Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 8, August 2014, pg.395 – 403

SURVEY ARTICLE

Survey on Dynamic Resource Allocation Techniques in Cloud Environment

Mr. S.Natarajan[1], Dr. R.Pugazendi[2]

[1]Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode, TamilNadu, India

[2]Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode, TamilNadu, India

[1]snatarajan1988@gmail.com; [2] pugazendi_r@rediffmail.com

Abstract— The growth of cloud computing infrastructures carries innovative ways to make and control computing system by means of the flexibility present with virtualization technologies. In this framework, it concentrates on two main goals. First to provide initial virtualization and cloud computing infrastructures to make distributed large scale computing platforms from various cloud providers approved to run software involving large volumes of computation power. Subsequently developing methods to formulate these infrastructures are more dynamic. This method offers inter cloud live migration planning innovative approaches to utilize the inherent dynamic environment of distributed clouds. Cloud computing permits business customers to extent up and down their resource usage founded on requires. Several of grows in the cloud model appear from resource multiplexing during virtualization technology. In this survey, it presents detailed explanation of the dynamic resource allocation approaches in cloud for cloud consumers. From this study that utilizes virtualization technology to distribute data center resources dynamically based on application requires and maintains green computing by optimizing the number of servers in employ. This manuscript introduces the idea of "skewness" to determine the unevenness in the multi-dimensional resource utilization of a server. By reducing skewness, it can include different kinds of workloads properly and improve the overall utilization of server resource. Furthermore importance of limitations and advantages of using Resource Allocation in Cloud computing systems is also examined.

Keywords— Cloud computing, Virtualization, Migration, Green computing, resource management, Resource allocation

I. INTRODUCTION

Cloud computing permits users to extent up and down their resources based on requires. Cloud computing technology formulates the resources as a single point of access to the consumer and cost is pay per usage. Cloud computing is an emerging technology, where a pool of resources are associated in private and public networks and to offer these dynamically scalable infrastructure for application. Cloud computing is not application oriented and this is a service oriented. It suggests that the virtualized resources to the cloud consumers. Cloud computing offers dynamic provisioning and thus can distribute machines to store data and add or remove the machines following to the workload requires. Cloud computing platforms namely, those offered by Microsoft, Amazon, Google, IBM. Cloud computing is a platform for sharing resources exclusive of the knowledge of the infrastructure and can formulates it possible to access the applications and its connected data from wherever at any time.

Cloud environment offers the four types of cloud.

1. Public cloud.
2. Private cloud.
3. Hybrid cloud.
4. Community cloud.

Cloud computing present's three types of services

1. Software as a service (SaaS).
2. Platform as a service (PaaS).
3. Infrastructure as a service (IaaS).

The cloud computing environment guarantees to give that it attaches to the service level agreement with giving a resource as service and with requires. Moreover, day by day subscribers' requires are rising for computing resources and their needs have dynamic heterogeneity and platform insignificance. But in cloud computing environment, resources are distributed and if they are not appropriately distributed next it will result into resource wastage. An additional important role of cloud computing platform is to dynamically balance the load between different servers in order to avoid hotspots and improve resource utilization. Consequently, the main issues to be resolved are how as well as efficiently deal with the resources.

Need of resources are increasing significantly day by day. So it is important to distribute the resources correctly but static allocations have some boundaries. So it is important to move in to the dynamic resource allocation [10]. For using the virtualization techniques [1], can migrate virtual machines to physical machines efficiently. By doing this some machines goes to the idle state and turning off these machines will direct to save the energy. So maintains the green computing and thus resources can be dynamically distributed correctly. On a cloud computing platform, dynamic resources can be efficiently handled using virtualization technology. The subscribers with additional demanding SLA can be guaranteed by willing to help all the needed services within a virtual machine image and then mapping it on a physical server. This assists to resolve difficulty of heterogeneity of resources and platform irrelevance. Load balancing of the whole system can be switched dynamically by using virtualization technology where it becomes promising to remap virtual machine and physical resources according to adjust in load. Due to these benefits, virtualization technology is being systematically implemented in cloud computing. Moreover, in order to accomplish the best performance, the virtual machines have to completely utilize its services and resources by adapting to the cloud computing environment dynamically.

