Chapter 2

Taxonomy and Survey of Energy Efficient Cloud Approaches

Chapter abstract Due to enormous popularity of the Cloud in last few years, the number of Cloud data centers has increased considerably worldwide. This has lead to an issue of energy consumption and CO$_2$ emission by these data centers. Researchers, over recent times, have been showing keen interest in addressing the concern. Out of many different facets to deal with the issue, several researchers have been working in the directions of making optimum utilization of available resources and to turn-off not-required computing resources to save energy. Further, along with energy consumption, performance of Cloud system is also considered as important parameter for research, as Cloud works on the concepts of pay-as-you-go. Hence, it is required to monitor and measure the performance of Cloud services before making the final billing statements. In this chapter, we aspire to address the key terminologies involved with performance and energy efficiency parameters of Cloud computing. We begin with preamble to Cloud and Cloud architecture. Further, we move on to virtualization and virtual computing resource management. Later, we bring in the issue of energy efficiency, performance and virtual machine migration. At the end of the chapter, we illustrate some significant issues and challenges in Cloud. We wrap up the chapter by introducing multi-objective optimization technique, which is motivation for us to address the issue of energy consumption.
2.1 Overview of Cloud computing

Cloud is emerged as popular technology because of its pay-per-use model. It has come up with more facilities compared to other technologies like utility and grid computing that handle huge number of end users around the globe. In this section, we provide overview of various terminologies involved with fundamentals components of Cloud computing.

2.1.1 Definitions

NIST (Mell & Grance (2009)) defines Cloud computing as “a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. This definition covers important Cloud parameters with Cloud architectures, security aspects and deployment strategies. Particularly, five vital elements of Cloud computing are expressed as follow (Dillon et al. (2010)):

1. On-demand self-service: A user with resource request can avail computing resources at a requested time slot (as and when required) in automatic (without human intervention) and on-demand (the resources can be increased or decreased depending on demand) manner.

2. Broad network access: Computing resources in Cloud are delivered over the network (mostly via Internet) and used by numerous client applications on heterogeneous platforms like mobile phones, laptops and PDAs.

3. Resource pooling: A Cloud service provider’s computing resources are pooled together in datacenter to satisfy end user’s request using either the multi-tenancy or the virtualization model. Physical resources are dynamically allocated and de-allocated in the form of virtual resources. Two major factors considered to setting up such a pool-based computing paradigm are: economies of scale and specialization. Due to this model, these resources, their locations and originalities remain hidden from end users. Users need not to worry
about where their data are transferred or about the place where computing is carried out.

4. Rapid elasticity: Elasticity is the prime characteristic of Cloud. For end users, computing resources are available immediately. They can scale up the resources when needed and scale down when not required. Moreover, resource provisioning is a continuous process in Cloud and resources should be made available at any time.

5. Measured Service: Though computing resources are pooled and shared by multiple customers, the service providers use appropriate mechanisms to monitor and measure the usage of these resources for each individual customer. These metered mechanisms provide pay-per use facility to end users.

Service Model: With these five essential characteristics, the following sub-section categorizes three service models in Cloud:

1. Software as a Service (SaaS): Cloud customers deploy their applications/software on a hosting environment, which can be accessed and shared by various clients through Internet. Cloud customers do not have any control on the Cloud infrastructure. In SaaS, different Cloud consumers’ applications are organized and managed in a single logical environment to achieve optimization in speed, security, availability, disaster recovery and maintenance. Google Mail, SalesForce.com, Google Docs, Zoho etc. are few of the examples of SaaS.

2. Platform as a Service (PaaS): PaaS provides development platform that support the full 'Software Lifecycle’ to Cloud customers to develop Cloud applications and different services directly on the Cloud. While SaaS only hosts completed/full/ready-to-use Cloud applications, PaaS offers a development platform that allows customers to hosts both completed and in-progress Cloud applications. Different APIs are available on Cloud for appropriate application development. It provides and support full development environment including programming and OS support, different development tools, configuration management etc. Examples of PaaS are GoogleAppEngine(Googleappengine (2015)), Microsoft Azure(azure (2015)) etc.
3. Infrastructure as a Service (IaaS): Cloud customers can directly request the Cloud underlying infrastructures viz. processing elements, storage, networks and other primitive computing resources available in the shared pool of resources. Virtualization is the key technology behind IaaS. Virtualization creates virtual resources over physical resources. The basic strategy of virtualization is to keep abstraction between the multiple users sharing the same resources by providing the independent virtual machines (VM) that are isolated from both the underlying hardware and other ongoing VMs. Examples of IaaS are Amazon’s EC2 (EC2 (2015)), Microsoft Azure (azure (2015)) etc. With these basic three services couple of more services can be included based on their importance in Cloud.

