

A Multi-Agent Based Data Replication Mechanism for Mobile Grid

¹Senthilnathan, T. and ²T. Purusothaman

¹Department of IT,

Sri Krishna College of Engineering and Technology, Coimbatore, India

²Department of CSE, Government College of Technology, Coimbatore, India

Abstract: Problem statement: In Mobile grid, the mobile devices can be effectively incorporated in to the Grid. It enables both the mobility of the users requesting access to a fixed grid and the resources that are themselves as part of the grid. There are number challenging issues for mobile grid; such as connectivity, availability, maintainability, consistency and fault tolerance. This study addresses the issue of data replication solution that maintains consistency, which improves the availability of replicated data in a small scale mobile grid environment. **Approach:** In this study we propose a structural design that improves the availability in a mobile grid environment by placing the replica in an optimized manner so that the performance of the mobile grid environment can be improved and a multi agent based approach to maintain the consistency between the replicated data effectively by propagating the latest updates intelligently. **Results:** The study presents the results depicting the advantageous of using agents in data replication, which includes reduction in data communication cost under different circumstances like change in mobility of nodes, read write ratio of nodes and replication schema. **Conclusion:** The proposed method will select the optimum number of locations to place the replica such that the maintenance overhead such as update propagation, consistency maintenance, storage cost and communication cost will be reduced.

Key words: Mobile grid, multi agent, data replication, SNR

INTRODUCTION

Mobile, nomadic and fixed wireless devices form new type of resource sharing networks called wireless grids. Grid computing (Foster *et al.*, 2001) lets devices connected to the Internet, provides an overlay for peer-to-peer networks and dynamically shares network connected resources. The wireless grid extends this sharing potential to mobile, nomadic, or fixed-location devices temporarily connected via ad hoc wireless networks. Wireless mobile devices have become an indispensable tool for large and small scale businesses, especially for those where employees must perform their duties away from the office such as field workers or sales representatives (Ahuja and Myers, 2006). The increasing reliance on these devices has motivated the pace at which applications for these devices are developed, as well as expanding the scope and functionality of these applications.

wireless technology and grid computing technology (Birje *et al.*, 2006). Rapid advances in miniaturization, increasing processing power and feature-rich operating systems and applications, along with the proliferation of wireless access points have quickly expanded the usefulness of these devices and made them increasingly capable of taking part in grid networks.

The use of mobile devices in grid environments may have two interaction aspects: Devices are considered as the users of grid resources or as grid resource providers (Elizabeth and Sivagami, 2010). Due to the constraints on energy and processing capacity of mobile devices, their integration into the grid as resource providers and not just consumers is a challenging issue. Because of the limited bandwidth and frequent disconnections in mobile grid environment significant work has already been carried out to improve the performance and reliability of mobile grid systems.

Wireless grid computing: Wireless grid computing is evolving because of the fast developments in

Data replication issues and challenges: Replication is the process of sharing information so

Corresponding Author: Senthilnathan, T., Department of IT, Sri Krishna College of Engineering and Technology, Coimbatore, India

as to ensure consistency between redundant resources, such as software or hardware components, to improve reliability, fault-tolerance, or accessibility. It could be data replication if the same data is stored on multiple storage devices, or computation replication if the same computing task is executed many times. Replication management provides local replicas for remote applications in order to quickly access and process remote data, avoiding a great deal of data transferring, improving the efficiency of data access and the capability of fault tolerance.

In a mobile grid environment, data replication technique is used to improve availability and access cost. In recent years, more and more works focused on the replica management in parallel and distributed systems. But most of them concerned on replica location, replica replacement and consistency strategies of replica, or building infrastructures for replica management. In fact, replica placement is one of important challenges to improve performance and good placement strategies can result in significant performance gains.

Agents: In computer science, a software agent is a piece of software that acts for a user or other program in a relationship of agency. Such "action on behalf of" implies the authority to decide which (and if) action is appropriate. The idea is that agents are not strictly invoked for a task, but activate themselves. Related and derived concepts include intelligent agents (in particular exhibiting some aspect of Artificial Intelligence, such as learning and reasoning), autonomous agents (capable of modifying the way in which they achieve their objectives), distributed agents (being executed on physically distinct computers), multi-agent systems (distributed agents that do not have the capabilities to achieve an objective alone and thus must communicate) and mobile agents (agents that can relocate their execution onto different processors).

We consider the following advantages to incorporate agents in the proposed data replication system.

Reducing network load: Agents on the mobile node and base station reduce number of message exchanges between mobile node and the base station thus helps in reducing the communication cost for data access.

Executing asynchronously and autonomously: The agent can operate autonomously even if the node from where it was launched is no longer available.

Overcome network latency: Mobile agents dispatched from the base station act locally and directly execute the instructions thus helps in overcoming the network latency in achieving replica consistency across the sites.

Adapting dynamically: The agents can be retracted, dispatched, cloned, or put to sleep as network and host conditions change. For example, as better agents are developed they can be sent out on the network to replace the older version.

