Energy-Efficient Resource Management Framework for Cloud Data Centers

Khulekani Sibiya

(Student Number: 21345559)

A thesis submitted to the Faculty of Engineering and the Built Environment in fulfillment of the requirements for the degree of

Master in Engineering (M. Eng)

May 2022

Approved for Final Submission

Supervisor: Professor B Nleya

Student: Khulekani Sibiya

_____ _ ___

Date_____

Date _30_May_2022_____

Declaration

As a result, I am declaring the dissertation to be my original work. It has not been submitted in part or in whole to any other university for a similar qualification.

I provide permission to the University to lend this work to anyone else for scholarly research reasons exclusively.

__ _ _30_May_2022_____

Signature Date

Khulekani_Sibiya_____ _21345559_____

Name Student number

Acknowledgment

I'd like to express my gratitude to God for all he's done for me. Also, thank you for your academic supervisor's unwavering support throughout the process. I would like to express my gratitude to my mother (Margaret Sibiya) for her unwavering support throughout my life. Thank you for being my everyday inspiration, Sinethemba Ndlovu and Aphilenhle Sibiya. My brothers Mduduzi Zondo, Nkululeko Sibiya, and Khayalethu Sibiya deserve special recognition for their contributions. Abongwe Zondo, Abusekho Zondo, and Nkuluthando Sibiya are three of my nieces, I'm doing this to show you that with God, everything is possible.

I'd also want to express my gratitude to my spiritual parents, Nolufefe and Thabo Mchunu, who have aided me in my development and given me the confidence to remain faithful to God and pray at all times. Also want to thank my spiritual brothers: Sandile Xaba, Smangaliso Hlatshwayo, Lindelani Ndlovu, Bhekani Nzimande, Vuyani Kaunda, Sphelele Skhakhane, Lindokuhle Ngubane Thulisani Ngubane, Sibusiso Zungu, Sandile Mazibuko, Lonathemba Mvelase, Luyanda Tenza, Nhlakanipho Shabalala, and Sbonelo Ngcongo.

A special thanks to my In-Laws for their support as well, my Mother In-Law Bongiwe Ndlovu and my Brothers in law, Sthembiso Ndlovu, Sabelo Ndlovu, and Mbuso Ndlovu.

Plagiarism Declaration

1. I know and understand that plagiarism is using another person's work and pretending it is one's own, which is wrong.

2. This report is my own work.

3. I have appropriately referenced the work of other people I have used.

4. I have not allowed and will not allow anyone to copy my work with the intention of passing it off as his/her own work.

Surname and Initials	Student Number Signature
_____K.W_____	_ _____

Copyright

The author of this thesis owns some copyright and has given the University (DUT) permission to use it for a variety of purposes, including administrative ones. Copies of this thesis may only be created in accordance with the copyright. Reproductions cannot and must not be made accessible for use without the owner's express written consent.

Dedication

Dedicated to my entire family, particularly my mother (Margaret Sibiya), Fiancée (Sinethemba Ndlovu), and Son (Aphilenhle Laluhle Sibiya).

Abstract

The continuing global surge in various cloud services, IoT, and Edge (Fog) computing has led to a sudden increase in the demand for Datacenters. By definition, a data center is **a** physical facility that corporations/organizations use to house their critical applications and data. A data canter's design is based on a network of computing and storage resources that enable the delivery of shared applications and data.

Notable advantages of Data Centers include but are not limited, to their ability to provide services to end-users based on affordable rates in various plans as per contractual agreements. They also offer a robust hardware ecosystem as well as software. In operational terms, data centers offer reliable and enhanced system performance by way of carefully distributing the traffic loads uniformly across the cluster nodes. In that way, end users are excused from maintenance responsibilities. Data centers also afford instant scalability based on changing capacity demands by users. To enhance the fail-safe abilities of data canters, backup systems are incorporated. A notable drawback of Datacenters is the high power consumption which up both CAPEX and OPEX costs. E.g it is prohibitively costly to erect robust cooling systems for a large-scale data center. The same cooling system ought to be scalable to accommodate future expansions of the data centers in terms of new services that may require new hardware to be incorporated. Thus scalability of energy supply capacity is quite a challenge. Thus, how to maximize power utilization and optimizing the performance per power budget is critical for data centers to deliver enough computation ability. Overall the operational costs of Data centers directly link the resource management algorithms implemented to assign virtual machines (VMs) to actual hardware servers and degrees of flexibility to relocate them elsewhere in case of emergencies usually associated with power losses of excessive heating of system elements. The main contribution of this thesis is in proposing and analyzing a hierarchical SLA-based distributed hierarchical resource allocation and optimization scheme, that considers constraints such as energy consumption and cooling-related energy consumption in addition to the scalability issue. We also incorporate a load-balancing algorithm to minimize the operational costs of the proposed scheme. We utilize CloudSim, which is a customizable tool that supports the modeling, and creation of several VMs, (as well as mapping tasks to appropriate VMs) for the scheme's performance evaluation. Ultimately obtained results show that the scheme significantly reduces the operational costs of the overall cloud data center system and at the same time ensures energy efficiency.

Table of Contents

List of Figures

List of Tables:

List of Abbreviations:

FCN - Fibre Channel Name.

WLAN - Wide Local Area Network.

HTTP - Hypertext Transfer Protocol.