Cloud computing is based on the virtualization technology. Virtualization technology is utilized to distribute the data center resources dynamically based on the application demands.

Virtualization having two types,

1. Para virtualization.
2. Full virtualization.

Live Migration.

Virtual machine live migration knowledge builds it feasible to assigning between the virtual machines (VM's) and the physical machines (PM's) even as applications are running. Live migration enhances the resource utilization and offers the improved performance result.

It is a policy concern remains as how to choose the mapping adaptively so that the resource demands of VM's are met even as the number of PM's used is reduced. This is demanding the resources requirements of VM's are heterogeneous due to the various set of applications they run and differ with time as the workloads enlarge and shrink. The capacity of PM's can also be heterogeneous since several generations of hardware co-exist in a data center.

The main goal of this work is to achieve two aims in the proposed algorithm. Overload avoidance: The capacity of a PM has to be enough to satisfy the resource needs of all VM's working on it. Or else the PM is overloaded and can direct to degrade performance of its VM's. Cloud Computing develop into a standard for computing, infrastructure as a services has been emerged as a significant concept in IT area. By concerning this concept it can abstract the fundamental physical resource such a CPUs, Memories and Storage and present this Virtual Resource to users in the reserved Virtual Machine. Several Virtual Machines are capable to run on a unique physical machine. An additional significant concerns in Cloud computing is provisioning technique for allocating resources to cloud consumers. Cloud computing environment consists of two provisions. The objective is to achieve an optimal solution for provisioning resource which is the mainly significant part in cloud computing. To formulate a best decision that the demand price and waiting-time uncertainties are engaged into account to adjust the trade-offs among on-demand and oversubscribed costs.

The load balancing and suitable allocation of resources have to be guaranteed in order to develop resource utility. Thus, the significant aims of this research are to be determining that how to improve resource utility, how to schedule the resources and how to achieve efficient load balance in a cloud computing environment. Dynamic resource allocation is completed by using the

virtualization technology. In this virtualization, migration of the VM's to PM's is needed. For the better allocation, migration approach is employed here [6]. By defining the hot spot and cold spot, can balance the load and it avoids the overloading in the system's as well as it maintain the energy effectiveness.

There is an inherent trade-off between the two objectives in the face of varying resource requires of VM's, overload avoidance. It must maintain the utilization of PM's low to decrease the possibility of overload in case the resource requires of VM's increasing later, for green computing it must maintain the utilization of PM's practically high to formulate capable use of their energy.

A warm spot is described which is lower than the hot spot and higher than the cold spot. Warm spot will balance the load and if distribute resources in as the warm spot situation, it will never goes to overload and the ideal situation. Here servers as a cold spot if the utilization of all its resources are below a cold threshold. This signifies that the server is mostly idle and a potential candidate to turnoff to save energy. Following the allocation, the cloud controller discovers idle system and by turning off this idle system, preserves the energy.

II. STUDY ON WORKLOAD IN DYNAMIC RESOURCE ALLOCATION IN CLOUD COMPUTING

Survey consists of the relative mechanisms and the methods which are employed earlier and also the advantages and disadvantages of each technique are described briefly. According to the survey of the earlier mechanism, it finds that the current system implemented has more advantages.

Christopher Clark *et al*. [2] Live OS migration tool is used for cluster administrators, which is used to allow the division of hardware and software considerations, and consolidating clustered hardware into a single coherent management domain. In the case of any physical machine failure, the administrator will attempt to migrate OS instances to the alternative OS machines. Thus the original machine will be allowed for maintenance. Likewise in the case of congested hosts, OS instances will be rearranged across machines. Like this kind of situations the combination of virtualization and migration significantly improves manageability.

