Lappeenranta University of Technology
School of Engineering Science
Erasmus Mundus Master's Programme in Pervasive Computing & Communications for Sustainable Development (PERCCOM)

Madhubala Ganesan

ENERGY-EFFICIENT CLOUD INFRASTRUCTURE FOR IoT BIG DATA PROCESSING

Examiners: Prof. Jari Porras (Lappeenranta University of Technology)
Prof. Eric Rondeau (University of Lorraine)
Assoc. Prof. Karl Andersson (Luleå University of Technology)

Supervisors: Professor Colin Pattinson (Leeds Beckett University)
Dr. Ah-Lian Kor (Leeds Beckett University)

This thesis is prepared as part of an European Erasmus Mundus Programme

PERCCOM - PERvasive Computing & COMmunications for sustainable development.

Co-funded by the Erasmus+ Programme of the European Union

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement).

Successful defense of this thesis is obligatory for graduation with the following national diplomas:

- Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Technology (Lappeenranta University of Technology)
- Master of Science in Computer Science and Engineering, specialization in Pervasive Computing and Communications for Sustainable Development (Luleå University of Technology)

ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Erasmus Mundus Master's Programme in Pervasive Computing & Communications for Sustainable Development (PERCCOM)

Madhubala Ganesan

Energy-efficient Cloud Infrastructure for IoT Big Data Processing

Master's Thesis

76 pages, 40 figures, 6 tables, 4 appendices

Keywords: IoT, Big Data Analytics, Cloud Data Centers, VM consolidation, Energy consumption

Internet of Things (IoT) along with Big Data Analytics is poised to become the backbone of Smart and Sustainable Systems which bolster economic, environmental and social sustainability. Cloud-based data centers provide computing power to churn out valuable information from voluminous IoT data. Multifarious servers in the data centers turn out to be the black hole of superfluous energy burn contributing to 23% of the global Carbon dioxide (CO_2) emissions in ICT industry. IoT energy concerns are addressed by researches carried out on low-power sensors and improved Machine-to-Machine communications. However, cloud-based data centers still face energy–related challenges. Virtual Machine (VM) consolidation is an approach towards energy efficient cloud infrastructure. Although several works show convincing results of the potential of VM consolidation in simulated environments, there is inadequacy in terms of investigations on real, physical cloud infrastructure for big data workloads. This work intends to evaluate dynamic VM consolidation approaches by combining algorithms from literature. An open source VM consolidation framework, Openstack NEAT is adopted and experiments are conducted on a Multi-node Openstack Cloud with Apache Spark as Big data platform. This work studies the performance based on Service Level Agreement (SLA) metrics and energy usage of compute hosts. The corresponding results are presented based on which the best combination of algorithms is recommended.

ACKNOWLEDGEMENT

To begin with, I would like to thank my supervisors Prof. Colin Pattinson and Dr. Ah-Lian Kor at Leeds Beckett University. They let me to work independently and offered expert advice as and when needed. I would cherish all those thesis meetings and the consultation hours we had. A special mention to Ah-Lian who took the role not only as my supervisor but also as a personal mentor and well-wisher.

Special thanks to Mr. Abiodun B Yusuf, Doctoral student of Leeds Beckett University for providing ideas, support and access to use his research 'Flight Simulation, Eye Tracking, and In-Flight Startle' experimental set up as the IoT system.

This acknowledgement would be incomplete if I do not extend my sincere gratitude to Prof. Eric Rondeau, the PERCCOM programme coordinator, who is nothing short of my personal guardian in Europe. From the first time I met you to until now, you have extended your infinite support and my thanks wouldn't be enough.

To Prof. Jari Porras, and Prof. Karl Andersson, the country coordinators and great Professors, thank you for sharing your expertise and for encouraging me come up with new ideas throughout the masters' programme.

To all my PERCCOM classmates (Cohort 4), the journey I had with you in the last two years was amazing. You are great peers and I feel gifted to have worked alongside you on various projects and a big thanks to all those great times we had. My best wishes to each and every one of you to follow your passion and come out with flying colors.

To the previous PERCCOM cohorts, you have been great seniors who shared your experiences and offered suggestions as friends, mentors and as seniors.

To my beloved father who have always encouraged me to pursue my passion in this field and persuaded me to go after my dreams despite all odds. I can never thank you enough. To my mother who kept me strong, pushed me forward when I lost all hope, thank you!

Special mention to my best friend, Nirmal, who has been my constant source of support and encouragement, who shared my happiness and sorrow and has always been my morale boost.

I also wish to thank all my friends and family, my support system, without whom I wouldn't be the person I am today.

Nancy, 14 July 2018

Madhubala Ganesan

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
BFD	Best Fit Decreasing
CDC	Cloud Data Center(s)
CO_2	Carbon dioxide
DPM	Dynamic Power Management
DVFS	Dynamic Voltage and Frequency Scaling
EDP	Elastic Data Processing
GDP	Gross Domestic Product
GHG	Green House Gas(es)
IaaS	Infrastructure as a Service
ICT	Information and Communication Technologies
IoT	Internet of Things
IPC	Idle Power Consumption
KVM	Kernel-based Virtual Machine
LOC	Loss of Control
LRR	Local Regression Robust
MAD	Median Absolute Deviation
MMT	Minimum Migration Time
NIC	Network Interface Card
PUC	Power Usage Characteristics
QoS	Quality of Service
RC	Random Choice
SA	Situational Awareness
SLA	Service Level Agreement
SPM	Static Power Management
THR	Threshold based Heuristic
VM	Virtual Machine
VMM	Virtual Machine Monitor
WoL	Wake-on-LAN

1 INTRODUCTION

Internet of Things (IoT) is the outcome of the emanating third wave of development of Internet. IoT, along with Big Data Analytics is poised to become the backbone of Smart and Sustainable Systems which bolster economic, social and environmental sustainability. Gartner (2014) predicts that IoT will hit mainstream by 2020 with almost 25 billion smart objects, which is more than thrice the number of human beings on earth. The degree of connectivity will extend to all everyday use products and smart phones will become even smarter. The boundless scope for IoT in various domains makes this period, the IoT era.

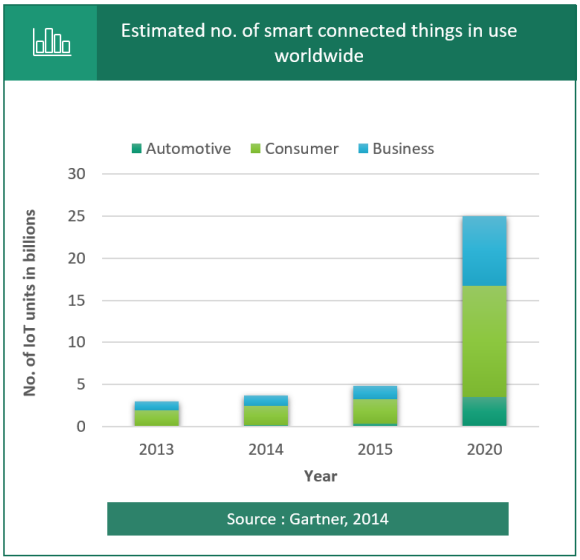


Figure 1. Estimated no. of smart connected things in use worldwide (Gartner, 2014)

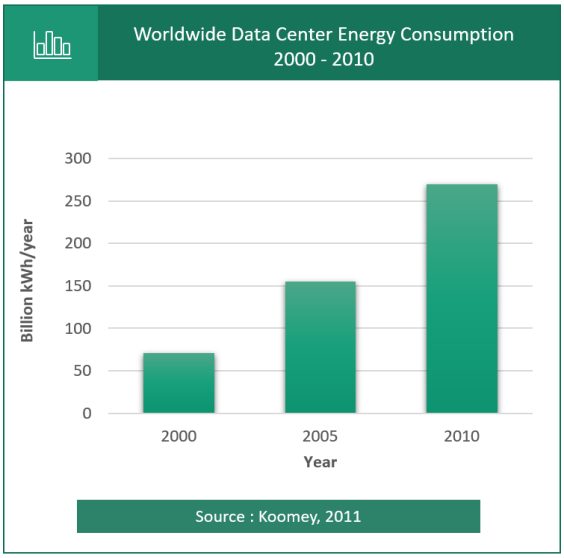


Figure 2. Worldwide Data Center Energy Consumption (2000-2010) [1]

Data is considered to be the next 'Oil' because of the value it brings in various sectors. Data Analytics help businesses increase revenues by boosting operational efficiency, target marketing, predicting progression in market etc., In addition, Data Analytics plays a vital role in getting insight out of voluminous data generated by sensors and devices [2]. Big Data refers to immense amount of raw and unstructured data that needs to be gathered and analyzed. This data is generated by diverse sources for various purposes. The purpose served by the data generated may not be the purpose intended when the data was generated. Traditional data-processing systems are replaced by sophisticated big data processing systems and platforms as the former is incapable of handling voluminous and complex data generated by IoT [3].

To meet the exponential growth of IoT data and user demands, the cloud computing paradigm provides vast computational resources for fast and reliable data processing. IaaS is a cloud service model where CDCs deliver computing, networking and storage resources on-demand over the internet. Hundreds of thousands of servers are provisioned worldwide. IaaS providers such as AWS EC2, Rackspace, Google Compute Engine provide virtual machines of different types based on the amount of resources [4].

Data center facilities that house networked computers and equipment play a crucial role in providing elastic computing resources creating an illusion of infinite resources [5]. Consequently, there is a need for a large amount of energy to power the Information and Communication Technologies (ICT) equipment. Increasing energy needs calls for the need for more coal and fossil fuels to generate electrical energy. Statistically, the energy consumption of data centers all over the world has increased by 56% within a short span of five years between 2005 and 2010 [6].

According to Gartner (2007), Data Centers account for 23% of global carbon dioxide (CO_2) emissions from ICT industry. The report also points out that data centers contribute to 2% of the global CO_2 emissions which is on par with that of the aviation industry. In spite of the overall value that ICT brings, Gartner indicates that this trend is unsustainable. There is a need for the innovations in ICT services and products to reduce environmental impact. Gartner provides some recommendations to the IT organizations to address this problem: measuring power consumption, consuming fewer servers by optimizing utilization, using devices on low power state, analyzing energy usage of the Data Center periodically. The European energy policies and climate targets for the years 2020 and 2030 have energy efficiency as a key part [7]. Energy consumption not only increases the operational cost of the data center but also acts as a major contributor of global carbon emissions.

1.1 Motivation

Eye tracking technology is getting significant attention to understand visual perception and the relationship between vision and attention. Research in this area has become an entrenched method in evaluating eye-movement behavior of participants. Eye trackers are being used in real-world systems and in fields such as psychology, neuroscience, aviation and human computer interaction [8].

In aviation industry, based on the data gathered from flight incidents, airline regulators are finding ways to improve safety across the industry. Data4Safety, a program launched by The European Aviation Safety Agency (EASA) for data collection and analysis for risk detection using data from multiple sources such as safety reports, in-flight telemetry, sensors and devices data, air traffic surveillance information, and weather data [9]. Analysis of data generated by Internet of Things (IoT) devices such as 'wearable' eye tracker device is gaining significant attention in detecting human errors of pilots while flying [9]. Factors such as visual distractions and fatigue can lead to loss of control and accidents [10]. The aim of this program is to identify potential safety risks and determining the right actions to mitigate the risks. The weak links in the aviation chain are identified by chewing over terabytes of data [10].

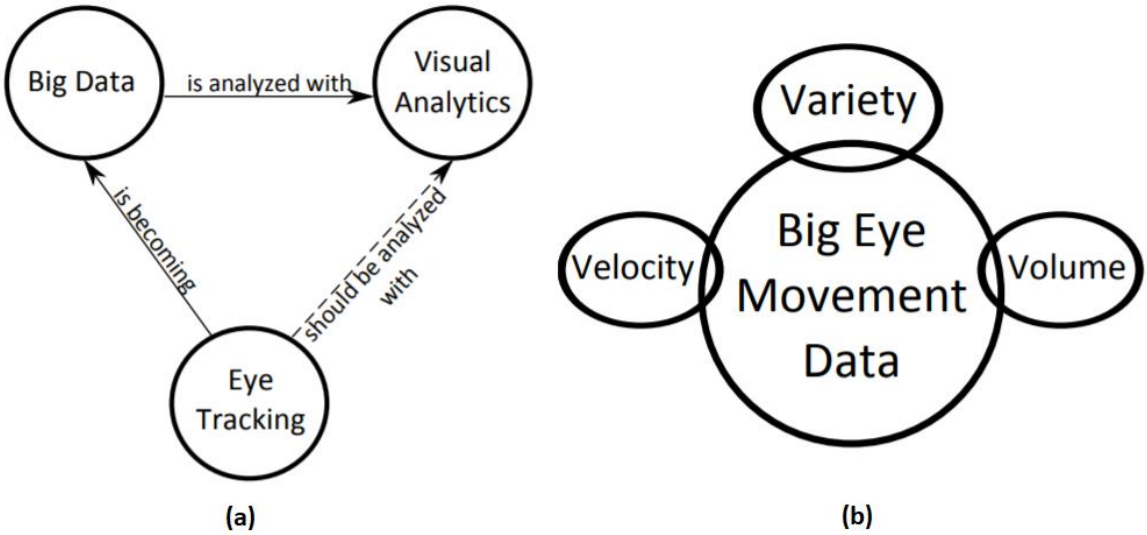


Figure 3. (a) Relationship between eye-tracking and big data analytics, (b) Three V's of eye tracking data [8]

In general, eye-tracking glasses, interactive eye tracking devices or smartphones are being used. Most eye-tracking devices are considered as 'wearable devices' which continuously sense eye-movements [11]. Considering the three characteristics or three V's of big data, the data generated are in extensive amounts of varied data types which makes the eye-tracking data labelled as 'big data' (Figure 3). The volume of eye-tracking data collected is immense and technologies such as head-mounted eye tracker devices are enabled with high tracking speeds (such as 500 data points per second) [8]. The velocity of data generation has increased drastically as recording devices have become cheaper and easily available. In addition, eye tracking is often integrated with different data sources such as electroencephalography (EEG), motion trackers, functional magnetic resonance imaging (fMRI), mouse and keyboard interactions. This has led to a large variety in the data formats [8]. Sophisticated approaches to identify patterns in the data beyond the statistical dimension has become essential. Movement measures, fixation counts, numerosity measures, distance measures are a few possible directions in analyzing eye-movement data [11].

The concept of ‘black box in the cloud' is considered to be a boon to the aviation industry. In this service, data is streamed from the aircraft to the airline in real-time. Position reporting, pilot's attention while flying using sensors can be reported in real-time to prevent aircraft accidents and disappearance far less likely [10]. The cloud infrastructure of the airline will do the required processing as the computational and storage resources needed cannot be compactly fit into the aircraft [10]. On the other hand, the data center infrastructure used is a major contributor of enormous power usage [12]. As aviation and ICT industries are the two major energy consumers, the situation calls for optimization of energy consumption of data centers.

Figure 4. Goal of Data Centers

The computing industry has always focused on improving the system performance. With efficient system designs and increasing the number of components based on Moore's law [13], this objective is being achieved. Though the output per watt is improving, there is no reduction in the total power consumed by the computing system. On the contrary, it is increasing in an exponential rate [14]. This trend has led to a situation where the cost for energy consumption of the server has exceeded the actual hardware cost and this is particularly true for large-scale computing infrastructures such as cloud data centers which are impacted by energy problem [1] [15].

ICT infrastructures and cloud providers are looking for effective systems and methods to address the overwhelming utility bills and their carbon footprint [15]. Green Computing focuses on optimizing computing technologies and practices to reduce the negative impact on the environment without compromising on performance [16]. The goal of computing infrastructure industry has shifted to energy efficiency coupled with high level of QoS for customers as depicted in Figure 5. Energy optimizations within a data center revolves mainly around improving cooling systems and ICT equipment. These are the major energy consumers within a data center [14]. In case of an IoT system that requires a cloud-based Big Data processing platform, resources need to be elastic to meet the needs. Analysis of energy efficient cloud systems and approaches for such a dynamic IoT system can provide useful insights towards building an energy efficient infrastructure.

VM consolidation is one of those green practices where the number of active computing devices are reduced by transitioning inactive servers to ‘energy saving’ mode [17]. Infrastructure as a Service (IaaS) providers consider a number of metrics to define their performance to meet Service Level Agreements (SLA). This work involves a study on energy challenges in a cloud-based IoT data processing system and state-of-the-art energy efficient techniques. It also involves investigation of the power usage characteristics of compute and storage nodes and the impact of VM consolidation technique to migrate Virtual Machines and Block Storage (big data storage) by building a cloud infrastructure followed by conducting appropriate experiments. The research focuses on evaluating VM consolidation algorithms in terms of SLA metrics and energy consumption.

1.2 Problem Definition

While a number of different approaches have been proposed in previous research work in this area, a few gaps still persist. The following are the gaps identified:

1. While there have been studies on big data applications for IoT and challenges faced regarding resource allocation for big data platforms on cloud, energy challenges in cloud for processing big data workload is not considered [18][19]. As CPU traces from PlanetLab or Google Cloud DataStore (GCD) are often used as workload for simulations [20], there is a lack in studying the performance for processing IoT related big data workload on compute hosts.

2. While several studies have compared the energy efficiency and performance of algorithms for each of the four VM consolidation sub-problems individually [21][20], there is a lack of rigorous evaluation and comparison of the overall system when algorithms for all the four sub-problems implemented together.

3. Although several works have tested the potential of VM consolidation in simulated environments using tools such as CloudSim, there is inadequacy in terms of investigations on physical cloud infrastructure [22].

4. VM Consolidation Frameworks for cloud platforms that are proposed in the past research works are not evaluated enough for different algorithms and workloads [23].

This thesis attempts to bridge the above-mentioned gaps in research. Different VM consolidation techniques are evaluated for processing big data and in the context of our research, the big data comes from IoT 'wearable' - in the form of an eye tracker.

1.3 Research Questions and Objectives

This section presents the research questions and the corresponding objectives that should be achieved in this research.

RQ1. What are the energy efficient approaches in cloud infrastructures?

RO1. ***Investigation of state-of-the-art***

RO1.1 Investigation of current energy challenges

RO1.2 Investigation of energy efficient cloud systems, models and approaches

RQ2. How can VM consolidation bring about energy efficiency in a cloud infrastructure?

RO2. ***Investigation of Power usage characteristics of compute servers***

RO2.1 Building a cloud infrastructure

RO2.2 Conducting baseline experiments and measure energy consumption

RO2.2 Implementing VM consolidation and analyzing energy usage

RQ3. Which is the most effective VM consolidation strategy for processing a big data workload?

RO3. ***Evaluation and comparison of VM consolidation strategies in cloud***

RO3.1 Selecting algorithms for VM consolidation

RO3.2 Implementing VM Consolidation Workflow

RO3.3 Designing and conducting experiments

RO3.4 Identifying performance and energy metrics for evaluation

RQ4. What is the impact of the solution on Sustainability?

RO4. ***Conducting a Sustainability Analysis using frameworks and open data***

1.4 Delimitation

In order to test VM consolidation on real cloud infrastructure, a data center infrastructure is required. Considering the limited available resources, a small lab with traditional PCs as compute hosts is set up. The set-up mimics a small-scale multi-node cloud infrastructure. The experiments are designed based on the availability of resources. The system is not adaptive or energy aware in real time because it is based on asynchronous data. Some of the experiments are repeated over a period of 24 hours which required data to be processed continuously. The data from sensors and devices are collected and stored over time. The collected data is continuously streamed and processed during the 24-hour long experiments to mimic a real world IoT system.

Some early decisions on selection of tools and platforms for testing the VM consolidation is done based on literature study. Openstack is chosen as the cloud platform and Apache Spark as Big Data Platform. Underload, Overload detection algorithms, VM selection & VM placements algorithms are chosen from the literature based on their performance on simulated workloads.

1.5 Contributions

The following are the key contributions of this work:

1. Investigation of VM consolidation on physical cloud infrastructure and a VM Consolidation Workflow is designed and implemented with open source technologies, tools and platforms.

2. Comparing different algorithms for VM consolidation sub-problems and evaluating their performance and energy efficiency. Providing recommendations on the most effective combination of algorithms for processing a IoT related big data workload using an Aviation system use case.

3. Analysis of the impact of the solution on Sustainability using open data and Sustainability analysis framework.

1.6 Organization of the Thesis

This structure of this manuscript is as follows:

- ***Chapter 1*** : ***Introduction*** provides an understanding of the background, motivation, objectives and contribution of the work presented in this thesis.
- ***Chapter 2*** : ***Review of Related Work*** covers a a broad study of IoT and Big data, state-of-the-art energy-efficient systems, methods and approaches, Resource Management in Cloud, VM consolidation approaches and algorithms.
- ***Chapter 3*** : ***Methodology*** dives into details about the systematic approach of the work with images that are relevant for the experimental set up and describes the architecture in detail, design and conduct of the experiments.
- ***Chapter 4*** : ***Results and Discussion*** present the results of the experiments conducted with relevant performance and energy metrics and the inferences.
- ***Chapter 5*** : ***Conclusion and Future work*** presents the outcome of the thesis and possible direction of future work.

2 REVIEW OF RELATED WORK

This chapter is divided into five sub-sections to discuss the following topics: (a) IoT and Big Data, (b) Energy-efficient computing systems, (c) Cloud-based data centers, (d) Cloud resource management, and (e) Consolidation of Virtual Machines. Several work in these areas are reviewed to highlight the state-of-the-art and identify relevant research gaps.

2.1 IoT and Big Data

Internet of Things will be positioned as an added value for applications in the future. It will involve vast deployment of sensors and devices which incurs a huge sum of money [24]. Effects of IoT (being a pervasive technology) on the environment must be taken into account [25]. For long term use, the whole system has to be optimized for energy efficiency and resource utilization. Radio-Frequency IDentification (RFID), Machine to Machine (M2M) communications, green cloud computing and data centers are the key focus areas of green IoT [16]. [26] discusses various green initiatives at various levels of a cloud based IoT system. [25] focuses on narrowband IoT (NB-IoT) which is a Low Power Wide Area Network (LPWAN) radio technology. NB-IoT has a great impact on improving battery life of devices and also on the cost. LPWAN is suitable for transmitting small amount of data over long periods of time which is highly suitable for IoT devices. Various issues and technology solutions related to green IoT for reducing power consumption are discussed in [16]. It also provides future directions for open problems in Green IoT, one of which being optimal utilization of cloud data center resources for energy efficiency. Five principles are proposed to be adopted for a greener IoT world by 2020. The principles explained by means of a case study on smart phone as an IoT device [26].

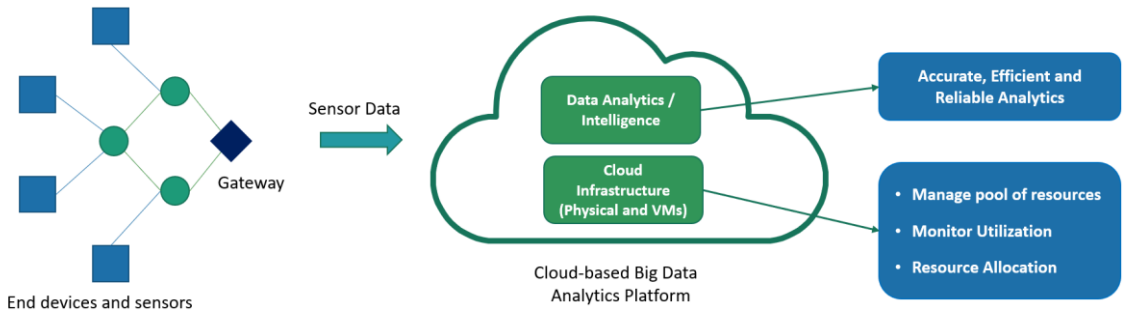


Figure 5. Cloud-based IoT System

The two main components in the cloud for a cloud-based IoT system are the data analytics application or software and the cloud infrastructure (physical and virtual) [27]. Such a system is represented in the Figure 5.

Figure 6. Data Processing Goals, Platforms and Constraints

The goals of data processing are to facilitate faster decisions, provide reliable results with low latency for batch and stream processing and sophisticated methods for making 'better' and more informed decisions [5]. In order to achieve these goals, powerful platforms are imminent. Hardware and software play crucial role in attaining these goals. IoT data is voluminous and complex for which scalable systems are necessary [28]. CPU time, I/O time, storage resources and energy efficiency are examples of constraints for efficient data processing. So, energy efficiency in data centers for IoT data processing is important as servers used for computations are energy hungry [26].

2.2 Energy-efficient Computing Systems

In this section, the state-of-the-art systems and techniques used for energy efficient computing are studied. Gordon E. Moore, in the year 1965, stated, "With unit cost falling as the number of components per circuit rises, many as 65,000 components on a single silicon chip" [13]. With increase in the number of components, size has decreased, while speed has increased. This is applicable to the increase in number of cores in a CPU [29]. Increased speed is invaluable for mission critical tasks but the need for more energy adversely affects the system. A simple circuit theory can be used in calculating the power consumption of CPU [29]. In this case, CPU can be considered as a variable resistor that changes resistance with increase in workload. Power dissipation can be depicted in Eq. (1),

$$P_{CPU} = V_{supply} * I \qquad (1)$$

The relationship between CPU utilization and total power consumption of a server was modelled in [30]. It states that with growth in CPU utilization, power consumption grows in a linear fashion from the idle state power consumption up to when server is fully utilized. This relationship is expressed in the Eq. (2),

$$P\ (u) = P_{idle} + \left(P_{busy} - P_{idle}\right) \times u \qquad (2)$$

Where P is estimated power consumption, P_{idle} is the idle server power consumption, P_{busy} is the power consumption when server is fully utilized and u is the current CPU utilization. The problem of energy wastage is addressed by an energy proportional computing system, where the energy consumed by the computing systems is proportional to the workload [31]. For single-core processor, the relationship is approximated by a quadratic function and on the other hand, the relationship is approximated by a linear function for dual core processor [32]. In case of embedded systems, power consumption of the device is a critical design consideration.

