Object Caching Model for Cell Phone Mobile Database

Khubaib Ahmed Qureshi
Department of Information Technology, Hamdard Institute of Management Sciences
Hamdard University, Main Campus, Karachi, Pakistan

Abstract - *Inherent limitations of mobile computing environment such as limited bandwidth, limited resourced mobile phones, and instability of wireless environment require summarized mobile database at mobile phones. Therefore the critical database summarization is done through an adoptive model incorporating object accessibility pattern in database queries by making optimal use of exponentially weighted moving averages. The database will keep only most frequently access objects at mobile phone to improve performance and provide high data availability for disconnected operations. Mathematical model and summarization scheme is proposed, along with simulation study, which identifies most frequently accessed objects to ensure high data availability for small memory mobile phones.*

Keywords: DEWMA Scheme, Cell-Phone Database Minimum Object Replication Identification, Dynamic Adoptive Object Access Pattern Identification.

1. Introduction

Applications of mathematics are very much diversified and especially regarding computer science in database systems, it is always productively utilized. Therefore in prevalent field of mobile computing to fulfill the typical requirement of reliable and scalable mobile database computing; we introduce entirely different and unique application of mathematics in constructing mathematical model for replication in mobile database. The mathematical model ensures high data availability by finding mostly required objects for a MH over rarely required objects for keeping them in limited storage to cater mobile database computing requirements. Optimal Dynamic Exponentially Weighted Moving Average of durations (DEWMA) technique is implemented for adapting hidden dynamics in object calling pattern by the queries at MH. It is required to cater light-weight mobile database requirements over conventional commonly employed principle of locality based techniques which has widely been adapted by most database systems.

Conventional techniques are less efficient in point-to-point paradigm and require more, storage and power. The research demonstrates the feasibility of proposed techniques by the simulation study.

A new light-weight high-level object oriented mobile database model is introduced which is mainly supported by the adoptive replication model based on DEWMA technique. It enables mobile database to request objects from the server very rarely, in point-to-point paradigm, by adapting the hidden dynamics in object calling pattern of queries and keeping minimum set of mostly required objects at MH.

2. Mobile database system

Impact analysis of mobile computing environment in the area of data management demands required optimal implementation of mobile databases at MUs which are supposed to be stripped-down version of their server-based counterparts, provided mostly with task specific data for offline data availability and accessibility with consistent database operations support, taking into account the limitations of wireless medium and limited resources at mobile unit. Provision of such replicated data at MU with high data availability and minimum required level of management functionalities, will surely be a solution. It mainly requires dynamic adoptable model to predict mostly required objects by adopting access pattern of user queries, for high data availability appropriate for mobile database computing.

3. Mobile database summarization model

Referring to our mobile database [1] where mobile phones are provided with small memory, low battery and scare bandwidth of expensive wireless computing environment, we provide high data availability at low memory MH by incorporating the mechanism of identifying mostly required objects for mobile client, which will always be made available through replacement.

Employing any conventional technique based on locality of reference per-page basis is not suitable for mobile database computing because in point-to-point paradigm every mobile client exhibit certain directed object calling pattern in different queries of specific intervals, therefore our proposed scheme adapts the hidden dynamics of object calling pattern in different queries and kernel level implementation of the scheme will provide required level of transparency.

In DEWMA, δ represents priority weight such as with δ=0.1 show higher priority weight for recent duration and δ=0.9 for lower priority weight to duration become aged. Considering r=1means when object i is first time accessed in mobile database, then the metric can be computed as "(1)", where d is duration from clock.

$$\overline{d}_i^{\delta_i} = (1-\delta_i)d_{i,1} \qquad \text{for r = 1} \qquad (1)$$

On the other hand, r>1 means object i is accessed more than once in mobile database, then the changing metric will be calculated as "(2)", such as d is inter object access duration.

$$\overline{d}_i^{\delta_i} = \delta_i \overline{d}_{i,r-1}^{(\delta_i)} + (1-\delta_i)d_{i,r} \qquad \text{for r > 1} \qquad (2)$$

By taking d: duration from current clock, in our scheme DEWMA, enables the feature to adapt quickly to changes in access pattern of specially those objects that are not being accessed for so long. Flow charts of the scheme can be seen in Figure 1.1 & Figure 1.2.

