Data Management in the Mobile Cloud Using Surrogate Object

Ravimaran Shanmugam and Maluk Mohamed M. A.

***Abstract*—Cloud computing had resulted in reducing the infrastructure and administrative costs and creating more efficient, flexible and collaborative computing models for business growth. However, managing the data over the cloud which includes mobile devices, namely the mobile cloud, faces several challenges. Building a consistent, secured and efficient transactional data management system for managing large volume of data is one of the major challenges in mobile cloud. In this paper, we propose a Surrogate Object based Cloud Caching (SOCC) mechanism which provides mobile device network to self heal by storing data needed for transactions across multiple surrogate objects over the cloud by analyzing customer requirements and context. The proposed SOCC mechanism ensures low latency, reliability and fault tolerance in transaction management over the mobile cloud. Extensive analysis shows that our proposed mechanism is highly efficient, provides better data management and effectively utilizing the underlying distributed data's over the mobile cloud.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

I. Introduction

Cloud computing is a new computing paradigm in which the computing resources are delivered as a utility like water, gas, and electricity in order to reducing the infrastructure and administrative cost of most of the business. Mobile cloud is the collection of mobile devices such as smart phones, lap tops and PDA's over the cloud extended by mobility and a new ad-hoc infrastructure. It provides mobile users with data storage and transaction processing services on a cloud computing platform. Mobile cloud computing devices use cellular connections to perform data storage and transaction processing services on a cloud computing platform. As per IDC(International Data Corporation)and Gartner estimation, the average number of consumer mobile devices used by enterprises employees is increasing day by day, estimated enterprise data growth in the next five years will be 600 % and percentage of businesses that will use cloud based services by 2012 will be 44 %. Managing and delivering an improved level of data management in this promising mobile cloud environment is very important. Although wireless cellular connectivity can be lost or restricted due to limitations in terms of energy, storage, power, lack of cellular coverage, the situation will be improved by constructing a reliable mobile cloud for an efficient data and transaction management.

In this paper, we propose an autonomic mobile cloud using surrogate object in which devices can self heal in response to limited cellular connectivity and uses distributed data caching for dynamic data support and transaction processing. The systems also provide effective and convenient data storage outside mobile devices. The surrogate object technique [1], [2], [3] for mobile cloud defines an architecture that allows mobile devices such as smart phones to participate seamlessly in data storage, transaction processing, computing and communication. The technique involves bridging the mobile device and cloud platform, using a placeholder namely the surrogate object. This is done by creating the surrogate object in the cloud platform to act on behalf of each mobile device. One consequence of using the surrogate object model is that mobile devices would be transparent to the instability of wireless communication. The surrogate object can remain active, maintaining information regarding the current state and plays an active role on behalf of the device and also act as the part of cloud. The surrogate object also acts as a data cache centre that can realize local caching for faster information access.

Introducing the surrogate object into cloud platform, can allow mobile devices to continue to perform data storage and transaction processing inspite of limited wireless connectivity by storing the application data on the cache of surrogate object. This approach can provide high throughput, minimum latency transactions, and high performance reliable transaction processing over cloud.

The rest of this paper is organized as follows: section 2 describes the related works and section 3 presents surrogate object based cloud caching mechanism among surrogate object and mobile devices in mobile cloud and describes the algorithm for the same. In section 4, the future scope of the proposed method is discussed and section 5 gives the conclusion to the paper.

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 don't support for better serializable transaction and data management over the mobile cloud.

Jing Zhao and Xiangmei Hu [4] study the performance of query processing on structured data and propose MapReduce

Manuscript received April 10, 2012; revised May 5, 2012.

The authors are with Software System Group, Department of Computer Science and Engg, M. A. M. College of Engineering, Anna University Tiruchirappalli,India (Corresponding author. Tel.: + 09443076222, fax: +914312650377, e-mail: ssg_ravimaran@mamce.org).

techniques. In this technique, a user query is divided into subqueries. With help of replicas in cloud, eachsub 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] propose 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] proposes technical challenges involved in designing hosted, multi-tenanted data management systems which focus on efficient cost management , averaging usage peaks and enabling higher-level services

Qiming Chen [9] proposes 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 deployes the 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 implemantation 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] present a novel idea for designing highly scalable distributed storage systems that are optimized for heavy data access concurrency and also proposed a different algorithms for data and meta data that enables a high throughput under concurrency.

In paper[13], Michael Grossniklaus address 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 join 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 emerges 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 causes a different problems such as delay in transaction execution, failure of transaction, low reliability and no guarantee for completing the transaction execution.

