MAPGrid: A New Architecture for Empowering Mobile Data Placement in Grid Environments

Yun Huang, Nalini Venkatasubramanian and Yang Wang
School of Information and Computer Sciences
University of California, Irvine, CA 92697
{*yunh, nalini, yangwang*}@*ics.uci.edu*

Abstract

The rising popularity of mobile applications and devices has brought about an enhanced interest in infrastructure support for mobile computing. Our work focuses on the development of a mobile grid infrastructure called MAPGrid (Mobile Applications on Grids), where grid resources are exploited as proxies to enable advanced mobile applications. However, intermittent availability of grid resources presents challenges for data-intensive mobile applications. In this paper, we propose novel methodologies for placing mobile data on grid proxies. We introduce a notion of two-tier architecture for MAPGrid, where the upper tier captures grid related features and the lower tier represents features associated with mobile environments. We further develop an intelligent mobile data placement mechanism that effectively balances tradeoffs between replication cost and data access cost by leveraging knowledge of grid availability and mobile data request patterns. Through extensive experimentation, we illustrate the superiority of the proposed techniques over several popular data placement strategies.

1 Introduction

The next generation of mobile applications such as mobile multimedia and gaming are data and resource intensive and must execute on devices that are limited in resources (memory, storage, battery power). A recent trend is to utilize resources in the path of service, i.e. network proxies, to provide localized computation and storage to enable distributed and ubiquitous service availability [20, 24]. Research efforts have demonstrated that the Grid infrastructure provides an ideal setting to enable proxy-based mobile applications [14,16,19]. In a mobile grid environment, grid proxies connect to mobile hosts via wireless access points such as wireless routers and base stations. Placing data requested by mobile users on nearby grid proxies can significantly reduce the load and thus prevent bottlenecks on the original data servers that maintain data objects and handle service requests. Besides, mobile users will experience shorter data access delays and thus better Quality-of-Service (QoS). Proxy-based techniques can also be applied on grid machines to overcome innate constraints of mobile devices, e.g., offloading tasks and QoS adaptations [20] can alleviate energy tensity when running computational-intensive applications.

In mobile grid environments, data placement decisions that made prior to the service request can affect resource discovery decisions at the time of service; this in turn influences the runtime system performance substantially. The overarching goal of this article is to address the issues of data placement on grid proxies for mobile applications. Many challenges arise due to the dynamic and heterogeneous nature of mobile grid environments. Firstly, the fact that mobile users can move freely and unpredictably in a wireless network may result in varying network connectivity. A grid proxy that is optimal for mobile users at a time may not be optimal throughout the service duration. In order to reduce network bandwidth consumption and to improve end-to-end QoS performance, a single mobile request may be serviced by multiple grid proxies during its whole service period. Leveraging user mobility patterns for devising efficient data allocation schemes has been demonstrated by [21], where mining techniques are applied on individual user's moving patterns. However, new challenges emerge from intermittent availability of heterogeneous grid proxies. Essentially, long term placement problems can not be directly solved by short-term caching techniques, e.g., [4, 10]. Resource allocation also needs to take into account varying resource capacities at a proxy for overall resource utilization (e.g. storage and network resources) improvement. Existing methods such as [23] and [13] store replicas on mobile hosts in wireless networks. These methods can be applied to the situation where resources of mobile devices are harnessed and low-power devices are integrated into a grid [22]. In this paper, we focus on how to place data on fixed grid proxies. Our assumptions on the availability of the resource providers (fixed grid machines) and the connectivity between mobile users and the resource providers are different from the ones in the aforementioned methods (where mobile devices are actually the resource providers). Therefore, their approaches will not have optimal performance in our scenario.

A number of data placement heuristics have been proposed for proxy-based content distribution networks [9, 17, 18], and we also developed data placement strategies [15] in the context of intermittently available of grid environment. However, these schemes neglect the mobility of users and services in their decision making and thus they may incur unnecessary replication operations and increase replication cost. In order to provide user better QoS services (lower data access cost) with lower system cost (lower data replication cost), both user mobility and intermittent availability information of heterogeneous grid proxies should be taken into consideration when making data placement decisions for mobile applications.