3.1. DEWMA scheme dynamics:

The scheme, Dynamic Exponentially Weighted Moving Average of durations DEWMA is finally proposed after the development and testing of our own versions of techniques, and best suited with object calling pattern of small size memory mobile phones provided for mobile database computing. DEWMA focuses on such features which are necessary to adapt the hidden dynamics of small memory client's object calling pattern within the period of time, infect DEWMA exhibits all those features like: systematically maintain object calling rate, earlier rates must be minor additive, dynamically update the rate to reflect the change by some intelligent means when they are not called for long period of time, provide priority to recently called objects.

3.2. Object replacement policy

When mobile database is filled, and new request is encountered, then mobile database checks the availability of new object within the database; incase, when miss is reported with no space for new object then, object having maximum calling duration rate, calculated by DEWMA, among all other object will be replaced with new object, requested by mobile client, otherwise when the object is hit then only changing metric is calculated.

3.3. Role of server in optimization of MDB

We recommend that if mobile database send object calling rate of every object with the request of new object from the server, before being replaced, then the sever can continue to calculate the weight of every object for each MDB, which will be transmitted with object itself, to make DEWMA more optimized.

3.4. Object broadcasting paradigm

High priority object within the server, with highest calling rate, evaluated by server with the cooperation of MDB must be selected for broadcast to reduce the transmission overhead for MU and easy availability without request. It provides better idea for more optimization [2]

4. Feasibility of scheme by simulation:

In this section we will explain simulation strategy for our scheme followed by typical results to demonstrate the feasibility of our scheme.

4.1 Simulation study of DEWMA:

In this simulation study we included our three schemes Hybrid, Clock Difference Exponentially Weighted Moving Average of durations (CDEWMA) and optimal DEWMA for effectiveness comparison with other related, MA, EWMA [3, 4], and conventional LRU, schemes.

Hybrid scheme is functional coupling of EWMA and MA, that can be calculated as "(3)", where w is maximum no of inter object access durations of object *i*.

$$\overline{d}_{i,r+1}^{(\delta,w_i)} = \delta_i \overline{d}_{i,r}^{(\delta_i,w_i)} + (1-\delta_i)d_{i,r+1} \qquad (3)$$

CDEWMA can simply be defined as "(4)", when r=1, means object *i* is first time requested in mobile database, where *d* is duration from clock to the time object first accessed, similarly the metric can be calculated as "(5)"

$$\overline{d}_i^{\delta_i} = d_{i,1} \qquad \text{for r = 1} \qquad (4)$$

when r>1, means object *i* is accessed more than once, where d is inter object access duration.

$$\overline{d}_i^{\delta_i} = \delta_i \overline{d}_{i,r-1}^{(\delta_i)} + (1-\delta_i)\,\mathrm{d}_{i,r} \qquad \text{for r > 1} \qquad (5)$$

In accordance with the realistic situations for mobile database computing using mobile phones, we have simulated series of schemes for three different configurations, from which two of them based on inter operations arrival patterns and last one based on object access rate. First configuration resembles with situation when frequency of operations arrival start getting increased dramatically at peak working hours of mobile clients for certain time interval when most of the operations are clustered, that is represented as Poisson arrival. Second configuration poses situation when mobile client performs same no of operations on all objects in equal intervals of time, without giving any preference to any particular object, which is very rare case in mobile database computing that is represented as Uniform arrival in our simulation, and finally the most suitable and frequently occurring configuration for mobile database computing is represented as Hotspot which represents such situation when each specific mobile client access some of the database objects pertaining to his interest within many of his routine

