SNMP for Cloud Environment Energy Efficiency

Marta Chinnici (✉ Marta.Chinnici@enea.it)

ENEA <https://orcid.org/0000-0001-8123-2791>

Asif Iqbal

Leeds Beckett University - City Campus: Leeds Beckett University

ah lian kor

Leeds Beckett University - City Campus: Leeds Beckett University

colin pattinson

Leeds Beckett University - City Campus: Leeds Beckett University

eric rondeau

Université de Lorraine Site Nancy: Universite de Lorraine

Research Article

Keywords: Data Center, Cloud Computing, Virtual Servers, Energy Efficiency, Networks, SNMP

Posted Date: September 13th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-720622/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

SNMP for Cloud Environment Energy Efficiency

Abstract. Cloud computing has seen rapid growth and environments are now providing multiple physical servers with several virtual machines running on those servers. Networks have grown larger and have become more powerful in recent years. A vital problem related to this advancement is that it has become increasingly complex to manage networks. SNMP is one standard which is applied as a solution to this management of networks problem. This work utilizes SNMP to explore the capabilities of SNMP protocol and its features for monitoring, control and automation of virtual machines and hypervisors. For this target, a stage-wise solution has been formed that obtains results of experiments from the first stage uses SNMPv3 and feed to the second stage for further processing and advancement. The target of the controlling experiments is to explore the extent of SNMP capability in the control of virtual machines running in a hypervisor, also in terms of energy efficiency. The core contribution based on real experiments is conducted to provide empirical evidence for the relation between power consumption and virtual machines.

Keywords: Data Center, Cloud Computing, Virtual Servers, Energy Efficiency, Networks, SNMP

1 Introduction

In recent years, cloud computing has seen rapid growth and to provide Virtual Machines (VMs) multiple users, a significant number of servers are deployed in Data Centers (DCs). Networks have grown larger and have become more powerful in recent years, bringing convenience to government, business, educational institutes and individuals. A vital problem that is related to this advancement is that it has become increasingly complex to manage networks. Existing standards such as TMN, CMPI and SNMP have been applied to deal with network management issues [1]. These standards cover protocols, services and management information bases. Among these standards, Simple Network Management Protocol (SNMP) has been considered as an important and most widely used protocol for Transmission Control Protocol/Internet Protocol (TCP/IP) based networks. SNMP can be used for controlling and management of virtual machines running in various hypervisor environments. Large and complex networks require automated and efficient network management system because just human efforts are not enough to manage them [2]. Additionally, Simple Network Management Protocol (SNMP) has been applied as *de facto* [3]. Management systems need to monitor information values in real-time because real-time monitoring can help to keep track of the network's bandwidth, performance, availability and troubleshooting [4]. An effective energy savings strategy would be to reduce electricity consumption by minimising the resource load [5], [6] for keeping servers (in the computer system) if properly managed, servers running multiple VMs can be made energy efficient. Focuses on the monitoring and managing of DC resource energy consumption, this work aims to enhance power consumption of virtualized computing systems based on the development of models, methods, and

tools for power consumption analysis by taking into account CPU load of VMs. This work focuses a rigorous cloud computing literature review and addresses the following research objectives:

1. Analyze tools for virtual machine monitoring in control and management in cloud/virtual environments;
2. Critically review existing systems and their underlying principles which incorporate the use of SNMP to manage cloud/virtual environments;
3. Develop a test-bed and experimental setup for monitoring and controlling of virtual machines in cloud/virtual environments;
4. Develop a model simulating CPU load based on several CPU processes benchmarks;
5. Develop a method/technique to predict server power consumption based on the number of virtual machines and their load;
6. Recommend practical implementation of the proposed method and models based on (3-5).

The remainder of the paper is organized as follows: Section 2 is dedicated to discussion of related work; Section 3 introduces the paper methodology; Section 4 contains Results and discussion; while Section 5 concludes the paper.

2 Background: Related Work

This section outlines the literature review and related work of the critical analysis in the domains of five pivotal elements for VM- problem analysis:

1. **Monitoring:** VM minimal overhead and high security become a significant challenge. It is crucial to investigate whether details of resource consumption can be obtained through monitoring hypervisor alone or hypervisor and virtual machines together.
2. **Controlling:** in the cloud environment, VMs run on numerous physical servers and to control them (e.g. SNMPv3) become essential to reduce the amount of energy consumption.
3. **Automation:** to appropriately change the state of a VMs or create new ones when the load exceeds a certain threshold. To control power based on demand can reduce energy consumption.
4. **Energy efficiency:** The energy consumption of today's systems is rapidly increasing with the growth of digital devices usage. Consequently, applications executed on those digital devices increasingly demand more resources including power. Monitoring, control and management of power utilization by these systems is necessary to effect energy efficiency [7].
5. **Virtual Local Area Networks (VLANs) impact:** A vital advantage of using VLANs is that it provides flexibility to allow any logical LAN to be implemented on any physical infrastructure.

The management of the network is necessary to ensure security, reliability, and network availability [8]. Systems used to assist network administrators in supervising individual components, and network services are known as network management systems. To unravel any issue in the network, it is appropriately monitored, and its efficiency is measured using the NMS [9]. Many applications are proposed to monitor network traffic, and most of these applications rely on the simple network monitoring protocol [10]. The monitoring of the network is essential to ensure no issue in the flow of data in the entire network. This is to ensure the problem-free flow of data. Network monitoring is used to identify the status of systems attached to a network and their behaviour. To detect any unusual traffic or intruders, network monitoring plays an important role [11]. Simple Management Network Protocol (SNMP) was first introduced in 1988 to manage TCP/IP based networks as a short-term solution (Stallings, 1998). Monitoring and controlling in TCP/IP networks is made possible using SNMP's Set, Get and Trap operations (Stallings, 1998). Deployment and implementation of SNMP management systems is considered quite important where TCP/IP is dominant as SNMP provides features like fault management, account management and configuration management [12]. Initial and original SNMP suite had several deficiencies and limitations, so in 1993 a new version SNMPv2 was introduced and published. A new set of Request for Comments (RFCs) documents, collectively known as SNMPv3, was presented to enhance and incorporate features such as security and remote configurations. The goal of these documents was to raise a single standard for core SNMP functions for the next generation [13], [14], [15]. There are multiple ways to use SNMP. Monitoring the status of the system or controlling resource allocation are some of the tasks that a network administrator can do. Powering up or shutting down the device, allocating resources to a particular virtual machine (VM), involves networking monitoring. SNMP is designed to make networking monitoring easier and because of that majority of current Network Monitoring Tools use SNMP at the primary level [16]. Three core components of SNMP are SNMP Agent, SNMP Manager and the Management Information Base (MIB) [17]. SNMP Manager, also known as management station, acts as an interface for the network management system's human network manager. SNMP manager monitors and controls the network management process. SNMP agent or simply an Agent is also an essential component of any network management. Devices such as Workstations, Bridges, Servers and routers can be monitored and controlled by an SNMP manager accompanied by an SNMP agent. Management information collected and recorded for network elements is communicated to the manager. The third component, Management Information Base (MIB), is introduced, and it contains local configuration and traffic information. A global MIB is maintained by the management station, which includes a summary of all the agents' knowledge. Agents use two techniques to collect management information and make it available to the management system. One of the methods is polling, and the other is event reporting. In polling, the manager queries the agent's information, which consults its MIB and responds with relevant information. Whereas, in event reporting, the manager listens for event reports generated by agents [14]. To network management, a set of applications is considered. Primary applications for configuration control, performance monitoring and accounting are minimally provided in any system [18]. SNMP is one such set that can be used for controlling and monitoring. This work

focuses on using SNMP to monitor and control systems running in the cloud environment.