In DVFS, depending on the demand for resources, voltage and frequency of the CPU are dynamically altered which has resulted in saving 30% of power in low-activity states of the desktops and servers [31]. On the other hand, dynamic power ranges of other components are less to negligible. It is less than 50% for Dynamic Random Access memory (DRAM) and 15% for network switches [30]. The fact that CPU is the only component that supports low-power modes is the actual reason behind the variation of dynamic power ranges. Other components do not support low power modes and can only be switched off. Nonetheless, there is a considerable effect on performance caused by the transition between active and inactive states. Idle state power incapability of the server components has led to a limited dynamic power range of the server to 30% [30]. A study on benchmarking power usage characteristics of embedded processor and analyzing the idle power consumption shows that the energy consumed by the processor significantly diminishes the instant the processor enters an idle state [33]. This study also states that during the idle state, energy consumption reduces without invoking hardware-based frequency scaling or Dynamic Voltage and Frequency Scaling (DVFS) methods thus, being effective with less overhead [34].

2.3 Cloud-based Data Centers

National Institute of Standards and Technology (NIST) defines cloud as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [35].

Cloud providers are looking for efficient ICT infrastructures to address their overwhelming utility bills and carbon footprint [15]. The goal of computing infrastructure industry has shifted to energy efficiency coupled with high Quality of Service (QoS) level for customers. End-users have an impact in terms of increased resource usage costs which is decided based on the total cost of ownership (TCO) by the provider [36]. Higher energy consumption inadvertently increases utility bills and impacts the requirement for more cooling systems, Uninterruptible Power Supplies (UPS) and Power Distribution Units (PDU). Several studies have shown that reducing power consumption of a system effectively extends the overall running time which results in longer lifetime of the device [33].

Most cloud data centers use blade servers which provide more computational power and less space consumption. However, blade servers are hard to cool as the components inside each rack are densely packed. In line with an example stated in [14], 60 blade servers can be mounted to a rack of 42U where ‘U’ is called the rack unit, a measure of height of a server [22]. On the contrary, the rack requires up to 4000 W for power supply to the servers and cooling systems compared to a rack with 1U servers which require only 2500 W. The sustainability of data centers and their efficiency measures are listed down in [37]. Power infrastructure, Cooling, Airflow Management and IT efficiency are the key factors that determine the efficiency of a data center. Power Usage Effective (PUE) and Data Center Infrastructure Efficiency (DCIE) are the widely used energy efficiency metrics which were originally proposed by Greed Grid Consortium [38]. PUE is a ratio of energy consumed by the data center to the energy supplied to the computing equipment. DCIE is the inverse of PUE [37].

2.4 Cloud Resource Management

Several studies state that the objectives of cloud resource management are to provide highly elasticity and scalability, provide efficient, reliable resource utilization [39][40][41]. It is a set of processes to effectively and efficiently manage resources and also guarantee Quality of Service (QoS) for the consumers [42]. [43] describes resource management in two phases: Ab-initio Resource Assignment followed by Periodic Resource Optimization. Ab-initio resource assignment is the first phase where the user requested resources or application requirements are fulfilled by following a series of steps as follows:

1. Request Identification
2. Resource Gathering
3. Resource Brokering
4. Resource Discovery
5. Resource Selection
6. Resource Mapping
7. Resource Allocation

The second phase of cloud resource management is periodic resource optimization which is classified into two types for non-virtualized resources and for virtualized resources [43]. The latter differ from the former in the ability to be gathered and dispersed to meet the requirement posed. This process requires periodic optimization either by bundling or by fragmenting resources [44]. Periodic resource optimization involves continuous resource monitoring and VM consolidation [21]. Resource allocation / reallocation can be achieved by applying policies which are broadly classified into four types based on the expected outcome: Load Balancing, Server Consolidation, SLA / QoS based allocation and Hybrid allocation of resources (Illustrated in Figure 7). The following figure represents the four policies for resource allocation and optimization [43]. This work focuses on guaranteeing QoS by meeting SLA and energy efficiency.

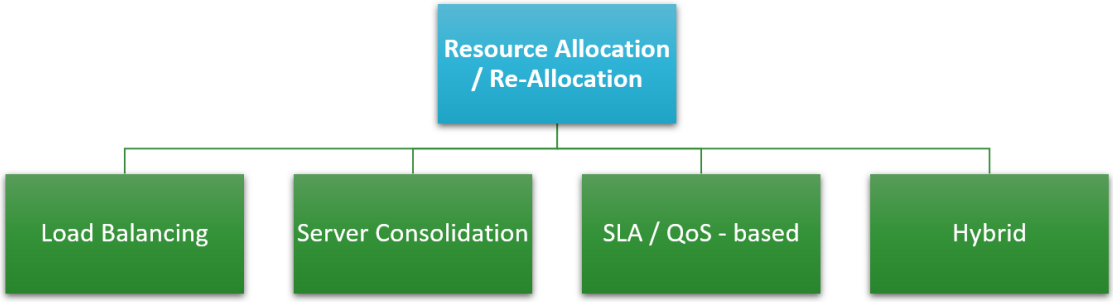


Figure 7. Factors involved in Resource Allocation / Re-allocation [43]

Several researchers have classified cloud resources into two types: physical and logical or hardware and software resources [42][44]. Another approach to classify the cloud resources based on its utility is proposed in [43]. The author classifies the resources into five types: Fast Computation Utility, Storage Utility, Communication Utility, Power / Energy Utility and Security Utility. This work focuses on fast computing utilities: Processor and Memory that provide the computing power for data processing.

There exists a fundamental belief that only hardware efficiency has an effect on energy consumption of devices. Several studies have proven that resource management systems also play a role in energy consumption of devices. The relationship between different layers in the computing systems and energy consumption are discussed in [14]. Figure 8 shows the interconnection between the layer as explained in [14].

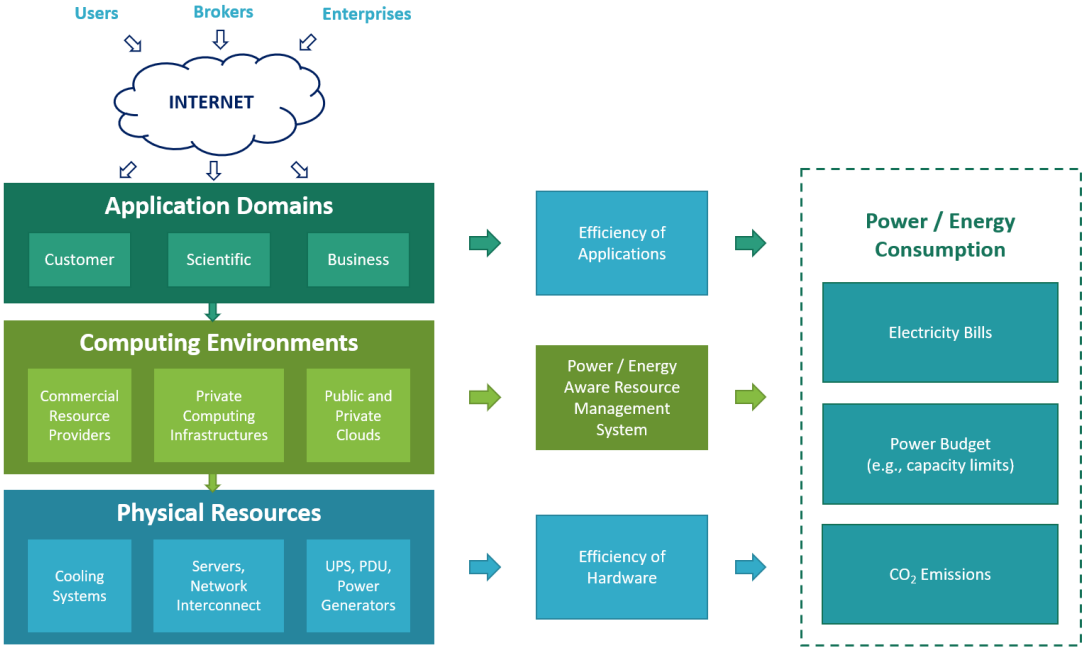


Figure 8. Interconnection between layers in computing systems [14]

The data centers' computing infrastructure consists of three main divisions: application domains, computing environments and physical resources. Efficiency of applications such as the concept of multi-threading to utilize cores of the processor for computation has an effect on power consumption. Hardware efficiency of the devices and cooling systems also effect power consumption. The middle layer, that provides the computing environment is often the least targeted layer for energy optimization. But in the recent studies, the concept of visualization is useful in altering the energy behavior of systems. This includes commercial providers, private computing resources and cloud.

2.5 Consolidation of Virtual Machines

Dynamic Power Management (DPM) techniques help reducing energy consumption temporarily whereas Static Power Management (SPM) is for permanent reduction in energy consumption [14]. VM consolidation is a potential DPM solution for improving resource utilization and reducing energy consumption [21][45]. The classification of Power Management Techniques in a data center is represented in Figure 9.

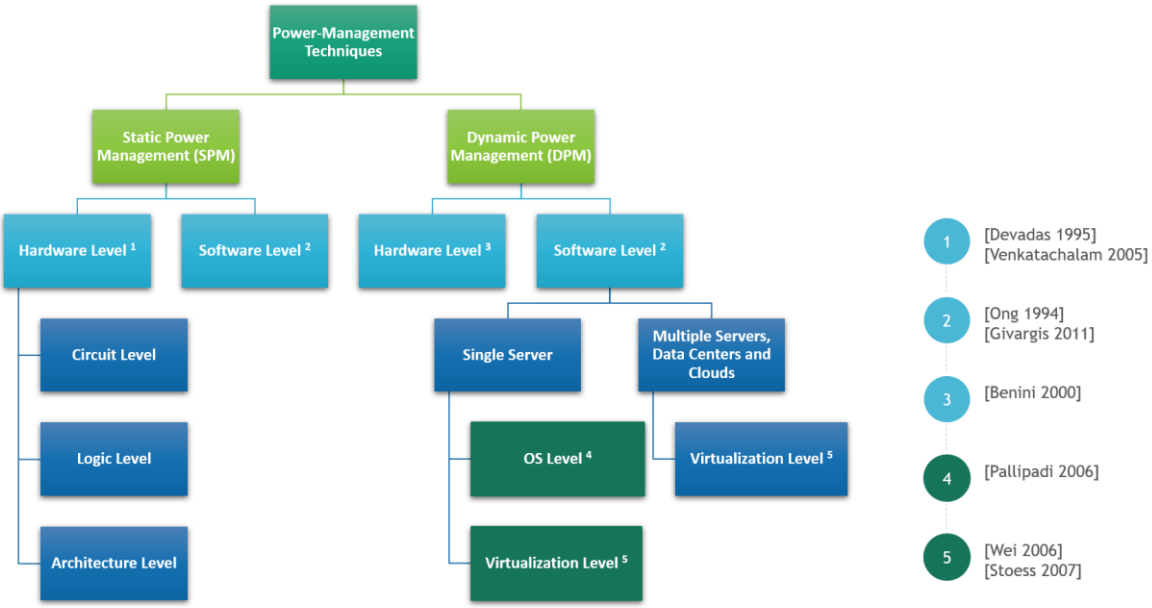


Figure 9. Classification of Power Management Techniques [40]

Virtualization facilitates provision of multiple virtual machines on a single physical host [46]. As a result, resources are better utilized thus, increasing the Return On Investment (ROI) [22]. VM consolidation technique achieves energy saving by eliminating idle power consumption through switching of idle hosts to low power mode such as sleep or hibernate

modes [47]. One of the capabilities of Virtualization is the ability to relocate a VM between compute nodes, called migration [48]. Performing VM migration with no downtime is called live migration [48][49]. There are two main states when VMs are migrated: when some physical hosts are under-utilized, VMs are migrated to keep the physical servers to be minimum in number; and relocate VMs from overloaded hosts to prevent deterioration of performance [20].

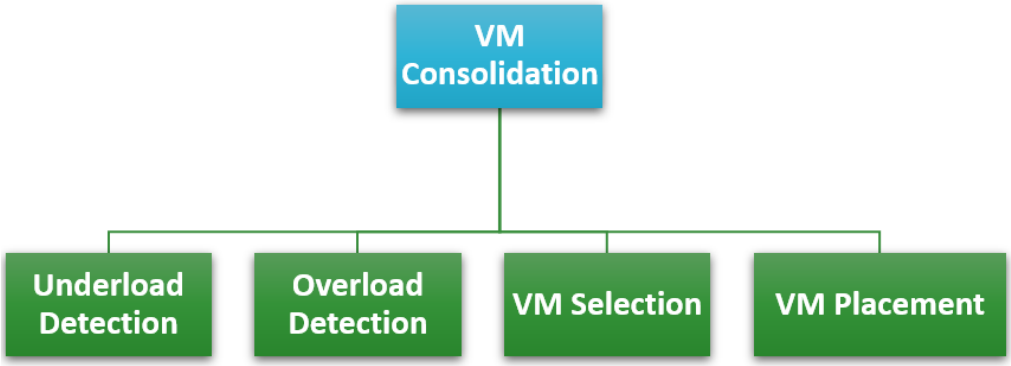


Figure 10. VM Consolidation Sub-problems [46]

Dynamic VM consolidation is a complex real-time decision-making problem that involves four sub-problems as shown in Figure 10 : underload detection, overload detection, VM selection and VM placement [50][38]. A tiered software system for VM consolidation was proposed in [46]. In this system, Virtual Machine Monitors (VMM) continuously observe the resource utilization and thermal state of VMs in each physical host. Local managers placed in the VMM observe the resource utilization of VMs and send the information to the Global Manager. . Commands to apply DVFS and turning hosts to idle nodes are issued by the Global Manager that manages a set of nodes. Figure 11 is the system architecture proposed in [46].

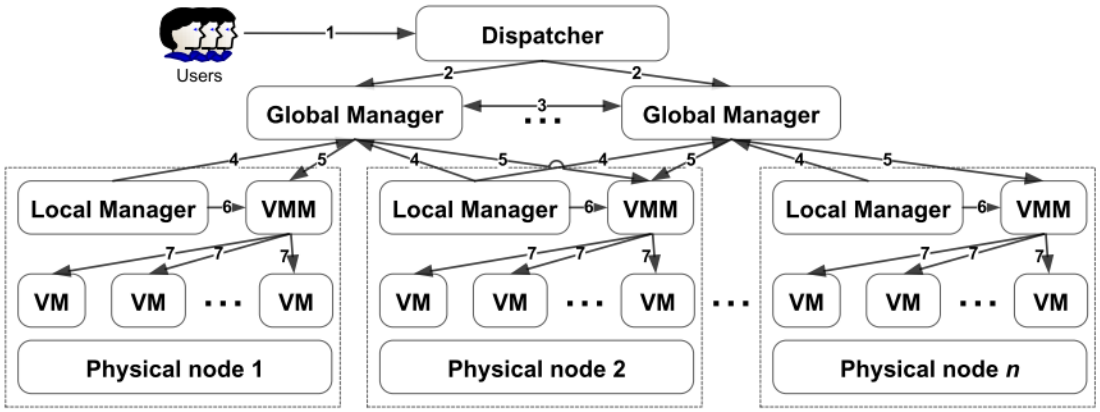


Figure 11. Tiered model for VM consolidation [46]

Several algorithms have been proposed and the performance of the algorithms are tested for individual sub-problems in [50][51] but the performance of the whole system when different algorithms are combined together and applied is not adequately studied. This work focuses on analyzing the effect of VM consolidation when different combination of algorithms are applied for an IoT big data workload.

A static threshold based underload detection is commonly applied as CPU utilization does not drop below a threshold very often and complex algorithms cause unnecessary overhead [52]. On the other hand, sudden peaks are noticeable with varying workloads that overloads hosts and cause performance degradation. Three category of algorithms are proposed for overload detection in [50][52]: Static-threshold based [50], Adaptive utilization based [50] and Regression based algorithms [50]. We select one algorithm from each category to analyze the suitability for big data workload. Threshold based Heuristic (THR) [22], Median Absolute Deviation (MAD) [50] and Local Regression Robust (LRR) [50] are the algorithms chosen for overload detection. When an overloaded or underloaded host is detected, selecting the right VM is important to cause minimal performance degradation. Random Choice (RC) and Minimum Migration Time (MMT) are the two common algorithms proposed for VM selection [38][53].

Combo	**Underload Detection**	**Overload Detection**	**VM Selection**	**VM Placement**
Combo1	THR[1]	THR	RC[2]	BFD[3]
Combo2	THR	MAD[4]	RC	BFD
Combo3	THR	LRR[5]	RC	BFD
Combo4	THR	THR	MMT[6]	BFD
Combo5	THR	MAD	MMT	BFD
Combo6	THR	LRR	MMT	BFD

Table 1. Combinations of Algorithms

[1] Threshold based Heuristic; [2] Random Choice; [3] Best Fit Decreasing;
[4] Median Absolute Deviation; [5] Local Regression Robust; [6] Minimum Migration Time

VM placement is regarded as a bin packing problem with varying bin sizes. Compute nodes are referred to as bins and VMs as items. The available CPU capacities are referred to as bin sizes. Best Fit Decreasing (BFD) algorithm sorts VMs in the decreasing order of CPU utilizations and VMs are placed on the host which will experience the least increase in power consumption. We chose BFD as it performs better than First Fit Decreasing (FFD) for any workload [38]. Based on the selected algorithms, we form six different combinations to be tested. Table 1 shows six combinations proposed for this research.

To summarize, IoT data is complex and voluminous that requires scalable, efficient systems for data processing. Both hardware and software play a key role in analyzing the 'big' IoT data generated. Cloud based data centers provide the necessary compute, network and storage resources for this purpose. Energy efficiency in data centers for IoT data processing is important as servers used for computations are energy hungry. We understand that cloud resource management not only provides high elasticity, scalability and reliable resource utilization but also makes the resources to be utilized in an energy efficient manner. VM consolidation is one such energy-saving technique. In the past researches, VM consolidation is tested on simulated cloud environment using simulation tools such as CloudSim [22][40] on simulated workloads with CPU traces from PlanetLab or Google Cloud Datastore (GCD) [50][38][49]. However, the performance is not tested on cloud infrastructure for complex data processing. This research aims to evaluate the combination of algorithms on Openstack Cloud as it is a potential cloud platform for big data processing [54][55]. The open-source platform provides energy efficiency capabilities using APIs of Nova compute service. Openstack NEAT is dynamic VM consolidation framework developed as an add-on package for Openstack instances. The framework is proposed in [56] but is not evaluated for big data workloads.

3 METHODOLOGY

Methodology discusses the systematic approach of the work, the experimental set up, architectural design and conduct of the experiments in detail.

3.1 Research Methodology

Research is defined as answering unanswered questions or exploring something that currently does not exist [57]. Research methodology is a systematized effort to gain new knowledge [58]. This thesis is approached systematically in five phases as illustrated in the figure below:

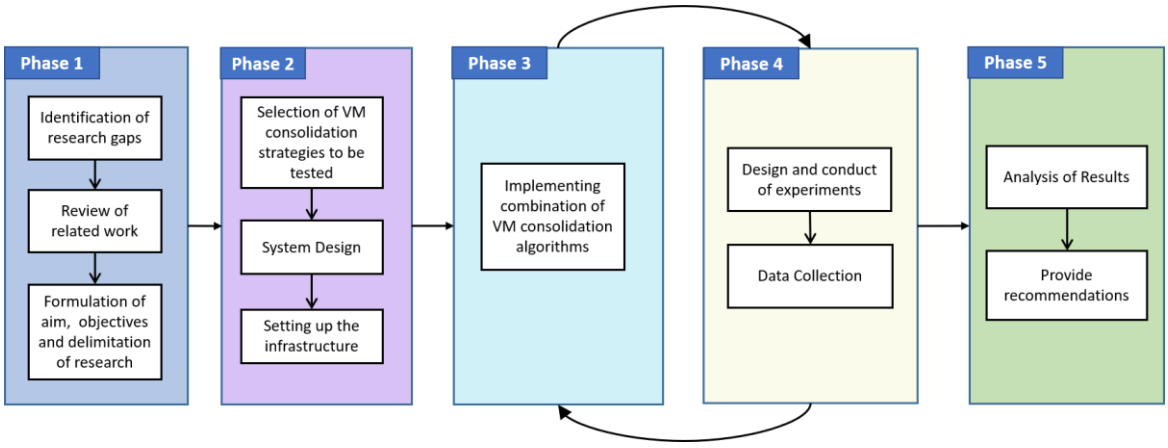


Figure 12. Research Methodology

Phase 1: Defining the problem

In the first phase, the challenges and research gaps in the area of energy-efficient cloud infrastructure is studied. Several works on this domain are critically reviewed for understanding the state-of-the-art systems and approaches. Based on the literature study and identified gaps, a set of research questions, goals and limitations of the research are identified and formulated. A refined list of objectives to be achieved are defined.

Phase 2: Designing the system

During the second phase, different VM consolidation strategies are studied and selected to be evaluated. An architecture of the system is designed with details of each component and its functions. The architecture designed is set as the base for setting up the physical infrastructure.

Phase 3: Implementation

During this phase, a set of algorithms selected during the previous phase are implemented based on the VM consolidation workflow, on the cloud set up.

Phase 4: Conducting Experiments

In order to evaluate the VM consolidation strategies, a set of experiments are designed and conducted. Each experiment has specific parameters and outcomes. Each experiment is conducted (run) multiple times to obtain reliable results. For some experiments that assess cumulative results, the task is run over a specific period of time. Data is collected during the whole period of experiments. After testing a combination of algorithms, the previous step is repeated by implementing the next combination of algorithms. This phase is repeated for all six VM consolidation strategies (six combination of algorithms).

Phase 5: Providing Recommendations

In the final phase, the data obtained by running experiments are analyzed and mapped to SLA and energy metrics. Based on the results obtained, the effective combination of algorithms for VM consolidation is recommended for the IoT big data workload.

3.2 Cloud System Architecture

The overall architecture of the cloud based IoT data processing system used for conducting experiments in this research is depicted in Fig. 13. The system is built using open-source tools and platforms such as Openstack, Apache Spark and Openstack NEAT [56]. The Aviation System discussed in Section 1.1, consisting of the flight simulator and wearable eye-tracker device is the IoT Set up. For simplicity, REST API is used for integration between the IoT system and the cloud platform. Other messaging protocols such as Message Queuing Telemetry Transport (MQTT) or COnstrained Application Protocol (CoAP) can also be used. The components of the system are described in the following sections.

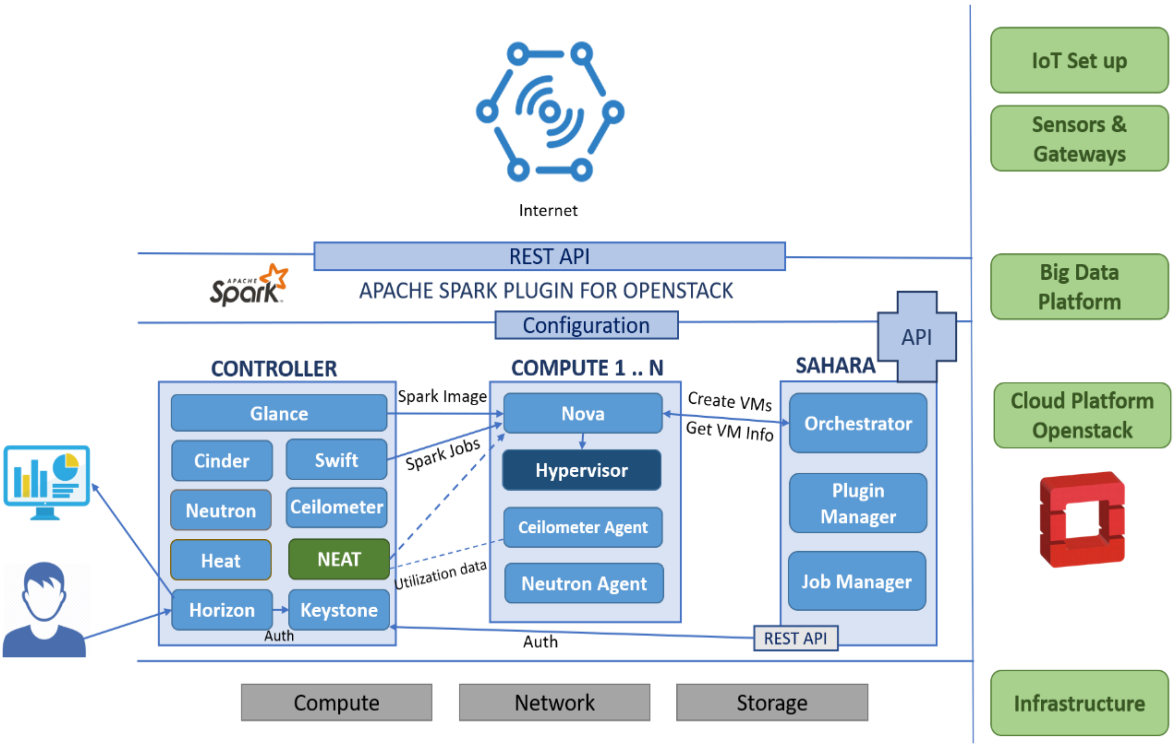


Figure 13. System Architecture

3.2.1 Cloud Platform - Openstack

The bottom-most tier is the physical infrastructure consisting of compute, network and storage resources. Openstack cloud platform is deployed on the infrastructure to virtualize the resources. Openstack is an open-source platform for creating and managing cloud infrastructure which is commonly used by IaaS providers [54]. Openstack project was begun with an aim to build a "massively scalable cloud operating system" [59].

