**CHAPTER 2**

**LITERATURE SURVEY**

Cloud Computing is emerging as a mainstream technology that provides virtualized physical resources and infrastructure. Cloud Computing is focused on everything-as-a-Service paradigm. There are three technologies used for providing cloud services: Software as a Service, Platform as a Service and Infrastructure as a Service. Although Software as a Service was the first adopted provision model, from the cloud customer’s perspective, the last two technologies offer more flexibility by providing the virtualized resources as service rather than just software. Thus, the number of clouds providing Infrastructure as a Service is starting to grow. This model promises to offer optimized and maintained virtualized resources.

Cloud computing can offer on-demand capacity as utility and can provide the abilities to use insufficient resources effectively. Therefore, selecting proper resource and managing the resource efficiently are critical for the performance of executions. That is, we need to identify application characteristics first and then provide an optimal resource provisioning in order to attain good application execution on cloud.

The following papers presented, offer a wide picture of the current trends in resource provisioning and capacity building within cloud. The following papers also offer a brief idea of associating Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS) with the concept of resource provisioning.

**2.1 Efficient Manager for Virtualized Resource Provisioning in Cloud System**

**Abstract**:

Cloud Systems provide the computing infrastructure and on-demand capacity required to host services. This paper [1] presents a new provisioning mechanism for Cloud Systems the project addresses the key requirements in managing resources at the infrastructure level. The resource manager allocates virtual resources in a flexible way, provides on-demand elasticity, which are the most important feature of Cloud Computing compared to the traditional hosting strategies. New resources can be dynamically allocated on-demand or policy-based. This is one of the novel contributions of this paper. For resource scheduling mechanism was developed that takes into account virtual machines’ capabilities and well defined policies and it uses a genetic algorithm. A QoS constraint base algorithm is used in order to maximize the performance, according to different defined policies.

**Specific contribution**:

* This paper proposes a solution to the problems regarding virtual resource management in Cloud Systems. A resource manager was developed for a cloud environment that filters user’s requests accordingly to defined policies, schedules the requested virtual machines and deploys them in the Cloud. Another idea specified in this paper is the possibility to dynamically allocate new resources according to a predefined policy.

* There are several cloud software platforms that are doing the resource management at the infrastructure level. Examples of such of platforms include OpenNebula, Eucalyptus and Nimbus. Based on a series of conducted experiments, the authors justified that their approach has a better average utilization rate than any of the above schedulers.
* A solution with a better performance for the VM’s scheduling algorithm is optimized genetic algorithm, which uses the shortest genes and introduces the idea of dividend policy in economics.
* The proposed architecture consists of three components : User Interface (UI),

Subscriber Server (SS) and Resource Manager (RM). Each of those components is directly connected to a Cloud Front-End. The customers have access to the Cloud through the User Interface. The customer must first create an account by using the subscription mechanism. The User Interface will send the VMs requests to the Resource Manager and the subscription messages to the Subscriber Server. The messages exchanged between those three components use a XML format. The User Interface also provides the capability of customizing the virtual machines.

* Scheduling is one of the key issues in the workflow management, especially in the cloud systems. It is the process that maps and manages the execution of interdependent tasks on the distributed resources. For this implementation they have used an improved genetic algorithm. The scheduler implements a timer that raises a Request event at predefined intervals. This algorithm makes use of chromosome encoding, the fitness function, population initialization and the genetic operators.

.

* This application will reside on the Front-End, along with the Apache server running the web application with the user interface that allows users to subscribe and make VM requests. The front-end has RedHat Scientific Linux as OS, a MySQL database, the Apache server and the OpenNebula middleware. The hardware configuration for the front-end consists of: 1 GB of memory, Intel Core 2 Duo E8200 processor with 2.66 GHz frequency, and a hard-disk capacity of 500 GB.
* The solution proposed in this paper is efficient and has a good utilization rate. The entire architecture is independent from the Front-End. A genetic algorithm was implemented in order to find the best scheduling solution. Also, the algorithm besides the virtual machine capabilities is considering well defined policies in order to find the proper planning solution.