This is achieved by using a pre-copy approach where the pages of memory are iteratively copied from the source machine to the destination host, all without ever stopping the execution of the virtual machine being migrated. Snap shop consistent can be ensured by using page level protection hardware and an impact of migration traffic on real time services can be controlled by using rate-adaptive algorithm. If there is remaining page to be executed the final phase will pauses a virtual machine to copy the remaining pages to the destination. After copying the pages it will resume the execution there. In this work "pull" approach is avoided in order to avoid the missing page fault across the network. Hence this method adds a residual dependency of arbitrarily long duration, as well as providing poor performance.

Live migration is defined as running virtual machines or application without disconnection of client or application. In this process memory, storage and network connectivity of the original virtual machines will be transferred to the destination.

Migration processes have certain steps to perform.
Stage 0: Pre-Migration: This will start with an active VM present on physical host. The future migration can be speed up by selecting the target host previously. So that the resources required for migration will be guaranteed.
Stage 1: Reservation: A request is issued to migrate a from host A to host B. Initially confirm that the necessary resources are available on B and reserve a VM container of that size. Failure to secure resources here means that the VM simply continues to run on A unaffected.
Stage 2: Iterative Pre –Copy: In this stage all the available pages will be transferred from A to B. The forwarding iteration will copy only pages that are stained during the previous phase.
Stage 3: Stop-and-Copy: In this stage the running OS will be suspended at A and its traffic will be redirected to B. After redirection of OS instances the state of CPU and remaining inconsistent memory pages will be transferred to B. Finally there will be a consistent suspended copy of the VM at both A and B. The consistent copy present at A is considered to be primary and is resumed in case of failure.
Stage 4: Commitment: In this stage Host B indicates to A about its successful reception of consistent OS image. By receiving this message Host A will acknowledges this message as commitment of the migration transaction. In this case host A may discard the original VM, and host B becomes the primary host.
Stage 5: Activation: In this stage the migrated VM on B will be activated. The device drivers will reattached to the new machine and advertise their new version of IP address by using post migration code.

Marvin McNett *et al.* [3] propose an Usher which is used to balances the imposing requirements with the help of combination of abstraction and architecture. Usher used to provide a simple view of logical cluster of virtual machines, or virtual cluster. In this method users are allowed to create virtual clusters of arbitrary size, while Usher trying to multiplexes

virtual machines on available physical machine hardware. The core of this method is used to implement basic virtual cluster and machine management mechanisms, such as creating, destroying, and migrating VM's. This core is used by the Usher clients to manipulate the virtual clusters. The usher clients are the one who acts as the interface to the users and also used by higher-level cluster software. For example, an Usher client provides a command shell for users to interact with the system. And also it acts as an adapter for a high-level execution management system. This adapter will operate as an Usher client who creates and manipulates virtual clusters on its own behalf.

In this work two modules are proposed. In the first module Usher is enabled to interact with broader site infrastructure. Usher implements default behaviour for common situations, e.g., newly created VM's in Usher can use a site's DHCP service to obtain addresses and domain names. In addition to that Usher can be customized to implement specialized policies. Like in UCSD, Usher will allocate the IP address ranges to VM within the same virtual cluster. Second, pluggable modules enable system administrators to express site-specific policies for the placement, scheduling, and use of VM's. As from the result of this work, Usher give permission to administrators to take decision about the configuration details for their virtual machine environments and determine the appropriate management policies.

Alternatively, Usher creates a framework to allow the system administrators to state site-specific policies depends upon their needs and goals. This Usher core provides best-effort computing environment for general purpose. It imposes no restrictions on the number and kind of virtual clusters and machines, and performs simple load balancing across physical machines. Here believe this usage model is important because it is widely applicable and natural to use. Users need to express their resource requirement based on needs. For example it will be challenge for users since they do not know when and how long they require resources. Further, resource utilization will be limited from allocation and reservation. So that it guarantees that some resources will not go to idle state and it can be used for some other purposes. Conversely, elaborate policies are specified in Usher for controlling the placement, scheduling and migration of VM's.