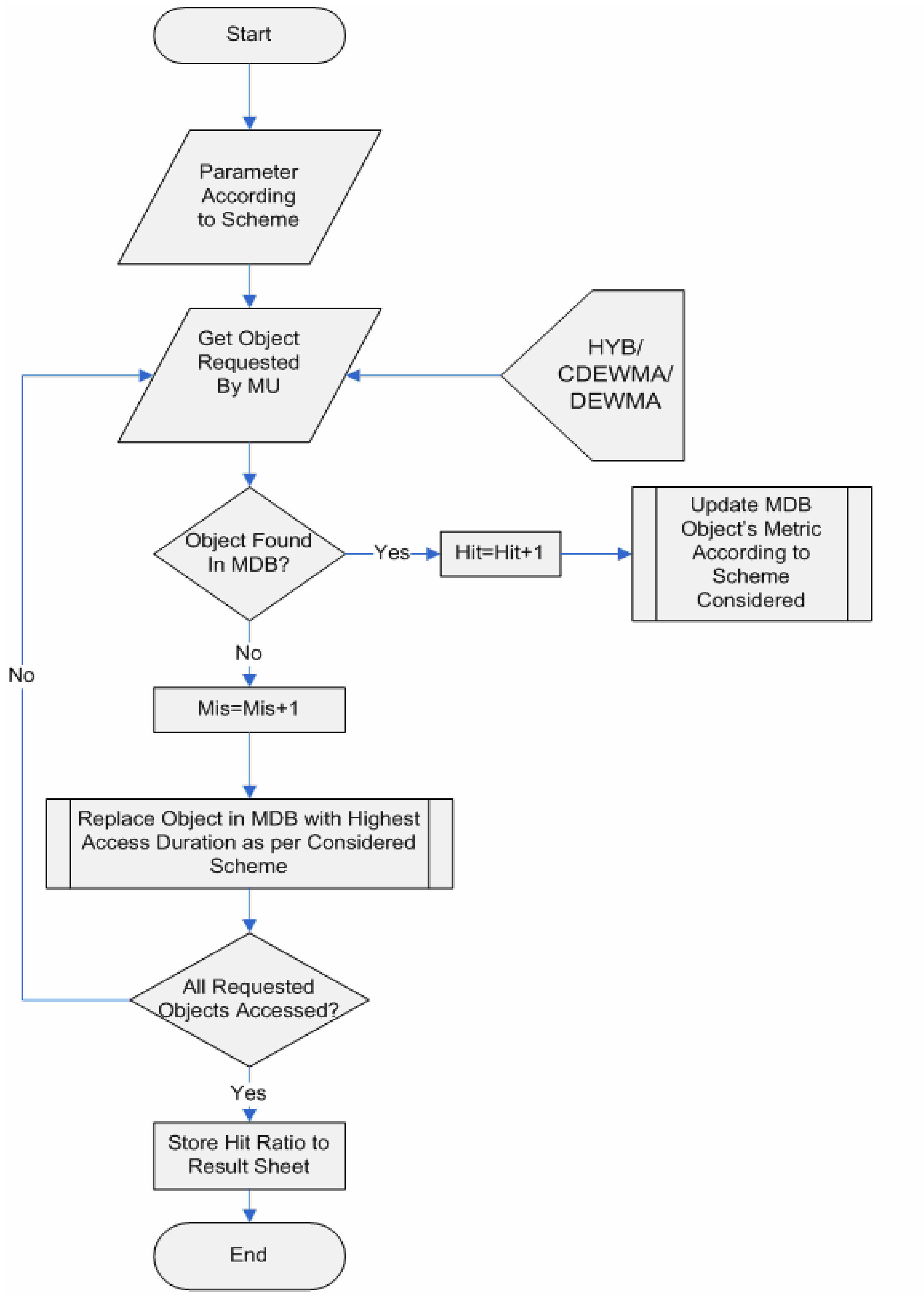


Figure 1.1. Flow Chart of Main Scheme