Operating in heterogeneous environments: The agents are dependent only on their execution environment, they facilitate heterogeneous system integration. This advantage is vital in mobile environment due to device heterogeneity (i.e., smart phones, PDA's, laptop's are involved in data grid formation).

Robust and fault tolerant: Mobile agents can react dynamically to unfavorable changes in the environment which helps to create a robust and fault-tolerant grid system.

Proposed scheme: This study addresses the problem of maintaining consistency and improving availability of replicated data for small scale distributed systems that operate in mobile environments, called as Mobile Grid. Distributed dynamic replication based on multi agent system with fault tolerance is designed. The proposed Replication system is divided into two levels base station level and mobile node level. Maintaining consistency and improving availability in replication architecture components is achieved with the help of three types of agents identified in this study, base station agent runs the expansion, contraction, switch test and fault tolerance algorithm to keep the system stable, node agent monitors the mobile node and keeps updated information about mobile node, it also acts as tokenizer and is responsible for passing back the list of changes that took place during replica updation to base station, update agent is responsible for updating the mobile replicas and maintaining consistency.

MATERIALS AND METHODS

Wolfson and Milo (1991), the authors have developed a new update propagation method, the minimum spanning tree write, based on efficient multicast algorithms. According to the minimum

spanning tree write, a given processor in the network should multicast an update of a logical data-item to all the processors that store replicas of the items, along the paths which form a minimum spanning tree. Limitations: read and write requests alone are not sufficient to decide the replication sites, signal strength and performance should be considered in deciding the replication site, also this system lacks the fault tolerance support.

Wolfson *et al.* (1997), a distributed dynamic replication algorithm for data object placement in a network with tree topology is proposed. Each replica site decides whether to duplicate its replica to its neighbors or whether to de-allocate its own replica based on the read and update access pattern. Limitations: this system cannot handle the burst which degrades the response time; also the system is not fault tolerant.

Matiasko and Zabovsky (2008), it has been proposed that rather than throughput and latency a Signal to Noise Ratio (SNR) is involved in designing replication model. SNR affects both characteristics throughput and latency in significant way. SNR directly impacts the performance of a wireless connection. Limitation: higher SNR only ensures the connection strength but other parameters like battery, performance and load on the node should be considered while deciding the replication node which requires an intelligent entity.

Elizabeth and Sivagami (2010), The Replica Supporting Fault-Tolerance (RSFT) algorithm is formulated to support QoS-aware replica placement, balancing the load of replicas and to reduce the communication cost in mobile grid environment using bottom-up dynamic programming approach. Then, a resource selection algorithm is integrated with bottom-up dynamic programming approach to support fault tolerance in mobile grid environment by considering the dynamic characteristics of mobile devices. Limitations: although this system considers all the parameters like battery, performance and load into account Connection strength is not taken into account which may result in frequent failures, also there is need for intelligent entity to decide the threshold value to optimize the replication.

A multi-master scheme is used in (Monteiro *et al.*, 2007), that is, read-any/write-any. The servers allow access (read and write) to the replicated data even when they are disconnected. To reach an eventual consistency in which the servers converge to an identical copy, an adaptation in the primary commit scheme is used. Limitations: read and write

requests alone are not sufficient to decide the replication sites, signal strength and performance should be considered in deciding the replication site, also this system lacks the fault tolerance support.

A hybrid replication strategy is presented in (Abawajy *et al.*, 2006) that have different ways of replicating and managing data on fixed and mobile networks. In the fixed network, the data object is replicated to all sites, while in the mobile network, the data object is replicated asynchronously at only one site based on the most frequently visited site. Limitations: this system cannot handle the burst which degrades the response time; also the system is not fault tolerant.

Cedar (Tolia *et al.*, 2007) uses a simple client-server design in which a central server holds the master copy of the database. At infrequent intervals when a client has excellent connectivity to the server (which may occur hours or days apart), its replica is refreshed from the master copy. Limitation: although this system improves the availability it lacks consistency of the replica, since replicas are refreshed hours or days apart, also this system requires an intelligent entity to resolve the update conflicts at the central server.

A price based distributed energy constrained resources allocation optimization algorithm (Li and Li, 2010) is proposed which aims to reduce energy consumption and also to improve the application utility in a mobile grid environment with a limited energy change, ensuring battery lifetime and also the deadlines of the grid application. Limitations: this system aims only in improving the battery life time of node and the response time of the grid application, incorporating parameters like signal strength and performance, intelligent entity in handling requests can yield better results.

IEEE 802.11 based mobile grid architecture (Ghosh *et al.*, 2007) is proposed which discusses a generic node mobility prediction framework for mobile grid environment. This framework can be used to formulate cost effective job allocation schemes based on a predetermined pricing strategy at the Wireless Access Point to be distributed to the mobile nodes under it. Limitations: this system aims only in predicting the mobility of the node to formulate cost effective job allocation, incorporating intelligent entity in handling request and in deciding the job allocation site can yield better results also parameters like battery, performance and load should be considered.