Such kinds of policies are ranged from batch schedulers to share of dedicated physical resources. In order to avoid the confusion between policy and mechanism Usher maintains a clean separation between them. It provides a small set of mechanisms which is crucial for VM management. And also it provides a set of hooks which helps to integrate with existing infrastructure. The three main components of Usher system are: local node managers, a centralized controller, and clients. A client with set application will utilizes an usher client library send VM management request to the controller. The LNM in each and every physical node will interact with the VMM to process management operations like creating, deleting, and migrating. The role of local node managers is to collect data of resource usage from the VMM's and monitor local events. Resource usage updates and events will be reported by the LNM to the controller for use by plug-in and clients. The central component of the Usher system is the controller. When it receives authenticated requests from clients, immediately it will issue approved order to the LNM's. It maintains the proper communicates with the LNM's to gather usage details of data in order to manage VM running on each physical node. The controller provides event notification to clients and plug-in registered to receive notification for a particular event. In Plug-in module, the persistent system-wide state information is maintained to perform DDNS updates and doing external environment preparation and cleanup.

The API of every application is present in the client library which is used to interact with the usher controller. Whenever client need VM to manipulate or requires additional VM it will send request to the controller. The controller will decide whether to accept or reject these requests based on operational policy dictates. Client will act as an user interface to the system and users make use of client to manage their VM and monitor system rate. In general arbitrary applications make use of client library to register call backs for events of interest in the Usher system.

The running usher system is supported by few services. Several combinations of services can be obtained based on functionality desired and infrastructure provided by a particular site. Those combination are a database server for maintaining state information or logging, a NAS server to serve VM file systems, an authentication server to provide authentication for Usher and VM's created by Usher, a DHCP server to manage IP addresses, and a DNS server for name resolution of all Usher created VM's. Usher is configured by administrator to support any set of services not only the preceding list.

***Gong Chen** et al.* [4] proposed load skewing algorithms to allow considerable amount of energy reduction without avoiding user experiences, i.e. maintaining very small number of SID's.

With the knowledge of power consumption by connection servers (CPU, network, and memory intensive servers) provides insights on energy saving strategies. There is no IO can be obtained in normal mode except infrequent log writing. TO avoid this problem memory is pre allocated to avoid run-time performance hit. The considerable power consumption can be reduced by using pack connections and login requests to a portion of servers, and keep the rest of servers hibernating. Still the integration of login requests results in high utilization of those servers, which may degrade the performance and user experiences. So, it is important to understand the user experience model before address the power saving schemes for large-scale Internet service.

The processing transaction load can be kept under control if the number of connection and the login rates are enclosed. Due to these enclosures it will not take transaction delay as QoS metric. This proves that the maintenance of connection server load is very important. The shape of connection server load is achieved by using DS and load dispatching algorithm. The better

power savings with good user experience can achieved by maintaining better interaction among CS, DS, the provision algorithm and load dispatching algorithm in terms of SNA and SDA. This can also be achieved by trying to schedule before switch off the server. In general server with least amount of connections will be identified by dispatchers in order to schedule them to connect to other servers. The connection with other servers will made with controlled slow pace manner to avoid burden for remaining active servers. The number of SID's usage can be reduced by starving or shutting down a server for a period of time before doing schedule. In case of messenger servers, this method will lead to the exponential decay of number of connections made. This exponential decay is produced in the case of natural departure rate caused by normal user log with a time constant a little less than an hour, meaning that the number of connections on a server decreases by half every hour. For example, if the server starving for two hours then the number of SID will be less than quarter of what without starving. This shows that increasing starving time reduces efficiency in saving energy. The goal of this method is to keep a less number of tail servers which have fewer connections. So that, whenever user sends login requests, these servers will be act as reserve to handle login increases and surge, and give time for new servers to be turned on.