In the remainder of the paper, we concentrate on tackling the above unique challenges of mobile data placement on intermittently available grid proxies in such mobile grid environments. More specifically, in Section 2, we will introduce a two-tier architecture, MAPGrid, for maintaining both the intermittent availability information of grid proxies and the data request patterns of mobile users. In Section 3, we propose an intelligent MAPGrid-based data placement technique that can effectively balance the data replication cost and the data access cost. We evaluate our approach in Section 4, and conclude in Section 5.

2 The Two Tier Architecture of MAPGrid

The performance of applications in the mobile grid infrastructure is influenced by *proxy related* factors, e.g. capacity of grid proxies and *user related* factors , e.g. data request patterns. Data placement decisions, in particular, are affected by the (a) the intermittent availability of the grid resources and (b) e.g. user mobility patterns(i.e. mobility of requested data). Fig. 1 (A) shows a two tiered architecture that facilitates the separation of these two concerns. The upper tier (Grid Proxy Tier) contains information about intermittent proxy availability and the lower tier (Mobile Data Tier) captures data request patterns of mobile users. This architecture enables efficient collections of proxy-related and user-related parameters independently. We fuse the spatio-temporal information into a comprehensive interval tree based representation (as shown in Fig. 1 (B)), which is subsequently used to make data placement decisions.

Interval tree [5] based representations have recently

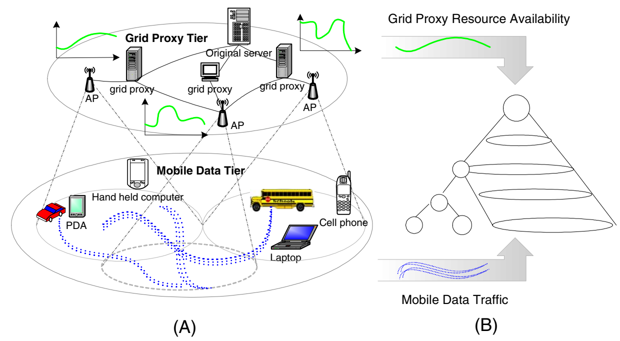

Figure 1. The two-tier MAPGrid architecture

gained popularity [8, 11]. A distinguishing aspect of the MAPGrid architecture is that the generation and maintenance of the interval tree is driven by the grid proxy features such as the availability and capacity of grid resources (Fig. 2 (A)). For efficient data representation, information about grid proxy availability and resources are stored from the top of the tree (Fig. 2 (B)) and the initial basic time intervals are defined by the grid availability according to the algorithm described by [1]. Information of Mobile Data Tier is initially collected at the bottom level of the tree for finer granularity, but it may be rearranged and be stored at higher level of the tree when data placement algorithm is executed. (We will present how to restore mobile data information in section 3.) Note that the generation of the interval tree is also based on two user/application defined parameters (i) a period that determines how often the trees are regenerated (i.e. reflecting the frequency of making data placement decisions) and the (ii) height of the interval tree. Each interval node contains two items. One is a grid proxy list (PList) including the grid proxies that are available for this interval. The other one is a Data Access Table of Mobile Users (MDT) which is a matrix shown by Fig. 2 (B). Because spatial information does not change frequently over time, we use an individual ID number to identify the spatial area of a wireless coverage. More specifically, the MDT contains information about the number of requests to data objects by each user at different wireless regions. According to information contained in the lower tier, correlations among a user's data request patterns over wireless regions and resource requirements of different wireless regions can be evaluated.