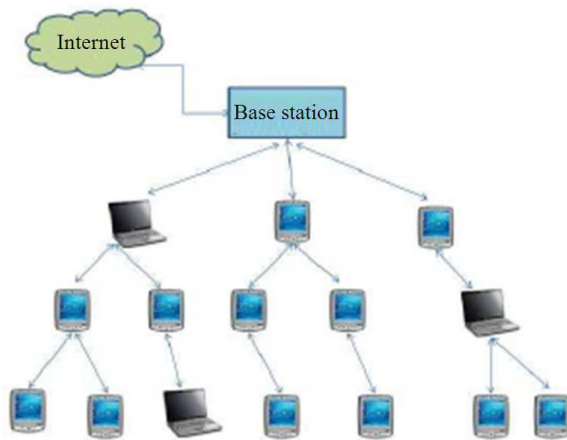


Fig. 1: Mobile grid replication architecture

Proposed method: The proposed replication strategy composed of two components: 1. Replication architecture and 2 agent based replication method. Replication architecture serves the purpose of providing a comprehensive infrastructure for improving data availability and supporting small number of replicas in mobile grid environment by determining the required components which are involved in replication process. Agent based replication method serves the purpose of transferring data updates between the components of the replication architecture, so has to achieve consistency of data and improve availability of recent updates to interested host. The strategy is hybrid in nature consists of both pessimistic and optimistic replication approaches (Fadellemloula *et al.*, 2009). The pessimistic approach is used for restricting updates of infrequently changed data to a single replica. An optimistic replication, in contrast, allows multiple replicas to be concurrently updatable based on the optimistic presumption that update conflicts are rare. Conflicting updates are detected and resolved once they have occurred.

Replication architecture: The study considers an environment which consists of fixed host (Base station), mobile hosts, a replication manager on base station and a replicated database on each host. A replicated database is called as mobile database when it is stored on a mobile host. Fixed host represent server with more storage and processing capabilities than the rest. The replicated database contains a set of objects stored on the set of hosts.

The proposed replication architecture considers a total geographic area (Fig. 1) divided into two levels: 1. Base station level and 2. Mobile Host level.

We model the grid environment as a connected graph. $G(V, E)$, where V is the set of nodes and E is the set of links connecting the nodes. We use 'n' to denote the total number of nodes in the given network, i.e., $n = |V|$. Each edge has a non-negative weight associated with it. A single data item is considered in the network, which is to be replicated at selected grid nodes. For each node $i \in V$, the frequency of reading the data item is R_i , the frequency of writing the data item is W_i and the cost of replication of the data item is $cost_{q,i}$. The replica management problem in the above modeled grid environment can be defined as follows. Given a grid $G(V, E)$ and number M ($1 \leq M \leq n$), select at most M replication nodes such that total (read and write) cost minimized, meeting QoS requirements of the node and also the selected node should have a SNR value greater or equal to the critical SNR (SNR_c) value and mobility, battery, performance (load) values should be in the acceptable range. The linear mathematical model for throughput T , prediction based on previous observations looks as follow (Na *et al.*, 2006): SNR is a measure of signal strength relative to background noise. The ratio is usually measured in decibels (dB). SNR, in decibels is given by the formula Eq. 1:

$$SNR = 20 \log_{10} (V_s/V_n) \quad (1)$$

Where:

V_s = The incoming signal strength in micro volts
 V_n = The noise level, also in micro volts Eq. 2 and 3:

$$T = T_{max}, SNR_0 > SNR_c \quad (2)$$

$$T = A \times (SNR_0 - T_0), SNR_0 \leq SNR_c \quad (3)$$

where, T_{max} is saturation throughput, A defines the slope, T_0 is a breaking point where T_{max} is changing to a curve described by A , SNR_0 is a cutoff SNR specified by the hardware vendor. Remaining battery information B_i is calculated as shown below Eq. 4 (Elizabeth and Sivagami, 2010):

$$B_i = B_{ic} - B_{ip} \quad (4)$$

Where:

B_{ic} = The current amount of battery before a job is assigned

B_{ip} = The amount of battery to process a job

Mobility information: Job should not be assigned to a replica node if it has a frequent movement, which is the probability of a replica mobile device v that keeps moving out of a mobile grid network, which is less than 40% by predicted path.

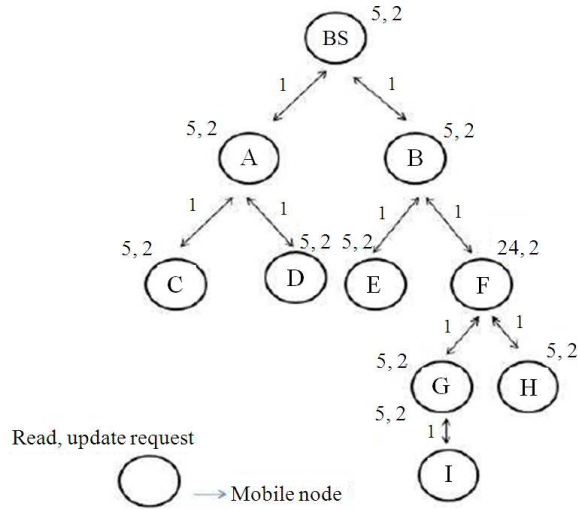


Fig. 2: Tree topology of Mobile grid

Performance information: Performance information is those of CPU, memory and storage. Performance information reflects dynamic status of a mobile device. If the load on the replica mobile node is too low, that is less than 70% of total capacity and then job is assigned.