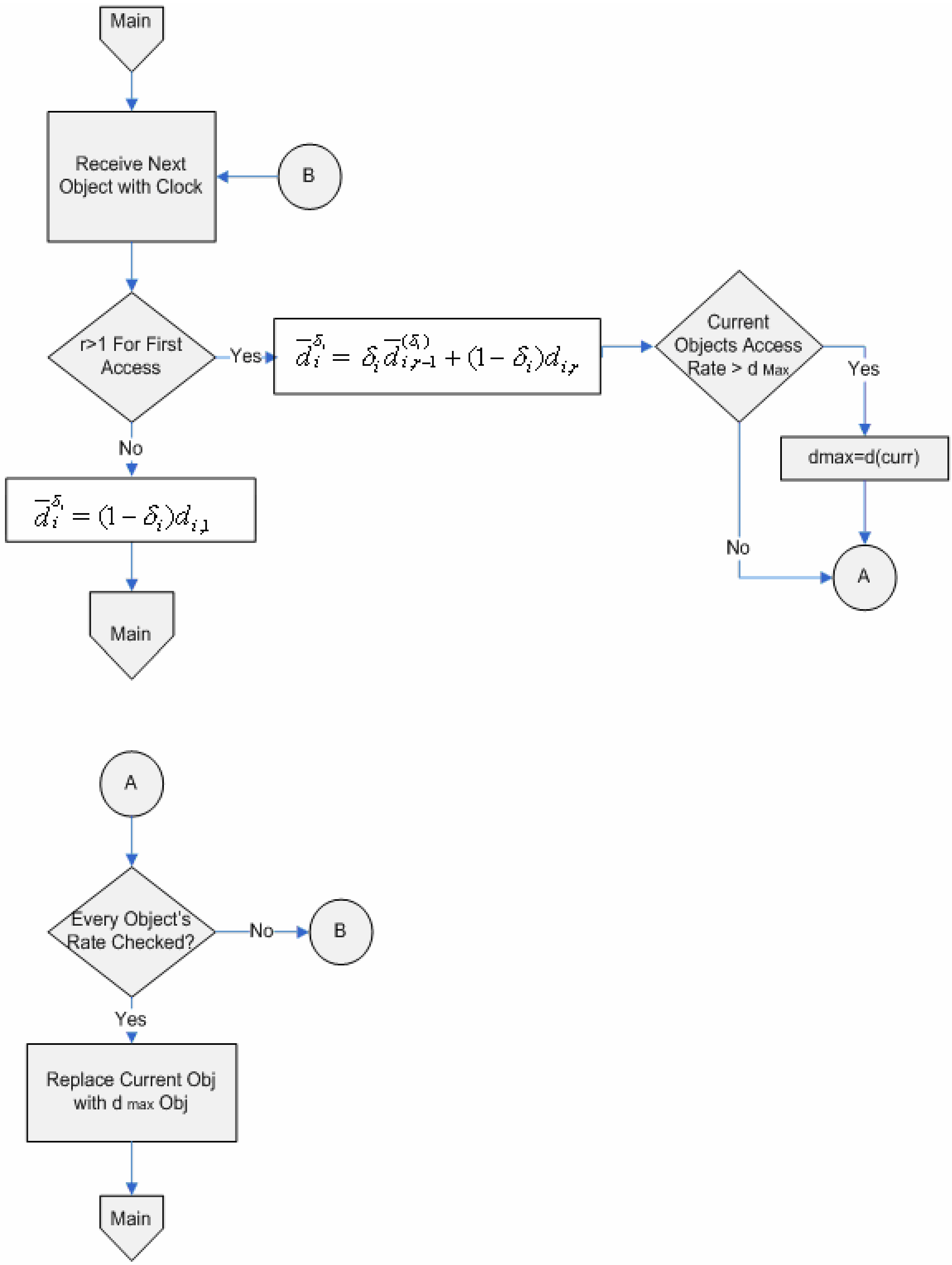
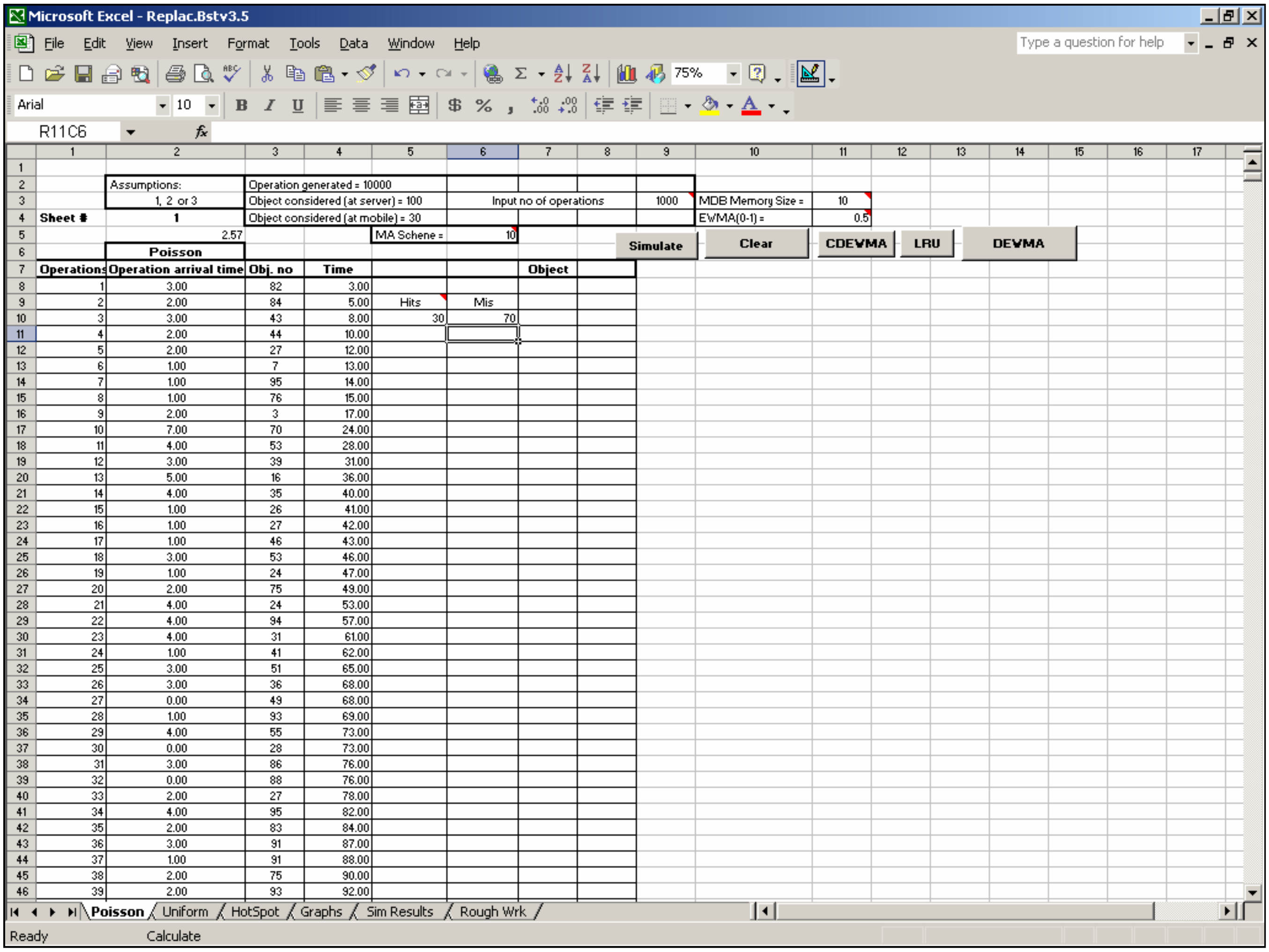


