AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Algorithms for Classification of VM migration scheme Inner Cloud Data Centre

[1]M. Padma and [2]G. Geetharamani

[1]*Research Scholar, Department of Computer Science and Engg, BIT Campus, Anna University::Chennai, Thiruchirapalli, Tamil Nadu, India.*
[2]*Assistant Professor, Department of Mathematics, BIT Campus, Anna University::Chennai, Thiruchirapalli, Tamil Nadu, India.*

ARTICLE INFO

Article history:
Received 19 September 2014
Received in revised form
19 November 2014
Accepted 22 December 2014
Available online 2 January 2015

Keywords:
Cloud computing, IaaS data centre, resource migration algorithm

ABSTRACT

Cloud computing has become increasingly popular among the widely deployment of multiple cloud infrastructures. Infrastructure-as-a-service (IaaS) cloud computing replaces bare hardware. The user will use cloud virtual machines (VM) to fulfill your computing needs. In this paper, we investigate how to reduce fragments of resources and the resources allocated in data centers. Therefore, a new scheme for estimating the VM migration to effectively reduce resources fragments the resources is proposed. The scheme improves the utilization of servers in data centers. Moreover, our proposed scheme can be applied to multiple resources. Among software components in the IaaS cloud stack, the module resource migration is very important as selecting appropriate virtual machine and place to run virtual machines. This paper focuses on the study and classification algorithms used in module resource migration. Questions of how to apply these algorithms are also discussed.

To Cite This Article: M. Padma and G. Geetharamani., Algorithms for Classification of VM migration scheme Inner Cloud Data Centre. ***Aust. J. Basic & Appl. Sci.,*** *9(1): 91-105, 2015*

INTRODUCTION

Cloud computing has become more and more popular with the widely deployment of several cloud infrastructures (Rimal, B.P., *et al*., 2009). The core principle of cloud computing is delivering services from shared hardware. The goal of this computing model is to make a better use of distributed resources, put them together to make higher throughput and be able to handle large-scale computation problem. People often categorize Cloud computing into three levels of use model or cloud computing services as presented in Fig. 1.

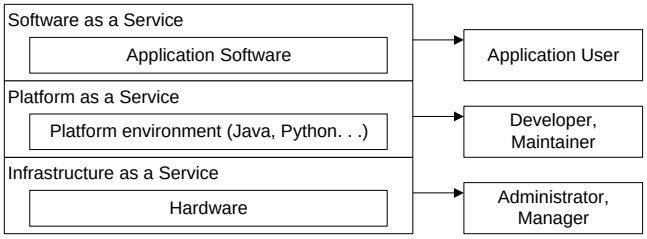

Fig. 1: General Cloud categorization.

Infrastructure-as-a-service (IaaS): Cloud computing replaces bare computer hardware. Users of IaaS have the ability to support operating systems and applications, but don't wish to buy server, storage and networking hardware and a data centre to house the hardware. Examples of those providers are companies such as Amazon (http://aws.amazon.com/ec2/), ENKI (http://www.enki.co/), GoGrid (http://www.gogrid.com/).

Platform-as-a-service (PaaS): Cloud computing replaces an execution environment for a computer language by providing a system ready to execute the user's software. The user of PaaS is the programmer. Examples of those providers are companies such as Engine Yard (http://www.engineyard.com/products/cloud) or Google (http://www.google.com/apps/intl/en/business/cloud.html).

Software-as-a-Service (SaaS): The user interacts directly with the Cloud-hosted software, and often pays for "seats" or "users" instead of computer time. Examples of those providers are NetSuite

Corresponding Author: M. Padma, Research Scholar, Department of Computer Science and Engg, BIT Campus, Anna University::Chennai, Thiruchirapalli, Tamil Nadu, India.
E-mail: padmamayan@gmail.com

(http://www.netsuite.com/portal/home.shtml), SalesForce.com (http://www.salesforce.com/ap/?ir=1), Google Apps (https://developers.google.com/appengine/).

Within the scope of this paper, we focus on IaaS Cloud. Figure 2 presents the typical architecture of an IaaS cloud. An IaaS cloud has many computing nodes grouped together to form clusters. For each node, there is virtualization component. It is a special purpose operating system that creates and maintains the VMs as well as serves their requests for accessing to hardware resources.

An NC (Node Controller) executes on every node that host VM instances. An NC makes queries to discover the node's physical resources – the number of cores, the size of memory, the available disk space - as well as to learn about the state of VM instances on the node. The information collected is propagated up to the Cluster Controller.

The Cluster Controller (CC) generally executes on a cluster front-end machine. CC has three primary functions: issue running instances to specific NCs, control the instance virtual network overlay, and gather/report information about a set of NCs.

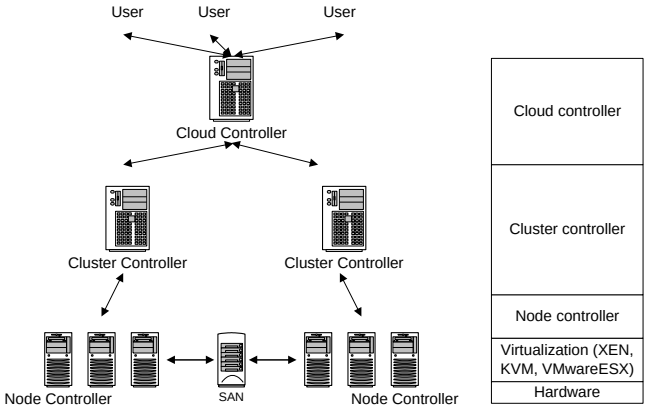

Fig. 2: Typical IaaS cloud architecture.

Cloud Controller is the entry-point into the cloud for users and administrators. It queries node managers for information about resources, makes resource migration decisions, and implements them by making requests to cluster controllers.

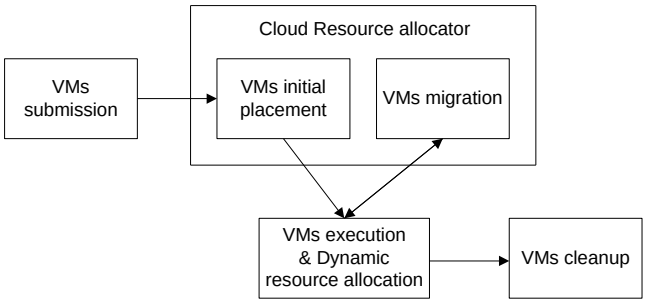

Fig. 3: Role of resource migration in VM life cycle.

Among the components of IaaS cloud software stack, the resource migration module is very important as it assigns resources to VMs. Figure 3 presents the role of resource migration in VMs' life cycle. When a user submits VMs to the IaaS cloud system, the cloud resource migration module will find the acceptable VMs and find the initial places to run those VMs. During the VMs execution process, the cloud system may migrate VMs out of the initial place to other computing nodes. The cloud resource migration module decides which nodes to migrate VMs to. While the node is executing VMs, the OS of the node may perform coarse-grained dynamic resource migration to VMs (Waldspurger, C.A., 2002).

In this paper, we want to concentrate on the natural distributed character of the IaaS cloud, but not on the OS within a physical node. Thus, this paper focuses on studying and classifying algorithms used in the cloud resource migration module to decide which VMs to run and define the computing nodes to run those VMs. The rest of the paper is organized as follows. Section 2 summarizes related survey works. Section 3 presents taxonomies for resource migration algorithms. Section 4 surveys various algorithms that have used for cloud resource migration. Finally, we end this paper with discussion on open issues, lesson learned in section 5 and conclusion in Section 6.

