SOCCER: Self-Optimization of Energy-efficient Cloud Resources

Sukhpal Singh[1], Inderveer Chana[2], Maninder Singh[3] and Rajkumar Buyya[4]

[1,2,3] *Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India-147004*

[4]*CLOUDS Lab, Department of Computing and Information Systems, The University of Melbourne, Australia*

[1]ssgill@thapar.edu, [2]inderveer@thapar.edu, [3]*msingh@thapar.edu*, [4]rbuyya@unimelb.edu.au

Abstract

Cloud data centers often schedule heterogeneous workloads without considering energy consumption and carbon emission aspects. Tremendous amount of energy consumption leads to high operational costs and reduces return on investment and contributes towards carbon footprints to the environment. Therefore, there is need of energy-aware cloud based system which schedules computing resources automatically by considering energy consumption as an important parameter. In this paper, energy efficient autonomic cloud system (SOCCER) is proposed for energy efficient scheduling of cloud resources in data centers. The proposed work considers energy as a QoS parameter and automatically optimizes the efficiency of cloud resources by reducing energy consumption. The performance of the proposed system has been evaluated in real cloud environment and the experimental results show that the proposed system performs better in terms of energy consumption of cloud resources and utilizes these resources optimally.

KEYWORDS - Cloud computing, Energy, Virtualization, Resource Scheduling, Autonomic Computing, Self-optimization, Green Computing, Heterogeneous Workloads, Aneka Cloud Application Platform

1. Introduction

Cloud offers three types of services (Infrastructure, Platform and Software) using pay per use model. Data centers are the backbone of the modern economy; from the server rooms that power small-sized to medium-sized organizations to the enterprise data centers that support corporations and the server farms that run cloud computing services hosted by Amazon, Facebook, Google, and others [1]. However, the explosion of digital content, big data, e-commerce, and Internet traffic is also making data centers one of the fastest-growing consumers of electricity in developed countries. Further, cloud data centers, provide effective and reliable infrastructure services to the end users [2]. Presently, customer satisfaction and performance is increased by the data centers without considering energy consumption in those datacenters which leads to high operational cost and reduces return on investment. Many governments have also imposed constraints to reduce the carbon footprints which effects environment. IT companies (Microsoft, Google, Amazon, IBM etc.) are increasing their data centers every year to provide services to the cloud users in a better way [3] [4]. Due to large energy consumption, temperature increases gradually which leads to the failure of the system and violates the Service Level Agreement (SLA). Literature reports that data center infrastructure generates over 70% of total heat generated [5]. Another reason of wastage of energy is resources are running in idle or underutilized state. Energy efficient resource scheduling in cloud is a challenging job and the scheduling of appropriate resources to cloud workloads depends on the QoS requirements of cloud applications and energy consumption of computing resources [6]. Energy saving in case of heterogeneous cloud workloads is very difficult to improve. Therefore, there is need of cloud based system which schedules computing resources automatically by considering energy consumption as a significant aspect.

Scheduling of resources in cloud is an important part of resource management system. Mapping of cloud workloads to appropriate resources is mandatory to improve QoS parameters like time, cost, energy consumption etc. In our earlier work [1-4], we have identified various research issues related to QoS and SLA for cloud resource scheduling and have developed a QoS based resource provisioning technique (Q-aware) to map the resources to the workloads based on user requirements described in the form of SLA. Further, QoS based Resource Scheduling Framework (QRSF) has been proposed, in which provisioned resources have been scheduled by using different resource scheduling policies (cost, time, cost-time and bargaining based). Based on QoS requirements, scheduler finds and maps the resources and workloads. Resource scheduling in previous work [2] [3] [4] has been done in following steps: i) understand the expectations and requirements of cloud user, ii) analyze and cluster the workloads through *k*-means clustering algorithm, iii) find the required number of resources, iv) map the resources and workloads and v) schedule and execute the workload on appropriate resources with minimum time and cost. QRSF framework executes the workloads without self-optimization of resources. To incorporate self-optimization, QRSF has been further extended by proposing Energy-aware Autonomic Resource scheduling TecHnique (EARTH) [1], in which IBM's autonomic computing concept has been used to schedule the resources automatically by optimizing energy consumption where user can easily interact with the system using available user interface. But EARTH can execute only homogenous cloud workloads and the complexity of resource scheduling in EARTH increases with the increase of number of workloads. To address this issue, proposed system (SOCCER) clusters the heterogeneous cloud workloads and executes them with minimum energy consumption.

The motivation of this paper is to design energy efficient autonomic cloud computing system called **SOCCER** ***(S**elf **O**ptimization of **C**loud **C**omputing **E**nergy-efficient **R**esources)* for effective scheduling of resources which considers energy consumption as a QoS parameter. The main aim of this research work is: i) to propose an autonomic resource management technique for execution of heterogeneous workloads by considering generic property of self-management, ii) to optimize the energy consumption and iii) to implement and perform evaluation in a real cloud environment for clustered heterogeneous workloads. The rest of the paper is organized as follows. Section 2 presents related work of existing energy-efficient systems. Proposed system is presented in Section 3. Section 4 describes the experimental setup and presents the results of evaluation. Section 5 presents conclusions and future scope.

2. State-of-the-art

Scheduling of resources in cloud has been done through different techniques as reported by literature but efficiency saving is an important factor that is difficult to optimize automatically. To reduce energy consumption, researchers proposed the concept of VM (Virtual Machine) consolidation to detect overload, under-load and VM selection [5] [6] [25]. Bin et al. [16] described the overview of data center networks for cloud computing. Further, detailed descriptions of virtualized infrastructure, physical architecture and Dynamic Circuit Network (DCN) routing have been discussed.

Mohsen et al. [7] proposed Preemption-Aware Energy Efficient (PAEE) technique for virtualized datacenters which adjusts the energy consumption based on user performance requirements and reduces SLA violations. PAEE reduces energy consumption up to 18% but it considers only homogenous workloads. Shaolei et al. [8] proposed provably-efficient online scheduling algorithm for geographically distributed datacenters which optimizes the energy cost and fairness among different organizations subject to queueing delay constraints and it reduces energy consumption which is closed to optimal offline algorithm. Steven et al. [9] developed mechanisms to better utilize installed power infrastructure, reducing reserve capacity margins and avoiding performance throttling. Further, it reduces reserve capacity requirements to tolerate a single power distribution unit (PDU) failure. In addition, power routing is proposed to schedule workload dynamically across different servers. Rahul et al. [10] proposed an Lyapunov optimization technique based online control algorithm that can optimally exploit these devices to minimize the time average cost. It operates without any knowledge of the statistics of the workload or electricity cost processes, making it attractive in the presence of workload and pricing uncertainties. Siqian et al. [11] formulated several stochastic optimization models for trading off between energy footprints and QoS associated with server consolidation in cloud computing data centers. Further, they consider finite service times with uncertain workloads at each period and minimize the expected energy consumption.