Figure 1.2. Flow Chart of DEWMA Scheme



Operations	Operation arrival time	Obj. no	Time
1	3.00	82	3.00
2	2.00	84	5.00
3	3.00	43	8.00
4	2.00	44	10.00
5	2.00	27	12.00
6	1.00	7	13.00
7	1.00	95	14.00
8	1.00	76	15.00
9	2.00	3	17.00
10	7.00	70	24.00
11	4.00	53	28.00
12	3.00	39	31.00
13	5.00	16	36.00
14	4.00	35	40.00
15	1.00	26	41.00
16	1.00	27	42.00
17	1.00	46	43.00
18	3.00	53	46.00
19	1.00	24	47.00
20	2.00	75	49.00
21	4.00	24	53.00
22	4.00	94	57.00
23	4.00	31	61.00
24	1.00	41	62.00
25	3.00	51	65.00
26	3.00	36	68.00
27	0.00	49	68.00
28	1.00	93	69.00
29	4.00	55	73.00
30	0.00	28	73.00
31	3.00	86	76.00
32	0.00	88	76.00
33	2.00	27	78.00
34	4.00	95	82.00
35	2.00	83	84.00
36	3.00	91	87.00
37	1.00	91	88.00
38	2.00	75	90.00
39	2.00	93	92.00

Figure 2.1. Simulation program

operations, that is the most realistic case for MDB.

Simulation assumes that there are total 10,000 operations generated by mobile client which are accessing total of 1000 object available at sever to be accessed by mobile client but mobile can store only very few of them locally.