The main contribution of this paper is to support sharing of data and processing of data among different transctions 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 faster data access. The paper provides two main contributions to the data management in the mobile cloud : [1] provide proper support to execute the tansaction without help of database server(ensures Low latency) , [2] provide support to access data across nearby surrogate object in response to the limited cellular connectivity which allows mobile devices in cloud environment to access the data autonomically in order to provide optimal utilization of wireless bandwidth (ensures reliabilty and fault tolerance).

III. Surrogate Object Based Cloud Caching (Socc)

A. Model Description

In this section, we propose a new transaction model, called Autonomic Distributed Surrogate Object based Cloud Caching **(SOCC).** The surrogate object model for distributed mobile cloud systems define architecture **(Fig.1)** that allows mobile devices to participate seamlessly in transaction processing over the cloud. The fundamental idea behind this model is that it involves bridging the mobile devices and its supporting environment using the surrogate object. This is done by creating the surrogate object in the cloud platform to act on behalf of each mobile device. By this way, mobile users can continuously carry out the transaction processing without support from the database server. We aim to deal with autonomic transaction management when databases are used and deployed as service (DaaS) over the mobile cloud.

The surrogate object remain active, maintain information regarding the current state, storing application specific data on the cache, acts as data sink that can collect data form database server (under DaaS mode) and plays an active role on behalf of the device and reduce the need for data request by the multiple and concurrent transaction processing across the cloud network, thereby allowing devices to perform the transaction processing under the following circumstances: i) in the absence of database server, ii) when wireless connectivity becomes weaker iii) when wireless connectivity becomes extremely latent iv) when database server is located far away. The surrogate object is a software entity which contains information relevant to the mobile device and set of methods to act upon them, which is initially hosted on some mobile support station (MSS) and later its cache is replicated and move to multiple devices present within a predefined range of cellular regions and assume that all these devices are seamlessly participating in mobile cloud environment.

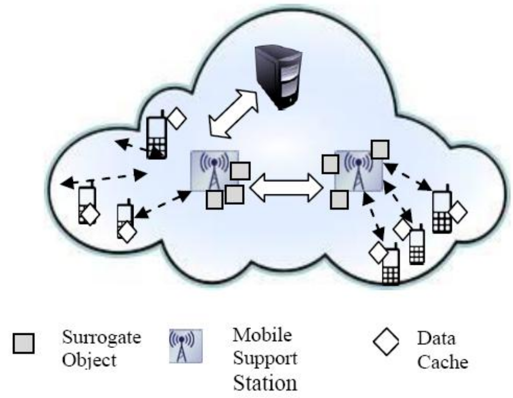

Fig. 1. Surrogate object based cloud caching

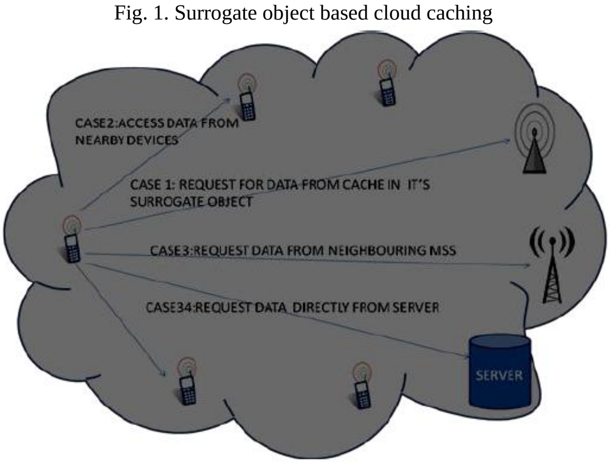

Fig. 2. SOCC abstract flow of operations

In general, when mobile user initiates the transaction request, the request is send either to other powerful mobile device or to powerful static device in the network, perform the remote computation and receive the resulting data from the recipients. This process leads to communication overhead and energy overhead. The proposed model helps in reducing the communication and energy overhead and also reducing the pricing for accessing the data resources with the help of DaaS scheme (Pay-as-You-Use Model). The process of transaction execution among mobile devices in mobile cloud is described in fig 1. In our model, whenever a mobile device newly joins the mobile cloud system, it registers itself with a mobile support station. The MSS assigns a unique identification for the mobile device (say MDID) and passes the information to underlying middleware in the same MSS about the entry of a new mobile device into the cloud. The middleware then creates an object (namely Surrogate Object) corresponding to the mobile device and assigns a unique id (say SOID) and adds both MDID and SOID entry in the catalogue register for future reference. Once the surrogate object is created in the respective MSS, it acts as data sink and uses dynamic caching mechanism which allow the system to replicate some or all the data from the database server using DaaS scheme and stored in the cache of the surrogate object. The model also moves the data stored in the mobile device to the surrogate object through pay-as-you-use mechanism in order to reduce the operational and maintenance cost.