2.1 Cloud Environment

Cloud computing is a paradigm for enabling useful, on-demand and global network access to a shared pool of configurable computing resources that can be promptly provisioned and released with remote management work or service provider involvement [19], [20]. Therefore, a cloud is service-oriented and services are provided on-demand based on hardware and software resources. Cloud computing is the result of the evolution of technology which begins from Grid computing [21] to utility computing, utility computing to SaaS computing and SaaS computing to Cloud computing [22] [23]. Services implemented as a part of this work are also based on IaaS concept. When all an organizations' computing requirements, such as servers, storages, and hosting, etc. are maintained or deployed on separate and dedicated or personal hardware and probably, on a private location, it is referred to as a private cloud. However, when all these services are located virtually on some remote areas, and only the required portion of resources is rented, without knowledge of hardware which underlies the benefits as well as the physical location of hardware, such clouds are referred as Public Cloud [24]. Some researchers submitted a draft to IETF in 2012 on Cloud Reference Framework [25]. The IETF draft on the cloud reference framework refers to different layers for integration, interoperability and virtualized applications operations [25]. The IETF proposed draft outlines the regulation of cloud functional elements and the interfaces between the functions. The horizontal Layers are divided into the following groups (as shown in Figure 1).

- Application/Service Layer
- Resources Control Layer
- Resources Abstraction and Virtualization Layer
- Physical Resources Layer

2.1.1 Open Stack

OpenStack is a type of software developed by NASA in 2010 with a vision to provide Data Center (DC) resources in stackable and shared basis, irrespective of underlying hardware limitations [26]. It is managed by the Open Stack Foundation which looks after the developments and advancements in OpenStack and brings the OpenStack community together believing that OpenStack is the future of cloud computing. The open stack is a Content Management System (CMS), a DC that is hardware resources in most straightforward possible words. It provides user experience benefits that a CMS like WordPress or Joomla provide in website management [27]. Talking very specifically about OpenStack, the complete DC' operating system manages all the available resources like storage, processing, computing, networking etc. in the physical DC [28]. Deploying open stack in a DC is possible through a web-based dashboard of all the underlying resources, making resource management very easy for public and private or any other kind of cloud [29].

2.2 Hypervisors

The hypervisor is a hardware virtualization technique that can create and run virtual machines by running multiple guest operating systems of different or same type on a single computer system [30]. This includes RAM, CPU, etc... into multiple independent virtual machines known as guests. Each virtual machine or guest can run its own OS giving an illusion that the virtual machine has its own RAM, CPU and other resources. To make it appear as such and provide such impression of independent resources, it needs to have feature access and support of underlying processor such as AMD-V on AMD or VT-X on Intel [31]. The hypervisor's computer system is called the host machine while each virtual machine is referred to as the guest machine. The hypervisor is also called a Virtual machine monitor or VMM. The traditional server architecture consists of a single operating system per physical server and one application per single OS. In VMware virtualization, a hypervisor is installed on a physical server to allow multiple unique and isolated VMs to run on the same physical server. Once the OS is separated from the server's hardware, the hypervisor can be utilized to present a complete virtual x86 platform to many virtual machines [32]. A hypervisor is basically a technique of virtualizing the physical computing environment of the host by means of splitting its resources [33]. Hypervisor essentially emulates the hardware of the physical machine which is very critical and affects the virtual machine's performance. Hypervisor operates on the principle of isolation which in other words, means that operation of one guest system does not have any effect or impact on the other guests or even host even if it gets crashed.

2.3 Benefits and Barriers of VMs

As defined initially by Popek and Goldberg, "virtual machine as an efficient, isolated duplicate of a real computer machine [34]. Modern use consists of virtual with no direct connection to any real hardware [35]. Virtual machines are based on computer architectures and offer a physical computer's functionality. Their implementations may comprise software, specialized hardware, or combination of both. Some of the critical properties of virtual machines include [36]:

- **partitioning** in running multiple operating systems on one single machine and dividing system resources between virtual machines.
- **isolation** in terms of fault and security isolation at the hardware level and advanced resource controls reserve performance
- **encapsulation** in terms of the entire state of the virtual machine as a set of files and easy migration and transferability of virtual machines
- **hardware independence** in terms of any virtual machine to be portable to any physical server

There are considerable benefits that can be realized using virtualization. They can be classified as:

- **consolidation**: To reduce the number of servers by simultaneously sharing the server's computing resources among multiple services and applications.
- **containment**: ongoing effort to virtualize new applications and manage the growth of existing apps

- **Availability:** introducing virtualization to enhance application availability and data recoverability

The benefits of virtualization shall be discussed. The following are some of the disadvantages of virtualization [37].

- **Upfront Cost:** A high cost required to purchase the virtualization software, and probably extra hardware to make the virtualization possible;
- **Limited Applications:** Virtualization environment does not support every application. It means the company or individual may need to establish a hybrid system to work correctly;
- **Learning:** Implementation and management of virtualization require staff with expertise in virtualization. For a person with knowledge of non-virtualization environment may need a long time to understand virtualization environment.

Many systems have been developed to make a network administrators' job more comfortable and efficient in managing the network. Some of the standard network monitoring systems are open source and discussed and comparison. It is crucial to obtain statistical information about the use of the network through some monitoring techniques such as SNMP [38], RMON [39] and NetFlow [40], to perform an efficient monitoring of networks. As aforementioned at the beginning of the section, the authors provide a critical analysis of the five pivotal elements for VM- problem analysis.

Monitoring: Research published by [41] and [42] focused on monitoring of virtual machines for improvement in I/O processes. Cherkasova used a light-weight monitoring tool to monitor the performance of CPU overhead in I/O processes by creating a web server based workloads. Ongaro used a similar approach to measure CPU performance by running various applications on guest virtual machines [41]. Other papers focused on virtual machine monitoring using some custom-built library or tool like Xenoprof, Xenaccess and Java-based object-oriented VMM technique. Each tool and technique focused on monitoring virtual machines using various configurations of virtual machines and applications. Research published by Peng et al. [43] proposed SNMP-based monitoring method. Peng et al. focused on utilizing standard MIBs and defining new custom MIBs for getting vital monitoring information while Shin used custom polling technique to achieve monitoring goal. The approach in this thesis is similar to approaches used in these papers. As an example, SNMPv3 is used to monitor virtual machines to get specifications, physical memory and processor load. However, there are several differences: firstly the cited research papers do not provide any insight into the use of Zabbix for collecting and displaying of results. In this work, Zabbix has been used to collect and display results in graphical and tabular formats. Furthermore, monitoring of bare-metal and hosted hypervisors has not been discussed in any of the cited research papers [44].

