DECISION MAKING FRAMEWORK FOR RESOURCE ALLOCATION IN DYNAMIC VM's AND CLOUD MACHINE ENVIRONMENT DURING DISASTER RECOVERY.

A.PUNITHA* & Dr. NANCY JASMINE GOLDENA**

*RESEARCH SCHOLAR, MANONMANIAM SUNDARANAR UNIVERSITY, ABISHEKAPATTI, TIRUNELVELI, TAMILNADU, INDIA

**DEPARTMENT OF COMPUTER APPLICATIONS, SARAH TUCKER COLLEGE, TIRUNELVELI, TAMILNADU, INDIA.

Received: July 10, 2018 Accepted: August 21, 2018

ABSTRACT *Challenges are related with the allotment of limited resource assets to relieve the effect of natural disasters inspire fundamentally a new theoretical inquiries for dynamic decision making in coupled human and natural systems. In this work an intelligent agent based dynamic scheduling framework is proposed. It comprises of two independent components: the agent and the resource optimization. The agent chooses the most suitable priority rule according to the conditions in real time, while resource optimization performs scheduling activities utilizing the rule chosen by the job scheduler.*

Keywords: *Resource Allocation, Dynamic Agents, Dynamic Resource Machines, Decision Making Framework.*

1. Introduction

Over the past million of peoples lost their lives due to disasters worldwide. Natural and man-made disasters, such as earthquakes, floods, plane crashes, or major nuclear facility malfunctions, pose an ever-present dare to public emergency services. Disaster response and recovery efforts need timely interaction and coordination of public emergency services in order to save lives and property. Existing so-called disaster management systems [16, 17] usually are mere information systems, which are utilized for graphical representation of disaster-relevant data. However, all these systems do not permit the next and more main procedure, namely decision support by providing an optimized schedule for the obtainable resources to the areas requesting facilitate.

Resource Allocation is the progression of passing on obtainable resources to the required by the client over the internet during the disaster time. Resource allocation starves services if the allocation is not controlled accurately. Resource provisioning solves that problem by allowing the service providers / agents to maintain the resources for each individual module. Resource Allocation Strategy is all about integrating virtual machine or cloud machine activities for utilizing and allocating scarce resources within the limit of VM or cloud environment so as to gather the needs of the cloud / web application. It requires the type and amount of resources required by each application in order to complete a user job. The order and time of allocation of resources are also an input for an optimal RAS.

To progress the identification and management of response assets in a mass-casualty incident, as well as to help coordinate the starting response, this work propose a decision making framework for resource allocation in disaster management.

2. Review of Literature

The essential element of resource management is the discovery procedure. It engages seeking for the suitable resource types obtainable that match the application requirements [3]. The process is supervised by the cloud service provider. J. Espadas et al. in paper [4] discuss about over and under provisioning of cloud resources, although the peak loads can be successfully calculated, thus without having an efficient elasticity model, costly resources are exhausted during nonpeak times or revenues from probable customers are lost after experiencing a deprived service.

V. Vinothina et al. [5] surveyed that resource contention, scarcity of resources, resource fragmentation and over provisioning should be avoided, so that allocation and usage of the resources will be optimal. There is a necessary of a new method to discover the causes of the computing resources become exhausted and out of capacity to serve the client. A new solution must present a new technique that will trigger and improve the computing power to offer the wanted resources on time to clients without starving its current capacity. The proposed strategy may be appropriate at the infrastructure provider site since the important resources for the cloud is controlled by the service provider.

These proposals are based on machine learning techniques to model and manage application performance. For example, Wildstrom et al. [6] use M5 trees to learn a model for predicting whether the revenue of a cloud provider will increase by changing the memory allocation. In the approach presented by Bodik et al. [7], a machine learning technique is used to learn a performance model. Kundu et al. [8] use ANN to model application performance in virtualized infrastructures using multiple resource knobs.

Wildstrom et al. [9] also use machine learning for modeling performance based on low level system metrics to find the best configuration to increase the throughput. The difference of our approach to these proposals is that they train a single model offline for all workload and resource allocation combinations, while we train the model online. Bitirgen et al. [10] use ANNs to predict performance and support online model training, but they consider resource allocation at the multiprocessor chip level, while we manage resources at the CPU and memory level in a virtual environment.

3. Proposed Framework

3.1 Problem of Existing Researches

The existing methodologies such as IGTRAP and IGTRAP-PRRAPS are employed to perform the resource allocation. IGTRAP approach gives proficient management of resources for services running in vast scale geographically distributed systems. In the event that a natural disaster ought to happen in the information resource center, provoke data recovery can be effortlessly and safely accomplished by making utilization of a monstrous number of generally distributed wired PCs, versatile PCs, PDAs managed by supervisory servers which are expanded however functionally joined.