It is built on the concept of distributed system with asynchronous messaging. It consists of seven major services for compute, storage, network, monitoring, orchestration and image along with authentication and dashboard services. Fig. 14 depicts Openstack as a Cloud Operating System (COS).

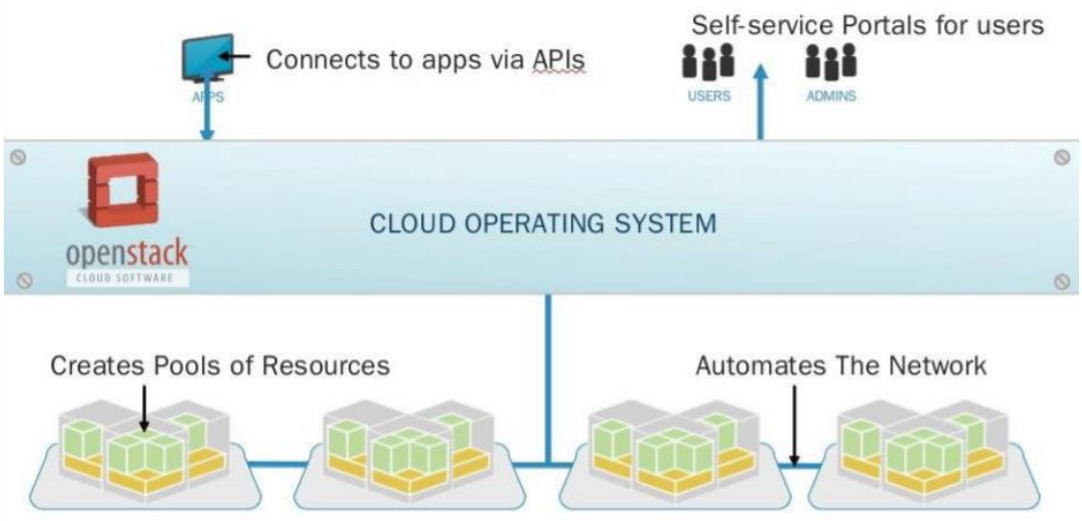


Figure 14. COS Openstack [60]

The services of openstack are loosely coupled inorder to be fully distibuted. The following are the core services of Openstack.

Compute

Nova is responsible for compute provisioning and management. It acts as the management layer for a list of supported hypervisors. The RESTful API exposed by the services are used for automating tasks and management.

Core Nova service comprises of:

- Compute Nodes – hypervisors that run virtual machines
    - Supports multiple hypervisors KVM, Xen, LXC, Hyper-V and ESX
- Distributed controllers that handle scheduling, API calls, etc
    - Native OpenStack API and Amazon EC2 compatible API

Compute controller is responsible for managing VMs on compute hosts. For the purpose of modelling a system, we created a four-node cloud set up with one controller and three compute nodes.

Object Storage

Swift is the Object Storage service in Openstack. It supports storing and retrieving data in the cloud. Swift service provides a native API as well as an AWS S3 compatible API. Swift differs from traditional file system storage and is the best type of storage when used for static data (media files such as MP3s, images, videos, VM images, and backup). A high degree of resiliency through data replication is provided. It handle petabytes of data.

Block Storage

Cinder is the Block Storage service in Openstack which provides persistent block storage for compute instances. The service manages the life-cycle of block devices : right from creation of volumes, attachment to instances to the release.

Networking

Neutron, the Openstack networking service provides services such as DNS, DHCP, IP address management, security groups which has network access rules such as firewall policies and network load balancing. Pluggable integration for various SDN solutions are available. The guest network configurations can be altered and managed by the cloud tenants.

Dashboard

Horizon, the dashboard provides a web-based interface of the Openstack platform. Through this dashboard cloud administrators and cloud tenants can provision, manage and monitor resources that are available.

Identity service

Keystone, the identity service is a shared service that provides authentication and authorization services for Openstack. There is pluggable support for multiple forms of authentication.

Image service

Glance, the image service provides disk image management services. It provides image discovery, registration, and delivery services to the Nova, as needed. QCOW, VMDK, VHD, ISO, OVF & AMI/AKI are the image file formants supported. Backend storage could be Filesystem, Swift, Amazon S3

Others Services:

The other services / modules include the following:

- Orchestration (Heat)
- Telemetry (Ceilometer)
- Database (Trove)
- Elastic Data Processing (Sahara)
- Bare metal (Ironic)
- Messaging (Zaqar)
- Shared file system (Manila)
- DNS (Designate)
- Key manager (Barbican)

In the experimental setup, Openstack version 'Newton' was used. Fig. 15 illustrates the list of services and openstack components that are enabled in the controller and compute nodes. KVM is used as the hypervisor on each compute node.

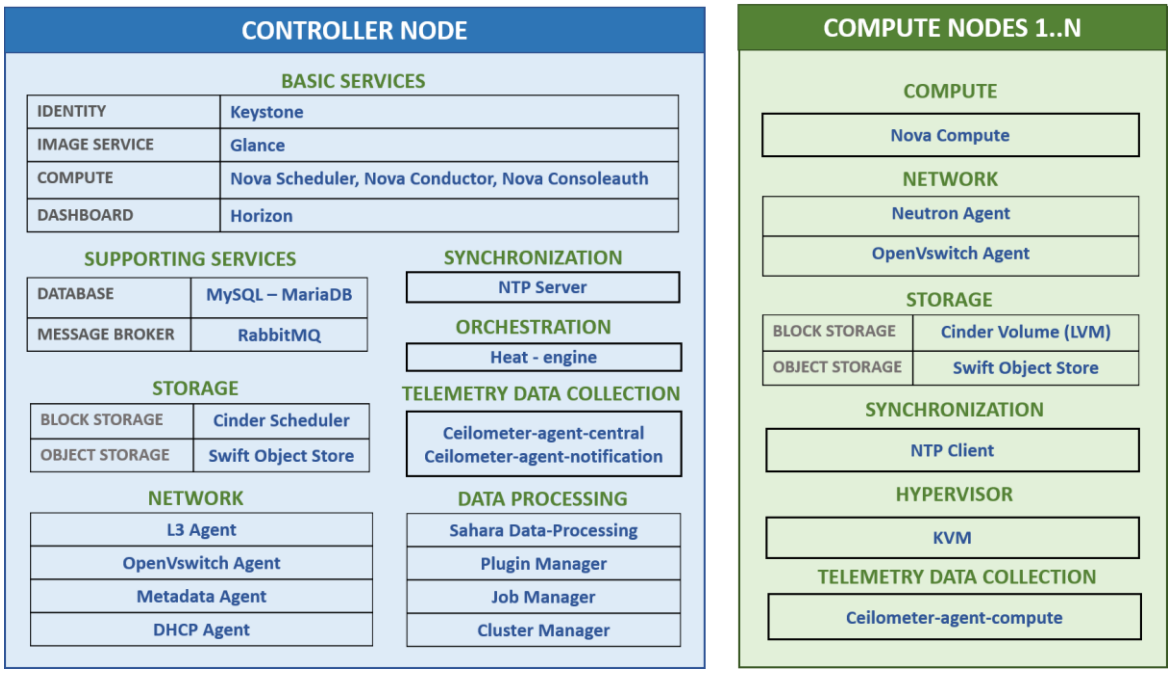


Figure 15. Openstack Components

3.2.2 IoT System

The experimental set up of a doctoral research work (at Leeds Beckett University) on Gaze pattern recognition to interpret vision cognitive behavior of pilots during in-flight startle is used as the IoT system. Fig. 16 illustrates the set-up which consists of a Flight Simulator, Flight controls and an Eye tracker device.

Figure 16. IoT Set-up

The relationship between startle and loss of situational awareness (SA) as a causal factor of Loss of Control (LOC) which leads to aviation accidents and fatalities can be understood by studying the pilot's eye fixations. The potential relationships that may exist within the problem space is examined by combining machine learning and statistical modelling of eye tracking data. Flight simulator and eye tracker generate performance, gaze fixations and pupil position data during 15 flying tasks with different startle scenarios. The data from this IoT system is diverse, voluminous, and demands a reliable big data processing platform to perform statistical analysis and classify the pilots based on performance. More details about the IoT set-up and tasks are explained in appendix 1.

3.2.3 Big Data Processing Platform

The data obtained from the IoT system is processed as Spark jobs. Apache Spark, the in-memory data processing engine which is suitable for both batch and stream processing, is used as the big data platform [61]. ‘Sahara’ is the renamed Openstack project ‘Savanna’ which provides a means for big data application clustering on Openstack. The plugins that are available for creating data-intensive application cluster are Hadoop, Spark and Storm.

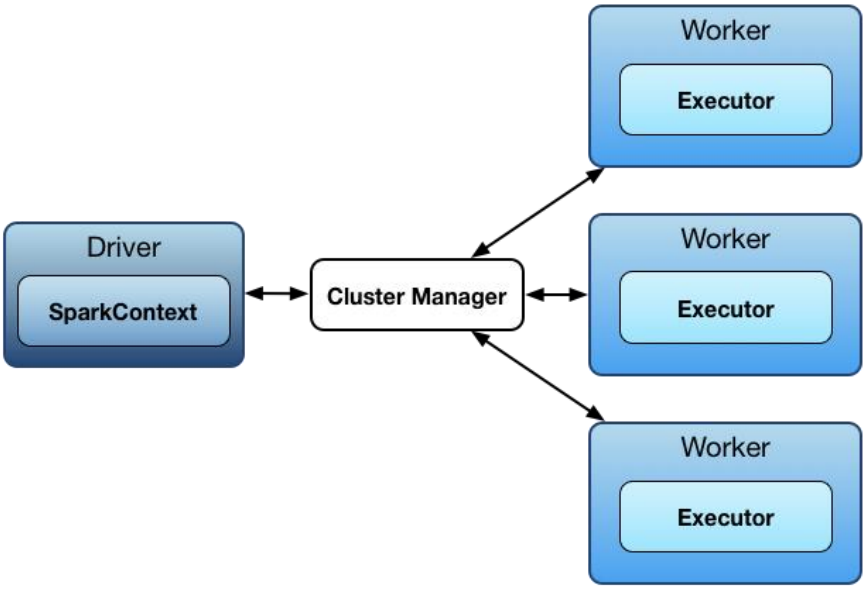


Figure 17. Apache Spark Architecture [62]

Spark uses a master/worker architecture. There is a driver that talks to a single coordinator called master or the cluster manager that manages workers in which executors run. The driver and the executors run in their own processes [62]. In this use case, the data processing jobs run on a cluster of VMs provisioned in Openstack by Sahara service – Elastic Data Processing (EDP). Setting up a cluster on Sahara EDP is explained in appendix 4. When a cluster is configured and launched, Sahara orchestrator sends a create VM request to Nova which in turn requests 'glance' for Apache Spark image. Virtual Machines are launched and block storage based distributed file system is created by communicating to the hypervisor (KVM) and orchestrated by heat. The data and job to be processed are stored in the object storage 'swift'. The spark jobs are then obtained by Nova API and processed by the infrastructure managed by Sahara Job Manager.

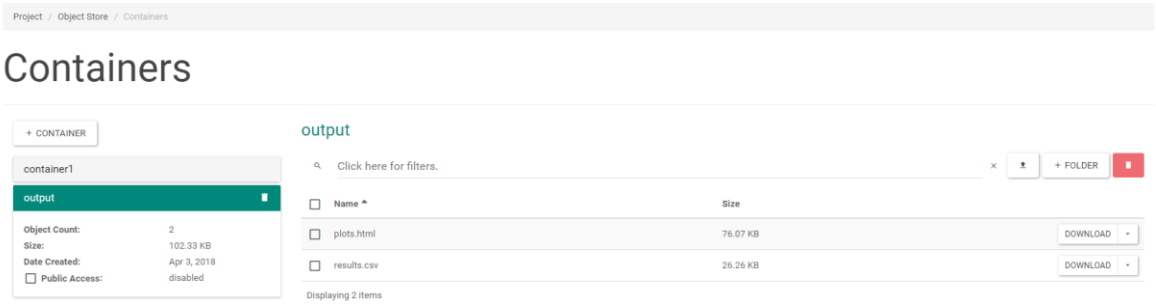


Figure 18. Output of data analysis stored in Swift

3.2.4 Openstack NEAT

In addition to the openstack controller components, the NEAT Global manager also runs in the controller node. Fig. 19 shows the components of Openstack NEAT. NEAT Global Manager makes decisions about mapping virtual machines to compute hosts and initiating migration of the selected VMs. A Local manager runs on each compute host which makes decisions on underload or overload situations and VM selection for migration. A data collector runs on compute nodes locally to collect resource utilization data from hypervisor and sends the data to the central database in the controller [23].

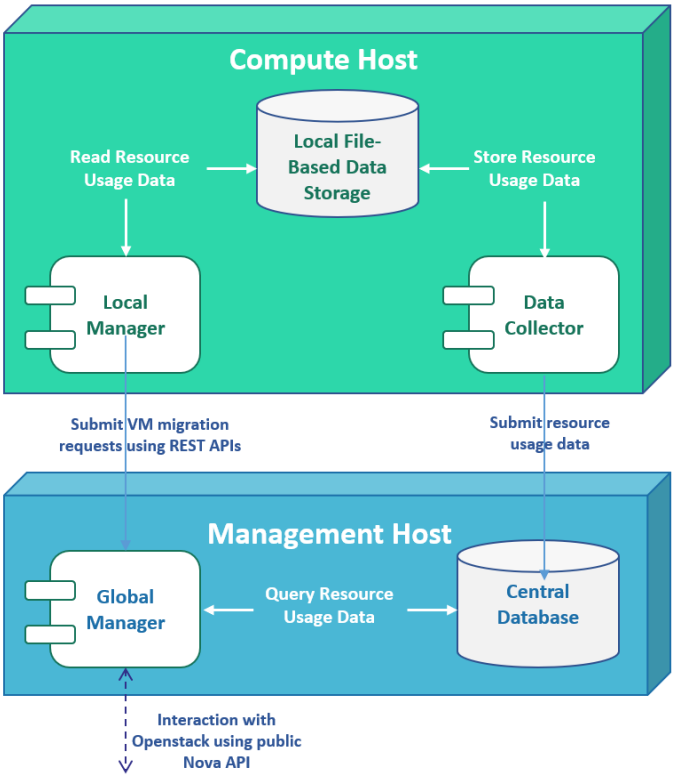


Figure 19. Openstack NEAT Architecture [56]

3.3 VM Consolidation Workflow

A VM consolidation workflow to test and compare the six combinations of algorithms is presented in Fig. 20. Local managers collect resource utilization data from 'Ceilometer' openstack service periodically.

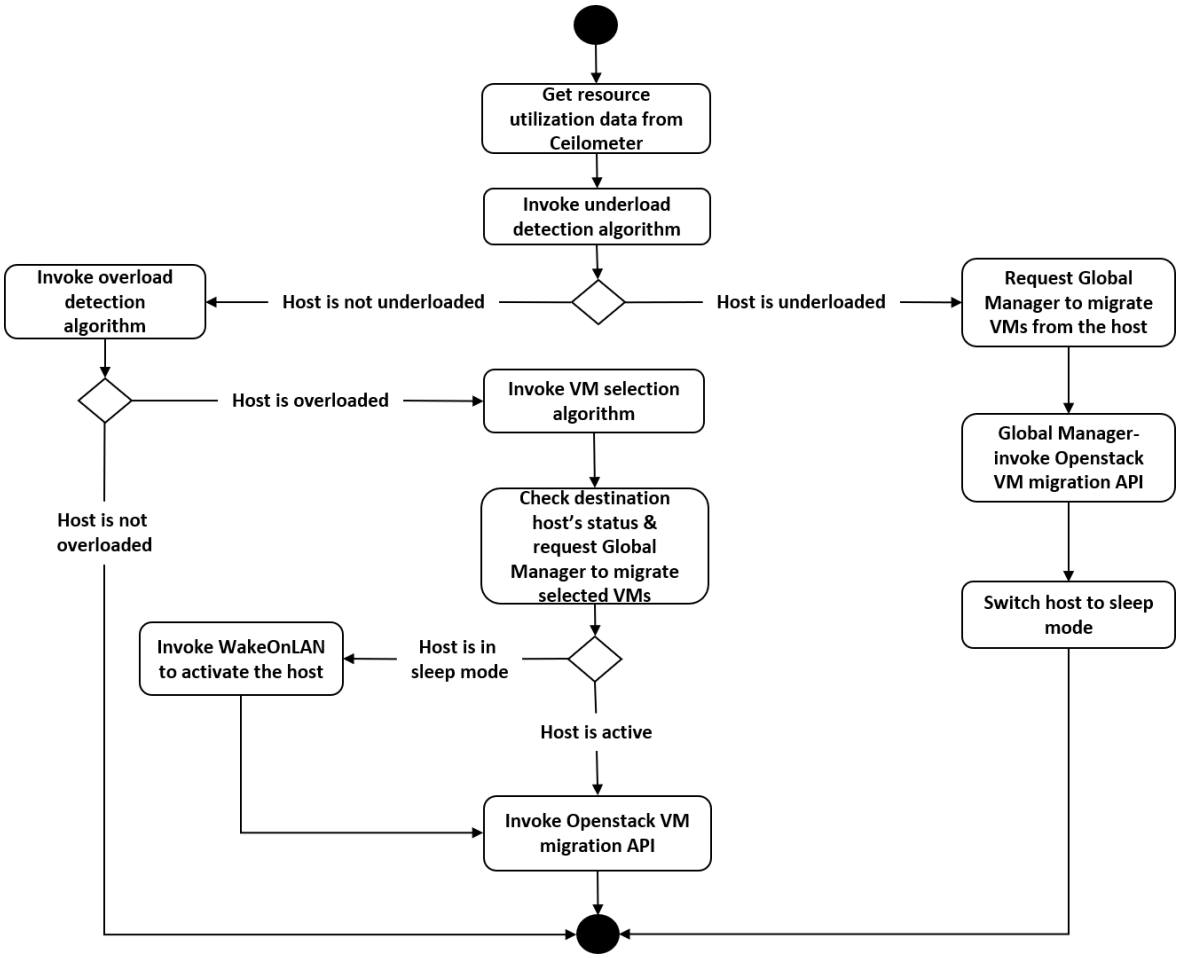


Figure 20. VM Consolidation Workflow

Underloaded hosts are identified by invoking underload detection algorithm; when the host is under-utilized, the local manager requests the global manager to migrate the VMs from the host using openstack VM migration API [63] and puts the host to sleep mode. On the other hand, if the host is not under-utilized, overload detection algorithm is invoked. If the host is not overloaded, the resource monitoring processes continues. If the host is overloaded, VMs to be relocated are selected by invoking the VM selection algorithm. The status of the destination host is checked before the global manager migrates the VMs along with the associated block storage for data processing. If the host is in sleep mode, the host is awakened by sending magic packets using WakeOnLAN standard [64]. The command to live-migrate an instance is shown below:

```
openstack server migrate d1df1b5a-70c4-4fed-98b7-423362f2c47c --live HostC
```

Wake-on-LAN (WoL) is a networking standard that allows to easily turn on one or more computers remotely by sending magic packets to the remote computers. WoL collects MAC addresses of all computers, and save the computers list into a file. It allows you to turn on a computer from command-line, by specifying the computer name, IP address, or the MAC address of the remote network card [64].

3.4 Experimental Details

The design of experiment for this research has three phases: Plan, Execute and Analyze as illustrated in figure 21. The aim, objectives and expected outcome are defined and the required equipment is identified in the 'plan' phase. The experiment is executed for repeated runs or repeated for a specified amount of time. Data is collected at the end of each experiment and saved as csv files. The collected data is analyzed, interpreted and validated. The findings are documented for further study.

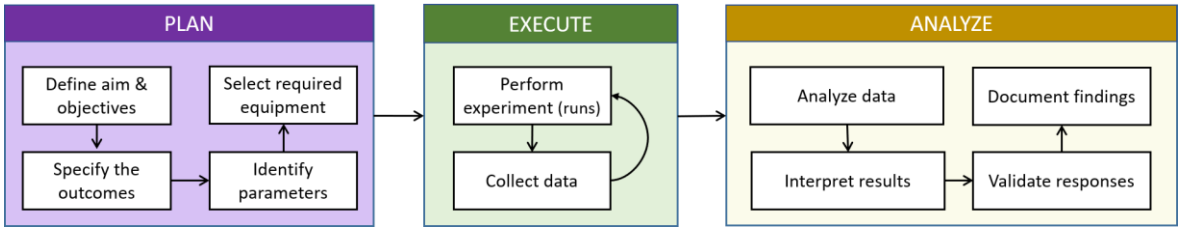


Figure 21. Design of Experiments

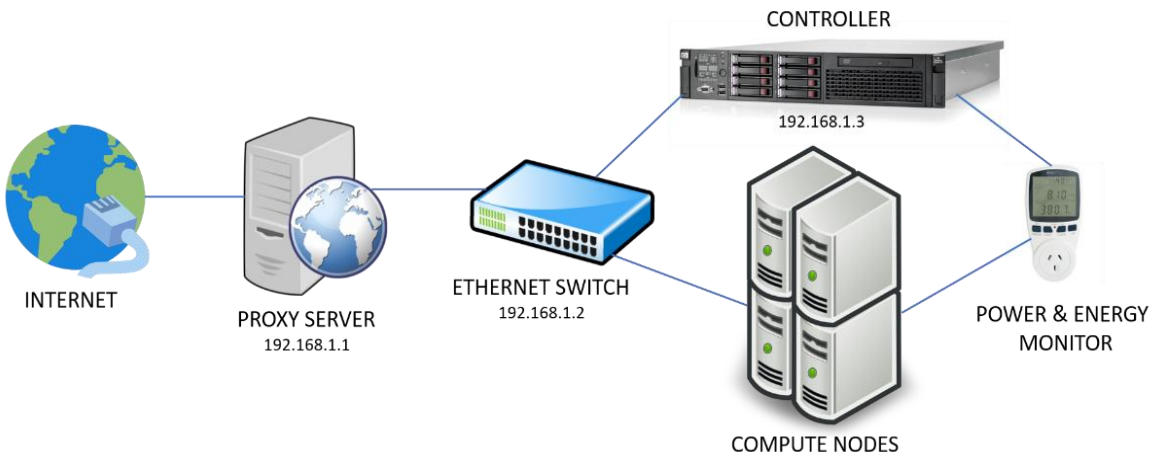


Figure 22. Experimental Set-up

Fig. 22. illustrates the experimental set up. The controller and compute nodes are plugged into the power source through plug-in power and energy monitors. The nodes are connected to the internet through a secure (Squid) proxy server via a 24-port ethernet 'Nortel' switch. Two Network Interface Cards (NICs) are present for each node, NIC1 provides access to the internet whereas NIC2 is connected to the Management or Internal network. In Openstack terms, the public IP obtained by each virtual machine is called the floating IP address [54]. The network architecture of the Openstack set-up is presented in figure 23.

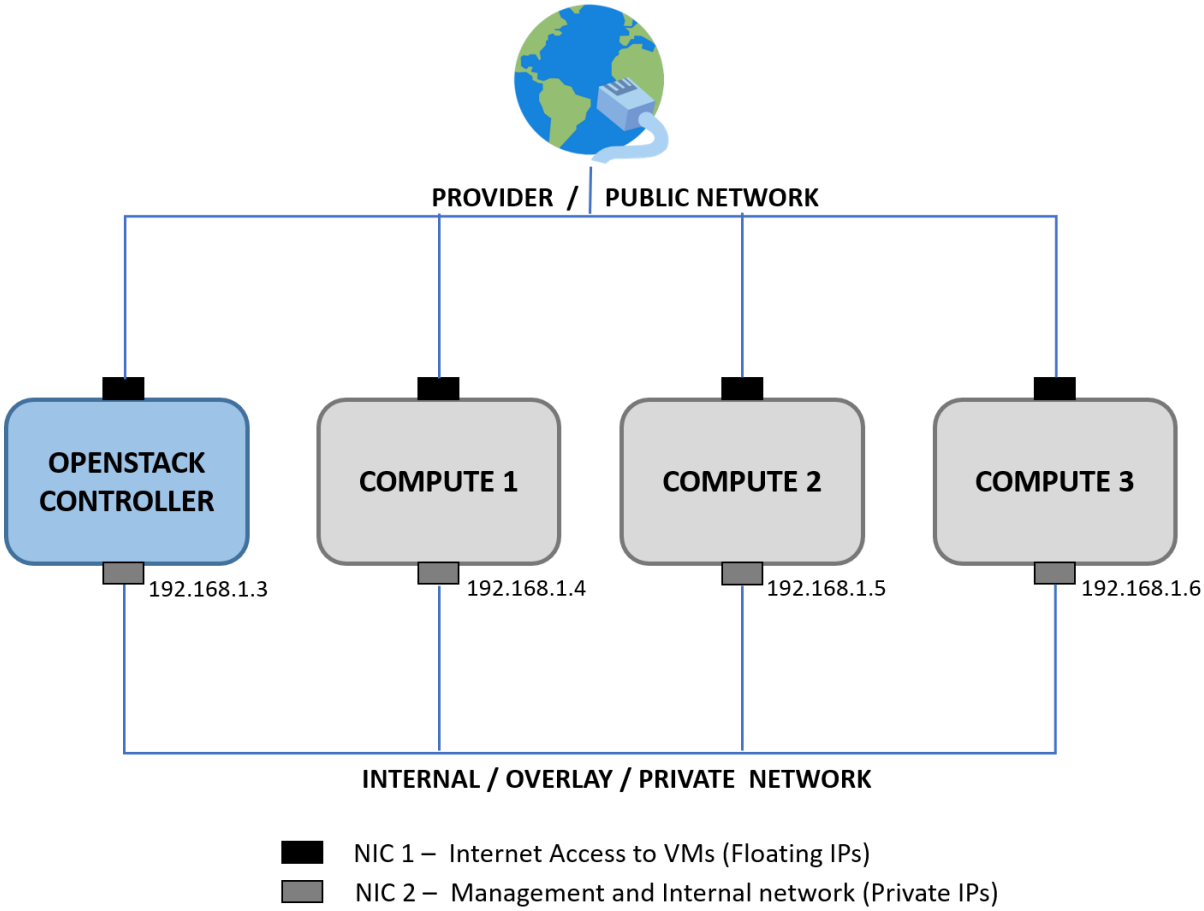


Figure 23. Network diagram

The compute nodes vary in capacity and configurations. Compute1 is an Intel Core i7-3779 CPU @ 3.40 GHz with 8 cores whereas Compute2 is an Intel Core 2 Duo CPU E8400 @ 3.00 GHz with 2 cores and Compute3 is an Intel Core 2 Duo CPU E8500 @ 3.16 GHz with 2 cores. Table. 2 presents the configuration and idle power consumption (IPC) of the nodes.

