An Extended Approach based on Genetic Algorithm for Multi-Tenant Data Management

Reyhaneh Kheiri1, Alireza Taghizadeh2, Elaheh Kheiri3 and Mostafa Ghobaei Arani4

1 Department of Computer Engineering, Mahallat Branch, Islamic Azad University, Iran
2 Department of Computer Engineering, Parand Branch, Islamic Azad University, Iran
3 Department of Computer Engineering, Mahallat Branch, Islamic Azad University, Iran
4 Department of Computer Engineering, Mahallat Branch, Islamic Azad University, Iran

1 reyhaneh.kheiri@gmail.com, 2 alireza.taghizadeh@gmail.com,
3 elaheh_kheiri2002@yahoo.com 4 mostafaghabaye@yahoo.com

Abstract

In recent years, the applications of cloud computing technology has been increased significantly. Multi-tenant property of cloud computing allows data sharing between clients. The multi-tenant data management system breaks the cost of hardware, software and professional services among a large number of tenants and thus reduces per-tenant cost. A multi-tenant database system should provide high performance with small storage space and excellent scalability. One major challenge is to reach a database scheme with high-quality. In this paper, we offer a compatible ADAPTED database schema based on genetic algorithms, and compare it with similar schema. Simulation results show that the proposed approach (GA-ADAPT) besides considering criteria of quality of service, provides higher performance comparing the previous ones in terms of selection time of the important features and the average response time to the tenants.

Keywords: Cloud computing, Multi Tenancy, Data Management

1. Introduction

Today, cloud computing is one of the most discussed topics in the world of information technology. Its multi-tenant service-oriented architecture provides the possibility of providing the infrastructure, platform or software as a service to multiple customers on demand. Customers rent the services based on their need and pay cost of what they have used. This makes cloud computing an attractive choice for organizations [1-3]. Cloud computing architecture is primarily a multi-tenant service-oriented architecture. Multi-tenancy refers to the manner in architectural design of systems that offer software as a service.

One of the important factors for the success of multi-tenancy is data management [4]. The cloud service providers face with ever-increasing number of users. If they want to use the previous method, will certainly encounter many difficulties and challenges, including the complexity of managing, distrust and discontent users [5].

Three approaches have been introduced to multi-tenant data management [5-8]: Independent Databases and Independent Database Instances (IDII), Independent Tables and Shared Database Instances (ITSI), Shared Tables and Shared Database Instances
STSI). However, they suffer from some limitations. In IDII hardware is shared by different tenants. The service provider creates one specific database to serve each tenant. IDII has good data isolation and security. However, the maintenance cost of IDII is high and its scalability is very poor.

ITSI has poor scalability since it needs to maintain large numbers of tables. However, STSI achieves good scalability at the expense of poor performance and high space overhead. Thus it calls for an effective schema design method to address these problems. ADAPT is one of the proposed methods to solve this problem. It builds the physical tables by the attribute level instead of the tenant level. It identifies the important attributes and uses them to generate an appropriate number of base tables. For the remaining attributes, it constructs supplementary tables.

In result, ADAPT has much smaller number of tables than the total number of distinct attributes which do not grow in proportion to the number of tenants.

In ADAPT in order to make the base tables achieve excellent performance with low space overhead and better scalability, it is necessary to determine a practical number of base tables. The practical number means to build an appropriate number of base tables with proper table width, few NULLs in which the performance, space and scalability can all be close to perfect in the database system. If the number of common attributes is quite small, we can enumerate the number of clusters and determine an appropriate value. In this case, the enumeration cost is not very large. However, when the number of the common attributes is very large, the enumeration method is rather expensive [5].

Given the above explanation, now we have to optimize ADAPT to select additional and important features when the number of features is very high and to have suitable time to respond to the requests. In this paper, compatible ADAPT database schema is provided using genetic algorithm.

The rest of this article is organized as follows: The second section is devoted to the related works, in the third section the proposed approach of GA-ADAPT is introduced. Evaluation and simulation results of the proposed approach are placed in the fourth section which is followed by conclusion and future works in last section.

2. Related Works

Various researches have been accomplished in the field of resource auto-scaling. The aim of some of them is to offer a scalability approach for a special application like server of web and others have done for optimizing the mechanism of scalability. In this part we are going to consider some of the previous researches about auto-scaling.