3 Mobile Data Placement on Intermittently Available Proxies

In this section, we will present how to leverage the MAPGrid to address data placement problems in such a mobile grid environment, and the problem we address is described as follows. Given knowledge of intermittent availability of

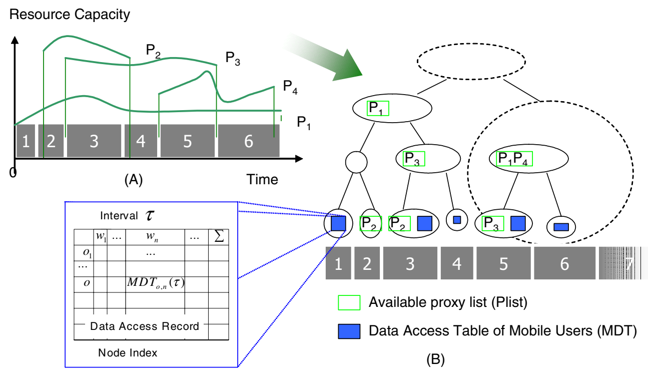

Figure 2. MAPGrid Data Representation: an interval tree-like data structure for capturing grid availability and data request patterns of mobile users

grid proxies, and users' historical mobility information, we aim at optimizing the network performance and the overall system resource utilization over time by: (1) determining the number of replicas for each user's requested data, (2) choosing when to replicate them and (3) selecting grid proxies for each replication, so that the total replication and data access cost will be reduced. The major contribution of this paper lies in how we tackle the issue of finding intermittent available grid resources to meet mobile users' data needs by applying spatio-temporal locality so as to balance the trade-off between replication cost and data access cost. We now develop a mobile data placement algorithm that determines the mapping between objects and proxies.

The high level procedure of the grid-based mobile data placement algorithm is as follows: from the first level of the tree until the leaf level, for each interval node, if there are proxies available for that period (i.e. the PList is not null), we attempt to create replicas of data objects requested in that period. To do this efficiently, we need to determine the interval-tree level at which the replication must occur. This level dictates the time at which replication must occur and the duration for which the replica will reside on the proxy. Note that the replica and proxy mapping at higher level of the tree indicates that replicas are created earlier and kept for a longer duration. Hence the choice of data objects to be replicated at higher levels must be done carefully. We apply statistical analysis techniques to select appropriate data objects for interval nodes. In particular, we define the notion of "regularity" of mobile data request patterns for an object in an interval tree node. If the regularity is high, the data object is chosen for replication. Otherwise, the data object will be considered for replication at lower levels of the interval tree. This selection will continue until either all predicted requests for this data object can be serviced by assigned proxies or available proxy resources for this period have been exhausted.

3.1 Analyzing Variance of Data Request Patterns

We now explain how to evaluate 'regularity" of data request patterns of mobile users by utilizing statistical techniques. Let τ_f be an interval node in the TARGET tree that spans a number of elementary intervals at the leaf level, as shown in Fig. 3. Each child interval node τ_{f_k} may contain data request patterns $\overrightarrow{MDT_o(f_k)}$ to data object o of a mobile user, where $1 \leq k \leq C$ and C is the number of Children of node τ_f. We compare them to recognize the regularity among data request patterns of the same mobile user. If they are similar, we can consider selecting grid proxies for the mobile user i at the node τ_f within its $PList$, instead of choosing grid proxies at each child node τ_{f_k} ($1 \leq k \leq C$). More specifically, the null hypothesis at stake is that data access patterns among all interval nodes are similar. We can use standard statistical techniques to test the hypothesis. For simplicity, we apply two-way ANOVA with F test [2] for illustration purpose. We can also use non-parametric alternatives to the above techniques, if the data is not normally distributed in practice.

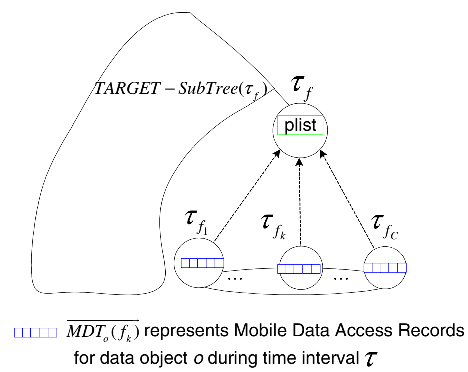

Figure 3. The subtree of τ_f

In this study, we let MDT_{o,n,f_k} be the number of accesses to one data object o from a specific wireless region w_n during one time period τ_{f_k}, then MDT_{o,n,f_k} is regarded as a dependent variable. While time interval f_k and wireless region n are regarded as independent variables. We use two-way ANOVA with F test to determine whether all rows of the data matrix shown in Fig. 4 have similar data access patterns. The F test will generate a F test statistic which follows the F distribution with (C-1) numerator and $(W-1)(C-1)$ denominator degrees of freedom. Given the F test statistic, a P-value ranging from 0 to 1 is also calculated. A small P-value indicates that real differences exist between at least some of the data access patterns across time intervals. If the