II. Related works:

According to our knowledge, we have not noticed any comprehensive classification journal article on IaaS cloud resource migration approaches. However, a number of related surveys and review book chapters that referred to IaaS cloud resource migration have been published. In this section, we describe only those surveys and reviews. The detail description of referred algorithms and their original reference are in section IV.

In (Endo, P.T., *et al*., 2010), the authors studied open source cloud platforms. This work compared solutions and their business model (hardware, middleware and user level) according to configuration flexibility. It also compared the service, infrastructure and users of those systems. The important of cloud resource migration was stated, but none of detail issues were discussed.

The work in (Rimal, B.P., 2009) extended a classification and survey of cloud computing system to both open source and commercial cloud platforms. The cloud systems are mainly characterized with architecture, virtualization management, service, fault tolerance and security. Related to resource migration , the authors referred only the load balance feature. In which, most studied systems use simple algorithms such as Round Robin, Greedy or server load equalization at IaaS level.

In (Beloglazov, A., *et al*., 2011), the author presented the classification and survey of energy-efficient data centres and cloud computing systems. This work discussed many energy-saving techniques ranging from hardware level, OS level, Virtualization level to data centre level. At data centre level, the authors described several research works about saving energy techniques. Those techniques mostly are based on DVFS (Dynamic Voltage and Frequency Scaling), VM consolidation and power switching.

In (Teng, F., 2012), the authors discussed various scheduling techniques for traditional distributed systems, grid computing systems and cloud computing systems. However, the author only touched the surface of real works for cloud computing. The IaaS cloud employs the VM concept. Each VM could have multiple CPUs and must be allocated completely within a physical machine. From the point of resource migration , this is the distinguished character of Cloud IaaS compared with traditional distributed system and Grid computing. In other types of distributed system, one job including many processes can be spread out to multiple physical machines.

The work in (Silpa, C.S., S.S.M. Basha, 2013) took some scheduling algorithms for cloud computing and performed experiment to do comparative analysis. The main comparative criteria include execution time, resource use rate and cost of algorithm. Also following this way, the work in (Do, T.V., C. Rotter, 2012) focused on scheduling schemes for on-demand IaaS requests. However, the authors of (Do, T.V., C. Rotter, 2012) used the analytical model and studied the ability of reducing energy consumption.

III. Resource migration algorithms classification:

A***. Overview of the cloud resource migration problem:***

The resource migration algorithm responds for finding the resource migration solution that satisfies a specific goal of the cloud provider. This goal could be optimizing power consumption, optimizing cost, ensuring SLA, etc. The typical resource migration architecture for IaaS cloud is presented in Figure 4.

In general, the input for resource migration includes resource information and workload information. Based on this input information, the resource migration algorithm finds out resource migration solution.

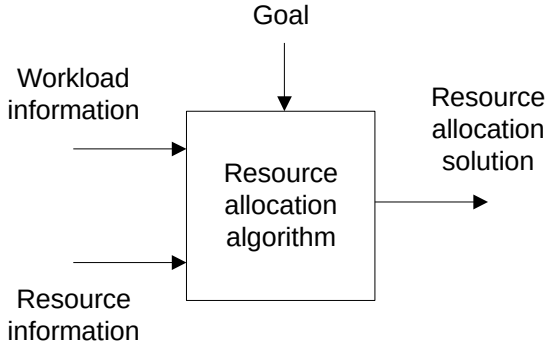

Fig. 4: Typical resource migration architecture.

B. Classification of resource migration algorithms:

From our point of view, we classify IaaS's resource migration algorithms according to three main criteria: execution phase of VMs, business model of providers and goal of the algorithm. We build the hierarchy of those algorithms as presented in Fig. 5. In this section, we discuss in detail each criterion.

1) Execution phase of VMs:

This classification is based on VM lifecycle operations of the cloud system. In the beginning when it comes to a VM system, the system will make the initial placement. If necessary, VM during the execution, the system

Australian Journal of Basic and Applied Sciences, 9(1) January 2015, Pages: 91-105

will make VM migration. At each stage of the execution of each task and workload parameters variations dialectical resource migration process is different.

Initial placement - When the virtual to cloud users, the initial placement algorithms executed. Where their work will determine if it can be allowed to run the VM. Methods, including static and dynamic information resource information required. Static information dynamic information such as CPU load, memory load, network load as the foundation of the present application, such as calculating the number of nodes, CPU's number, the number of cores memory capacity, storage capacity, such as the computing node has information, etc. At this point, the user option must be respected. In this case, the source of their required workload is a group of VM. This is usually the number of vCPUs storage and bandwidth, is the amount of memory. In some cases, this information includes the user's bid price.

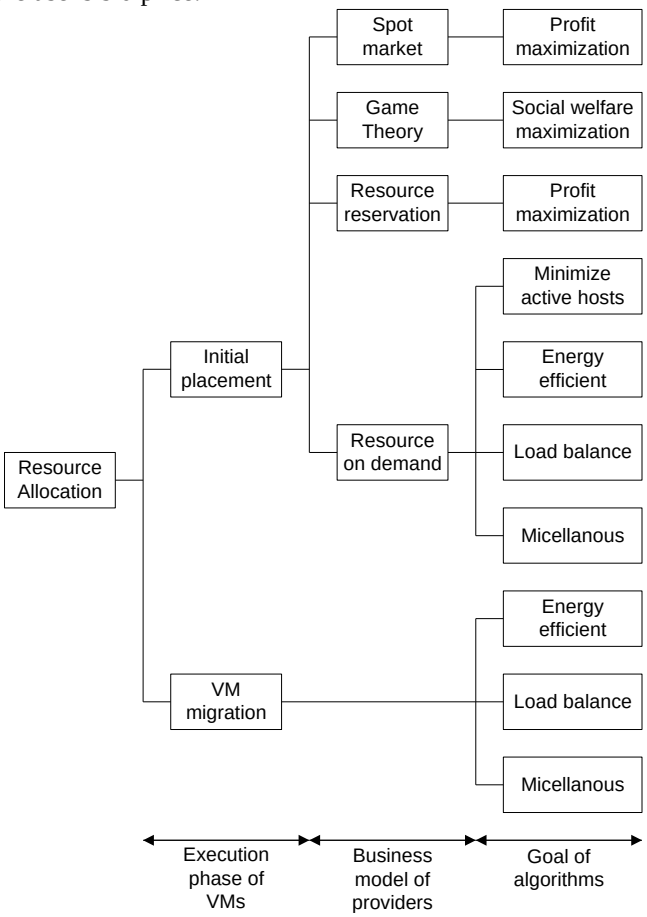

Fig. 5: Classification of resource migration algorithms.

VM migration - During the operation process of the virtual data center migration pathways induced by an internal data center policy. Unlike the initial placement algorithms, and not worry about discarding some virtual migration algorithms. Their goal is to complete the data center running VM's trying to find a new place. To perform this task, resource information is the same as in the previous case. However, current information on the use of their resources, in this case the workload running in the data center all the VM's. Storage usage and bandwidth usage, CPU usage, it is usually the ratio, the amount of memory usage, amounts. Business model of providers

IaaS cloud providers and cloud users to distribute this classification is based on the resource. IaaS cloud providers in the sample space of the market model, the resource allocation model, game theory model or resource can distribute the resource. Different from each other based on the difference in the business model of the dialectic as the source of migration algorithms.

Spot market in the spot market model, the user if he / she is willing to pay per hour resource, is to claim the maximum price. The current spot price is greater than the maximum bid price for the user, the user can run within a virtual resource received. Then, the next time the user to gain the right to reclaim the resource is used. There are two separate steps, it should be noted that the market mechanism is the migration of the resource. The first step in its interaction with the system is allowed to bid price, which will determine the VM. This is the second step VM physical machine (PM) will determine admitted to running.