**2.2 Energy- aware resource allocation heuristics for efficient management of data centers for Cloud computing**

**Abstract:**

Cloud computing offers utility-oriented IT services to users worldwide. Based on a pay-as-you-go model, it enables hosting of pervasive applications from consumer, scientific, and business domains. However, data centers hosting Cloud applications consume huge amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. Therefore, Green Cloud computing solutions are required that can not only minimize operational costs but also reduce the environmental impact. In this paper [2] an architectural framework and principles for energy-efficient Cloud computing are presented as follows.

**Specific Contributions:**

* Until recently, high performance has been the sole concern in data center deployments in the cloud computing scenario, and this demand has been fulfilled without paying much attention to energy consumption.
* As energy costs are increasing while availability dwindles, there is a need to shift the focus from optimizing data center resource management for pure performance to optimizing them for energy efficiency, while maintaining high service level performance.
* Therefore, Cloud service providers need to adopt measures to ensure that their profit margin is not dramatically reduced due to high energy costs.
* There is also increasing pressure from governments worldwide aimed at the reduction of carbon footprints, which have a significant impact on the climate change.
* .Green Cloud computing is envisioned to achieve not only the efficient processing and utilization of a computing infrastructure, but also to minimize energy consumption.
* As far as Green Cloud Architecture is concerned, there are four main entries:
  1. Consumers/Brokers: Cloud consumers or their brokers submit service requests from anywhere in the world to the Cloud.
  2. Green Service Allocator: Acts as the interface between the Cloud infrastructure and consumers.
  3. VMs: Multiple VMs can be dynamically started and stopped on a single physical machine according to incoming requests, hence providing the flexibility of configuring various partitions of resources on the same physical machine to different requirements of service requests.
  4. Physical Machines: The underlying physical computing servers provide the hardware infrastructure for creating virtualized resources to meet service demands.
* Green Cloud Computing plays a significant role in the reduction of data center energy consumption costs, and thus helps to develop a strong and competitive Cloud computing industry.

**2.3 Towards autonomic detection of SLA violations in Cloud infrastructures**

**Abstract**:

Cloud computing has become a popular platform for implementing scalable computing infrastructures provided on-demand on a case-by-case basis. Self-manageable Cloud infrastructures are required to comply with users’ requirements defined by Service Level Agreements (SLAs) and to minimize user interactions with the computing environment. Thus, adequate SLA monitoring strategies and timely detection of possible SLA violations represent challenging research issues. Methods in Detecting SLA Violation infrastructure (DeSVi) architecture and sensing SLA violations through sophisticated resource monitoring are presented in this paper [3].

**Specific Contributions**:

* Service provisioning in the Cloud relies on Service Level Agreements (SLAs) representing a contract signed between the customer and the service provider including non-functional requirements of the service specified as Quality of Service (QoS).
* SLA considers obligations, service pricing, and penalties in case of agreement violations.
* Prevention of SLA violations avoids penalties providers have to pay and on the other hand, based on flexible and timely reactions to possible SLA violations, user interaction with the system can be minimized, which enables Cloud computing to take roots as a flexible and reliable form of on-demand computing.
* The detection of possible SLA violations relies on the predefined service level objectives and utilization of knowledge databases to manage and prevent such violations.
* In order to assist Cloud providers in detecting SLA violations through resource monitoring, the DeSVi architecture is made use of.
* Based on the user requests, DeSVi allocates computing resources for a requested service and arranges its deployment on a virtualized environment.
* The main components of the DeSVi architecture are:
  + Automatic VM deployer
  + Application Deployer
  + LoM2HiS frame work
* The automatic VM deployer allocates necessary resources for the requested service and arranges its deployment on a virtual machine (VM).
* After service deployment, LoM2HiS frame work monitors the VMs and translates the low-level metrics into high-level SLAs using the specified mapping rules.
* To realize autonomic SLA management DeSVi utilizes a knowledge database for the evaluation of the monitored information in order to propose reactive actions in case of SLA violation situations.
* Advantages of flexible and reliable Cloud infrastructures are manifold. Hence unnecessary SLA violations can be prevented by subscribing to the methods mentioned above.