Host Name	Device	Operating System	Memory (GiB)	Disk (GB)	IPC (Watt)
Controller	HP Proliant DL360p Gen8	Ubuntu 16.04 LTS	70.8	219.1	6.2
Compute1	HP Compaq Elite 8300 MT	Ubuntu 16.04 LTS	15.5	487.7	2.4
Compute2	HP Compaq dc7900 SFF	Ubuntu 16.04 LTS	3.8	242.9	1.9
Compute3	HP Compaq dc7900 SFF	Ubuntu 16.04 LTS	3.6	242.1	1.8

Table 2. Configuration of Servers

Power consumption of compute server varies widely during data processing. The average of the power consumed during a specific period of time is known as energy consumption and the peak value during the period is peak power consumption. Reduction in peak consumption has a positive impact on cost related to power supply and distribution [14]. A set of baseline experiments are run on the infrastructure to analyze the power usage of the compute nodes. The first set of experiments are conducted with simulated load generated using stress-ng. It is a stress test utility to test OS interfaces and sub-systems [65]. The peak power consumption of the controller and compute nodes are observed for different CPU intensive, generic input/output and RAM (Virtual Memory Stressor) workloads.

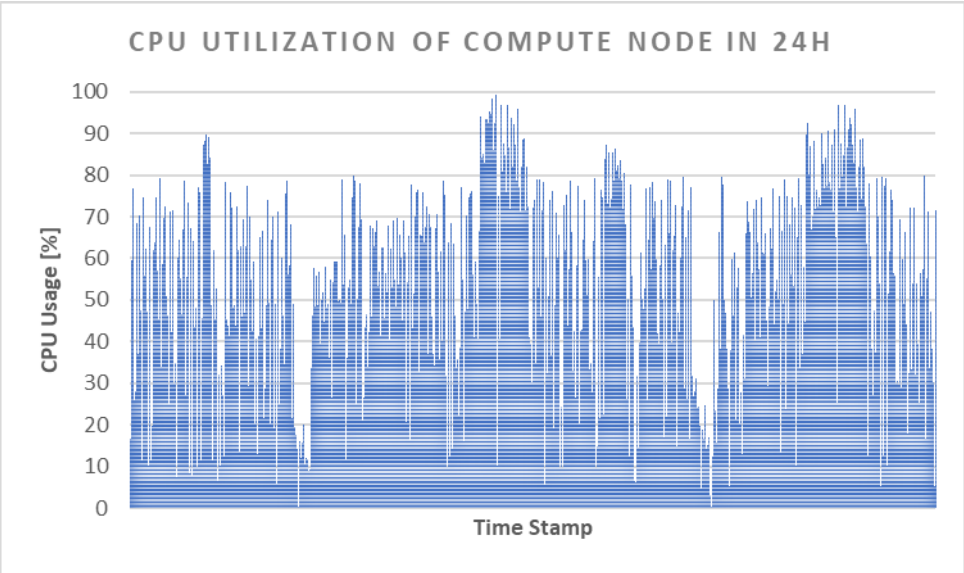


Figure 24. A sample CPU trace of a compute node in 24h

To compare the six combinations of VM consolidation algorithms (presented in Table 1), an experiment was conducted by enabling each ‘Combo’ on Openstack NEAT. Openstack allows over-committing CPU resources at a ratio of 16:1, thus the scheduler can allocate up to 16 virtual cores per physical core [66]. Considering the above fact and the available CPU and RAM resources, the number of VMs that run on the cluster at a time is set to a minimum of 16 and a maximum of 96. As discussed in section C, a big data workload from the IoT system (Fig. 16.) was processed as Spark jobs on the cluster of virtual machines for 24 hours when both power consumption and performance data were collected. The experiment is repeated for each ‘combo’. It is often argued that virtualization causes overhead on servers. Several work conclude that CPU and memory overhead caused by virtualization is insignificant [67]. In this paper, it is relevant to understand the effect of the virtualization layer on power consumption as well. For this baseline is experiment, no workload is applied on the compute nodes. The peak power consumption, CPU and memory utilization of the compute nodes are recorded when no virtualization is enabled and when KVM, Openstack and Openstack NEAT services are enabled. The results of the above discussed experiments are presented in the next chapter.

4 RESULTS AND DISCUSSION

This chapter presents the results of the experiments conducted and the inferences.

4.1 Peak Power Consumption for Synthetic Workloads

As discussed in the previous chapter, stress-ng is used to synthetically stress the compute nodes with CPU, I/O and RAM workloads. The number of cores to be stressed, number of I/O tasks and amount of RAM are provided as input. The experiment is conducted for a time-period of 60 seconds. Workload is applied in percentages from 0 to 100 in an interval of 10. This experiment is repeated 10 times and the average Peak Power Consumption (PPC) in Watt is observed. The results are tabulated in Table 3 and depicted as graphs in Figures 25 and 26.

Workload (%)	Average Peak power Consumption in Watt					
	Compute1		Compute2		Compute3	
	1 core	All cores	1 core	All cores	1 core	All cores
0	26.8	30.9	39.2	39.5	39.8	40.2
10	32.9	42.1	44.0	44.9	42.8	45.1
20	35.1	48.2	47.4	48.9	47.5	49.1
30	37.1	50.8	49.4	54.0	49.7	54.4
40	38.1	55.8	50.9	57.9	50.8	58.6
50	38.6	60.2	51.6	58.3	52.8	62.1
60	40.0	71.1	52.4	60.6	54.5	66.6
70	40.5	80.1	55.0	61.3	56.0	68.0
80	42.1	83.3	56.4	63.4	58.9	69.9
90	47.1	84.9	57.3	65.9	60.9	71.2
100	52.1	87.6	58.7	68.4	63.1	72.1

Table 3. Peak Power Consumption for Synthetic Workloads

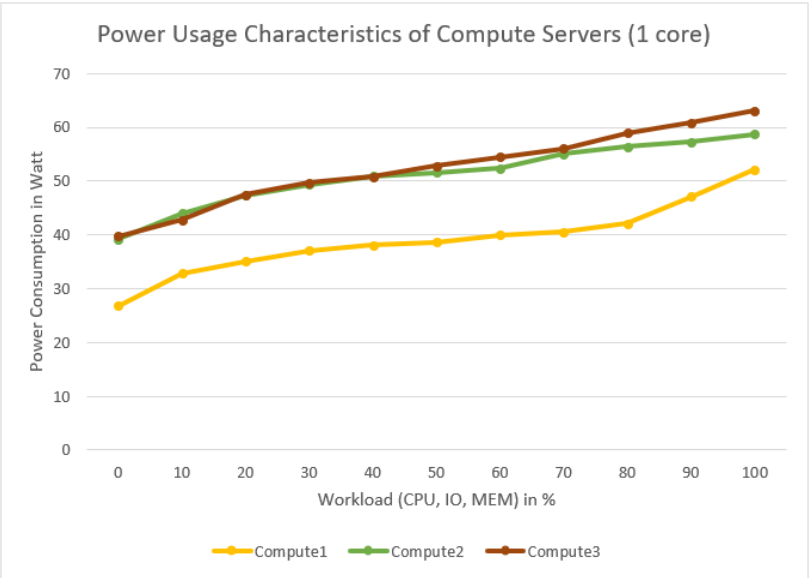


Figure 25. PUC of Compute nodes (1 Core)

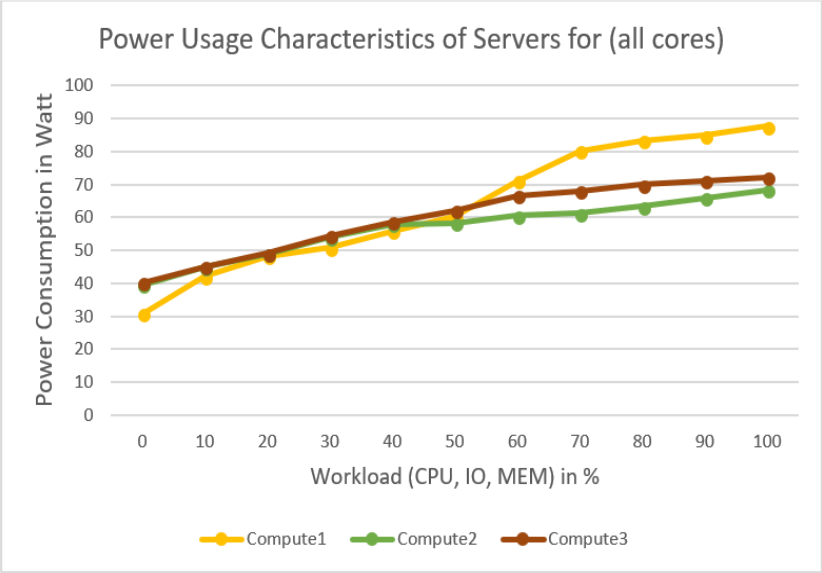


Figure 26. PUC of Compute nodes (all cores)

It is observed that for similar workloads, the Power Usage Characteristics (PUC) of servers with single core and multiple-cores are different [21]. From Table 3 and Figure 25, it is evident that Compute1 consumes less power compared to the other compute nodes when one of the cores is stressed. The i7 processor is optimized for power consumption compared to the Core 2 Duo processors [21]. On the contrary, when all the cores are stressed, the PPC of Compute1 changes drastically when the workload is increased to 60%. The turbo boost feature of i7 processor is responsible for this behavior. The turbo boost feature of i7 processors reduce up to 6% percent of the execution time at the cost of increasing the energy consumption by 16% [51].

On comparing Compute2 and Compute3, the PPC on an average is 56.2W when workload is 50% for both cases. When the workload increases from 60 to 100%, Compute3 tends to consume more power than Compute 2. Several factors could be responsible for this behavior, one of which is the electronic hardware ageing phenomenon [52]. The above analysis shows that to reduce overall energy consumption during data processing, it is important to reduce the peak power consumption by effectively identifying the underloaded and overloaded hosts, followed by reducing the number of active hosts by putting the others to an idle mode. The idle mode power consumption of the compute nodes is negligible as shown in Table 2.

4.2 Performance Metrics

To compare the efficiency of the six VM consolidation approaches, the following metrics are used to evaluate the performance.

4.2.1 Total Energy Consumption (E)

It is the sum of energy consumed by the compute servers as a result of application workloads over a specific period of time. It is measured in kilowatt hour [43].

4.2.2 Number of VM Migrations

VMs are selected to be migrated once a host is identified to be underloaded or overloaded. Minimizing the time for migration is a crucial step which is achieved by reducing the total number of VM migrations.

4.2.3 Power State Changes

The number of state changes (on and off) of compute nodes must be minimal to avoid unnecessary loss of energy.

4.2.4 Service Level Agreement (SLA)

QoS requirements of a system are devised in the form of SLA, determined by attributes such as throughput or response time which are application dependent. In case of IaaS, QoS can be evaluated using SLA metrics that depend on VM and compute resources [52]. IaaS – SLA violations (SLAV) can be measured using two metrics:

- ***SLATAH:*** SLA violation Time per Active Host is the period of time when a host experiences 100% CPU utilization and the requested performance is not delivered as it is limited by the node's capacity causing a violation of the SLA as shown in Eq. (3):

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \qquad (3)$$

  where N is the number of compute nodes, T_{si} is the total time during which the host i experienced 100% utilization leading to an SLA violation, T_{ai} is the total time when host i actively provided VMs [68].

- ***PDM**:* The overall degradation in performance experienced during migration of virtual machines as shown in Eq. (4):

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{rj}} \quad (4)$$

  where M is the number of Virtual Machines, C_{dj} is the performance degradation of VM_j caused by migrations, C_{rj} is the total processor capacity requested by VM_j. In general, C_{dj} is assumed to be 10% of CPU in Million Instructions Per Second (MIPS) during migrations [68].

As SLATAH and PDM are two independent metrics, ***SLA Violation (SLAV)*** is a metric that combines both performance degradation caused by overloading as well as VM migrations as shown in Eq. (5):

$$SLAV = SLATAH * PDM \quad (5)$$

It denotes the violation that takes place when the promised QoS is not met [68].

4.2.5 Energy and SLA Violations (ESV)

Energy Consumption (E) of compute nodes and SLAV are negatively correlated as energy consumed can be reduced at the cost of increased SLA violations. Whereas, the goal of an energy-efficient system is to minimize energy as well as SLA violations. Hence, a combined metric Energy and SLA Violations (ESV) proposed in [46] is shown in Eq. (6):

$$ESV = E * SLAV \quad (6)$$

A lower ESV value indicates that energy saving is higher than SLA violations.

4.3 Performance Evaluation

The results of the experiment which investigates on the impact of six different VM consolidation 'combos' on energy consumption is illustrated in Fig. 27.

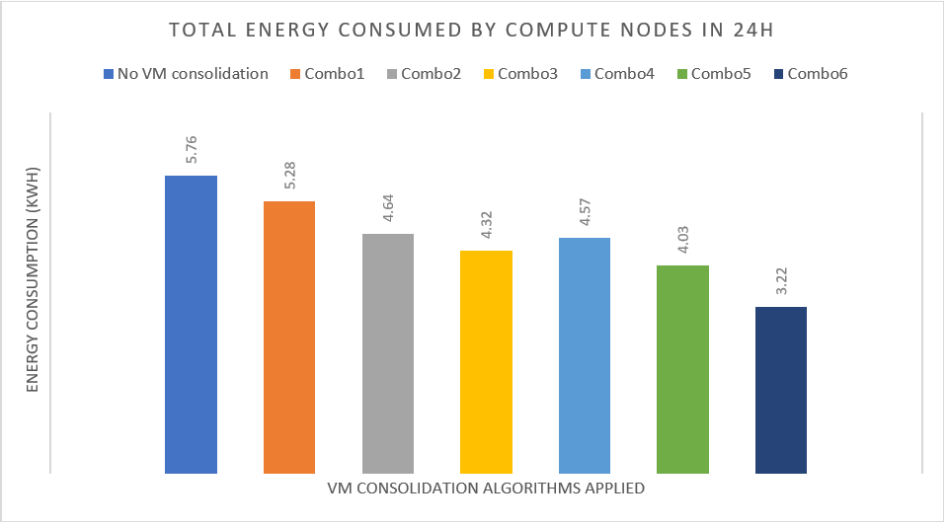


Figure 27. Total Energy Consumption of Compute Nodes in 24 hours

It is observed that VM consolidation in general, saves as little as 8.33% of energy. 'Combo6' clearly outperforms all the other algorithms by reducing energy consumption by 44.1% saving 2.54 kWh (within a 24-hour duration) of electrical energy by switching the underloaded compute nodes to sleep mode. Next in line, Combo5 and Combo3 save up to 1.73 kWh and 1.44 kWh respectively. It is observed that VM selection plays a crucial role in energy saving as Random Choice (RC) cause aggressive migrations consuming energy which is mitigated by Minimum Migration Time (MMT) algorithm [20]. MMT when applied along prediction based Local Regression Robust (LRR) and statistical Median Absolute Deviation (MAD) algorithms accomplish substantial energy saving.

Effective identification of overloaded / underloaded hosts and VMs to be migrated is crucial in VM consolidation as aggressive VM migrations lead to unnecessary energy loss [52]. In addition, power state changes between sleep and on states should be kept to a minimum [45]. Fig. 28 and 29 compare the number of VM migrations and power state changes of the six approaches.

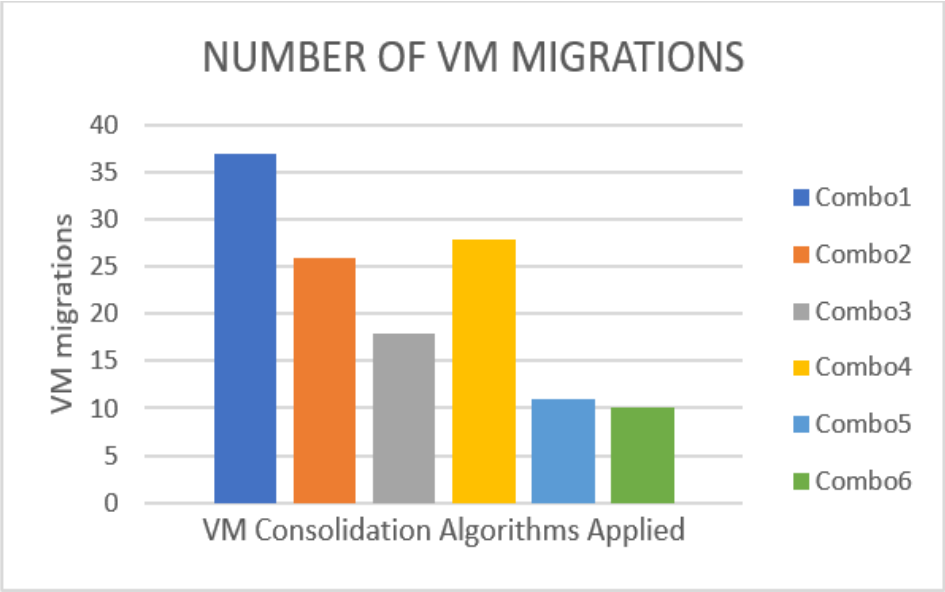


Figure 28. Number of VM Migrations

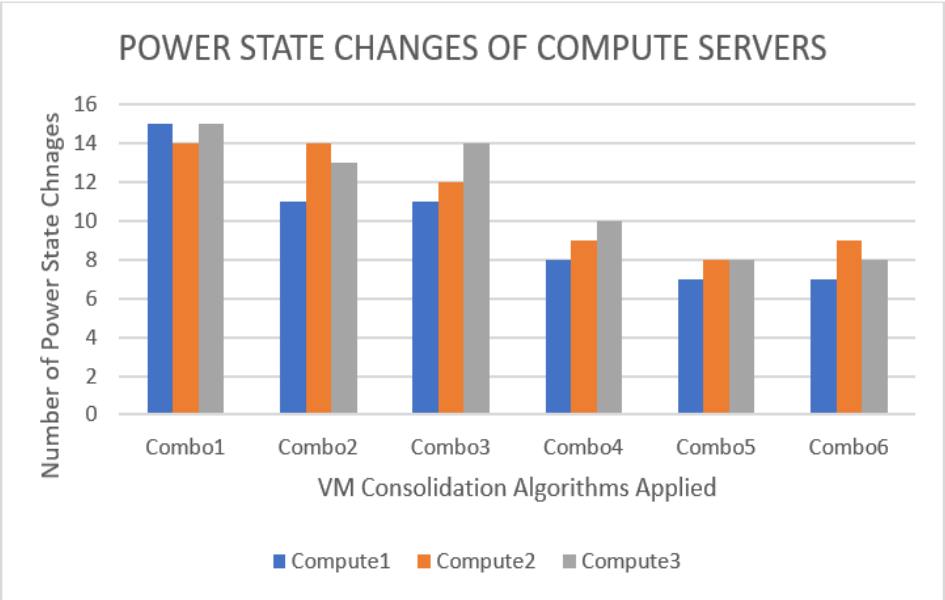


Figure 29. Power State Changes

From Fig. 28, it is seen that Combo6 has the least number of VMs migrated. The combination of LRR's prediction of resource utilization and MMT's strategy to select VMs based on minimum time taken to migrate is effective in saving energy causing least migration overhead. Combo5 is the second-best method with less VM migrations. From Fig. 29, it is observed that the Minimum Migration Time (MMT) algorithm performs better than Random Choice (RC) in keeping power state changes to an optimal level as 'combos' that employ MMT, Combo 4, 5 and 6 have less changes in power states than combo 1, 2 and 3 that apply RC algorithm.

The following discussion summarizes the results obtained by studying the impact of the six VM consolidation approaches in terms of SLA compliance. SLA violations are caused by both over-utilization of resources (performance degradation due to 100% resource utilization) and degradation caused by extensive VM migrations as defined in [46]. SLA metrics that define SLA violations are SLATAH and PDM as discussed in the previous section. Table 4 presents the comparison of Energy and SLA violation metrics of the six approaches.

Combo	E	SLATAH	PDM	SLAV	ESV
Combo1	5.28	17.67	0.31	5.4777	28.92226
Combo2	4.64	14.54	0.28	4.0712	18.89037
Combo3	4.32	18.56	0.22	4.0832	17.63942
Combo4	4.57	19.25	0.23	4.4275	20.23368
Combo5	4.03	21.42	0.09	**1.9278**	7.769034
Combo6	**3.22**	65.23	0.03	1.9569	**6.301218**

Table 4. Energy and SLA Violation Metrics

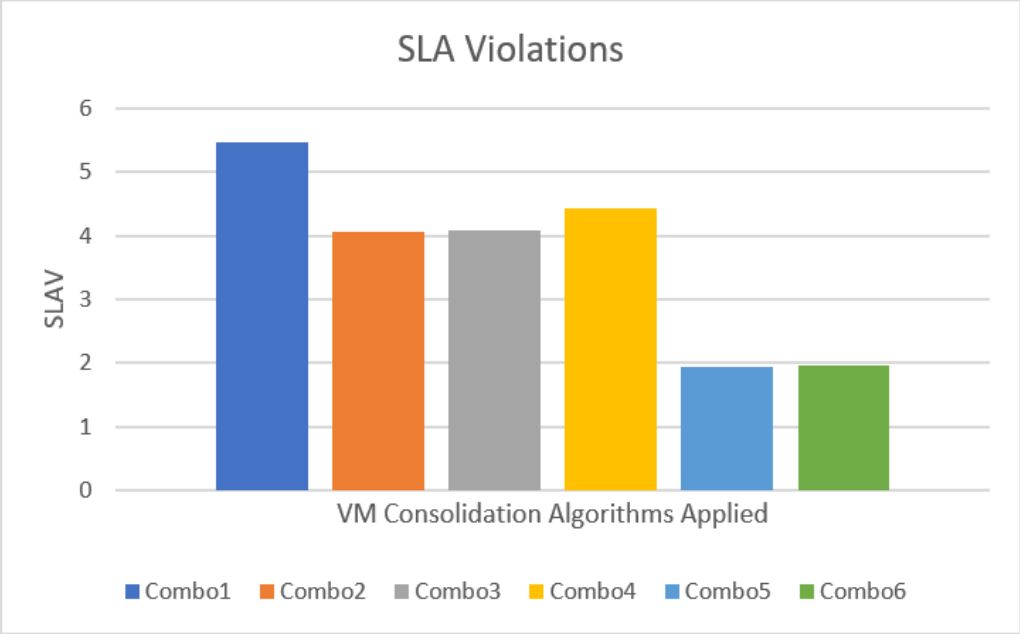


Figure 30. SLAV

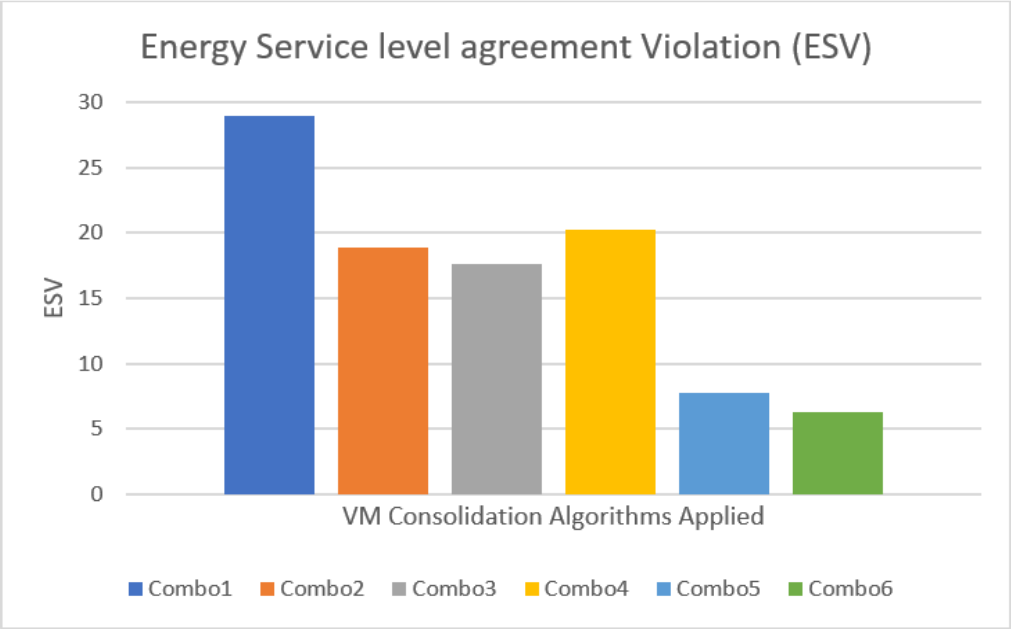


Figure 31. ESV

SLA Violations (SLAV) metric is computed from SLATAH and PDM for each combo. From Table 4, it is observed that Combo5 (MAD and MMT algorithms applied) has the least SLA violations followed by Combo6 (LRR and MMT). Threshold based Heuristic (THR) and Random Choice (RC) algorithms cause most SLA violations and are not as effective as MMT, MAD and LRR. The Energy Service level agreement Violation (ESV) metric in case of Combo6 is reduced by the energy consumption factor. The balance between energy saving and SLA violations is expressed by ESV [50]. Though Combo5 has less SLA violations, the energy consumed (4.03 kWh) is more compared to Combo6 (3.22 kWh). Fig. 30 and 31 present graphs of SLAV and ESV respectively. The results obtained are similar to the results presented in [46].