XaaS - Everything as a service.

PaaS - Platform as a service.

SaaS - Software as a service.

EA - Enterprise Application.

IOS - iPhone Operating System.

TCP/IP - Transmission Control Protocol/Internet Protocol.

ISP - Internet Service Provider.

SMTP - Simple Mail Transfer Protocol.

MTA - Mail Transfer Agent.

EoR - end of the rack.

ToR - Top of the rack.

VLB - Valiant Load Balancing.

COTS - Commodity Off-The-Shelf.

BFS - Breadth-First Search.

LSA - Link State Advertisement.

CPS - Control and protection switching gear.

LISP-MN - Locator/Identifier Separation Protocol.

NFC - Near Field Communication.

OTA - Over-the-Air Technology.

FFD - first-fit decreasing.

RAM - Random Access Memory.

FCN - Field Control Node.

OSPF - Open Shortest Path to forwarding.

SQL - structured query language.

JSON - JavaScript Object Notation

MQTT - MQ Telemetry Transport.

SLA - Service-Level Agreement.

1. Introduction

1.1. Motivation

By definition, a data center typically refers to a collective system(facility) that renders shared access to computing aiding resources such as applications and data via a defined network, The network itself typically comprises storage facilities and computing hardware and systems. Various Standardisation bodies are actively defining the construction designing and maintenance of typical data centers and associated infrastructures to enhance data privacy, security as well as system availability, and resilience. An example Data center is shown in Figure 1-1, [1].

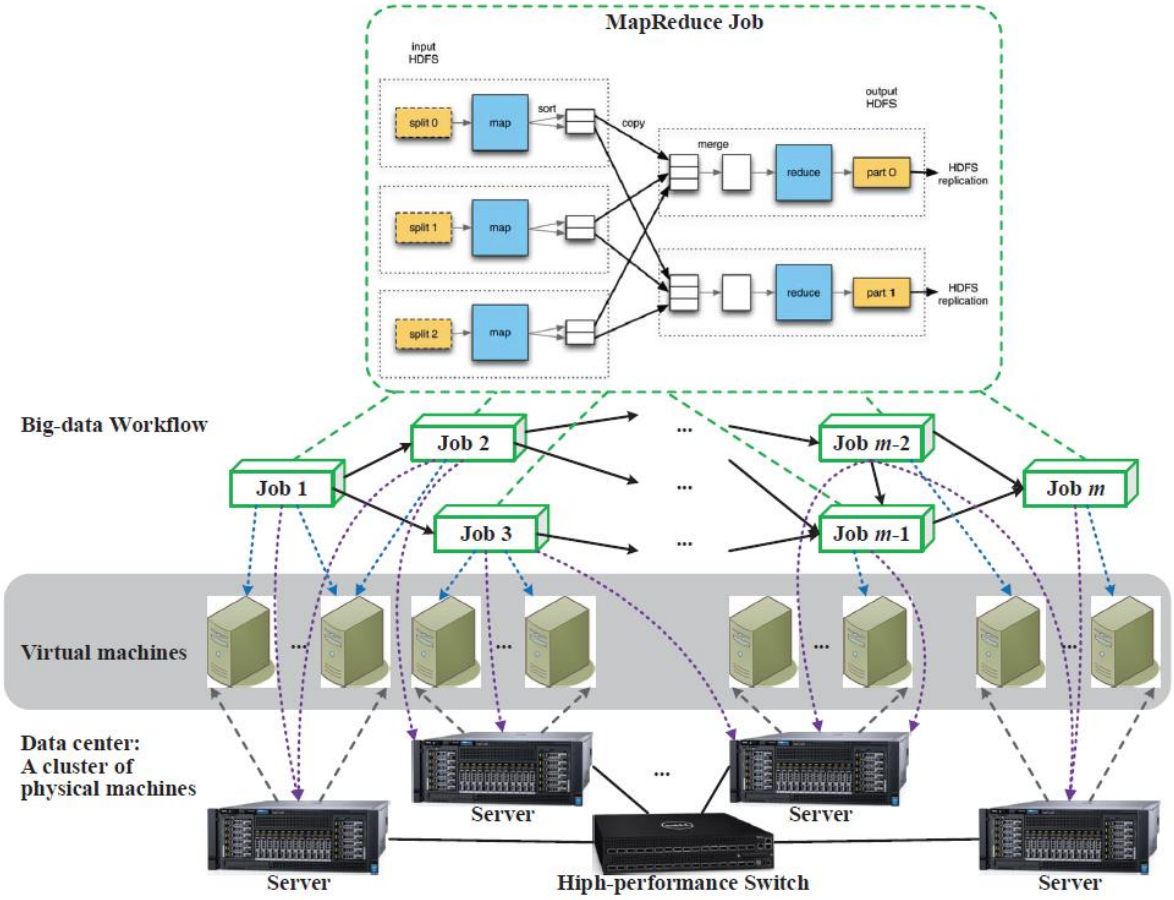

Figure 1.1: Example Data Center Configuration

Data centers vary in size and geographical coverage even though they all serve similar goals, i.e to provide computing facilities to enterprises and individuals. Nowadays the same systems

have since evolved from on-premises infrastructure oriented to one that connects on-premises systems with cloud infrastructures where networks, applications, and workloads are virtualized in multiple private and public clouds. We can enlist examples as follows, [2], [3]:

- A Cloud data center is mostly widely distributed in nature and often accessible via third-party managed service providers.
- An Enterprise data center is owned by an individual large corporation to cater for internal purposes. Most large multinational corporations such as Microsoft, Dell , Siemems AG, and others do have their individually owned Enterprise data centers.
- Colocation data centers are rented out computing space and resources of an existing data center.
- Managed service data centers will serve end-users directly by providing data storage, computing, and other services. They often operate on a third-party basis.