Consider Tree $T = (V, E)$ rooted at the base station (Fig. 2), where node set V is the set of base station and other mobile devices and $E = V \times V$ is the set of wireless communication links. For each edge $E(u, v)$, a communication cost $d(u, v)$ is associated with it. Each node $v \in V$ in tree has two weights read (v) and write (v) of that node respectively. And the request is served by the node named $p(v)$, which is the first node in the path from node v to root and there is a replica stored on it. BS calculates the total number of request $t(v)$ for each node so as to find suitable replica nodes to balance the load, satisfy the performance requirements of user and to reduce communication cost. If the mobile node is a leaf node of the tree, its read (v) + write (v) will be its $t(v)$. Else the sum of read (v) + write (v) of that node and its children will be its $t(v)$. Therefore, the dynamic programming equation of workload is given as Eq. 5 and 6:

$$T(v) = \text{read}(v) + \text{write}(v), \text{ if } v \text{ is leaf} \quad (5)$$

$$t(v) = \sum_{v_j \in \text{ch}(v)} t(v_j) + (\text{read}(v) + \text{write}(v)), \text{ if } v \text{ is not leaf } v_j \in \text{ch}(v), v_j \in S \quad (6)$$

Each node q in the overall system and a data object i are associated with a non-negative read rate

$\text{Read}_{q,i}$ and a non-negative write rate $\text{Write}_{q,i}$. If Y is the write cost for a given object i and X is the read cost for the same object, then $Y/X = \alpha_i$ is the ratio of the write cost for each node. If there are no replicas for object i in the system then the total data transfer cost for this object at node q is Eq. 7 (Lamehamed *et al.*, 2002):

$$\text{cost}_{q,i} = (\text{Read}_{q,i} + \text{Write}_{q,i}) \times \text{size}(i) \times d(q, d) \quad (7)$$

where d is the node containing the object i , $d(q, d)$ is the cost of sending a unit of data along the path from d to q .

Essentially the proposed method is to select a set of optimum number of replication nodes M , such that the total cost: $\text{cost}_{q,i}$ is minimized. However the given grid environment is dynamic in nature, the resources, user requests and the network statistics changes in a timely fashion. Therefore after finding the optimum number M nodes, the assignment of the requests to replica should consider the number of user requests and network characteristics (i.e., SNR value, SNR_c value, mobility, battery, performance (or load), QoS) for the current period. However the candidate site that holds the replicas currently may not be the best site to fetch nor cannot handle the access request, if the user request and network latency changed. Therefore relocation to be considered to maintain good performance.

Agent based replication method: The proposed replication method based on a multi-agent system is a 5-Tuple $\langle T, S, D, I, U \rangle$ entity, where.

$T = \{t_1, t_2, t_3, t_4\}$ is a finite set of replication agent types. A type t_i maps each replica system agent to a certain level (i.e., the type determines the location or functionality of agent).

$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ is a finite set of replica agent states. Each state represents the current activity that is carried out by the model.

$D = \{d_1, d_2, \dots, d_n\}$ is a finite set of data items that are required to store recent updates that are performed on the similar data items, which are stored in the database.

$I = \{i_1, i_2, \dots, i_n\}$ is a finite set of primitives/instructions that are required to perform the agent activities and the transitions.

$U: T \rightarrow \{1, 2, 3, \dots, k\}$ is a function for assigning a unique identifier for the agent based replication in the system. According to the above mentioned formal definition, the proposed scheme consists of code and database, which has type, state and unique identifier.

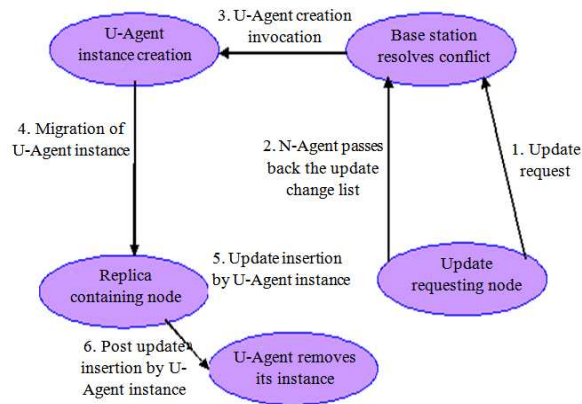


Fig. 3: State transition diagram for update propagation

Agent types: Agents used in this replication scheme are classified into three types according to the level in which the agents carries out its activities or based on the functionality. The agent types are.