Host	Virtualization State	Peak Power Consumed (Watt)	CPU (%)	Memory (%)
Compute1	A[1]	26.7	0.34	0.29
	B[2]	27.8	0.42	0.36
Compute2	A	39.2	0.25	0.19
	B	39.9	0.33	0.25
Compute3	A	39.8	0.27	0.21
	B	40.9	0.35	0.28

[1] No Virtualization enabled;
[2] Virtualization enabled by KVM, Openstack and Openstack NEAT

Table 5. Effect of Virtualization on Energy Consumption and Resource Utilization

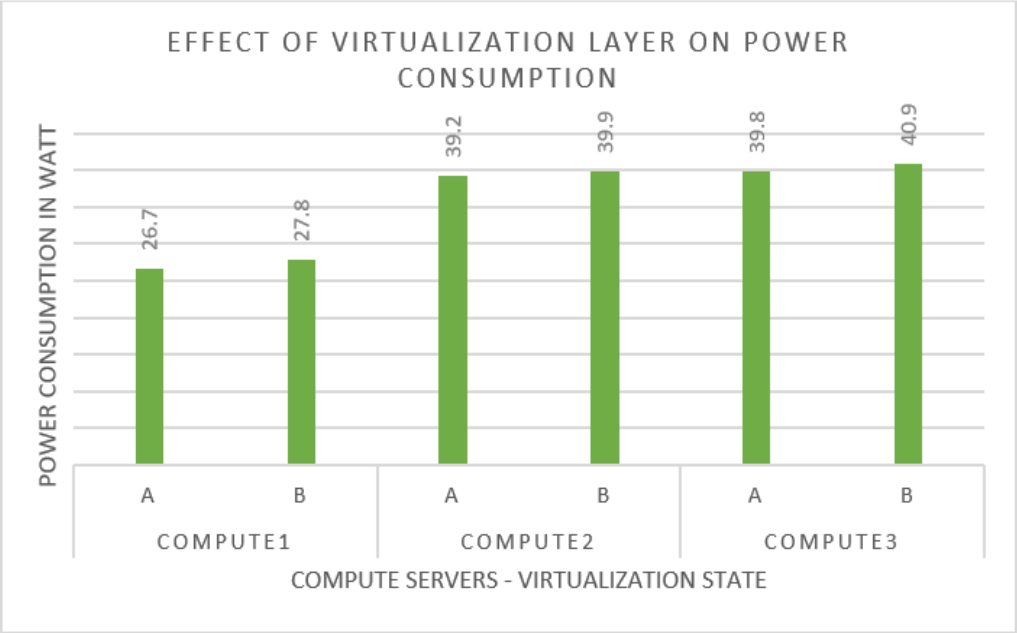


Figure 32. Effect of Virtualization layer on Peak Power Consumption

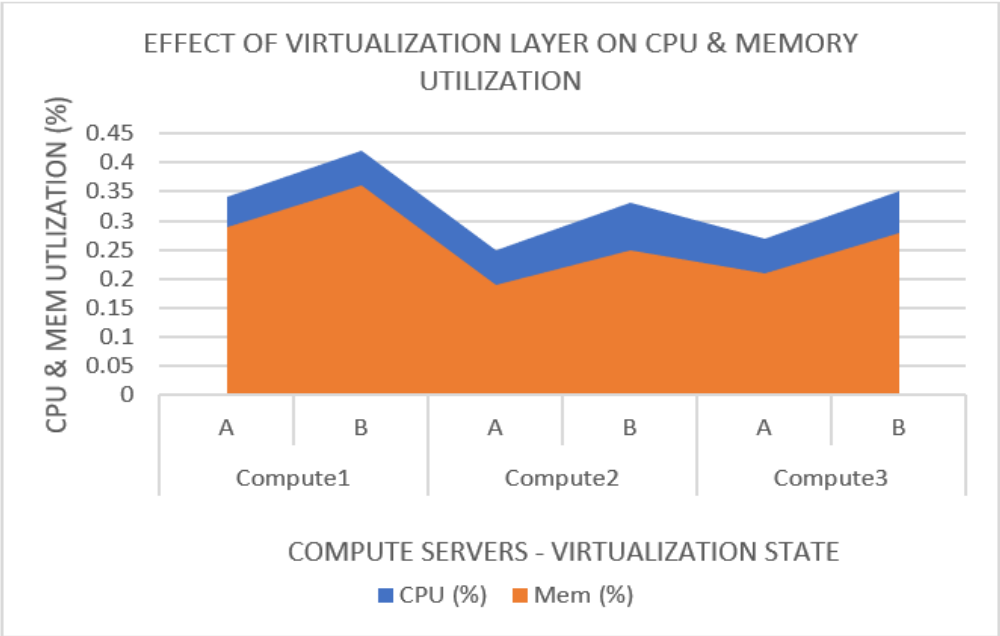


Figure 33. Effect of Virtualization layer on CPU & Memory Utilization

The experimental results for the effect or overhead caused by the virtualization layer on peak power consumption, CPU and memory utilization are presented Table 5 and figures 32 and 33. Notations ‘A’ and ‘B’ denote ‘No virtualization enabled’ and ‘Virtualization enabled by KVM, Openstack and Openstack NEAT’ respectively. It is clear that the increase caused by the virtualization layer is negligible of the order of less than 1 Watt of power and less than 1% of CPU and memory.

4.4 Sustainability Analysis

The PERCCOM program aims at combining ICT with environmental awareness to build cleaner, greener, more resource and energy efficient cyber-physical systems [69]. It is important to highlight the theme of this research work with respect to sustainability. A development is considered to be sustainable when "it meets the needs of the present without compromising the ability of future generations to meet their own needs" [70]. Correlating this statement to energy, it is a resource that is depleting day by day. Two main approaches to solve this problem are, effectively utilizing the available energy and finding ways to utilize renewable sources. This research work contributes towards the former by making efficient use of available energy.

Relating to the three-pillar or three-dimensions approach to sustainability, this research work contributes towards two of the three pillars directly.

- ***Environmental**:* This work focuses on reducing CO_2 emission by reducing the numbers of active compute hosts that consume electrical energy. This is achieved by making optimal use of the host's resources. This not only decreases the amount of energy utilized but also increases the lifetime of the equipment in the data center thereby reducing e-waste.
- ***Economical**:* The decrease in energy consumption translates to reduction of operational and utility costs in the cloud data center as well as the cost involved in production and transmission of electrical energy. Furthermore, the increase in lifetime of hardware resources reduces cost of procuring new resources.

Still, the analysis of sustainability aspects of the solution cannot be restricted to the three-dimensional approach as it can have indirect or enabling effect on more inter-related dimensions.

4.4.1 Five Dimensions of Sustainability

Becker et. al [71] proposed a five dimensional model for sustainability to understand sustainability issues from a broader perspective. The following are the five dimensions proposed.

- Individual
- Social
- Economic
- Technical
- Environmental

The five-dimensional approach considers the immediate effects, longer running aggregate effects (enabling effect) and the cumulative impact (structural effect) on each of the dimensions. Based on the model, we conducted a sustainability analysis of the system which is depicted in Fig. 34.

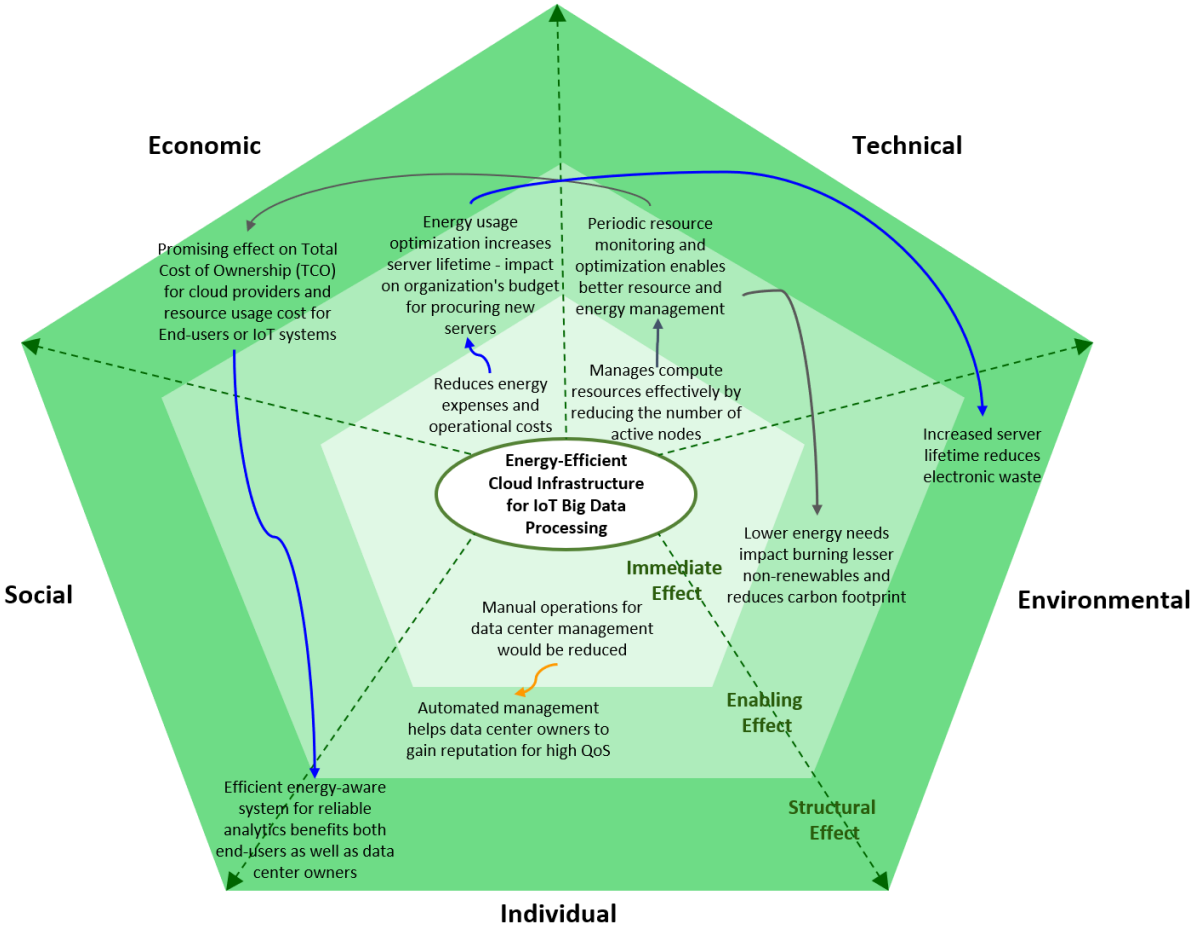


Figure 34. Five-dimensions of Sustainability

4.4.2 Analysis based on World Energy Council Data

The World Energy Council (WECouncil) consists of a community of energy practitioners and leaders who aim to encourage economical, reliable and environmentally sustainable energy system that helps everyone. WECouncil define energy sustainability based on three dimensions namely, energy security, energy equity and environmental sustainability. Together, a balance attacined in these three goals form the 'Energy Trilemma' which forms the basis of accomplishment and competitiveness among countries.

The definitions of the goals of energy trilemma as stated by the WECouncil Report 2017 are:

Energy security: "Effective management of primary energy supply from domestic and external sources, reliability of energy infrastructure, and ability of energy providers to meet current and future demand" [72].

Energy equity: "Accessibility and affordability of energy supply across the population" [72].

Environmental sustainability: "Encompasses achievement of supply and demand-side energy efficiencies and development of energy supply from renewable and other low-carbon sources" [72].

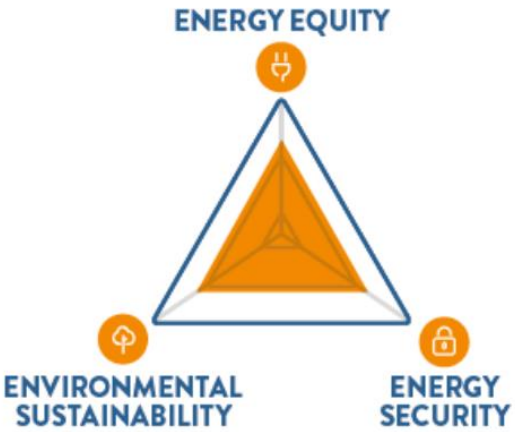


Figure 35. Three-dimensions of Energy Trilemma [72]

125 Countries were assessed based on the three aspects and the results and ranking of countries were presented as the 'World Energy Trilemma Index 2017'. The top ten countries for the year 2017 in the list are presented below:

1. Denmark
2. Sweden
3. Switzerland
4. Netherlands
5. UK
6. Germany
7. Norway
8. France
9. New Zealand
10. Slovenia

In addition to the ranking, the report provides open data for the listed 125 countries on factors such as Energy Intensity, Electricity cost, CO2 intesity etc., Table. 6 presents some of the open data provide for a selected list of 14 countries. The countries which are prime locations of hyperscale datacenters [73] and countries that are suitable locations for data centers [74] are chosen. The data presented in the table and graphs as defined by WECouncil [72] are as follows:

Industrial sector (% GDP) - % of total GDP that is in the industrial sector (CIA World Fact Book, 2014)

Energy intensity (koe per US$) - Measures how much energy is used to create one unit of GDP (Enerdata & World Energy Council, 2014)

Electricity cost (US$/kWh) - Average cost of electricity (IEA, Eurostat, World Energy Council, World Bank, 2015)

CO2 intensity (kCO2 per US$) - Measures CO2 from fuel combustion to generate one unit of GDP in PPP (Enerdata and World Energy Council , 2014)

GHG emission growth rate 2010 – 2014 (%) - Greenhouse gas emission growth rate from the energy sector between 2000 and 2012, (WRI/CAIT, 2012)

Country	Industrial Sector (% of GDP)	Energy Intensity (koe per USD)	Electricity cost (USD per kWh)	CO2 intensity (kCO2 per USD)	GHG emission growth rate % (2010-2013)
US	20	0.09	0.21	0.34	-0.8
UK	19.4	0.05	0.24	0.17	-1.4
Sweden	26.3	0.08	0.22	0.1	-2.6
Denmark	22.9	0.07	0.34	0.16	-2.2
France	19.5	0.07	0.2	0.15	-1.5
Germany	30.5	0.07	0.33	0.24	-0.7
Finland	26.9	0.13	0.18	0.25	-0.9
Latvia	22.9	0.1	0.18	0.19	0.2
Canada	27.7	0.13	0.16	0.4	0.4
Brazil	22.3	0.08	0.14	0.17	3.4
New Zealand	21.8	0.1	0.1	0.26	0.1
Japan	28.9	0.07	0.1	0.27	0.4
China	40.9	0.12	0.09	0.54	8.5
India	29.6	0.08	0.08	0.3	5.2

Table 6. WECouncil 2017 Open Data

The following figures (Fig. 36, 37 and 38) depict the comparison of Industrial Sector (% of GDP) and the growth rate of Green House Gas Emissions, CO_2 Intensity and Ranking of selected countries in terms of Overall performance and in terms of environmental sustainability respectively. The data presented in these graphs are based on the open data provided in WECouncil 2017 Report [72].

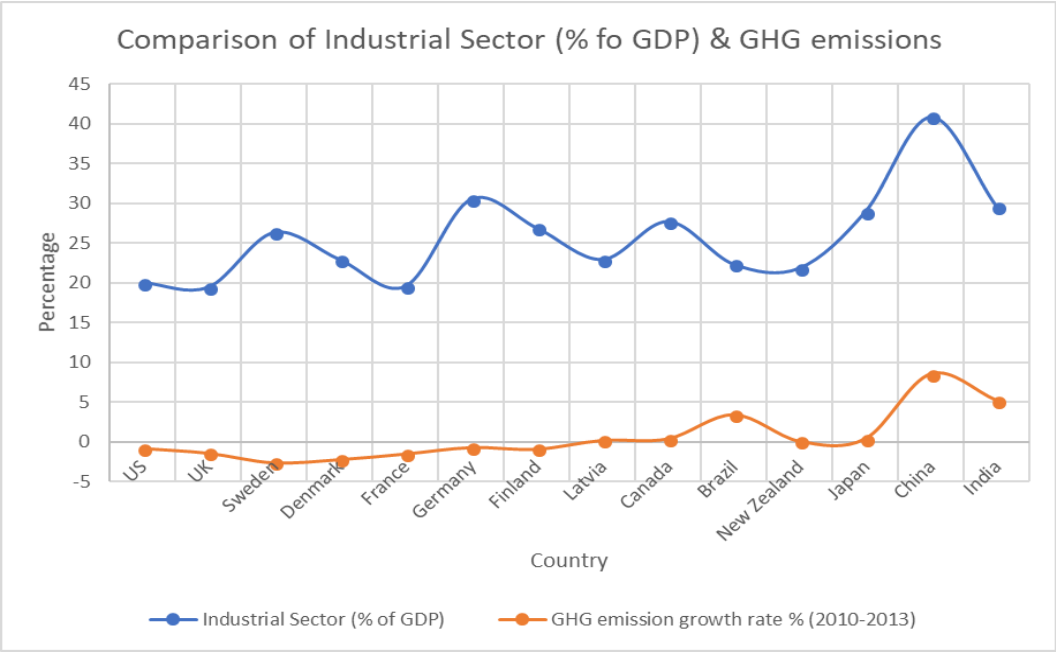


Figure 36. Comparison of Industrial Sector (% of GDP) and GHG emissions growth rate (%)

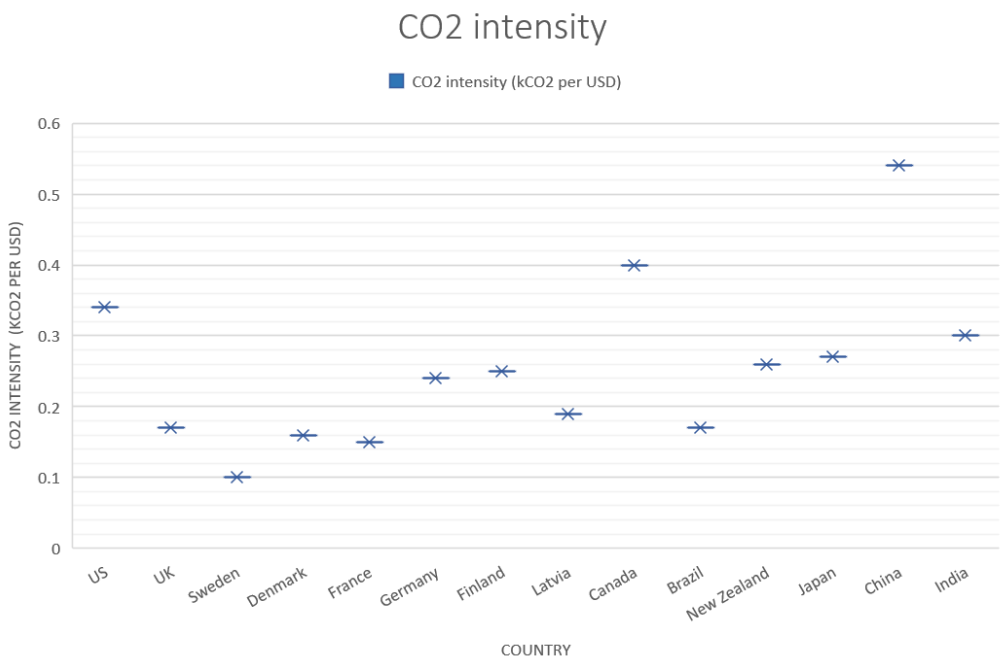
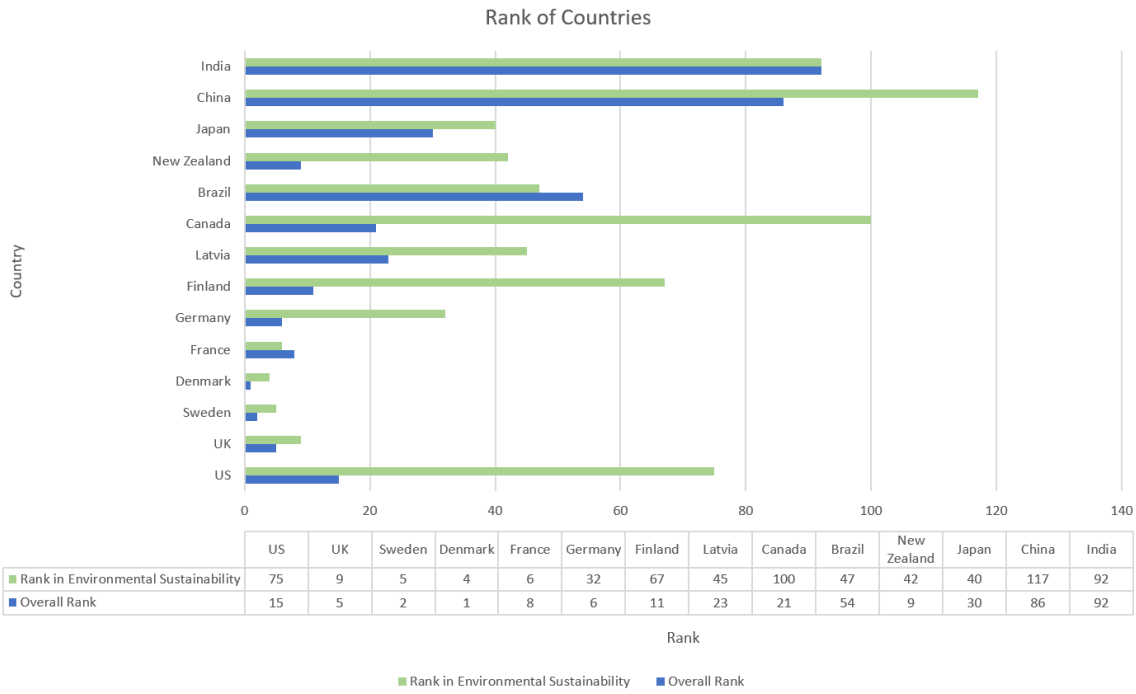


Figure 37. Carbon dioxide Intensity of 14 selected countries



	US	UK	Sweden	Denmark	France	Germany	Finland	Latvia	Canada	Brazil	New Zealand	Japan	China	India
Rank in Environmental Sustainability	75	9	5	4	6	32	67	45	100	47	42	40	117	92
Overall Rank	15	5	2	1	8	6	11	23	21	54	9	30	86	92

Figure 38. Ranking of Selected Countries

Based on the results obtained from Performance evaluation of different VM Consolidation strategies, it is evident that Combo6 is the best combination of VM consolidation algorithms for the given IoT Big data workload. In order to understand the economic and environmental sustainability implications of Combo6, a projection on the cost of electrical energy for running the compute nodes and carbon emissions for the required energy generation for a period of 30 days is calculated. Since the scope of this research is to study the energy usage of compute nodes, energy cost is calculated only for running the compute nodes; controller and other ICT equipment such as the ethernet switch are not taken into account.

Energy cost in USD and carbon dioxide emission in KgCO2 for a month is calculated using Energy Council (2017) data presented in Table 6 on electricity-specific energy generation cost and carbon emission of countries [75]. Cost and carbon emission for wastage during energy generation and transportation are not taken into account. A comparison was made between ‘When no VM consolidation is applied’ and ‘Combo6’ for processing the same IoT Big Data workload. Fig. 39 and 40 represent the projected energy cost and projected carbon dioxide emission for required energy generation in various countries for a period of 30 days.