Various studies have been conducted in the field of data management in multi-tenancy, each of them having their own advantages and disadvantages. In the following, we introduce some of them.

Aulbach et. al., [9] focuses on providing extensibility for multi-tenant databases. They introduce chunk tables to store data in extension attributes. However, its table is too big and plenty of costly self-join operations happen on it and is proved to be slower than the conventional tables. They also mentioned the extension table layout which is well suited to the customization of the multi-tenant application.

Jiacai et. al., [5] introduce three highly advanced approach to data management of multi-tenancy. These approaches include: independent databases and their examples, independent tables and examples of common database, common tables and examples of common database. They also presented a consistent database schema design method for multi-tenant applications in order to achieve good scalability and high performance with low space requirements called ADAPT. To this end, we identify the important attributes and use them to generate several base tables. For each of other attributes, we construct supplementary tables. We discuss how to use the kernel matrix to determine the number
of the base tables, apply the graph-partitioning algorithm to construct the base tables and evaluate the importance of attributes using the well-known PageRank algorithm. We develop a cost-based model to adaptively generate the base tables and supplementary tables.

Haitham et. al., [10] propose an architecture design to build an intermediate database layer to be used between software applications and Relational Database Management Systems (RDBMS) to store and access multiple tenants’ data in the Elastic Extension Table (EET) multi-tenant database schema. This database layer combines multi-tenant relational tables and virtual relational tables and makes them work together to act as one database for each tenant.

They also propose a novel multitenant database schema design to create and configure multitenant applications, by introducing an Elastic Extension Tables (EET), which consists of Common Tenant Tables (CTT), Extension Tables (ET), and Virtual Extension Tables (VET). Further, they proposed an access control method for a tenant’s users to grant them an access to a tenant’s data stored in EET multi-tenant database schema. This access control permits the tenant’s users to access the table columns and rows based on a number of groups or roles assigned to these users.

Madhu et. al., [11] compare the performance of accessing data from EET and UTSM. Experimental study shows a significant performance improvement result for our EET in comparison with the performance of UTSM (Universal Table Schema Mapping) and the query execution time is much faster when we access data from EET than when we access it from UTSM. However, the execution time of inserting rows in EET is slightly slower than UTSM.

Haitham et. al., [12] propose a multi-tenant data management service called Elastic Extension Tables Schema Handler Service (EETSHS), which is based on a multi-tenant database schema called Elastic Extension Tables (EET). This data management service satisfies tenants’ different business requirements, by creating, managing, organizing, and administering large volumes of structured, semi-structured, and unstructured data. Furthermore, it combines traditional relational data with virtual relational data in a single database schema and allows tenants to manage this data by calling functions from this service.

Yifeng Luo et. al., [13] propose a novel and cost-efficient mechanism to support multitenant database as a service (MTD BaaS) in cloud for small businesses. In MTD BaaS, tenants get rid of the expensive cost of owning and maintaining database systems. The overall aim of MTD BaaS techniques is to obtain good system scalability by enhancing the consolidation of tenants, so that higher resource utilization and lower operational cost can be achieved, so that high data consolidation, good system scalability and low operational cost could be achieved by service providers.