Resource reservation - Resource allocation model, the user can choose to run a virtual instance of the use and the virtual. He / she is still, while requiring the user if he / she is able to run multiple instances of the

booking reserved cases, offer a capacity reservation. Amazon EC2 as a long-term resource allocation, five, Reserved Instances and the user if he / she wants to record every instance have a lower willingness to pay a onetime fee. In turn, he / she receive a significant discount, for example, hourly rates. Elastic cloud environments, the provider, make sure they have enough resources to meet demand. Otherwise, the provider does not have the death penalty and its Reservation VM instances when they wanted to pay compensation to those clients.

Game theory - In game theory model, the problem is considered by many customers at the same time using the Cloud. Each round of scheduling, the system considers all workloads and new workloads. The first scheduled workloads should be prioritized. Then, the new demands of users select the right resources available directly from the pool. The selection process takes place through several rounds until it reaches equilibrium. The balance of the customers and their ability to optimize their resource usage and cloud resources are used near the (near) optimal prices that are charged.

Resource on demand – This is the current most popular resource distribution model for IaaS cloud. In the resource on demand model, when users need resource to run their VMs, they can get them from the cloud. They can use them as long as they wish. The price of using resource for each VM instance is usually fixed.

2) Goal of the algorithm:

This classification is based on the expectancy of the cloud providers with their resource usage. Some main focuses are energy-efficient, profit maximization, load balance or number of active PMs minimization. Besides that, some other goals include fault tolerance, ensuring SLA, etc. It is clear that different goals lead to different resource migration algorithms.

Profit maximization - Profit maximization is an important goal for commercial cloud providers. There are two ways to gain profit maximization. If the cloud providers have many prices for a single resource, in the spot market for example, selling the same amount of resource with highest price will reach profit maximization. If the cloud providers have fixed price for a single resource, maximizing the workload running by the same amount of resource also ensures profit maximization.

Social welfare maximization – Social welfare maximization is a goal of economic that tried to apply to cloud environment. The mechanisms manage to distribute resources in a way that reaches equilibrium. This equilibrium ensures that clients are charged (near) optimal *prices* for their resource usage and resources of the cloud are used near their optimal capacity.

Energy efficient - Saving money in the energy budget of a data centre, without sacrificing SLAs is an excellent incentive for data centre owners. It would at the same time be a great success for environmental sustainability. Usually the power consumption of a cloud data centre is calculated through the power consumption of the IT equipment and the PUE (Power usage effectiveness) value of the data centre.

$$Pcenter = Pit\_equipments * PUEcenter \quad (1)$$

With a data centre, its PUE is relatively stable. Thus, efforts to cut energy consumption of a data centre focuses on reducing the energy consumption of servers. The power consumption model of a server is further divided to power consumption of CPU, disk, memory, etc. Detail server power consumption model can be seen in [13]. Allocating the same amount of workload using minimal amount of energy will assure energy-efficient.

Number of active PMs Minimization – Online bin-packing problem derived from the target, the target. The number of bins used in different modules in a way that reduces the ability of HIV to a method defined in a bundle. IaaS cloud environment, the resource used to block migration, PMS PMS in a way that reduces the number of a defined need to assign a set of virtual.

Load balance - Load balance is one of the main challenges in cloud computing. It requires distributing the dynamic workload across multiple nodes to make sure that no single node is overwhelmed. It helps in optimal use of resources and hence in enhancing the performance of the system. For the IaaS cloud, the load of each host is calculated with:

$$Load = \frac{\sum VM\_entitlement}{Host\_capacity} \quad (2)$$

Usually, the VM entitlement is the occupied CPU resource or memory resource of the VM running on the host. The host capacity is the available resource of the host, which can be provided to VMs.

Miscellaneous – Besides main goals as stated above, the literature also recorded some research works about IaaS cloud resource migration with other goals such as ensuring SLA, fault tolerance, multi objectives, etc.

IV. Survey of resource migration algorithms:

VMs initial placement algorithms:

3) Profit optimization algorithm for clouds support resource reservation:

The work in (Sotomayor, B., *et al*., 2008) proposed a mechanism to allocate computing resource to workloads with various mixes of best-effort and advance reservation requests. Incoming advance reservation leases are always scheduled right away, where best-effort leases are put on a queue. The scheduling function

periodically evaluates the queue. It uses an aggressive backfilling algorithm (Jalaparti, V., *et al*., 2001; http://opennebula.org/) to decide whether any best-effort leases can be scheduled. When an advance reservation lease comes, the scheduler will try to choose nodes that will not need preempting another lease.

In (Wang, X., 2011), the authors presented a scenario of cloud platforms that supports resident applications and resource reservation. If some resource reservation requests come, VMs running resident applications can easily be shifted somewhere (Zhao, M. and R.J. Figueiredo, 2007) to squeeze out enough resources for the reservation. However, doing this might lead to negative impact to the applications originally resident due to limited total resource amount. The authors defined a revenue function and the adaptive QoS-aware resource reservation management algorithm. With each physical node, the algorithm checks the capacity, forms a new configuration according to the reservation and forecasts the revenue. It then select the best node giving the largest revenue.

From the description above, we can see that both algorithms in (Sotomayor, B., *et al*., 2008) and (Wang, X., 2011) support reservation as well as non-reservation workloads. They both give higher priority to reservation request. When a new reservation request comes, the algorithm in (Sotomayor, B., *et al*., 2008) and (Wang, X., 2011) may suspend or migrate non-reservation VMs to have enough cloud resources for this request. The difference between two algorithms is the behavior with the affected VMs. While the algorithm in (Sotomayor, B., *et al*., 2008) does not care much about the affected non-reservation workloads, the algorithm in (Wang, X., 2011) has to consider the fine before deciding to migrate the non-reservation VMs.

4) Profit optimization algorithms for clouds support spot market:

Motivated by low resource use, Amazon EC2 introduced the spot instance mechanism to allow customers to bid for unused Amazon EC2 capacity (http://aws.amazon.com/ec2/). Amazon EC2 at each availability zone has fixed number of virtual machine types and fixed number of instances of each VM type. Amazon EC2 runs one spot market for each VM type in each availability zone. A customer submits a request that specifies the type, the number of instances, the region desired and the biding price per instance-hour. The provider assigns resources to bidders in decreasing order of their bids until all available resources have been allocated or all resource requests have been satisfied. The selling price (i.e. the spot price) is equal to the lowest winning bid. The actual VM – physical host assignment is done with Round Robin algorithm (Rimal, B.P., *et al*., 2009), which is described in detail in Section 4.1.4.

Unlike Amazone EC2, in (Zaman, S. and D. Grosu, 2010), the authors developed the model to allow users to bid for bundles of items. This model is called combinatorial auction. Determine the set of winning users with the goal of maximizing the income in this model is a NP-hard problem (Rimal, B.P., *et al*., 2009). To solve this problem, the authors proposed the CA-GREEDY algorithm. It collects users bundle of VMs and bid price, does weighted sum of VM of each bid, calculates the bid density, sorts bids by density and then allocates highest to lowest, until resources exhaust.

