

IMPROVING DATA AVAILABILITY IN MOBILE ENVIRONMENT USING DATA ALLOCATION

**Moaiad Ahmad Khder, Zulaiha Ali Othman,
Abdullah Mohd Zin and Salha Abdullah**

School of Computer Science, Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, 43600, Bangi, Malaysia

Received 2013-05-08, Revised 2013-05-16; Accepted 2013-07-05

ABSTRACT

Data distribution is one of the crucial issues in Data Base Management Systems (DBMS) in general and in Mobile environment in Particular. It is important because, if not properly managed, it will cause reduction in data availability, which in turn causes more rejections in transactions. Replication algorithms (e.g., CCM) are used to improve data availability. However, the database replication algorithms in general will increase the storing and communication costs for updates especially when the DB is very large and MU number is also large; this will lead to a congested network. An alternative approach is to use data allocation (e.g., TMM-MDB). The data allocation algorithm used in TMM-MDS doesn't allocate data fairly for MU and data availability reduced over time. Our study consists of simulation supported by a statistical method. We examined our proposed algorithm for data distribution called Data allocation using weight factor for mobile environment. The simulation evaluates the past history and the current claims of the data allocation in order to find out an improved data distribution method for the mobile environment. Our simulation results proved that our proposed method increases the data availability in mobile environment by 75% and distribute data fairly.

Keywords: Data Allocation, Data Replication, Mobile Database, Transaction Management, Mobile Environment

1. INTRODUCTION

Data distribution is a very important issue in distributed database, the database fragments need to be assigned to nodes in the computer network. Three replication scenarios exist: A database can be full replicated, partially replicated, or unreplicated. Data allocation describes the process of deciding where to locate data, data allocation strategies are: centralized data allocation, partitioned data allocation and replicated data allocation. Data distribution over a computer network is achieved through data partition, through data replication, or through combination of both.

1.1. Mobile Database Allocation

Data fragmentation is a technique for data organization that allows efficient data distribution and

processing. Each fragment obtained corresponds to a different physical file and is allocated to a different server which is running on sites, the result being the allocation schema.

The allocation problem involves finding the "optimal" distribution of fragments on sites (Ozsu and Valduriez, 2011). The optimality can be defined with respect of two measures: minimal cost and performance. The cost function consist of the cost of storing each fragment at a site, the cost of querying a fragment at a site, the cost of updating a fragment at all sites where it is stored and the cost of data communication. The allocation attempts to find an allocation scheme that minimizes the combined cost function.

Fragments allocation across the nodes must consider some factors: the data will be stored near to sites that use them; the data must be available even in the case of site

Corresponding Author: Moaiad Ahmad Khder, School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Malaysia

failure using data replication on many sites; the fragment allocation must implies minimal storing costs and communication costs. There are four alternative strategies for data allocation on sites: centralized, partitioned, complete replication, selective replication:

- Centralized strategy assumes to have one database and one DBMS, both of them stored to one site, having users distributed on network. In this case, the communication is costly because all accesses from users out of central site use communication lines. The liability and availability of this type of distributed system are low, because a failure occurred on central node will guide to a total system loss
- Partitioned strategy supposes to partition the database in disjoint fragments, each of these stored to a site. If the data are placed on the site which uses them frequently, then the local character of reference is
- High. Because the fragment is not replicated, the store costs are low, but the liability and availability are also low, but higher then centralized systems. System performances could be good and communication costs could be low if the distribution is correctly designed
- Complete replication strategy assume to have a complete database copy on every site. In this situation, the local character of reference, the liability and the availability are excellent, but the storing and communication costs for updates are the largest possible. A compromise solution is snapshot use, which will be bringing up to date periodically
- Selective replication strategy represents a combination between partitioning, replication and centralization. Some items on database are partitioned to obtain a high local character of reference and other items, frequently used on many sites and rarely updated are replicated. The rest of items are centralized. The objective of this strategy is to obtain the others strategies advantage, but none of those disadvantages, minimizing costs and maximizing performances. Because its flexibility, this is the most used data allocation strategy on sites for distributed systems

Fragments allocation is the simplest solution. The fragments allocation determining method, also named the best choice method (Atzeni and Paraboschi, 1999; Kifer *et al.*, 2006), consists in every possible allocation measurement and to choose the site with the

best measure. This method offer a solution which exclude the possibility to place a fragment to a site where is stored a related fragment. Data replication increase the design complexity because the replication degree of every fragment became allocation variable and then, the read accesses became complicated because application must select, from many alternatives, the sites to access fragments.

