Improving Data Replication in Mobile Grids using Mobility Prediction

Hala M. Abbas*, Iraky Khalifa**
*(Faculty of Computers and Information, Helwan University, Egypt)
** (Faculty of Computers and Information, Helwan University, Egypt)

ABSTRACT

Data replication is a technique used in mobile grid environments to enhance system reliability by increasing data availability and reducing access latency and network utilization. Due to the dynamic nature of mobile grids, replica placement becomes one of the most important challenges. It has a great impact on the performance of the whole system. Efficient placement strategies should consider both system capacity and user mobility to avoid increasing replication cost. In this paper, we integrated two simple replica placement algorithms (greedy and random) into an accurate mobility prediction to investigate the effect of this integration on the accuracy of placement. Our simulation results showed that the performance of replica placement improved by considering users' current and future locations.

Keywords – **Data Replication, Mobile Grids, Mobility Prediction, Replica Placement.**

1. Introduction

The main role of Grid in order to solve large scientific problems and workloads is to combine a large amount of geographically dispersed computing resources and servers in a heterogeneous environment as if they are located at a single site [1]. Grids can carry out tasks range from data storage to complex calculations which are too intensive to be handled by a single machine by using the wasted CPU cycles of computers in a network and providing common and general purpose interfaces and protocols to all these resources. Early grids were implemented with high performance and physically fixed resources connected through reliable and high speed networks, but emerging grids extended the scope of these resources to include sensors and wireless devices to acquire information in order to solve real-time problems. Because they serve as a basis for today's anytime, anywhere computing paradigm, mobile wireless devices have won strong support from users. But these devices were restricted to small amount of battery and poor performance which makes its usage in solving computing problems very difficult. Grid computing can be used to overcome these constraints by pooling the resources of several willing and cooperative mobile devices to resolve a computationally-intensive task. This integration led to the new research field of mobile grid.

Mobile grids addresses the mobility issue by providing a platform which enables both fixed and mobile users to have access to both fixed and mobile grid resources, at the same time it should manage this transparently to the users and possibly to the application itself taking into account the system performance. As a result, improving the performance and increasing reliability of mobile grid systems gain significant interest in the research field.

Data replication is the process of creating multiple data copies for remote applications in order to quickly access and process remote data. It helps avoiding data transferring overhead, improving the efficiency of data access and the capability of fault tolerance, minimizing user latency and thus providing better Quality-of-Service (QoS) [2]. To ensure efficient replication, many problems should be resolved such as deciding which data to replicate, and for how long to keep it (data replacement techniques), where to keep replicas (replica placement techniques), and how to reduce or even avoid the inconsistency between replicated data. In fact, replica placement is one of important challenges to improve performance and good placement strategies can result in significant performance gains.

In mobile grid environments, grid proxies are connected to mobile hosts via wireless access points such as base stations and wireless routers. Maintaining data objects in the original data servers to handle service requests increase workload and bottlenecks. Placing data requested by mobile users on nearby grid proxies can prevent these bottlenecks and provide shorter data access delays and thus better Quality-of-Service (QoS). In contrast to traditional wired grids, data placement decisions in mobile grid environments should take into consideration user mobility to avoid increasing replication cost by incurring unnecessary replications. A considerable amount of work has been done in replica placement which considers not only user mobility but also availability of grid proxies, load balancing of replicas, applying fault tolerance mechanisms in case of replica node failure and meeting users' QoS requirements. But it is found that all these strategies place replicas in a reactive manner (i.e., decisions for moving and placing replicas are based on previously observed data requests). A better solution is to make data placement decisions proactive by using mobility prediction. That is, placing the requested data prior to user movement and in the direction of the predicted next location of the mobile user and thus, the user will observe shorter data access delays and get better QoS which in turn will improve the overall system performance.

In this paper, we investigate the effect of using simple replica placement algorithms in combination with an accurate mobility predictor on the performance of mobile grid system. We aim at reducing data access cost by enhancing placement accuracy. Two replica placement algorithms have been used (popularity based greedy data placement and popularity based random data placement)[3]. These algorithms have been regarded as efficient solutions since they select powerful grid proxies to place the most popular users' data requests but they are still reactive. They don't consider user mobility in the placement decision. So the main contribution of this paper lies in how we can find the user's next location using mobility prediction to guide the placement process in the direction of user's movement, and to what extent the placement accuracy will be affected by the prediction accuracy.