Base Station Agent (BS-Agent): BS-Agent is responsible for analyzing the context of the environment, i.e., it analyses system requirements and detects any changes in the grid environment and reports it to replication manager (Fig. 3). The only possible state of the BS-Agent is:

Monitoring: In this state, the BS-Agent monitors the connection with the other devices through interaction with its environment (i.e., hosted device) via message passing and detects any changes in the environment.

Node Agent (N-Agent): N-Agent is responsible for monitoring the mobile node and collecting mobility, battery, performance (load), SNR and SNRc information and maintains the read and write update counter for the replica. The possible states of the N-Agent are:

Monitoring: In this state, the N-Agent monitors the connection with the other devices through interaction with its environment (i.e., hosted device) via message passing. N-Agent also monitors the mobility, battery, performance (or load) information, signal-to-noise ratio, read and write requests.

Tokenizer: Write Once Read All (WORA) method is used in this scheme, N-Agent of mobile node which wants to update the replica acts as a tokenizer

and updates the mobile replica, the update/change list is passed to base station via message passing, in case of any update conflicts base station is responsible for resolving the update conflicts.

Update Agent (U-Agent): U-Agent is responsible for updating the mobile node replica and maintaining consistency across the mobile grid. U-Agent follows the Spanning tree path maintained by the localization manager, whenever the path splits U-Agent splits itself and takes different path to update the mobile copy of replica, once the update task is completed U-Agent removes itself. The possible states of the U-Agent are:

Creating instance: The U-Agent creates an instance of it and stores the set of recent updates on this instance.

Migration: The U-Agent instance migrates from the base station to other mobile device which contain replica that is inconsistent.

Insertion: In this state, the U-Agent instance inserts it's stored recent updates in the database of the mobile host.

Removing: The migrated instance of U-Agent removes itself after completion of the insertion process (Fig. 3).

Base station level: This level contains the master replica, which must be synchronized with the replicas from the nodes at the mobile host level. The server in this level is responsible for synchronizing all changes that have been performed on infrequently changed data with the lower level. This level contains the replication manager which acts as the core of the proposed Replication method; two types of agents are associated with this level. Following are the components of replication manager (Fig. 4).

Replica manager: The replica manager selects the optimum number of locations to place the replica such that the maintenance overhead such as update propagation, access cost and storage cost can be minimized with minimum number of replications. Replication manager makes use of self stabilizing distributed dynamic replication algorithm for data object placement in a network with tree topology. The distributed dynamic replication algorithm works by three tests, namely, expansion test, contraction test and switch test.

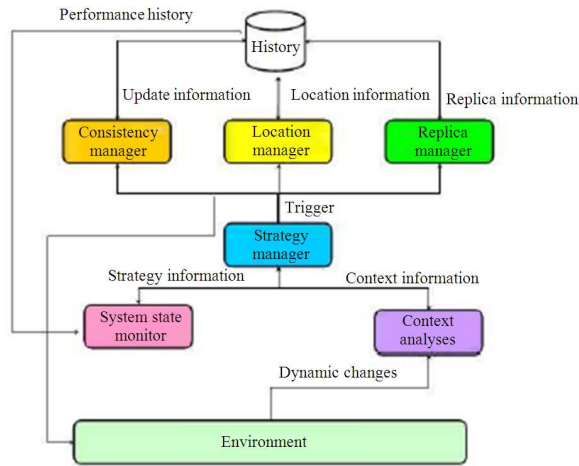


Fig. 4: Replication manager architecture

The expansion test is executed at each neighbor of i , if the changes like one or more mobile node getting added to the wireless grid, detection of drastic changes in read and update requests of mobile nodes in a particular instance triggers the execution of expansion test by the BS-Agent and is responsible for replica schema expansion, aging factor can be used to modify the third step to improve the efficiency of the expansion test. What motivates is, it is easier to deal with prevention of node failures due to communication network than to solve the node failure, the proposed method prevent data replication at nodes which has the tendency towards failure.

Localization manager: The localization manager keeps track of the nodes containing the replica and is responsible for creation and allocation of T-Agent upon the update request.

Consistency manager: The consistency manager ensures replica consistency after each write operation and resolving update conflicts and creating the instance of U-Agent which is responsible for inserting update to each replica site and helps maintaining consistency.

Context analyzer: The context analyzer detects the pertinent change in required context information and provides context information, runs the fault tolerance algorithm to maintain stability in the system and also contains an instance of BS-Agent which helps in detecting the changes in the system environment. In addition to expansion, contraction and switch tests, fault tolerance algorithm (Algorithm4) is used to make the system stable in

case of node failures; fault tolerance algorithm is triggered by the node failure.

Strategy manager: The strategy manager adapts the replication strategy to context or system state variations. For example, the system changes its strategy from pessimistic to an optimistic one in order to improve the data availability.