Controlling: Research by [45], [46], [47], [48] and [49] focused on controlling virtual machines using various methods and techniques. Each method used migration concept to balance load. In this method, the entire operating system is migrated to a new server either impacting very less or not at all to applications performance. This work also followed the same steps of using XEN to balance loads. But instead of migrating the entire operating system, commands are used provided by SNMP to change the state of the guest operating system for example from running to paused etc. Monitoring results (obtained from methods mentioned in this section) are used, and

virtual machines are controlled through manual commands. Different approaches have been used in various researches. Chen et al. [50] in their research used OpenStack cloud and built their system on KVM hypervisor while this research uses the local system for experiments, and most importantly, XEN is used for all experiments. Nikbazm et al. [53] used a similar approach as in [50]. Even though Nikbazm used XenServer hypervisor, their research mainly focused on network traffic. Furthermore, there is no mention of control features. Another similar approach is also followed by Yang et al. [51]. Yang's research is basically related to the live migration of virtual machines under cloud environments. The system proposed by Yang is based on KVM. No experiments have been performed on the XEN server. Furthermore, virtual machines are either migrated or shut down; no other option such as suspending or resuming is available. Another approach followed to monitor system using SNMP was the implementation of custom-made MIBs. XenMIB and NCNU-VM-MIB were implemented by [52] and Peng et al., [43] respectively. Both these types of research only provide monitoring features, and there is no mention of changing virtual machine parameters which is an integral part of controlling. An approach closest to this thesis work is Hillbrecht et al. [55] work on monitoring and control of virtual machines using SNMP on KVM hosts. Additionally, this related work also used custom-made MIB for monitoring purpose, and their proposed future research direction was to add new management features to the MIB rather than following an automated approach to control virtual machines. Daitx et al. [54] conducted research that main focused on identifying interface problems in virtual networks. Their research basically investigated the use of SNMP in virtual routers as a management interface. They do not provide any theory or experimental evidence on how to control various virtual machines running in XEN or any other hypervisor. Furthermore, their research is targeted towards the configuration of virtual links and virtual hosts. Santos's research is basically an extension to Daitx et al. [54] research. Santos et al. [57] presented their research work to evaluate SNMP, Restful Web services (RWS) and NETConf for Network Virtualization Environment (NVE). Research conducted by Santos et al. (2015) did not present or use any automatic approach for management of virtual routers. Furthermore, their focus was on Virtual Routers rather than virtual machines. Kim et al. [58] used SNMP and Web technology to present a Load Cluster Management System for monitoring and controlling of the entire load cluster group. Many other management approaches other than SNMP were also discussed (e.g. Web-based and REST-based). Web-Based Enterprise Management system was presented for the management of distributed computing environments. Metsch et al. [59] presented a REST-based approach to provide remote management for API for IaaS model based services. Both these approaches come under the umbrella of monitoring and controlling domain, but these approaches do not use SNMP.

Automation: Existing patents suggested automated monitoring virtual machines and using some benchmark to reallocate resources from one virtual machine to other virtual machines. There was also a focus on the reallocation of resources for an application that is running database applications or programmable logic controllers. The concept from patents to automatically monitor virtual machines and CPU utilization as a benchmark to balance load by either changing state of a VM or by creating an entirely new guest operating system has been used. This research concept is aligned to Patents in this regard, but it differs in a sense that it uses a custom-

developed C program to do automation and controlling of virtual machines. SNMP is used as a base for the program, and testing is done on XEN – bare-metal – hypervisor. Patents provide a base for managing resource allocation. A concept is discussed in these patents, that is, creating new virtual machines can balance resource allocation. But these patents do not give an idea about how to automatically achieve this. Furthermore, SNMP is not used in these patents. Therefore, a gap left by research in this regard is filled by the research in this paper. Research conducted by Yang et al. [51] and Guazzone et al. [56] assumed the same objective, i.e. achieving a green cloud computing environment. Both of them presented a different system in order to achieve their goals. Yang et al. called their system Green Power Management (GPM) that reduces power consumption while the system presented by Guazzone et al. reduces power consumption and reduces QoS violations. None of them talked about the use of SNMP in their system. Both of them did not provide anything on the management of virtual machines such as creation, suspension or resumption. Adding SNMP with VM management can facilitate power consumption reduction and load balancing. Minhas et al. [64] research followed some basic concepts of load balancing presented by Yang et al. in their research. In their research, they proposed a system to provide recommended VM configurations based on the workload of database management systems. Their research also does not provide anything on controlling parameters of virtual machines. In other words, creation or resuming of virtual machines is not discussed in their research. Hyser et al. [65] presented an algorithm which follows the basic concept as discussed in this paper, i.e. launch of new virtual machines based on workload. Their research also used an algorithm to maintain balance workload environment among virtual machines.

Energy efficiency: Numerous researches applied the same concept of developing a methodology or technique to efficiently allocate resources in a cloud computing environment. In this regard, some research classified energy efficiency techniques and some research treated tasks as a single unit to find a correlation between system performance and resource utilization. Some research used data-intensive jobs to suggest energy efficiency methodology. Some research used consolidation technique to achieve the same goal of energy efficiency. Stress provided a model to identify the power consumption of each hardware device for both energy aware and energy-unaware guest operating systems. The research in this paper also used the same technique to attain objectives. Power consumption has been used for calculating formula, which provides brief information about energy utilization/consumption by each resource. Difference between various methodology and thesis methodology is that it uses properly defined formulas for result calculations [60], [61], [62], [63].

Virtual Local Area Networks (VLANs) impact: Most research used either enterprise network or campus network to obtain information about VLAN usage so that VLANs advantage or overhead can be decided. Some research used inferencing algorithm to find traffic patterns of VLANs and some used tools like EtherTrace to find out dependency between Layer-2 and layer-3. Results discussed in these research works suggested that virtualization causes inefficiencies in performance and VLANs complicate network configuration management. Research carried out on VLANs in this paper; also do similar work, i.e. obtaining CPU utilization for various VLANs configurations. Results discussed as part of this work also suggested that VLANs add overhead and cause extra CPU utilization. However, other existing work has not used various stress/workloads to find out the impact of VLANs on CPU utilization.

Experiments have been conducted on various VLANs configurations to further support the results.