Fragment allocation on sites must be done according with performance-cost balance. Performance could be obtained with a good response time from the system and an increased availability. The cost is composed from hardware cost, which includes the processing cost and the storage cost and communication cost respectively. The reason for having distributed databases is not that of maximizing the interaction and the necessity of transmitting data via networks. On contrary, the planning of data distribution and allocation should be done in such a way that the largest number possible of application should operate independently on a single server, to minimize the execution cost that is typical to distributed application.

1.2. Structure of the Study

The remaining of this study is structured as per the following. In previous studies, other models involved in data replication and allocation is described, followed by explanation of the problem statement of this research. Moreover, we illustrate the proposed solution comprised of the statistical method and the simulation of this method, which is utilized by our algorithm. The Result is devoted to data analysis, which depicts the result of applying the proposed method on a dataset used for the past working history of the typical data distribution methods. Discussion however, describes the innovative values of this research. Finally, in Conclusion we conclude this study and suggest some further works as complements to our proposed solution.

2. MATERIALS AND METHODS

2.1. Related Work

According to (Serrano-Alvarado *et al.*, 2004) data distribution models in the past, paid more attention to database replication as a solution to data availability and concurrency problems.

Vijay-Kumar *et al.* (2006) and Prabhu *et al.* (2004) introduced new concurrency control management

mechanism to overcome the weaknesses in data replication by proposing cached copy of database, where the model keeps a limit Λ for the amount of change that can occur on the replica at each MU, thus Λ_i denotes the total maximum change allowed in a replica of D_i at a MU.

Example

Consider a data object X representing total number of movie tickets. Let N_x be the number of replicas of X . Initially $X = 180$ and $N_x = 3$. X is replicated at MU_1 , MU_2 and MU_3 . In this example the function $f_x(X, N_x)$ that calculates Λ_x is $\Lambda_x = f_x(X, N_x) = (X/2) / N_x = X/2 N_x = 30$. Note that we divide X by 2 so that we keep some tickets for the request transaction, which cannot be executed at the MU. **Figure 1** showing the data distribution for this example.

However, the database replication in general and CMM will increase the storing and communication costs for updates especially when the DB is very large and MU number is also large; this will lead to a congested network.

Therefore, utilizing database replication cannot be an effective plan without investigating its pros and cons. Although database replication has many positive impacts on different aspects of the transaction management models (Serrano-Alvarado *et al.*, 2004), it can also bring harm and loss to the database without precise investigation on failure factors in adapting it. In this direction, (Abdul-Mehdi *et al.*, 2008; 2010) proposed Transaction Management Model for Mobile Database System (TMM-MDS) supports mobile concurrent disconnections of team members in a system. The system model of TMM-MDB contains BSs in a fixed network and Mobile Nodes (MN) in a wireless network which connect to the BS as shown in **Fig. 2**. The master data is stored in the BS. The BS makes changes to updates and parts of the master data for the team members. The team members' MNs are given the permission to connect to the server BS during the system time and make disconnections from part of the master data. The BS transfers part of the master data with the same timestamps to the MNs. This is done through the wireless network in connected mode. MNs make necessary changes to their data parts locally within the limit of their received data parts, during the validation of the timestamp. Before the timestamp process completes, the MNs are reconnected to the BS and the changes made to the updates are sent over to the BS.

The master data is seized and managed by the BS. Data is distributed to MHs which may update the data according to the equation:

$$\delta_i = f_i(d_i, m_i, n_i) = [(AV + (\epsilon * r)) * d_i / (m_i + n_i)]$$

The explanation of TMM-MDB Values shown in **Fig. 3**.

The data allocation algorithm used in TMM-MDS doesn't allocate data fairly for any successor mobile unit that connects to the MSS as shown in **Table 1**. For the first time we have two MUs each one of them will be allocated $\delta_1 = 450$, the successor MU will be allocated $\delta_1 = 165$.

2.2. Research Hypothesis

In database systems (Ozsu and Valduriez, 2011; Atzeni and Paraboschi, 1999; Kifer *et al.*, 2006) the profit that the database replication earns, the penalty that it has to pay due to an unsatisfied deal and the inconsistency is tightly coupled with the data availability. Since data replication suffers from high storing capacity and communication costs for updates especially. Therefore, we believe that finding a way to approximately distribute the data with less inconsistency and more data availability to generate a realistic initial plan, which in turn prevents the system from the risk of using data replication and helps the system to increase the data availability.

