrogate object resides on the wired network is free to move.
10) Cloud Transaction Manager [CTM] at each MSS is responsible for all Transaction Execution initiated at MSS.
11) Global Cloud Data Distributor [GCDD] distributed across the cloud platform will support caching necessary data for transaction execution into surrogate object.
12) Cloud Reconcile Manager [CRM] at each MSS will support updating of database to database server after each transaction commits.
13) To yield proper load balancing, surrogate object is free to move independently of the MH anytime. In order to avoid heavy traffic and increased latency due to this object movement, always move the surrogate object to the MSS which is the source of maximum queries or to a less loaded MSS and this movement pattern are properly stored in Migration Table.

The initial startup phase of this model is given below.

1) Before submitting a transaction request, a mobile node must register its identity in the cloud through the respective MSS.
2) A Unique mobile node Id is allocated and cloud manager at MSS creates an object corresponding to the mobile node and assigns unique surrogate object identifier. Both ids are entered into the object table maintained at the same MSS.
3) When mobile node wants to issue the transaction request, a node must start a virtual machine [VM] in the cloud with the support of Cloud Transaction Manager [CTM]. The CTM receives the transaction request, is responsible for scheduling them [if it's long lived transactions] and coordinates execution and it allocates and de-allocates VM based on the current status of the transaction execution. This new transaction entry is updated into the Transaction table.
4) CTM at each MSS Sends data hoarding request to database server with necessary locking request and GCDD at database server sends the required data set as cache to different surrogate objects. The shared data are cached at the surrogate objects only if database server grants non-conflict lock. When a mobile node issues a transaction request in cloud environment and the data is cached to the surrogate object, then the transaction has immediate access to the data on surrogate object. So that the transaction can be executed very quickly with minimum latency than if it is stored just in the database server. By this way, the proposed model yields 35 % of reduction in server and infrastructure costs.

A. Phase I Execution

1) When mobile node submits the transaction request, the MSS to which it belongs, uses the naming services of the middleware through CTM and gets the object reference of the surrogate corresponding to that node. Using the object reference, CTM sends the transaction request to Local Transaction Manager [LTM] located at the surrogate object. A surrogate object receives one or more transaction request from the CTM at a time, executes them, and after that informs the CTM about their completion.
2) The lock requests on the surrogate object are handled with the support of Cloud Lock Manager [CLM] at the respective MSS. The transaction must issue a lock request to CLM before performing a read or write action on surrogate object. This is the case if there is a cache hit at the surrogate object. If there is a cache miss, then transaction needed to be executed as mentioned in Phase II.
3) Each Transaction is required to lock surrogate object before any unlock. This guarantees the consistency on the cache of the surrogate object.
4) When a transaction is committed at the surrogate object, the LTM provides a logging service to ensure that the committed transaction result will not be lost and ensures transaction integration [if it is long lived transaction].
5) The complete algorithm is described in the next section which is executed after submitting every transaction request.

Get Transaction Request ( )

input : *Tran_Details and Mob_id*

process :

Send Transaction request along with mob_id & Tran_Details to MSS;

CTM assigns unique transaction id : tran_id;

Refer Object Table and Fetch the corresponding Surrogate object id : sid;

Call FindDataCache();

output : *tran_id, sid, mob_id updated in tran_table and initialize VM for transaction*

FindDataCache ( )

input : *tran_id, sid*

process:

For each transaction tran_id
Check the existence of surrogate object in the respective MSS;

If surrogate object exist in the respective MSS
Transaction request is forwarded to surrogate object along with tran_id;
Else
Refer Object Migration table and forward the request to the corresponding MSS;

When surrogate receives Transaction request, it checks its cache for required data item

If DataCache exist

transaction request to LTM

Else

transaction request forwarded back to CTM
CTM sends a request to the location server (which is located in the static portion of the network)
Get the object reference and its current location and forwarded to CTM.