The main objective of IGTRAP-PRRAPS is to achieve the utilization of the node while allocating the resources. Thus to construct an optimized resource allocation model by enhancing the IGTRAP of the adaptive resource allocation system with the parameter selection is designed. The enhanced IGTRAP-PRRAPS is employed, to attain the optimized resource allocation scheme that determines the optimal virtual machine to achieve the resource effectively hence, it reduce the resource allocation time.

The existing research work towards resource allocation doesn't focus on synchronous factor of VMs for which reason task scheduling among the VM is still unsolved issue. The decision of resource allocation are mainly implemented using deterministic approach which is known for its overhead creation over highly distributed network like cloud.

Therefore multiple research contribution in existing system there is no a single benchmarked model to be spoken of in this regard. The potential characteristics of VM were never fully exploited and were always seen as a single entity whereas a novel framework with chain of VM with some significant behavioral characteristics modeling may add new functionalities to minimize task completion time.

Consequently different research commitment in existing framework there is no a solitary benchmarked model to be discussed in such manner. The potential qualities of VM were never completely misused and were dependably observed as a solitary substance while a novel structure with chain of VM with some huge social attributes displaying may add new functionalities to limit errand fruition time.

Hence, there is a need of a novel model that performs autonomous decision making practices in the dynamic environment of cloud in order to facilitate resource allocation.

3.2 Proposed Model

In this work focuses on vertical resource scaling and present an approach to assist the **Triple-N Resource Allocation Framework for vm's and** cloud to optimize resource allocation to VMs running on the physical machines. This framework manages the N-Customers, N-Agents and N-VMs. The proposed approach addresses the mentioned performance-power tradeoff by expressing the two conflicting objectives of application performance and power consumption in a utility function and optimizes the utility function at runtime.

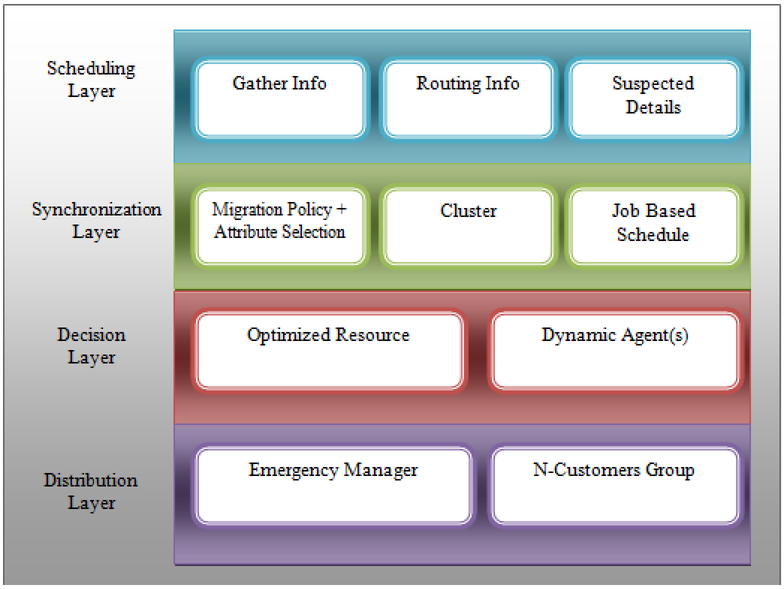

Figure 1: Architecture Diagram of Triple-N Resource Allocation Framework

The proposed framework considers the procedure of resource management for a large-scale in VM / cloud environment. Such an environment incorporates the physical infrastructure and associated control functionality that empowers the provisioning and management of cloud services. The perspective here by take is that of a cloud service provider, which hosts sites in a cloud environment. The cloud service provider claims and administrates the physical infrastructure, on which cloud services are given. It provides hosting services to site owners through a middleware that executes on its infrastructure.

Site owners provide services to their individual clients via sites that are hosted by the cloud service provider. Therefore, the client's demands are transformed to this virtual cloud server. Through this effective technique, the client's demands will be fulfilled successfully by serving the client without waiting. In this manner, the resources will be allocated dynamically. This work contributes towards building a middleware layer that performs resource allocation in such a cloud environment, with the following plan objectives

1) Performance objective: To think about computational and memory resources and the objective is to accomplish max-min fairness for computational resources under memory requirements.

2) Adaptability: The resource allocation procedure must dynamically and efficiently adapt to modify in the demand for cloud VM's allocation service.

3) Scalability: The Resource allocation process must be versatile both in the quantity of machines in the cloud and the number of sites that the cloud hosts. This implies that the resources consumed per machine in order to goal a required performance objective must improce sub linearly with both the number of machines and the number of sites.

Every layer of proposed framework described as follows:

1) Scheduling Layer

A) Gather Info

In addition to the fundamental resource controls exhibited earlier, administrators and clients can indicate flexible resource management policies for groups of VMs. This is encouraged by introducing the idea of a logical resource pool, a container that can be utilized to specify an aggregate resource allocation for a group of VMs. A resource pool is an entity that with related settings for every managed resource. It collected the information from the machines from various data centers such as Response Time, Transfer Rate, RTO, RPO ,Rate of QOS and Total Disk Size etc.