Ying et al. [12] described DVS (Dynamic Voltage Scaling) based energy aware technique to execute workloads with minimum execution time and energy consumption. Fitness function is defined based on methods of double and unify fitness and genetic algorithm is used to identify the resources with minimum energy consumption. Yan et al. [13] described control dependence graph based energy aware resource scheduling technique to execute the HPC applications in distributed environment within deadline with least energy consumption. Design approximation and traditional multiprocessor scheduling algorithms are extended to formulate the problem after analysis and completion of worst-case performance. Further based on energy consumption and desired deadline of tasks, pricing scheme is designed for their execution Nakku et al. [14] proposed Energy Credit Scheduler [ECS] used to estimate the consumption of power in VM based on the number of workloads executed on VM. Scheduling algorithm for virtual environment is designed based on this estimation model to execute the tasks on computing resources based on minimum energy consumption and budget and implemented in Xen virtualization system and it reduces energy consumption with minimum error rate. Changbing et al. [15] proposed holistic workload based resource scheduling policy for geographical distributed data centers to improve energy efficiency and MinBrown (workload scheduling technique) is designed and consider constraints like availability of green energy and cooling power.

Lin et al. [17] proposed virtualization based unified optimization framework to improve the energy efficiency of data center networks by using the concept of multipath routing protocol and hierarchical feature of the topology. Lin et al. [18] proposed an efficient energy saving technique for data center networks by scheduling and routing “deadline-constrained flows” where the transmission of every flow has to be accomplished before a rigorous deadline, being the most critical requirement in production data center networks. Zhaogang et al. [19] proposed architecture of Cloud-integrated Cyber-Physical Systems to execute complex industrial application in a controlled manner. Lin et al. [20] proposed an energy efficient framework (GreenDCN) by assigning virtual machines to servers to reduce the amount of traffic and to generate favorable conditions for traffic engineering. Further, GreenDCN reduces the number of active switches and balance traffic flows which improves energy efficient of data center network.

Marco et al. [21] explored the benefit of electricity price variations across time and locations and discussed the problem of scheduling batch jobs to multiple geographically-distributed data centers. Further, provably-efficient online scheduling algorithm (GreFar) is proposed to optimize the energy cost and fairness among different organizations subject to queueing delay constraints. Qiang et al. [22] proposed industrial cluster oriented Cloud Manufacturing Service System (CMSS) in order to fulfill the real time designing and manufacturing information interaction among the collaborative partners in an industrial cluster area. Further, lightweight application of CMSS is designed to analyses the services using virtual cloud environment. Lena et al. [23] designed an auction-based online mechanism for VM provisioning, allocation, and pricing in clouds that considers several types of resources. Further, it allocates VM instances to selected users for the period they are requested for, and ensures that the users will continue using their VM instances for the entire requested period. Dan et al. [24] studied traffic-aware resource provisioning mechanisms for distributed clouds and examines cloud traffic characteristics and optimizations produced fine-grained traffic-awareness approaches that can more efficiently reduce energy costs for distributed clouds with dynamic, diverse traffic. SOCCER (Self Optimization of Cloud Computing Energy-efficient Resources) has been compared with existing frameworks as described in Table 1.

Table 1: Comparison of SOCCER with Existing Frameworks

Framework	Mechanism	Workload Type	Clustering of Workloads
PAEE [7]	Non-Autonomic	Homogenous	×
ECS [14]	Non-Autonomic	Homogenous	×
GreenDCN [20]	Non-Autonomic	Homogenous	×
GreFar [21]	Non-Autonomic	Homogenous	×
CMSS [22]	Non-Autonomic	Homogenous	×
SOCCER	Autonomic	Homogenous and Heterogeneous	√

None of the existing research work considers heterogeneous cloud workloads, clustering of workloads and autonomic management of resources. In addition; SOCCER needs to consider the basic features of cloud computing in order to execute the heterogeneous cloud workloads automatically with minimum energy consumption and maximum energy efficiency.

2.1 Our Previous Contributions

To incorporate self-optimization, QRSF [4] has been further extended by proposing Energy-aware Autonomic Resource scheduling TecHnique (EARTH) [1], in which IBM’s autonomic computing concept has been used to schedule the resources automatically by optimizing energy consumption where user can easily interact with the system using available user interface. Architecture of EARTH is shown in Figure 1.

2.2. EARTH Execution

EARTH used fuzzy logic system to process the data in an effective way [1]. Fuzzy inputs include Workload Waiting Time (WWT), Workload Execution Time (WET) and Resource Energy Consumption (REC) and fuzzy output is Workload Processing Priority (WPP). All the four variables are changing continuously due to this, and hence these variables are considered. Based on this, fuzzy rule set is created to define the behavior of fuzzy system and setting the relationship among inputs and outputs. Three membership functions are considered for every input and output variables: Low, Medium and High. Based on these input and output variable, inference engine is making decisions. Value of membership functions can be changed based on the requirements and conditions of every workload. After the inputs and outputs of a fuzzy system are selected, they must be partitioned into appropriate conceptual categories. Based on selected inputs and outputs of the fuzzy system, *member functions* are created for better representation of relationship among input and output variables. Each of these categories actually represents a fuzzy set on a given input or output domain. WWT, WET and REC are antecedents and WPP is consequent. Three membership functions consider for every three inputs. *Fuzzification* is used to find the degree of truth for every rule, membership function defined on every input variable is applied to their actual value. We used most popular operator "AND" operator for fuzzy implementation. This function returns the lowest value of among these values entered. *Defuzzification* is used to convert the value of fuzzy output into crisp output value. We used MAXIMUM method in this research work for *Defuzzification*. Note: Detailed execution of EARTH is described in our previous research work [1].

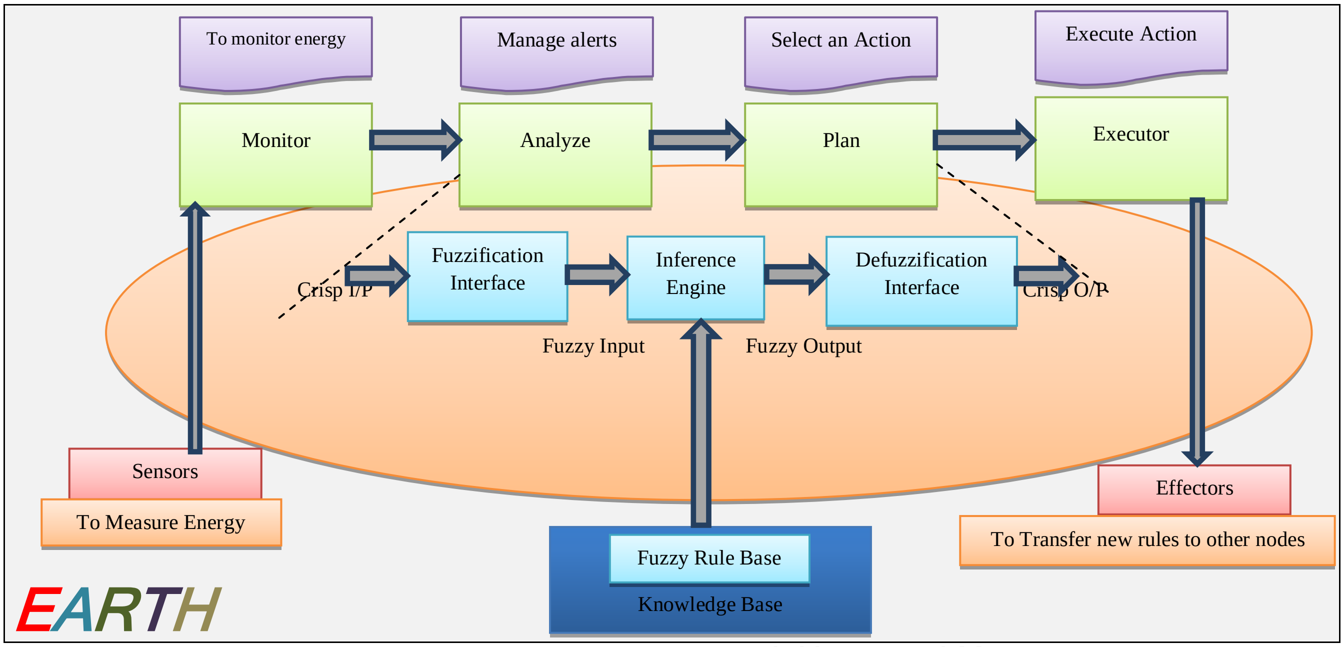

Figure 1: Energy-aware Autonomic Resource Scheduling Framework [1]