4. Data storage as a Service (DaaS): It is a special type of IaaS. The provision of virtualized storage on demand to end users at any time introduces a separate Cloud service called data storage service. It allows customers to pay for only the part of database that they actually use rather than for the entire database. DaaS not only offers traditional storage interfaces such as RDBMS and file systems, but also provides table-style abstractions that are designed to scale out to store and retrieve a huge amount of data within a very compressed time frame. Examples of DaaS include Amazon S3, Google BigTable, Apache HBase, etc.

5. Identity as a Service (IaaS): An identity service is one that stores the information/identity associated with a digital entity in a secured form that can be queried and managed efficiently for electronic transactions. A data store, query engine, and policy engine that maintain data integrity are the core functions of the Identity services.

After the brief discussion on various types of Cloud service model, we move on to different Cloud deployment models.

Deployment Model:
Deployment models are the way the Cloud services are provided to end user based on the usage and requirement. There are four types of Cloud deployment models which define the way Cloud can be deployed depending upon various purposes. Diagram of deployment models of Cloud is
1. Private Cloud: It is a Cloud where the infrastructure is built for a single organization and managed by either a single organization or third party at on or off the premise. The motivation to setup a private Cloud within an organization has several aspects. Proper utilization of resources, data privacy and data transfer cost from local to public Cloud are some of the reasons for companies to move for private Cloud.

2. Community Cloud: Several organizations jointly construct and share the same Cloud infrastructure as well as policies, requirements, values, and concerns. The Cloud community forms into a degree of economic scalability and democratizes equilibrium. The Cloud infrastructure could be hosted by a third-party vendor or within one of the organizations in the community.

3. Public Cloud: This is the leading model of current Cloud computing deployment model. The public Cloud is used by the general public Cloud end users and the Cloud service provider is an owner with its defined policies. Examples of public Cloud providers are Amazon EC2 (EC2 (2015)), S3, Google AppEngine (Googleappengine (2015)), and Force.com (Force (2017)).

4. Hybrid Cloud: The combination of two or more Clouds (private, community, or public) is called as hybrid Cloud. Many organizations use the hybrid Cloud model in order to optimize their resources to increase their core capacity by moving onto the Cloud while controlling core activities by keeping it to on-premise through private Cloud.

### 2.1.2 Architecture

Cloud architecture includes the components and sub-components required for Cloud computing. This paradigm has a four layered architecture (Zhang et al. (2010)) as shown in figure 2.2. The figure covers and describes the components required at each layer. Each layer has clearly identified responsibilities. The lowest layer is the hardware layer which is responsible for managing hardware resources like server, switches, power, cooling system etc. Second last is the infrastructure
layer, which is also known as virtualization layer responsible for creating storage and computing resources by dividing the physical resources using virtualization technologies. The infrastructure layer plays an important role since many key features, such as dynamic resource assignments are only available through virtualization technologies (Zhang et al. (2010)). Virtualization of underlying hardware and software is done by creating an image/instance of physical machine called as Virtual Machine. Virtual machine monitor (VMM) is responsible for creation of virtual machine. Third layer from the bottom is the platform layer which consists of operating systems and application frameworks. The upper most forth layer is application layer which consists of the actual Cloud applications (Zhang et al. (2010)). All these layers are loosely coupled i.e. this layered architecture handles the maintenance and management overhead of Cloud computing. The different services like infrastructure, platform and software correspond to these layers in the architecture.
2.1.3 Other Technologies

In this section, we discuss and compare the other technologies like Service Oriented Computing (SOC), Grid Computing and High Performance Computing (HPC) with Cloud.

1. Cloud and Service-Oriented Computing: Many substantial features like encapsulation, componentization, decentralization and integration capability are provided by SOC. It defines the architectural principles, standardized protocols and software specifications to connect computers and devices across the internet (Huhns & Singh (2005)). Actually, the concept of Cloud is more or less based on the development on SOC, specifically in the SaaS. There are many lan-
guages like Web Services Description Language (WSDL) and the REST protocol are very popular interface languages to describe Web services. Similarly, Cloud computing can benefit SOC in several important ways. PaaS model of Cloud can host service-oriented development. For example, Google’s App Engine facilitates a complete development platform in which developers can develop and deploy Java Web services to build their applications. Also, developers of web services could use vast computing resources of public Cloud for the simulation of real-world automated machine requests and network for testing of different services (Dillon et al. (2010)).