2.3. Problem Statement

With advances in mobile processing and distributed computing that occurred in the operating system arena, the database research community did considerable work to address the issues of data distribution, distributed transactions management, distributed query processing (Connolly and Begg, 2009). One of the major issues in data distribution is replicated data management at the Mobile Host (MH). Replication can improve data availability; however by using replication, the distributed system will suffer from data inconsistency, data access delay and network overhead (Pamila and Thanushkodi, 2010). Data allocation is suggested to overcome these problems.

2.4. Proposing Data Allocation Algorithm Using Weight Factor

The new algorithm is proposed and implemented. The main idea of our proposed algorithm is to distribute the data between mobile network and fixed network using weight factor that representing the need or demand for the data.

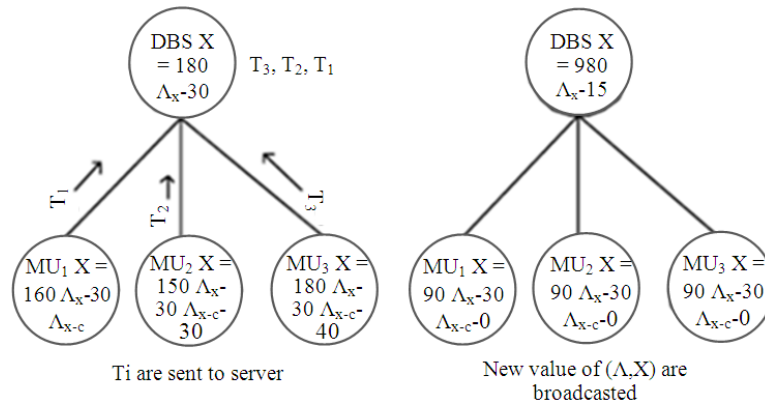


Fig. 1. Data distribution in CCM

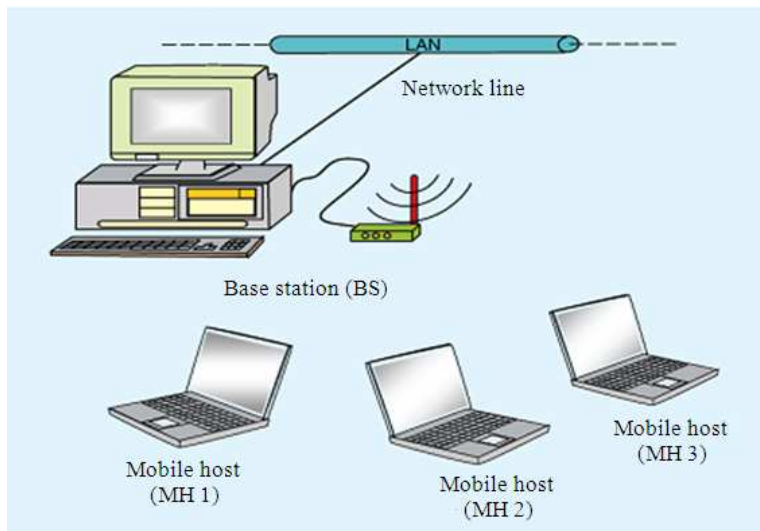


Fig. 2. Transaction management model for mobile databases architecture

Values	Representation
D	A set of data items
$i \in N$	A set of natural numbers (non-negative integer)
D_i	Data item related to a distributed reservation system (number of flight seats, number of football tickets, etc)
n_i	The number of MHs requesting a data item
m_i	The number of MHs formerly disconnected
δ	The total maximum changed allowed at each MH
Δ_i	A portion of D_i
β_i	The balance of D_i
f_i	The function of D_i, n_i, m_i