But EARTH can execute only homogenous cloud workloads and the complexity of resource scheduling in EARTH increases with the increase of number of workloads. To address this issue, proposed system (SOCCER) clusters the heterogeneous cloud workloads and executes them with minimum energy consumption.

3. SOCCER: Self-Optimization of Cloud Computing Energy-efficient Resources

SOCCER focuses on energy efficient scheduling of computing resources in virtual data centers for execution of both homogenous and heterogeneous cloud workloads. SOCCER focuses on autonomic execution of clustered heterogeneous cloud workloads in order to improve energy efficiency. Finally, we have validated SOCCER using real cloud environment and measured the value of energy consumption of different clusters of workloads. This research work mainly focuses on one important aspect of self-optimization i.e. energy consumption.

3.1 Objective Function

The goal of cloud provider is to minimize the actual energy consumption. The cloud workload will be executed only when the Actual Energy Consumption denoted as EnC_{actual} is less than the threshold value of energy consumption (E_t). For a particular cloud workload, the information on its energy consumption and processor utilization can be used to measure the energy consumption of resources for execution of heterogeneous cloud workloads.

Energy Consumption is calculated using ([Eq. 1]-[Eq. 5]). The energy model is devised on the basis that resource utilization has a linear relationship with energy consumption [1] [2]. Energy Consumption (EnC) of using resources can be expressed as the following formula [Eq. 1]:

$$EnC = EnC_{Datacenter} + EnC_{Transceivers} + EnC_{Memory} + EnC_{Extra} \qquad (1)$$

$EnC_{Datacenter}$ represents the datacenter's energy consumption, $EnC_{Transceivers}$ represents the energy consumption of all the switching equipment. EnC_{Memory} represents the energy consumption of the storage device. EnC_{Extra}

represents the energy consumption of other parts, including the fans, the current conversion loss and others. The above formula can be further disintegrated; a cloud computing environment with d datacenters, t' transceivers equipment and a centralized memory device, its energy consumption can be expressed as [Eq. 2]:

$$EnC = d(EnC_{Processor} + EnC_{PrimaryStorage} + EnC_{secondarystorage} + EnC_{Motherboards} + EnC_{NetworkCards}) + t'(EnC_{Hardware} + EnC_{LANcards} + \sum_{f=0}^{F} d_{connectors,f} + EnC_f) + (EnC_{NetworkAnalysisServer} + EnC_{MemoryManager} + EnC_{NetworkAttachedStorageArrays}) + EnC_{Extra} \quad (2)$$

The energy consumed by a transceiver and all its ports can be defined as: where $EnC_{Hardware}$ is related to the energy consumed by the transceiver, $EnC_{LANcards}$ is the energy consumed by any active network LAN card, EnC_f corresponds to the energy consumed by a connector (port) running at the frequency f. In the equation [Eq. 2], only the last component appears to be dependent on the link frequency while other components, such as $EnC_{Hardware}$ and $EnC_{LANcards}$ remain fixed for all the duration of transceiver operation. Therefore, $EnC_{Hardware}$ and $EnC_{LANcards}$ can be avoided by turning the transceiver off or putting it into sleep mode. $EnC_{t',i}$ is the energy consumption at given time t is defined in [Eq. 3]:

$$EnC_{t,i}(r) = q \times EnC_{max} + (1 - q) \times EnC_{max} \times ru \quad (3)$$

Where EnC_{max} is maximum energy consumption while resource is fully utilized, q is fraction of energy consumed by idle resource and *ru* is resource utilization. Resource utilization is change over time and it is function of time and presented as $\mathrm{ru}(t)$. For a resource r_t at given time t, the resource utilization $ResU_t$ is defined as [Eq. 4]:

$$ResU_t = \sum_{i=1}^{n} EnC_{t,i}(ru(t))\, dt \quad (4)$$

where *n* is the number of cloud workloads running at time *t*. The actual energy consumption EnC_{actual} of a resource ru_t at given time *t* is defined as [Eq. 5]:

$$EnC_{actual} = (EnC_{max} - EnC_{min}) \times ResU_t + EnC_{min} \quad (5)$$

where EnC_{max} is the energy consumption at the peak load (or 100% utilization) and EnC_{min} is the minimum energy consumption in the active/idle mode (or as low as 1% utilization).

3.2 SOCCER Architecture

Architecture of SOCCER is shown in Figure 2. SOCCER is based on IBM's autonomic model [1] that considers four steps of autonomic system: 1) Monitor, 2) Analyze, 3) Plan and 4) Execute. SOCCER comprises of following units:

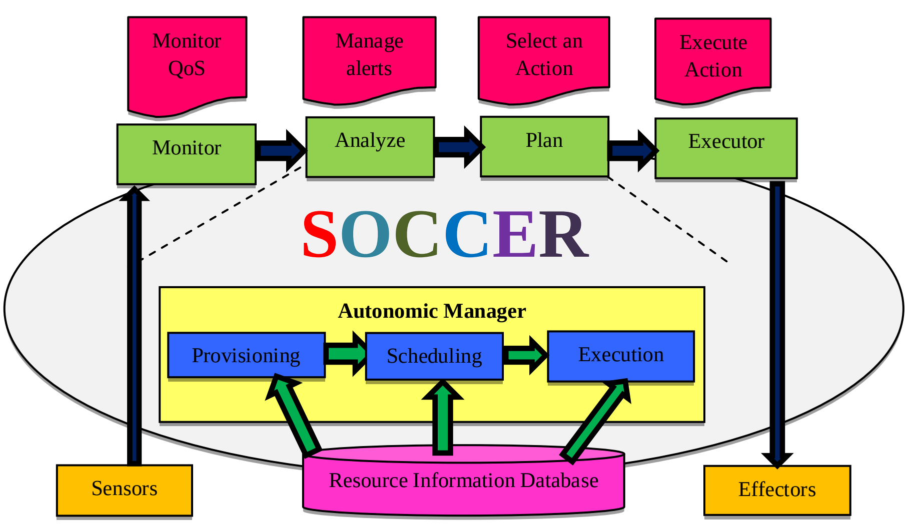

Figure 2: SOCCER Architecture

3.2.1 Monitors [M]

Initially, Monitors [M] are used to collect the information from sensors (*Sensors* get the information about energy consumption of all the systems working under cloud and update the information time to time) for monitoring continuously the value of energy consumption as shown in [ALGORITHM 1: Monitoring Unit (MU)] and transfer this information to next module for further analysis.