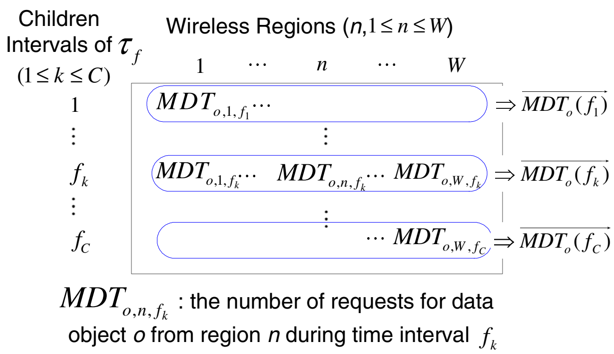

Figure 4. The data matrix

resulted P value is equal to or less than .05, we interpret that the data access patterns across time intervals are statistically different and thus do not select proxy available for this time period to replicate the data object. The F test statistic is calculated as follows: MSE is the mean square error from wireless region, and MSI is the mean square from time interval. MDT_{o,n,f_k} is the number of requests for data object o from wireless region n during time interval f_k. MDT bar $o, n.$ is the average number of requests for data object o of wireless region n, and similarly MDT bar $o.f_k$ is the average of time interval f_k. MDT bar $o..$ is the grand mean of all values.

$$
\begin{aligned}
F_{o,f} &= \frac{MSI_{o,f}}{MSE_{o,f}} \\
MSE_{o,f} &= \frac{\sum_{n=1}^{W}\sum_{k=1}^{C}(MDT_{o,n,f_k} - \overline{MDT}_{o,n.} - \overline{MDT}_{o.f_k} - \overline{MDT}_{o..})^2}{(W-1)(C-1)} \\
MSI_{o,f} &= \frac{W\sum_{k=1}^{C}(\overline{MDT}_{o.f_k} - \overline{MDT}_{o..})^2}{(C-1)}
\end{aligned}
\tag{1}
$$

3.2 Balancing Tradeoffs Between Replication Cost and Data Access Cost

The above data placement strategies are designed to balance tradeoffs between data replication cost and data access cost. The following lemma shows that the proposed approach maintains low data access cost without increasing data replication cost.

Lemma 1 *If $\overrightarrow{MDT(\tau_k)}$ ($1 \le k \le C$ and C is the number of children) are merged as $\overrightarrow{MDT(\tau_f)}$ into the parent node, as long as there is at least one proxy available for selection, data placement decision based on $\overrightarrow{MDT(\tau_f)}$ will not increase the replication cost and will not decrease the data retrieval cost compared to that decisions made by using each MDT elements of $\overrightarrow{MDT(\tau_k)}$ ($1 \le k \le C$). This is true when data placement is preformed from higher level to lower level.*

Proof 1 *If $p_k == p_f$ ($1 \le k \le C$), replication cost and data access cost remain identical. If $p_{k-1} \ne p_k$ ($2 \le k \le C$), as replication cost depends on the total size of the objects and one-time cost for replicating the data onto proxies, replication cost onto p_f will not be increased, as the total size of the data object is not changed; in fact, the one-time cost for replicating the data is reduced. However, the data access cost will not be decreased,*

$$
\begin{aligned}
&\overrightarrow{MDT(\tau_f)} \cdot \overrightarrow{AC_{p_f}(\tau_f)} \Rightarrow \sum_{1\leqslant k\leqslant C} \overrightarrow{MDT(\tau_k)} \cdot \overrightarrow{AC_{p_f}(\tau_f)} \\
&\Rightarrow \sum_{1\leqslant k\leqslant C} (\overrightarrow{MDT(\tau_k)} \cdot \overrightarrow{AC_{p_f}(\tau_k)}) \\
&\geqslant \sum_{1\leqslant k\leqslant C} (\overrightarrow{MDT(\tau_k)} \cdot \overrightarrow{AC_{p_k}(\tau_k)})
\end{aligned}
$$

otherwise, p_k ($1 \le k \le C$) will be replaced by p_f, then $p_{k-1} == p_k$ ($2 \le k \le C$) , which is in direct contradiction to the assumption $p_{k-1} \ne p_k$.