2. Cloud and Grid Computing: Grid computing is a distributed computing that takes the hardware and software from distinguish places to process the large task. Both, Cloud and Grid Computing, use resource virtualization. Grid creates resource sharing to form single virtual organization whereas Cloud is owned by single physical organization. Cloud shares physical resources in the form of virtual resource in isolated way while grid targets to provide the maximum computing capacity for a huge task through resource sharing. Cloud aims to suffice as many small-to-medium tasks as possible based on users’ real-time requirements. Therefore, multi-tenancy is a very important concept for Cloud computing.

3. Cloud and High Performance Computing: High-performance computing (HPC) facilitates the solutions to advanced computation problems. HPC has been widely used for scientific tasks whereas Cloud computing targets serving business applications. Key difference of Cloud and HPC lies in elasticity. In HPC, the capacity is often fixed and requires considerable human interaction. For example, HPC application running on cluster requires tuning based on a particular cluster with a fixed number of homogenous computing nodes. Information of physical machines are unknown to end users in Cloud.

### 2.2 Taxonomy and Terminology

This section presents taxonomy associated with Cloud computing covering the hardware components, operating system, virtualization, different
characteristics and technology with the aim to provide conceptual information.

### 2.2.1 Resource Management

Cloud computing is a new popularly adopted technology and as its still growing, there are number of challenges that need to be considered. One of the key challenges faced by Cloud infrastructure providers is the effective and efficient management of physical resources of Cloud infrastructure. Resource management refers to the efficient and effective deployment of computational resources based on the requirement. Many factors needs to be taken into consideration for effective and efficient resource management viz. sharing of physical resources, SLA between service provider and users and different costs of hosted services on virtualized resources. These factors are handled by ability of server consolidation, availability, performance of services and cost management. One of the major considerations in resource management is the optimization of the current allocation of VMs. Therefore, it’s important to identify different efficient allocation policy for optimized resource management.

### Virtualization

Virtualization is a process to create virtual variants of computing resource, such as a server, storage device, network or operating system where the structure partitions the resource into one or more execution environments. It creates virtual machines (VMs) over physical machine based on request for resource by Cloud consumers. Isolation and resource sharing are the two major benefits of virtualization technology (Reubcn (2007)). Resource sharing is the sharing of physical resources available like memory, disk and network devices in shared pool through virtualization. Isolation property provides isolation between multiple virtual machines running on the same physical machine. There are numerous ways to implement a virtualization. There are four main strategies of virtualization viz. full virtualization, para virtualization, operating system-level virtualization and native virtualization (Walters et al. (2008)). Brief overview of all these virtualization strategies is as follow:
– **Full virtualization:** In full virtualization, VMM provides the image of physical machine to the each VM request. In this way, it executes the unmodified OS to execute the privileged instructions. It is successful as it provides pure isolation between the programs executed (Mishra & Jaiswal (2012)). QEMU, Microsoft Hyper-V, Oracle VirtualBox, VMware Workstation, VMware Server, Microsoft Virtual PC are some of the examples of full virtualization.

– **Para virtualization:** In para virtualization, each VM has an abstraction of the hardware that is similar but distinct to the underlying physical hardware. VMs are executed by guest operating systems. Hence, it provides a near-native performance. Methods of para virtualization are still in development and thus, it has limitations of security issues like the guest OS cache data, unauthenticated connections and “guest” OS needs to be modified (Abels et al. (2005)). Disaster recovery, migration, capacity management are the benefits of para-virtualization (Fenn et al. (2009)). Xen (Cherkasova & Gardner (2005)) and KVM (Barham et al. (2003)) are few examples of para-virtualization.

– **OS-level Virtualization:** Contrasting to para-virtualization and full virtualization, OS-level virtualization does not depend on a hypervisor. In this technique, it modifies the operating system securely to isolate multiple instances of an operating system within a host machine. Performance up to near-native speeds is the advantage to OS-level virtualization. No hypervisor/instruction is required separately. OpenVZ and LXC are few examples of OS-level virtualization.

– **Native Virtualization:** Multiple unmodified operating systems are allowed to run along side one another in this technique and operating systems are capable of running on the host processor directly. That is, in native virtualization, it does not emulate a processor. This virtualization gives full processor to unmodified OS, which results into a poor performance. Intel and AMD supports virtualization through the Intel-VT and AMD-V virtualization extensions (Walters et al. (2008)) are few example of this technique.
Virtual Machine

An image or an instance of physical machine is called a virtual machine (VM). Virtualization makes it possible to run multiple computing resources (in form of VMs) on the same (physical) server at same time, concurrently. Virtual Machine Monitors (VMM) also commonly known as hypervisors create and provide VM abstraction.