ALGORITHM 1: Monitoring Unit (MU)
 # AUTONOMIC MONITORING
Start
Set of Available Resources: R= $\{R_1, R_2, \ldots\ldots\ldots\ldots R_m\}$
Add Resources: RA= $\{R_1, R_2, \ldots\ldots\ldots\ldots R_o\}$ where o ≤ m
Allocate resources to process data based on QoS requirements
for workload processing, Calculate resource requirements
if (Required Resources < Provided Resources) **then**
Start execution of resources to process data
elseif
Generate Alert
end if
end for
for all resources (RA), Calculate EnC_{actual}
if ($[EnC_{actual} \leq E_t]$ == '$TRUE$') **then**
Continue execution of resources to process data
else
Generate alert
end if
end for

3.2.2 Analyze and Plan [AP]

Analyze and Plan [AP] module starts analyzing the information received from monitoring module and makes a plan for adequate actions for corresponding alert as shown in [ALGORITHM 2: Analyzing and Panning Unit (AU)]. In this step, based on QoS requirements of workload(s), resources are provisioned, scheduled and executed. It comprises of following subunits:

ALGORITHM 2: Analyzing and Panning Unit (AU)
 # AUTONOMIC ANALYZING AND PANNING
 # Check Resource Requirement
if (Provided Resources < Required Resources) **then**
Allocate new resources by using [Algorithm 3]
elseif
Generate Alert
end if
 # Check Energy Consumption
if ($EnC_{actual} \leq E_t$) **then**
start the resource execution
elseif
Reallocate resources by using [Algorithm 3]
elseif
Current resource is declared as dead node
Allocate new resources by using [Algorithm 3]
elseif
Generate Alert
end if

3.2.2.1 Resource Provisioning

Monitor continually checks the status of resources provisioned, workloads queued and energy consumption. The objective of resource provisionor is to provision the resources for execution of heterogeneous cloud workloads without violation of SLA [3]. The workloads submitted should be executed with minimum energy consumption. SOCCER provisions and schedules the resources based on energy consumption to the workloads automatically as shown in Figure 3.

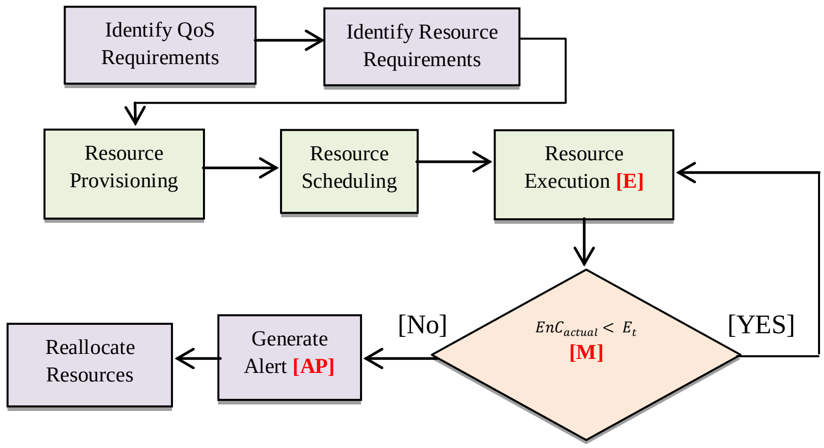

Figure 3: Automatic Execution of SOCCER

Workload submitted by user to resource provisionor is stored into bulk of workloads for their execution. All the submitted workloads are analyzed based on their QoS requirements described in terms of SLA as shown in Table 2.

Table 2: Cloud Workloads and their QoS Requirements

Workload	QoS Requirements
Websites	Reliable storage, High network bandwidth, High availability
Technological Computing	Computing capacity, Reliable storage
Endeavour Software	Security, High availability, Customer Confidence Level, Correctness
Performance Testing	Computing capacity, Network bandwidth, Latency
Online Transaction Processing	Security, High availability, Internet accessibility, Usability
E-Com	Variable computing load, Customizability
Central Financial Services	Security, High availability, Changeability, Integrity
Storage and Backup Services	Reliability, Persistence
Productivity Applications	Network bandwidth, Latency, Data backup, Security
Software/Project Development and Testing	User self-service rate, Flexibility, Creative group of infrastructure services , Testing time
Graphics Oriented	Network bandwidth , Latency, Data backup, Visibility
Critical Internet Applications	High availability, Serviceability, Usability
Mobile Computing Services	High availability, Reliability, Portability

Workload patterns are identified for better classification of workloads then pattern based clustering of workloads is done. QoS metrics for every QoS requirement of each workload are identified [2] [3] [4]. Based on importance of the attribute, weights for every cloud workload are calculated. After that, workloads are re-clustered based on *k*-means based clustering algorithm for better execution. Final set of workloads is shown in Table 3. Note: Detailed description of clustering of workloads is described in our previous research work [4].

Table 3: Clustering of Workloads

Cluster	Cluster Name	Workloads
C1	Compute	Technological Computing, Performance Testing
C2	Storage	E-Com and Storage and Backup Services
C3	Communication	Websites, Critical Internet Applications, Mobile Computing Services
C4	Administration	Endeavour Software, Online Transaction Processing, Central Financial Services, Productivity Applications, Software/Project Development and Testing and Graphics Oriented

Estimate the value of energy consumption of cloud resources based on their previous statistics of execution available in Resource Information Database (RID). If the value of energy consumption of workloads executes within range [Actual Energy Consumption denoted as EnC_{actual} is less than the threshold value of energy consumption (E_t)] then it will provision resources otherwise generate alert for analyses the workload again after reallocation of resources by autonomic manager. After finding the workload priority using EARTH [1], SOCCER calculates the resource requirements to check whether the provided resources are sufficient for execution of workload (s) are provided or not. If the sufficient resources are provided then start scheduling of resources for workload execution otherwise add new resources from pool of reserved resources. Resources are again allocated for further execution after finding the minimum value of energy consumption (also less than threshold value). The outcome of resource provisioning is set of provisioned resources, which is stored in RID.

3.2.2.2 Resource Scheduling

After successful provisioning of resources, Resource Scheduler (RS) takes the information from the appropriate workload after analyzing the heterogeneous workload details which cloud consumer demanded [2]. [Algorithm 3: Energy Aware Resource (EAR) Scheduling Algorithm] is used to schedule the resources effectively with minimum energy consumption. RS then collects the information of available resources from RID. RID contains details of all the resources available in resource pool and reserve resource pool. Based on cloud consumer details RS assigns resources and executes heterogeneous cloud workloads. Workload with highest priority is put into the categories of urgent workloads and remaining will be considered as non-urgent workloads. SOCCER automatically checks the total workloads in the workload queue after each new workload is added. Priorities of workloads are changing adaptively. The reason for changing priorities might be that priority of newly added workload is higher. For this workload deadline is mandatory to consider. Otherwise, new workload with higher priority waits for long time which leads to starvation and reduce user satisfaction. Therefore, we used [Algorithm 3] for this purpose.