This paper is organized as follows. Section 2 presents related work on replica management. Section 3 demonstrates the proposed model. Experimental results are presented in Section 4. Finally, in Section 5 we conclude our paper.

2. Related Work

One of the most important challenges in mobile grids is replica management [4], although it provides an efficient way to access and process remote data and helps to reduce user's latency, it adds additional cost for optimizing data placement, replacement and ensuring consistency. For consistency management, [5] proposed a hybrid approach that combines the pessimistic and optimistic approaches and benefits from their advantages to reduce the conflicts between replicas and the degree of divergence. For replica replacement [6] proposed an algorithm based on value-cost prediction, it was divided into two stages. The first stage, calculates replica value by prediction to make sure that low value replicas are replaced by high value replicas. In the second stage, the best network link is chosen through the prediction of network bandwidth to reduce the replacement cost. [7] Proposed a decision-tree-based predictive file replication strategy that predicts files' future popularity based on their characteristics on the Grids. [8] Also proposed an arithmetic method that calculates replica value to enhance data access.
Most of the work that has been done for the replica placement problem focused on wired grid, Proportional Share Replication a heuristic algorithm was proposed in [9], but this algorithm doesn't provide an optimal solution. [10] Proposed a set of placement algorithms that consider user requests and network latency, they used p-median, p-center and multi-objective models to select the best candidate sites to place replicas. The drawback of these algorithms is that it uses a static number of replication sites to host a replica. A dynamic replica creation and placement algorithm has been proposed in [11].

For mobile grid replica placement, QoS requirements of the mobile users as well as load balancing and communication cost acquired a great interest from the researchers to improve performance of mobile grid systems. [12] presented a two-step solution for QoS-aware replica placement in a tree based topology, the first step was a bottom-up dynamic programming approach used to select replica sets that satisfy the QoS requirements of mobile users and workload constraint of replicas, then, a binary search based algorithm was used to place these replica sets in mobile grid environments. It was a feasible and efficient algorithm, but it did not support general graph architectures. An improvement to [12] was proposed in [13] to consider the reliability problem by applying fault tolerance mechanisms in case of replica node failure. Also, [14] proposed an algorithm based on minimum spanning tree algorithm to find the path for placing replicas. For general graph architectures, [15] proposed an intelligent mobile data placement technique based on a two tier architecture that can effectively balance the data replication cost and the data access cost, and [16] proposed a flexible management system for dynamic placement of replicas that deals with limited topology knowledge and provides good adaptability to network dynamicity, although this algorithm is efficient but it is very complex, and it has been used with a mobility model in [17] to proactively place replicas.

3. Proposed work

Fig. 1 shows the architecture of an internet based mobile grid system which provides pervasive access to internet resources. In this system the mobile user can directly access internet data kept in the nearby grid proxy (if the data is not exist in that proxy, a data replication process is made by coping the requested data through the wired network). Due to users' movement through the network, a single data request by a mobile user may be serviced by multiple proxies during the service lifetime. As shown in the figure, the system consists of three key elements: mobile users sensed by access points, data items that represent mobile users' requests and grid proxies that are used for replicating data items to serve access requests from mobile users. The mobile replica placement problem in such a system is how to make decisions about replicating a requested data item and find an optimal proxy set that satisfies users' requests and reduces access delays.

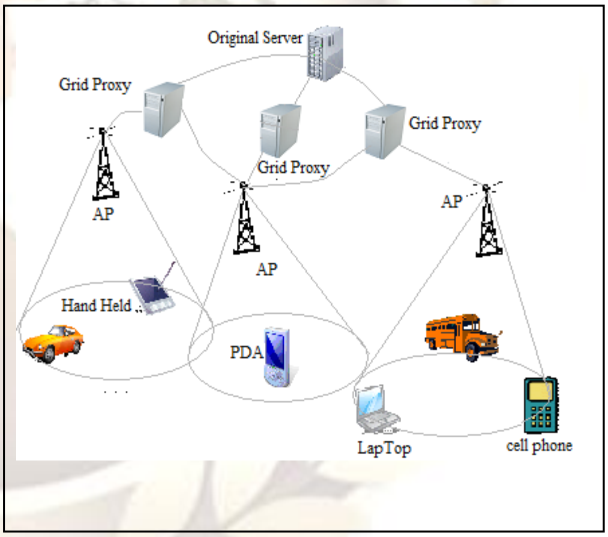


Figure 1: a mobile grid system.