It is generally noted that the surging numbers of operational data centers and associated services coupled with the emergence of the Internet of Things(IoT) as well as have triggered more demand for data centers globally, [4]. These data centers require lots of electrical power and indirectly this contributes to elevated carbon emissions. Consequently, a sizeable fraction of existing data centers may ultimately run out of operational electrical power capacity soon. Predictions are that the global power consumption by this sector may surpass 1,500TWh in the next decade [5]. This amount of power demand will further exacerbate carbon emissions.

It is noted that the electrical power consumption of data centers contributes significantly to both OPEX and CAPEX. On average it is estimated that the power demands as well as associated infrastructure, coupled with the cooling system, typically cost about ZAR 1500-2250 per watt of IT critical. Moreover, it will cost multiple millions of Rands to further expand (capacitate) and existing data center in cooling capacity terms. Needless to also emphasize that the power infrastructure costs are likely to exceed the energy two-fold over a 20-year lifespan. The ever-rising power tariffs globally have also escalated the OPEX costs hence making it a major significant cost of a data center's operating expense. Blending traditional generating sources with renewable equivalents by Cloud service providers is lowering overall energy costs as well as

carbon emissions. It is therefore quite imperative to maximize the utilization of data center capacities to reduce the Total Cost of Ownership(TCO).

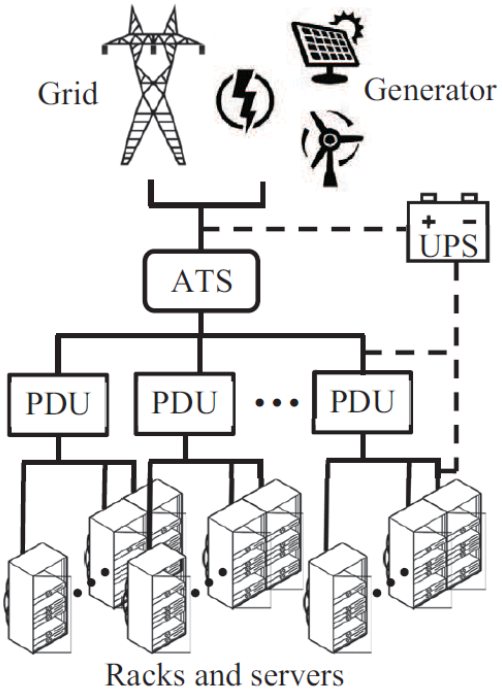

Figure 1.2: Energy Consumption

Figure 1-2 illustrates power usage in a typical data center. This is done hierarchically to minimize TCO and at the same time maximize utilization of the data center's computing resources. Typically data centers incorporate fossil-based power generators, uninterrupted power supplies (UPS), as well as power distribution units (PDUs). As further illustrated in Figure 1-2, it is noted that the power hierarchy comprises a high-voltage grid power feeding high voltage (HV) to the data center via an automatic transfer switch (ATS). The latter switches between grid power and hot standby generation (during utility failures). Ultimately the power is stepped down to the 220 Volts level before being fed to the UPS. The UPS thus provides a protected power supply to various power distribution units (PDUs) that in turn supply power to various racks, [6].

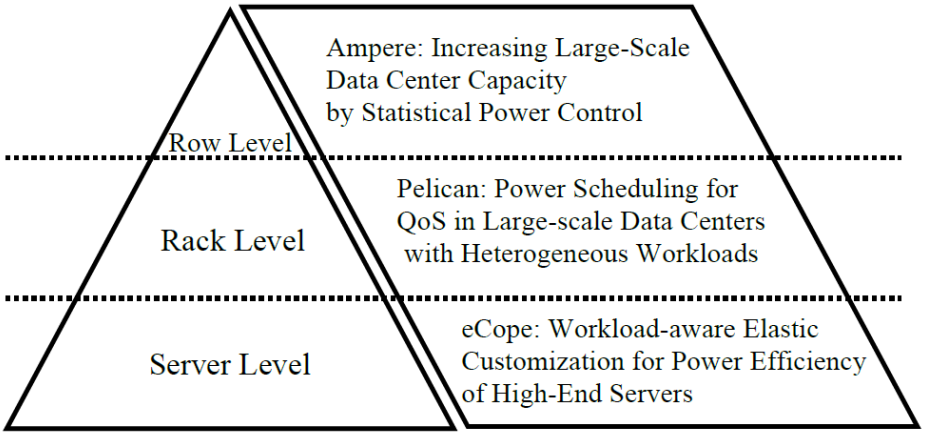

Figure 1.3: Typical Power Power Hierarchy in a Data Center

There exists a rack PDU feeding directly computing elements. They (servers) are supplied following the prorated electrical energy budget. The rated supply is determined by each sever's maximum power rating. To afford to enhance power protection it may be necessary to provide extra UPS before each row/rack PDU further complementing the already existing centralized UPS. Note that UPSs placed before an ATS and level PDUs output voltages at HV or MV levels hence are relatively costly. However, the PDUs at the rack level and other UPSs output relatively lower voltages and are thus correspondingly cheap, [7].