Virtual Machine Monitor (Hypervisor)

Hypervisors, also known as Virtual Machine Monitors (VMM) create, manage and run virtual machines on physical machines. There are many open source hypervisors available viz. KVM, Xen, OpenVZ, VirtualBox, Lguest, LXD etc. There are three types of VMM/hypervisor viz. type 2 VMM, type 1 VMM and hybrid VMM as shown in fig 2.3.

![Figure 2.3: Hypervisor (Younge et al. (2011))](image)

The type 2 VMM runs inside the host operating system. All guest environments are operated above the level of host OS. Java Virtual Machine and Microsoft’s Common Language Runtime are examples of these guest environments. The hybrid model is used to implement Virtual PC, Virtual Server and VM Ware GSX. Control is with host operating system that shares control of the hardware with the VMM. The type 1 VMM controls the guest operating systems instance executes on hardware at an upper level of it. Xen and Windows Server virtualization are the examples of type 1 hypervisor.
Host/Server

A physical machine on which requests are deployed through virtual machine is called host/server in data center. Many methods are available to create VM on physical machine/server. Server consolidation is also an important process carried out by various procedures for efficient use of machine and power. In our research, we use the terms host, server or node interchangeably.

Task/Job

Task/job is a small piece of work or a part of process carried out on virtual computing resource. An effective and efficient task scheduling is required for efficient resource utilization. Numerous techniques are available for task scheduling that emphasize on various parameters of Cloud datacenter such as resource utilization, response time, service level agreement (SLA) violation, makespan, power consumption etc. Scheduling in Cloud environment generally categories into three groups: resource scheduling, workflow scheduling and task scheduling. Task scheduling methods further categorizes into two sub-groups: centralized and distributed. In centralized scheduling, single scheduler does all mappings whereas in distributed scheduling, the entire process is partitioned among different schedulers. In case of distributed scheduling, it has high implementation complexity whereas centralized scheduling is easy to implement (Mathew et al. (2014)). Figure 2.4 summarizes classification of scheduling methods.

Datacenter

Data center in Cloud is a shared pool of physical resources including processing elements, memories, storage and network components like routers and switches where thousands of servers are arranged in racks and interconnected through switches, routers or other communication channels. Typical issues at datacenter level are for hardware covering configuration of components, traffic management, fault-tolerance, power and cooling of equipment and efficient resource management (Zhang et al. (2010)).
Energy Efficiency

Improving energy efficiency is one of the major issues in Cloud computing and hence energy management of Cloud data center is important. It has been identified that the cost of powering and cooling is 53% of the total operational expenditure of data centers (Hamilton (2009)). In 2006, data centers in the US consumed more than 1.5% of the total energy generated in that year, and the percentage is projected to grow 18% annually (Li et al. (2009)). Hence, infrastructure providers are under enormous pressure to reduce energy consumption. The goal is not only to cut down energy cost in datacenters, but also to meet government regulations and environmental standards. Designing energy-efficient data centers has recently received considerable attention. Cloud computing naturally leads to energy-efficiency by providing the following characteristics:

- Economy of scale due to elimination of redundancies.
- Improved utilization of the resources.
- Location independent VMs can be moved to a place where energy is cheaper.
- Scaling up and down resource usage can be adjusted to current
requirements.
– Efficient resource management by the Cloud provider.

**Virtual Machine Migration**

Virtualization offers significant benefits in Cloud computing by enabling virtual machine migration. It is a useful technology for migrating operating system instances across multiple physical machines. It is used to cover many objectives like fault management, load balancing, low-level system maintenance and reduce energy consumption. In addition, virtual machine migration enables robust and highly responsive provisioning in data centers. It has been developed from process migration techniques (Zhang et al. (2010)). Many open source hypervisors, like Xen (Barham et al. (2003)) and VMWare (Chiueh & Brook (2005)) have implemented “live” migration of VMs with very negligible downtime around ten milliseconds to one second of applications. Live migration allows VMs to be migrated without considering downtime. The transfer of a VM actually talks about the transfer of its state. This includes memory, internal state of the devices and virtual CPU. Between all these parameters, the most time-consuming one is the memory transfer. There are two important parameters which need to be measured while performing live VM migration are:

1. **Down time** - Down time refers to the time for which the VM service is not available due to migration.
2. **Migration Time** - Migration Time refers to the total amount of time required to transfer a virtual machine from one physical node to another without affecting its availability.

There are two types of VM Migration methods:

1. **Hot (live) migration** - Virtual machine remains in running state, while migrating and does not lose its status information. In this type, the state of a virtual machine to migrate is transferred first. The state of VM consists of its local file system and memory contents.
2. **Cold (non-live) migration** - The status of the VM loses and user can notice the service interruption. In this type, initially VM is
suspended and its state is transferred, after all transfer, VM is resumed at destination host.