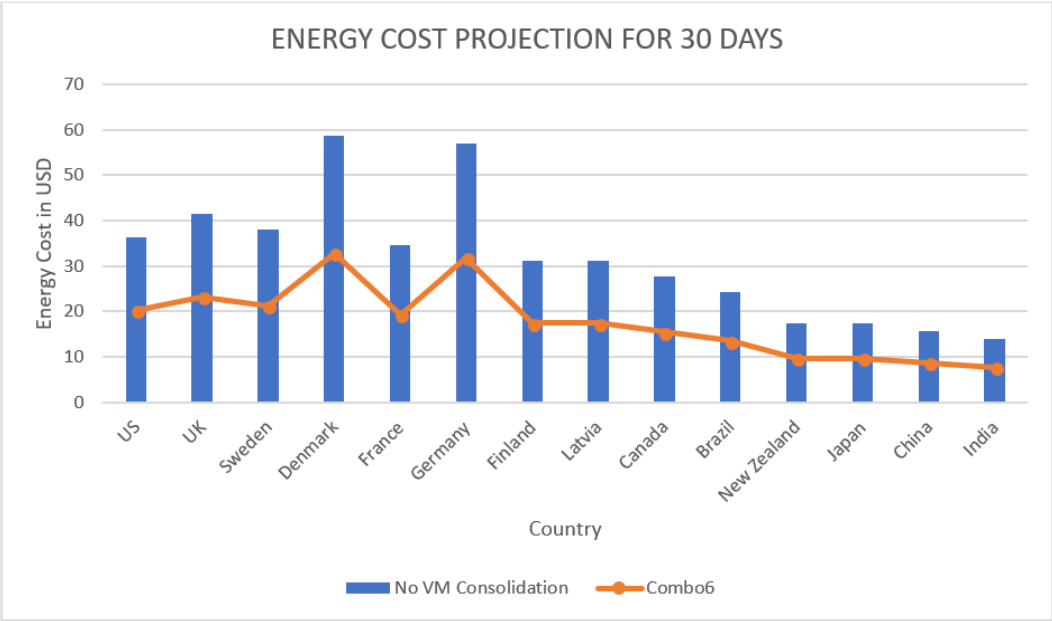


Figure 39. Energy Cost Projection for 30 days

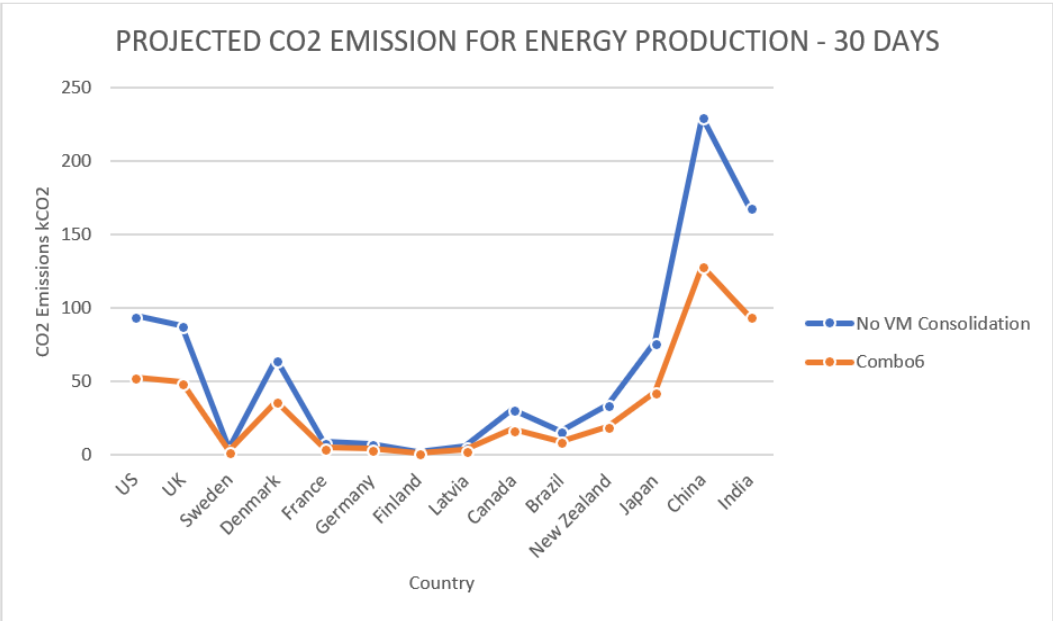


Figure 40. Projected Carbon dioxide emission for Energy generation - 30 days

It is observed that there is a significant decrease in the energy cost and carbon dioxide emission with Combo6 for VM consolidation in each country. In countries such as Denmark that generate energy from renewable energy sources, cost per kWh is as high as 0.34 USD. Applying Combo6 for VM consolidation can save up-to 25.908 USD for a small 3-node set up. Countries with colder climates are often preferred for locating data centers as there is no need for additional cooling systems. Countries such as China, India and Japan are becoming popular data center locations for the availability of labor, connectivity and cost of electrical energy. Though the cost of electricity generation in these countries is less, (Eg., China - 0.09 USD per kWh), the amount of carbon dioxide and other Green House gases (GHG) emitted are very high (Eg., China - 1.33 kgCO2/kWh) compared to countries like Finland (0.01 kgCO2/kWh) and Sweden (0.02 kgCO2/kWh) that primarily use renewable sources of energy. With Combo6, there would be 101.5 kg less carbon dioxide emission in China and 0.8 kg less in Finland for a period of 30 days to run the three compute nodes. Applying energy saving systems and approaches such as the most suitable VM consolidation technique in data centers can save substantial amount of money and a great impact on the environment not only by reducing carbon dioxide emissions but also by increasing the lifetime of the computing systems, therefore less electronic waste [33].

5 CONCLUSION AND FUTURE WORK

This thesis has investigated the energy consumption and performance of compute hosts for IoT Big data processing in a cloud infrastructure. The results obtained from real compute resources overcome the drawback of testing VM consolidation on simulated environments with simulated workloads. From a data center's perspective, compute hosts and cooling systems are major consumers of energy. In addition to hardware and application efficiency, cloud resource management also plays a key role in energy saving. VM consolidation reduces the overall energy consumption thereby reducing the utility and operational costs. From the observations made, it is clear that power consumption varies based on workload. So, it is important to choose the apt VM consolidation algorithms for each workload.

Further, an energy-efficient system is effective when it meets the QoS requirements in addition to energy saving. Therefore, selection of apt VM consolidation algorithms for IoT big data workload considering energy and less SLA violations to meet QoS requirements is essential. For IoT Big data workload, regression based LRR algorithm outperforms static threshold-based THR and adaptive threshold-based MAD algorithms for overload detection. Combo6 with Local Regression Robust (LRR) overload detection algorithm and Minimum Migration Time (MMT) VM selection predicts resource utilization and chooses VMs that require minimum time to migrate, is recommended as it performs better than other combinations.

The additional overhead caused by virtualization on the compute hosts is negligible considering the value it brings in. It can also play a vital role in countries that generate electricity from fossil fuel thereby reducing the negative impact on the environment by burning lesser non-renewables. This work aptly falls under the theme 'Green Technologies and IT'. The system can be altered to become energy-aware by enabling 'Energy Monitoring as a Service' for the compute hosts. Further, an energy-efficient cloud system must be robust and scalable. The global manager of Openstack NEAT is centralized; a distributed model of the VM consolidation framework can avoid single point of failure. Analyzing VM consolidation algorithms by applying such a distributed framework for more number of compute nodes and different big data platforms could be the future direction of this research.

REFERENCES

[1] J. G. Koomey, "GROWTH IN DATA CENTER ELECTRICITY USE 2005 TO 2010," 2011.

[2] M. Marjani *et al.*, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," *IEEE Access*, vol. 5, no. c, pp. 5247–5261, 2017.

[3] R. L. Aguiar *et al.*, "Big Data, IoT, ... Buzz Words for Academia or Reality for Industry?," *Proc. 21st Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '15*, pp. 550–551, 2015.

[4] P. P. Ray, T. Pflanzner, and A. Kertesz, "A survey of IoT cloud providers," *Futur. Comput. Informatics J.*, vol. 1, no. 1–2, pp. 730–735, 2017.

[5] C. Standards and C. Council, "Deploying Big Data Analytics Applications to the Cloud : Roadmap for Success," *Cloud Stand. Cust. Counc.*, p. 21, 2014.

[6] J. G. Koomey, "Worldwide electricity used in data centers," *Environ. Res. Lett.*, vol. 3, no. 3, 2008.

[7] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in Data Centre Energy Consumption under the European Code of Conduct for Data Centre Energy Efficiency," *Energies*, vol. 10, no. 10, 2017.

[8] T. Blascheck, M. Burch, M. Raschke, and D. Weiskopf, "Challenges and Perspectives in Big Eye-Movement Data Visual Analytics," 2015.

[9] Hortonworks, "How Big Data in Aviation Is Transforming the Industry," 2018. [Online]. Available: https://hortonworks.com/article/how-big-data-in-aviation-is-transforming-the-industry/. [Accessed: 27-Aug-2018].

[10] Inmarsat Aviation, "How big data drives safety," 2018.

[11] T.-C. DAO, "Real-time Eye tracking Analysis from Large scale Dataset using Cloud computing," School of Computing, University of Eastern Finland, 2014.

[12] N. Chaichi, J. Lavoie, S. Zarrin, R. Khalifa, and F. Sie, "A comprehensive assessment of cloud computing for smart grid applications: A multi-perspectives framework," in *Portland International Conference on Management of Engineering and Technology*, 2015, vol. 2015–Septe, pp. 2541–2547.

[13] Gordon E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, 1965.

[14] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, *A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems*, vol. 82. 2011.

[15] M. Dayarathna, Y. Wen, and R. Fan, "Data Center Energy Consumption Modeling: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[16] C. Zhu, V. C. M. Leung, L. Shu, and E. C.-H. Ngai, "Green Internet of Things for Smart World," *IEEE Access*, vol. 3, pp. 2151–2162, 2015.

[17] M. Sami, M. Haggag, and D. Salem, "Resource Allocation and Server Consolidation Algorithms for Green Computing," *Int. J. Sci. Eng. Res.*, vol. 6, no. 12, pp. 313–316, 2015.

[18] R. Buyya, K. Ramamohanarao, C. Leckie, R. N. Calheiros, A. V. Dastjerdi, and S. Versteeg, "Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions," *Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS*, vol. 2016–Janua, pp. 75–84, 2016.

[19] L. E. Bautista Villalpando, A. April, and A. Abran, "Performance analysis model for big data applications in cloud computing," *J. Cloud Comput.*, vol. 3, no. 1, pp. 19–38, 2014.

[20] N. T. Hieu, *Virtual Machine Management for Efficient Cloud Data Centers with Applications to Big Data Analytics*. .

[21] I. Caglar and D. T. Altilar, "An energy efficient VM allocation approach for data centers," *Proc. - 2nd IEEE Int. Conf. Big Data Secur. Cloud, IEEE BigDataSecurity 2016, 2nd IEEE Int. Conf. High Perform. Smart Comput. IEEE HPSC 2016 IEEE Int. Conf. Intell. Data S*, pp. 240–244, 2016.

[22] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, 2012.

[23] A. Beloglazov and R. Buyya, "OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds."

[24] A. Botta, W. De Donato, V. Persico, and A. Pescape, "On the integration of cloud computing and internet of things," *Proc. - 2014 Int. Conf. Futur. Internet Things Cloud, FiCloud 2014*.

[25] S. K. Routray and K. P. Sharmila, "Green initiatives in IoT," in *Proceedings of the 3rd IEEE International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics, AEEICB 2017*, 2017, pp. 454–457.

[26] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15667–15681, 2017.

[27] L. Hou *et al.*, "Internet of Things Cloud: Architecture and Implementation," Sep. 2016.

[28] R. Jimenez-peris, "Big Data and Cloud Challenges from IoT."

[29] M. Travers, "CPU Power Consumption Experiments and Results Analysis of Intel i7-4820K."

[30] X. Fan Wolf-Dietrich Weber Luiz André Barroso, "Power Provisioning for a Warehouse-sized Computer," 2007.

[31] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *Computer (Long. Beach. Calif).*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[32] W. Dargie, "A Stochastic Model for Estimating the Power Consumption of a Processor," *IEEE Trans. Comput.*, vol. 64, no. 5, pp. 1311–1322, May 2015.

[33] S. Daud *et al.*, "The effects of CPU load & idle state on embedded processor energy usage," in *2014 2nd International Conference on Electronic Design (ICED)*, 2014, pp. 30–35.

[34] F. Cao and M. M. Zhu, "Energy efficient workflow job scheduling for green cloud," *Proc. - IEEE 27th Int. Parallel Distrib. Process. Symp. Work. PhD Forum, IPDPSW 2013*, pp. 2218–2221, 2013.

[35] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology."

[36] I. S. Moreno and J. Xu, "Customer-aware resource overallocation to improve energy efficiency in realtime Cloud Computing data centers," *Proc. - 2011 IEEE Int. Conf. Serv. Comput. Appl. SOCA 2011*, p. 8, 2011.

[37] E. Masanet and H. Robert, "Chapter 20 : Data Center IT Efficiency Measures," *Unif. Methods Proj. Methods Determ. Energy Effic. Savings Specif. Meas.*, no. January 2015, p. 33, 2015.

[38] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," in *Concurrency Computation Practice and Experience*, 2012, vol. 24, no. 13, pp. 1397–1420.

[39] F. Wuhib, R. Stadler, and H. Lindgren, “Dynamic resource allocation with management objectives - Implementation for an OpenStack cloud,” *2012 8Th Int. Conf. Netw. Serv. Manag. 2012 Work. Syst. Virtualiztion Manag.*, pp. 309–315, 2012.

[40] A. Beloglazov and R. Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers,” *2010 10th IEEE/ACM Int. Conf. Clust. Cloud Grid Comput.*, pp. 826–831, 2010.

[41] V. Awasare and S. Deshmukh, “Survey and Comparative Study on Resource Allocation Strategies in Cloud Computing Environment,” *IOSR J. Comput. Eng.*, vol. 16, no. 2, pp. 94–101, 2014.

[42] S. Singh and I. Chana, “QoS-Aware Autonomic Resource Management in Cloud Computing,” *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1–46, Dec. 2015.

[43] S. M. Parikh, N. M. Patel, and H. B. Prajapati, “Resource Management in Cloud Computing: Classification and Taxonomy,” Feb. 2017.

[44] G. E. Gonçalves *et al.*, “5 Resource Allocation in Clouds: Concepts, Tools and Research Challenges.”

[45] M. A. Khoshkholghi, M. N. Derahman, A. Abdullah, S. Subramaniam, and M. Othman, “Energy-Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers,” *IEEE Access*, vol. 5, pp. 10709–10722, 2017.

[46] A. Beloglazov, “Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing,” no. February, pp. 1--232, 2013.

[47] A. Ashraf, B. Byholm, and I. Porres, “Distributed Virtual Machine Consolidation: A Systematic Mapping Study,” 2016.

[48] Z. Zhou, Z. Hu, J. Yu, J. Abawajy, and M. Chowdhury, “Energy-efficient virtual machine consolidation algorithm in cloud data centers,” *J. Cent. South Univ.*, vol. 24, no. 10, 2017.

[49] N. T. Hieu, M. Di Francesco, and A. Yla-Jaaski, “Virtual Machine Consolidation with Usage Prediction for Energy-Efficient Cloud Data Centers,” *Proc. - 2015 IEEE 8th Int. Conf. Cloud Comput. CLOUD 2015*, vol. XX, no. X, pp. 750–757, 2015.

[50] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb, “Virtual machine consolidation enhancement using hybrid regression algorithms Virtual machine consolidation enhancement,” *Egypt. Informatics J.*, vol. 18, no. 3, pp. 161–170, 2017.

[51] Nguyen Trung Hieu, M. Di Francesco, and A. Yla-Jaaski, “A virtual machine placement algorithm for balanced resource utilization in cloud data centers,” in *2014 IEEE 7th International Conference on Cloud Computing*, 2014, pp. 474–481.

[52] A. Najari, S. EnayatOllah Alavi, and M. Reza Noorimehr, “Optimization of Dynamic Virtual Machine Consolidation in Cloud Computing Data Centers,” *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 9, 2016.

[53] G. Han, W. Que, G. Jia, and L. Shu, “An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing.,” *Sensors (Basel).*, vol. 16, no. 2, p. 246, Feb. 2016.

[54] A. Corradi, M. Fanelli, and L. Foschini, “VM consolidation: A real case based on OpenStack Cloud,” *Futur. Gener. Comput. Syst.*, vol. 32, no. 1, pp. 118–127, 2014.

[55] W. Dai, L. Qiu, A. Wu, and M. Qiu, “Cloud Infrastructure Resource Allocation for Big Data Applications,” *IEEE Trans. Big Data*, vol. 7790, no. c, pp. 1–1, 2016.

[56] A. Beloglazov and R. Buyya, “OpenStack Neat: A framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds,” *Concurr. Comput.* , vol. 27, no. 5, pp. 1310–1333, 2015.

[57] “Research methodology : an introduction / Wayne Goddard and Stuart Melville - Details - Trove.” [Online]. Available: https://trove.nla.gov.au/work/11191016. [Accessed: 18-Apr-2018].

[58] J. R. Sampey, “The Romance of Research (Redman, L. V.; Mory, A. V. H.),” *J. Chem. Educ.*, vol. 11, no. 1, p. 62, Jan. 1934.

[59] O. Litvinski and A. Gherbi, “Experimental evaluation of OpenStack compute scheduler,” *Procedia Comput. Sci.*, vol. 19, no. Ant, pp. 116–123, 2013.

[60] A. Beloglazov, S. F. Piraghaj, and M. Alrokayan, “Deploying OpenStack on CentOS Using the KVM Hypervisor and GlusterFS Distributed File System ∗,” pp. 1–47, 2012.

[61] M. A. Lopez, A. G. P. Lobato, and O. C. M. B. Duarte, “A performance comparison of open-source stream processing platforms,” *2016 IEEE Glob. Commun. Conf. GLOBECOM 2016 - Proc.*, 2016.

[62] M. Zaharia *et al.*, “Apache Spark,” *Commun. ACM*, vol. 59, no. 11, pp. 56–65, Oct. 2016.

[63] Openstack Community, “OpenStack Docs: Live-migrate instances.” [Online]. Available: https://docs.openstack.org/nova/pike/admin/live-migration-usage.html.

[64] M. Popa and T. Slavici, “Embedded server with Wake on LAN function,” in *IEEE EUROCON 2009*.

[65] J. Zhan, “CHARACTERIZATION AND ARCHITECTURAL IMPLICATIONS OF BIG DATA WORKLOADS,” 2015.

[66] Openstack Community, “OpenStack Docs: Overcommitting CPU and RAM.” [Online]. Available: https://docs.openstack.org/arch-design/design-compute/design-compute-overcommit.html. [Accessed: 23-May-2018].

[67] L. Chen, S. Patel, H. Shen, and Z. Zhou, “Profiling and Understanding Virtualization Overhead in Cloud,” in *2015 44th International Conference on Parallel Processing*, 2015, pp. 31–40.

[68] R. Nadimi and R. Ahmadi Khatir, “VM Consolidation by using Selection and Placement of VMs in Cloud Datacenters,” *Casp. J. Math. Sci. Univ. Maz. Iran CJMS*, vol. 6, no. 2, pp. 120–130, 2017.

[69] J Porras, A Seffah, E Rondeau, K Andersson, A Klimova, " PERCCOM: A Master Program in Pervasive Computing and COMmunications for Sustainable Development " 29th IEEE Conference on Software Engineering Education and Training, 2016

[70] L. M. Hilty and M. D. Hercheui, “ICT and Sustainable Development,” vol. AICT-328, Springer, 2010, pp. 227–235.

[71] C. Becker *et al.*, “Requirements: The key to sustainability,” *IEEE Softw.*, vol. 33, no. 1, 2016.

[72] WECouncil, “World Energy Trilemma Index | 2017 MONITORING THE SUSTAINABILITY OF NATIONAL ENERGY SYSTEMS.”

[73] “Total number of hyperscale data centers reaches 300,” *DatacenterDynamics*, 2018. [Online].

[74] “The Top Ten Best Countries in the World for Locating Data Centers.”

[75] A. M. Brander *et al.*, “Electricity-specific emission factors for grid electricity,” *Ecometrica*, no. August, pp. 1–22, 2011.

APPENDIX 1. IoT System Set-up

Tasks:

Gaze Patterns Recognition to interpret Cognitive Behavior is the main idea of the Doctoral work. The user performs 15 different tasks of flying an aircraft in the Flight Simulator. The user wears an Eye Tracker during the whole process.

Equipment Set Up:

Flight Simulator - "Qalhata_2016" - Multi-Screen Capable Flight Sim

1 x Intel Core i7 6700K Quad Core (4.0GHz, 8MB Cache, overclocked 4.4GHz+, Hyperthreading)
1 x Corsair Hydro Series H80 High Performance Liquid CPU Cooler
Tuning (BIOS, Driver and OS Tweaks)
1 x NVidia GeForce GTX 970 4GB GDDR5 PCI Express Graphics Card
1 x 16GB Standard DDR4 2400MHz Dual Channel Memory Kit (2 x 8GB)
1 x Asus Z170-P Intel Motherboard (4 Phase Power)
1 x Samsung 250GB 850 Evo Series SATA III 6Gb/s Solid-State Drive
1 x Seagate 1TB Barracuda 7200 64MB Cache SATA III Hard Disk Drive
1 x FSP Non-Modular 750W Power Supply (Silver 80 Plus Certified)
1 x Onboard HD 7.1 Audio

Flight Controls

1 x Saitek X52 Flight Control System
1 x Saitek Pro Flight Yoke System
1 x Saitek Pro Flight Rudder Pedals
1 x Saitek Pro Flight Multi Panel (Autopilot/Auto Throttle/Flaps/Trim)
1 x Saitek Pro Flight Radio Panel (COM1/COM2/NAV1/NAV2)
1 x Saitek Pro Flight Switch Panel (Engines/Gear/Lights)
1 x Natural Point TrackIR 5 with Track Clip Pro (To be added)

Eye Tracker

Pupil Labs Eye tracking kit

FoV Monitors

3 x 27" AOC HD Monitors

Operating System

1 x Microsoft Windows 10 Home (64-bit) – Running X-Plane 10 & 11, Tacview Analysis Tool and the Pupil Labs Capture and Playback software tools

Flight Sim Chassis

1x Pagnian Flight Simulator Chassis and Frame – Extendable To fit a motion platform

The Experiment:

The Eye tracker captures the reflection of the eye. It initially calibrates the position of eyes for each user with 9 distinct points. The position of the pupil is identified based on the reflection from the infrared illumination. Multiple parameters regarding these distinct points, pupil dilation, size and position of pupil are all recorded. The eye tracker also captures a video of eye movements during the tasks.

(continues)

During the 15 flying tasks, the user controls the aircraft using flight controls. His performance during the tasks are recorded by the flight simulator based on his actions on the actuators. Data such as take of speed, landing speed, angle of inclination are a few of the factors which determine the performance of the user. Unexpected startle scenarios are created during the tasks to identify the cognitive behavior. 'Startle' is experiencing an unexpected or not experiencing the expected such as disturbances - turbulence or change in weather conditions. Fixation is a parameter from the Eye tracker device which determines the number of times a user looks at a particular object. Since an aircraft is equipped with numerous devices and components in the dashboard, it is important for the pilot to understand which component needs attention during a startle scenario. An experienced pilot looks at the device which is important at the situation and the cognitive behavior is fast because of experience. A Novice will have more fixation counts than an expert. Vision Indicator Average is a measure of average of take-off speed score and climb speed score of the user. This measure helps in relating vision with cognition during a flight. Fixation data helps in understanding the Visual Behavior along with the performance data from the Flight Simulator. The outcome of this work can help in classifying different kinds of pilots and train them based on their strengths and weaknesses to improve their flying performance. This will greatly impact reducing aviation accidents due to Loss of Control.

Data Sources:

In thie scenario, there are multiple data sources.