The above works all assume fixed number of virtual machine types and fixed number of instances of each VM type. In (Zaman, S. and D. Grosu, 2011), the authors proposed a model called Dynamic VM Provisioning and Migration (DVMPA). It allows users to bid for a bundle of VMs of different types. It can configure the set of available computing resource into different numbers and types of VM instances. The author proposed the CA-PROVISION algorithm to solve the problem. It first collects users' bundle of VMs of different types and bid price, does weighted sum of VM of each bid, calculate the bid density, sort bids by density and then allocates highest to lowest, until resources exhaust. Once the winners are determined, the mechanism determines the VM configuration to fulfill the winning users' requests.

From the description above, we can see the increasing complexity of the workload and resource assumption along the line of description. The algorithm in (http://aws.amazon.com/ec2/) and (Zaman, S. and D. Grosu, 2010) assume single VM and bundle VMs bid with fixed amount of available resources. The algorithm in (Zaman, S. and D. Grosu, 2011) assumes bundle VMs bid and a pool of resources. With the increasing complexity in input description, the algorithm is more complex and more efficient.

5) Social welfare maximization algorithms for clouds support game theory:

In the work of (Jalaparti, V., *et al*., 2010), the authors defined a new class of games called Cloud Resource Migration Games (CRAGs). CRAGs solve the resource migration problem in clouds using game-theoretic mechanism. At the start of each round, all the clients submit the jobs for that round to the cloud. The provider first calculates his resource migration vector for jobs that are running on the cloud. The provider then advertises the amount of resources left at each machine to the clients. Each client chooses the machines that would satisfy its resource need and minimize its total cost. Next, the clients will actively change their resource migration as long as they can decrease their costs. They can follow any of the standard update mechanisms in the literature (Nisan, N., *et al*., 2007) to determine how they change their strategy. All the clients must follow the same update mechanism. When no client can decrease its cost by changing its strategy alone, the system has achieved a stable equilibrium.

Australian Journal of Basic and Applied Sciences, 9(1) January 2015, Pages: 91-105

Also applying game theory to resource migration, however, instead of letting user take part in many migration rounds, the work in (Wei, G., *et al*., 2010) proposed a practical approximated solution with the following two steps. First, each participant solves its optimal problem independently. A Binary Integer Programming method is proposed to solve the independent optimization. Second, an evolutionary mechanism is designed to change multiplexed strategies of all initial optimal solutions with the goal of minimizing their efficiency losses. The algorithms in the evolutionary mechanism take both optimization and fairness into account.

From the description above, we can see that the algorithm in (Jalaparti, V., *et al*., 2010) is not as convenient as the algorithm in (Wei, G., *et al*., 2010). The requirement of complex users' attendance in the scheduling process makes the algorithm in (Jalaparti, V., *et al*., 2010) impractical. It should be used only as the source for further refinement. The algorithm in (Wei, G., *et al*., 2010) fixed the drawback of the algorithm in (Jalaparti, V., *et al*., 2010) but its execution time could be long with the evolution mechanism.

6) Load balance algorithms for clouds support resource on demand:

Several commercial and open source cloud systems use simple load balance algorithms like Round robin (http://aws.amazon.com/ec2/; http://www.gogrid.com/; http://opennebula.org/; http://www.enomaly.com/) random (http://www.enomaly.com/), least connect (http://www.gogrid.com/), weighted selection (http://opennebula.org/; Chandrasekaran, B., *et al*., 2007) as described in (Rimal, B.P., *et al*., 2009).

Round robin is among the most widely used migration algorithms for VM initial placement. The servers are in a circular list. There is a pointer pointing to the last server that received the previous request. The system sends a new resource migration to the next node with enough physical resources to handle the request.

Random load balancing, requests are routed to servers at random. Random load balancing for each server instance where a similarly configured system running on the same cluster deployments, is recommended. The cumulative number of requests to the server instances in the cluster increases the random load balancing distributes requests evenly throughout. Over a small number of requests, the load can be balanced correctly.

During the study, the methods are simple. Setting a new link to the node that has the least number of current connections passing. Servers or other equipment with similar capabilities, not least in the context of the methods work best. With the dynamic load balancing methods. Their current number of connections per node, such as a node or a fast response time, server performance in terms of various aspects distribute connections.

In many corporate environments, unequal power and performance characteristics of the server nodes are used to provide services. It would not be burdened with unreasonable demands, some servers to distribute the load on the server capabilities to individual cases.

Weighted Round Robin blank round robin algorithm that eliminates the drawbacks of an improved version of the round robin. Weighted round robin algorithm to assign a weight to each server in a group. If a server is capable of handling the load as the other two times more powerful server receives a weight of 2. In such cases, the scheduler is assigned to every demand weakened a powerful server will assign two requests.

Such as weighted round robin, weighted random load balancing policy with respect to each other server systems and allow you to specify a processing load distribution ratio. In addition to weight, making the final selection based on the weight of the purified using a random distribution. Higher probability of receiving the request to the server is overweight.

The work in (Hu, J., *et al*., 2010) presented a scheduling strategy on load balance of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence on the system after the deployment of the needed VM resources. It then chooses the least-affective solution, through which it achieves the best load balance and reduces or avoids dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling.

The work in (Randles, M., *et al*., 2007) discussed resource migration methods for large-scale Cloud Systems. The first method is Biased Random Sampling. In the large-scale Cloud system, each computing node in the cloud has a set of neighbor nodes. When a request comes, the node will choose randomly a neighbor node according to the light loaded level. The walk like that will continue k steps and the lightest load nodes will be selected for migration.

The other method discussed in (Randles, M., *et al*., 2007) is Active Clustering. Active Clustering works on the principle of grouping similar nodes together and working on these groups. Each computing node has a set of neighbor nodes. When a request comes, the initial node selects another node called the matchmaker node from its neighbors. This selection satisfies the criteria that it should be of a different type than the former one. The so-called matchmaker node then forms a connection between neighbors of it that is of the same type as the initial node. The matchmaker node then detaches the connection between itself and the initial node. The walk like that will continue k steps and the lightest load nodes will be selected to migration.

In (Wang, S., *et al*., 2010), the authors presented a load balance method for a three-level cloud-computing network: the service node, the service manager and the request manager. The proposed two-phase scheduling

algorithm integrates OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) to assist in the selection for effective service nodes. First, it finds all nodes having available resource > threshold. It then randomly distributes sub-tasks to those nodes (OLB algorithm). With set of sub-tasks and set of available nodes belong to a service manager, it applies LBMM. It sorts available nodes according min execution time, sorts sub-tasks according to min completion time, assigns min completion time sub-task to min execution time nodes, moves the assigned sub-task out of list, moves the assigned node to the end of the node list and repeats assign process until all sub-tasks assigned.

The work in (Galloway, J.M., *et al*., 2011) proposed a power aware load balance algorithm. The PALB algorithm first gathers the utilization percentage of each active compute node. If all compute nodes n are above 75% utilization, PALB instantiates a new virtual machine on the compute node with the lowest utilization. Otherwise, the new virtual machine (VM) is booted on the compute node with the highest utilization (if it can accommodate the size of the VM). If all currently active compute nodes have utilization over 75%, PALB sends turning on command to power on additional compute nodes (as long as there are more available compute nodes). If the compute node is using less than 25% of its resources, PALB sends a shutdown command to that node.

From the description above, pure load balance algorithms such as round robin, random, least connect are only suitable for homogeneous cloud system as they treat each computing node equally. With heterogeneous system, using those algorithms may create unbalanced state and weighted round robin or weighted random algorithms are more suitable. Algorithms in (Randles, M., *et al*., 2010; Wang, S., *et al*., 2010) mainly applied to large-scale cloud system. Unlike other algorithms, the algorithm in (Galloway, J.M., *et al*, 2011) considers not only load balance, but also energy saving by turning off servers.