Checks for the existence of replicas and required data sets in the above said locations;

CTM need to wait for an acknowledgment; gets the required dataset and forward the request either to corresponding LTM or to CTM;

Call CloudTran_Execution();
Output: *required dataset for transaction execution*

CloudTran_Execution( )
input: *tran_id, sid, DataCache,*

process:
LTM send lock request to CLM

LTM uses appropriate Lock migration strategy to maintain Cache consistency and to support concurrent transaction execution;

If lock request (either Read or Write) is granted,

Surrogate object performs transaction execution on cached data items;

After execution, cache is updated and transaction status is updated as "committed" in the transaction table;

Cloud Reconcile Manager [CRM] at each MSS updating database server after each transaction commit;

Call Tran_ACK();
Output : *result of transaction submitted by mobile node*

CloudTran_ACK( )
Input : *tran_id, result ;*

Process:
A result of the transaction is forwarded to respective surrogate object;
If all subtractions (if submitted transaction is long lived transaction) are committed
CTM forwards the result of the transaction to the mobile device;
Sends an acknowledgement of the receipts of the transaction results to CTM;

CTM removes transaction entry from the transaction table;
If the mobile device is not in their communication region (due to mobility)
Retains the result of the transaction in the respective surrogate object;

Else

Surrogate object takes opportune time to deliver the result of the transaction to its mobile device when it reconnects to its mobile support station.

Output: *Result delivered to mobile node with ACK*

B. Phase II Execution

If DataCache Missed

CTM issues further transaction requests either to the various mobile devices present nearby MSS or to other surrogate objects; Checks for the existence of replicas and required data sets in the above said locations;
CTM needs to wait for an acknowledgement; get the required dataset and forward the request either to corresponding LTM or to CTM;

Call CloudTran_Execution();

If required datasets are not found in any of the above said locations

Transaction request forwarded directly to the Database server and executed in the server

IV. Performance Evaluation

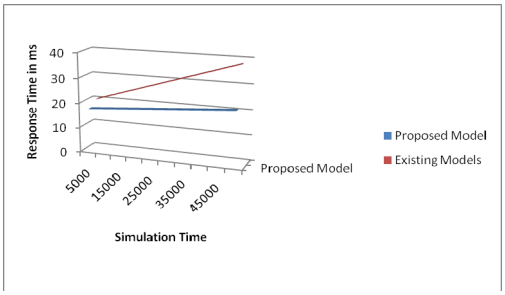

Fig. 2. Average latency Vs simulation times

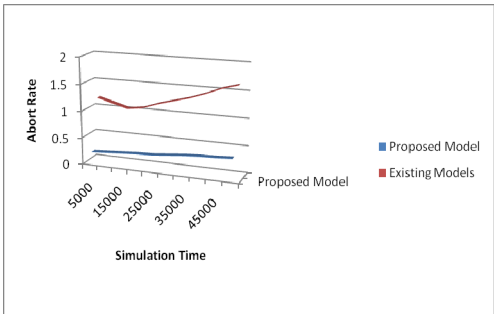

Fig. 3. Average abort rate VS simulation times

We measured the latency and abort rate of transactions in order to show the improvement in the network life time and fault tolerance in our simulations. We compare our approach with the existing approaches which are implemented based on two of the closely related works [4],[15] and simulation shows that the lifetime improves with the support of surrogate objects and its cache. This is also leads to better bandwidth utilization and only 5% of abort (implies 95 % of Success achieved) because more data items are cached into the surrogate objects. Due to this, most of the time the transaction requests are sent to the surrogate objects instead of database server. Thus improve the network lifetime, ensures minimum abort rate and achieves minimum latency in transaction execution over the mobile cloud as shown in Fig. 2 and 3.

V. Conclusion

In this paper we have evolved a surrogate model of transaction for mobile cloud that overcomes the latency

problem found in several strategies proposed in the literature. The proposed strategy caches the data in the surrogate object which helps in reducing the average transaction. The required reliability and reduced abort rate are provided by sending the transaction request to surrogate object and network lifetime is maximized by migrating the surrogate objects in a way to avoid improper load balancing. Currently we are in the process of developing a simulation prototype to evaluate our method among mobile devices on the cloud and moving our method to execute concurrent transactions with different workloads to test its feasibility.