The surrogate object at the respective MSS can be considered as a common sharing workspace. In general, full replication of data is not suitable for many scenarios. Instead, partial replication can be used where only data that frequently used and accessed are replicated and distributed to the nearby mobile devices within a predefined range using conciliation. The Conciliation is the process by which the surrogate object and mobile devices that are present in a mobile network interact with each other to reach an agreement for various services and pricing and then sending partial replication to all the agreed devices and considers all these replication as local workspaces. Whenever the device initiates the transaction request, first it uses their own common workspace loaded in the respective MSS for transaction execution. If any changes happens to mobile cloud environment such as slow or unreliable wireless connections, the SOCC model automatically chooses the suitable method for transaction execution and device uses wireless communication technology such as Bluetooth, Wifi, Wimax and Ultra Wide Band to access the cached data from the nearby devices which are part of cloud network in their communication range instead of transmitting transaction request to a database server, thereby minimizing the need for cellular connections.

In our proposed model, during mobile transaction execution, the mobile device first works on the common workspace located in the surrogate object and then re-conciliates the results into the database server. Further if necessary, it performs the transaction execution using local workspace and these replicated data can be inconsistent across different local workspaces. The effect of transaction execution is first updated with local workspace, later it updates the result with common workspace and then it re-conciliates with the database server.

The big challenge in transaction execution in mobile cloud is mobility of the device. Mobility of the device is one of the major issues that differentiate transaction execution in mobile cloud environments form traditional cloud environments. Due to real time movements and frequently changing the location of mobile devices, mobility will have strong impact on transaction execution and it requires additional time on transaction execution and reconfiguration. Also, the transaction execution in the mobile cloud is severely affected by various constraints such as lower bandwidth, unstable network, frequent disconnection in wireless environment and limited energy, limited storage, limited functionality in the mobile devices. These complicated issues could be handled effectively without affecting the ongoing transaction execution using our proposed model.

In our model, the transaction execution in mobile cloud environment need not bother about whether mobile device is in mobility or out of coverage area. When mobile device initiate a transaction, our proposed model simply submits the transaction request to appropriate surrogate object for processing which is residing in some mobile support station. The surrogate object takes sufficient time to execute the transaction with the help of data cache in the surrogate object or with help of replicated data cache residing in another nearby mobile devices in the respective cellular region. By this way, the model could help in continuing with the transaction execution during disconnection period which happens due to limited and extremely latent wireless connectivity. In some situation, due to mobility, the mobile devices have moved to far away MSS with respect to surrogate object. Due to this, heavy traffic and low latency may be occurred in the cloud platform. The surrogate object model supports migration of surrogate object in which surrogate object could migrate either to the less loaded MSS or less loaded mobile devices as pay-as-you-use model.. This is achieved by implementing Database as Service (DaaS) scenarios. Here, suitable method is deployed to deal with data object being modified by the mobile devices and to reconcile different replicated versions in local workspace or to maintain a single updated version in common workspace in order to maintain the consistency among all replicas. To show the usefulness of our proposed model, the following section suggests an algorithm for the execution of transaction over the mobile cloud environment. The abstract flow of operations involved in this model is depicted in fig 2.

B. SOCC Algorithm

In this section, we present SOCC algorithm that allows required data to be granted for transaction processing during normal and disconnected period using surrogate object based caching mechanism. The following notations and assumptions have been considered.

Ti → Transaction id where i = 1,2,3….

Tij → Sub Transaction id for Ti where i, j=1,2,3…

MSSij → Mobile Device Identifier

i represents MSS id ; j represents mobile device id in the MSS (i,j = 1,2,3..)

MSSij_k → Surrogate object id hosted at ith MSS on behalf of jth mobile device

Di → required datasets for Ti

Tij-Xr → request read lock on data X for Tij

Tij-Xw → request write lock on data X for Tij

CTM → Cloud Transaction Manager is virtual machine which is closely associated with surrogate object, replicas and database server for effective transaction management system over the mobile cloud.

Catalog 1 at each MSS: maintains list of MDID and SOID; whenever mobile device enters into mobile cloud environment, system updates catalogue 1 with MDID and object reference SOID..