In the following subsections, we will illustrate the functionality of the greedy and random data placement and how it can be integrated into our mobility predictor.

3.1 Greedy and Random Data Placement

Given historical data access patterns for mobile users, the greedy algorithm works as follows: all users' data accesses are ranked according to their popularity every T time period, then the most powerful grid proxy (the one with available network bandwidth and with enough storage) is greedily selected to place replicas for the most popular data request. The random algorithm like the greedy is based on data popularity but the proxy selection is random. An extension to these concepts is introduced in this work to include the mobility information of the users.

Let $M = \{m_i\}(1 \leq i \leq |M|)$ be the set of mobile users (|M| is the total number of mobile users), $D = \{d_j\}(1 \leq j \leq |D|)$ be the data items requested by mobile users, and $P = \{p_k\}$ $(1 \leq k \leq |P|)$ represents the group of grid proxies (including the original server) which have a set of resources *R* such as storage and bandwidth. We assume that we have a data access table *DAT* for the mobile users (a table that contains the predicted users' data access requests ranked according to their popularity), and $DAT_{i,j}$ determines that mobile user mi requested data item dj, Hi (L_{c-2}, L_{c-1},L_c) is a history set that contains the current and the past two locations which will be used in the next location predictor (illustrated in the next subsection). The greedy placement procedure is introduced in Fig. 2, it begins by collecting the locations information for all users and inputs this information to our mobility predictor to get the next n locations. The next step is to put the popular data in the direction of users' movement by finding the set of nearby proxies to the predicted next locations the replication is done if the duration for which the users will remain in these locations is greater than a threshold value *val* proportional to the duration of sessions. As we mentioned previously the random placement is the same but powerful proxy is selected randomly.

```
Repeat
  For each m_i ∈ M  Do
    Get H_i = (L_c-2, L_c-1, L_c ), DAT_i
    Find {L_c+1, ..., L_c+n} from the predictor
    For each predicted location Do
      Determine the waiting duration
      Find the nearest proxies set from P
      If (duration (L_c+1) > val)  Then
        Greedily select more powerful proxy
        Replicate the most popular d_j ∈ DAT_i
      Else
        Do the same for L_c+2 ,...,L_c+n
      End
    End
  End
Every T time period
```

Figure 2: the greedy placement procedure.

3.2 The Mobility Predictor

We introduced a mobility predictor in [18], it is a straight forward mobility predictor that can predict the future locations of mobile users and the duration for which the users will remain in those locations and does not need a training step like Markov models. It is based on a real-world dataset (The UCSD dataset which includes users' movement traces collected during an 11 week trace period for 300 mobile users issued by PDA's which are sensed by access points to build the users' history set)[19]. The predictor includes two phases: a preprocessing phase to convert the raw data to a form suitable for the next phase. Fig. 3 and Table 1 present the format of the original dataset and a sample of the resulting data traces after preprocessing respectively.

```
USER_ID,SAMPLE_TIME,AP_ID,SIG_STRENGTH,
AC_POWER,ASSOCIATED
123,09-22,00:00:00,359,8,0,0
123,09-22,00:00:00,363,5,0,0
123,09-22,00:00:00,365,11,0,1
191,09-22,00:00:00,355,31,0,1
101,09-22,00:00:00,353,8,1,0
101,09-22,00:00:00,362,30,1,1
129,09-22,00:00:00,369,31,0,1
156,09-22,00:00:00,360,19,1,1
```

Figure 3: a portion of the UCSD dataset.

Table 1: Sample of records for user #200.

User	Start Time of Session	Associated Access Point	Duration of Session
200	26/09/2002 07:18:27 pm	93	0:01:20
200	26/09/2002 07:19:47 pm	48	2:03:45
200	26/09/2002 09:23:32 pm	352	0:01:00
200	26/09/2002 09:24:32 pm	354	2:18:33
200	26/09/2002 11:43:05 pm	352	0:54:00
200	30/09/2002 02:52:20 pm	352	0:00:22
200	02/10/2002 10:01:36 pm	352	0:01:42
200	03/10/2002 09:34:59 am	352	3:08:47

The next phase is the main location predictor it uses the preprocessed mobility patterns and the user's current mobility information to predict the next locations of the mobile user. The following steps illustrate how the predictor operates.