Resource pools are helpful in splitting large capacity into logically set users. Organizational administrators can utilize resource pool hierarchies to mirror human organizational structures, and to carry delegated administration.

B) Routing Info (Need to write)

Routing info focuses on the management of network resources by exploiting joint route selection and VM placement. A tenant usually subscribes for each of her VM a certain amount of resource, e.g., CPU and disk, and the uplink/downlink bandwidth SLAs from the data center operators. Fortunately, the operators have control over both where to place to the VMs that meet the resource demand, and how to route the traffic between VMs, at the time when a tenant is admitted. Routing info formalize it as an optimization problem, in which given a sequence of job arrivals, the network operator needs make the routing and placement decisions in order to minimize the network congestion in the long run. Routing info utilizes both synthesized and real traffic trace reported from an operational data center to evaluate for selection Resources.

C) Suspected Details

In the Cloud Computing service / Web Server HTTP Service, it is essential to allocate resources to clients as much as possible in any situation. So it is important issue to manage resources for reducing consumption of resources caused by implementing IDS. Some considerations when deploying IDS for defensive every single VM in Cloud Computing system are as follows. The security issues bring much more economic loss in Cloud Computing than in another type of systems.

This work proposes the Multi-level IDS method for implementing effective IDS in Cloud Computing system. Multi-level IDS method leads to efficient resource handling by applying differentiated level of security strength to clients based on the degree of anomaly. It is true that Cloud Computing is easy to be target of attack. For this reason, it is possible to judge all users and administrators as potential attacker and apply strong security policy to all traffic, but it is not efficient at all.

Suspected detail phase that divides security level into three, such as High, Medium and Low for effective IDS construction. High-level is a set which relates patterns of all recognized attacks and a section of anomaly detection procedure when it requires, for given that strong security services. Medium-level is a set of middle ranking which affect patterns of all recognized attacks to rules for providing comparatively strong security service. Finally, Low-level is a set for flexible resource management which applies patterns of chosen malicious attacks that arise with high frequency and that involve fatally to the system.

In Multi-level IDS scheme is an intrusion detection system that uses further resource when giving higher level security, because higher level security affect many rules than lower level. On the other hand, if intrusion detection system provides lower level security policy, then the amount of resource usage is decreased although the detecting power of attacks also drops. The assignment of virtual machine to a client is firmed in accordance with security level. The grade of virtual machine is proportional to client criteria of anomaly level. Anomaly levels of users are estimated by their behaviors during the usage of service based on saved user anomaly level in system.

2) Synchronization Layer

A) Migration Policy and Feature Selection

Dynamic consolidation of virtual machines (VMs) using live migration and switching idle nodes to the sleep mode allow Cloud / Data Center providers to optimize resource usage and reduce energy consumption. Due to the variability of workloads experienced by modem applications, the VM placement should be optimized continuously in an online manner. These policies create the system able to automatically alter its behavior depending on the workload patterns evaluated by the applications, once the behavior is altered then the energy efficiency and performance will be improved. The sequence of virtual machine Migration policies first checks if the host is overloaded, second virtual machine selection policy is used to select virtual machine's that need to be migrated from the host, third the selected virtual machine would be ready to be migrated from the overloaded hosts and the virtual machine placement algorithm is invoked to find a new placement for the virtual machine's to be migrated.

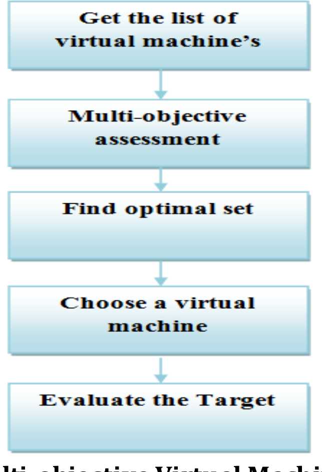

Figure 2: Flow of Multi-objective Virtual Machine migration policy

The work has investigated the efficiency of virtual machines migration based on various single objectives, and then verified the impact of using multi-objective evaluation to discover an optimal solution and to manage this solution to ensemble various situations.

Algorithm: MOVMP (Multi-objective Virtual Machine migration policy)

{

1: migration list L = {}
2: migration process x_{ij} = null
3: Calculate targets T-CPU , T-mem
4: Construct matrix X
5: X = MigrationPolicy(X)
6: While (True)
7: Add(L, x_{ij})
8: If (TCPU and Tmem) met
9: break
10: else
11: Update(X,i,j)
12: EndWhile
13: return L
14: Update (X,i,j)
15: remove row(i)
16: update column(j)

}

Feature Selection

In cloud computing / virtual machine environment all the resource are sharing by utilizing the virtualization innovation, hence the resources are depicted to as virtual machines and each virtual machine has a restricted limits, for instance processing power, memory size and system transfer speed .The speed of each virtual machine is ascertained in millions regulation for every seconds, work has preparing necessity acquired as far as millions directions.