Catalog 2 at each MSS: Maintains Ti, Tij, MDID, SOID and Status entries; also stores Read / Write lock request for Tij. Updates whenever Tij request from the mobile device and during the execution of Tij.

Here, mobile device need to access the data through the cache in surrogate object in normal period (ie wireless network is available for communication) and access the replica of data cache at nearby devices during weak and extremely latent cellular connectivity period. The surrogate object would be stored as middleware on cloud environment and data cache in surrogate object is partially replicated and stored in the nearby devices within the predefined cellular region as pay-as-you-use model. In this case, transaction need not query the required datasets directly from the database server, instead it needs to access the datasets either in the surrogate object (say common workspace) or in the replicas in nearby mobile devices (say local workspace),thereby minimizing the need for wireless cellular connection in most of the time. In the worst case, the model provides access to other mobile devices located on other MSS's and remote database server. The following algorithm describes how the transaction would be executed with help of surrogate object technique.

Algorithm: Surrogate object based Cloud Caching (SOCC)

Routine 1:

Initially:

a) Transaction request form mobile device; Ti

b) Sending Transaction request to CTM located at respective MSS;

c) Create sub transaction if required; Tij

d) Update the catalogue 2 with Ti, Tij, MDID entries.

CTM then sends a message to catalogue 1 to find the existence of surrogate object for the mobile device MDID.

If found,

Update SOID entry in the catalogue 2;

Else

Perform the tasks mentioned in Routine 2.

For each sub transaction Tij

If SO exist,

CTM checks if any other Transaction (say T) with the lower timestamp is being processed by any other CTM

If YES,

Wait till its turn to send transaction request comes

Else

Tij is forwarded to SO

Update the status for Tij as "Sent to SO" in catalogue 2.

When SO for receives Tij request, it checks its dataset cache Di for existence of required data item (say X) for Tij execution.

If SO holds data item X in datasets Di (cache Hit)

Request the read lock on data item X (say Tij-Xr)

If Tij-Xr lock request is granted,

SO performs transaction execution for Tij on cached data items;

After Execution, Request the write lock on data item X (say Tij-Xw);

If Tij-Xw lock request is granted,

SO update its cache and reconcile with common workspace. Update the status of Tij as "Committed" in catalogue 2.

If any conflict in read or write, inform to CTM for necessary action.

Else (in case of cache Miss)

Transaction execution for Tij is forwarded to CTM and update the status for Tij as "forwarded Back" in catalogue 2 for further action.

CTM then issues transaction request Tij to the various mobile devices present within a predefined range and checks for the existence of replicas and required data sets Di in the local workspace and CTM required to wait for acknowledgement from only the current replicas in local workspace.

If replica and Di exist,Forward the request Tij to replica node

Request the read lock on data item X (say Tij-Xr);

If Tij-Xr lock request is granted, Node performs transaction execution for Tij on replica data sets

After Execution, Request the write lock on data item X (say Tij-Xw);

If Tij-Xw lock request is granted, Node updates its replica (local workspace) and reconcile with common workspace.

Update the status of Tij as "Committed" in catalogue 2.

If any conflict in read or write, inform to CTM for necessary action.

CTM then issues write request to other replicas in the local workspace for maintaining the consistency among replicas.

A result of the transaction is forwarded to respective surrogate object for delivery of final transaction result to mobile device. if all subtractions Tij of Ti are committed; CTM forward the result of the transaction to mobile device and in turn device sends an acknowledgement of the receipts of the transaction result to CTM and CTM removes transaction entry for Ti in catalogue 2 .if the mobile device is not in their communication region (due to mobility) , the result of the transaction targeted at a mobile device need not bother about whether the mobile device is in motion or out of coverage area. All that needs to be done is to retain the result of the transaction Ti in the respective surrogate object. The surrogate object takes opportune time to deliver the result of the transaction to its mobile device when it reconnects to its mobile support station.

Routine 2:

If SO not found in the respective MSS (due to SO migration), CTM sends a request to the location server requesting it the object reference of the surrogate object of mobile device MDID.

Locations server looks up the entry for the surrogate object of MDID and return the object reference and its location to CTM.

Upon receipt of the location of surrogate object, transaction request Ti is forwarded to new location (another MSS) and mark status as "sent to SO in new MSS"

Repeat all the steps mentioned in Routine 1 for transaction execution.

If all the subtractions Tij of Ti are committed properly and all conflicts are resolved, the committed transaction result in the common workspace (surrogate object) will be finally reconciled at the database server. When required datasets Di doesn't exist either in SO (common workspace) or in the replicas (local workspace), the transaction request is directly forwarded to database server for execution.