**2.4 Dynamic job Scheduling in Cloud Computing based on horizontal load balancing**

**Abstract:**

Cloud computing is a latest new computing paradigm where applications, data and IT services are provided across dynamic and geographically dispersed organizations. Improving the global throughput and utilizing the Cloud computing resources proficiently and gain the maximum profits with job scheduling system is one of the Cloud computing service providers’ ultimate objectives. The motivation in this paper [4] here is to establish a scheduling mechanism which follows the Lexi –search approach to find an optimal feasible assignment. The cost for assigning a task into a resource is probabilistic result considering the above criteria.

**Specific Contributions**:

* The scheduling algorithms in distributed systems usually have the goals of spreading the load on processors and maximizing their utilization while minimizing the total task execution time.
* Load balancing is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job.
* Lexi-Search approach is used to assign tasks to the available resources.
* The cost matrix is generated from a probabilistic factor based on some most vital condition of efficient task scheduling such as task arrival, task waiting time and the most important task processing time in a resource.
* The load balancing mechanism in the central middleware reduces the overhead of scheduling on a single middleware by partitioning the job queue and thus scalability issues are well maintained.
* Making the replication of the partitioned job queue ensures fault tolerance in the cloud since if any of the client fails, then that job could be reassigned into another client by another local middleware as the local middleware interact each other for every job updates.

**2.5 Heterogeneous Workload Consolidation for Efficient Management of Data Centres in Cloud Computing**

**Abstract:**

Cloud computing is a recent innovation, which provides various services on a usage based payment model. The rapid expansion in data centers has triggered the dramatic increase in energy used, operational cost and its effect on the environment in terms of carbon footprints. To reduce power Consumption, it is necessary to consolidate the hosting workloads. A Single Threshold technique for efficient consolidation of heterogeneous workloads is briefly illustrated in this paper [5]. The technique focuses on the energy consumption of the data centre due to the heterogeneity of the workloads and also gives information about the SLA violations.

**Specific contributions:**

* The objective in this case, is to demonstrate an efficient technique to consolidate the heterogeneous workloads in Cloud Computing.
* Workload Consolidation Techniques can be categorized into two forms: (1) **Homogeneous Workload Consolidation Techniques**These techniques include the similar kind of workloads which is having fixed number of parameters such as length, number of CPU’s required and buffer size of input and output files. Workload consolidation in data centers is achieved with the help of virtualization.

(2) **Heterogeneous Workload Consolidation Techniques**For this technique Green Cloud Architecture is used. The proposed technique works on the PaaS (Platform as a service) layer as allocation policies are implemented on PaaS layer which are followed by the IaaS (Infrastructure as a service) layer. These techniques include different kind of workload which is having varying parameters like length; number of CPU’s required and buffer size of input and output files.

* **VM Allocation Policy**The VM allocation Policy used is Single Threshold. In this policy, the upper utilization of the host is fixed and virtual machines are placed according to the current CPU utilization. For the execution of the applications, First Come First Serve scheduling algorithm is used. On arrival of the heterogeneous applications, the virtual machine is assigned to the application. Later, when the utilization of the host reaches up to the set threshold, virtual machines are migrated to the hosts whose threshold is below the fixed threshold.
* **Performance Metrics**In order to get the optimal points where energy consumption in KWh of the data centre is minimum two metrics are chosen. First performance metric gives the total energy. Consumption of the data centre by the resources due to the execution of heterogeneous workloads and the second performance metric gives the percentage SLA violations.