- Flight Simulator : Task information, Startle scenarios, Performance Data
- Eye Tracker : Fixation Data, Pupil Positions
- Flight Controls: Brakes, Throttle, Acceleration etc.,

Data Processing:

In order to make sure the truthfulness or veracity of the data, a reliability analysis is needed. This process of removing noise by performing a summary analysis by selecting random points from a big sample of data of widely differing values is called bootstrapping or data massaging. An Empirical Cumulative Distribution Function (ECDF) is used for data massaging. A series of Statistical analysis is performed which includes, correlation of different parameters such as fixation counts and performance of the user. A Poisson distribution computes the probability of a given number of events occurring in a fixed interval of time or space, to analyze the cognitive behavior. The Statistical analysis is run on different parameters to find the relationship between Fixation & Cognition and to classify the user based on his performance.

	norm_pos_x	norm_pos_y	dispersion	confidence	method	gaze_point_3d_x
0	0.418033	0.682182	0.783270	1.000000	pupil	-70.599354
1	0.418436	0.715165	0.958956	1.000000	pupil	-71.751534
2	0.387143	0.587924	0.919865	0.999773	pupil	-89.720900
3	0.427755	0.716043	0.949755	1.000000	pupil	-64.321441
4	0.461076	0.716780	0.737886	1.000000	pupil	-38.816434
5	0.457116	0.712485	0.957530	1.000000	pupil	-41.665021
6	0.425112	0.612741	0.798015	1.000000	pupil	-62.173130
7	0.414723	0.680539	0.755507	1.000000	pupil	-73.031288
8	0.381604	0.685340	0.959929	1.000000	pupil	-100.044507
9	0.373617	0.651517	0.998707	1.000000	pupil	-104.468371
10	0.368645	0.656394	0.977853	1.000000	pupil	-108.906418
11	0.365828	0.598492	0.982291	1.000000	pupil	-107.226137
12	0.455839	0.616038	0.795401	1.000000	pupil	-40.055183
13	0.380415	0.662111	0.835342	1.000000	pupil	-99.567626
14	0.381367	0.644499	0.971332	1.000000	pupil	-97.693569
15	0.442738	0.591758	0.925602	1.000000	pupil	-48.625070
16	0.382838	0.612389	0.998410	0.999841	pupil	-94.455595
17	0.412757	0.683172	0.950117	1.000000	pupil	-74.634113
18	0.390239	0.653580	0.950103	0.999204	pupil	-91.006391
19	0.421420	0.623810	0.987744	1.000000	pupil	-65.310958
20	0.427325	0.638338	0.342577	1.000000	pupil	-61.478469
21	0.425016	0.714017	0.918028	1.000000	pupil	-66.333512
22	0.383054	0.687808	0.725834	1.000000	pupil	-99.245685
23	0.446778	0.682735	0.896022	1.000000	pupil	-48.700166

Data from eye tracker device

(continues)

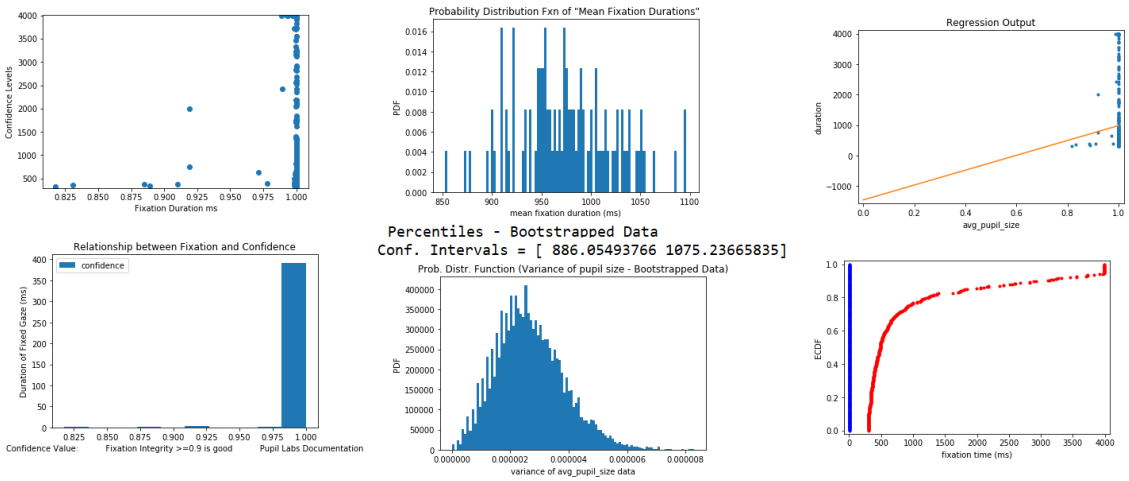
REST API to fetch data - GET

http://192.168.1.10:3000/inflightdata/

Sample Output:


```
[
    {
        "id": 1,
        "start_timestamp": 3617.2559,
        "duration": 352.8000000001157,
        "start_frame_index": 5,
        "end_frame_index": 16,
        "norm_pos_x": 0.4180327103552838,
        "norm_pos_y": 0.6821823808521131,
        "dispersion": 0.783269508688,
        "confidence": 1,
        "method": "pupil",
        "gaze_point_3d_x": -70.59935355869726,
        "gaze_point_3d_y": -84.51290920574947,
        "gaze_point_3d_z": 456.9785679734375,
        "data_points": "3617.2559 3617.2643 3617.2727 3617.2811 3617.2895
3617.2979 3617.3063 3617.3147 3617.3231 3617.3315 3617.3399 3617.3483
3617.3567 3617.3651 3617.3735 3617.3819 3617.3903 3617.3987 3617.4071
3617.4155 3617.4239 3617.4323 3617.4407 3617.4491 3617.4575 3617.4659
3617.4743 3617.4827 3617.4911 3617.4995 3617.5079 3617.5163 3617.5247
3617.5331 3617.5415 3617.5499 3617.5583 3617.5667 3617.5751 3617.5835
3617.5919 3617.6003 3617.6087"
    }
]
```

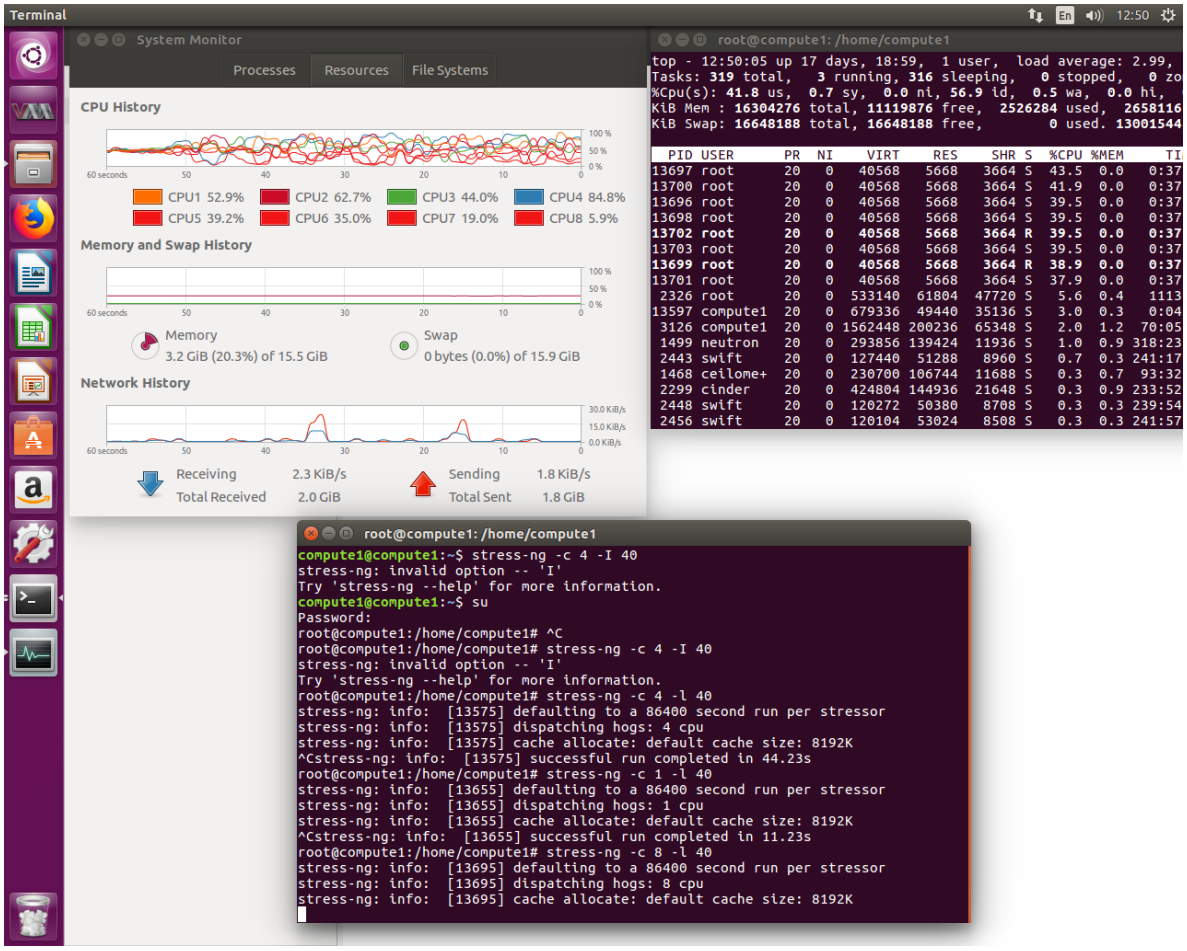

Visualized data:



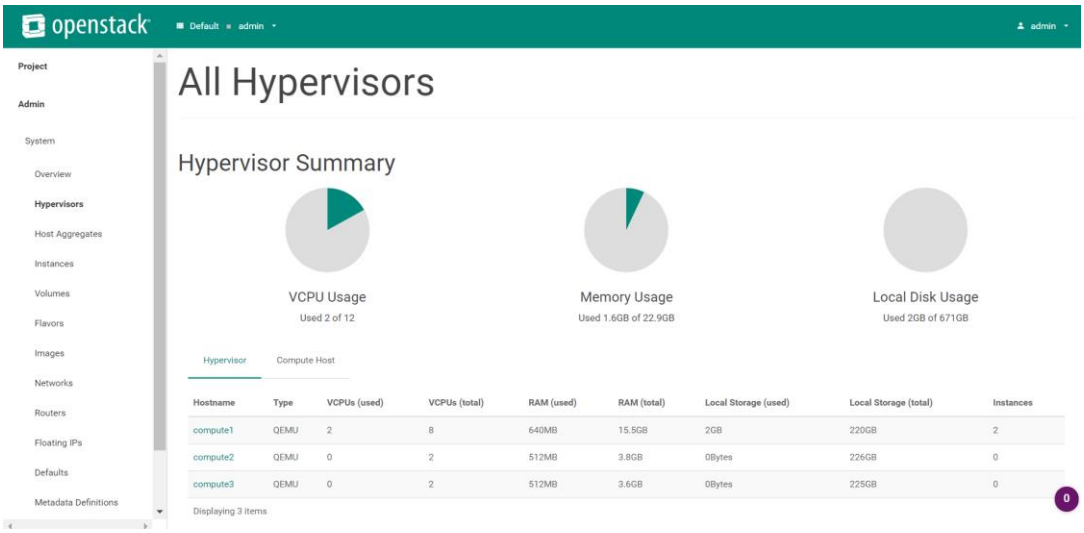
Visual output of IoT data analysis

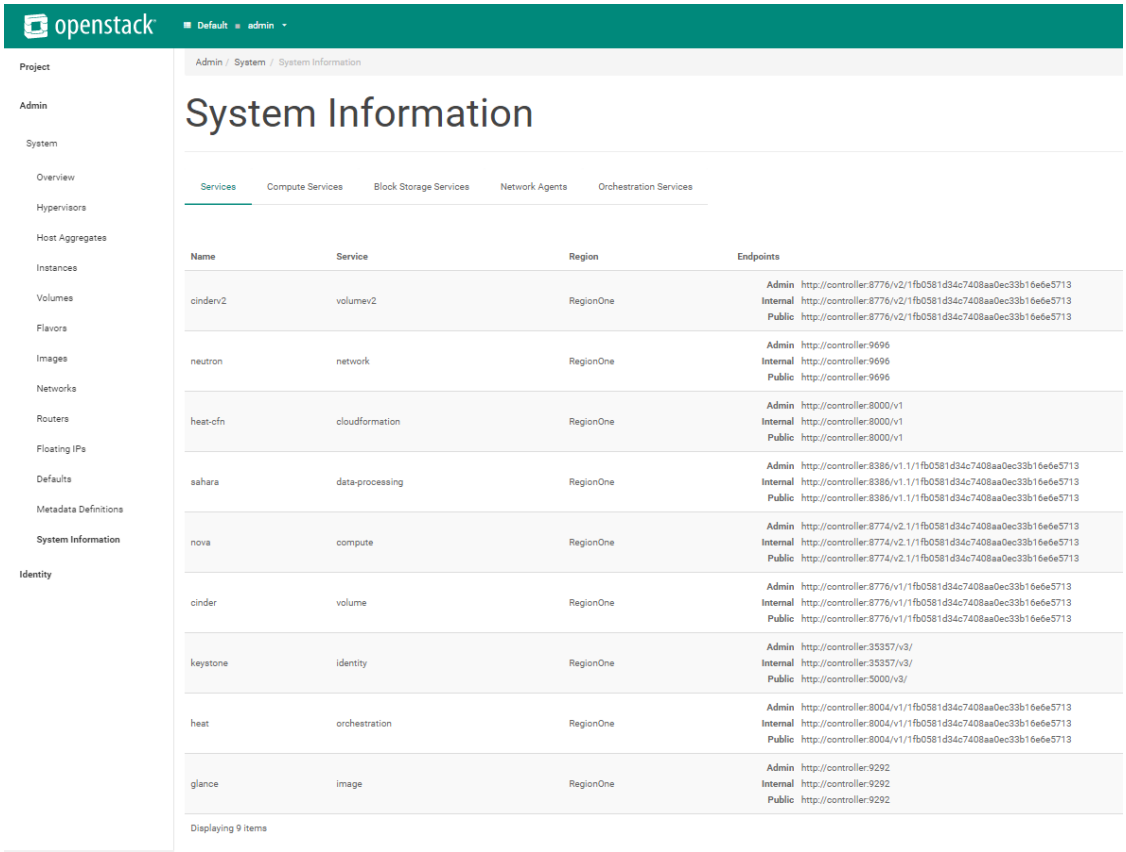
APPENDIX 2. Synthetic Workload using stress-ng

Stress-ng is a Linux utility to create stress situations on a server's resources. Individual cores of the CPU can be stressed by a simple command that is to calculate square root many times over.

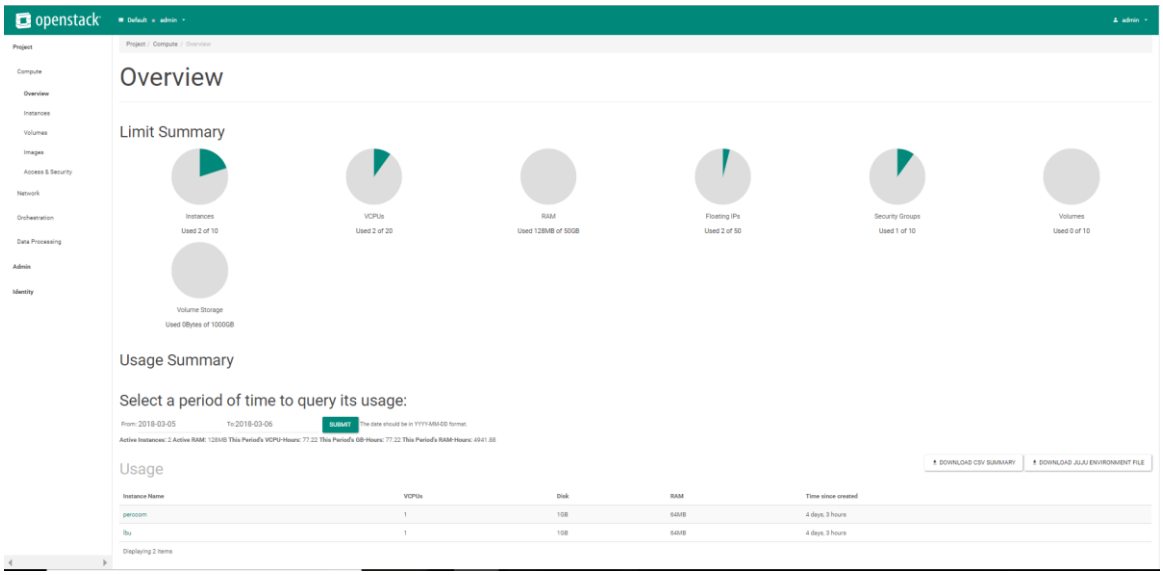


APPENDIX 3. Openstack Dashboard Screenshots





(continues)



Flavors

Filter

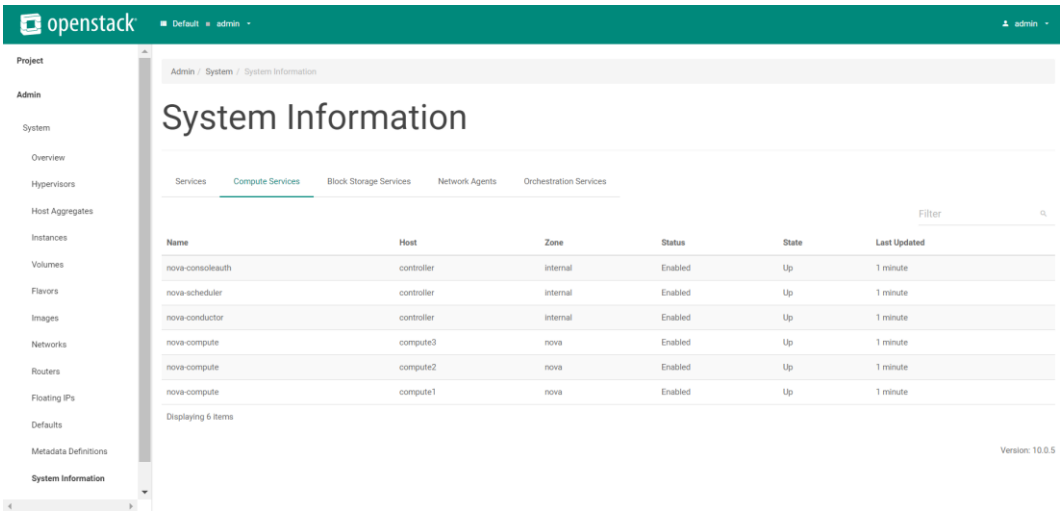
+ CREATE FLAVOR

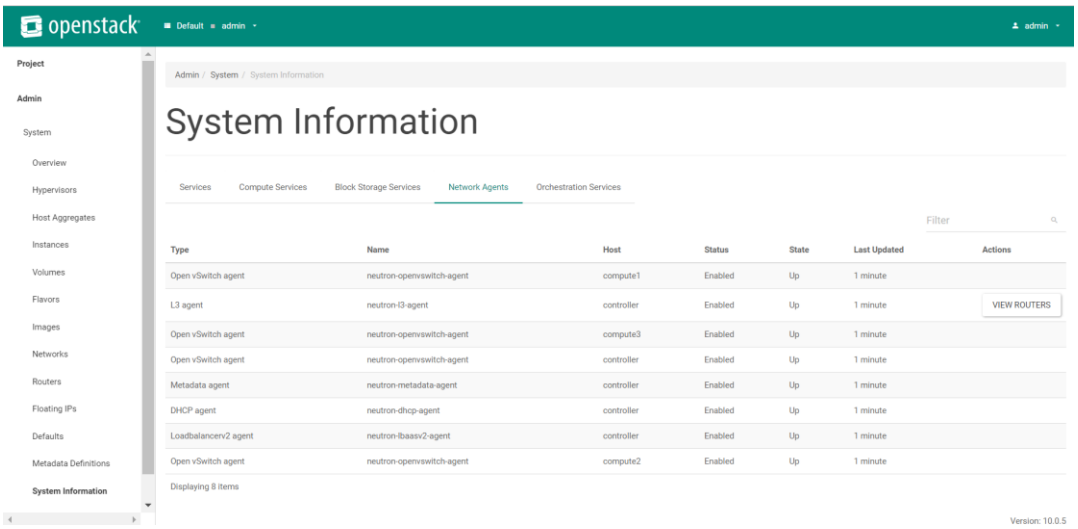
DELETE FLAVORS

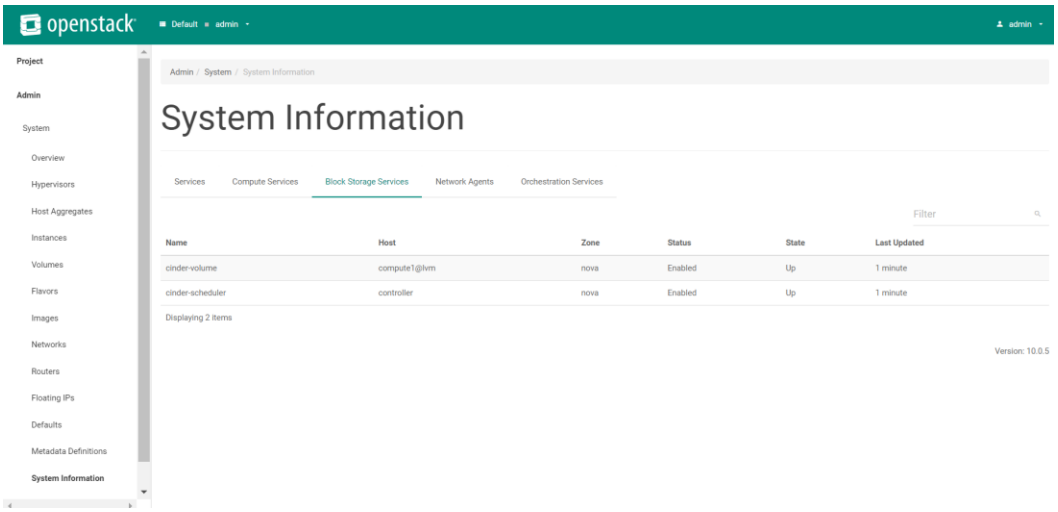
Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
m1.nano	1	64MB	1GB	0GB	0MB	1.0	0	Yes	No	EDIT FLAVOR
m1.spark-ubuntu	1	256MB	2GB	0GB	0MB	1.0	4140f606-d74b-43ff-815f-8b1b13e3455e	Yes	No	EDIT FLAVOR
m1.tiny	1	512MB	1GB	0GB	0MB	1.0	98de8c6a-f1bb-481a-b219-518ffeeaa28d	Yes	No	EDIT FLAVOR
m1.ubuntu1	1	512MB	5GB	0GB	0MB	1.0	265b893c-8b56-4d2b-bad7-40937048cb63	Yes	No	EDIT FLAVOR
m1.ubuntu2	2	1GB	10GB	0GB	0MB	1.0	b1594dcd-b9c9-4f50-bd92-88a615da2029	Yes	No	EDIT FLAVOR
m1.ubuntu3	3	2GB	15GB	0GB	0MB	1.0	87b389c6-a158-4002-bec9-275e846811cb	Yes	No	EDIT FLAVOR

Displaying 6 items

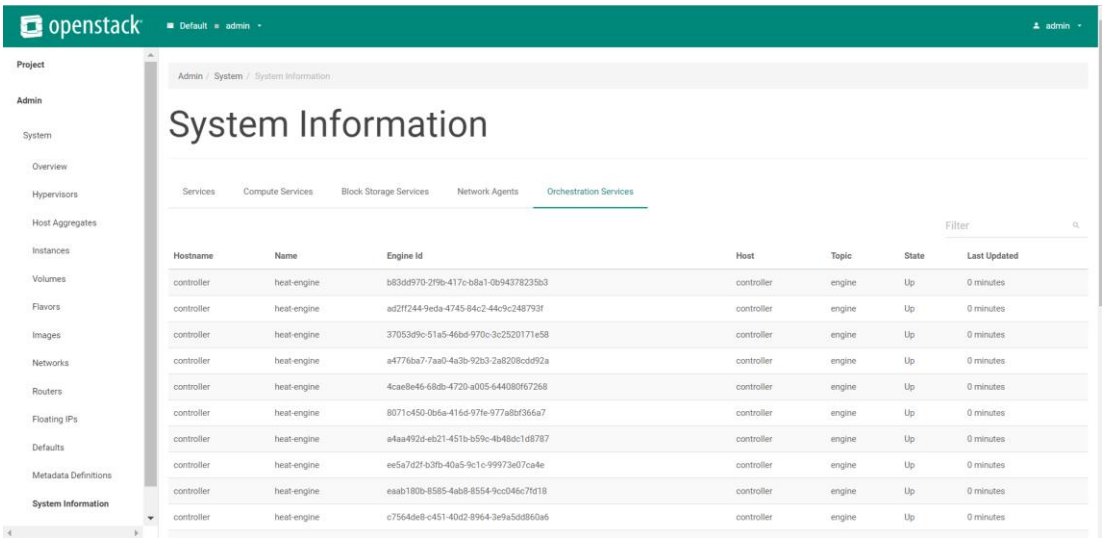
(continues)

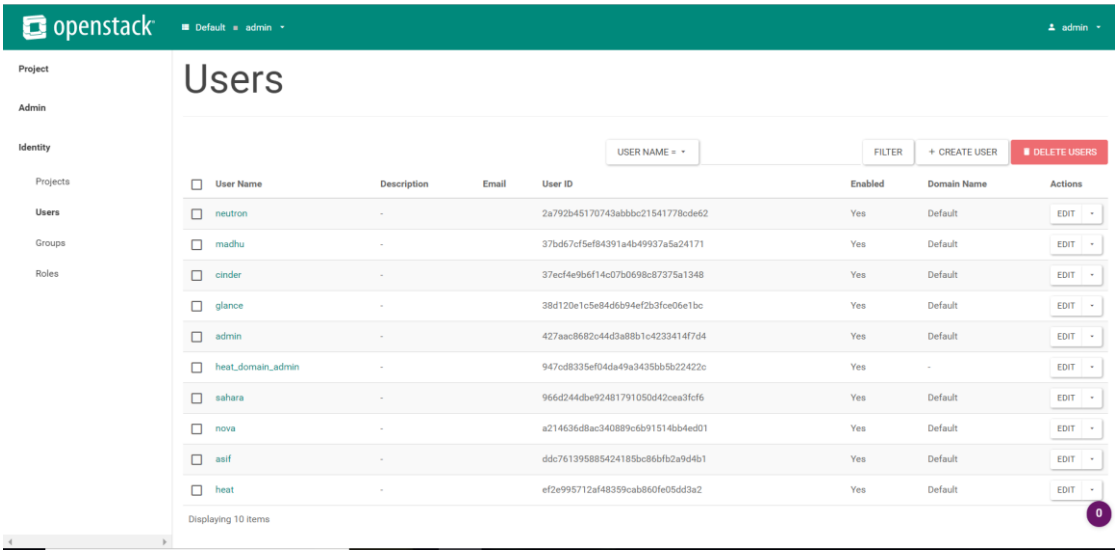

openstack
Default
admin
Project
Admin
System
Overview
Hypervisors
Host Aggregates
Instances
Volumes
Flavors
Images
Networks
Routers
Floating IPs
Defaults
Metadata Definitions
System Information
Admin / System / System Information
System Information
Services
Compute Services
Block Storage Services
Network Agents
Orchestration Services
Filter
Name Host Zone Status State Last Updated
nova-consoleauth controller internal Enabled Up 1 minute
nova-scheduler controller internal Enabled Up 1 minute
nova-conductor controller internal Enabled Up 1 minute
nova-compute compute3 nova Enabled Up 1 minute
nova-compute compute2 nova Enabled Up 1 minute
nova-compute compute1 nova Enabled Up 1 minute
Displaying 6 items
Version: 10.0.5