ALGORITHM 3: Energy Aware Resource (EAR) Scheduling Algorithm

1. **Start**
2. Set maximum value of energy consumption to a certain threshold value (E_t).
3. Get cloud workload
4. Set limit of execution of workloads → L
5. Get list of available resources→ R
6. **if** available workload≤ L
7. **do** execution
8. Assign resource *Ri* from *R* to workload based on QoS (EnC_{actual}) requirement of Workload(s)
9. Repeat steps 5 to 6 **until** condition 4 is **true**
10. Calculate energy consumption (EnC_{actual})
11. **if** ($EnC_{actual} < E_t$)
12. Complete the execution
13. **else**
14. Start scheduling the workloads according to priority.
15. **if** new workload *(NW)* arrive > *L*
16. **if** (the workload is urgent)
17. Assign reserve resources to execute new workload *(NW)*
18. Recalculate Energy Consumption [**Goto** Step 9]
19. **else** put new workload *(NW)* into waiting queue.
20. **Stop**

3.2.2.3 Resource Execution

During execution of a particular cloud workload, the Resource Executor (RE) will check the current workload. If the resources are sufficient for execution then it will continue with execution otherwise request for more resources. RE will monitor the value of energy consumption continually. If the value of Actual Energy Consumption (EnC_{actual}) is less than the threshold value of energy consumption (E_t) then RE will execute workloads otherwise RE will generate alert for rescheduling of resources. After successful execution of cloud workloads, RE releases the free resources to resource pool and RE is ready for execution of new cloud workloads.

3.2.3 Executor [E]

Executor implements the plan after analyzing completely. To reduce the energy consumption is a main objective of executor as shown in [ALGORITHM 4: Executing Unit (EU)]. Based on the output given by analysis and executor tracks the new workload submission and resource addition, and generates the alert. *Effector* is used to transfer the new policies, rules and alerts to other nodes with updated information.

ALGORITHM 4: Executing Unit (EU)

 # AUTONOMIC EXECUTION
if (Required Resources > Provided Resources) **then**
    Allocate new resource from reserve resource pool with minimum value of EnC_{actual} and [$EnC_{actual} \leq E_t$]
end if
for all resources
  if ($EnC_{actual} \leq E_t$) = = '*FALSE*' **then**
    Restart the resource and start execution
  else if
    Declared resource as dead node and removed
    Resource(s) is required to process data without degradation in performance
    if (Availability of Resource(s) = = '*TRUE*')
    Allocate new resource from resource pool with minimum value of EnC_{actual} and [$EnC_{actual} \leq E_t$]
    else
    Resource(s) is required to process data without degradation in performance
    Add new resource from reserve resource pool with minimum value of EnC_{actual} and [$EnC_{actual} \leq E_t$]
  end if
end for

4. Experimental Setup and Results

We have used empirical methods to evaluate the performance of SOCCER. Tools used for setting cloud environment for empirical evaluation are Microsoft Visual Studio 2010, NetBeans IDE 7.1.2, Oracle Java SDK V.6, Aneka, SQL Server 2008 and, JADE Platform (for agents). In this experimental setup, three different cloud platforms are used: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The integration of multiple environments used to conduct experiments is shown in Figure 4. Cloud user interacts with Cloud Workload Management Framework through Cloud Workload Management Portal (CWMP) to submit the workload details. Note: CWMP is described in our previous research work [26]. *At software level*, Microsoft Visual Studio 2010 is used to provide user interface in which user can access service from any geographical location. *At platform level*, Aneka cloud application platform is used as a scalable cloud middleware to make interaction between IaaS and SaaS, and continually monitor the performance of the system. At *Infrastructure level*, three different servers (consist of virtual nodes) have been created through Citrix Xen Server and SQL Server has been used for data storage. Scheduler runs at IaaS level on Citrix Xen Server. Computing nodes used in this experiment work are further categorized into three categories as shown in Table 4. Energy consumption is measured in Kilo-Watt-Hour (kWh) using *Joule Meter*. Experiment setup using 3 servers in which further virtual nodes (12 = 6 (Server 1) +4 (Server 2) +2 (Server 3)) are created. Every virtual node has different number for Execution Components (ECs) to process user request and every EC has their own value of energy consumption (kWh/EC time unit (Sec)). Table 4 shows the characteristics of the resources used and their Execution Component (EC) access energy consumption per time unit in kWh.

Table 4: Configuration Details of Thapar Cloud

Resource_Id	Configuration	Specifications	Operating System	Number of Virtual Node	Number of ECs	Energy Consumption (kWh/EC time unit)
R1	Intel Core 2 Duo - 2.4 GHz	1 GB RAM and 160 GB HDD	Windows	6	18	18
R2	Intel Core i5-2310- 2.9GHz	1 GB RAM and 160 GB HDD	Linux	4	12	21
R3	Intel XEON E 52407-2.2 GHz	2 GB RAM and 320 GB HDD	Linux	2	6	25

4.1 Experimental Results

In order to validate SOCCER, we have selected two existing energy efficient resource scheduling approaches i.e. Preemption-Aware Energy Efficient (PAEE) [7] and Energy Credit Scheduler (ECS) [14] as discussed in *Section 2*.

Energy consumption of SOCCER, PAEE and ECS is compared based on four different clusters of workloads [a) Compute (C1), b) Storage (C2), c) Communication (C3) and d) Administration (C4)] as shown in Table 3. With increase the number of workloads, energy consumption is increasing.