Zhi Hu Wang et. al., [8] study the technologies to build a cost-effective, secure and scalable multi-tenant infrastructure, especially in data tier and explore all kinds of potential implementation patterns of data tier multi-tenancy on aspects of isolation, security, customization & scalability etc. Furthermore, evaluate the performance of these design patterns on aspects of isolation and security via a series of experiments and simulations and the cost of these patterns study from the infrastructure, management and development aspects by using different kinds of measurement metrics. For the limit space, this paper only focuses on the performance evaluation via a set of simulations, and identifies potential performance bottlenecks, corresponding optimization approaches and best implementation practices for different multitenant business usage models. A brief comparison of the discussed related works is shown in Table 1.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Technic</th>
<th>advantages</th>
<th>disadvantages</th>
</tr>
</thead>
</table>
| Aulbach et al[9]  | Generation of extensibility for multi-tenant databases | high efficiency                | -Being greater than tables limit  
                              -Costly self-joining operations                                      |
| Jiacai et al[5]   | IDII                                           | -Information Isolation property  
                              -High security                                                             | - The number of databases per server to the number of tenants increases  
                              -high cost of preservation  
                              -poor scalability                                                        |
|                   | ITSI                                           | -lower maintenance costs compared to IDII  
                              -better scalability than IDII                                                | -private tables in a common database increases on the number of tenants  
                              -scalability is limited by the number of tables that could be supported by database system |
|                   | STSI                                           | -reduced maintenance costs      
                              -better scalability than previous approaches of IDII and ITSI               | -creating many NULL in the table  
                              -low yield  
                              -low efficiency  
                              -a lot of wasted space                                                    |
|                   | ADAPT                                          | -Does not provide many NULLs     
                              -number of tables will not increase in terms of the number of tenants       | -As the number of features is increased, a long time is spent to find the main features and remaining characteristics |
<p>|                   |                                                | -good performance               |                                                                                |
|                   |                                                | -high efficacy                  |                                                                                |
|                   |                                                | -low space overhead and required space |                                                                                |
| Haitham et al[10] | Suggestion of an interface database layer to be used between the application software and relational database management systems to store data for multiple tenants and access them using flexible extension tables | -can be used specially as a base to build software programs in general and SaaS applications -converts unstructured data of tenants into structured data -uses EET for storage of data collected via e-mail, news, and online texts -enables database service providers to create a single database application that can support several tenants based on the same hardware and software -relief from the high cost of writing SQL query codes and Backend data management code |  |
| Madhu et al[11] | Comparing performance of access to data via GET and ITSM | -Improved performance of search, retrieve, update and delete data by EET compared to UTSM -EET avoids storing rows with NULL values -EET provides possibility to create virtual relationships between shared physical tables of tenants and virtual tables -EET allows tenants to use the three database models to redirect -the time of taking place the row in ESET is slower than UTSM |  |</p>
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Description</th>
<th>Benefits/Features</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Haitham et al[12]   | proposing a multi-tenant data management services called controller service of flexible extension tables schema                              | -Different needs of tenants are met with the creation, management, organization and implementation of large-volume structured, semi-structured and unstructured data  
-This will allow tenants to manage their data without writing SQL queries  
-The time of the search, placing and update CTT rows are slightly faster than the rows of VET  
-EET scheme is a good representative to manage multi-tenant data for SaaS and Big Data applications | -Elimination of CTT rows are slower that VET                                                        |
| Luo et al[13]       | Proposing MIDBasS mechanism                                                                                                                   | -Get rid of the high costs of ownership and maintenance of database systems  
- Storing tenants data on the side server MTDBaaS and then search and update data through the browsers or server application  
-Integration of large amounts of data  
-Good scalability of the system  
-Low operating costs                                                                                      | -Consumption of more sources                                                                         |
| Wang et al[8]       | Investigation of multi-tenant executive patterns of data rows in terms of aspects of isolation, security, upgrades and improvements and scalability | -Isolation pattern of specific database for larger tenants who need relatively higher TPS is more appropriate and has better separation capacity compared to other separation pattern  
- separation model schema / specific table achieves the highest performance  
- yield loss in isolation model and model-specific database schema/ shared table by an increasing number of tenants  
- yield loss in isolation model/ shared table using LBAC technology                                   |                                                                                                       |
3. The Proposed Approach

An efficient Scalable mechanism is able to meet the desired quality of service, also optimal cost for users, On the other hand, load balancer using appropriate distribution in system samples will be able to reduce the frequency of the system needed to be scaled up as much as possible to preserve stability of system. As mentioned before, the purpose in this paper is to present an approach to optimize cost of scalability by use of learning automata.

The proposed approach, which is called GA-ADAPT is developed based on ADAPT. To do this, a genetic algorithm is utilized to overcome problems of ADAPT method and optimize response time once the number of features increases. Genetic algorithm is able to quickly and confidently solve complex issues that resist against the traditional solutions [14]. The fundamentals of genetic algorithms are:

1. Generating random population of n chromosomes
2. Checking the fitness function of each chromosome x in the population
3. Creating a new population by repeating following steps:
   3.1 Selection of two parent chromosomes from a population based on their fitness
   3.2 Considering certain amount for the possibility of applying the crossover operator and then combine parents to create offspring. Without any combination, the offspring will be the parents.
   3.3 Considering the possibility of mutation and then change of children per locus
   3.4 Replacement of new children in the new population
4. Using new population for the algorithm executions
5. To finish execution of the algorithm if stop circumstances are observed and returning the best solution in the current population
6. Going to step 2

According to the idea of genetic algorithms, the proposed approach stages are explained according to the following steps:

**Step 1. Initial Population**

The initial population of GA-ADAPT is equal to a set of chromosomes which, each chromosome is equivalent to a feature and the length of each chromosome is equivalent to the number of tenants or genes. So, the first generation of the population of n chromosomes is randomly generated.