The different possible scenarios of the proposed model are: The CTM administrates cached data in the surrogate object & replicas. When mobile device initiates the transaction request, the proposed SOCC mechanism supports transaction execution either at the surrogate object or at the mobile devices which have replicas. Local managers (say LTM) at the mobile device takes responsibility for managing the submitted transaction request Ti. Both CTM and LTM can interact with each other for transaction execution to specify and submit transactions.

Consider a case that the mobile device is moved to the area in which cellular connectivity becomes incrementally weaker (measured through predefined threshold) and also consider the data cache is missed in the respective surrogate object. In this case, the CTM at the MSS forward the transaction request to one of the mobile device in their communication area (possibly less loaded mobile device).When wireless connectivity becomes completely latent, the proposed model automatically use the short range wireless communications such as Bluetooth or wireless USB to establish a connection among the mobile devices in their communication area and execute the transaction on shared replica datasets. Once the transaction is executed, the result will be stored in the device and it will dispose the result to surrogate object and effect of transaction result is reconciled with common workspace when wireless network is regained in the environment.

The proposed method provides an excellent technique for transaction execution either at the surrogate object or at the mobile device to share the data sets on both common and local workspace while the device being on move or disconnected from the MSS due to poor and extremely latent wireless cellular connection. The above mentioned algorithm transparently spreads the data over many surrogate objects and many mobile devices. This is accomplished via DaaS and pay-as-you-use caching method. This proposed method allows transaction to avoid unnecessary data lock-in, delay in getting the required data sets from the data centers, reduce the cost of switching providers and multiple data centers for getting the required data sets for execution, and also provide better fault tolerance during disconnection and failures. We have build and evaluate a middleware for SOCC technique and estimate latency rate, success rate of transaction, wireless and wired access rate for execution, support for disconnection, estimates the costs incurred and benefits reaped.

IV. Future Work

This work is in progress. We are implementing this idea on a mobile cloud as a simulation prototype. This SOCC technique aims to measure average waiting time for transaction execution, average success rate of transaction during dynamic changes of environmental conditions in order to measure the throughput of the proposed algorithm for concurrent transaction execution and also estimates the costs incurred for the same. We have an idea to implement this technique for various transaction loads in order to test the feasibility of our method. Further, we have to develop a commit protocol for mobile cloud transactions, migration protocols for surrogate objects. The commit of a sub transaction in one surrogate object might also depend on the other sub transactions that are being executed either at another surrogate object or at another mobile device. So we need to develop a suitable commit protocol for transaction management over the mobile cloud.

V. Conclusion

In this paper, we are trying to develop an algorithm for transaction management over the mobile cloud to achieve the high throughput, minimum latency transactions, and high performance reliable transaction processing. The proposed SOCC technique supports both surrogate object and mobile device based transaction execution during normal, poor and extremely latent wireless connectivity. In the mobile cloud environment, the constraints on the mobile devices and wireless cloud environment are relatively high. We have developed an algorithm for supporting transaction execution on the surrogate object or on the replicas over the cloud network with minimum usage of wireless bandwidth, more number of transaction commit without exceeding the budget and dead line for transaction execution. 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.

References

[1] M. A. Maluk Mohamed, D. Janaki Ram, and Mohit Chakraborty, "Surrogate Object Model: A New Paradigm for Distributed Mobile Systems," *Proceedings of the 4th International Conference on Information Systems Technology and its Applications* (ISTA'2005), May 23-25, 2005 - 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," *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," *Proceedings of MoMM2011_ERPAS, ACM 9th International Conference on advances in mobile computing and multimedia*, 2011, iiWAS2011, vol. 5-7 December, 2011, Ho Chi Minh City, Vietnam. Copyright 2011 ACM 978-1-4503-0784-0/11/12. 2011.

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

[5] Adrian Daniel Popescu, Debabrata Dash, Verena kantere, and Anastasia 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. J. Shi, X. F. Meng, 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. 2009.

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

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

[11] X. Y. Zhang, J. Ai, Z. Y. 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 L. Bouge, "BlobSeer: Next generation data management for large scale infrastructure," *Journal of Parallel and distributed computing, volume 71 issue 2*, February 2011, doi: 10.1016/j.ipdc.2010.08.004, 2011.

[13] Michael 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," *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," *Proceedings of 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, 2010, doi:10.1109/CCGRID.2010.46, 2010.