4 Performance Evaluation

In this section, we present and analyze our simulation results. In the simulation study, we model a cellular network system with 100 cells, 25 evenly distributed proxies and one data server that keeps the original copies of all data objects. Kondo et al. have studied resource availability in enterprise desktop grids [7]. In our simulation, For simplicity, without losing generality, we model proxy availability using service *TimeMap* [15], which provides information on when a grid proxy will be available during a day. Given space limitation, we show results of *Uniform TimeMap* model.

Mobility of individual user is characterized by applying the incremental mobility model [12], where mobile users can move at walking or driving speed in a closed coverage area. We model each user's data requests following the zipfian [17] distribution and popularity-unpopularity (PopUnPop) data object distributions (where data objects are classified into two groups) [6]. These two request patterns determine the number of requests that will be issued to different data objects by users. The duration of each data request varies from 30 minutes to 4 hours. We assume that the disk space required for replicating each data object is proportional to the service duration. The basic configurations include that each proxy has 100 GB storage and 100Mbps network bandwidth and that each data access from the mobile user will consume network transmission bandwidth ranging from 500 kbps to 1.3 Mbps.

We use the following metrics to evaluate the system performance. (1) We measure the distance from the selected proxy to the access point of a mobile host (the number of network hops) as the data access cost. Also because the access cost at any instance within the whole service period may change due to users' movement, for each user, we calculate his average access cost over his whole service duration. Note that the data access cost from the original server to a mobile user is defined to be higher than the maximum access cost among all proxies to the mobile user [3]. (2) Data replication cost is affected by lots of factors, e.g. data consistency, network and CPU utilizations. These are all counted when a data replication operation is performed. Thus without losing generality, we take the number of replication operations as the replication cost. (3) We also calculate the percentage of load that is saved from the original data server by servicing them on proxies.

We have performed a set of experiments to evaluate our proposed solutions under various configurations of the simulated system. Given space constraints, we will show the basic performance of our proposed mobile data placement (MDP), which compared with that of three selected data placement strategies, i.e. popularity-based greedy data placement (Greedy), popularity-based random data placement (Random) and caching on-demand technique (Caching). Our other results draw the same conclusions.

As shown in [17], popularity-based greedy data placement has been regarded as an efficient solution. The greedy policy was implemented as follows: at the end of each period, we first rank all users' accesses based on their popularity (given total number of accesses in a history window), then decide the number of replicas for each user's data by considering the system capacity. We greedily select a more powerful proxy (with more available network bandwidth) to allocate replicas for popular objects that are predicted to have more accesses in the next period of time with less storage requirement. The Random policy is also based on data popularity but it is different from the Greedy policy in that selecting a proxy for a replica is random. The third benchmark policy, an on-demand caching strategy, proceeds as follows: when a mobile user issues a request, the system will try to find a nearby cached replica. If there is no replica available on proxies, and the nearest proxy has resources to store a replica and to serve this request, a data replica will be placed on the selected proxy. Subsequent request can also reuse the replicated data. If there is no available proxy with sufficient resources to provide data services, the system will try to make a LFU cache replacement; if all replicated objects are in use that they can not be replaced, the mobile request will be connected to the original server. During service time, if mobility information is provided by users, the system can pre-schedule service connection along user's trajectory, otherwise, it will perform dynamic re-allocation according to users' current location.