**Step 2. Fitness Function**

The main goal is to find the number of basic and complementary tables, therefore, similarity between features should be found to minimize category that, to do it we can use Jaccard interval.

\[
Jaccard(V_x, V_y) = 1 - \frac{|V_x \cap V_y|}{|V_x \cup V_y|}
\]  

(1)

Then, for each chromosome population, the fitness function is calculated.

**Step 3. Selecting**

We choose the Selection operator for GA-ADAPT. The proper number of chromosomes are selected based on fitness rate and Selection Operator. Chromosomes that have a lower
fitness number may be chosen several times in the production processes, while the chromosomes that have high fitness may not be selected at all.

**Step 4. Reproduction**

The next step is the production of second-generation population. At this step, a new population is selected to enter the next stage of the algorithm. This is done by comparing the fitness of chromosomes. Chromosome with minimum value of fitness is selected. One of the methods of selecting new population is that a part of the future population are from the current population and the rest are selected from the new children. So the chromosomes with less fitness value are selected.

**Step 5. End**

Once the similarities between all feature vectors are evaluated, the algorithm reaches to the final stage and we have a series of basic tables includes original features and some additional tables with additional features. In Figure 1 and 2 executive steps of GA-ADAPT and ADAPT approaches are shown.

![Figure 1. The Executive Steps of GA-ADAPT](image-url)
4. Performance Evaluation

Cloudsim [15] provides an extensible and generalized simulated framework capable of modeling, simulation and testing of structures and functional services of emerging cloud computing, which allows users to concentrate on designing specific systems in accordance with their demands without importance of having a low-level details related to cloud-based services and structures [16]. To determine the performance of the proposed method, input data, which includes fields for each table, the number of tenants and how to deploy data on tenants, have been changed at each round of simulation. In this structure, each tenant has a table that holds a number of records. For recognition of effective fields or the main features of each tenant, if the field is NULL, then it is that main field of table. After identifying the main features and extras, simulation with Cloudsim would be done. Then both the method ADOPT and the proposed method GA-ADAPT in terms of selection time of important features and average response time to the tenants are compared. Sent requests randomly are generated by the normal distribution, so in all cases of simulation situation of sent requests would be similar. Simulation parameter values that have been selected for our simulation are shown in Tables 2 and 3. In this simulation, there are three hosts that its characteristics are shown in Table 3.
### Table 2. Specification of Data Center

<table>
<thead>
<tr>
<th>Architecture</th>
<th>X86</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Linux</td>
</tr>
<tr>
<td>Virtual Machine Management</td>
<td>XEN</td>
</tr>
</tbody>
</table>

### Table 3. Host Specification

<table>
<thead>
<tr>
<th>Server Name</th>
<th>Processor kind</th>
<th>Number of cores</th>
<th>Frequency (MIPS)</th>
<th>main memory (GB)</th>
<th>Band width</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Intel Xeon 2697</td>
<td>12</td>
<td>4061</td>
<td>16</td>
<td>1 Gbit/s</td>
</tr>
<tr>
<td>S2</td>
<td>Intel Xeon 5660</td>
<td>6</td>
<td>3070</td>
<td>8</td>
<td>1 Gbit/s</td>
</tr>
<tr>
<td>S3</td>
<td>Intel Xeon 5620</td>
<td>4</td>
<td>2170</td>
<td>4</td>
<td>1 Gbit/s</td>
</tr>
</tbody>
</table>

We define four scenarios to evaluate the proposed approach. Each scenario has a number of fixed clients and number of fields is changed. In each simulation by changing the number of fields, the number of requests is changed from 1000 to 20000 so the changes in response time is evaluated with two methods. Two criteria of selection time of features and response time to requests are investigated and their results are analyzed.