Fig. 3. Explanation of TMM-MDB Values

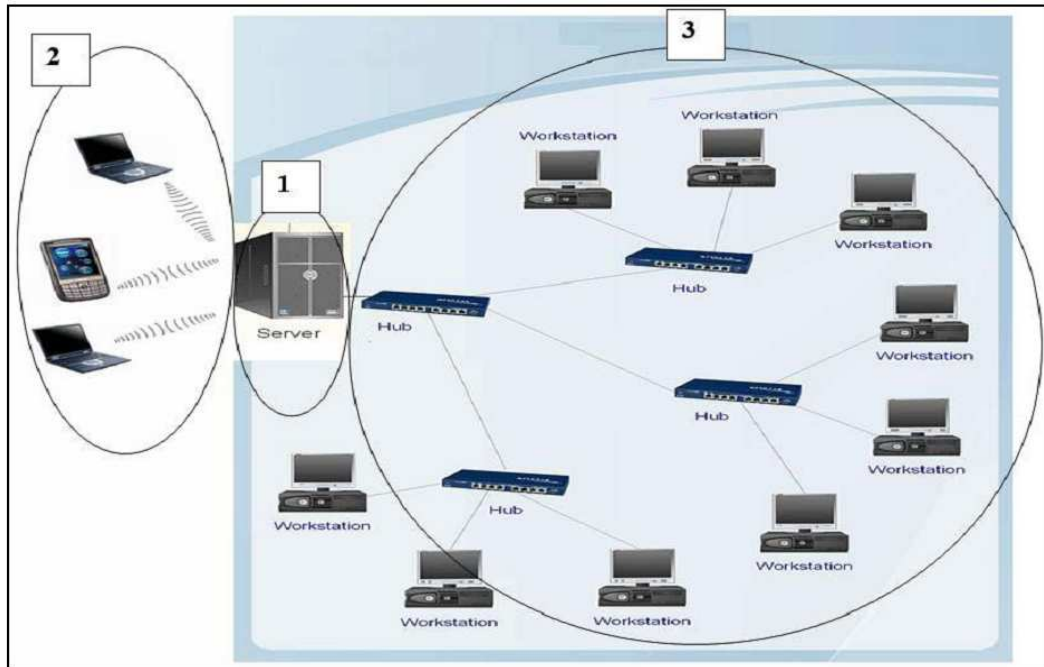


Fig. 4. FETOTM Architecture

Data Allocation can be used to improve data availability and reduce rejected transactions in distributed database environments. In such a system, a mechanism is required to maintain the consistency of the data. Fixed Network can be in different topologies. In this model, we proposed a technique where a data will be allocated to some selected nodes in the fixed network and mobile hosts.

The basic concept of the algorithm is to allocate the data to the base station (Fig. 4(1)), the mobile network nodes Fig. 4(2) and some selected nodes in the fixed network (Fig. 4(3)) Fig. 4.

Assume the Data will be D, so:

- BS allocated data = D/z
- where, $z = 3$, because we have 3 main components in our proposed system namely fixed network, mobile network and BS.
- $FN_MN_d = X =$ The data will be reserved for the fixed network and the mobile network
- $X = FN_MN_d = D - D/z = 2/3 D$
- Mobile Network allocated data = $X * D * \text{Mobile network weight}$
- Fixed network allocated data = $X * D * \text{Fixed network weight}$

Where:

Mobile network weight + Fixed network weight = 1:

$$\text{One Mobile unit allocated data} = \frac{\text{Mobile network allocated data}}{\text{No of the MU}}$$

$$\text{One Fixed network node allocated data} = \frac{\text{Fixed network allocated data}}{n \text{ int}(\sqrt{n})}$$

where, n is the number of the fixed network hosts.

The Data will be distributed to 3 parts:

- Fixed network (FN)
- Base station (BS)
- Mobile Network (MN)

2.5. Distribution Process

Step1:

BSd = The data will be reserved for the Base Station (BS)

$$BSd = 1/z * \text{Data} = 1/3 \text{ Data}$$

$$BSd = \frac{1}{3} D$$

Step2:

FN_MN_d = X = The data will be reserved for the fixed network and the mobile network

$$X = FN_MN_d = D - D/z = 2/3 D$$

$$X = \frac{2}{3} D$$

Step3:

XF = the data will be reserved for the fixed network

XM = the data will be reserved for the mobile network
 $XF = X \times FNweight$
 $XM = X \times MNweight$

The data will be distributed between the fixed network and the mobile network according to the weight
 Step4: The fixed network data will be distributed between selected fixed network nodes. The selected nodes will be chosen by using the square root of the total number of the fixed network nodes.

In the previous models they use to distribute that data to all nodes or to selected nodes like the DRG model and the distribution be as replication not as allocation.
 Y = the number of the selected fixed network nodes:

$$Y = \lfloor \sqrt{N} \rfloor$$

XF_i is the data will be allocated to the selected fixed network node:

$$XF_i = XF / y$$

where, y is the number of the selected fixed network nodes.
 Step5: The mobile network data will be distributed between the mobile network nodes:.

$$XM_i = XM / m$$

where, m is the number of the mobile network nodes:.

$D = BSd + XF + XM$
 $XF = XF_1 + XF_2 + \dots + XF_y$
 $XM = XM_1 + XM_2 + \dots + XM_m$

where, m is the No of the mobile network nodes and y is the number of the selected fixed network nodes.