7) Energy efficient algorithms for clouds support resource on demand:

In (Do, T.V., 2011), a simple energy-aware policy incorporating migration scheme of virtual servers is proposed to achieve the aim of green computing. The migration schemes are popular strategy such as round robin, first fit, etc. The policy automatically governs physical hosts to a low-energy consuming state when no virtual servers are allocated in a specific physical host. It automatically manages a physical host into the operating state of full functionality when virtual servers are assigned.

The open source IaaS Cloud package Eucalyptus has integrated Power save policy as a scheduling option (http://open.eucalyptus.com/.; Nurmi, D., *et al*., 2009). The core of the Power save policy is the First - Fit heuristic. The First- Fit method attempts to deploy a virtual machine to the first machine in a physical machine list that can accommodate this virtual machine. If no physical machine is found, then a new physical machine will be booted to host this virtual machine.

In the work of (Mazzucco, M., *et al*., 2010), the authors tried to save energy by optimizing the computing resource usage. The main idea is turning on a number of sufficient servers to meet the migration demand. Other servers are turned off. To realize this idea, the common techniques are determining the demand and determining suitable set of available servers to meet the demand for the next period. This strategy is very similar to the idea of (Lubin, B., *et al*., 2009; http://en.wikipedia.org/wiki/Gomory%E2%80%93Hu_tree) applying to the generic data centre. The works in (Do, T.V., C. Rotter, 2012; Mitrani, I., 2013; Mitrani, I., 2011; Do, T.V., U.R. Krieger, 2009) have the same idea but does not predict the demand. Instead, the authors build cloud-managing models to determine the optimized number of server to be turned on. This optimized value must balance the energy consumption and other criteria such as high performance (Mitrani, I., 2013), customer impatience (Mitrani, I., 2011), blocking probability (Do, T.V., C. Rotter, 2012; Do, T.V., U.R. Krieger, 2009).

Both works in (Quan, D.M., *et al*., 2011; Beloglazov, A., *et al*., 2012) used the same energy aware Best Fit strategy to allocate resource for the new coming VM. At first step, a power consumption model for data centre is defined. After that, the algorithm walks through all servers to determine if the server has enough resources to host the VM. With the feasible one, it estimates the future power if the VM is deployed on that machine. The feasible server with the smallest future power will be selected.

In (Srikantaiah, S., *et al*., 2008), the authors proposed an algorithm for Energy aware VM consolidation. With the experiment, the authors found that there exists an optimal combination of CPU and disk utilization. At this optimal point, the energy per transaction is minimum. Next, as each request arrives, it is allocated to a server, resulting in the desired workload distribution across servers. The used heuristic maximizes the sum of the Euclidean distances of the current migration s to the optimal point at each server. If the request cannot be allocated, a new server is turned on. Then all requests are re-allocated using the same heuristic, in an arbitrary order.

From the description above, we can see the difference in saving energy mechanism of described algorithms. The work in (Do, T.V., 2011) saves energy by setting servers to the lower power consumption state when there is no VM on the machine. The works in (http://open.eucalyptus.com/.; Nurmi, D., *et al*., 2009; Mazzucco, M., *et al*., 2010; Mitrani, I., 2013; Mitrani, I., 2011; Do, T.V., U.R. Krieger, 2009) try to use sufficient number of servers to host load and other free machines are turned off. The works in (Quan, D.M., *et al*., 2011; Beloglazov,

A., *et al*., 2012) go further by allocating load to the least power consumption servers. Unlike other algorithms that deal with VM, the work in (Srikantaiah, S., *et al*., 2008) goes deeper with transactions distribution among VMs.

8) Number of active hosts minimization algorithms for clouds support resource on demand:

The placement algorithm for VMs in a data centre allocates various resources such as memory, bandwidth, processing power, etc. from a physical machine (PM) to VMs with the goal of minimizing the number of PMs used. This problem can be viewed as a multi-dimensional packing problem. In (Lee, S., *et al*., 2011) the authors discussed several heuristics to solve this problem. Each host is presented by the host's vector of capacities H = (h_1, h_2, . . ., h_d). Each VM is represented by its vector of demands V = (v_1, v_2, …, v_d).

The popular heuristic for bin packing problem is the First Fit Decreasing (FFD). This heuristic orders the bins and the objects in size decreasing order. Starting with the first bin, it iterates over the object. It places objects into the first bin till no more objects can be placed into it. It then considers the first bin to be filled and proceeds to the second bin with the same procedure.

The important task for applying the FFD heuristic is determining the size of the bin and the size of the object. For cloud computing, the authors presented several ways to handle this task and thus, created several variations of the FFD heuristic.

FFDProd heuristic sorts the servers and VMs according to

$$\text{Volume(V)} = \prod_i v_i \quad (3)$$

$$\text{Volume(H)} = \prod_i h_i \quad (4)$$

FFDSum sorts the servers according to

$$\text{Volume(V)} = \sum_i w_i * v_i \quad (5)$$

$$\text{Volume(H)} = \sum_i w_i * h_i \quad (6)$$

$$w_i = \sum_{VM} \frac{v_i}{h_i} \quad (7)$$

Dot-Product heuristic - At time t let H(t) denote the vector of remaining or residual capacities of the current open host, i.e. subtract from the host's capacity the total demand of all VMs currently assigned to it. It places the VM that maximizes the dot product with the vector of remaining capacities $\sum_i w_i * v_i * h(t)_i$ without violating the capacity constraint.

Norm-based Greedy heuristic - for the l_2 norm distance metric, from all unassigned VMs, it places the VM *v* that minimizes the quantity $\sum w_i * (v_i - h(t)_i)^2$ and the assignment does not violate the capacity constraints.

Do not use heuristic, the work in (Bellur, U., *et al*., 2010) modeled the problem as the quadratic programming problem to be solved with (Kozlov, M.K., *et al*., 1980) and the integer linear programming problem to be solved with a standard LP solver (lp-solve. http://lpsolve.sourceforge.net/5.5/).

In (Van, H., and F. Tran, 2009), the author described the Global Decision Module (GDM) which is responsible for two main tasks: determining the VM migration vectors *Ni* for each application *ai* (*VM Provisioning*), and placing these VMs on PMs in order to minimize the number of active PMs (*VM Packing*). These two phases are expressed as two *Constraint Satisfaction Problems* (CSP) which are handled by a *Constraint Solver*.

From the above description, we can see that there are two classes of algorithms. Algorithms described in (Lee, S., *et al*., 2011) use heuristics and they are quite fast. Algorithms in [40,43] use global optimization techniques such as quadratic programming or linear programming. Thus, they may have long execution time.

9) Miscellaneous objectives algorithms for clouds support resource on demand:

The work in (http://en.wikipedia.org/wiki/Gomory%E2%80%93Hu_tree) proposed a mechanism called CLUSTER-AND-CUT to improve the Scalability of Data Centre Networks. It first partitions VMs into VM-clusters and partitions slots into slot-clusters. VM-clusters are obtained via classical min-cut graph algorithm (http://aws.amazon.com/ec2/). The data centre operators can obtain slot-clusters manually. The algorithm then maps each VM-cluster to a slot cluster. For each VM-cluster and its associated slot-cluster, it calls cluster-and-cut for a smaller problem size.

In (Machida, F., *et al*., 2010), the authors recognized the serious risks of host server failures that induce unexpected downs of all hosted virtual machines and applications. To protect required high-availability

applications from unpredictable host server failures, redundant configuration using virtual machines can be an effective countermeasure. The proposed method estimates the requisite minimum number of VMs according to the performance requirements of application services. It then decides an optimum VM placement so that minimum configurations survive at any k host server failures.