1.2 Problem Statement and Overview

Key challenges in present-day data centers include insufficiency in the capacity to handle computational demands by various end-users at the desired grade of service (GoS). Multiple factors and challenges are constraining the capacity of a data center. These can be summarily listed as follows:

- Industry regulations: Depending on the nature of business, and possibly data center placement (location), various levies can be incurred and in the process, this affects capacity planning. Such levies include carbon emissions and Privacy & data protection taxes Enhanced privacy and security issues must be taken care of public service customers. To achieve external accreditation, adherence to set ISO 9001/2001 standards (Europe and Africa) or ASHRAE standards (North America) may be necessary, [8].

- Rack optimization: Required operational power typically ranges from 5kW to 0kW per rack. It is necessary not to over-provision data center resources. Typically one may

want to focus on utilization maximization by way of considering high-density racks rather than standard cabinets thus in the process considerable savings can be achieved.

- Power utilities: power related CAPEX and OPEX costs are a key integral aspect of capacity planning and provisioning. Thus to establish a new data center it might be necessary to consider power tariffs at various global locations so that one can locate it appropriately. Long terms power fixed power tariff contracts, typically 15-20 years provide a reasonable degree of certainty in terms of viability for the foreseeable future.

- Cooling: A thorough comprehension of cooling requirements is critical in accurately sizing a data center. This is because the required cooling capacity varies seasonally, as well as daily.
- Greening and environmental responsibilities: Corporations and other industries that adhere to the carbon footprint commitment by way of greening their operations, always thrive to lower the Power Usage Effectiveness (PUE) consistently. Thus substituting fossil-based generating plants with greening ones is a crucial aspect of data center capacity planning.
- Budget: The scalability of existing data center capacity can drastically reduce future expansion costs. As such smart planners typically prolong the lifespan of existing facilities by scaling extra components as well as incorporating flexible hardware.

Overall challenges faced by service providers concerning the implementing and management aspects include the following, [8]:-

- Real-time Monitoring and Reporting: Measures have to be put in place to constantly monitor and detect failures before they can propagate throughout the data Real-time monitoring of running applications, hardware as well as cooling and power distribution systems, must be enforced at all times. The acquired reporting metrics on fault occurrences as well as general performance will ultimately assist the service provider in making improved capacity planning decisions.

- Capacity Planning and Management: Often service providers over-provision resources of a data center as a way of ensuring resilience and reliability However this often leads to wastage of resources, space, and power. Thus appropriately dimensioning required

resources and space will help alleviate this problem. By implementing a proper data center infrastructural management (DCIM) system all physical space, capacity, power, cooling, and other resources can easily be identified. In so doing massive savings will be achieved as it will become relatively easy to optimize capacity as well as curb running costs and expenditures.

- Uptime and Performance Maintenance: Precise determining of the overall performance as well as ensuring the uptime of a data center system is crucial. This will also involve power maintenance and cooling at the desired levels. Note that a DCIM system as discussed can help in the acquisition of PUE metrics in real time. Once again this will assist in optimizing and managing the uptime s and other performance-related metrics.

- Energy Efficiency and Cost Cutting: It is important to implement measures and mechanisms that will promote the overall energy efficiency of the data center and consequently this will cut down on costs.

- Staff Productivity Management: The implementation of a DCIM system will help in tracking, analyzing, and reporting all faults in the data center infrastructural system. In that way, Staff and other maintenance personnel will focus on other duties and consequently leading to an improvement in staff productivity.

Having outlined the challenges and capacity planning challenges we now go on to explore the challenges of addressing improving power efficiency in the data center. In this regard we look at it from various levels e.g data sever and, row (rack) levels. Note that at the server level, most modern Computer/server designs are designed to maximize power efficiency. In our view, it is desirable that given specific service and workload limits, hardware customization of the server must be implemented such as to achieve energy efficiency. Specifically, we should always thrive to determine an optimized flexible workload power function complemented with hardware customization at the server level. At rack levels, it might be necessary to thoroughly analyze the behavior of, electrical power and associated tasks in a data center. This must be further complemented with appropriate electrical power scheduling in the case of heterogeneous workloads being served at the data centers. We briefly discuss some of these approaches in the next few subsections.

1.3 Energy Efficiency by way of Server Customization

Most server hardware components such as CPUs are generally designed to be energy efficient. However, at the moment the servers themselves are yet to be energy proportional. Lots of literature have explored the energy efficiencies of servers in data centers in this regard. Notably, several authors have proposed a "performance per watt" metric. Research has gone as far as determine the same metric for both homogenous and heterogeneous workloads. However, with a specific focus on heterogeneous loads, it is generally noted that the characteristics of the workload are never steady, nor are they predictable. This is partly attributed to the mere fact that services and applications tend to be agile as well as flexible. As a result, energy proportionality design becomes intricate. Research has also demonstrated individual servers have service/application-specific energy efficiency ratings depending on the service or application running, the magnitude (size) of the VM, the workload, and the extent of the scalability. Furthermore, turnaround time variations are also quite pronounced. Based on this brief literature survey, job-aware flexible customization for energy efficiency of CPUs was further proposed In this way advantages can be taken of any configurable hardware to enhance the energy proportionality for heterogeneous workloads, [9].