2.6. Case Study

Assume we have Data (D)=1800 , 6 mobile Host ($n = 6$) and our Fixed network consists of 25 nodes ($y = \lfloor \sqrt{25} \rfloor = 5$). Data will be allocated to BS:

$$BSd = \lfloor Data / 3 \rfloor = \lfloor 1800 / 3 \rfloor = 600$$

Data will be allocated for mobile network and fixed network will be:

$FN_MN_d = X = 2/3 * D$
 $FN_MN_d = X = 2/3 * 1800 = 1200$

First Example

where, $MN_weight = FN_weight$ which it means the demand and distribution of the data will be equally between MN and FN.

Data will be allocated for mobile network will be:

$$\begin{aligned}
 XM &= X \times MNweight \\
 XM &= 1200 * 0.5 = 600
 \end{aligned}$$

The mobile network data will be distributed between the mobile network nodes:

$$XM_i = XM / m = 600 / 6 = 100$$

Data will be allocated for fixed network will be:

$$\begin{aligned}
 XF &= X \times FNweight \\
 XF &= 1200 * 0.5 = 600
 \end{aligned}$$

The fixed network data will be distributed between the selected fixed hosts:

$$XF_i = XF / y = 600 / 5 = 120$$

This example details showin in **Fig. 5**.

Second Example

where, $MN_weight < FN_weight$ which it means that the FN demands more data than MN.

Data will be allocated for mobile network will be:

$$\begin{aligned}
 XM &= X \times MNweight \\
 XM &= 1200 * 0.4 = 480
 \end{aligned}$$

The mobile network data will be distributed between the mobile network nodes:

$$XM_i = XM / m = 480 / 6 = 80$$

Data will be allocated for fixed network will be $XF = X \times FNweight$:

$$XF = 1200 * 0.6 = 720$$

The fixed network data will be distributed between the selected fixed hosts:

$$XF_i = XF / y = 720 / 5 = 144$$

This example details showin in **Fig. 6**.

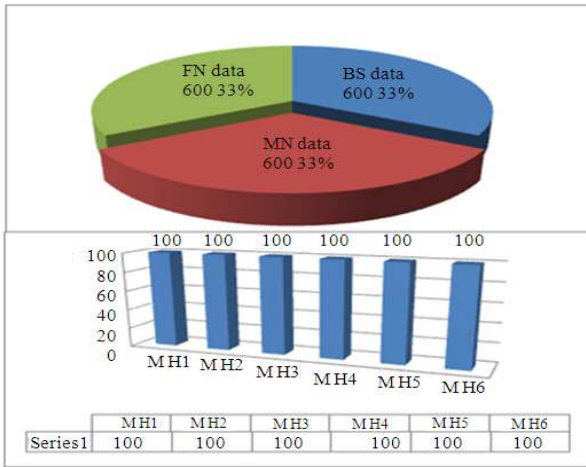


Fig. 5. MN_weight = FN_weight in FETOTM

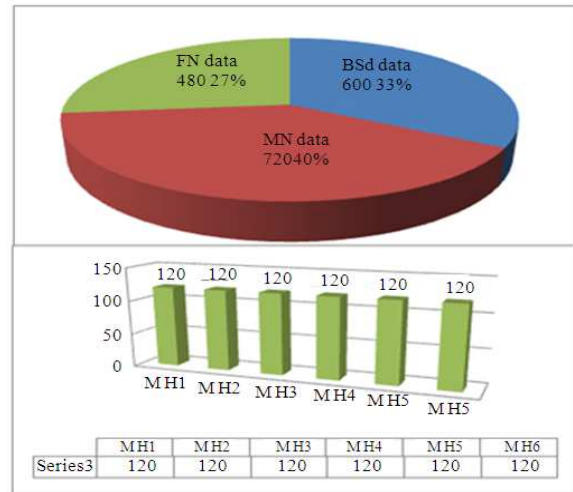


Fig. 7. MN_weight > FN_weight in FETOTM

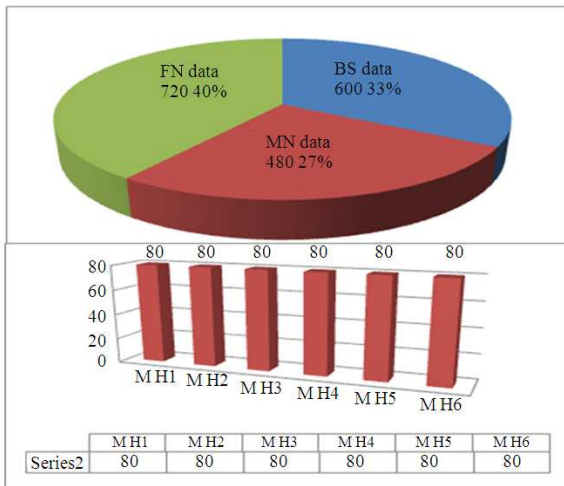


Fig. 6. MN_weight < FN_weight in FETOTM

Third Example

where, $MN_weight > FN_weight$ which it means that the MN demands more data than FN.