Selection time of features: Time that algorithm spent to select the original and additional features.

Response time: is the exact time difference between the time of request and the time of delivery to the user. Each request includes random access to 100 rows of the table of main features and additional features. Table 5 shows the structure and characteristics of each scenario.

### Table 4. Evaluation Scenarios

<table>
<thead>
<tr>
<th>description</th>
<th>Objective</th>
<th>Maximum number of fields</th>
<th>Number of tenants</th>
<th>scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of requests at the maximum limit of</td>
<td>Calculation of feature selection time of and</td>
<td>5, 7, 15, 20</td>
<td>50</td>
<td>1st. scenario</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10, 15, 20, 30</td>
<td>100</td>
<td>2nd. scenario</td>
</tr>
</tbody>
</table>
the number of fields is changed from 1,000 to 10,000 every 1,000 cycles so, using it we can calculate the response duration.

<table>
<thead>
<tr>
<th>response time in two methods of GA-ADAPT and ADAPT</th>
<th>50, 70, 100, 150</th>
<th>500</th>
<th>3rd. scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>150, 200, 250, 300</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

4.1. Evaluation of First Scenario

Given that the main objective of data management is for multi-tenant structure, one of the most important indicators is selection time of original and additional features. In this section, the number of tenants has been considered 50 and maximum of fields have been considered 5, 7, 15, 20. Selection time of original and additional features in two GA-ADAPT and ADAPT methods in 4 described cases have been investigated in Figure 3.

![Feature Selection Time](image)

**Figure 3. Selection Time of Original and Additional Features (the Number of Tenant 50)**

Investigating Figure 3, it is determined that with the lower number of fields, ADAPT algorithm works faster, but when the number of fields increases, genetic algorithms consumes less time for feature selection.

Figure 4 shows number of selected fields in each case with two GA-ADAPT & ADAPT methods.
To check the response time of the algorithm GA-ADAPT and ADAPT, the maximum number of fields 20 is considered. Then, the number of requests has been changed from 1,000 to 10,000 with 1,000 intervals to calculate the response time. Figure 3 shows response time to the requests in both algorithms.

According to Figure 5, response time of algorithm GA-ADAPT is less than ADAPT algorithm. Specifically, using GA-ADAPT algorithm in data management and identifying original and additional features increase speed to respond the requests compared to ADAPT algorithm.

4.2. Evaluation of Second Scenario

In this section, number of the tenants has been considered 100 and a maximum field of 10, 15, 20 and 30 have been considered. Selection time of original and additional features in two GA-ADAPT and ADAPT methods in 4 described modes has been investigated in Figure 4.
Figure 6. Selection Time of Original and Additional Features (the Number of Tenants 100)

Focusing on Figure 6, it is determined that the performance speed of GA-ADAPT algorithm to choose original and additional features is higher than ADAPT algorithm.

Figure 7 shows selected main fields in each case with two GA-ADAPT and ADAPT methods.

Figure 7. The Number of Selected Main Fields (the Number of Tenants 100)

According to the previous simulation, requests are sent randomly with a normal distribution to measure response time to requests. Figure 8 shows response time to requests in two GA-ADAPT and ADAPT algorithms.

Figure 8. Response Time to Requests (the number of tenants 100)
According to Figure 8, the speed of response of GA-ADAPT algorithm is higher than ADAPT algorithm.

4.3. Evaluation of Third Scenario

In the third evaluation we have considered 500 tenants and a maximum fields of 50, 70, 100 and 150 have also been considered. Selection time of original and additional features in two GA-ADAPT and ADAPT methods in 4 mentioned cases has been studied and compared in Figure 9.

![Feature Selection Time](image)

**Figure 9. Selection Time of Original and Additional Features (the Number of Tenants 500)**

Looking at the Figure 9 it is clear that performance speed of GA-ADAPT algorithm to choose original and additional features is higher than ADAPT algorithm. Figure 10 shows selected main fields in each case with two GA-ADAPT and ADAPT methods.