When user login requests rise down, these servers can be slowly drained and shut down. The problem of reduced efficiency is addressed by shutting down only tails servers. So that the number of SIDs will also be reduced and no re-login request or connection movement is created. But in the case of hardware side, the frequent turning off and turning on will lead to reliability problem.

This work tries to avoid the frequent turning on and turning off the same machines. To achieve this load prediction is used to avoid short term decisions. A better solution is to rotate servers in and out of the active clusters deliberately.

Pradeep padala *et al*. [5] proposed an Auto Control. It is a resource control system which is designed to adapt any dynamic changes that are occurred in the shared virtualized infrastructure. It is a combination of an online model estimator and a novel multi-input, multi-output resource controller. The complex relationship between application performance and resource allocation are captured at the time of allocation of right amount of resources to achieve application SLO by MIMO. Virtualization is an emerging technology in enterprise data centres which gives rise to a new paradigm: shared virtualized infrastructure. In this virtualized infrastructure allocated resources are shared dynamically In this new paradigm, multiple enterprise applications share dynamically allocated resources. And also it is used to consolidate application where the infrastructure and operational system costs are reduced with the increase of resource allocation. The main challenge faced by data center administrators is to satisfy the service level agreements. But it will be complex where dynamic resource sharing and unpredictable interactions are occurred across many applications. The challenges faced by database administrator are:

Complex SLO's: The conversion of individual application SLO's into equivalent resource shares are non- trivial in the shared virtualized platform.

The variation of resource requirement and enterprise application workloads based on intensity will lead to a demand for individual resource type's changes over the lifetime of the application. This utilization variation over time for both resources will occur at different times of a day. This variation proves that static resource allocation can meet application SLO's only when the resources are allocated for peak demands, wasting resources.

Distributed resource allocation: In Multi-tier applications across multiple nodes, resource allocation can be done at all tiers to satisfy and meet end to end application SLO's.

Resource dependencies: The ability of application decides application level performance across multiple system resources. In a virtualized infrastructure, each and every application depends on one another based on its performance which makes it difficult to replicate its behaviour in pre-production environment. Application level SLO is used to address the problem of managing the allocation of computational resources in a shared, virtualized infrastructure.

The problem of allocation of resources is controlled by using Auto Control method. It is an automated resource control and adaptation system. Two main contributions of this method are: First, an online model estimator is designed to determine and capture the affiliation between application level performance and the allocation of individual resource shares dynamically. The complex behaviour of enterprise application (varying resource demands overtime, resource demands from distributed application components, and shifting demands across multiple resources types) are captured by using adaptive modelling approach. Second, two layered, multi-input, multi output controller is designed to schedules multiple types of resources to multiple enterprise applications automatically to achieve their SLO's. The first layer consists of a set of application controllers. The role of application controller is to determine the amount of resources required in automatic manner in order to achieve individual application SLO's, with the usage of estimated models and a feedback approach. The second layer is comprised of a set of node controllers. The role node controller is to detect resource bottlenecks on the shared nodes and properly allocate resources of multiple types to individual applications. At the time of overload, the node controllers will provide differentiation among the services by prioritization among different applications. Auto Control is used to detect and adapt to bottlenecks which is happened at both CPU and disk across multiple nodes. The architecture of Auto Control admits the placement of App Controllers and Node Controllers in a distributed manner. Node Controllers can be hosted in the physical node they are controlling. App Controllers can be hosted in a node where one of the application tiers is located. Here do not order this

placement, how-ever, and the data center operator can choose to host a set of controllers in a node dedicated for control operations.

A model estimator is the one which will learn automatically in real time. It is a model for the affiliation between an application's resource allocation and its performance. An optimizer will predict the resource allocation required for the application to meet its performance target based on estimated model.

One of the main goals of this optimizer is to determine the source allocation required in order for the application to meet its target performance. Other goal is to complete this in a secure manner, without affecting many oscillations in the resource allocation.

Qi Zhang *et al* [7] as a demand of each VM type can fluctuate independently at run time; it becomes a problem to dynamically allocate data center resources to each spot market to maximize cloud provider's total revenue.