In this paper, a proposed feature selection strategy *Preeminent Responsive Resource Allocation using Parameter Selection (PRRAPS)* is acquainted with allocates the resources in an effective way. The proposed methodology PRRAPS is a combination of Correlation based Feature Selection with Crow Search Algorithm. The key objective of this strategy is to search for the accessibility of the resources and allocate the undertaking for the virtual machines. The PRRAPS has their particular focal points while unraveling different issues. In this methodology the individual features of the algorithms are consolidated to display a novel hybrid algorithm to enhance the resource allotment scheme.

The virtual machine is get together with gathering of nodes utilizing correlation based selection, the fitness value is assessed based on the capacity of the virtual machine, and data transfer speed based on the function computed the P_{best} virtual machine is evalauted and in the following emphasis if fitness function is discovered superior to anything the past fitness value, displace the newer one as P_{best} lastly the best fitness value in the populace is considered as g_{best}. The procedure of crow search through that brings the optimal selection of the accessible virtual machine with ability for the resource distributed. The algorithm for the proposed PRRAPS is given below:

Algorithm PRRAPS

{

Input: J-Job List
Output: V_{0pt} –Virtual machine for optimized resource allocation
1: Initialize the random number n of virtual machine VM.
2: For each i= 1 to n
3: {
4: For each j= 1 to n
5: {
6: Fetch (vm_i,vm_j ϵ VM)
7: Evaluate Bandwidth (vm_i,vm_j) and Capacity (vm_i,vm_j)
8: If (vm_i > vm_j)
9: {
10: Gather Correlated VM from the list

```
VM_List = GetPopulation(vm_i)
}
}
}
For each i= 1 to m from VM_List
{
Compute Fitness (vm_i)
If (Fitness (vm_i) < P_best-i)
{
P_best = vm_i
}
Else
{
Choose next vm_i
}
Update P_best
}
g_best = evaluate best fitness value of P_best
Select gbest as optimal VM
New position < gbest VM in VM_List
If new position of VM in VM_List < previous VM in list
{
Elect new position of VM_List in optimal VM
Update the position of the virtual machine
Assign resource to virtual machine // for all VM_i.
}
}
```

At first the collected virtual machine is finished by utilizing correlation based selection among the n number virtual machines. Initiate the number of population in the job and the virtual machine, and after that assess the fitness capacity of each virtual machine by utilizing the crow search. The fitness value is computed by the data transfer speed and the limit of the virtual machine and for the resource, with the minimum execution time and deadline are acquired. The best fitness value is picked as the g_{best} virtual machine and after that allocates the resource to the optimal virtual machine.

B) Cluster:

Clustering is one of the most widely used methods for examining data analysis, with applications ranging from statistics, computer science, and biology to social sciences or psychology. In last year's, **spectral clustering** has become one of the majority popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm [15]. Compared to the traditional clustering algorithms such as k-means, spectral clustering can process non-convex data samples. In this work the object that needs to be clustered is overlap data which is non-convex and this is also here use spectral clustering but not k-means.

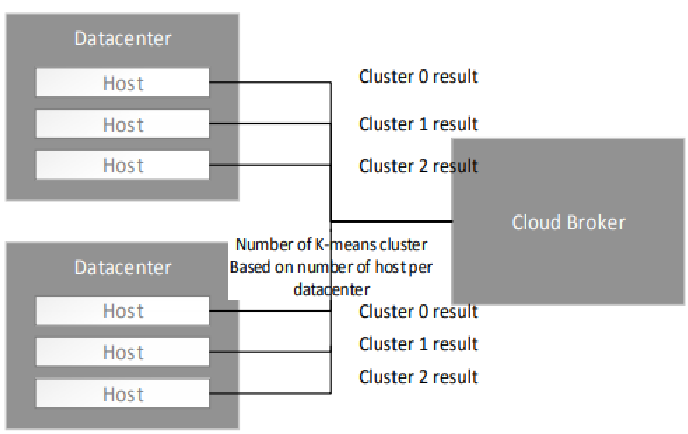

Figure 3: Clustering for each Datacenter

The major tools for spectral clustering are graph Laplacian matrices. Given a distance matrix C_{nxn} , the first step is to calculate the diagonal matrix D , where $D_{ii} = \sum_j C_{ij}$. Then we can obtain matrix L = D - C and L is the laplacian matrix. It is obvious that L is also a diagonal matrix. Then eigenvalue of the matrix L can be calculated which should be arranged from large to small. For k-cluster, feature vectors should be selected according to the first k eigenvalues and these vectors constitute a new matrix $R_{n\times k}$. The final step is to cluster row vectors of matrix $R_{n\times k}$ from which can gain clustering result. It is worth mentioning that the number of cluster k which decides the resource utilization is very important to the algorithm and should satisfy $k_{min} \le k \le k_{max}$.