3 Methodology

The concept of a cloud environment can be described as hundreds of VMs running on multiple physical servers. Monitoring of VMs in that environment becomes important because monitoring results can provide us with useful information about the energy consumption of CPUs or overheads for implementing features such as VLANs. Once relevant data are gathered and analysed, the finding can inform the control of VMs in order to reduce energy consumption. Controlling can be done either manually or automatically using scripted codes. This section discusses the research methodology, which encompasses different sets of experiments conducted. Monitoring is the first phase of this work. SNMPv3 protocol is used to monitor resource consumption of bare-metal hypervisors and virtual machines of both bare metal and hosted hypervisors. VMware ESXi 5.5 – bare-metal hypervisor – is installed on HP machine server and two virtual machines are configured on VMware ESXi. SNMPv3 protocol monitors the resource consumption for each VMs. Monitoring of the hypervisor provides information on what resources are being consumed by each VMs. Results of monitoring feed into the control of VMs. Controlling of VMs can be achieved using controlling features of SNMPv3. SNMP provides various commands and parameters to change the state of VMs, i.e. from running to paused, paused to shut off or paused to running. Controlling is very important in order to reduce energy consumption by VMs. While using controlling features and commands of SNMPv3, the state of VMs can be changed, but that process is done manually. In order to better utilize monitoring results, automation for controlling features is essential. Automation facilitated through C programs which utilize Libvirt API functions. Changing states of VMs or creating a new virtual machine based on the average load is part of this C program. Use of Virtual Local Area Networks (VLANs) is an important concept in cloud computing. Monitoring and analyzing the impact of VLANs on CPU utilization in a cloud environment is necessary. Microsoft Hyper-V is installed on Windows Server R2 with various VMs each having 2GB RAM and 1-24 processors. Stress command is used to put a load on VMs. Results generated are then plotted and analyzed for further decisions.

3.1 Experimental setup

The testbed for SNMP version 3 monitoring and controlling VMs was created on a hypervisor and is implemented on two physical machines with several virtual machines, where Processor, Network, Storage and Memory performance was monitored. Step by step installation and configurations was also described:

1. To conduct research experiments on bare-metal hypervisor, VMware ESXi 5.5 was installed on HP ProLiant DL380 Gen8 server.

2. Bare metal hypervisors run directly and do not need an Operating System installed on the machine. Bare metal hypervisor configuration: the laptop is directly connected to an HP server running VMware hypervisor.
3. Zabbix was installed on a laptop, and two VMs have configured on VMware ESXi 5.5 hypervisor. Zabbix Server is a Linux based monitoring tool, and VMware ESXi is Windows-based.
4. To solve this issue, Zabbix server was installed inside Ubuntu using Virtualbox.

Stored values in various object IDs were collected and displayed in Zabbix with graphs. Monitoring of hypervisors and VMs played an important part in controlling and automation. Controlling virtual machines can be done either manually or automatically. However, before conducting any experiment with automation, it is important to first explore SNMP with manual controlling. This work discussed experiments that helped to explore and identify controlling features of SNMP (e.g. change of VMs state). To start with controlled experiments, the configuration of Libvirt-SNMP with Xen was briefly outlined. Manual commands such as SNMP table and SNMP set were used to get information about virtual machines and change the state of virtual machines accordingly. State of virtual machines was changed from running to paused, and shutdown with the help of SNMP commands. Results of these experiments clearly showed that SNMP could control VMs. The methodology adopted to monitor running domains and controlling them was with using Libvirt API. In the experiments, the authors extended the capability of SNMP by automating the control and management of virtual machines through the design and implementation of an algorithm in C language code. The experiments could be summarized in 4 set-up and were performed 10 times with average used for graph plotting:

1. **Create VMs:** Expected results are an increase in CPU utilization whenever a new virtual machine is created. For this purpose, create VM experiments were performed with new virtual machine being created after each 60 seconds (see Fig. 1);
2. **Suspend VMs:** Expected results are a decrease in CPU utilization whenever a virtual machine is suspended. For this purpose, suspend VM experiments were performed with one virtual machine being suspended out of 24 running virtual machines after each 60 seconds (see Fig. 2 and Fig. 3);
3. **Resume VMs:** Expected results are an increase in CPU utilization whenever a virtual machine is resumed. For this purpose, resume VM experiments were performed with one virtual machine being resumed out of 24 suspended virtual machines after each 60 seconds (see Fig. 4 and Fig. 5);
4. **IDLE state:** Expected results are no or slight change in CPU utilization whenever no action is performed. For this purpose, idle state experiments were performed with new virtual machine being created and monitored for 5 minutes (see Fig. 6 and Fig 7).

3.2 Mathematical model

In various experiments, we have calculated the relation between CPU utilization and multiple parameters such as the number of processes (stress) being run on Hypervisor and number of VMs and their state. Each experiment provided a relation,

and using that relation; a mathematical model is developed. The graph in Fig. 8 between CPU utilization and the number of processes gives the equation between processes and CPU utilization.

$$CPU\ Utilization = Y = 1.92 \times P + C1 \quad \text{Eq. 1}$$

Where, P is the number of (similar) processes running on CPU and C1 is the load on CPU from other processes. Similarly, the graph between CPU utilization and VMs at Idle state from Figure 7 gives an equation between the number of VMs and CPU utilization. The idle state is selected for finding out equation because it provides a better relationship between VMs and CPU.

$$CPU\ Utilization = Y = 2.787 \times VMs + C2 \quad \text{Eq. 2}$$

Where, VMs is number of virtual machines in idle state (without any stress) and C2 is load on CPU from other processes. Both equations (1 and 2), give the relationship between CPU utilization, stress, and VMs in idle state. It's crucial to determine the relationship between CPU utilization and virtual machines when VMs are provided under pressure. Later on, this can help optimize the system for selecting virtual machines from the pool of VMs to perform some task. For this purpose, a similar set of experiments was performed, but this time VM was fixed at 1 while stress on VM was changed from 0 to 10.

Figure 1. CPU Utilization for 2-VMs Suspend

Figure 2. CPU Utilization for Resume VM Action

Figure 3. CPU Utilization for 2-VMs Resumed

Figure 4. CPU Utilization for Idle State

Figure 5. CPU Utilization at Idle State

Figure 6. CPU Utilization with Stress Command

Figure 9 shows that, for each case (without stress on CPU and with CPU stress), CPU utilization increased at a rate of 0.5 with increased pressure on the VM. So, equation to calculate the relationship between CPU utilization and stress on virtual machines can be written as:

$$CPU\ Utilization = Y = \sum_{i=0}^n 0.5 \times S_i + C3 \quad \text{Eq.}$$

Where: n is total number of VMs, S_i is stress on i th VM, C3 is load on CPU from other factors such as stress on CPU and/or load of VM. The combination of 3 equations provides a complete mathematical model to calculate CPU utilization with stress on CPU and VMs:

$$CPU\ Utilization = Y = (1.92 \times P) + (2.787 \times VMs) + \sum_{i=0}^n 0.5 \times S_i + C \quad \text{Eq.4}$$