There are two versions of simulation, over which each scheme is measured, one for small size memory and other for large size memory, which are capable to store only 30 objects and 10 objects from 1000 objects respectively. Priority weight δ is taken 0.5 for simulation purpose, which could be 0.1 in real cases where priority is known to be given.

We have run many passes of simulation with different parameter values to evaluate different effects such as priority weight δ, maximum no of inter object access durations w, no of operations, and size of mobile database etc, and therefore only significant results are presented in figure 3.1 and table 1 for small memory and figure 3.2 and table 2 for large memory. The results are typically generated after many iterations of the simulation (see figure 2.1).

4.2 Resulted effectiveness of simulated schemes

Summarizing the simulation results every scheme has typical results for a specific situation such as Moving Average MA10 is fail to adapt quickly to changes in object calling pattern for all configurations, where as EWMA responses only to those objects who are continuously accessed in frequent manner with short interval and does not update their rate dynamically if they are no more accessed or those who were newly accessed, therefore new objects achieve similar access rate against old once are same. That's why it is appreciated in hotspot with large database memory and in uniform arrival with small memory. Hybrid is almost following EWMA.

As per our simulation study, applying light weight conventional LRU scheme at object level instead of per-page level provides much better results which are significant in Poisson arrival with both small and large size memory.

CDEWMA is somewhere between LRU and DEWMA, well prominent in uniform arrival with large memory, and finally few configurations, which are very crucial to our proposed

OOMDB [1], are sufficiently supported by DEWMA scheme providing outstanding results in hotspot and poisson arrivals for small memory size, and also acceptable results in poisson and uniform arrivals for large memory size, therefore these results are enough to support our assumption that DEWMA adapt quickly and dynamically to changes in object calling pattern for small memory size mobile databases and resulted much suitable for our mobile database, especially where memory size is small, object access pattern is mostly similar to hotspot and some time to poisson which is most realistic configuration for MDB.

Table 1: Hit Ratios for Small Size Memory.

	MA10	EWMA	Hybrid	LRU	CDEWMA	DEWMA
Hotspot	7.07	7.17	7.17	38.39	35.55	40
Uniform	10	10.50	10.50	9.89	9.79	9.79
Poisson	8.58	9.29	9.29	10.30	9.79	9.89

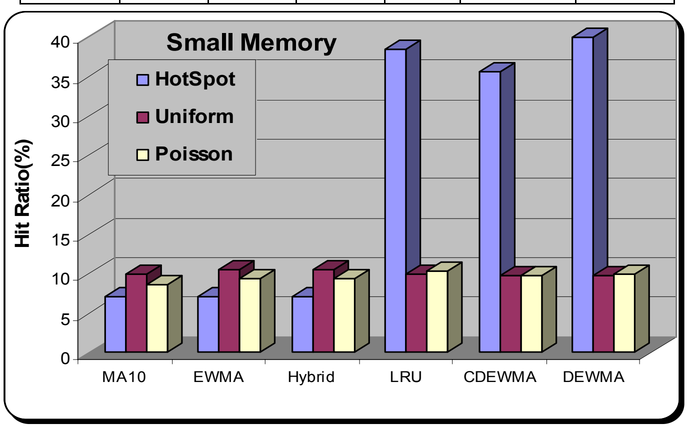


Figure 3.1: Graph Showing Simulation Results for Small Size Memory

5. Conclusions

Though there are other summarization schemes [7] available, but we presented a completely distinct and suitable scheme to identify minimum set of mostly required object for replication at MH through our light-weight object oriented mobile database model. It ensures high data availability at mobile host during disconnected operations, under wireless medium, which is vulnerable to frequent disconnection and provides limited bandwidth. Data availability is considered extremely important that is ensured by our proposed object replacement scheme DEWMA, suitable for mobile database computing paradigm and limited memory mobile phones, which is making our database model suitable for performance related concerns.

The final version of our scheme, DEWMA exhibits all those features like: systematically maintain object calling rate, earlier rates must be minor additive, dynamically update the rate to reflect the change by some intelligent means when they are not called for long period of time, provide priority to recently called objects. It is therefore fulfilling all requirements and giving required performance in small memory mobile phone.

