CHAPTER 4

A FRAMEWORK FOR EFFECTIVE RESOURCE ALLOCATION IN CLOUD ENVIRONMENT

4.1 INTRODUCTION

A Multi-Agent based Dynamic Resource Allocation (MADRA) for resource investigation and allocation process on the server side is presented in Chapter 3. Infrastructure of Cloud is a complex system due to large number of heterogeneous resources that need to be shared. Managing and allocating resources in Cloud is a major issue in Cloud computing. In this Chapter, a user-friendly framework is proposed for improving the resource management, and resource allocation in Cloud environment. This Chapter focuses on providing a request aware framework for improving the efficacy of resource allocation. This framework is integrated with a request processor, response provider, resource allocator, local manager and global manager. These components individually do their activities automatically and allocate the appropriate resources dynamically to the user.

4.2 AN EFFICIENT FRAMEWORK FOR RESOURCE ALLOCATION IN CLOUD ENVIRONMENT

The internal structure of the Cloud is considered as a client-server mode having better performance in terms of processing time and resource allocation efficiency during multiple user requests for a task. In this
framework, local managers and global managers are used to validate the user requests and allocate the resources.

In the proposed framework, the general process of resource allocation is as follows: Let \( R = \{R_1, R_2, R_3, \ldots, R_n\} \) be the set of \( n \) available resources and the entire details about the resources are stored in a server registry. The users \( U = \{U_1, U_2, \ldots, U_m\} \) send their requests \( REQ = \{Req_1, Req_2, \ldots, Req_k\} \) to Cloud for resources. The proposed framework allocates efficient resources to user request. It is well known that when a normal internet user becomes a Cloud user, they should register with their detailed information in Cloud. User information is represented as \( UI = \{U_{id}, U_{pwd}, U_{name}, U_{IP}\} \) which is stored in Database i.e. user information.

In Cloud, security can be provided in user level verification, data level verification and data storage level verification. In this section, the user level security is applied for improving the security. The user information such as id, Cloud password, name, and ip address and is verified for checking the user whether the user is valid user or not.

The resources can be allocated if the information given by local and global managers is valid. When a user \( U_i \) sends a request \( Req_i \) it will be forwarded to the local manager. The local manager verifies the original information about the user, user system, and the request pathway with the stored user information in the server and also validates the user requests. If user request is valid, then, the user request requirement information is passed to the Global manager. The Global manager retrieves the resource information from server registry and match with user requirement information. If both information matches, then, it directly connects the corresponding resources to the user, or else it replies that there is no resource like that as shown in Figure 4.1. If the same resource is requested by more
than two users at the same time the resources are allocated according to the priority of the users. After successful completion of the resource allocation, availability of the resources information is updated dynamically in the server registry. The status information maintained by the Local manager and the global managers are cleared dynamically for storing status of new resource information.

The server registry contains all the information about the deployed private and public resources in the Cloud. While deploying the resources the priority information is also registered and the utilization time is assigned according to the priority of the resources. The resources are divided into various categories such as No-Priority, Low-Priority, High-Priority. If the resource is having High-Priority, then, the time scheduling scheme provides only less time. The user can utilize the resources within a stipulated time.

![Diagram: Request Aware Framework for Resource Allocation](image)

**Figure 4.1. Request Aware Framework for Resource Allocation**
Figure 4.2 Steps Involved in Resource Allocation

Steps involved in resource allocation in Cloud as shown in Figure 4.2. The proposed algorithm for the resource allocation scheme is given below. The resource is allocated if the resource is free and status is true. The status of the resource becomes true then only it can be allocated to the requested user. The status of the resource will be true if the resource is unassigned to any other user and the resource should be in the nearest server.
ResourceAllocation()
{
    if (user information == valid) then
    {
        if (Ri ∈ R with status(Ri) = True ) then
        {
            ManagePriority();
            Ui gets Ri;
            status(Ri) = allocated;
        }
        else
            Request is rejected;
    }
    else
        Display (‘User Request is invalid’);
} // end of ResourceAllocation

ManagePriority()
{
    if (Resource priority == High)
        settime = less;  // for High_Priority resource
    else
        settime = normal; // for Less_Priority
} // end of ManagePriority()

4.2.1 Resource Management

In this section, the resource management is obtained by arranging the resources in such a manner that according to the index and the number of times the resources are requested frequently. Like web-ranking application, a variable is maintained for counting the number of times the resources are
requested and allocated. The resources are arranged in Cloud where the location of the resource behaves like cache memory in a computer system. It helps to retrieve the resources very fast and effectively.