It presents a solution to this problem that consists of 2 parts:

1. Market analysis for forecasting the demand for each spot market
2. A dynamic scheduling and consolidation mechanism that allocate resource to each spot market to maximize total revenue.
3. Cloud providers specify a fixed price for each type of VM offerings.
4. When total demand is much lower than data center capacity, the data center becomes under-utilized, i.e., the cloud provider is to encourage customers to submit more requests.
5. When total demand rises over the data center capacity, it is desirable for the cloud provider to motivate the customers to reduce their demand.

A promising solution to this problem is to use market economy to reshape the demand by dynamically adjusting the price of each VM type.

1. When total demand is high, the mechanism raises the price to ensure resources are allocated to users who value them the most.
2. When total demand is low, the mechanism lowers the prices and provides incentive for customers to increase their demand.

Dynamic resource allocation framework consists of the following components:

Market Analyser:

1. Analyse the market situation and forecast the future demand and supply level
2. Predict the future demands use AR(auto regressive model)

Capacity planner:

Capacity planner decides the expected price of each VM Different pricing schemes:

1. In the fixed pricing scheme, price of a VM type does not vary with the current supply and demand.
2. In the uniform pricing scheme, the price of a VM type is adjustable at run-time.

VM scheduler:

1. Make online scheduling decision for revenue maximization
2. Dynamic resource allocation policy has the multiple machine configurations and this will amplified when the demand pattern changes over the time.

Gunho Leey *et al* [8] in cloud computing environment data analytics are the key applications and it should be account for the heterogeneity of environments and workloads. In cloud computing environment does not provide the fairness among jobs when multiple jobs share the cluster. In Proposed, Resource allocation on a data analytics system in the cloud to hold the heterogeneity of the underlying platforms and workloads and the architecture shows how to allocate resources to a data analytics cluster in the cloud.

Technique:

Data analytic cluster: Data analytics workloads have heterogeneous resource demands because some workloads may be CPU whereas others are I/O-intensive. In cloud various resource demands of data analytics workloads, we scale the cluster according to demands.

For scaling process the resource allocation strategy having two levels

1. Divides machines into two pools - core nodes and accelerator nodes
2. Dynamically adjusts the size of each pool to reduce cost or improve utilization.

Data analytic cloud contains the components are,

Core nodes: Host the data and computations
Accelerator nodes: This is added to the cluster temporarily when additional computing power needed.
Analytic engine: Runs the application both the pools

Cloud driver:

1. This manages the nodes allocated to the analytic cloud and decides when to add/remove what type of nodes to/from which pool.
2. The user submits the job to the cloud driver with the hints about the job. Cloud driver keeps the history of routinely processed query, this is used to estimate the submission query and update the hints provided.
3. This is also monitor the storage system to estimate the incoming data rate. This will predict the resource requirements to process queries and to store data.
4. Many productions query are submitted with tight deadlines, the cloud driver will add the nodes to the accelerator pool temporarily to handle the job rather than allocating more core nodes.
5. When adding nodes, the cloud driver makes the decision on which resource container to use.

***Ying Song** et al.* [9] propose a two-tiered on-demand resource allocation mechanism, including the local and global resource allocation, based on a two-level control model. A waste of resources can be minimized and quality of the application can be guaranteed by using well designed on demand resource allocation algorithm. The optimization resource allocation can be achieved in VM by using local on demand resource allocation algorithm with allocation threshold value. And in global on demand resource allocation method resource allocation can be optimized by adjusting the allocation threshold of each local resource allocation.

In this work a novel two-tiered on-demand resource allocation mechanism with feedback is proposed to optimize the resource allocation for VM-based data center. Optimized resource allocation method is modelled to provide the guidance for the design of on-demand resource allocation algorithm. The local and global resource allocation algorithms are used to optimize the dynamic resource provision based on two-tired on demand resource allocation.