Figure 7. CPU Utilization at 1 VM

Figure 8. CPU Utilization vs Number of VMs

A comprehensive set of experiments were performed to evaluate the mathematical model outlined. These experiments included increasing stress put on virtual machines. Each experiment included two different types of stress. One pressure being put on CPU and one stress being put on VMs. Pressure on virtual machines ranges from 1 to 10, while CPU stress included only two variations: 0 and 4. Each such experiment was repeated 5 times, and the average value was calculated. A comparative graph is plotted in Figure 10 and discusses in the following. In this graph (Fig. 9), each line represents the relation between CPU utilization and the number of VMs under specific stress. Series Stress_1 represented CPU utilization when VMs were put under the stress of 1. Another critical point here is that, with the increase in VMs, the stress factor was reduced. For example, when only 4 virtual machines were running, stress factor was changed from 0 to 10, but when the total number of running virtual machines increased to 24 then stress factor was changed from 0 to 1. This is because each virtual machine put a load of nearly 2.8 times on CPU (as per Equation 2) and placing stress of 10 on each VM will exceed 100% CPU utilization (theoretically, as per Equation 3). But in practical CPU utilization cannot exceed 100% therefore, it is capped to some limited percentage which can be anywhere between 90 and 100. That's why at a higher number of VMs, the stress factor is kept low.

4. Results and Discussion

As previously discussed, we have conducted 4 experiment set-ups which demonstrated the concept of CPU utilization. Experiments conducted in the preceding section attempt to verify four posits relating to the change in CPU utilization with respect to different states (create, suspend, resume, and idle) and number of virtual machines. The experimental phase presented four experiments that discussed the relation between the number of virtual machines and CPU utilization. Each experiment was executed ten times with different predefined scheme of actions performed in each iteration. CPU utilization against the number of virtual machines and their states was plotted and presented in graphs. The first experiment showed an increase in the number of virtual machines brought about an increase in CPU utilization. The second experiment showed an increase in the number of virtual machines being suspended in CPU utilization. The third experiment showed that CPU utilization increased when virtual machines were resumed. The fourth and last experiment revealed that CPU utilization did not change nor featured any slight change when no action was performed on virtual machines. A mathematical model was formulated based on these experiments that presented a clear relationship between CPU Utilization, Total number of running VMs and Stress on each VM. The evaluation of the mathematical model demonstrated that certain points in CPU utilization could be achieved through different number of virtual machines and stress on each virtual machine. To further investigate the relationship between power consumption and CPU utilization, two similar experiments were performed. The first experiment investigated and demonstrated a linear relation between power consumption and the number of virtual machines. The other experiments extended the first experiment to calculate power consumption against the number of VMs under varying stress factors. A mathematical model is developed and evaluated in the previous section for the relation between the number of virtual machines and CPU utilization. Graphs and result discussion clearly stated that CPU utilization increases with an increase in the number of VM. In other words, the CPU utilization is directly proportional to the number of VMs running. It is essential to find a relation between the number of VMs and power consumption. For this purpose, the authors carried out a similar set of experiments but not included in this work. In particular, a set of experiments included creating a new virtually every five minutes and recording power consumption in watts using the energy-saving monitor.

4.1 Energy Efficiency management

This section explains the results of research experiments which aimed to investigate the server's energy utilization using SNMPv3. Sets of experiments relating to Automation revealed that power consumption is directly related to the number of VMs running on a hypervisor. Therefore, it is crucial to monitor and track each processor's average CPU Utilisation and overall total power consumption. A single VM acts as a process, and it puts some load on the server. In this section, the average CPU utilization of server was calculated for standard load and heavy load over 24 hours. Average load implied that a system was in an idle state or system was observed without executing stress command. Heavy load refers to a condition whereby a system was put under stress command, or multiple applications were concurrently

running. Results were calculated and plotted using Zabbix, which showed that the average CPU utilization of host machine increased whenever a heavy load was applied, which created new VMs. Hence, it could be concluded that with more VMs, load on the host machine increases, which affects more consumption of power. This section discusses the total power consumption of CPU and underlines as the CPU utilization plays an important role in energy efficiency improvement. Lower CPU utilization means lower power consumption or improved energy efficiency. For the experimental set up to manager, the energy efficiency in a cloud computing environment has used the SNMPv3. The work focuses on monitoring various CPU parameters to determine the machine's power utilization due to computer processing. HP ProLiant DL380p Gen8 Server with Vmware ESXi – bare-metal hypervisor – was used for experimentations and data was collected using SNMPv3 commands. Zabbix server was used to extract values from the server and save them in a database. In the following the authors present the experimental' campaign steps starting to an observed list of parameters to manage energy consumption within a server.

- 1) CPU Utilization of Processors Using SNMPv3: individual processors on the server machine were monitored to obtain CPU utilization. HP ProLiant server provides a total of 24 physical processors chips, and each processor's utilization was logged at specific intervals over some time. Data collected in this case was for an average load.
- 2) Average CPU Utilization: collecting average CPU utilization, Zabbix was used because multiple values were being collected every minute over a period which provided an accurate account of the average CPU utilization for the 24 processors.
- 3) Calculation of CPU utilization: various timing parameters has been used as shown in the following formula (Equation 5):

$$CPU\ Usage = 100 \times \left(\frac{utime + stime + nicetime}{totalTime} \right) \quad \text{Eq.5}$$

Where: **sTime** (System Time) is the time used by processor for system tasks; **uTime** (User Time) is the time used by processor for user tasks; **niceTime** (Nice Time) is the time used for low priority and background tasks and **totalTime** (Total Time) is the total running time of processors.

- 4) Process CPU Utilization: a number of processes running on a server at any given time and this means the CPU is utilized. The following formula (Eq. 6) was used to calculate processor CPU utilization.

$$Processor\ CPU = 100 \times \left(\frac{utime + stime}{totalTime} \right) \quad \text{Eq. 1}$$

- 5) Total Power Consumption of CPU: it can be calculated using the Eq. 7 (Blackburn, 2008):

$$Power\ Consumed = (P_{max} - P_{idle}) \times \left(\frac{CPU\ Usage}{100} \right) + P_{idle} \quad \text{Eq. 2}$$

Where: **P_{max}** is the Power utilization index under heavy load and **P_{idle}** is the Power utilization index under stress free.

4.2 Impact of VLANs in Hypervisors

This section focuses on the investigation of the impact of Virtual Local Area Networks (VLANs) on CPU utilization in a cloud environment. For experimentation, a test environment was established with Microsoft Hyper-V installed on Windows Server 2012 R2. Each virtual machine was created with 2048MB RAM and 1-24 virtual processors. Stress command was used to put the load on VMs. Stress command is a simple load generator for POSIX systems written in C and licensed under GPL-v2; it inflicts a configurable amount of CPU, memory, I/O, and disk stress on the network. The experimental campaign was carried out to explore average CPU utilization when:

- No-VLAN configuration was used. The aim of this experiment was to provide a benchmark for CPU utilization when there is no Virtual Local Area Network. The experiment was run five times for processor cores ranging from 1 to 24.
- 2, 3, 4, 5, 6, 7 VLANs configuration was used; an inter-comparison average CPU utilization of n-VLANs under various VM configurations obtained from running various experiments has provided.