Data will be allocated for mobile network will be:

$$XM = X \times MN_weight$$

$$XM = 1200 * 0.6 = 720$$

The mobile network data will be distributed between the mobile network nodes:

$$XM_i = \frac{XM}{m} = \frac{720}{6} = 120$$

Data will be allocated for fixed network will be $XF = X \times FN_weight$:

$$XF = 1200 * 0.4 = 480$$

The fixed network data will be distributed between the selected fixed hosts:

$$XF_i = \frac{XF}{y} = \frac{480}{5} = 96$$

This example details show in Fig. 7.

By applying the data allocation method of TMM-MDB and the proposed model for the above case study, we can get Fig. 8 which can clearly shows the fair distribution by FETOTM and the descending distribution by TMM-MDB.

On other hand, by applying the data distribution method of CMM and the proposed model for the above case study, we can get Fig. 9 which can clearly shows the fair distribution by FETOTM and the replication load on each MH by CMM.

2.7. Experiment Setup

We have used simulation model to measure the performance of the proposed data allocation using weight factor. Due to space limitation we do not include the simulation deep details. The execution of the simulation is controlled by a timing routine, which selects the event to occur from the events list in Fig. 10 and executes the appropriate event routine.

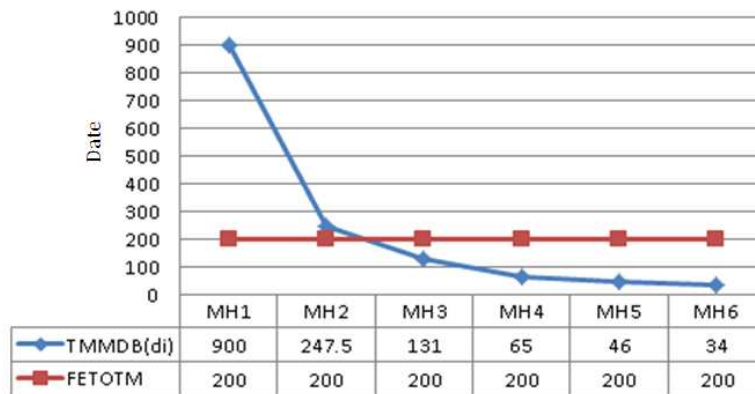


Fig. 8. (FETOTM vs. TMM-MDB) Data allocation

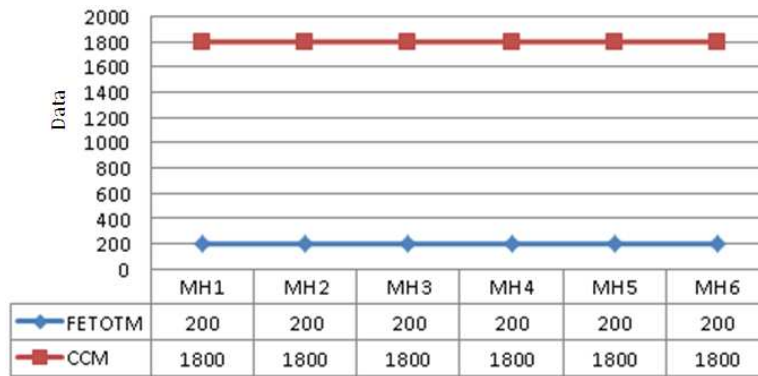


Fig. 9. FETOTM(allocation) vs. CMM(replication)