1.4. Statistical Power Control and Rack Level Capacity

As is well known, the overall electrical power budget in a data center is never fully. Normally only up to about 57% of the budgeted power is utilized. As a result this significantly increases both CAPEX and OPEX costs power-wise, [10].

Normally when designing data centers, the traditional approach is to ensure that the aggregated rate is not less than the power budget. However, the underutilization stems from the mere fact that most server systems incorporate power-saving routines such that the actual rated power is far less. Moreover, the actual consumed power is a function of the utilization The rather lowered utilization is often due to the fluctuations in the heterogeneous workloads as well as trying to maintain the service level agreement (SLA) for delay-sensitive services and applications. One way of increasing the computing capacity would be to consider having more servers than are needed. This is termed over-provisioning. Statistical power control has since been proposed. The idea behind this approach is briefly explained as follows. Rather than imposing a power restriction (limit) which otherwise would worsen the overall performance of the data center in terms of the jobs already in progress, a statistical form of power control is instead

implemented to dynamically manage the available power by way of varying the workload scheduling algorithm.

As such rather than exerting electrical power control at rack levels, rather power constraints are imposed at the row level, and in the process, more space is created for overprovisioning. In practice, fewer jobs are dispatched by the scheduler to rows that are already power constrained.

.

1.5. Power Scheduling for Heterogeneous Workloads in Data Centers

Conservative server often leads to very low energy efficiencies in existing data centers. Given that there exist both online and offline services, we thus can combine both on the same set of servers and in that way, power efficiency will drastically improve, even though that would pose additional challenges such as, [12], [12], [13]:

- The quantity of online service jobs is a function of end-user inputs.
- massive as well as intermittent power fluctuations are not unusual.
- Power increase demands are relatively shorter in comparison with the case of offline jobs.
- The scheduler's complexity increases when there is a demand for heterogeneous workloads to run harmoniously and as such a strong coupling exists between the degree of a job scheduler's complexity versus the amount of power consumption.

1.6. Research Problem and Aims

The work aims at working out solutions to reduce energy usage in the data center communication subsystem. The project's main goals are to find ways to increase the energy efficiency of cloud data centers and communications subsystems without impacting the functionality of the services and applications they serve. In summary, the following research issues are looked upon.

- Various elements of resource management and their potential impact on equipment/accessory energy consumption.
- Because cloud computing is a dynamic as well as a diverse environment with the same heterogeneity of services and applications, heterogeneity may be handled.
- Providing resources in a way that maximizes performance while consuming the least amount of energy possible. This includes the development of systems that increase energy efficiency by intelligently allocating available resources to diverse services and applications while maintaining consistent quality of service (QoS) standards.
- Identifying the many aspects that influence performance and energy use. This includes assessing the performance and power consumption implications of various management features available in current data center servers and applications.

- We also give simulations by developing simple online performance and power model monitors and estimators that dynamically determine the correlations between resource utilization and application-level performance and power consumption. Proposing energy-efficient resource allocation method methods that incorporate software and hardware strategies to increase energy efficiency while preserving service and application performance and minimizing resource allocation variations.

1.7. Thesis Objectives and Contributions

Network re-design, traffic engineering, power-aware networking, adaptive operation, proxying/virtualization, and energy efficiency network protection are the five basic methodologies employed in diverse literature. In general, re-designing the physical connections and core switches may dramatically reduce the energy consumption of the core network that offers interconnectivity to Cloud and Data Centers. Physical linkages have traditionally relied on wireless and optical technologies, whereas core nodes have mostly relied on electrical technology. Because electronic devices are slower and spend more energy than their optical component equivalents, most electronic networking devices and components must be gradually replaced with optical equivalents [2]. However, replacing electronic switching gear with an OXC allows for ultra-fast switching owing to the removal of bottlenecks caused by slow electronic processing in general. Furthermore, because optical devices require less energy, potential energy savings will be obtained [3].

Massive energy savings may also be gained by re-designing the physical architecture of such a network via the optimization of available connections [4]. Because link deployment expenses account for the majority of an optical core backbone network's CAPEX, it's sensible and rational to connect core nodes with the fewest number of connections feasible. This will have several advantages, including better hardware utilization and on-demand resource provisioning, as well as lower CAPEX and OPEX expenses [5, 6]. Such networks would require more flexible management methods to aid in the relocation and restart of virtual machines if they adopted a virtualization strategy. The following is a summary of the goals:
Designing a Fog to Cloud design for a Data Center Energy Efficient Approach based on the Internet of Things (IoT). The performance of the provided architecture is evaluated using analytical and simulation modeling.

1.8. Overview of the Thesis

The remainder of this thesis is divided into the following sections.

In Chapter 2, we will provide a general description of the general infrastructure of the data center system and key design and operational aspects. Virtualization and containerization technology principles were overviewed. Energy consumption and performance-related issues were also dealt with. An overview of key methods implemented to improve both energy and performance efficiency in distributed data center systems will also be covered. Note that the power efficiency approaches are addressed at hardware, applications, and resource management levels. We also provide insight into, energy, performance, and cost management issues in a distributed system.