REFERNCES

[1] M. A. Maluk Mohamed, D. Janaki Ram, and M. Chakraborty, "Surrogate Object Model: A New Paradigm for Distributed Mobile Systems," in *Proceedings of the 4th International Conference on Information Systems Technology and its Applications (ISTA'2005)*, May 23-25, New Zealand, pp.124-138, 2005.

[2] S. Ravimaran and M. A. Maluk Mohamed, "An Improved Kangaroo Transaction Model using Surrogate Objects for Distributed Mobile System," in *Proceedings of MobiDE 2011 Tenth International ACM workshop on data Engineering and for wireless and Mobile Access*, June 12th 2011, Athens, Greece (In conjunction with SIGMOD/PODS 2011.

[3] S. Ravimaran and M. A. Maluk Mohamed, "Surrogate Object based Data Mining for Distributed Mobile System," in *Proceedings of MoMM2011_ERPAS, ACM 9th International Conference on advances in mobile computing and multimedia*, 2011, iiWAS2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam. Copyright 2011 ACM 978-1-4503-0784-0/11/12.

[4] J. Zhao, X. Hu, and X. Meng, "An Efficient SQL query processing for cloud data management," CloudDB'10 ACM proceedings of the second international workshop on cloud data management, doi:10.1145/1871929.1871931, 2010.

[5] A. D. Popescu, D. Dash, V. kantere, and A. Ailamaki, "Benchmarking cloud based data management systems," CloudDB'10 ACM proceedings of the second international workshop on cloud data management, 2010, doi:10.1145/1871929.1871933. 2010.

[6] Y. Shi, X. Meng, and J. Zhao, "Benchmarking cloud based data management systems," CloudDB'10 ACM proceedings of the second international workshop on cloud data management, 2010, doi:10.1145/1871929.1871938, 2010.

[7] O. B. Chin, "Cloud Data Management Systems: Opportunities and challenges," Fifth International IEEE conference on semantics, Knowledge and Grid, 2009, doi: 10.1109/SKG.2009.110, 2010.

[8] R. Ramakrishnan, "Data Management in the cloud", IEEE International conference on Data Engineering, 2009, doi: 10.1109/ICDE.2009.175, 2010.

[9] H. Q. Chen and M. Hsu, " Data Stream analytics as cloud service for mobile applications," *OTM'10 proceedings of the 2010 international conference on On the move to meaningful internet systems, Springer-Verlag Berlin*, Heidelberg 2010, ISBN:3-642-16948-1 978-3-642-16948-9, 2010.

[10] Y. Simmhan, R. Barga, C. V. Ingen, "On building Scientific Workflow Systems for Data Management in the Cloud," ESCIENE'08 IEEE *proceedings of the 2008 Fourth Conference on eScience*, doi: 10.1109/eScience.2008.150.

[11] X. Zhang, J. Ai, and Z. Wang, "An efficient multi-dimensional index for cloud data management," CloudDB'09, *proceeding of the ACM first international workshop on cloud data management*, 2009, ISBN: 978-1-60558-802-5 doi:10.1145/1651263.1651267, 2009.

[12] B. Nicolae, G. Antoniu, and Bouge, "BlobSeer: Next generation data management for large scale infrastructure," *Journal of Parallel and distributed computing*, vol. 71, no. 2, February 2011, doi: 10.1016/j.ipdc.2010.08.004, 2011.

[13] M. Grossniklaus, "The case for object databases in cloud data management", ICOODB'10 proceedings of the Third international conference on objects and databases, Springer-Verlag Berlin, Heidelberg, ISBN: 3-642-16091-3 978-3-642-16091-2, 2010.