Performance

Performance considerations are necessary for the overall success of Cloud computing. Performance criteria includes the optimum cost of Cloud services, reliability and scalability. For each service of Cloud, there are different performance measurements. For SaaS, performance measures are business transaction response times, throughput, technical service reliability, availability and scalability of the applications. In PaaS, performance measures are indirectly perceived by users and defined as technical transaction response times, throughput, technical service reliability, availability, and scalability of the middleware. In IaaS, performance measures are defined as infrastructure performance, capacity, reliability, availability, and scalability.

Service Level Agreement (SLA)

An SLA is a mutual agreement between the Cloud service provider and the consumer of different services of Cloud on various Quality of Service (QoS) parameters or level of service to be delivered. Different attributes of QoS are performance, which covers response time and throughput, availability and security (Duijvestijn et al. (2010)). Number of execution problems does exist for the Cloud service providers such as efficient methods/algorithms to keep precise and regular variable real time information of the resource usage and smart algorithms to reject some resource request when SLA does not match. Even more advanced SLA algorithms are also required to incorporate the consumers’ feedback and more appropriate calculation mechanisms (Dillon et al. (2010)).

Power and Performance

Power consumption is considered as an important parameter in Cloud because expenditure on powering and cooling is 53% of the total operational expenditure of data centers (Hamilton (2009)).
has been also noticed that in 2006, the cost of data centers in the US consumed more than 1.5% of the total energy generated in that year, and the percentage is projected to grow 18% yearly Duijvestijn et al. (2010). Hence, an infrastructure provider needs to think on reduction of energy consumption. Designing energy-efficient data centers has gain significant attention. Also, one of the important requirements for a Cloud computing environment is to provide consistent QoS, that can be defined in terms of SLA that describing different characteristics such as minimal throughput, maximal response time or latency delivered by the deployed system. Although modern virtualization technologies can ensure performance isolation between VMs sharing the same physical computing node, due to aggressive consolidation and variability of the workload, some VMs may not get the required amount of resource when requested. This leads to performance loss in terms of increased response time, timeouts or failures in the worst case. Sometime, focusing on performance may increase energy consumption. In some of the cases when energy consumption is considered, underloaded machines are targeted to switch off by transferring VM on physical machine to another node, may lead to performance degradation. Therefore, Cloud providers have to deal with energy performance trade-off. There are many conventional methods introduced by researchers trying to handle and reduce this trade-off (Beloglazov & Buyya (2010a,b), Dong et al. (2013), Buyya et al. (2010), Beloglazov & Buyya (2012)).

The main focus of this research work is on energy-efficient resource management strategies that can be applied on a virtualized data center by a Cloud provider (e.g. Amazon EC2). VM migration is considered for our work. The ability to migrate VMs between physical hosts with low overhead gives flexibility to a resource provider as VMs can be dynamically reallocated according to current resource requirements and the allocation policy. Idle physical nodes can be switched off to minimize energy consumption. In this thesis, we present a decentralized architecture of the resource management system for Cloud data centers and propose the development of the following policies for continuous optimization of VM placement:

- Optimization over multiple system resources - at each time

frame, VMs are reallocated according to current CPU, RAM and network bandwidth utilization.

- Network optimization - optimization of virtual network topologies created by intercommunicating VMs. Network communication between VMs should be observed and considered in reallocation decisions in order to reduce data transfer overhead and network devices load.

- Thermal optimization - current temperature of physical nodes is considered in reallocation decisions. The aim is to avoid “hot spots” by reducing workload of the overheated nodes and thus decrease error-proneness and cooling system load.

### 2.2.2 Storage Technologies and Data Management

In Cloud environments, applications lay in the huge scalable data centers where physical computing resources are dynamically provided and shared to achieve noticeable scaling. Data management and storage capacity needs to be emphasized to create the maximum benefits. To overcome the problem of the traditional storage technology, big data on Hadoop and data warehouse integration platform are designed, which ensured the effectiveness and ease of the management and usage of data. These frameworks typically use Internet-scale file systems such as GFS and HDFS Software frameworks such as MapReduce and its various implementations such as Hadoop (Zhang et al. (2010)).