Chapter 3, devotes to scheduling now that it is a key integral part of facilitating (enabling) energy efficiency in data centers. We will carry out an extensive survey of energy-efficient scheduling algorithms. It is noted that the designing of a new generation of scheduling algorithms that are energy efficient was triggered by a rise of data center-associated workloads These will aim to appropriately dimension the available data center resources subject to certain objectives that include but are not limited to maximizing energy efficiency, maximizing utilization, minimizing live VM migration, preventing QoS degradations of jobs already being served. In terms of classification, we categorize them into global, local, centralized, distributed or hierarchical, static, and dynamic.

Chapter 4 Centers on the operation of energy-efficient data center systems. The same chapter also discusses the choice of configurations and operation procedures. Various paradigm architectures are overviewed. The focus is on mitigating an architecture that would best promote overall cost-effectiveness and energy efficiency. The implementation of appropriate resource allocation, as well as scheduling algorithms is will be further discussed taking into consideration the multi-objective goals that have to be met. Insights pertaining to features and characteristics that influence its overall performance will be explored. Given that selection of one paradigm over the other requires careful consideration, we will thus list down a set of criteria that designers can base on in choosing a particular paradigm.

Chapter 5, devotes mostly to discussing key drivers for achieving energy efficiency. In particular, we made reference to a Fog Cloud data center paradigm. The choice for such a paradigm is that the Fog servers are normally placed in proximity to end users and as such QoS related issues such as latencies for critical mission services and other applications can easily be overcome. We also note that the key to achieving energy efficiency in its operations would be sound load balancing among the active servers as well as appropriate scheduling. The chapter also extensively reviews typical Fog Cloud frameworks. We also carry out an in-depth classification of load-balancing approaches. A load balancing framework/scheme is proposed and a and analyzed in respect of the key QoS metrics that would affect both the users' satisfaction as well as overall energy efficiency A comparative performance evaluation of the algorithm was carried out at Fog as well as at Fog-Cloud data center levels. Overall it is concluded that load balancing at VMs level coupled with sound scheduling is key to achieving energy efficiency.

1.9. Chapter Summary

In this chapter, we defined the various data center types as well as factors to be considered in their design. Challenges are also explored Overall we note that notable advantages of Data Centers include but are not limited, to their ability to provide services to end-users based on affordable rates in various plans as per contractual agreements. They also offer a robust hardware ecosystem as well as software. In operational terms, data centers offer reliable and enhanced system performance by way of carefully distributing the traffic loads uniformly across the cluster nodes. End users are excused from maintenance responsibilities. Data centers also

afford instant scalability based on changing capacity demands by users. To enhance the fail-safe abilities of data canters, backup systems are incorporated. The objectives and aims of the thesis are also spelled out. The next chapter will mostly center on energy consumption and performance-related issues. An overview of key methods implemented to improve both energy and performance efficiency in distributed data center systems will be discussed. The power efficiency approaches will be addressed at various levels, namely hardware, applications, and; resource management. A taxonomy of the various performance management approaches will also be overviewed together with insights into energy, performance, and cost management issues in distributed systems.

2. Energy-Efficient Data Centers Overview

The chapter devotes to describing the general infrastructure of data centers and key design and operational aspects. Throughout we assume energy efficiency design and operation. As is known most utilize virtualization and containerization technologies in providing computing services to customers and other end, users. In this case, the required applications are provisioned via cloud centers. Thus the chapter will mostly center on energy consumption and performance-related issues. An overview of key methods implemented to enhance both energy and performance efficiencies in distributed data center systems is also covered. Note that the power efficiency approaches are addressed at hardware, applications, and; resource management levels. We also provide a taxonomy of the various performance management approaches. We also provide insights into, energy, performance, and cost management issues in distributed systems.

2.1. Introduction

Computing demands are steadily increasing worldwide and this is acting as an impetus for the deployment of more data centers. Side drivers include IoT and other network-based services and applications. To cite an example, at an individual level, online services such as banking, health services, commerce, and software as a service(SaaS) generate massive loads of data. Public and private entities have also upped demands for data processing and this happens to be pacing quite rapidly,[14].

On the background of all these Data centers have become the de facto backbone of current and future IT infrastructures. This sector (data centers) is steadily dominating in terms of overall global energy consumption.

As alluded to in the previous chapter, a significant proportion of power consumption is lost resource over-provisioning. Furthermore, the tendency by Service providers to provide required power based on aggregated rated peak only exasperates the problem of overall under-service level agreements (SLAs) between end users and the data centers to which they subscribe. The SLA generally requires elevated GoS throughout and thus peak power assignment is preferred to seclude any possible bottlenecks that may otherwise ultimately degrade performance (GoS**)**.

Achieving good power management in the entire system is also. This is because the limited power availability coupled with huge consumption demands by IT-related hardware intricates overall power management. Secondly, a large proportion of the power is consumed in the physical infrastructure. Thirdly, regulating the peak instantaneous power consumption must be carefully managed. Fourthly, the differences in the granularity nature of, power budgets in, clusters, racks, or even servers of a data center also make effective power management difficult to achieve. Finally, the trade-off between energy consumption versus system performance must be managed carefully the system's performance and the power manager must trade-off carefully in this regard. The varying cooling threshold demands (in critical temperature terms) of individual servers mean that the supply of inlet cooling air does not satisfy all the servers.