4.3 EXPERIMENTAL RESULTS AND DISCUSSION

The resource allocation process presented in this chapter is simulated in Cloud Simulator integrated in Netbeans software. In this simulation the number of users, number of tasks and number of servers are assigned. During the execution, the number of users, tasks and servers are changed and the results are verified. It is also assumed that some of the systems are considered as clients and from the clients the requests are passed to the nearest server. The local Manager and the global managers are software modules deployed in the network systems which are called as Distributed Component Object Model (DCOM) modules. It is a module in which .exe, .com or .dll are deployed in Cloud.

Using this environment the proposed approach is experimented and the results are verified. The incoming requests are verified as either valid or invalid requests according to status parameters, verified by the local Manager. The experiment is done for a varying number of rounds for a varying number of requests. For 10, 20, 30, 40 and 50 requests the number of invalid requests is 0, 2, 2, 4 and 3 respectively and the remaining requests are validated by the local manager. To avoid conflicts in scheduling the resources effectively, all the resources are assigned different priorities such as high priority, low priority and no-priority in terms of time and energy. According to the priority of the resources only the user can utilize the resources and the same resource can be re-allocated to the new users. According to the priority, the resources are arranged in the registry as shown in Table 4.1.
Table 4.1 Categories of Resources

<table>
<thead>
<tr>
<th>S.No</th>
<th>Resources Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High Priority</td>
</tr>
<tr>
<td>2</td>
<td>Low Priority</td>
</tr>
<tr>
<td>3</td>
<td>No Priority</td>
</tr>
</tbody>
</table>

The number of incoming requests is validated by the local manager and the global manager separately. The local manager verifies the user information and validates the requests. The global manager validates the requests and the resources and allocated the suitable resources to the appropriate requests. The validation of the requests done by the local manager and the global manager is shown in Figure 4.3. Once local and global managers complete the request validation, it is essential to check the status of the resources. If the requests are valid and the statuses of the resources are valid then the appropriate resources can be allocated.

The resources allocated to the corresponding requests are computed experimentally and the results are shown in Figure 4.4. From this figure, it is clear that after validation of the requests 80% of the requests are allocated by the relevant resources. This work presents an efficient methodology where the resource allocation is validated in three different stages as it reduces waste of time and energy. Also to evaluate the step by step procedure of the proposed approach the number of requests validated in each stage like by local manager, global manager, and status of the resources is validated separately and the resources are allocated are shown in Figure 4.5.
Figure 4.3 Valid vs. Invalid Requests

Figure 4.4 Number of Requests Allocated by Resources
This chapter mainly focuses on reducing the resource allocation time and increasing the accuracy of choosing the appropriate resources. Before allocating a resource to any incoming user requests, the request and the resources should be validated to avoid mis-allocation. Also, user validation provides user level security and increases the QoS in various aspects. If the user request is valid, the resource is also valid and appropriate, then the resource allocation is accurate. It improves the QoS in terms of response time, energy consumption and accuracy.

#### 4.4 CONCLUSION

This Chapter discusses resource allocation by assigning a local and global manager in the Cloud environment. Since the environment is vast and the number of resources, clients, and the servers is more in numbers, it is necessary to validate the requests in each stage for allocating the resources.
After successful validation, the resource is allocated for the valid requests. From these results, it is clear that after validation of the requests 80% of the requests are allocated by the relevant resources. In the experimental results, it is concluded that validation based on resource allocation reduces the response time, energy consumption, cost and accuracy.