**2.6 Policy based resource allocation in IaaS cloud**

**Abstract:**

In present scenario, most of the Infrastructure as a Service (IaaS) clouds use simple resource allocation policies like immediate and best effort. Immediate allocation policy allocates the resources if available, otherwise the request is rejected. Best-effort policy also allocates the requested resources if available otherwise the request is placed in a FIFO queue it is difficult for the cloud provider to satisfy request due to finite resources. Haizea is a resource lease manager that tries to address these issues by introducing complex resource allocation policies. Haizea supports four kinds of resource allocation policies: immediate, best effort, advanced reservation and deadline sensitive. This paper [6] provides a better way to support deadline sensitive leases in Haizea while minimizing the total number of leases rejected by it. Proposed dynamic planning based scheduling algorithm is implemented in Haizea that can admit new leases and prepare the schedule whenever a new lease can be accommodated.

**Specific contribution:**

* In IaaS cloud the resources (compute capacity and storage) are provided in the form of virtual machines to users. Scheduler can be used to decide when and where to place these virtual machines. The main goal of the scheduler described in this paper is to minimize request rejection rate while satisfying resource allocation policies offered by cloud provider. Presently, most of the cloud providers rely on simple resource allocation policies like immediate and best effort.
* Nimbus, Eucalyptus and OpenNebula are cloud toolkits which can be used to setup a cloud on local infrastructure. The integration of OpenNebula and Haizea provides the only Virtual Infrastructure (VI) management solution offering advance reservation of capacity and configurable VM placement policy.
* Sometimes it is not possible for cloud providers to satisfy all the requests which come to them on immediate basis due to lack of resources. Haizea is an open source resource lease manager that tries to address this issue. User requests computational resources from Haizea in the form of a lease. The lease is accepted by Haizea if and only if Haizea can assure the resource allocation policy requested by this lease. Haizea assumes that best-effort leases are pre-emptable and they do not have any time constraints.
* Proposed scheduling algorithm for deadline sensitive leases: To request resources from Haizea, consumer has to submit lease in a specific format. Main parameters in this request are number of virtual nodes, amount of physical resources for each node, start time, duration and deadline CPU and memory are the two computational resources that can requested by the user. The software specified in software field of the request is considered as VM image. These VM images are stored at a central repository.
* Swapping and backfilling: Whenever a new deadline sensitive lease is submitted to Haizea, the existing algorithm in Haizea to handle deadline sensitive leases tries to find a single time slot to satisfy the whole lease. If it is not able to find such a slot then it will reschedule already accommodated deadline sensitive leases to make space for new lease while rescheduling it does not consider the amount of resources requested by the leases. It just sorts the deadline sensitive leases based on slack value and reschedules them considering pre-emption. The existing algorithm of Haizea to handle deadline sensitive leases tries to find a single time slot to satisfy the whole lease. One enhancement is to consider several time slots, which together can satisfy the complete lease within its deadline. This way several new leases may get scheduled without requiring rescheduling of other leases to make space for them.

**2.7 Evolutionary Optimal Virtual Machine Placement and** **Demand Forecaster for cloud computing**

**Abstract:**

Cloud computing allows the users to efficiently and dynamically provision computing resources to meet their IT needs. Most cloud providers offer two types of payment plans to the user, i.e, reservation and on-demand. The reservation plan is typically cheaper than the on-demand plan but reservation plan has to be provisioned in advance. Reserving the resources would be straightforward if the actual computing demand (e.g., job processing) is known in advance. However, in reality, the actual computing demand can be observed only at the point of actual usage. Therefore, it is difficult to reserve the correct amount of resources during the reservation to meet the computing demands of the users. An “evolutionary optimal virtual machine placement”(EOVMP)algorithm with a demand forecaster will be the best option in this case and it has been briefly illustrated in this paper [7].