With regards to Virtualization approaches, a VM generally facilitates an application-hosting environment where independence between services /applications that share a common physical machine. is maintained [8], [15], 16]. In a way, Virtualization technology improves both power efficiency and utilization by way of consolidation. The consolidation was achieved as a result of the capability to assign multiple VMs to a single physical server. When this is done other physical machines can be migrated to sleep mode hence power is saved. However, a tradeoff has to be carefully maintained between the degree of virtualizing. This is because whereas assigning multiple VMs to a single physical machine promotes energy efficiency, the latter's resources might not be adequate hence degradation of performance may be experienced.

Migrating a VM from one physical server to another, results in downtime, typically 100 milliseconds. The distributed nature of most deployed data centers means that the peak power demands are also spread (distributed) and thus the local grid can be relieved of a sudden surge in peak power demands. Illustrated in Figure 2.1 are the key physical components of a data center, [15]

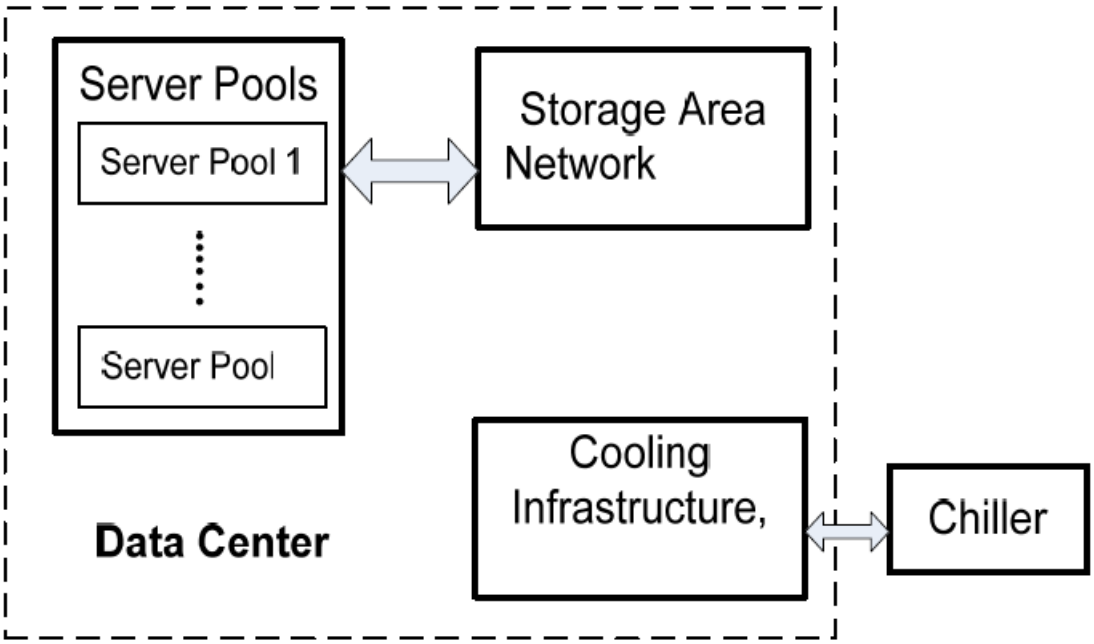

Figure 2.1: Key Components of a Data Center.

This comprises mainly components such as CRAC units and storage area networks (SANs), Servers are typically identical. An internal bus will interconnect the servers to the SAN.

A server is described as a system of a computer network that can be accessed by various users. The server is also known as software that is responsible for delivering service. The server hardware can be dedicated to its service or it may not be. The components used for the server provide high performance and these components are high grade and last for long period. A certain computation application determines the server hardware. Thus server hardware is split into two categories:

- **Components of Server Hardware:** At this level, few factors contribute to the bettering of server availability and performance. Processor performance, storage capabilities, and transmission duration, bus network interconnect capacity, HDD /magnetic tape peripherals performance, and network throughput are the factors to consider.
- **The architecture of Server Hardware:** The microprocessor's performance limits the server's processing capability. Servers have switched to multiprocessor servers to achieve excellence and availability. Multiprocessor servers are divided into two categories: Symmetrical Multiprocessors (SMP) and Loosely-coupled Multiprocessors (LMP) are two types of multiprocessors (Clusters or Massively Parallel systems).

Components of Server Hardware include the following:

- Processor: Microprocessor power appears to be doubling regularly as technology progresses and improves. The server's performance is directly influenced by the processor's performance and speed. The server will become faster and perform better as CPU and strength increase

- Server Memory: Processors, RAM, and drives all have different performance levels. It can be measured not just in terms of delays, but in throughput terms as well. Memory access times must be minimized. The access time is determined by the memory chip technology as well as the memory system's organization. The cache, or memory hierarchy, was designed to simulate very big and extremely fast storage by blending huge capacity and inexpensive storage with the much smaller and more efficient though relatively costly memory. The cache behaves similarly to huge fast memory, although it costs about the same as large slow memory.

- Processor, Memory, and I/O Interconnection: Input or Output Interconnection has several sections. These are:
    - The link between memory, I/O bus, and processors is implemented by a system controller. This is frequently a chip that serves as both a memory controller and an I/O controller.
    - The PCI bus and its expansions have become the industry standard for I/O buses.
    - An I/O bus connects the I/O controllers. Such controllers are attached to peripherals.
    - Magnetic peripherals that are connected directly to the computer.
    - Peripheral subsystems (SAN) and communication sub-systems are connected through specialized networks (LAN). Fibre channel is becoming more popular in its sector.