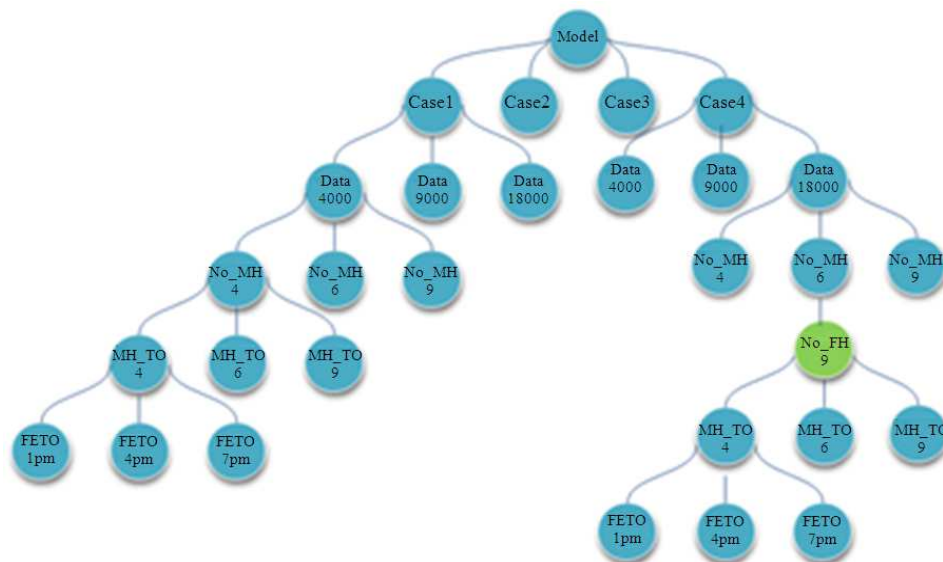


Fig. 10. Events list structure

Table 1. Data allocation sample in TMM-MDB

$D_i(X_{server})$	n_i	m_i	ϵ	AV	R	δ_i	δ_i/D_1 (%)	δ_{i+1}/δ_1 (%)
1800	2	0	0.05	0.5	0	450	50	-
900	1	2	0.05	0.5	1	165	18	37
735	3	3	0.05	0.5	2	74	30	45

Table 2. Simulation parameters

	Parameter	Input		
		Description	L	MH
Case	Case (Mobile network only, or with fixed network)			
Model	Transaction model (TMM-MDB, CMM/TCOT, and proposed algorithm)			
Data	Data	4000	9000	18000
No_MH	No of mobile hosts	4	6	9
No_FH	No of fixed hosts		9	
MH_TO	Mobile host timeout	4	6	9
FETO	Fixed end timeout	1pm	4pm	7pm
Data availability	Data availability			

Table 3. Data effects on data availability

Multiple comparisons data availability tukey HSD				
(I) data	(J) data	Mean difference (I-J)	Std. Error	Sig.
4000	9000	0.45889*	0.44909	0.01
	18000	0.41426*	0.44909	0.01
9000	4000	-0.45889*	0.44909	0.01
	18000	-0.04463*	0.44909	0.01
18000	4000	-0.41426*	0.44909	0.01
	9000	0.04463*	0.44909	0.01

Table 4. Number of MH effects on data availability

Multiple comparisons data availability tukey HSD				
(I) Number of mobile host	(J) Number of mobile host	Mean difference (I-J)	Std. Error	Sig.
4 host	6 host	-0.17*	0.51	0.01
	9 host	-0.69*	0.48	0.01
6 host	4 host	0.17*	0.51	0.01
	9 host	-0.52*	0.42	0.01
9 host	4 host	0.69*	0.48	0.01
	6 host	0.52*	0.42	0.01

Table 5. Transaction execution time comparison between the simulated models by using T-test

Multiple comparisons data availability tukey HSD				
(I) Model	(J) Model	Mean difference (I-J)	Std. Error	Sig.
FETOTM	TMM-MDB	-140.61468*	0.358	0.00
	TCOT	-200.61946*	0.35793	0.00
TMM-MDB	FETOTM	140.61468*	0.358	0.00
	TCOT	-140.00478*	0.37642	0.00
TCOT	FETOTM	200.61946*	0.35793	0.00
	TMM-MDB	140.00478*	0.37642	0.00

Table 2 summarizes the main simulation parameters and their descriptions and their values that used in this research will be in the consequence sections.

3. RESULTS AND DISCUSSION

In this study, three values were assigned to the data (4000, 9000 and 18000). Data effects on the data availability were analyzed statistically using post hoc comparison as shown in **Table 3**.

The data showed that there is a significant ($p < 0.05$) difference between data amount (Data) effects on the data availability. The comparison of 4000 data over 9000 ($M = 0.45889$, 95% CI), data 4000 over 18000 ($M = 0.41426$, 95% CI) and data 9000 over 18000 ($M = -0.04463$, 95% CI) clearly indicate that data gave significant difference on the data availability at $p < 0.05$.

In addition to the relation between data effects and the data availability, any possible effect of number of mobile host on the data availability has been checked. Three types of mobile host were 4, 6 and 9. Data effects on the data availability were analyzed statistically using post hoc comparison as show in **Table 4**.