**Specific Contributions:**

* It would be much more beneficial for an organization in the long term to outsource its IT needs to a third-party providers, e.g., Amazon EC2, Google AppEngine, and IBM Computing on Demand.
* When it comes to which third-party providers to choose, the most commonsensical choices would be the provider who offers these resources at the cheapest price. However, in reality, it is not as straightforward as it seems. Though providers may offer the prices at the lowest, they may not have sufficient computing resources to satisfy the needs of all the users (e.g., a provider has limited resources at a certain time period).
* Therefore, the users will have to optimize the resource allocation from multiple providers to meet their IT needs. Nonetheless, the user will now face a difficulty in deciding on how to split their demands to the various providers and ultimately to choose the best combination that will incur the smallest cost.
* This resource planning/provisioning problem can be classified as a combinatorial optimization problem.
* Combinatorial optimization problem is defined as finding the optimal (best) solution that either maximizes or minimizes a given revenue or cost function within a feasible problem set. Typically, such problems can be solved with any search algorithms or integer programming.
* The objective is to minimize the cost by predicting the demand of cloud users and optimizing the VM placement based on the user’s usage history.
* To achieve such objective, the evolutionary optimal virtual machine placement (EOVMP) algorithm and the demand forecaster have been introduced.
* First, a demand forecaster predicts the computing demand. Then, EOVMP uses this predicted demand to allocate the virtual machines using reservation and on-demand plans for job processing. The performance of the proposed schemes is evaluated by simulations and numerical studies.

**2.8 Improving Performance and Availability of Services** **Hosted on IaaS Clouds with Structural** **Constraint-aware Virtual Machine Placement**

**Abstract**:

The increasing popularity of modern virtualization based data centers continues to motivate both industry and academia to provide answers to a large variety of new and challenging questions on improvement of performance and availability of services hosted on IaaS clouds. Since resources are scarce, virtual machine placement is imperative. A structural constraint aware virtual machine placement (SCAVP) supports three types of constraints: demand, communication and availability. SCAVP is considered as an optimization problem which is explained in this paper [8].

**Specific Contributions**:

* Virtual machine (VM) placement has become a crucial issue in modern virtualization-based data centres since resources are scarce.
* Structural constraint-aware VM placement (SCAVP) is used to improve the performance and availability of services deployed in Infrastructure as a Service (IaaS) clouds.
* In SCAVP, an application is considered as a cluster of related VMs, rather than a set of individual VMs. By augmenting the placement system with awareness of availability and communication requirements, we can improve the overall performance and availability of the application.

* Communication constraints permits the minimization of overall communication costs by assigning VMs with large mutual communication costs to PMs in close proximity to one another.

* The core of the VM placement problem is to create a placement plan which dictates where VMs should be deployed in order to satisfy various constraints.
* Without loss of generality, an efficient initial placement plan is created by satisfying the following three structural constraints:
* Demand constraint, which is defined as the lower bound of resource allocations that each VM requires for the service to meet its Service level agreement.
* Availability constraint: Availability constraint is to improve the overall availability of the service.
* Communication constraint: Is a user-assigned value that represents the communication requirement between two VMs such that the application functions. The objective is to minimize the communication cost while satisfying both demands and availability constraints.
* Structural constraint-aware virtual machine placement is an effective way to improve the performance and availability of services hosted on modern datacenters, particularly on IaaS clouds.

* Careful VM placement can reduce communication costs significantly and improve overall system availability.

**2.9 Intelligent Cloud Capacity Management**

**Abstract:**

Cloud computing as a service promises many business benefits and it faces many challenges.One of the challenges is to effectively manage cloud capacity in response to the increased demand changes in clouds, as computing customers now can provision and de-provision virtual machines more frequently**.** This paper [9] explains cloud capacity prediction as a response to the challenge. An integrated solution for intelligent cloud capacity estimation is presented here. In this solution, a novel measure is introduced to quantify and guide the prediction process. The cloud capacity is estimated using the active virtual machines and the future provisioning/de-provisioning demands altogether.

**Specific contributions:**

* Capacity management in cloud is one of the challenges that has a critical impact on the service quality and the profitability of the cloud. If the demand is higher than the existing capacity the cloud has to turn down new customers and lose potential revenue. If the shortage is severe, even provisioning requests from existing customers have to be rejected, which defeats the promise that application in cloud can scaling-up whenever workload increases.
* This paper focuses on the problem of capacity estimation and management of cloud environment. It handles capacity estimation using a two-stage approach: an estimating stage and a controlling stage. The core of the capacity estimation system is the *Capacity Predictor*, which predicts the future cloud capacity based on the information of historical provisioning data, and notifies the *Power Management Module* to prepare the required resources (include active servers, active coolers, other auxiliaries, and scheduled labour) in advance. The *Capacity Controller* would keep tuning the capacity according to the actual workload of the cloud environment.
* The contribution of this paper can be summarized as follows: A solution is provided for the capacity estimation problem by decomposing it into the provisioning demand and de-provisioning demand. Authors have introduced a novel measure based on the uniqueness of cloud service to quantify the prediction result. By incorporating this measure, the prediction preference can be dynamically tuned according to actual running status of the cloud.