Fig. 5 shows how these policies perform. To make a fair comparison, we evaluate CDF (Cumulative Distribution Function) of user data access cost over time by restricting replication cost of these four policies. Specifically, the total numbers of replicas made by different policies are kept same when these policies are executed with same system configuration, i.e. the same system load and data request patterns. For example, there are 212 replicas created for zipf data objects in Fig. 5 (1,3) and 100 replicas generated for Pop-Unpop data objects in Fig. 5 (2,4). According to Fig. 5 (1) and (2), the MDP results are shown to have lower data access cost than the other three policies under different conditions for the majority of requests. Besides, the other three policies (especially the caching policy) are more sensitive to data request patterns and system workload. As shown in Fig. 5 (3) and (4), our proposed MDP also prevails in saving more load from the original server than other policies.

5 Concluding Remarks

In this paper, we address the issue of placing data requested by mobile users on intermittently available grid proxies. We designed an intelligent data placement strategy to address tradeoffs between data replication cost and data access cost. We first introduced a two-tiered architecture, MAPGrid that separates concerns of intermittent availability of grid proxies and user mobility, we then developed an efficient data representation to collect information of the tiered architecture efficiently. Subsequently, we presented how to leverage the MAPGrid architecture in making intelligent proxy data placement decisions. More specifically, we apply statistical technique, e.g. a Two-Way ANOVA to analysis regularity of information captured by the MAPGrid. Our simulation results show that the proposed approach achieves better performance than several popular data placement strategies. We currently focusing on incorporating and evaluating proposed algorithms in MAPGrid middleware system that is under development at UCI.

References

[1] http://www.dgp.toronto.edu/people/jamesstewart/378notes/22intervals/.

[2] http://www.statsdirect.com/help/analysis_of_variance/twayrep.htm.

[3] I. D. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. In *12th ACM-SIAM SODA*, 2001.

[4] C.-Y. Chow, H. V. Leong, and A. T. S. Chan. Group-based cooperative cache management for mobile clients in a mobile environment. In *ICPP*, 2004.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