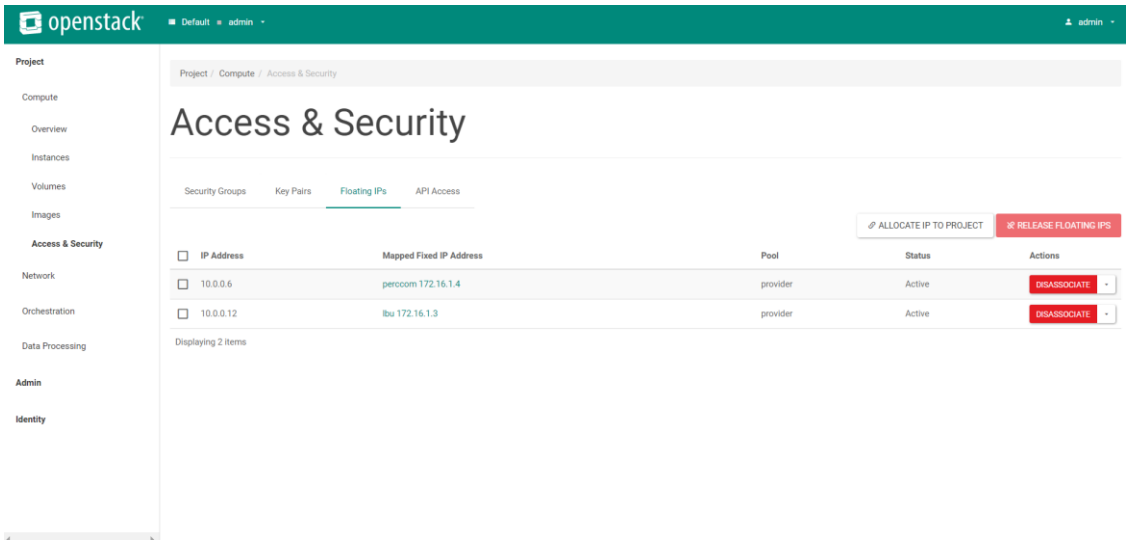

openstack
Default
admin
Project
Admin
System
Overview
Hypervisors
Host Aggregates
Instances
Volumes
Flavors
Images
Networks
Routers
Floating IPs
Defaults
Metadata Definitions
System Information
Admin / System / System Information
System Information
Services
Compute Services
Block Storage Services
Network Agents
Orchestration Services
Filter
Type Name Host Status State Last Updated Actions
Open vSwitch agent neutron-openvswitch-agent compute1 Enabled Up 1 minute
L3 agent neutron-l3-agent controller Enabled Up 1 minute VIEW ROUTERS
Open vSwitch agent neutron-openvswitch-agent compute3 Enabled Up 1 minute
Open vSwitch agent neutron-openvswitch-agent controller Enabled Up 1 minute
Metadata agent neutron-metadata-agent controller Enabled Up 1 minute
DHCP agent neutron-dhcp-agent controller Enabled Up 1 minute
Loadbalancerv2 agent neutron-lbaasv2-agent controller Enabled Up 1 minute
Open vSwitch agent neutron-openvswitch-agent compute2 Enabled Up 1 minute
Displaying 8 items
Version: 10.0.5


openstack
Default
admin
Project
Admin
System
Overview
Hypervisors
Host Aggregates
Instances
Volumes
Flavors
Images
Networks
Routers
Floating IPs
Defaults
Metadata Definitions
System Information
Admin / System / System Information
System Information
Services
Compute Services
Block Storage Services
Network Agents
Orchestration Services
Filter
Name Host Zone Status State Last Updated
cinder-volume compute1@lvm nova Enabled Up 1 minute
cinder-scheduler controller nova Enabled Up 1 minute
Displaying 2 items
Version: 10.0.5

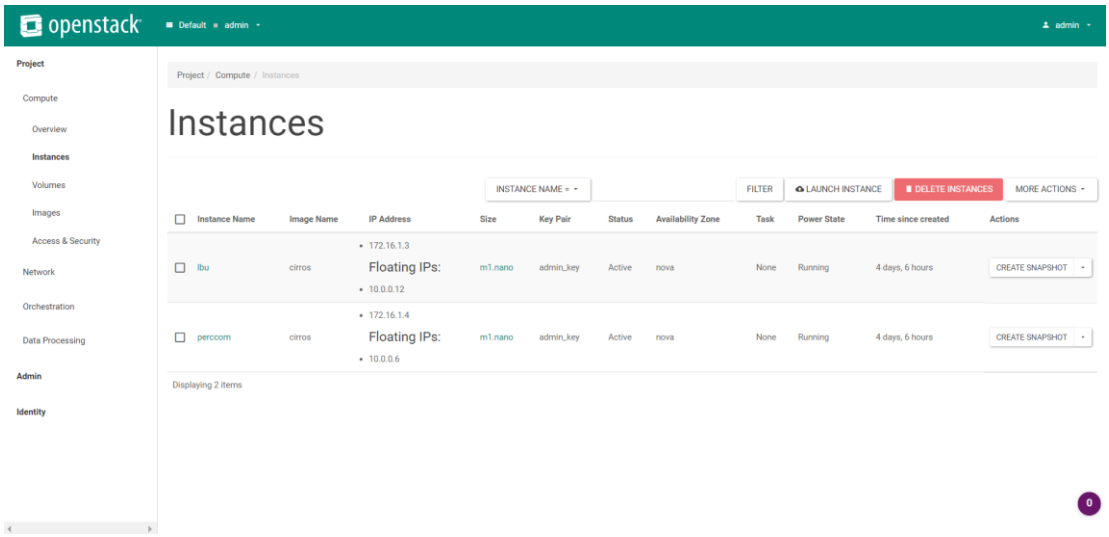
(continues)

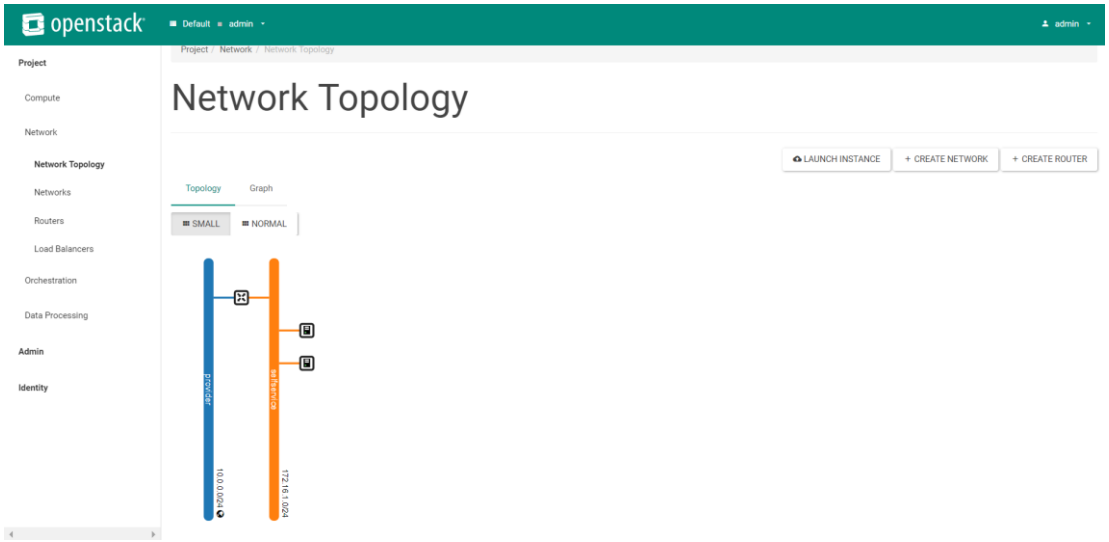

openstack
Default
admin
Project
Admin
System
Overview
Hypervisors
Host Aggregates
Instances
Volumes
Flavors
Images
Networks
Routers
Floating IPs
Defaults
Metadata Definitions
System Information
Admin / System / System Information
System Information
Services
Compute Services
Block Storage Services
Network Agents
Orchestration Services
Filter
Hostname | Name | Engine Id | Host | Topic | State | Last Updated
controller | heat-engine | b83dd970-2f9b-417c-b8a1-0b94378235b3 | controller | engine | Up | 0 minutes
controller | heat-engine | ad2ff244-9eda-4745-84c2-44c9c248793f | controller | engine | Up | 0 minutes
controller | heat-engine | 37053d9c-51a5-46bd-970c-3c2520171e58 | controller | engine | Up | 0 minutes
controller | heat-engine | a4776ba7-7aa0-4a3b-92b3-2a8208cdd92a | controller | engine | Up | 0 minutes
controller | heat-engine | 4cae8e46-68db-4720-a005-644080f67268 | controller | engine | Up | 0 minutes
controller | heat-engine | 8071c450-0b6a-416d-97fe-977a8bf366a7 | controller | engine | Up | 0 minutes
controller | heat-engine | a4aa492d-eb21-451b-b59c-4b48dc1d8787 | controller | engine | Up | 0 minutes
controller | heat-engine | ee5a7d2f-b3fb-40a5-9c1c-99973e07ca4e | controller | engine | Up | 0 minutes
controller | heat-engine | eaab180b-8585-4ab8-8554-9cc046c7fd18 | controller | engine | Up | 0 minutes
controller | heat-engine | c7564de8-c451-40d2-8964-3e9a5dd860a6 | controller | engine | Up | 0 minutes

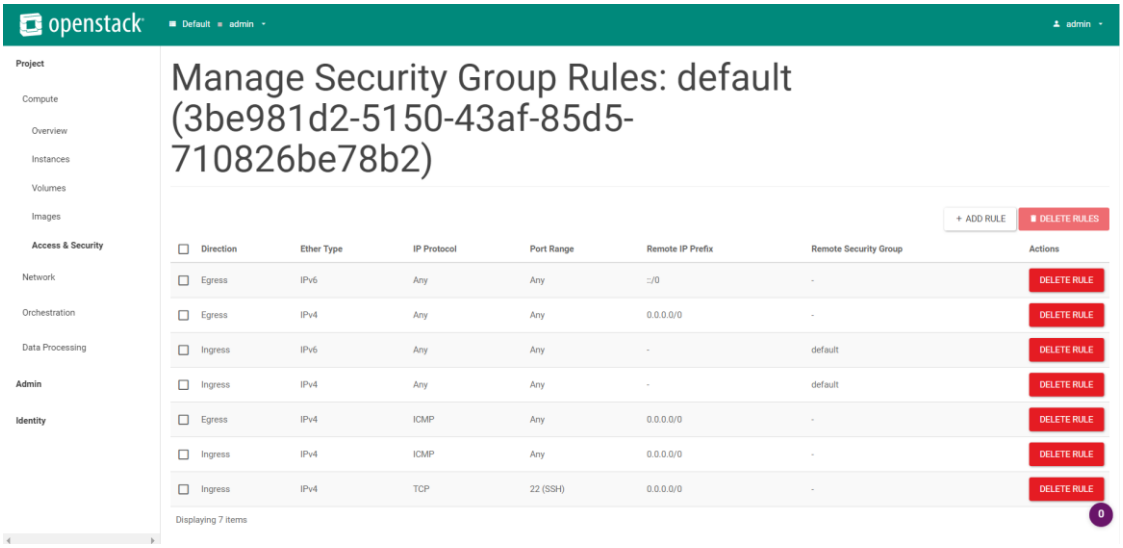

openstack
Default
admin
Project
Admin
Identity
Projects
Users
Groups
Roles
Users
USER NAME =
FILTER
+ CREATE USER
DELETE USERS
User Name | Description | Email | User ID | Enabled | Domain Name | Actions
neutron | - | 2a792b45170743abbbc21541778cde62 | Yes | Default | EDIT
madhu | - | 37bd67cf5ef84391a4b49937a5a24171 | Yes | Default | EDIT
cinder | - | 37ecf4e9b6f14c07b0698c87375a1348 | Yes | Default | EDIT
glance | - | 38d120e1c5e84d6b94ef2b3fce06e1bc | Yes | Default | EDIT
admin | - | 427aac8682c44d3a88b1c4233414f7d4 | Yes | Default | EDIT
heat_domain_admin | - | 947cd8335ef04da49a3435bb5b22422c | Yes | - | EDIT
sahara | - | 966d244dbe92481791050d42cea3fcf6 | Yes | Default | EDIT
nova | - | a214636d8ac340889c6b91514bb4ed01 | Yes | Default | EDIT
asif | - | ddc761395885424185bc86bfb2a9d4b1 | Yes | Default | EDIT
heat | - | ef2e995712af48359cab860fe05dd3a2 | Yes | Default | EDIT
Displaying 10 items


openstack
Default
admin
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Data Processing
Admin
Identity
Project / Compute / Access & Security
Access & Security
Security Groups
Key Pairs
Floating IPs
API Access
ALLOCATE IP TO PROJECT
RELEASE FLOATING IPS
IP Address | Mapped Fixed IP Address | Pool | Status | Actions
10.0.0.6 | percoom 172.16.1.4 | provider | Active | DISASSOCIATE
10.0.0.12 | lbu 172.16.1.3 | provider | Active | DISASSOCIATE
Displaying 2 items

(continues)


openstack
Default
admin
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Data Processing
Admin
Identity
Project / Compute / Instances
Instances
INSTANCE NAME
FILTER
LAUNCH INSTANCE
DELETE INSTANCES
MORE ACTIONS
Instance Name
Image Name
IP Address
Size
Key Pair
Status
Availability Zone
Task
Power State
Time since created
Actions
lbu
cirros
172.16.1.3
Floating IPs:
10.0.0.12
m1.nano
admin_key
Active
nova
None
Running
4 days, 6 hours
CREATE SNAPSHOT
perccom
cirros
172.16.1.4
Floating IPs:
10.0.0.6
m1.nano
admin_key
Active
nova
None
Running
4 days, 6 hours
CREATE SNAPSHOT
Displaying 2 items

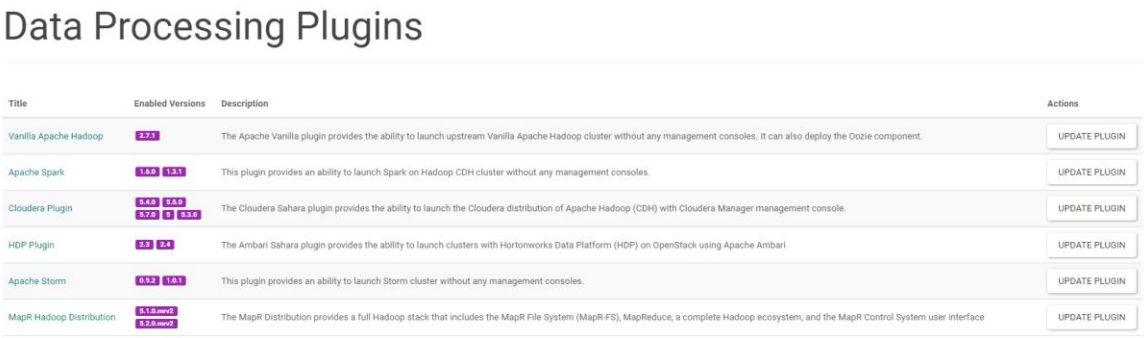

openstack
Default
admin
Project
Compute
Network
Network Topology
Networks
Routers
Load Balancers
Orchestration
Data Processing
Admin
Identity
Project / Network / Network Topology
Network Topology
LAUNCH INSTANCE
+ CREATE NETWORK
+ CREATE ROUTER
Topology
Graph
SMALL
NORMAL
provider
10.0.0.0/24
selfservice
172.16.1.0/24


openstack
Default
admin
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Data Processing
Admin
Identity
Manage Security Group Rules: default (3be981d2-5150-43af-85d5-710826be78b2)
+ ADD RULE
DELETE RULES
Direction
Ether Type
IP Protocol
Port Range
Remote IP Prefix
Remote Security Group
Actions
Egress IPv6 Any Any ::/0 - DELETE RULE
Egress IPv4 Any Any 0.0.0.0/0 - DELETE RULE
Ingress IPv6 Any Any - default DELETE RULE
Ingress IPv4 Any Any - default DELETE RULE
Egress IPv4 ICMP Any 0.0.0.0/0 - DELETE RULE
Ingress IPv4 ICMP Any 0.0.0.0/0 - DELETE RULE
Ingress IPv4 TCP 22 (SSH) 0.0.0.0/0 - DELETE RULE
Displaying 7 items

APPENDIX 4. Setting up Elastic Data Processing (EDP) on Sahara

Creating a cluster:

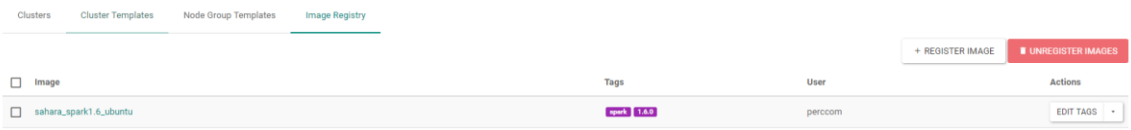
1. Choosing a plugin : In order to determins the type of cluster to run, there are several choices available based on the confirguration of Sahara. The 'choose plugin' option lists all the available data processing plugins. Choose a plugin along with the version number. Choosing this up front will allow the rest of the cluster creation steps to focus only on options that are pertinent to your desired cluster type.



Chosen plugin : Plugin: Apache Spark Version: 1.6.0

2. To launch instances of the cluster, the image needs to be registered. In the register image form, the image is chosen, username is entered tags are added for the chosen plugin.

OS	Flavor	vCPUs	RAM	Disk
sahara-mitaka-spark-1.6.0-ubuntu.qcow2	m1.spark-ubuntu	1	256 MB	2 GB



3. Next, the different types of virtual machines in the cluster is defined. This is done by defining a Node Group Template for each type of VM. Two templates, one for VMs running a "master" set of processes while another template for the set of VMs to be running the "worker" processes. The flavors of VMs and a sample node group template is presented below.

m1.tiny	1	512MB	1GB	0GB	0MB	1.0	98de8c6a-f1bb-481a-b219-518ffeeaa28d	Yes	No
m1.ubuntu1	1	512MB	5GB	0GB	0MB	1.0	265b893c-8b56-4d2b-bad7-40937048cb63	Yes	No
m1.ubuntu2	2	1GB	10GB	0GB	0MB	1.0	b1594dcd-b9c9-4f50-bd92-88a615da2029	Yes	No
m1.ubuntu3	3	2GB	15GB	0GB	0MB	1.0	87b389c6-a158-4002-bec9-275e846811cb	Yes	No

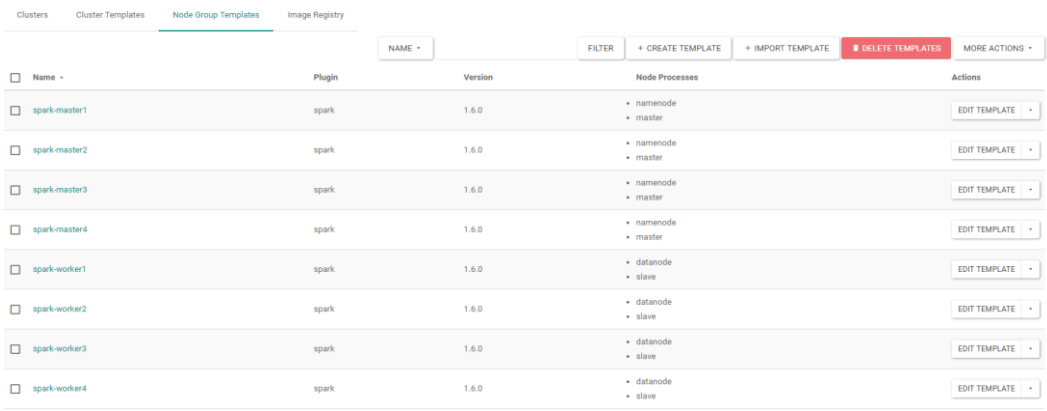
(continues)

Node Group Template in Json – Master & Worker

```
{
    "auto_security_group": true,
    "availability_zone": "",
    "description": "spark 1.6.0 master node group1",
    "flavor_id": "98de8c6a-f1bb-481a-b219-518ffeeaa28d",
    "hadoop_version": "1.6.0",
    "image_id": "d5de30b7-bff0-43bf-9d74-493a646d6606",
    "is_protected": false,
    "is_proxy_gateway": false,
    "is_public": true,
    "name": "spark-master1",
    "node_configs": {
        "HDFS": {}
    },
    "node_processes": [
        "namenode",
        "master"
    ],
    "plugin_name": "spark"
}

{
    "auto_security_group": true,
    "availability_zone": "",
    "description": "spark 1.6.0 worker node group1",
    "flavor_id": "98de8c6a-f1bb-481a-b219-518ffeeaa28d",
    "hadoop_version": "1.6.0",
    "image_id": "d5de30b7-bff0-43bf-9d74-493a646d6606",
    "is_protected": false,
    "is_proxy_gateway": false,
    "is_public": true,
    "name": "spark-worker1",
    "node_configs": {
        "HDFS": {}
    },
    "node_processes": [
        "namenode",
        "master"
    ],
    "plugin_name": "spark"
}
```

Screenshots of created Node Group Templates

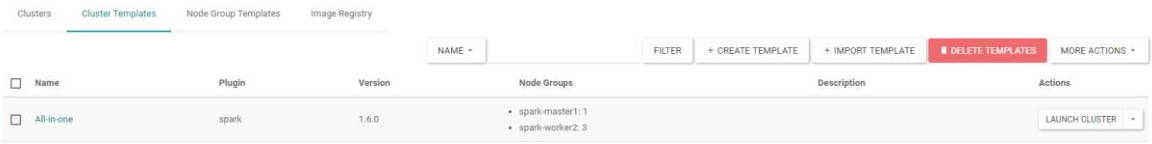


(continues)

```
+----------------------------+-------------------------------------+
| Property                   | Value                               |
+----------------------------+-------------------------------------+
| auto_security_group        | True                                |
| availability_zone          |                                     |
| created_at                 | 2018-04-03T11:16:44                 |
| description                | spark 1.6.0 master node group4      |
| flavor_id                  | 87b389c6-a158-4002-bec9-275e846811cb |
| floating_ip_pool           | None                                |
| hadoop_version             | 1.6.0                               |
| id                         | a184552a-e448-4aee-9aaf-7752f8b0b024 |
| image_id                   | d5de30b7-bff0-43bf-9d74-493a646d6606 |
| is_default                 | False                               |
| is_protected               | False                               |
| is_proxy_gateway           | False                               |
| is_public                  | True                                |
| name                       | spark-master4                       |
| node_configs               | {u'HDFS': {}}                       |
| node_processes             | namenode, master                    |
| plugin_name                | spark                               |
| security_groups            | None                                |
| tenant_id                  | 1fb0581d34c7408aa0ec33b16e6e5713    |
| use_autoconfig             | True                                |
| volume_local_to_instance   | False                               |
| volume_mount_prefix        | /volumes/disk                       |
| volume_type                | None                                |
| volumes_availability_zone  | None                                |
| volumes_per_node           | 0                                   |
| volumes_size               | 0                                   |
+----------------------------+-------------------------------------+
```

```
+----------------------------+-------------------------------------+
| Property                   | Value                               |
+----------------------------+-------------------------------------+
| auto_security_group        | True                                |
| availability_zone          |                                     |
| created_at                 | 2018-04-03T11:21:06                 |
| description                | spark 1.6.0 worker node group1      |
| flavor_id                  | 98de8c6a-f1bb-481a-b219-518ffeeaa28d |
| floating_ip_pool           | None                                |
| hadoop_version             | 1.6.0                               |
| id                         | a783960e-ac5e-4105-9984-38a23f1c4619 |
| image_id                   | d5de30b7-bff0-43bf-9d74-493a646d6606 |
| is_default                 | False                               |
| is_protected               | False                               |
| is_proxy_gateway           | False                               |
| is_public                  | True                                |
| name                       | spark-worker1                       |
| node_configs               | {u'HDFS': {}}                       |
| node_processes             | datanode, slave                     |
| plugin_name                | spark                               |
| security_groups            | None                                |
| tenant_id                  | 1fb0581d34c7408aa0ec33b16e6e5713    |
| use_autoconfig             | True                                |
| volume_local_to_instance   | False                               |
| volume_mount_prefix        | /volumes/disk                       |
| volume_type                | None                                |
| volumes_availability_zone  | None                                |
| volumes_per_node           | 0                                   |
| volumes_size               | 0                                   |
+----------------------------+-------------------------------------+
```

4. A cluster template is created by choosing the number of instances of each Node Group Template that will appear in the cluster.



5. Launching the cluster : A name for the cluster is given, the cluster template is chosen and the image to build instances is chosen to create the cluser. On clicking 'Create', the instances begin to spawn. The cluster is operational in a few minutes.