The memory load in a VM can be finding out by using local and lazy on demand memory allocation algorithm which is based on the static priority [12]. The activity refers to the threshold of idle memory for memory overload. If idle memory of each VM is higher than, no memory needs to be reallocated. If there is any additional VM present, then the MwmFlow-L will increase the memory size in the case less IMi.

The single point failure of the global scheduler can be addressed by selecting a server dynamically. The replacement server will be selected randomly in the case of any server failure. Even if there is no replacement global scheduler for the failed global scheduler, the local schedulers will not stop working. They will continue their work to allocate resource to the hosted VM without macro level optimization. Thus it proves that failure of global scheduler could not affect the other resource allocation job. The computation scale of the global scheduler is directly proportional to the number of application in the K-VM-1-PMmodel, because the constrained optimization in the K-VM-1-PM model is linear. Thus, the global scheduler is not the bottleneck even in a large-scale computing environment.

Zhen Xiao *et al* [10] Cloud computing allows business customers to scale up and down their resource usage based on needs. In this paper, using virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use

Technique:

1. Virtualization technology
2. Skewness

Goals:

1. *Overload avoidance:* The capacity of a PM should be sufficient to satisfy the resource needs of all VM's running on it.
2. *Green computing:* The number of PM's used should be minimized as long as they can still satisfy the needs of all VM's. Idle PM's can be turned off to save energy.

Virtualization technology: This technology used to allocate datacenter resources based on the application demands.

Skewness: This is used to measure the unevenness multidimensional resource utilization of a server. By combining different types of workloads skewness can be minimized. Skewness can be measured based on,

1. *Hot spot:* If the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VM's running on it should be migrated away.
2. Cold spot: If the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy.

Achieve the goals to make the following contributions,

1. Develop a resource allocation system that can avoid overload in the system

2. Skewness to measure the uneven utilization of a server.
3. Design a load prediction algorithm that can capture the future resource usages of applications accurately without looking inside the VM's.

Load prediction algorithm:

Exponentially weighted moving average (EWMA): Predict the CPU load and we measure the load every minute and predict the load in the next minute.

TABLE I
ANALYSIS OF METHODS

S.No	TITLE	AUTHORS	METHODS	ADVANTAGES	DISADVANTAGES
1	Live Migration of Virtual machines	Christopher Clark et. al[2]	Live migration: Migrating application into another system.	1. It is extremely powerful tool for clusters administrators. 2. Relieve the load on the congest hosts.	1. Consumes entire bandwidth when sending VM images iteratively 2. It only considers the live migration among the well-connected data center.
2	Usher: An Extensive Framework for Managing Clusters of Virtual Machines	Marvin McNett et. al [3]	Usher Framework: Plugin API is for adding modules.	1. Provide a best effort computing environment. 2. Performs load balancing across physical machines. 3. Usher can be used for controlling the placement scheduling, and migration of VM's if desired.	1. For using other sites in usher, existing plug-in are not matching for this. 2. For using the usher in another site needs to modify the existing plug-in or rewrite it. 3. No plug-in for managing clusters of physical machines is written.
3	Energy-Aware Server Provisioning and Load Dispatching for Connection- Intensive Internet Services	Gong Chen, et. al [4]	Skewness algorithm	1. Load prediction will help to reduce the frequently turning on and off servers. 2. Load prediction will reduce the power consumption. 3. Load balancing is considered.	1. Sometimes load prediction may cause the failure. Migration of the load is not considered. 2. Power is only considered as the parameter.
4	Automated Control of Multiple Virtualized Resources	Pradeep Padala, et. al [5]	Auto Control: An automatic control system	1. Performance assurance: All Applications can be meet their performance. 2. Without human intervention allocation decision should be made automatically. 3. Various workloads can be adopted. 4. Scalability can be achieved.	1. Auto Control only does not deal the bottleneck problems. 2. It does not control any memory control
5	Dynamic Resource Allocation for Spot Market in Cloud	Qi Zhang et. al [7]	Local Search Approximation Algorithm	Total revenue is maximized	VM pre-emption and migration are not addressed
6	Heterogeneity Aware Resource Allocation In Cloud	Gunho Leey, et. al [8]	Resource Allocation and Scheduling	Provide the fairness among jobs when multiple jobs are submitted	1. Migration of the load is not considered. 2. Power is not considered as the parameter.
7	A Two Tired On-Demand Resource Allocation Mechanism for VM Based Data Center [9]	Ying Song, et. al [9]	Two Tiered Allocation Mechanism 1)Local resource Scheduler	1. It addresses the problems of availability and scalability. 2. If global resource allocation failure occurs then the local resource	1. Application workload scheduling is not considered. 2. Mismatch between the on demand resource and workload dispatch.