In (Tsakalozos, K., *et al*., 2011), the authors presented a custom optimization mechanism. The mechanism allows operator to select one among following goals: Reserve a single PM for a specific VM; Minimize traffic; Spread VMs across separate PMs; Reduce the number of PMs used; Offload a specific PM. The mechanism is two-phase optimization process. During the first phase, it selects a subset of PMs called cohort with properties that best serve the VM placement. Resource is divided into level according to migration ability. The first phase algorithm gradually explores all cohort levels in search of a promising "neighborhood". In the second phase, the mechanism solves a constraint satisfaction problem that yields a near optimal VM-to-PM mapping.

From the description above, we can see the difference in goals of those algorithms. While the work in (http://en.wikipedia.org/wiki/Gomory%E2%80%93Hu_tree) focuses on the scalability of the cloud system, algorithm in (Machida, F., *et al*., 2010) tries to deal with failure by allocating redundant resources to load. The work in (Tsakalozos, K., *et al*., 2011) allows custom optimization.

VMs migration algorithms:

10) Load balance:

To balance the load, there are two main tasks: to detect if there is imbalance in the system and to move:

Well-known, VMware system (Epping, D., F. Denneman, 2010) "target load standard deviation" metric (CHLSD) (THLSD) ", the standard deviation of the current load of the hosts' first task is handled by comparing. CHLSD hosts "entitlement value consistent with standard deviation is calculated. Each host's ability to host the same host as described in the formula for the value of the amount calculated by dividing the entire virtual machine load.

$$Load = \frac{\sum VM\_entitlement}{Host\_capacity} \tag{8}$$

The operator predefines THLSD value. If the CHLSD exceeds the THLSD, the cluster is considered imbalanced. For the second task, VMware uses the following algorithm. It checks if the cluster is imbalanced (CHLSD > THLSD), simulates moving each VM from the highly load host to the lower load host to calculate CHLSD, adds migration information giving best improving CHLSD to a list and then repeats the process until CHLSD < THLSD.

In (Wood, T., *et al*., 2007; Khanna, G., *et al*., 2006), the authors proposed the Reactive Load Balancing mechanism using local state of each PM. The mechanism detects the imbalance of each physical machine. If a PM has the resource usage > threshold for any type of resource, it is considered imbalance. VMs from that PM must be migrated to the under loaded PM. To select VM to be migrated, the work in (Wood, T., P. Shenoy and Arun, 2007) defined the parameter

$$VSR = \frac{\frac{1}{1-cpu} * \frac{1}{1-net} * \frac{1}{1-mem}}{VM_{memorySize}} \tag{9}$$

The VM having minimum VSR will be selected.

Unlike the work in (Wood, T., P. Shenoy and Arun, 2007), the work in (Khanna, G., *et al*., 2006) defined the parameter L for VM selection.

$$L = migration\_cost_{vector} * utilization_{vector} \tag{10}$$

The mechanism chooses the VM having minimum L and migrate to the PM that has least enough residual capacity.

In (Arzuaga, E., and D.R. Kaeli, 2010; Singh, A., *et al*., 2008), the authors proposed the Proactive Load Balancing mechanism considering the global state of the PM. For a cluster, a PM is imbalance if the coefficient of variance of PM's load > threshold. In (Arzuaga, E., and D.R. Kaeli, 2010), the overloaded VM from overloaded PM will be migrated to the under loaded PM. The work in (Singh, A., *et al*., 2008) move under loaded VM from over loaded PM to the under loaded PM.

In (Zhao, Y., and W. Huang, 2009), the authors presented the Compare and Balance algorithm. Algorithms are executed concurrently on independent physical hosts, and having limited information or no information about what the other parts of the algorithm are doing. It picks a VM_i running in the current host, picks randomly another host in the set of active hosts, calculates usage level of the current host c and the selected host c', if c>c' migrates VM_i to the selected host with the probability of c – c', and then repeats the process with other VMs.

From the description above, we can see that the load balance algorithms depend on how they define the balanced concept. The work in (Epping, D., F. Denneman, 2010) considered the imbalance as CHLSD > THLSD. The works in (Wood, T., *et al*., 2007; Khanna, G., *et al*., 2006; Arzuaga, E., and D.R. Kaeli, 2010; Singh, A., *et al*., 2008) tried to detect the imbalance using a threshold value. Using the probability, the work in

(Zhao, Y., and W. Huang, 2009) considered that the probability of having imbalance increases along with the greater difference of resource usage of a PM compared with another PM.

11) Energy efficient:

In (Verma, A., *et al*., 2008), the authors proposed the *min Power Placement algorithm with History, mPPH*. *mPPH* algorithm tries to minimize migrations by migrating as few VMs as possible with two phases. In the first phase, it determines a target utilization for each server based on the power model for the server. In the second phase, the mechanism calls the bin-packing algorithm.

The work in (Takeda S., and T. Takemura, 2010) proposed a rank-based VM consolidation method for power saving in data centres. Every PM has a server rank that is a unique value representing selection priority of the PM. Usually, the rank is determined by the data centre operator. The mechanism sets 2 values R_{high} and R_{low}. The algorithm only considers VMs in PMs having load > R_{high} and load < R_{low} as candidate for migration. The remapping process is then done with the First Fit Decreasing heuristic. VMware also has the same idea as this one for its DPM (Dynamic Power Management) module (Epping, D., F. Denneman, 2010).

In (Li, B., *et al*., 2009), the author proposed a mechanism that uses migration in three basis activities of the system: Workload Arrival Event, Workload Departure Event and Workload Resizing Event. For insert, it uses Best-Fit algorithm. For departure, when a workload finishes its work and departs from node *x*, the algorithm reinserts the other workloads on *x*. For Workload Resizing, it can be transformed to a *Pop workload size x* and an *Insert* Procedure workload size y.

The work in (Lin, C.C., *et al*., 2011) proposed the dynamic Round-Robin algorithm for energy efficient VM migration including two rules. In the first rule, if a VM has finished and there are still other VMs hosted on the same physical machine, this physical machine will accept no more new VM. When the rest of the virtual machines finish their execution, this physical machine can be shutdown. In the second rule, if a physical machine is in the "retiring" state for a sufficiently long period of time, it will not wait for the residing virtual machines to finish. The physical machine will be forced to migrate the rest of the virtual machines to other physical machines, and shutdown after the migration finishes.

The work in (Beloglazov, A., *et al*., 2012) proposed a heuristic for energy efficient VM migration. The optimization of the current VM migration is carried out in two steps. At the first step, the algorithm selects VMs need to be migrated. At the second step, the chosen VMs are placed on the hosts using the modified best fit decreasing algorithm.

In (Quan, D.M., *et al*., 2011), the authors proposed an algorithm called F4G-CG to optimize the energy consumption of data centres using VM migration. The F4G-CG algorithm has two main phases. In the first phase, the algorithm moves the VMs from low load servers to higher load servers if possible in order to free the low load server. The free low load server can be turned off. In the second phase, the algorithm moves the VMs from the old servers to the modern servers. The free old servers can be turned off.

From the description above, we can see the difference in input parameters for migrations algorithm. While the work in (Verma, A., *et al*., 2008) considers all VMs of the cloud system for migration, the work in (Beloglazov, A., *et al*., 2012; Takeda S., and T. Takemura, 2010) focus on the VMs on very high or very low load servers. The work in (Li, B., *et al*., 2009) deals with even the individual work size change. All those works do not support server turning on/off policy. The work in (Lin, C.C., *et al*., 2011) moves VMs from low load servers to high load servers in order to turn off free servers. Also having the same idea, the work in [36] extends it by moving load from old servers to more modern servers to turn off old servers.