### 2.3 Issues and Challenges in Cloud

Though Cloud computing has been come up with successful paradigm and extensively accepted by the industries, the research on Cloud computing is in progressive stage. Many existing challenging issues are under research umbrella and have not been fully addressed, while new challenges keep rising from different services. In this section, we briefly address some of the research issues and challenges of Cloud environment.
2.3.1 Security and Privacy

Security and privacy play very vital role in Cloud. While using services of Cloud, end users store their data on Cloud and deploy their services on the Cloud infrastructure. In this scenario, as the users lose their direct control over their data, security and privacy become the most essential. Multi-tenancy model of Cloud creates the threat of stealing of data and privacy information through shared channel Dillon et al. (2010). In general, service providers are not having any control over infrastructure. They rely on infrastructure provider in case of security and hence, infrastructure provider must consider the important security parameters like confidentiality, for secure data access and transfer, and auditability, for verifying whether security setting of applications has been tampered or not. Cryptographic protocols are used to achieve confidentiality, whereas auditability can be achieved using remote verification techniques. At each and every architectural layer of Cloud, more secure algorithm needs to be implemented (Zhang et al. (2010)). Also, virtualization and VM migration technologies required to be fully secure (Zhang et al. (2010)).

2.3.2 Interoperability and Portability

The Cloud computing system is large, where many providers offering a large variety of Cloud services. When many providers and many services are available to end users, two terms, interoperability and portability play significant role for success of technology. The first term, interoperability means a measurement of the degree to which heterogeneous systems or components can successfully work together (Lewis (2013)). More formally, IEEE and ISO define interoperability as the ability for two or more heterogeneous systems or applications to mutually exchange information and use the information that has been exchanged. In case of Cloud computing, interoperability should be seen as the capability of different applications of public, private and other different Cloud systems to understand each other’s different characteristics like configuration, applications, service interfaces data formats etc. for proper commu-
nication and co-operations. Second term, portability can be defined as ability to move an entity of Cloud from one system to another so that it is usable on the target system. Cloud portability enables the migration of Cloud services from one Cloud provider to another or between a different Cloud models. There are numerous reasons for consumers to seek for the Cloud portability like increase in price by current service provider, broken service-level agreement or shifting to new provider for to fulfill business need. Cloud portability needs interoperability among Cloud providers, which means that one Cloud provider must be able to replicate the application environment that the previous Cloud provider had established for the service. A huge research on these two is going on. Many vendors are working together to provide interoperability and portability for consumers. There are many challenges associated with Cloud computing. In general, the interfaces and APIs of Cloud services are not standardized and different providers use different APIs. Data portability is a major consideration in moving from one Cloud service to another Cloud service (Lewis (2013)).

2.3.3 Reliability and Availability

A key benefit of Cloud computing is in offering the “pay-per-use” model for renting resources where running own services. As a concern, users will expect to have certain assurances of the service level to be provided once their applications run in the Cloud system. These expectations include the reliability and availability of the service. On these characteristics, service consumer/users and provider mutually agree on agreements referred to as Service Level Agreement (SLA). Reliability property refers to consistency in service of Cloud whereas, availability refers to presence of service when required by consumer (Sabahi (2012a)).

2.3.4 Traffic Management and Analysis

Analysis of data traffic is important for data centers. Traffic management and its analysis are used by many web applications to optimize performance and customer’s experiences. It is also use-
ful to network operators to make many of the management and planning decisions. There are several challenges for existing traffic measurement, load balancing and analysis methods. More heuristic methods and algorithms required to handle traffic and its analysis efficiently.

2.4 Approaches Addressing Energy Efficiency in Cloud

A rapid growth Cloud computing paradigm makes a challenge for Cloud providers to maintain the operational costs of the data centers. In these considerations, energy-efficiency of the servers plays an important role, as they include the electrical and the cooling components costs which found a key part of the total cost involved in the data center. There is noticeable energy consumption at different level of Cloud as shown in figure 2.5. Power-savings can be achieved at numerous components levels in a computer system. These components are processors, memory, devices, and I/O devices. Very well-known technologies like DVFS (Dynamic Voltage and Frequency Scaling) can be used at the processors level or use of P-state when CPU is running and P-sleep when CPU is idle. To save power at memory level, standards such as DDR3 have used to put idle memory banks into low-power states. Also, at devices level, individual devices can be put into low-power states, controlled and coordinated by a run-time power management framework in the operating system (Dharwar et al. (2012)). Many researchers have been working to identify the suitable strategy of VM allocation and server consolidation to manage energy consumption in Cloud. Xie et al. (2013) have proposed a heuristic algorithm to allocate VMs in the increasing order of their starting time. There is a set of servers with the enough resources, for each VM, this server is selected in such a way that the energy cost can be reduced to the minimum. Beloglazov & Buyya (2010b) have proposed and evaluated heuristic algorithm for dynamic reallocation of VMs to minimize energy consumption while providing reliable Quality of Service. The allocation of VMs can be divided in two parts: the first part is admission of
new requests for VM provisioning and placement of VMs on hosts and the second part is optimization of current allocation of VMs. For the first part, it uses modification of the Best Fit Decreasing (BFD) algorithm. Some studies (Verma et al. (2008), Buyya et al. (2010), Cardosa et al. (2009)) have proposed a Virtual machine allocation technique to reduce energy consumption by improving physical server utilization and reducing the number of hosts. More research has been carried out by Beloglazov & Buyya (2010a,b), Dong et al. (2013), Buyya et al. (2010), Beloglazov & Buyya (2012) on efficient methods of virtual machine allocation to minimize the energy consumption. We will discuss it in the next section where again the performance based energy efficient allocation of VM will be covered.