1. Construct a set SX that contains the past i locations and the current location Xc of a mobile user.
2. Match the sequence SX with all history sequences (preprocessed mobility patterns) to generate a history set SH which contains the past i locations, the current location and the next N locations.
3. Add the resulted sequences to a prediction set Plist.
4. Check if Plist is empty (i.e., no match was found), then the session with the shortest duration is removed from the past i locations. The history set is searched again until Plist contains at least one historical sequence.
5. Predict the next N locations by calculating the probability for each of the resulted locations according to how many times it appears in the history set.

We evaluated the prediction accuracy of our algorithm considering the current and the last two positions and compared it with a third order Markov predictor. The results showed that our predictor achieved greater prediction accuracy than markov model, also we have reached that the predictor has the following advantages over markov model

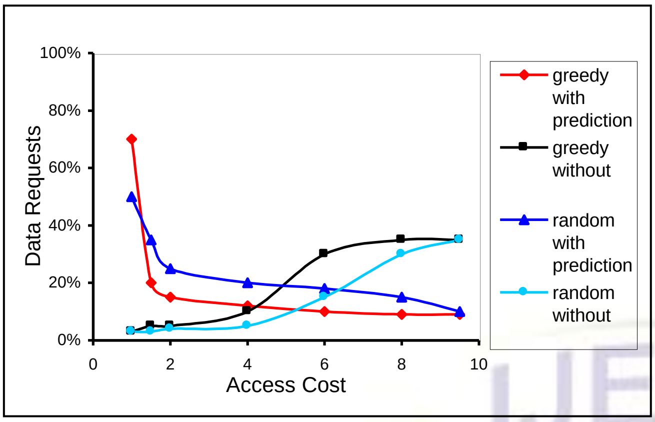
1. We can increase the number of past locations easily unlike markov model (increasing the model order leads to increase the number of transitions).
2. No training step is needed. Thus, new locations and history traces can be included faster unlike markov model (Adding new historical data needs rebuilding the movement patterns).

4. Experimental Results

As we mentioned previously, the mobility predictor is based on The UCSD dataset, it includes a file containing the access points' locations. We modeled a network and used this file to determine the locations of 30 distributed proxies to serve the 200 access points and one data server that keeps the original copies of all data objects. We configured each proxy to have 50 GB storage and 50 Mbps bandwidth, for users' data requests we followed our assumption in which each user has a data access pattern for data items ordered by their popularity and each data request required from 400 KB to 200 MB disk space for replication and consumed from 300 Kbps to 1 Mbps network bandwidth. In order to use the mobility predictor, we determined the current and the last two locations for each user. We used the following factor as the basis to evaluate the performance of our algorithm: data access cost which refers to the distance from the access point of a mobile user to the selected proxy for replication, and the placement accuracy is evaluated by the mean distances for all users during the whole service period. We assumed that the data access cost to the original server is the highest.

In order to validate our algorithm we performed a set of simulation experiments to compare the placement accuracy of greedy and random algorithms with and without the mobility predictor, Fig. 4 shows the results, it is obvious that greedy and random algorithms with prediction achieve lower data access cost since we reduced the proxy selection area toward the moving direction of the user which in turn reduced the distance between user requests and the replica locations. We run our simulation under different numbers of grid proxies with different configurations to get the average distance for the placement strategies. As Fig. 5 illustrates, the distance decreases if we increase the convergence between users and proxies. Finally it was clear that the prediction accuracy affects the accuracy of placement since it is considered the base which guides the placement process.