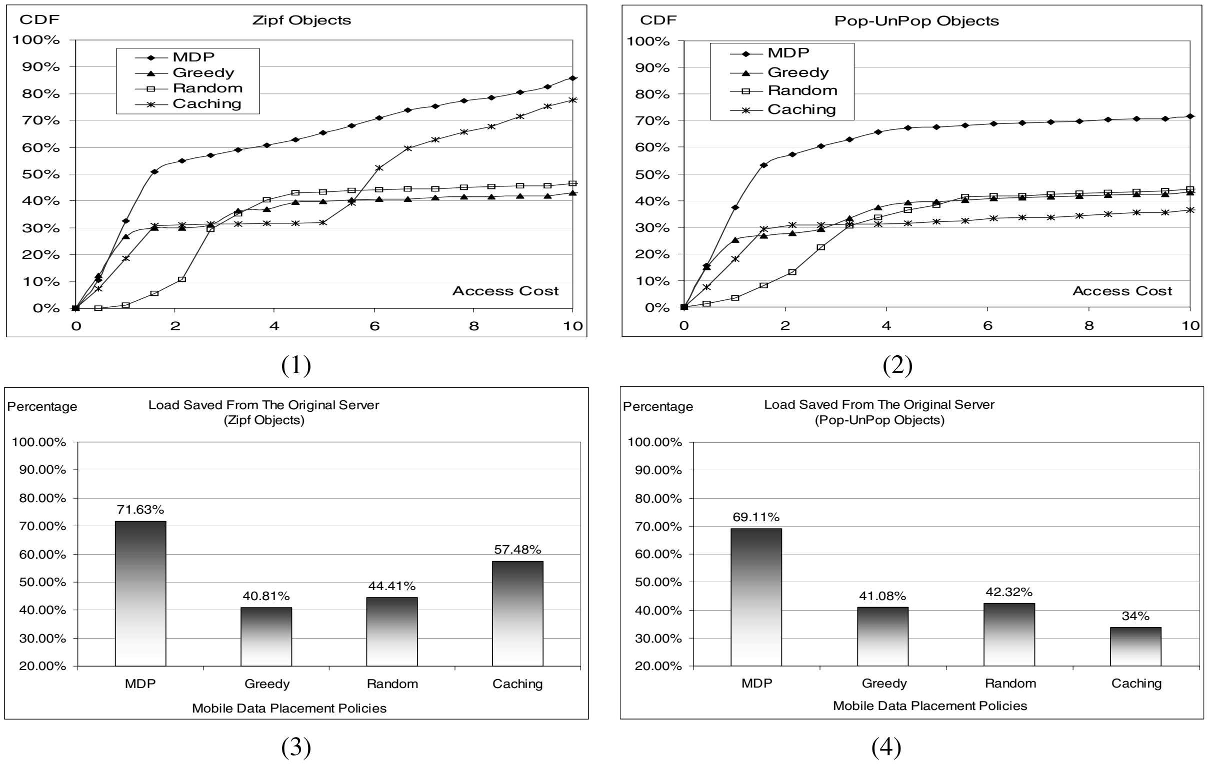

Figure 5. Basic performance of mobile data placement (MDP) on grid proxies

[6] A. Dan and D. Sitaram. An online video placement policy based on bandwidth to space ratio (bsr). In *SIGMOD '95*, 1995.

[7] F. C. A. A. C. Derrick Kondo, Gilles Fedak and H. Casanova. Resource availability in enterprise desktop grids. In *2nd CoreGRID WP4 Workshop*, 2006.

[8] P. Desnoyers, D. Ganesan, and P. Shenoy. Tsar: A two tier sensor storage architecture using interval skip graphs. In *Proceedings of ACM SenSys*, San Diego, CA, 2005.

[9] P. Felber and E. W. Biersack. Cooperative content distribution: Scalability through self-organization. In *Self-star Properties in Complex Information Systems*, 2005.

[10] A. Feldmann, R. Caceres, F. Douglis, G. Glass, and M. Rabinovich. Performance of web proxy caching in heterogeneous bandwidth environments. In *INFOCOM (1)*, pages 107–116, 1999.

[11] Y. Feng, H.-G. Li, D. Agrawal, and A. E. Abbadi. Exploiting temporal correlation in temporal data warehouses. In *DASFAA*, 2005.

[12] Z. Haas. A new routing protocol for the reconfigurable wireless networks, 1997.

[13] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *INFOCOM*, pages 1568–1576, 2001.

[14] Y. Huang, S. Mohapatra, and N. Venkatasubramanian. An energy-efficient middleware for supporting multimedia services in mobile grid environments. In *IEEE ITCC*, 2005.

[15] Y. Huang and N. Venkatasubramanian. Qos-based resource discovery in intermittently available environments. In *HPDC*, 2002.

[16] Y. Huang and N. Venkatasubramanian. Supporting mobile multimedia services with intermittently available grid resources. In *HiPC*, 2003.

[17] M. Karlsson and C. Karamanolis. Choosing replica placement heuristics for wide-area systems. In *ICDCS*, 2004.

[18] Y. Lin, B. Kemme, M. Patiño-Martínez, and R. Jiménez-Peris. Consistent data replication: Is it feasible in wans?. In *Euro-Par*, 2005.

[19] L. W. McKnight, J. Howison, and S. Bradner. Guest editors' introduction: Wireless grids–distributed resource sharing by mobile, nomadic, and fixed devices. *IEEE Internet Computing*, 8(4):24–31, 2004.

[20] S. Mohapatra and N. Venkatasubramanian. PARM: Power-Aware Reconfigurable Middleware". In *ICDCS-23*, 2003.

[21] W.-C. Peng and M.-S. Chen. Mining user moving patterns for personal data allocation in a mobile computing system. In *ICPP '00: Proceedings of the Proceedings of the 2000 International Conference on Parallel Processing*, 2000.

[22] T. Phan, L. Huang, and C. Dulan. Challenge: Integrating mobile wireless devices into the computational grid, 2002.

[23] M. Tu, P. Li, L. Xiao, I.-L. Yen, and F. B. Bastani. Replica placement algorithms for mobile transaction systems. In *IEEE TKDE*, 2006.

[24] S. Wee, J. Apostolopoulos, W. Tan, and S. Roy. Research and design of a mobile streaming media content delivery network. In *ICME*, 2003.