2.4.1 Introduction to Multi-Objective Optimization Technique

Multi-objective optimization is a mathematical method of systematic and simultaneous optimization of objectives’ set. Unlike the single objective optimization, multi-objective optimization has a set of non-dominated solutions considering all objectives known as pareto optimal solutions (Zitzler & Thiele (1998)). This pareto optimal outcome is called pareto front(Zitzler & Thiele (1998)). A simple single-objective optimization problem can be formulated as \( \min f(x) \), where \( x \in S \), where \( S \) is a set of constraints, whereas,
multi-objective optimization problem can be formulated as: \( \min f_1(x), f_2(x), f_3(x), ..., f_n(x) \) where \( x \in S \). There are four categories of methods in multi-objective optimization (Marler & Arora (2004)).

- **Methods with apriori articulation of preferences**: Preferences are articulated by Decision Maker (DM) or user for different objective functions. This technique is used when the priority of objectives is articulated. The user needs to give the preference/priority to obtain the desirable solution.

- **Methods with a posteriori articulation of preferences**: User selects the efficient solution from set of solutions and accordingly the preference of the objectives are finalized. This method is used when the user is aware of set of non-dominated solutions.

- **Methods with no articulation of preferences**: The DM cannot concretely define the preferences all the time. No preference is required in this method. It is a simplification of the apriori articulation of preference method. It performs operations directly on objectives.

- **Progressive articulation of preferences or Interactive method**: The DM needs to give the preferences at every fixed iteration. DM needs to be careful with the preference of objective to obtain an optimal solution.

Hierarchy shown in figure 2.6 summarizes the different methods of Multi-objective optimization. Selection of the method depends on the type of problem. Their comparisons are shown in table 2.1. Based on this comparison, a method of priory articulation of preference is identified as suitable for our work. This is because of the fact that the user is not aware about the possible solution/result, but can predict/give preferences to non-dominated objectives to generate pareto front. To do so, we prefer to use a priory articulation of preferences. Further, comparison on different methods of prior articulation of preference are shown in table 5.2.

Based on this comparison, we found weighted sum method of apriori articulation of preferences suitable for our problem, as accuracy in weight selection generates pareto optimal set.

In this method, prior weightages are given to different objectives. These weightages are actually a service providers’ priority to objec-
Figure 2.6: Different methods of Multi-objective optimization

tives as mention in table 2.2.
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Key characteristic</th>
<th>Solution Identified/Goal specified</th>
<th>Preference of objective</th>
<th>Solution Identified/Goal specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>A priori articulation of preferences</td>
<td>User indicates the relative importance of the objective functions or desired goals before running the optimization algorithm.</td>
<td>No required</td>
<td>No</td>
<td>Not required</td>
</tr>
<tr>
<td>A posteriori articulation of preferences</td>
<td>Which entails electing single solution from a set of mathematically equivalent solutions</td>
<td>Not required</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Progressive articulation of preferences</td>
<td>Decision-maker needs to continuously provide input during the running of the algorithm</td>
<td>Required continuously</td>
<td>Not required</td>
<td>Not required</td>
</tr>
<tr>
<td>No Articulation of Preferences</td>
<td>No preferences required</td>
<td>Not required</td>
<td>Not required</td>
<td>Not required</td>
</tr>
<tr>
<td>Method Name</td>
<td>Key characteristic</td>
<td>Preference articulation</td>
<td>Advantages</td>
<td>Disadvantages</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Weighted Global Criterion</td>
<td>All objective functions are combined to form a single function</td>
<td>Method parameters are used to define preferences</td>
<td>It can give optimal pareto set in accurately defined preferences.</td>
<td>The fixed value of power will limit prediction of the calculation on weight/preference.</td>
</tr>
<tr>
<td>Weighted Sum</td>
<td>All objective are combined to form a single function with weight attached with each objective.</td>
<td>A very less preference information required. User does not provide extensive input.</td>
<td>Easy and simple to implement. Accuracy in weight selection will give pareto optimal set.</td>
<td>Weights of objective function needs to be identified accurately otherwise pareto optimal points and set cannot be obtained.</td>
</tr>
<tr>
<td>Lexicographic</td>
<td>Objective functions are arranged in order of importance and afterwards the objectives are minimized.</td>
<td>Clear</td>
<td>Straight forward method</td>
<td>Computational expense increase as multiple problems solve individually.</td>
</tr>
<tr>
<td>Weighted Min-Max</td>
<td>The method is about to minimization and maximization of the objectives by introducing constraint.</td>
<td>Weightage is assigned to overall function. It is predefined.</td>
<td>It provides the complete Pareto optimal set by variation in the weights.</td>
<td>Number of constrain can increase the number of complexity</td>
</tr>
</tbody>
</table>
Table 2.2: Comparison of different methods of apriori articulation of preferences (continue)