For each VLAN, 7 Virtual Machines were used with various configurations to observe nature of VLANs. When plotted for average CPU consumption for each VLAN, an approximately straight line was observed. As shown in graph (Figure 9).

Figure 9. Average CPU Consumption for VLANs

As shown in Figure 11, with an increased number of VLANs, average CPU consumption almost remained the same regardless of Virtual Machine configuration. Hence, growing VLANs to the much greater number will not change VLANs' impact on CPU consumption. And VLAN number cannot be increased beyond 4096. Because there are only 12 Bits used for VLANs in 802.1q so only, $2^{12} = 4096$ VLANs are used starting from 0 to 4095. 0 and 4095 are reserved (Thaler, et al., 2011). The CPU consumption graph seems to be horizontally linear. This implies that the value of y appears to be constant at $y = 18.5\%$, now even at $x = 4095$. In conclusion, the increase in the number of VLANs seems not to affect the value of y (CPU consumption).

Monitoring and investigating the impact of VLAN on CPU utilization using Simple Network Management Protocol (SNMP). CPU utilization was recorded and plotted in graphs for various virtual local area networks (VLANs) configurations for 2-7 VLANs. Furthermore, CPU utilization was compared for VLAN and No-VLAN configurations. Results showed that having different virtual machine configurations had no effect on CPU utilization for the same VLAN configuration. Results also showed that having VLAN for any design put a load on CPU and caused overhead

compared to No-VLAN configuration. To better understand and concrete analysis of experimental results, statistical methods such as T-tests have also been used, which also demonstrated the same outcome as in graphs. Experiments conducted in this chapter provided us with the opportunity to monitor CPU utilization for Hyper-V hypervisor.

.5. Conclusions and Future Works

An environment with multiple physical servers and several virtual machines running requires monitoring VMs with minimal overheads. To monitor the consumption of resources via monitoring hypervisor alone or hypervisor and VMs is a significant challenge. Monitoring will provide informed decisions for the control of VMs to reduce the amount of energy consumed by each VM. The primary controlling feature includes a change of virtual machine state from running to paused or shut down, remained to shut down or to run. An essential aspect of control is exploring if it is beneficial to manage energy efficiency and balance load on different VMs. If controlling can decrease energy consumption by changing states of VMs, it can be extended to an environment where multiple servers with several VMs are running. Controlling in such environment through manual commands is not a feasible solution. Therefore automation is required in such a scenario. Automating such a system requires a system built on particular requirements such as creating a new VM when the load exceeds a certain threshold to balance load or shutting down a running VM to control resource consumption. Power up and powering down VMs based on automatic demand is critical to know if it can reduce energy consumption. The energy consumption of today's systems is rapidly increasing with the growth of digital devices usage. Consequently, applications executed on those digital devices increasingly demand more resources, including power. It is useful to determine resource consumption and power utilization by these systems to make them energy efficient. Another critical concept is the use of Virtual Local Area Network (VLANs) in the cloud computing environment. In enterprise networks, VLANs are widely utilized to group hosts together. A vital advantage of using VLANs is that it provides flexibility to allow any logical LAN to be implemented on any physical infrastructure. Enterprise network administrators often use VLANs to group users and control access to resources. There are more than fifty VLANs in use, even in small enterprise networks. Monitoring and analysing the impact of VLANs on CPU utilization in a cloud computing environment is vital. Monitoring impact of VLANs provides an answer to two basic but fundamental questions: whether or not to implement VLANs in a network environment. And what type of configuration is suitable for implementing VLANs. SNMPv3 has been used to monitor the bare and hosted hypervisors, which enabled us to know resource consumption by each virtual machine. However, to determine the VMs' respective resource consumption, each of them is to be monitored individually. VMs configured on hosted or type-II hypervisors were also monitored, and results were plotted in graphs using data collected by Zabbix. SNMPv3 provides various commands to manage virtual machines running on the bare and hosted hypervisor. These commands can be used to monitor and change the state of virtual machines running on XEN – bare-metal –

hypervisor. Controlling virtual machines enabled us to reduce energy consumption and resources by VMs. Monitoring virtual machines and hypervisors was a vital part of making decisions about controlling VMs. Owning and managing VMs using SNMPv3 commands was quite useful to reduce energy consumption but automating this procedure helped perform this task for any number of VMs. Controlling VMs can be done either manually or automatically. To date, there is limited experimental data available related to automated controlling of VMs. However, for any case, monitoring results were required to make decisions on VMs' status. The work's core contribution is the development of C-language based utility using Libvirt API that monitors and changes VMs based on CPU utilization. The key concept for this utility is based on experimental results of monitoring and controlling. Monitoring of virtual machines served as a testbed for implementation of SNMP version 3 on two physical machines with multiple virtual machines running. Memory, processor, network and storage performance was monitored to collect and analysis for controlling experiments. Controlling can be done either manually or with the use of scripts to automate the process. However, for both cases, monitoring results are required to make decisions on virtual machines status. Results from controlled experiments discussed that SNMP could be used to manage virtual machines running in XEN hypervisor. The first phase for utility development included testing of controlling based on manual execution of various commands. For this purpose, controlling experiments provided a base to work on Libvirt-SNMP orders to obtain virtual machines information and control them. Manual commands such as SNMP table and SNMP set used to get information about virtual machines and change the state of virtual machines accordingly. State of virtual machines was changed from running to paused, and shutdown with SNMP commands' help. Results of these experiments clearly showed that SNMP could control virtual machines. This utility extended the SNMP controlling feature to automatic management of virtual machines.

Power consumption concept was mainly demonstrated by this utility through a set of experimental results. To relate power consumption with virtual machines, experiments were carried out in three phases. The first phase of experiments helped to obtain average CPU consumption with changing virtual machines status. These experiments demonstrated that whenever a new VM was created or resumed from suspension state, the average CPU consumption was increased. Experiments in the second phase included eight different experiments that discussed the relation between the number of virtual machines and power consumption. Fifty iterations following a different scheme of actions were used for each experiment. Graphs were used to present and plot the relation between power consumption and the number of virtual machines. This phase's results showed that with the increase in the number of virtual machines, power consumption was also increased. The third and last phase of experiments explored the power consumption by a hypervisor when the load is put on virtual machines. Four experiments were executed in this phase for the different number of virtual machines and varying load. Results of these experiments were plotted for power consumption and virtual machine load. These experiments helped find out the fact that hypervisor's power consumption increases with the increase in load on virtual machine. To prove this fact, statistical analysis was also done based on data collected through these experiments. Z-Test was performed and explained for different categories, i.e. VM create, VM resume, VM suspend, no-action. Statistical analysis further proved the findings outlined in experimental results.