System state monitor: Replication system monitors its own state. This state is represented by system performance (load) parameters, data availability parameters and data consistency parameters. Replication scheme that is most adapted to the change in context by modifying the plans are stored in history.

Algorithm 1 Expansion test:

- 1: Begin
- 2: for Each neighbor $j \in M$ do
- 3: Examine R_i and W_i .
 R_i is the read request from j to i during last time period, W_i is the total number of update request from i and neighbor $6 = j$ to i .
- 4: if $R_i > W_i$ then
- 5: Mark those nodes as candidate nodes.
- 6: end if
- 7: end for
- 8: for Each candidate do
- 9: if Candidate is the site on which replica is currently located then
- 10: Goto step 21
- 11: else
- 12: For the candidate sites get SNR and SNR_c, battery, mobility, load values.
- 13: if SNR > SNR_c, battery > 10 %, mobility < 40 % and load < 70 % then
- 14: Replicate the data on the candidate node (thus, j joins M).
- 15: else
- 16: Choose site with the read requests > update requests, from the non candidate set and mark it as candidate for data replication.
- 17: Go to step 8
- 18: end if
- 19: end if
- 20: end for
- 21: End

The contraction test (Algorithm 2) is only done at the replica in M , contraction test is triggered by the detection mobile node failure by the BS-Agent and is responsible for replica schema

contraction, if any such node failure is noticed, then contraction test allows us to release the site from the replication schema.

Algorithm 2 contraction test:

```

1: Begin
2: for Each node  $i \in M$  do
3:   Examine  $W_j$  and  $R_i$ .
       $W_j$  is the number of update requests that  $i$ 
      received from  $j$  during the last time period,  $R_i$  is
      the number of read requests that  $i$  received in
      the last time period.
4:   Examine SNR, SNRc, battery, mobility, load
      values.
5:   if  $W_j > R_i$  or  $SNR \leq SNR_c$  or battery  $< 10\%$  or
      mobility  $> 40\%$  or load  $> 70\%$  then
6:     Release site from replication schema  $M$  (thus,  $i$ 
      exits  $M$ )
7:   else if The replica node  $i$  doesn't respond then
8:     Release site from replication schema  $M$  (thus,  $i$ 
      exits  $M$ )
9:   end if
10: end for
11: End

```

The switch test (Algorithm 3) is executed at the replica in M and is responsible for moving replica to the neighboring node by discarding its own copy. If the candidate node that is holding the replica currently is not the best site to fetch nor can handle the access request due to the change in user request and network latency. In such situation reallocation of replica is done by the switch test execution. The algorithm will converge to the optimal placement scheme if user access pattern remains stable.

Algorithm 3 Switch test:

```

1: Begin
2: if If there is only  $i$  in  $M$  then
3:   for Each neighbor  $k$  do
4:     Examine two integers denoted by  $R_k$  and  $R_k^*$ .
      Integer  $R_k$  is the number of read requests
      received by  $i$  from  $k$  during the last time period
      and  $R_k^*$  is the number of all other read requests
      received by  $i$  during the last time period.
5:     Examine SNR, SNRc, battery, mobility, load
      values.
6:     if  $R_k > R_k^*$  and if  $SNR > SNR_c$ , battery  $> 10\%$ ,
      mobility  $< 40\%$ , load  $< 70\%$  then
7:        $i$  sends the replica to  $k$  with an indication that  $k$ 
      becomes the new singleton processor in the
      replication schema (thus,  $i$  exits  $M$  &  $k$  joins  $M$ )

```

```

8:   end if
9:   end for
10: end if
11: End

```

Mobile host level: This level contains the mobile replica; each replica in this level is updated frequently and then synchronized with the master replica. Mobile host level uses the N-agent to monitor the connection with the other mobile devices through interaction with its environment via message passing. N Agent is also used to monitor the mobility, battery; performance (or load) information, signal-to-noise ratio, read and write counter values. Mobile host passes information to the replication manager, Context analyser with the help of BS-Agent which detects the changes and reports it to strategy manager, the replication manager takes the actions according to the changes and brings back the stability of the system by executing the fault tolerance algorithm. N-Agent also acts as tokenizer during update and passes the update/change list that took place during update to base station via message passing. N-agent and U-Agent works at the mobile host level and they are responsible for carrying the list of changes during update and update insertion after conflict resolution respectively.

Algorithm 4 Fault tolerance algorithm:

```

1: Begin
2: Scan the grid to find out failed node.
3: if Failed node has children and failed nodes
   previous level is base station then
4:   Base station becomes the parent node of failed
   node children. Find the communication cost to
   base station.
5:   Go to step 13.
6:   else
7:   Scan mobility, battery information and load
   information for other replica node which is at
   the higher level on the path to base station from
   failed node.
8:   if The replica node which is on the path to base
   station has  $SNR > SNR_c$ , battery  $> 10\%$ ,
   Mobility  $< 40\%$ , load  $< 70\%$  then
9:     That particular node becomes the parent node to
     the failed nodes children.
10:   end if
11:   end if
12: End

```


Update propagation using agents: The proposed agent based replication system uses top down propagation approach, the typical operations which are involved in top down propagation are:

- Consistency manager resolves the update conflict and creates an instance of U-Agent which contain updates list to be inserted in all replicas to maintain consistency.
- U-Agent takes the path given by localization manager to reach the mobile nodes which contains the replica to be updated.
- U-Agent replicates itself when it needs to take diverge path, once the update insertion is complete U-Agent removes its instance.