Methodology: The problem of cloud capacity management is to ensure that the amount of prepared resources in cloud is consistent with the real demands in the near future. Without loss of generality, cloud capacity is modelled as the number of virtual machines.

* Performance Evaluation Criteria: The consequences of over-estimation and under-estimation are different in cloud capacity prediction. An over-estimation has no negative effects on the customer and only causes idled cloud resources while an underestimation degrades the service quality. Considering the uniqueness of cloud demand, we propose a novel error measure called Demand Estimation Error (DEE). DEE is an asymmetric measure that models the overestimation and under estimation of the demand as different kinds of costs: the cost of idled resources and the penalty of SLA (Service Level Agreement).

* This paper has explained an effective framework for cloud capacity prediction based on the knowledge learnt from the requests history and leveraged the time series ensemble method to predict the provisioning demands and utilized the life span distribution of VM to estimate the de-provisioning requests.

**2.10 Performance and Power Management for Cloud Infrastructures**

**Abstract**:

A key issue for Cloud Computing data-centres is to maximize their profits and productivity by minimizing power consumption and Service Level Agreement (SLA) violations of hosted applications. In order to achieve the above mentioned prospects, brief details regarding a resource management framework combining a utility-based dynamic Virtual Machine provisioning manager and a dynamic VM placement manager are presented in this paper [10]. Both problems are modeled as constraint satisfaction problems. The VM provisioning process aims at maximizing a global utility capturing both the performance of the hosted applications with regard to their Service Level Agreement and the energy- related operational cost of the cloud computing infrastructure.

**Specific Contributions**:

* In a Cloud computing context, the objective is the amalgamation of applications belonging to different companies by minimizing power consumption and increasing productivity.
* The former class of applications tends to exhibit time-varying workloads with long-term variations and sudden workload spikes which are difficult to predict.
* The introduction of virtualization technologies in data centers has already allowed reducing the number of physical servers and has improved their utilization level.
* Migration to the cloud calls for a dynamic infrastructure in which resources are provisioned to the applications on the fly according to the actual workload of the service.
* A key issue with regard to resource provisioning and placement is to automate the decision-making process while providing high-level handles to the human administrator to control and tune the system.
* The VM Placement Manager handles the dynamic placement of the VMs on the datacenter physical hosts.
* The VM Provisioning Manager delegates to the VM Placement Manager the task of creating or destroying a VM.
* The VM provisioning process aims at maximizing a global utility capturing both the performance of the hosted applications with regard to their Service Level Agreement and the energy- related operational cost of the cloud computing infrastructure.
* The VM Provisioning Manager and Placement Manager are both implemented as a CSP program executed by a constraint solver.
* The Scheduler component takes care of the triggering of the control loop.
* The Application Environment is a composite component which wraps each hosted application on the cloud.
* Self-optimization is achieved through a combination of utility functions and a constraint programming approach. Hence with the help of the above mentioned techniques, the trade-off between power consumption and productivity is balanced.

**2.11 Conclusion**

This report provides a brief summary of the technical papers related to the framework for load balancing, resource provisioning, migration, power management issues in a cloud environment. Based on a systematic study of the above papers, we have incorporated some of the beneficial aspects to our research. The research involved the development of an indigenous algorithm called as the “Load Balancing Algorithm”. The above reports have motivated us to innovate on important aspects such as scalability which involves scaling of the CPU resources, live migration of virtual machines, and wireless access to the servers forming the base of the cloud through XenCenter.