Energy Efficiency, processes and applications efficiency can be determined by observing the various data from the device. One approach used to monitor energy utilization of server using SNMPv3. For this purpose, HP ProLiant server with 24 physical processors chips was used to obtain CPU utilization. Data for each processor was logged at specific intervals over 24 hours under both heavy and normal load. Various equations were used to calculate overall CPU usage, individual processor CPU utilization and total power consumption. These equations covered different time parameters such as system time, user time, nice time and total time and power utilization parameters such as Pmax and Pidle. Data from individual processor indicated varied CPU utilization and dependency of processing distribution among the processors. The average total CPU utilization over 24 hour's period showed an increase in CPU utilization in case of high load compared to low load. By comparing CPU utilization and power consumption of the CPU, it was indicated that there is power consumption by the CPU even when there is normal or low load. However, the power consumption of server providing cloud services depends upon the load on its processor. Results obtained through experiments were plotted in the form of graphs using Zabbix. Graphs presented for this purpose demonstrated that the average CPU utilization of host machine increased whenever a heavy load was applied. Monitoring and finding the impact of VLAN on CPU utilization using Simple Network Management Protocol (SNMP) helped us get to know about VLANs' impact on the server. CPU utilization was recorded and plotted in graphs for various virtual local area networks (VLANs) configurations for 2-7 VLANs for Hyper-V hypervisor. Furthermore, CPU utilization was compared for VLAN and No-VLAN configurations. Results showed that having different virtual machine configurations have no effect on CPU utilization for the same VLAN configuration. Results also showed that having VLAN for any configuration put the load on CPU and caused overhead compared to No-VLAN configuration. Statistical methods have been used to get concrete analysis and a better understanding of results. For this purpose, one-way ANOVA and T-Tests have been used that also demonstrated the same outcome as plotted in graphs. Core experiments were performed under a single server environment and results obtained were according to expectations. These experiments provided a platform to explore further options in SNMP category in the multi-server environment. Researching controlling and automation under multi-server environment can help to manage energy efficiently. Automation experiments can be expanded to act like an Agent connected with Zabbix to automatically record monitoring and control of virtual machines in MySQL database, making more efficient decisions based on the more prolonged duration of resources utilization. Other monitoring tools such as PRTG, Nagios and Pandora FMS can also be used to replace Zabbix for results comparison and analysis. Each tool provides a different set of features; hence, results comparison between various monitoring tools can lead to deciding the best tool for experiments with least overhead. Another prospect is the comparison of results obtained from XEN hypervisor with other Hypervisors. One such hypervisor is KVM. KVM is a popular tool as XEN, and a lot of similar research has been done on KVM. Results obtained from KVM experiments will provide useful analysis on the selection of hypervisor that more suitable for automation experiments. Development and execution of an automation algorithm with different programming languages on other Hypervisors can prove an interesting future topic. Python is one such programming language that is becoming popular among many researchers.

Implementing automation algorithm using Python as a base programming language can prove to be useful future research direction. The current implementation is in C language, which does not seem to put any considerable load on the system. Therefore, implementing the same functions in Python might not affect the performance of the system. However, the efficiency of languages could be future work. Additionally, implementing algorithms in different languages such as Python and C could render the system more versatile. Another promising prospect can be extending automation code to virtual LANs to identify VLANs' impact on power consumption and identify VLANs' best possible configuration for different scenarios.

References

1. Black, U. D., 1994. Network Management Standards: SNMP, CMIP, TMN, MIBs and Object Libraries. New York, NY, USA: McGraw-Hill, Inc.
2. Hämäläinen, S., Sanneck, H. & Cinzia Sartori, e., 2012. Autonomic Computing and Autonomic Management. In: LTE self-organising networks (SON): network management automation for operational efficiency. s.l.:John Wiley & Sons, p. 55.
3. Hunt, R., 1997. SNMP, SNMPv2 and CMIP—the technologies for multivendor network management. Computer Communications, pp. 73-88.
4. Mayfield, V. & Erickson, L., 2016 . The Value of Real-Time Monitoring Your Network.
5. GreenIT, 2016. Green Technology. [Online].
6. Murugesan, S. & Gangadharan, G. R., 2012. Harnessing Green IT. Principles and Practices. s.l.:Wiley and Sons Ltd.
7. Ram, A. & Murugan, M. B., 2018. Energy Estimation of Embedded Systems. In: Intelligent Embedded Systems. s.l.: Springer, pp. 275-284.
8. Ding, J., 2016. Advances in network management. s.l.: CRC press.
9. Zhou, E. et al., 2014. Network management system. Google Patents | US Patent 8,782,282.
10. Gavalas, D., Greenwood, D., Ghanbari, M. & O'Mahony, M., 2000. Advanced network monitoring applications based on mobile/intelligent agent technology. Computer Communications, pp. 720-730.
11. Bowman, W. B., 1997. System and method for detecting fraudulent network usage patterns using real-time network monitoring. US Patent 5,627,886.
12. Hunt, R., 1997. SNMP, SNMPv2 and CMIP—the technologies for multivendor network management. Computer Communications, pp. 73-88.
13. Harrington, D., 2003. SNMP Version 3. [Online]. Available at: <https://datatracker.ietf.org/doc/charter-ietf-snmpv3/>
14. Stallings, W., 1998. SNMP and SNMPv2: the infrastructure for network management. IEEE Communications Magazine, pp. 37-43.
15. Case, J., Mundy, R., Partain, D. & Stewart, B., 2002. Introduction and applicability statements for internet-standard management framework.
16. Feit, S. M., 1993. SNMP: A guide to network management. s.l.:McGraw-Hill Professional.
17. Harnedy, S. J., 1996. Total SNMP: exploring the simple network management protocol. s.l.: CBM Books.
18. Schmid, K. & D. M., 2005. Chapter 1. Introduction to SNMP and Network Management. In: Essential SNMP. s.l.: O'Reilly Media, Inc.
19. Mell, P. & Grance, T., 2011. The NIST definition of cloud computing. pp. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg.