C) Dynamic Job scheduling based on Thresholding (DJST):

Job scheduling of cloud computing or virtual machine refers to the process of adjusting resources between different resource users according to certain rules of resource use under a given cloud or VM environment. Resource management and job scheduling are the key technologies of resource allocation strategies. At present, there is not a uniform standard for job scheduling in cloud or virtual machine during resource allocation. Over-reliance on the scheduler may lead to some virtual machines excess while others are idle after dispatcher assigning tasks according to the load of virtual machines.

When this happens only solution is to allot tasks for next period according to what the response scheduling device gets. The method in different virtual machines is self-sufficient. A virtual machine does not have contact to other virtual machines' running conditions. So if one of the following two problems occurs, the execution efficiency of the tasks will be affected. One of the problems is that the dispatcher is out of joint, and the other one is the environment of virtual machines changes, resulting in the problems of some virtual machines and creating them not capable to send the data back to the scheduler.

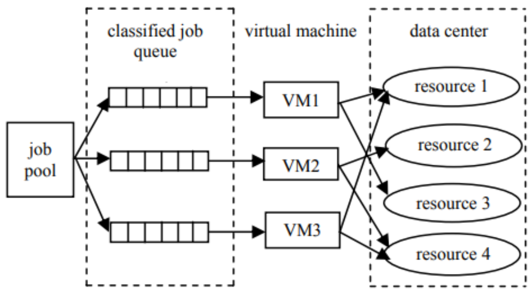

Figure 4: Job based Scheduling Model

To get the real-time response of the state of virtual machine, there are two ways. One of them is to construct a set of response mechanism between dispatcher and virtual machines to get the real-time response of the tasks load on virtual machine, and then create a real-time modification on job allocation upon the detail of virtual machines. The other one is to use the dynamic scheduling amongst virtual machines themselves to get the real-time state of the load of virtual machines. If overload happened then tasks could be modified and redistributed among virtual machines. By dynamic transmit in virtual machines, dynamic scheduling algorithm based on threshold can assign jobs and resources flexibly and diminish the efficiency impact caused by the synchronization among virtual machines.

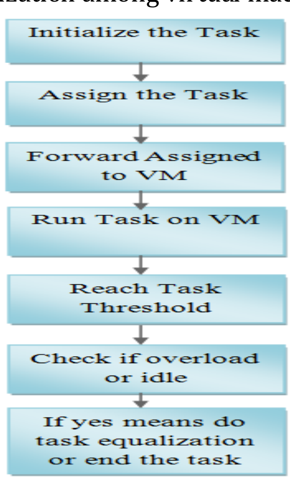

Figure 5: Process Flow of Job based Scheduling Model

The system depicts that if there are some virtual machines overload and some idle at a certain time, dynamic job adjustment is conducted to shorten the total cost time, thereby enhancing efficiency. However, task allocation between virtual machines refers to synchronization problems, which is also the major issue of the dynamic scheduling algorithm based on threshold. Since every virtual machine is independent to all together, in the other word they are non-interfering. They can carry out tasks in parallel. If virtual machines are synchronized, they inescapably bring affects to their performance. Therefore, the synchronization operation should be reserved to minimum range.

In order to diminish the impact of synchronization, two measurements are taken. Initially set the threshold. Synchronization is accomplished only when the virtual machine achieves a threshold. The larger the threshold is, the smaller the impact of synchronization will be. Second, limit the synchronization caused by two virtual machines. The smaller the number of virtual machines for synchronization is, the weaker the contact it brings.

3) Decision Layer

In this layer, the decision making procedure is experienced with the performance of optimization and Choosing Dynamic Agents for the optimized resource allocation for the planned job from the past layer. This layer comprises of the following perspectives:

A) Multi-Objective Virtual Machine Resource Optimization (MOVMRO):

In this work proposes a multi-objective resource optimization algorithm for mapping tasks to virtual machines in order to improve the throughput of the datacenter and reduce the cost without violating the Service Level Agreement for an application in cloud environment. Scheduling refers to the mapping or assigning a task to a specific virtual machine such that resource utilization increases. An efficient task scheduling algorithm improves the overall system performance and helps service provider to provide good quality of services.

In Resource selection broker plays an important role. Brokers have the list of virtual machines and its quality of service. A high performance virtual machine assign with the high quality of services. Broker takes the requests from the user and sends the request to the one of the virtual machine which meets the user requirement and the service level argument. The tasks in the group are chose sequentially and proposed to the Virtual Machine. After receiving the list of virtual machines cloud broker assigns quality of service to the virtual machines. Each virtual machines have different capability to execute different quality of service's tasks. Millions of instructions per second (MIPS) of a virtual machine is considered for assigning virtual machines quality of service. Virtual machine with high MIPS is a high quality of service virtual machine and virtual machine with low MIPS is low quality of service virtual machine.