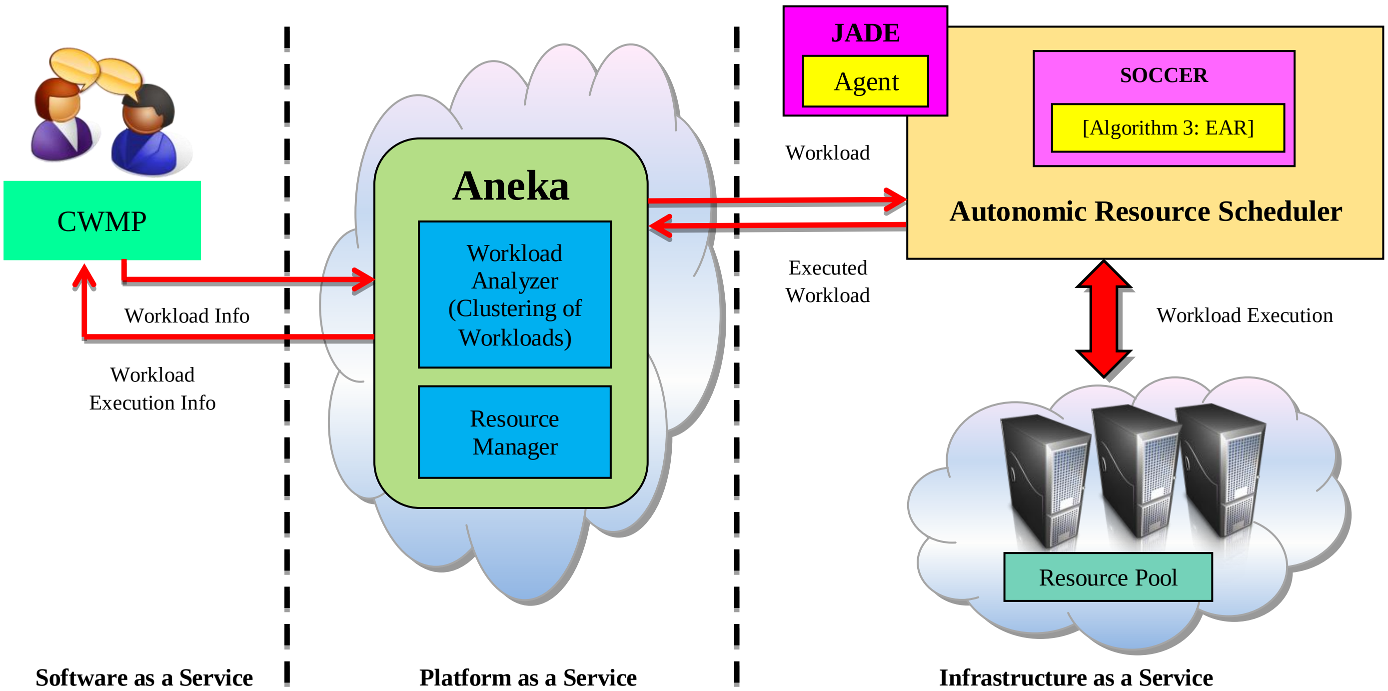

Figure 4: Cloud Environment at Thapar University

Table 5 shows the different cloud workloads considered for different test cases.

Table 5: Cloud Workloads Considered for Different Test Cases

Test Case	Cluster	Workload	Description
Test Case-1	Compute (C1)	Performance Testing	[Processing Larger Image File of Size 713 MB], in which SOCCER converts an image file from JPEG format to PNG format. Conversion of a single JPEG file into PNG is considered as a single workload.
Test Case-2	Storage (C2)	Storage and Backup Data	Store larger amount of data (5 TB) and creates backup of data
Test Case-3	Communication (C3)	Website	Website of Thapar University [http://www.thapar.edu] Website is accessed by a large number of users during Admission Period
Test Case-4	Administration (C4)	Software Development and Testing	Developed and tested Agri-Info [to manage agriculture related information] Software in a controlled environment. [http://www.cloudbus.org/reports/AgriCloud2015.pdf]

Test Case 1: Energy Consumption with Different Number of Workloads for Compute Cluster (C1)

Figure 5 shows the energy consumption of different number of workloads (10-60) for SOCCER, PAEE and ECS in Compute Cluster. It is clearly shown that the PAEE and ECS consuming more energy than SOCCER at different workloads. At 50 workloads, energy consumption in SOCCER is 8.37% lesser than PAEE and 6.42% lesser than ECS. Average energy consumption in SOCCER is 5.47% and 6.66% lesser as compared to PAEE and ECS respectively.

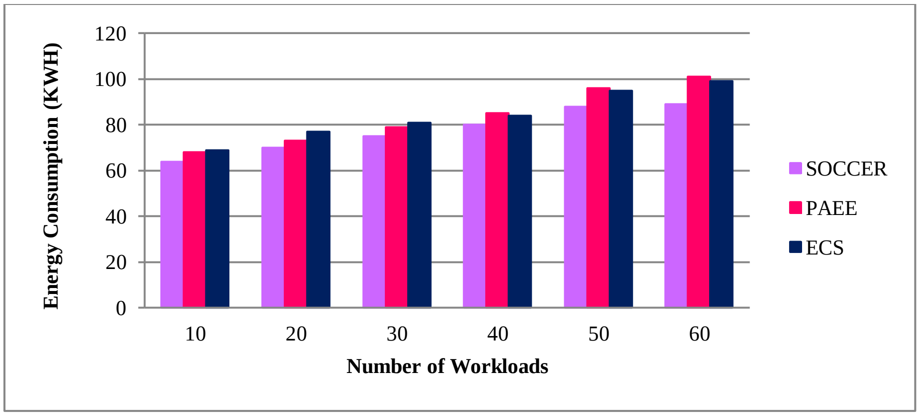

Figure 5: Comparison of Energy Consumption with Different Number of Workloads for Compute Cluster (C1)

Test Case 2: Energy Consumption with Different Number of Workloads for Storage Cluster (C2)

Comparison of energy consumption for SOCCER, PAEE and ECS in Storage Cluster with different number of workloads is shown in Figure 6. SOCCER performs better than PAEE and ECS in terms of energy consumption. SOCCER consumes 12.36% lesser than PAEE and 14.31% lesser than ECS at 50-60 workloads. SOCCER reduces 6.16%-8.71% and 7.95%-9.82% average energy consumption as compared to PAEE and ECS respectively.

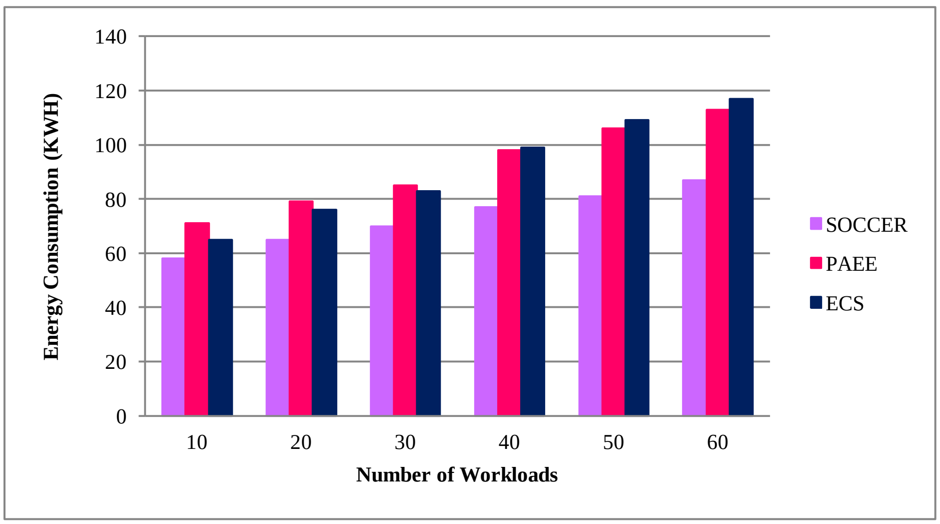

Figure 6: Comparison of Energy Consumption with Different Number of Workloads for Storage Cluster (C2)

Test Case 3: Energy Consumption with Different Number of Workloads for Communication Cluster (C3)

Figure 7 shows the energy consumption of different number of workloads for SOCCER, PAEE and ECS in Communication Cluster. It is clearly shown that the PAEE and ECS consuming more energy than SOCCER at different workloads. At 40 workloads, energy consumption in SOCCER is 5.65% lesser than PAEE and 7.12% lesser than ECS. Average energy consumption in SOCCER is 4.71% and 6.12% lesser as compared to PAEE and ECS respectively.