RESULTS AND DISCUSSION

Consider the simulation setup with one base station (which holds the original data item) and nine mobile hosts labeled from A to I (Fig. 2), mobile host A and B are directly connected to base station at an equal distance of 1m, node C,D and E,F are connected to node A and B respectively at an equal distance of 1m , node G and H are connected to node F at equal distance of 1m and finally node I is connected to node F at a distance of 1m. All nodes have read request, write request of 5,2 respectively except the node F which has 24 read request and 2 update request. Results are plotted comparing communication cost by varying different parameters, such as read, write ratio, mobility of a node, node failure.

The first set of results (Fig. 5) are obtained by comparing communication cost with and without data replication, clearly agent based data replication has reduced the communication cost by considerable amount with the help of optimized replication achieved through the expansion, contraction and switch test.

The second set of results (Fig. 6) are obtained by comparing communication cost with and without fault tolerance support, let suppose node F has failed and the request from the child node should be served by node B, but if node B fails to serve then the request has to be forwarded to BS, so if we know in prior with the help of fault tolerance algorithm that node B cannot handle the request then the requests are directly sent to BS thus reducing communication cost.

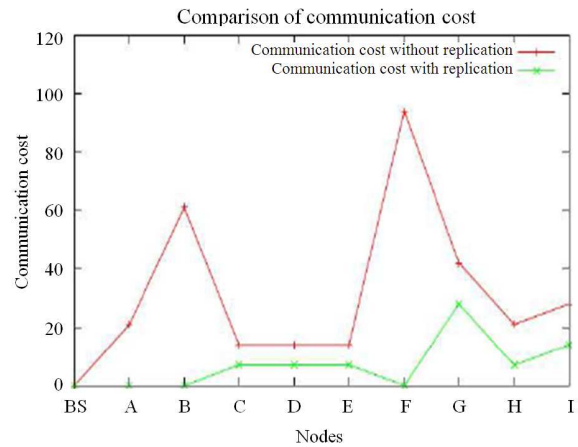


Fig. 5: Communication cost comparison with and without replication

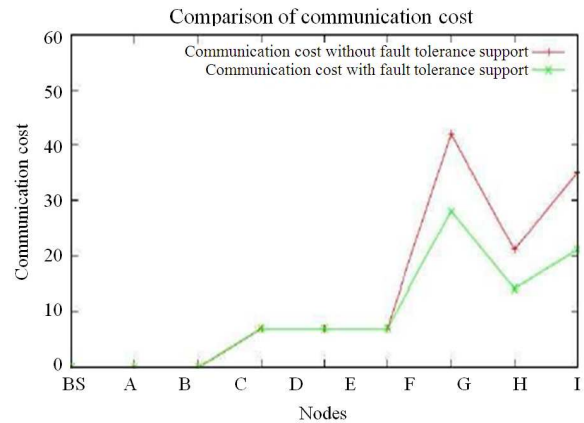


Fig. 6: Communication cost comparison with and without fault tolerance support

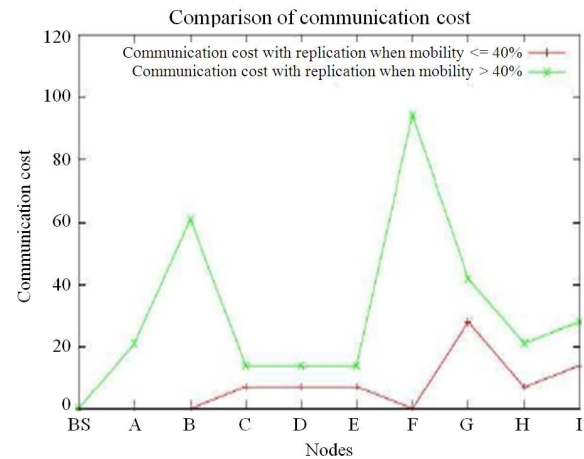


Fig. 7: Communication cost comparison by varying mobility of nodes

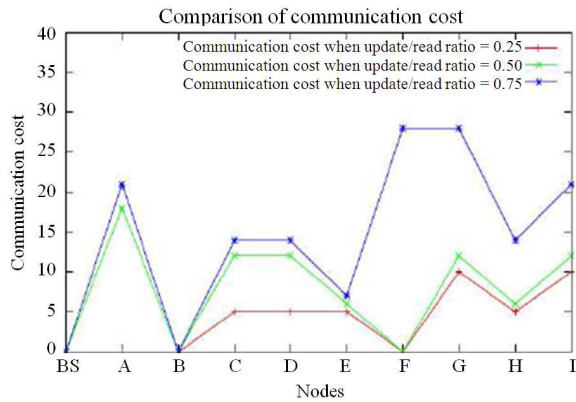


Fig. 8: Communication cost comparison by varying update/read ratio

The third set of results (Fig. 7) are obtained by varying the mobility of the node, if the mobility of the node is less than 40% then the communication cost gets reduced considerably. The fourth set of results (Fig. 8) is obtained by comparing communication cost by varying update(write)/read ratio, initially update/read ratio is equal to 0.25, i.e., all nodes have 1 update and 4 read requests, in the next step update/read ratio is increased to 0.5, i.e., all nodes have 2 update and 4 read requests, in the final step update/read ratio is increased to 0.75, i.e., all nodes have 3 update and 4 read requests, this clearly shows the impact of update/read ratio on data replication and communication cost.

CONCLUSION

In this study Distributed dynamic replication based on multi agent system with fault tolerance is proposed. A self stabilizing distributed dynamic replication algorithm for data object placement in a network with tree topology is discussed in detail. The proposed method will select the optimum number of location to place the replica such that the maintenance overhead such as update propagation, access cost and storage cost can be minimized with minimum number of replications. In future, assigning replica to a mobile node even if it has more frequent move from one mobile grid network to other will be considered. Also as a part of our future research, we plan to develop the required interfaces to implement the proposed strategy in different mobile grid application environments such as M-Learning, Mobile Healthcare.