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Key characteristic</th>
<th>Preference articulation</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Code complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential Weighted</td>
<td>Exponential is introduced in weighted sum method.</td>
<td>Predefined</td>
<td>Give pareto optimal set</td>
<td>Large values of parameters used in method may lead to numerical overflow.</td>
<td>Complex</td>
</tr>
<tr>
<td>Weighted Product</td>
<td>Weight is apply as a power of objective function.</td>
<td>Unclear</td>
<td>A functions with different significance are handled.</td>
<td>As the characteristics of the weights are unclear cannot obtain efficient pareto optimal set.</td>
<td>Average</td>
</tr>
<tr>
<td>Goal Programming</td>
<td>Goals are specified for each objectives.</td>
<td>Goals are clear.</td>
<td>It has wide range of application as it achieves/optimize goals one by one.</td>
<td>There is no guarantee of a Pareto optimal solution. Cannot handle larger objectives.</td>
<td>Average</td>
</tr>
<tr>
<td>Bounded Objective Function</td>
<td>Minimizes the single most important objective function, all other objectives are used to create additional constraints for objective function.</td>
<td>Not required, rather it sets limit on the objectives.</td>
<td>Consistent variation in null parameters, may obtain the complete Pareto optimal set.</td>
<td>Selection of parameter to get feasible region is complex.</td>
<td>Average</td>
</tr>
<tr>
<td>Physical Programming</td>
<td>It creates utility function based on objective functions, constraints, and goals</td>
<td>Clear.</td>
<td>Customize a more complex and accurate individual utility function for each objective.</td>
<td>Significant knowledge of objectives, constraint and goals required.</td>
<td>Complicated</td>
</tr>
</tbody>
</table>
2.5 Introduction to Fuzzy-based Approach

Fuzzy logic is an approach to computing where “degree of truth” is considered rather than any fix value like true-false represented by 0 and 1. Few important terms used in fuzzy set theory are defined as follow:

- Linguistic Variables: Linguistic variables are the defined input or output variables of the system having self-explanatory words or sentences from natural language, instead of numerical values.
- Fuzzification: It is the process of converting a crisp input data to fuzzy values.

This conversion is relying on membership function.

2.5.1 Membership Functions

Membership function maps the non-fuzzy input values to a fuzzy values. It is define as $\mu_A: x \rightarrow [0,1]$. where, $x$ is non-fuzzy input values.

The different forms of membership functions are available like triangular, Gaussian, singleton, trapezoidal, or piecewise linear as shown in figures 2.7, 2.8, 2.9, 2.10. Triangular, trapezoidal, and Gaussian shapes are the most common categories of membership functions (Mendel (1995)).

2.5.2 Fuzzy Rules

A fuzzy rule is a simple IF-THEN rule with a condition to derive conclusion. Fuzzy rules define output of the system.

2.5.3 Fuzzy Set Operations

Fuzzy set operations are used to evaluate fuzzy rules and combine the results of individual rules. These operations are different than the one apply on non-fuzzy set.
2.5.4 Inference

The process of combining a result of each rule to obtained final result is called an Inference. There are different ways to combine the result of different rules like maximum, bounded sum, normalized sum.

2.5.5 Defuzzification

Defuzzification is the process of converting the fuzzy results obtained by inference to crisp output. Defuzzification depends on the membership function of output variable. There are many algorithms like Center of Gravity, Center of Gravity for Singletons, Left Most Maximum and Right Most Maximum for defuzzification. Fuzzy logic system is defined by figure 2.11. fuzzification and defuzzification depends on a membership function while inference engine uses different method to perform operation on rules. Hierarchy of fuzzy inference methods is shown in figure 2.12. Mamdani and sugeno are the most popular method of direct type.
Figure 2.8: Trapezoidal function

Figure 2.9: Singleton function
Figure 2.10: gaussian function

Figure 2.11: Fuzzy logic system

Figure 2.12: Categories of fuzzy inference method