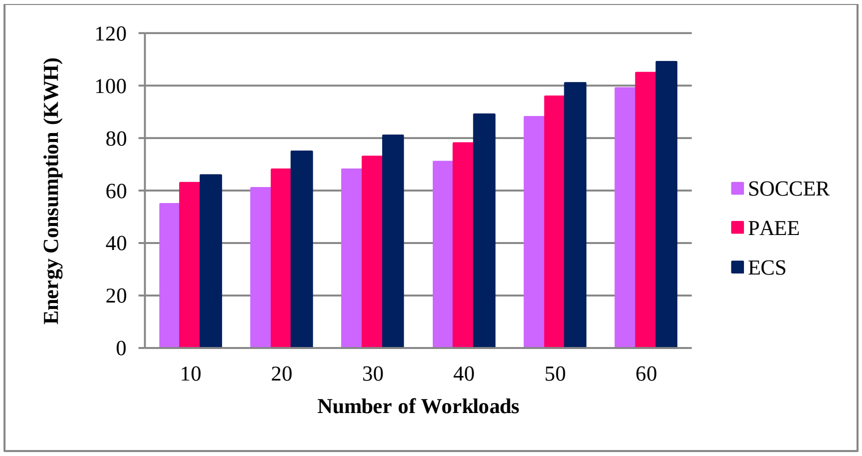

Figure 7: Comparison of Energy Consumption with Different Number of Workloads for Communication Cluster (C3)

Test Case 4: Energy Consumption with Different Number of Workloads for Administration Cluster (C4)
Comparison of energy consumption for SOCCER, PAEE and ECS in Administration Cluster with different number of workloads is shown in Figure 8. SOCCER performs better than PAEE and ECS in terms of energy consumption. SOCCER consumes 13.76% lesser than ECS and 17.71% lesser than PAEE at 40-60 workloads. SOCCER reduces 7.75%-9.46% and 10.69%-12.77% average energy consumption as compared to ECS and PAEE respectively.

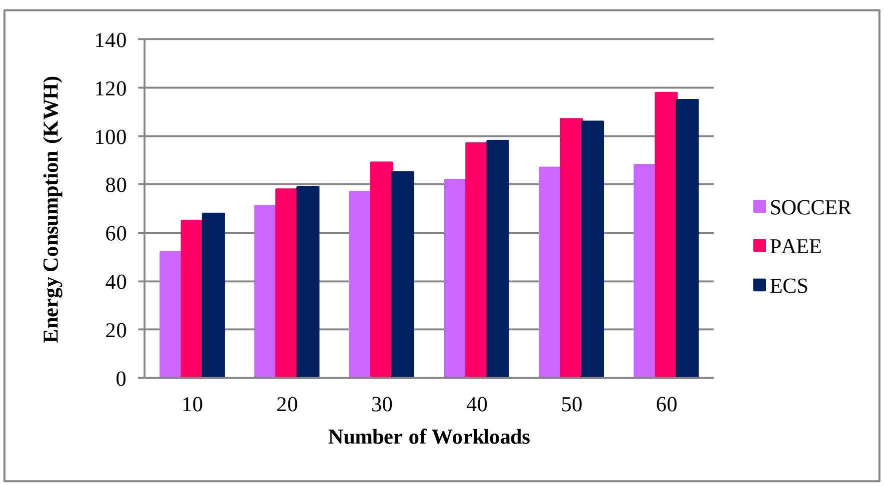

Figure 8: Comparison of Energy Consumption with Different Number of Workloads for Administration Cluster (C4)

Test Case 5: Energy Efficiency

Energy Efficiency is a ratio of number of workloads successfully executed in a data center to total energy consumed to execute those workloads. [Eq. 6] is used to calculate energy efficiency.

$$Energy\ Efficiency_i = \sum_{i=1}^{n} \left(\frac{\text{number of workloads successfully executed in a data center}}{\text{total energy consumed to execute those workloads}} \right) \quad (6)$$

In this test case, energy efficiency is measured for all the four clustered with different number of resources (Execution Components) as shown in Figure 9. It has been depicted from Figure 9; the value of energy efficiency is increasing with increasing in number of resources.

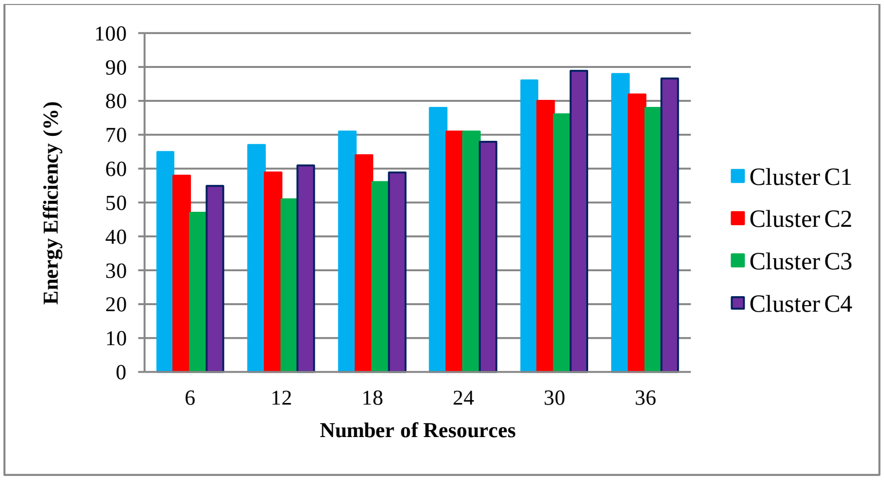

Figure 9: Energy Efficiency of different Workload Clusters with Number of Workloads

5. Conclusions and Future Directions

In this paper, energy efficient autonomic cloud computing system is proposed for energy efficient scheduling of cloud computing resources in data centers. The proposed system (SOCCER) has been validated in real cloud environment. *SOCCER* focused on energy efficient scheduling of computing resources in virtual data centers for execution of heterogeneous cloud workloads and uses the autonomic model to optimize the energy consumption. The experimental results show that the proposed system performs better in terms of energy consumption as compared to existing systems. Further, SOCCER can be extended by developing a QoS aware autonomic resource provisioning and scheduling technique which will consider self-healing (find and react to sudden faults), self-optimization (maximize resource utilization and cost and time efficiency), self-configuration (capability to readjust resources) and self-protecting (detection and protection of cyber-attacks).

Acknowledgement

One of the authors, Sukhpal Singh (SRF-Professional), acknowledges the Department of Science and Technology (DST), Government of India, for awarding him the INSPIRE (Innovation in Science Pursuit for Inspired Research) Fellowship (Registration/IVR Number: 201400000761 [DST/INSPIRE/03/2014/000359]) to carry out this research work. We would like to thank all the anonymous reviewers for their valuable comments and suggestions for improving the paper.

References

[1] Singh, S., Chana, I.: EARTH: Energy-aware autonomic resource scheduling in cloud computing. Journal of Intelligent & Fuzzy Systems, **30**(3), 1581-1600 (2016).