REFERENCES

- Abawajy, J.H., M.M. Deris and M. Omer, 2006. A novel data replication and management protocol for mobile computing systems. *Mobile Inform. Syst.*, 2: 3-19.
- Ahuja, S.P. and J.R. Myers, 2006. A Survey on wireless grid computing. *J. Supercomput.*, 37: 3-21. DOI: 10.1007/s11227-006-3845-z
- Birje, M.N., S.S. Manvi and B. Prasad, 2006. Agent based resource brokering and allocation in wireless grids. *Proceedings of the IEEE International Conference on Services Computing*, Sept. 18-22, IEEE Xplore Press, Chicago, IL., pp: 331-334. DOI: 10.1109/SCC.2006.17
- Elizabeth, L.A. and V.M. Sivagami, 2010. Yet another improvement in RSFT algorithm in mobile grid. *Int. J. Comput. Appl.*, 5: 0975-8887.
- Fadellmoula, A.A., P.D.D. Dominic, A. Abdullah and H. Ibrahim, 2009. A binary replication strategy for large-scale mobile environments. *Int. J. Comp. Sci. Secu.*, 3: 83-104.
- Foster, I., C. Kesselman and S. Tuecke, 2001. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15: 200-222. DOI: 10.1177/109434200101500302
- Ghosh, P., N. Roy and S.K Das, 2007. Mobility-based cost-effective job scheduling in an IEEE 802.11 mobile grid architecture. The University of Texas at Arlington.
- Lamehamedi, H., B. Szymanski, Z. Shentu and E. Deelman, 2002. Data replication strategies in grid environments. *Proceedings of the 5th International Conference on Algorithms and Architectures for Parallel Processing*, Oct. 23-25, IEEE Xplore Press, pp: 378-383. DOI: 10.1109/ICAPP.2002.1173605
- Li, C. and L. Li, 2010. Energy constrained resource allocation optimization for mobile grids. *J. Parallel Distributed Comput.*, 70: 245-258. DOI: 10.1016/j.jpdc.2009.06.003
- Matiasko, K. and M. Zabovsky, 2008. A data replication for mobile environment. *J. Elect. Eng.*, 59: 277-280.
- Monteiro, J.M., A. Brayner and S. Lifschitz, 2007. A mechanism for replicated data consistency in mobile computing environments. *Proceedings of the 2007 ACM Symposium on Applied Computing*, Mar. 11-15, ACM, Seoul, Republic of Korea, pp: 914-919. DOI: 10.1145/1244002.1244203

- Na, C., J.K. Chen and T.S. Rappaport, 2006. Measured traffic statistics and throughput of IEEE 802.11b public WLAN hotspots with three different applications. *IEEE Trans. Wireless Commun.*, 5: 3296-3305. DOI: 10.1109/TWC.2006.05043
- Tolia, N., M. Satyanarayanan and A. Wolbach, 2007. Improving mobile database access over wide-area networks without degrading consistency. *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, Jun. 11-14, PR, USA., pp: 71-84. DOI: 10.1145/1247660.1247672
- Wolfson, O. and A. Milo, 1991. The multicast policy and its relationship to replicated data placement. *ACM Trans. Database Syst.*, 16: 181-205. DOI: 10.1145/103140.103146
- Wolfson, O., S. Jajodia and Y. Huang, 1997. An adaptive data replication algorithm. *ACM Trans. Database Syst.*, 22: 255-314. DOI: 10.1145/249978.249982