12) VM migration with Miscellanous objectives:

In (Xu, J., and J. Fortes, 2010) work in the virtualized data center environments, a method is proposed for multi-objective virtual machine placement. The objectives of minimizing resource waste and reduce power consumption and dissipation which reduce costs. An improved fuzzy multi-objective evaluation of the genetic algorithm to efficiently search large solution space and possibly conflicting objectives combining the proposed facility.

In (Bobroff, N., *et al*., 2007) employed the SLA violations and proposed a method to manage the placement of dynamic virtual machines. Overall demand for paper in recent years, a series of resource-based approach is to predict future needs. Future resource needs, based on the best-fit algorithm, November VM remaps.

From the above description, we can see the mechanism of goal difference as well as the methods described. (Xu, J., and J. Fortes, 2010) Supports multiple purposes, making it time-consuming to use the genetic algorithm. (Bobroff, N., *et al*., 2007) The work aims to manage the SLA violation, and it uses heuristics to speed up the run-time best suited to heuristic with fast run-time.

Discussion:

In this section we each subsection in section 4. The difference between the methods of analysis, we focus on the applicability of the solutions studied up to the real environment. Public and Private Clouds: real environment, IaaS cloud data centers, there are two main types.

Public clouds or clouds the ability to provide resources for everyone to afford the cost of resource usage for business. IaaS cloud infrastructure for data centers offer a wide range of products is the source of the trend. For example, VM is a representation of the events on the market at the time (http://aws.amazon.com/ec2/), Amazon EC2 VM instances allocated resource. This policy provides flexible options for users using cloud resources. Thus, it forces the change from traditional computing, cloud computing. The main purpose for the benefit of the public cloud providers. As discussed, the same or a higher price for a fixed amount of resources to be gained by increasing the workload of the target resource. The first approach, etc. (Zaman, S. and D. Grosu, 2011) Such market mechanisms, can be used. The second approach is more efficient methods of energy can be realized (Quan, D.M., *et al*., 2011; Beloglazov, A., *et al*., 2012). Since workload grip rage trying to use a small number of energy efficiency measures. Showed significant improvement in the data center to the cloud to achieve that goal, compared to the literature, the initial placement of VM migration mechanism,. The initial placement algorithm energy efficiency, VM migration to increase the scale of a few percent.

Will be smaller than the total capacity available on demand, there should be no problem, and provides a wide range of policy. If demand is greater than the available capacity of the situation becomes even more complicated. In this situation, how to allocate resources among a number of products with the optimization of the profit is still an open issue. Another situation is that there is a peak in demand in a short period of time. Provision of adequate resources to deal with this situation would lead to inefficient resource utilization. Virtual (Sotomayor, B., *et al*., 2008) In addition to this issue, an initiative was proposed as a possible solution to the challenge. However, Amazon, a strong atmosphere of such a comprehensive study (http://aws.amazon.com/ec2/) It is still necessary.

Resources to provide users with a system limit for private clouds. Depending on company policy, may be different in private clouds. Each goal, the system can apply different solutions. Complex algorithms use simple heuristic: In general, both the VM migration initial virtual resource gathering methods, employment and migration is divided into two sections. Like for instance, a round-robin (http://aws.amazon.com/ec2/; http://www.gogrid.com/; http://opennebula.org/; http://www.enomaly.com/), random (http://www.enomaly.com/), at least (http://www.gogrid.com/), weighted selection [23, you can use the virtual job placement, load balancing for the purpose of starting a simple heuristic 46] or, genetic algorithm (Hu, J., *et al*., 2010) is applicable. Simple to use and easy to implement heuristics to speed up the execution. Generally has better performance by using complex algorithms. However, they are slower and more complicated to implement. It's simple heuristics real systems (Rimal, B.P., *et al*., 2009) seems to be the priority.

Conclusion:

Cloud computing is a promising model for the provision of services and computing applications. Each resource block IaaS cloud migration is an important part of the system. In this paper, we analyze the physical machines to virtual machines and cloud computing systems in treating different algorithm to map. Recent research developments will be discussed and execution stages, business models and classify the resource migration targets.

IaaS cloud computing systems efficiently, a well-known and widely used resource migration is to study the problem. Such migration results in the need for the market, game theory, resource allocation and resource under a different business model IaaS cloud infrastructure is made in both homogeneous and heterogeneous. Public cloud (or business: the proposed mechanisms of migration, etc. We are satisfied with the programming and control of the two main types of cloud data center solutions suited to the study was discussed, such as the genetic algorithm for linear programming, the well-known methods of cloud applications ranging from simple heuristics) and private cloud. From the survey, said the open issues and future direction.

REFERENCES

Rimal, B.P., E. Choi, I. Lumb, 2009. A Clasification and Survey of Cloud Computing Systems, Proceeding of the Fifth International Joint Conference on INC, IMS and IDC, pp: 44-51.

http://aws.amazon.com/ec2/

http://www.enki.co/

http://www.gogrid.com/

http://www.engineyard.com/products/cloud

http://www.google.com/apps/intl/en/business/cloud.html

http://www.netsuite.com/portal/home.shtml

http://www.salesforce.com/ap/?ir=1

https://developers.google.com/appengine/

Endo, P.T., G.E. Gonçalves, J. Kelner, D. Sadok, 2010. A Survey on Open-source Cloud Computing Solutions, Proceedings of the 28th edition of the Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2010), pp: 3-16.

Beloglazov, A., R. Buyya, Y.C. Lee, A.Y. Zomaya, 2011. A Clasification and Survey of Energy-Efficient Data Centers and Cloud Computing Systems. Advances in Computers, 82: 47-111.

Teng, F., 2012. MANAGEMENT DES DONN´EES ET ORDONNANCEMENT DES TˆACHES SUR ARCHITECTURES DISTRIBU´EES, PhD thesis of E´COLE CENTRALE PARIS ET MANUFACTURES.

Basmadjian, R., N. Ali, F. Niedermeier, H.d. Meer and G. Giuliani, 2011. A Methodology to Predict the Power Consumption for Data Centres, Proceedings of e-Energy, pp: 1-10.

Sotomayor, B., K. Keahey, I.T. Foster, 2008. Combining batch execution and leasing using virtual machines, Proceedings of HPDC, pp: 87-96.

Lifka, D.A., 1995. The ANL/IBM SP scheduling system, Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing, IPPS '95, pp: 295-303.

Mu'alem, A.W. and D.G. Feitelson, 2001. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. IEEE Trans. Parallel Distrib. Syst., 12(6): 529-543.

Wang, X., 2011. Research on Adaptive QoS-Aware Resource Reservation Management in Cloud Service Environments, Proceedings of 2011 IEEE Asia-Pacific Services Computing Conference (APSCC 2011), pp: 147-152.

Zhao, M. and R.J. Figueiredo, 2007. Experimental study of virtual machine migration in support of reservation of cluster resources, Proceedings of the 2nd international workshop on Virtualization technology in distributed computing, pp: 1-8.

Zaman, S. and D. Grosu, 2010. Combinatorial auction-based migration of virtual machine instances in clouds, Proceedings of the 2nd IEEE Intl. Conf. On Cloud Computing Technology and Science, pp: 127-134.

Zaman, S. and D. Grosu, 2011. Combinatorial Auction-Based Dynamic VM Provisioning and Migration in Clouds, Proceedings of CloudCom, pp: 107-114.

Jalaparti, V., G.D. Nguyen, I. Gupta, M. Caesar, 2010. Cloud Resource Migration Games, Illinois Technical Report, pp: 124-133.