20. Qian, L., Luo, Z., Du, Y. & Guo, L., 2009. Cloud computing: An overview. *Cloud computing*, pp. 626-631.
21. Foster, I., Zhao, Y., Raicu, I. & Lu, S., 2008. Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop, 2008. GCE'08*, pp. 1-10.
22. ComputerWeekly, 2016. A history of cloud computing. [Online]. Available at: <http://www.computerweekly.com/feature/A-history-of-cloud-computing>
23. Cantu, A., 2011. The history and future of cloud computing. *Forbes online*, p. 2012.
24. Vikas, S., Gurudatt, K., Vishnu, M. & Prashant, K., 2013. Private vs public cloud. *International Journal of Computer Science & Communication Networks*, p. 79.
25. Khasnabish, B. et al., 2011. Cloud Reference Framework draft-khasnabishcloud-reference-framework-02.
26. Curry, J., 2010. Introducing OpenStack. [Online]. Available at: www.openstack.org
27. Sefraoui, O., Aissaoui, M. & Eleuldj, M., 2012. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*.
28. OpenStack, 2016. OpenStack Summit | Austin 2016. [Online]
29. Pepple, K., 2011. *Deploying openstack*. s.l.:O'Reilly Media, Inc.
30. Golden, B., 2011. *Virtualization for dummies*. In: s.l.:John Wiley & Sons, p. 54.
31. Singh, H. & Yip, M., 2018. *NEXT-GEN VIRTUALIZATION FOR DUMMIES - VMware Special Edition*. s.l.:John Wiley & Sons.
32. Morabito, R., Kjällman, J. & Komu, M., 2015. Hypervisors vs. lightweight virtualization: a performance comparison. In: *Cloud Engineering (IC2E), 2015 IEEE International Conference on*. s.l.:s.n., pp. 386-393.
33. Golden, B., 2011. *Virtualization for dummies*. In: s.l.: John Wiley & Sons, p. 54.
34. Popek, G. J. & Goldberg, R. P., 1974. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, pp. 412-421.
35. Smith, J. E. & Nair, R., 2005. The architecture of virtual machines. *Computer*, pp. 32-38.
36. Ferrari, M., 2014. Getting Started with OpenStack and VMware vSphere. *virtualization.info*.
37. Logan, K., 2015. The Advantages and Disadvantages of Virtualization. [Online]. Available at: <http://milner.com/company/blog/technology/2015/07/14/the-advantages-and-disadvantages-of-virtualization>
38. Schoffstall, J. C. M. F. M. & Davin, C., 1990. RFC 1157: Simple network management protocol (SNMP). IETF, April.
39. Waldbusser, S., "Remote Network Monitoring Management Information Base", RFC 1757, 1995.
40. Claise, B., 2004. RFC 3954. Cisco systems netflow services export version.
41. Ongaro, D., Cox, A. L. & Rixner, S., 2008. Scheduling I/O in virtual machine monitors. s.l., ACM, pp. 1-10.
42. Cherkasova, L. & Gardner, R., 2005. Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor.. *USENIX Annual Technical Conference, General Track, Volume 50*.
43. Peng, Y.-S. & Chen, Y.-C., 2011. SNMP-based monitoring of heterogeneous virtual infrastructure in clouds. *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pp. 1-6.
44. Iqbal, A., Pattinson, C., & Kor, A. L., 2015. Performance monitoring of Virtual Machines (VMs) of type I and II hypervisors with SNMPv3. *2015 World Congress on Sustainable Technologies (WCST)*, pp. 98-99.
45. Kaneda, S. & Murakami, K., 1989. US, Patent No. 4860190.
46. Clark, C. et al., 2005. *Live migration of virtual machines*. s.l., USENIX Association, pp. 273-286.
47. Barham, P. et al., 2003. Xen and the Art of Virtualization. *Proc. of the 19th ACM Symp. on Operating Systems Principles*, p. 164-177.

48. Nelson, M., Lim, B.-H. & Hutchins, G., 2005. Fast Transparent Migration for Virtual Machines.. s.l., s.n., pp. 391-394.
49. Nagarajan, A. B., Mueller, F., Engelmann, C. & Scott, S. L., 2007. *Arun Babu Nagarajan, Frank Mueller, Christian Engelmann, Stephen L Scott*. s.l., ACM, pp. 23-32.
50. Chen, C.-C. et al., 2015. Construction of a Green Power Virtual Machine Management Method on OpenStack. In: COMPSAC, IEEE 39th, pp. 627-630.
51. Yang, C.-T., Liu, J.-C., Chen, S.-T. & Huang, K.-L., 2017. Virtual machine management system based on the power saving algorithm in cloud. *Journal of Network and Computer Applications*, Volume 80, pp. 165-180.
52. Schonwalder, J. & Muller, A., 2001. Reverse engineering internet MIBs. In: *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*. s.l.:IEEE, pp. 423--436.
53. Nikbazm, R., Dashtbani, M. & Ahmadi, M., 2015. Enabling SDN on a special deployment of OpenStack. In: *Computer and Knowledge Engineering (ICCKE), 2015 5th International Conference on*. s.l.:IEEE, pp. 337--342.
54. Daitx, F. F., Esteves, R. P. & Granville, L. Z., 2011. On the use of SNMP as a management interface for virtual networks. In: *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. s.l.:IEEE, pp. 177-184.
55. Hillbrecht, R. & de Bona, L. C. E., 2012. A SNMP-based virtual machines management interface. *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, pp. 279-286.
56. Guazzzone, M., Anglano, C. & Canonico, M., 2012. Exploiting VM migration for the automated power and performance management of green cloud computing systems. In: *International Workshop on Energy Efficient Data Centers*. s.l., Springer, pp. 81-92.
57. Santos, P. R. d. P. F., Esteves, R. P. & Granville, L. Z., 2015. Evaluating SNMP, NETCONF, and RESTful web services for router virtualization management. In: *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. s.l.:IEEE, pp. 122-130.
58. Kim, M.-S., Choi, M.-J. & Hong, J. W., 2002. A load cluster management system using SNMP and web. *International Journal of Network Management*, pp. 367--378.
59. Metsch, T. & Edmonds, A., 2010. Open cloud computing interface-infrastructure. In: *Standards Track, no. GFD-R in The Open Grid Forum Document Series, Open Cloud Computing Interface (OCCI) Working Group, Muncie (IN)*. s.l.:s.n.
60. M. Chinnici, A. Capozzoli, and G. Serale: Measuring energy efficiency in data centers. In *Pervasive Computing Next Generation Platforms for Intelligent Data Collection*, Chapter 10, pp. 299-351, 2016.
61. M. Chinnici, et al: Data Center, a Cyber-Physical System: Improving Energy Efficiency Through the Power Management,” in *2017 IEEE (DASC/PiCom/DataCom/CyberSciTech)*, 2017, pp. 269–272.
62. M. Chinnici, et al: Measuring energy efficiency in data centers, Book chapter, *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection*, 2016, pp. 299–351
63. M.Chinnici et al: A machine learning solution for data center thermal characteristics analysis, *Energies*, 2020, 13(17), 4378
64. Minhas, U. F. et al., 2010. Automatic virtual machine configuration for database workloads. *ACM Transactions on Database Systems (TODS)*.
65. Hyser, C. D. & McKee, B. A., 2012. Starting up at least one virtual machine in a physical machine by a load balancer. s.l.:Google Patents US Patent 8,185,893.

Figures

Figure 1

CPU Utilization for 2-VMs Suspend

Figure 2

CPU Utilization for Resume VM Action

Figure 3

CPU Utilization for 2-VMs Resumed

Figure 4

CPU Utilization for Idle State

Figure 5

CPU Utilization at Idle State

Figure 6

CPU Utilization with Stress Command

Figure 7

CPU Utilization at 1 VM

Figure 8

CPU Utilization vs Number of VMs

Figure 9

Average CPU Consumption for VLANs

Image not available with this version

Figure 10

Image not available with this version.

Image not available with this version

Figure 11

Image not available with this version.