Algorithm MOVMRO

{

Submit VMs List and Job List to the Intermediate
Create a list of job Job_{List} and VMs VM_{List} received
Initially create Job_i in the list
For each i=1 to Size (Job_{List})
{
For each j=0 to Size (Job_{List})
{
If (Job_j dominates Job_i)
Include Job_j to Job_{List}
Else if (Job_i dominates Job_j)
Include Job_i to VM_{List}
Else
Include Job_i and Job_j to VM_{List}
}
}
Sort the job list Job_{List} regarding the VM_{List}
Sort the VM_{List} in descending order
For each i=0 to Size(Job_{List})
{
If (j>=0)
{
Job_i is binds to VM_j
}

```
Else if(j==SUM(VM))
{
j=0
}
}
}
```

Broker is in charge for mediating negotiations between Service and cloud provider and such negotiation are driven by quality of service requirements. In multi-objective problems multiple objective functions are considered. In the proposed work, the main goal is to minimize the execution time of a task and it is achieved by selecting a task with minimum task size and minimum (low) quality of service value. The main problem is to bind set of tasks received by the broker to the received list of virtual machines, so that execution time of workload is reduced to minimal optimized time by using multi- objective task scheduling algorithm. This list is restructured in fixed time interval and enthusiastically at the pick time when number of requests enlarge rapidly.

B) Select Dynamic Agents

The issue of distributed resource allocation has been addressed from the starting point of agent-based research and application. Here algorithms or techniques have been created that particularly consider into account the decentralized system structure of multi-agent systems and their capacity to communicate and standard. An optimization problem comprises of searching the best solution, according to a given criterion, among a set of feasible solutions. Optimization algorithms are general step-by-step methodology for solving optimization issues. In other words optimization algorithm is said to resolve the issue if it can be applied to any instance of that problem to produce a feasible solution. Optimization algorithms can be exact if they discover the optimal solution or heuristic if the discover a good solution not essentially the optimal one.

The scheduling system outlined in this framework consists of two important parts: the Resource Optimization (RO) and the Dynamic Agent. RO is responsible for optimizing the scheduling process dynamically by communicating with the agent whenever it needs to select a job to assign to any machine. Whenever a job has arrived, the RO determines its details and puts it into related queue for its first operation prior to identifying the inter-arrival time for the next job. If new assignment and/or departure events happen, RO carries out such duties; otherwise it improves the simulation time up to the next arrival and accepts the next job. The simulation carries on as long as the stopping criteria are unsatisfied. In this case, the simulation stops when the predefined number of jobs is scheduled.

Whenever any assignment event happens, RO communicates with the agent asking it to identify an appropriate priority rule. The agent perceives the current situation from the RO, evaluates it and suggests a dispatching rule based on the available information, selecting the appropriate one of the following three rules: SPT (Shortest Processing Time), COVERT (C over T), and CR (Critical Ratio). RO applies the rule to select one of the waiting jobs in the queue. Then it assigns the job to the machine for the current operation and calculates it's finishing time on this machine as well as removing the job from the queue.

4. Evaluation Metrics

The performance metrics evaluation for the resource allocation framework with its existing system and methodologies to frame this work is illustrated in this section. The performance evaluation is computed for Clustering Techniques, Job Scheduling Techniques, and Resource Optimization Techniques. To measure the performance analysis the metrics such as Time Consumption, No. of clusters, accuracy, etc,.

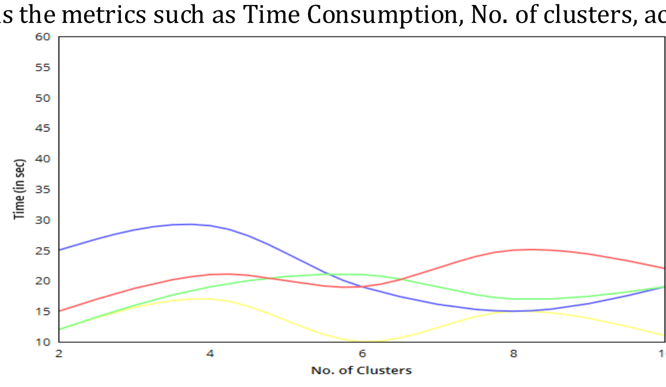

Figure 6: No. of Clusters Vs Time Consumption

Figure 6 depicts the clusters build up with the time. The figure depicts that the number of clusters build up with the time consumption. The figure describes that the time consumption for the cluster creation is less in proposed clustering k-means.

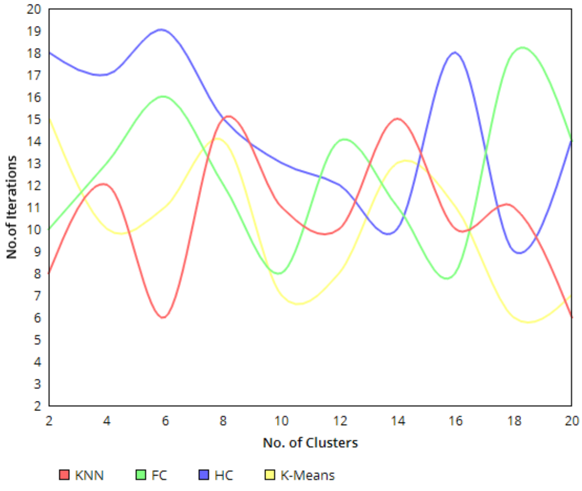

Figure 7: No. of Clusters Vs No. of Iterations

The no. of iterations for the cluster creation are reduced in level for the proposed clustering K-means is depicted in Figure 7.