Wei, G., A.V. Vasilakos, Y. Zheng, N. Xiong, 2010. A game-theoretic method of fair resource migration for cloud computing services, J Supercomputer, 54: 252-269.

http://opennebula.org/

http://www.enomaly.com/

Hu, J., J. Gu, G. Sun and T. Zhao, 2010. A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment, Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp: 89-96.

Nisan, N., T. Roughgarden, E. Tardos and V.V. Vazirani, 2007. Algorithmic Game Theory. Cambridge University Press.

Randles, M., D. Lamb and A. Taleb-Bendiab, 2010. A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, pp: 551-556.

Wang, S., K. Yan, W. Liao and S. Wang, 2010. Towards a Load Balancing in a Three-level Cloud Computing Network, Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp: 108-113.

Galloway, J.M., K.L. Smith, S.S.Vibsky, 2011. Power Aware Load Balancing for Cloud Computing, Proceedings of WCECS2011, pp: 127-132.

Do, T.V., 2011. Comparison of Migration Schemes for Virtual Machines in Energy-Aware Server Farms, The Computer Journal, 54(11): 1790-1797.

http://open.eucalyptus.com/.

Nurmi, D., R. Wolski, C.Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov, 2009. The eucalyptus open-source cloud-computing system, Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID '09, pp: 124-131.

Mazzucco, M., D. Dyachuk and R. Deters, 2010. Maximizing cloud providers' revenues via energy aware migration policies, Proceedings of the IEEE International Conference on Cloud Computing, pp: 131-138.

Lubin, B., J.O. Kephart, R. Das, D.C. Parkes, 2009. Expressive Power-Based Resource Migration for Data Centers, Proceedings of the 21st international joint conference on Artifical intelligence, pp: 1451-1456.

Chase, J.S., D.C. Anderson, P.N. Thakar, A.M. Vahdat, R.P. Doyle, 2001. Managing energy and server resources in hosting centers, ACM SIGOPS Operating Systems Review, 35(5): 103-116.

Quan, D.M., R. Basmadjian, H.d. Meer, R. Lent, T. Mahmoodi, D. Sannelli, F. Mezza, L. Telesca, C. Dupont, 2011. Energy Efficient Resource Migration Strategy for Cloud Data Centres, Proceedings of ISCIS, pp: 133-141.

Beloglazov, A., J.H. Abawajy, R. Buyya, 2012. Energy-aware resource migration heuristics for efficient management of data centers for Cloud computing, Future Generation Comp. Syst., 28(5): 755-768.

Srikantaiah, S., A. Kansal, F. Zhao, 2008, Energy aware consolidation for cloud computing, Proceedings of the 2008 conference on Power aware computing and systems, pp: 1-10.

Lee, S., R. Panigrahy, V. Prabhakaran, V. Ramasubrahmanian, K. Talwar, L. Uyeda and U. Wieder, 2011. Validating Heuristics for Virtual Machine Consolidation, Microsoft Research, MSR-TR-2011-9, pp: 1-14.

Bellur, U., C. Rao and M. Kumar, 2010. Optimal Placement Algorithms for Virtual Machines, Proceedings of CoRR, pp: 103-110.

Kozlov, M.K., S.P. Tarasov and L.G. Khachiyan, 1980. The polynomial solvability of convex quadratic programming, USSR Computational Mathematics and Mathematical Physics, 20(5): 223-228.

lp-solve. http://lpsolve.sourceforge.net/5.5/

Van, H., and F. Tran, 2009. Autonomic resource management for service host platforms, Proceedings of Workshop on Software Engineering Challenges in Cloud Computing, pp: 1-8.

Meng, X., V. Pappas and L. Zhang, 2010. Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement, Proceedings of IEEE 2010 INFOCOM, pp: 1-9.

http://en.wikipedia.org/wiki/Gomory%E2%80%93Hu_tree

Chandrasekaran, B., R. Purush, B. Douglas and D. Schmidt, 2007, Virtualization Management Using Microsoft System Center and Dell OpenManage, Dell Power Solutions, pp: 40-44.

Machida, F., M. Kawato and Y. Maeno, 2010. Redundant Virtual Machine Placement for Fault-tolerant Consolidated Server Clusters, Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium, pp: 32-39.

Tsakalozos, K., M. Roussopoulos and A. Delis, 2011. VM Placement in non-Homogeneous IaaS-Clouds, Proceedings of 9th International Conference on Service Oriented Computing (ICSOC 2011), pp: 172-187.

Epping, D., F. Denneman, 2010. VMware vSphere 4.1 HA and DRS Technical Deepdive, CreateSpace, ISBN-10: 1456301446.

Wood, T., P. Shenoy and Arun, 2007. Black-box and gray-box strategies for virtual machine migration, NSDI 2007, pp: 229–242.

Khanna, G., K. Beaty, G. Kar and A. Kochut, 2006. Application performance management in virtualized server environments, Proceedings of 10th IEEE/IFIP Network Operations and Management Symposium NOMS 2006, pp: 373-381.

Arzuaga, E., and D.R. Kaeli, 2010. Quantifying load imbalance on virtualized enterprise servers, Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering, pp: 235-242.

Singh, A., M. Korupolu and D. Mohapatra, 2008. Server-storage virtualization: Integration and load balancing in data centers, Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, pp: 1-12.

Zhao, Y., and W. Huang, 2009. Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, pp: 170-175.

Verma, A., P. Ahuja and A. Neogi, 2008. pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, pp: 243-264.

Takeda S., and T. Takemura, 2010. A rank-based vm consolidation method for power saving in datacenters. Information and Media Technologies, 5(3): 994-1002.

Lin, C.C., P. Liu, J.J. Wu, 2011. Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems, 2011 Fourth IEEE International Conference on Utility and Cloud Computing, pp: 81-88.

Li, B., J. Li, J. Huai, T. Wo, Q. Li, L. Zhong, 2009. EnaCloud: An Energy-saving Application Live Placement Approach on Cloud Computing Environments, IEEE International Conference on Cloud Computing, 2009. CLOUD '09, pp: 17-24.

Lee, C.C., and D.T. Lee, 1985. A simple on-line bin-packing algorithm. Journal of the ACM, 32(3): 562-572.

Xu, J., and J. Fortes, 2010. Multi-objective Virtual Machine Placement in Virtualized Data Center Environments, Proceedings of the 2010 IEEE/ACM Conference on Green Computing and Communications, pp: 179-188.

Bobroff, N., A. Kochut and K. Beaty, 2007. Dynamic Placement of Virtual Machines for Managing SLA Violations, Proceedings of the 10th IFIP/IEEE Symposium on Integrated Network Management, pp: 119-128.

Waldspurger, C.A., 2002. Memory Resource Management in VMware ESX Server, ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, pp: 181-194.

Silpa, C.S., S.S.M. Basha, 2013. A Comparative Analysis of Scheduling Policies in Cloud Computing Environment. International Journal of Computer Applications, 67(20): 16-24.

Do, T.V., C. Rotter, 2012. Comparison of scheduling schemes for on-demand IaaS requests. Journal of Systems and Software, 85(6): 1400-1408.

Mitrani, I., 2013, Managing performance and power consumption in a server farm. Annals OR 202(1), pp: 121-134.

Mitrani, I., 2011, Service center trade-offs between customer impatience and power consumption. Perform. Eval. 68(11): 1222-1231.

Do, T.V., U.R. Krieger, 2009. A Performance Model for Maintenance Tasks in an Environment of Virtualized Servers. In: IFIP/TC6 NETWORKING 2009, pp: 931-942.