![Main Feature Selection](image)

**Figure 10. The Number of Selected Main Fields (the Number of Tenants 500)**

1000 to 10000 requests are sent and their response time is measured. Figure 9 shows response time to requests in two algorithms GA-ADAPT and ADAPT.
According to Figure 11, response time of GA-ADAPT algorithm is less than ADAPT algorithm. So, using the GA-ADAPT algorithm in data management and identifying original and additional features increase the speed of response to the requests of the ADAPT algorithm.

4.3. Evaluation of Fourth Scenario

In the fourth evaluation we have considered 1000 tenants and a maximum fields of 150, 200, 250 and 300 have also been considered. Selection time of original and additional features in two GA-ADAPT and ADAPT methods in 4 mentioned cases has been studied and compared in Figure 12.

According to Figure 12, the speed of performance of GA-ADAQPT algorithm in original and additional features selection is higher than ADAPT algorithm. Figure 13 shows the number of selected main fields in each case with two GA-ADAPT and ADAPT methods.
Figure 13. The Number of Selected Main Fields (the Number of Tenants 1000)

Figure 14 shows response time to requests in two algorithms GA-ADAPT and ADAPT.

Figure 14. Response Time to Requests (the Number of Tenants 1000)

According to Figure 14, response time of GA-ADAPT algorithm is less than ADAPT algorithm. So, using the GA-ADAPT algorithm in data management and identifying original and additional features increase the speed of response to the requests of the ADAPT algorithm.

4.3. Total Evaluation

According to the results of assessments carried out, average of two important criteria of previous evaluations that included the selection rate of original and additional features and response time, were determined and compared. According to Figure 15, if the number of features and tenants is higher, using GA-ADAPT algorithm has much higher performance than the ADAPT method.
Figure 15. Comparison of Average Feature Selection Time in Two Algorithms GA-ADAPT and ADAPT

Figure 16. Comparison of Average Response Time in the Algorithms of GA-ADAPT and ADAPT

According to Figure 16, as the number of tenant increases, response rate in data management method of GA-ADAPT increases compared to ADAPT method.

5. Conclusion and Further Work

Cloud computing has been one of most interested topics in the area of multi-tenant data management. In this article we provide a genetic-based approach to achieve high scalability, and performance with small storage space requirements. The proposed approach, GA-ADAPT uses cloud and ADAPT approach to improve the quality of services and to provide data management more dynamically. In addition to take into account QoS metrics such as cost, higher performance and scalability were achieved. The important performance metrics including feature selection time and average response time were evaluated by simulations. The results show that GA-ADAPT has better performance compared to ADAPT by increasing the number of tenants as well as the number of fields, the response time is still shorter using this approach. For future research one can work on comparing to other approaches of data management and improving service quality, using the GA-ADAPT method in the Federation of clouds and the use of meta-heuristic and
smart algorithms, such as mutation of leap, frog, bat, cuckoo search rather than genetic algorithms.

References


Authors

Reyhaneh Kheiri received the B.S.C degree in Software Engineering from University Azad Arak, Iran in 2009, and M.S.C degree from Azad University of Mahallat, Iran in 2015, respectively. Her research interests include Cloud Computing, Multi Tenancy and Data Management.

Alireza Taghizadeh received his BS and MS in Computer Engineering from Iran University of Science and Technology (IUST) and Sciences and Research Branch of Azad University (IAU), Tehran, Iran. He obtained his PhD in the area of Network and Communication from University Sains Malaysia (USM) in
2013. He was formerly with Iran Telecom, Research Center (ITRC) as a Senior Research Engineer in IP-based core networks. He is currently a lecturer in Islamic Azad University of Parand. His research interests include Cloud computing, IP mobility, handover modeling and evaluation.

Elaheh Kheiri received the B.S.C degree in Software Engineering from University Azad Arak, Iran in 2009, and M.S.C degree from Azad University of Mahallat, Iran in 2015, respectively. Her research interests include Cloud Computing, Multi Tenancy, Utilization and Resource Allocation.

Mostafa Ghobaei Arani received the B.S.C degree in Software Engineering from IAU Kashan, Iran in 2009, and M.S.C degree from Azad University of Tehran, Iran in 2011, respectively. He’s Currently a PhD Candidate in Islamic Azad University, Science and Research Branch, Tehran, Iran. His research interests include Grid Computing, Cloud Computing, Pervasive Computing, Distributed Systems and Software Development.