Finally presented simulation study proves the effectiveness of our scheme DEWMA against other related schemes and also describe the generated results to form productive conclusion of the simulation.

Table 2: Hit Ratios for Large Size Memory

	MA10	EWMA	Hybrid	LRU	CDEWMA	DEWMA
Hotspot	58.39	83.29	81.66	69.24	69.25	69.33
Uniform	28.24	30.16	28.81	30.30	30.59	30.39
Poisson	26.08	30.25	27.72	30.82	30.12	30.77

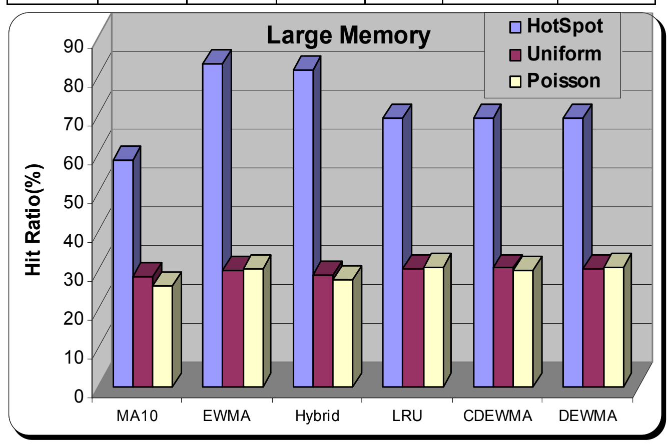


Figure 3.2: Graph Showing Simulation Results for Large Size Memory

6. Future research directions

The research work is not ended, rather more advance, adoptive, and light weight predictive model is expected by using fractals, to efficiently promote mobile database computing even for small mobile phones.

6.1. Fractals to predict object access pattern:

To study the underlying dynamics of the system we use the data generated by the system. This data is the record of the system state at discrete time intervals. In cache memory management the biggest challenge is to keep the most used objects in smallest cache as possible and we should have those objects, which are accessed most of the time with fewer replacements possible. In this scenario, we have the data of small number of accesses on the basis of this we have to decide about the objects to keep in the cache [5, 6]. Fractals may also be looked into as a possibility for checking the utility of the objects and then decide about its status of replacing.

Some features of fractals are such, which helps us in applying this concept:

1. It is basically random in nature as the most of the access patterns of the objects
2. It works on the concept of self-similarity; each object has the probability to be accessed again.
3. Fractals also work in short data sets. In cache we have a very big issue of cache size
4. Since object are accessed independent of time (stationarity), which is a necessary requirement for fractals
5. Artificial neural network already being used in computer networking successfully, here the weights of neurons layer can be estimated using fractal dimensions. So for ANN the requirement of large data training set can be compensated.
6. This techniques is scale invariant, so can also be applied on any scale of application as efficiently as on cache management.

   This also implies that instead of applying it one a small cache we can work on a large-scale system and then scale down the whole approach as required for application.

7. References

[1]. A. Khubaib, M. Nadeem, "Mobile Database Computing Performance Related Issues and Replication with Recommendations" JISR, Vol. 3, No. 1, Dec 2005.

[2]. Rajeswari Malladi, Karen C. Davis. "Applying Multiple Query Optimization in Mobile Databases," hicss, p. 294b, 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9, 2003.

[3] Si A., and Leong H.V. "Adaptive caching and refreshing in mobile databases.", Personal Technologies, Sep. 1997.

[4] Iman R. L. "A Data-based Approach to Statistics," International Thomson Publishing, 1994.

[5]. D. Thiebaut, "On the Fractal Dimension of Computer Programs and its Application to the Prediction of the Cache Miss Ratio," IEEE Trans. Computers, Vol.38, No. 7 July 89.

[6]. J. Voldman, B. Mandelbrot, L. W. Hoevel, J. Knight, and P. Rosenfeld, "Fractal Nature of Software-Cache Interactions". IBM J. Research and Development, Vol. 27, pp. 164-170, Mar. 1983.

[7]. Darin C., and John. F., R. "Context-Sensitive Mobile Database Summarization". ACSC2003, 26th ACSC Conf. in Adelaide, Aust., Reproduction in Conf's. Research and Practice in IT, Vol. 16. 2003.