			2)Global resource scheduler	allocation will work, vice versa. 3. No failure of resource allocation is occurred.	
8	Dynamic Resource Allocation Using Virtual Machine for Cloud Computing Environment	Zhen Xiao, et. al [10]	1. load prediction algorithm 2. Skewness Algorithm	Server overload is minimized	Future load prediction algorithm does not always give the actual result.

III. CONCLUSION

This paper deals with the theoretic study of different dynamic resource allocation techniques in cloud environment. The detail explanation of the techniques is briefed and also summarizes the advantages with parameters of the different techniques in cloud computing environment. Many of the gains in the cloud model come from resource multiplexing through virtualization technology. In this survey conclude efficient algorithm as a system that uses virtualization technology to allocate data center resources dynamically based on application needs and support green computing by optimizing the number of servers in use. "skewness" algorithm is used to measure the un-evenness in the multidimensional resource utilization of a server. By minimized skewness, different workloads can be combined and improve the over-all utilization of server resources. We discuss a set of heuristics that avoid overload in the system efficiently while saving energy used.

REFERENCES

[1]. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization", Proceedings of the nineteenth ACM symposium on Operating systems principles, New York, USA, Oct. 2003.

[2]. Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Prattand and Andrew Warfield, "Live Migration of Virtual Machines", Proceedings of the 2nd conference on Networked Systems Design & Implementation, UK - Volume 2, May. 2005.

[3]. Marvin McNett, Diwaker Gupta, Amin Vahdat, and Geoffrey M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Machines", Proceedings of the 21st conference on Large Installation System Administration Conference, USENIX Association Berkeley, USA, Nov. 2007.

[4]. Gong Chen, Wenbo He, Jie Liu, SumanNath, Leonidas Rigas, Lin Xiao, and Feng Zhao "Energy-aware server provisioning and load dispatching for connection-intensive Internet services", Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, Berkeley, USA, Apr. 2008.

[5]. Pradeep Padala, Kai-Yuan Hou Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, and Arif Merchant, "Automated Control of Multiple Virtualized Resources", Proceedings of the 4th ACM European conference on Computer systems, New York, USA, Apr. 2009.

[6]. M. Mishra and A. Sahoo, "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach", Proceedings of the IEEE 4th International Conference on Cloud Computing, Washington, USA, Jul. 2011.

[7]. Qi Zhang, Eren Gurses, Raouf Boutaba and Jin Xiao, "Dynamic Resource Allocation for Spot Markets in Clouds", Proceedings of the 11th USENIX conference on hot topics in management of internet, cloud, and enterprise networks and services, USA, Dec. 2011.

[8]. Gunho Leey, Byung-Gon Chunz, and Randy H. Katzy, "Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud", Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, USENIX Association Berkeley, USA, 2011.

[9]. Ying Song, Yuzhong Sun and Weisong Shi, "A Two-Tiered On-Demand Resource Allocation Mechanism for VM Based Data Centres", IEEE transactions on services computing, ISSN: 1939-1374, vol. 6, no. 1, Jan. 2013.

[10]. Zhen Xiao, weijia song and Qi chen "Dynamic Resource allocation using Virtual Machines For Cloud Computing Environment", IEEE Transactions on parallel and distributed systems, ISSN: 1045-9219, vol.24, No.6 Jun. 2013.