[2] Singh, S., Chana, I.: Resource provisioning and scheduling in clouds: QoS perspective. The Journal of Supercomputing, **72**(3), 926-960 (2016)

[3] Singh, S., Chana, I.: Q-aware: Quality of service based cloud resource provisioning. Computers & Electrical Engineering, **47**, 138-160 (2015)

[4] Singh, S., Chana, I.: QRSF: QoS-aware resource scheduling framework in cloud computing. The Journal of Supercomputing, **71**(1), 241-292 (2015)

[5] Chen, K., Hu, C., Zhang, X., Zheng, K., Chen, Y., Vasilakos, A. V.: Survey on routing in data centers: insights and future directions. IEEE network, **25**(4), 6-10 (2011)

[6] Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J. M., Vasilakos, A. V.: Cloud computing: Survey on energy efficiency. ACM Computing Surveys (CSUR), **47**(2), 1-36 (2015)

[7] Salehi, M. A., Krishna, P. R., Deepak, K. S., Buyya, R.: Preemption-aware energy management in virtualized data centers. In the Proceeding of 5th IEEE International Conference on Cloud Computing (CLOUD). 844-851 (2012)

[8] Ren, S., He, Y., Xu, F.: Provably-efficient job scheduling for energy and fairness in geographically distributed data centers. In the Proceeding of 32nd IEEE International Conference on Distributed Computing Systems (ICDCS), 22-31 (2012)

[9] Pelley, S., Meisner, D., Zandevakili, P., Wenisch, T. F., Underwood, J.: Power routing: dynamic power provisioning in the data center. ACM Sigplan Notices **45**(3) 231-242 (2010)
[10] Urgaonkar, R., Urgaonkar, B., Neely, M. J., Sivasubramaniam, A.: Optimal power cost management using stored energy in data centers. In the Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems. 221-232 (2011)
[11] Shen, S., Wang, J.: Stochastic modeling and approaches for managing energy footprints in cloud computing service. Service Science, **6**(1), 15-33 (2014)
[12] Changtian, Y., Jiong, Y.: Energy-aware genetic algorithms for task scheduling in cloud computing. In the Proceedings of the Seventh IEEE ChinaGrid Annual Conference. 43-48 (2012)
[13] Ma, Y., Gong, B., Sugihara, R., Gupta, R.: Energy-efficient deadline scheduling for heterogeneous systems. Journal of Parallel and Distributed Computing, **72**(12), 1725-1740 (2012)
[14] Kim, N., Cho, J., Seo, E.: Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems. Future Generation Computer Systems, **32**, 128-137 (2014)
[15] Chen, C., He, B., Tang, X.: Green-aware workload scheduling in geographically distributed data centers. In the Proceeding of 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom). 82-89 (2012)
[16] Wang, B., Qi, Z., Ma, R., Guan, H., Vasilakos, A. V.: A survey on data center networking for cloud computing. Computer Networks. **91**, 528-547 (2015)
[17] Wang, L., Zhang, F., Vasilakos, A. V., Hou, C., Liu, Z.: Joint virtual machine assignment and traffic engineering for green data center networks. ACM SIGMETRICS performance evaluation review, **41**(3), 107-112 (2014)
[18] Wang, L., Zhang, F., Zheng, K., Vasilakos, A. V., Ren, S., Liu, Z: Energy-efficient flow scheduling and routing with hard deadlines in data center networks. In the Proceeding of 34th IEEE International Conference on Distributed Computing Systems (ICDCS) 248-257 (2014)
[19] Shu, Z., Wan, J., Zhang, D., Li, D: Cloud-integrated cyber-physical systems for complex industrial applications. Mobile Networks and Applications, 1-14 (2015)
[20] Wang, L., Zhang, F., Aroca, J. A., Vasilakos, A. V., Zheng, K., Hou, C, Liu, Z: GreenDCN: a general framework for achieving energy efficiency in data center networks. IEEE Journal on Selected Areas in Communications, **32**(1), 4-15.
[21] Polverini, M., Cianfrani, A., Ren, S., Vasilakos, A. V.: Thermal-aware scheduling of batch jobs in geographically distributed data centers. IEEE Transactions on cloud computing, **2**(1), 71-84 (2014)
[22] Liu, Q., Wan, J., Zhou, K.: Cloud manufacturing service system for industrial-cluster-oriented application. Journal of Internet Technology, **15**(3), 373-380 (2014)
[23] Mashayekhy, L., Nejad, M. M., Grosu, D., Vasilakos, A. V.: An online mechanism for resource allocation and pricing in clouds. IEEE Transactions on Computers, **65**(4), 1172-1184 (2016)
[24] Xu, D., Liu, X., Vasilakos, A. V: Traffic-aware resource provisioning for distributed clouds. IEEE Cloud Computing, **2**(1), 30-39 (2015)
[25] Choi, S., Chung, K., Yu, H.: Fault tolerance and QoS scheduling using CAN in mobile social cloud computing. Cluster Computing, **17**(3), 911-926 (2014)
[26] Singh, S., Chana, I.: Efficient Cloud Workload Management Framework. Masters Dissertation. Thapar University, India. (2013). Retrieved From: http://dspace.thapar.edu:8080/jspui/bitstream/10266/2247/1/sukhpal_singh_me_thesis.pdf

Author's Bibliography

Sukhpal Singh joined Computer Science and Engineering Department of Thapar University, Patiala, India, in 2016 as Lecturer. Dr. Singh obtained the Degree of Master of Engineering in Software Engineering from Thapar University, as well as a Doctoral Degree specialization in "Autonomic Cloud Computing" from Thapar University,. Dr. Singh received the Gold Medal in Master of Engineering in Software Engineering. Dr. Singh is a DST Inspire Fellow [2013-2016] and worked as a SRF-Professional on DST Project, Government of India. He has done certifications in Cloud Computing Fundamentals, including Introduction to Cloud Computing and Aneka Platform (US Patented) by ManjraSoft Pty Ltd, Australia and Certification of Rational Software Architect (RSA) by IBM India. His research interests include Software Engineering, Cloud Computing, Operating System and Databases. He has more than 30 research publications in reputed journals and conferences.

Inderveer Chana joined Computer Science and Engineering Department of Thapar University, Patiala, India, in 1997 as Lecturer and is presently serving as Professor in the department. She is Ph.D. in Computer Science with specialization in Grid Computing, M.E. in Software Engineering from Thapar University and B.E. in Computer Science and Engineering. Her research interests include Grid and Cloud computing and other areas of interest are Software Engineering and Software Project Management. She has more than 100 research publications in reputed Journals and Conferences. Under her supervision, more than 30 ME thesis and four Ph.D thesis have been awarded and four Ph.D. thesis are on-going. She is also working on various research projects funded by Government of India.

Maninder Singh received his Bachelor's Degree from Pune University in 1994, and holds a Master's Degree, with honors in Software Engineering from Thapar Institute of Engineering & Technology, as well as a Doctoral Degree specialization in Network Security from Thapar University. Dr. Singh is currently working as Associate Professor in Computer Science and Engineering Department at Thapar University. Dr. Singh is on the Roll-of-honour at EC-Council USA, being certified as Ethical Hacker (C—EH), Security Analyst (ECSA) and Licensed Penetration Tester (LPT). Dr. Singh has successfully completed many consultancy projects for renowned national bank(s).

Rajkumar Buyya is a Fellow of IEEE, Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council, and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercialising its innovations in Cloud Computing. He has authored over 500 publications and four text books. He is one of the highly cited authors in computer science and software engineering worldwide (h-index 100+, 50000+ citations). He has served as the foundation Editor-in-Chief (EiC) of IEEE Transactions on Cloud Computing and now serving as Co-EiC of Journal of Software: Practice and Experience.