Figure 4: access cost comparison



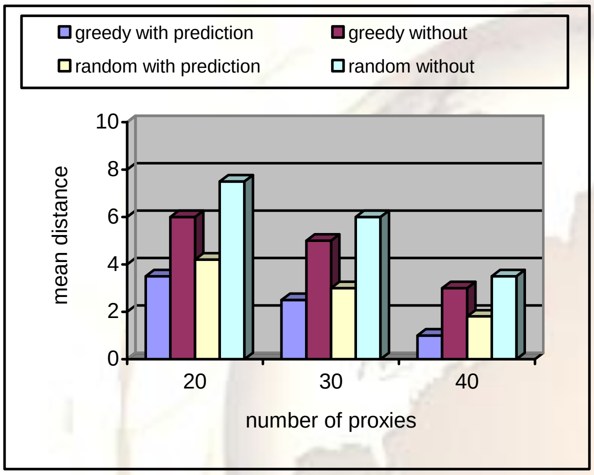


Figure 5: overall distance for different configuration

5. Conclusion

In this paper we addressed the problem of replica placement in mobile grid environments, we used two existing algorithms (greedy and random placement) and compared its performance against an integration between them and a developed mobility predictor based on real world data traces. Our simulation results showed that the performance of replica placement could be improved significantly if the prediction accuracy increased.

References

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid", *International Journal of Supercomputer Applications*, 2001.

[2] V. Tripathi and Y. Breitbart, "Replica management in content distribution networks", *IADIS International Journal on WWW/Internet Vol. 4, No. 1*, 2009, pp. 43-54.

[3] M. Karlsson and C. Karamanolis, "Choosing replica placement heuristics for wide-area systems". *Proc. of ICDCS*, 2004.

[4] V. Tripathi and Y. Breitbart, "Replica management in content distribution networks", *IADIS International Journal on WWW/Internet Vol. 4, No. 1*, 2009, pp. 43-54.

[5] G. Belalem, et al, "Consistency Management of Replicas in Wireless Grid Environment", *international Journal of Grid and Distributed Computing Vol. 3, No. 4*, December, 2010.

[6] T. Tian et al: "A Replica Replacement Algorithm Based on Value-Cost Prediction", *Proc. of 11th international conference on CSCWD*, April 2007.

[7] C. Liao et al: "Predictive file replication on the Data Grids ", *International Journal of Grid and High Performance Computing, Volume 2, Issue 1, IGIP*, 2010.

[8] S. Bin and D. Yueli, "A efficient data replica replacement algorithm for Mobile Grid", *Proc. of 2nd International Conference on Advanced Computer Control (ICACC)*, 2010.

[9] J. H. Abawajy, "Placement of file replicas in data grid environments". *Proc. of ICCS 2004, Lecture Notes in Computer Science (LNCS), Vol.3038*, 2004, pages 66–73.

[10] M. Rashedur, et al, "Replica placement strategies in data grid", *Journal of Grid Computing, 6*, 2008, pp. 103–123.

[11] K. Sashi and Dr. A. Thanamani, "Dynamic Replica Management for Data Grid ", *International Journal of Engineering and Technology, Vol.2, No.4*, August 2010.

[12] X. Fu et al, “A replica Placement Algorithm in Mobile Grid Environments”. *Proc. of the International Conferences on Embedded Software and Systems*, 2009,

[13] L. A. Elizabeth, V.M. Sivagami, "Yet another Improvement in RSFT Algorithm in Mobile Grid", *International Journal of Computer Applications, Vol. 5– No.4*, August 2010, pp. 0975 – 8887

[14] T. Senthilnathan and T. Purusothaman, "A Novel Approach for Optimized Multi Site Replica Placement in Mobile Grid", *European Journal of Scientific Research Volume 63 Issue 3* November, 2011

[15] Y. Huang and N. Venkatasubramanian, " MAPGrid: A New Architecture for Empowering Mobile Data Placement in Grid Environments", *Proc. of 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*, 2007.

[16] J. Gossa et al, " FReDi: Flexible Replicas Displacer", *Proc. of International Conference on Networking, ICN06* , Los Alamitos, CA, USA 2006, pp. 16–25.

[17] J. Gossa et al, "Proactive Replica Placement Using Mobility Prediction", *Proceedings of the 9th International Conference on Mobile Data anagement Workshops*, 2008.

[18] I. Khalifa, H.M. Abbas, "Mobility Prediction In Dynamic Grids", *international journal of Computer and Information Science, Vol. 5, No. 3*, May 2012, pp.90-96.

[19] M. McNett," UCSD Wireless Topology Discovery Trace ", *available on: http://sysnet.ucsd.edu/wtd.*