[14] C. Gang, "Data Center Management Plan in Cloud Computing Environment," in *Proceedings of IEEE 3rd International conference on information management*, Innovation Management and industrial Engineering, 2010, doi: 10.1109/ICIII.2010.575, 2010.

[15] A. Beloglazov and R. Buyya, "Energy Efficient Resource management in virtualized cloud data centers," *in Proceedings of 2010 10th IEEE/ACM international conference on cluster*, cloud and grid computing, 2010, doi:10.1109/CCGRID.2010.46, 2010.

[16] D. J. Abadi, "Data management in the cloud: Limitations and Opportunities," *Bulletin of the IEEE computer society Technical committee ondata engineering*, 2009.

S. Ravimaran has received his B.E. in computer science and engineering from National Institute of Technology, Tiruchirapalli, India in 1997 and his M.E.computer science and engineering from Periyar Maniammai College of Technology, Vallam, Tanjore, Anna University, Chennai, Tamil Nadu, India in 2004.Since 2008, he has been a research scholar, pursuing a Ph.D at Anna University, Tiruchirappalli, Tamilnadu, India. Since January 1992 to December 1999, He was worked as a SENIOR FACULTY CUM HEAD in Academic Courses Techno Services Pvt Limited, Tiruchirappalli, Tamilnadu, India. From December 1999 he was worked as a LECTURER, SENIOR LECTURER, ASSISSTANT PROFESSOR and currently he is working as a Professor and HEAD, Department of Computer Science and Engineering, M.A.M. College of Engineering, Tiruchirappalli, Tamil Nadu, India. He has published papers such as A Transaction Processing System for Sharing Mobile Databases in Wireless Environment in National Conference on Recent Trends in Information Technology at RVS College of Engineering and Technology in March 2010.Another paper titled Performance Analysis in Data Replication in Distributed Data base Environment was Published in National Conference on Networking and Database NCND organized by PABCET, on 17 -18 Mar 2005. His area of interests is Distributed database, Mobile transaction, Grid Computing. Professor S.Ravimaran was a member of IEEE and IEEE Computer Society,ISTE, CSI and Institution of Engineers

M. A. Maluku Mohamed has received his BE in Electronics and Communication Engineering from Bharathidhasan University, Tiruchirapalli, India. in 1993 and his M.E. in Computer Science and Engineering from National Institute of Technology Tiruchirapalli, India. in 1995.He has done his Ph.D. in Communication and Computing Paradigms for Distributed Mobile Systems in from Indian Institute of Technology Chennai, India in 2006. The previous year book publication was Information Systems Reengineering for Modern Business Systems: ERP, SCM, CRM and E-Commerce Management Solutions at USA published by IGI-Global in 2011.He has published journal papers such as Handling Disconnections in LTE Network-based Mobile Grid in International Journal on Internet and Distributed Computing Systems, Vol. 1, No. 2, pp. 68-75 in 2011 and another journal paper named DEEPG: Dual Heap Overlay Resource Discovery Protocol for Mobile Grid from International Journal on Scalability Computing: Practice and Experience, Vol.12, No.2, pp.239-255 in 2011.His research interests are distributed computing, mobile computing, wireless sensor network, cluster computing and grid computing. Dr.M.A.Maluk Mohamed was a member of ACM, IEEE and IEEE computer society, IACSIT.He was awarded Vijay Rattan by India International Friendship Society, New Delhi in 2005 for specializing in science and Technology.

S. Nazreen has received her B.E in Computer Science and Engineering from M.A.M. college of Engineering, Anna University of Technology, Chennai, Tamil Nadu, India in 2009.Since July 2009 to June 2010 she worked as a LECTURER in Department of Information Technology. M.A.M College of engineering and Technology, Tiruchirappalli, Tamilnadu, India. Currently she is pursuing M.E in computer science and Engineering, M.A.M College of Engineering, Tiruchirappalli, Tamilnadu, India. Her research interests includes grid computing, mobile computing, Data structures, operating system.