The comparison of 4 host over 6 host ($M = -0.16750$, 95% CI), 4 host over 9 host ($M = -0.69196$, 95% CI) and 6 host over 9 host ($M = -0.52446$, 95% CI) clearly shows that the data gave significant difference on the data availability at $p < 0.05$.

Data availability comparison between the simulated models has done. Three types of models used in this study were TMM-MDB, FETOTM and TCOT. Transaction execution time comparison between the simulated models was analyzed statistically using post hoc comparison. The data showed that there is a significant ($p < 0.05$) difference between FETOTM and TMM-MDB ($M = -140.61468$, 95% CI) $p = 0.000$. Moreover FETOTM and TCOT ($M = -200.61946$, 95% CI) $p = 0.000$ also were statistically significant ($p < 0.05$). In addition TMM-MDB and TCOT ($M = -140.00478$, 95% CI) shows any significant difference in the transaction execution time as show in **Table 5**.

4. CONCLUSION

In conclusion, the result of this research proves that data allocation as a data distribution method is much better in mobile environment comparing to the replication and our proposed algorithm is more efficient comparing to other data allocation Data distribution systems which are implemented using data allocation and in particular this approach are more generic, adaptable and consistent in comparison with other approaches.

In this study we have proposed and formulated a method to manage the data distribution in mobile environment. The proposed model is based on evaluation of the past working history of data distribution methods. The main objective of the proposed model is to improve data availability and introduce a new data distribution method.

Furthermore, the mechanism of using the weight factor is an extra effort for the data allocation that can ensure the fair distribution of the data between all the participants. Finally, the proposed method has been applied on a mobile environment system consists of mobile network, fixed network and mobile support station. The results of our observation and analysis reveal that the proposed method increases the overall data availability for a data distribution by 75% in average. This rate is a considerable figure that proves the efficiency and applicability of the proposed method.

However, despite of the proven efficiency of the proposed method, there are many other factors that can be added to the formal method in order to increase the data availability. Some of these factors are: the cost of storing each fragment at a site, the cost of querying a fragment at a site, the cost of updating a fragment at all sites where it is stored and the cost of data communication.

5. REFERENCES

- Abdul-Mehdi, Z., A.B Mamat, H. Ibrahim and M.M. Dirs, 2010. A model for transaction management in mobile databases. *IEEE Potentials*, 29: 32-39. DOI: 10.1109/MPOT.2010.936929
- Abdul-Mehdi, Z.T., A.B. Mamat, H.I. and M.M. Dirs, 2008. A Transaction consistency management model for mobile databases. *IETECH J. Commun. Tech.*, 2: 089-98.
- Atzeni, P. and S.C.S. Paraboschi, 1999. *Riccardo Torlone-Database Systems. 1st Edn., Concepts, Languages and Architectures-McGraw-Hill.*
- Connolly, T.M. and C. Begg, 2009. *Database Systems: A Practical Approach to Design, Implementation and Management. 8th Edn., Cengage Learning, Boston, Mass., ISBN-10: 1423902017, pp: 704.*
- Kifer, M., A.J. Bernstein and P.M. Lewis, 2006. *Database Systems: An Application Oriented Approach. 2nd Edn., Pearson Education, Boston, ISBN-10: 0321312562, pp: 1235.*
- Ozsu, M.T. and P. Valduriez, 2011. *Principles of Distributed Database Systems. 3rd Edn., Prentice-Hall, New York, ISBN-10: 1441988343, pp: 845.*
- Pamila, J.C.M.J. and K. Thanushkodi, 2010. Recovery algorithm for transaction framework in Wireless Mobile Environ. *Int. J. Compu. Sci. Eng.*, 2: 1228-1232.

- Prabhu, N., V. Kumar, I. Ray and G.C. Yang, 2004. Concurrency control in mobile database systems. Proceedings of the 18th International Conference on Advanced Information Networking and Application, (INA' 04).
- Serrano-Alvarado, P., C. Roncancio and M. Adiba, 2004. A survey of mobile transactions. Distributed Parallel Data., 16: 193-230. DOI: 10.1023/B:DAPD.0000028552.69032.f9
- Vijay-Kumar, M., H. Wu, R. Jones, G. Grant and B. Babbitt et al., 2006. Flagellin suppresses epithelial apoptosis and limits disease during enteric infection. Am. J. Pathol., 169: 1686-1700. doi: 10.2353/ajpath.2006.060345