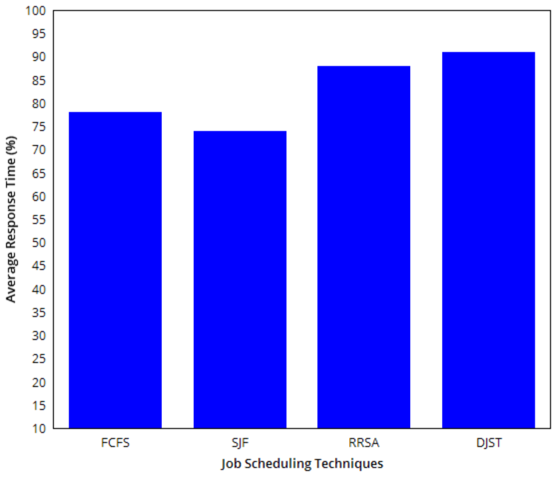

Figure 8: Average Response Time of Job Scheduling Techniques

The proposed strategy for job scheduling gives high response time for the requesting for job while booking is depicted in Figure 8.

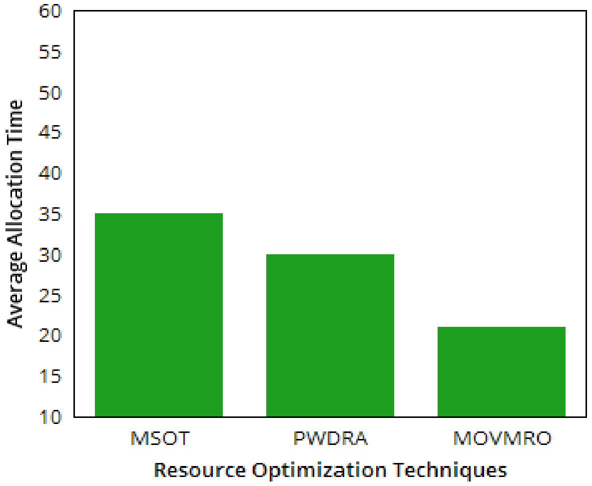

Figure 9: Average Resource Allocation Time Evaluation for Resource Optimization Techniques

The average resource allocation time evaluation is depicted in Figure 9. The resource optimization techniques are evaluated to measure the resource allocation time for the job scheduled, the proposed method MOVMRO consumes and responses in less time.

Figure 10 illustrates the wait jobs resource allocation with respect to the time response. The figure clearly depicts that the proposed optimization technique responses in less time with reduced waiting jobs than the other methodologies.

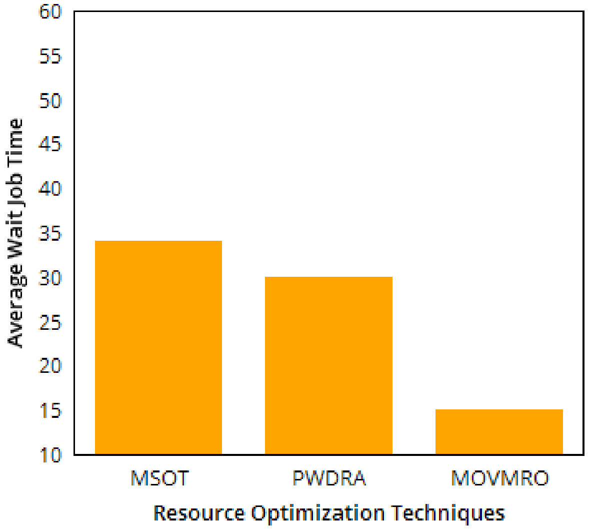

Figure 10: Average Wait Job Time Evaluation for Resource Optimization Techniques

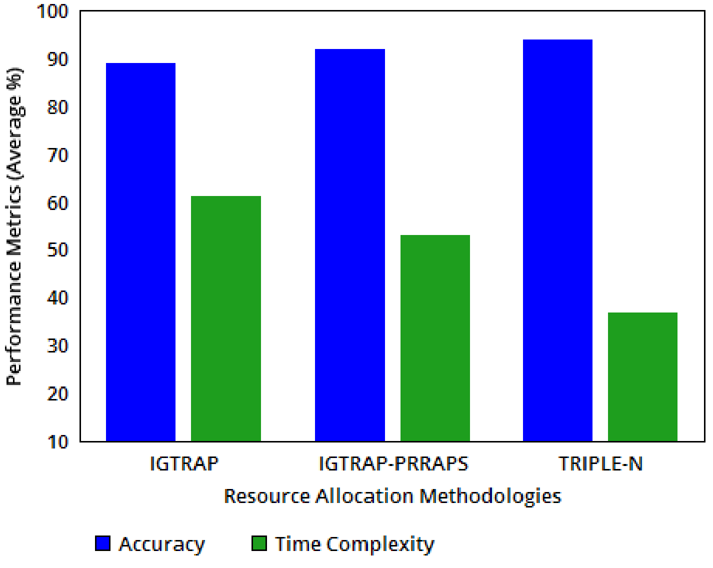

Figure 11: Performance Evaluation for TRIPLE-N Framework with other Techniques

Conclusion

In this work resource management approach for VM resource allocation in cloud / virtual machine has been presented. The proposed methodology TRPLE-N Framework is employed with the efficient allocation of the resources for the required clients with the dynamic allocation of the resources for the VM. The framework entirely includes the process of N-Clients, N-Agents and N-VMs to perform this efficient allocation of the resources during disasters. The task are scheduled with the job scheduling algorithm to know the required clients and the job are allocation through the framework with the verification of VMs availability and the capability to allocate the resource is acquired. Then the available VM is allocated to the number of clients for the efficient allocation of the resource. The performance evaluation of the proposed TRIPLE-N Framework provides better results on allocation of the resource than the existing methodologies such as IGTRAP and IGTRAP-PRRAPS.

6. References

1. Hsu M, Anen C, Quartz SR: The right and the good: distributive justice and neural encoding of equity and efficiency. Science 2008, 320:1092–1095.
2. G. Clarke. Mckinsey: Adopt the cloud, lose money: Virtualize your datacenter instead. http://www.theregister.co.uk/ 2009/04/15/mckinsey_cloud_report.
3. R. Buyya and R. Ranjan, "Special section: Federated resource management in grid and cloud computing systems," Future Generation Computer Systems, vol. 26, no. 8, pp. 1189–1191, June, 2010.
4. J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha, "A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures," Future Generation Computer Systems, vol. 29, no. 1, pp. 273-286, 2013.
5. V. Vinothina, R. Sridaran, and P. Ganapathi, "A survey on resource allocation strategies in cloud computing," International Journal of Advanced Computer Science and Applications, vol. 3, no. 6, pp. 97– 104, 2012.
6. J. Wildstrom, P. Stone, and E. Witchel. Carve: A cognitive agent for resource value estimation. In Proc. 5th IEEE International Conference on Autonomic Computing, pages 182–191. IEEE Press, 2008.
7. P. Bod´ık, R. Griffith, C. Sutton, A. Fox, M. Jordan, and D. Patterson. Statistical machine learning makes automatic control practical for internet datacenters. In Proc. Conference on Hot topics in Cloud Computing. USENIX Association, 2009.
8. S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta. Modeling virtualized applications using machine learning techniques. In Proc. 8th ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments, pages 3–14. ACM Press, 2012.
9. J. Wildstrom, P. Stone, E. Witchel, and M. Dahlin. Machine learning for on-line hardware reconfiguration. In Proc. 20th International Joint Conference on Artifical Intelligence, pages 1113–1118. Morgan Kaufmann Publishers Inc., 2007.
10. R. Bitirgen, E. Ipek, and J. F. Martinez. Coordinated management of multiple interacting resources in chip multiprocessors: A machine learning approach. In Proc. 41st Annual IEEE/ACM International Symposium on Microarchitecture, pages 318–329. IEEE Computer Society, 2008.
11. International Federation of Red Cross and Red Crescent Society (IFRC), 2010. World Disaster Report 2010; Focus on Urban Risk. Retrieved Jan 07, 2014, from
12. Haddow, G., Bullock, J., Coppola, D. P., 2007. Introduction to emergency management. Butterworth-Heinemann
13. Jyothi Sekhar, Getzi Jeba, "Energy Efficient VM Live Migration in Cloud Data Center", IJCSN International Journal of Computer Science and Network, Vol 2, Issue 2, April 2013.
14. Anju Bala ,Inderveer Chana ,"VM Migration Approach for Autonomic Fault Tolerance in cloud computing", Thapar University, Patiala, Punjab, India,2013.
15. U. von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, pp. 395-416, December 2007.
16. Fiedrich, F., F. Gehbauer, and U. Rickers, "Optimized Resource Allocation for Emergency Response after Earthquake Disasters," Safety Science, Vol. 35, 41-57, 2000.
17. Fiedrich, F., "An HLA-Based Multiagent System for Optimized Resource allocation after Strong Earthquakes," Proceedings of the 2006 Winter Simulation Conference, 2000.
18. Sherali, H.D., Desai, J. and Glickman,"Allocating emergency response resources to minimize risk under equity considerations," American Journal of Mathematical and Management Sciences, Vol. 24(3/4), pp.367-410. 2004