The connections types for input/output systems are:

- **The link between system controllers and processors is as follows**: It has the controller memory and an I/O interface built-in. It mostly relies on telecommunication transportation. The bus connects all of the elements. The main advantage of using the bus is its simplicity. Its throughput is determined by the number of bits it can send in parallel and the frequency at which it operates. However, its frequency is limited by its length and linked parts. There are two kinds of buses.
    - Synchronous bus: It has a high frequency of operation. It has to be short in length.
    - Asynchronous bus: To govern data transport, it requires a protocol. It has low throughput however the distance has limitations
- **Peripheral Component Interconnect (PCI)**: It is the microprocessor's backbone bus. It has the following characteristics:
    - There are two different data transmission widths: 64 bits and 32 bits.
    - There are two clock speeds available: 66 MHz and 33 MHz
    - Various bandwidth options (532, 266, and 133 MB/sec) are offered.
    - Support for both 32/64 controllers on the same bus.
    - The ability to find device configurations on a bus (plug and play)
    - Hot plug's capability
- **Small Controller Single (SCSI)**: Interface has long been the industry standard for connecting magnetic peripherals. It has the following features:
    - The maximum length of a link is limited.
    - It can only hold a limited amount of gadgets.
    - Bulk.
- **Fibre Channel – Arbitrated Loop (FC-AL):** It has a much wider range of device connectivity than SCSI. It is not bulky in any way. Fibre channel is becoming more popular in the industry. Fibre Channel makes connectivity easier and more reliable.

- Data Storage: In business, data is a vital, highly accurate, and critical component. One of the most important reasons for having a server is to concentrate data so that business-critical data can be managed more effectively. The characteristics of storage systems are important considerations when selecting a server. The requirement to keep data available

at all times made memory subsystem critical components of servers, leading to the development of RAID.

2.2. Server Power Usage

Illustrated in Figure 2-2 is an arrangement of the racks in hot air/cold air setup. Typically each rack will house several servers. The power is fed from Power Delivery Unit (PDU) of the rack. Normally an individual rack is provisioned with a single PDU that supplies power to all its housed servers, [16].[17, 18].

.

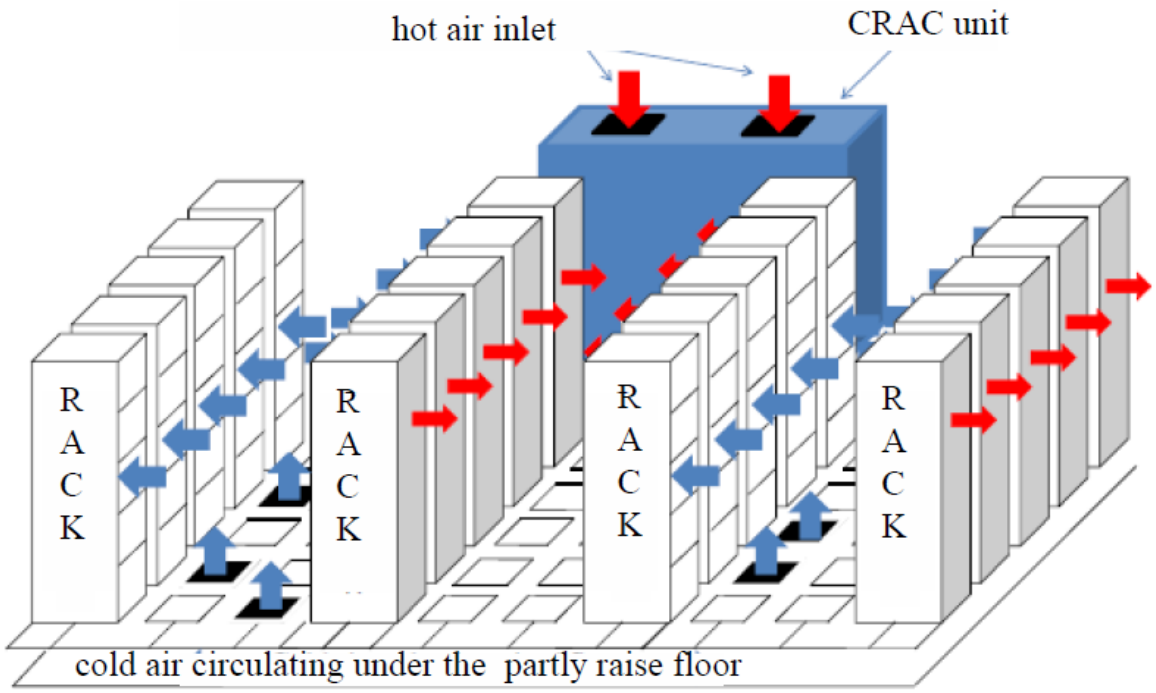

Figure 2.2: Data Center's Internal Layout

The racks are interconnected by rack nodes (switches). Most of the data exchanges are among servers housed in the same rack and as such intra-rack switch nodes typically have higher bandwidth capacities than inter-rack equivalents. The CRAC is the main cooling infrastructure. The chiller unit reticulates the returned hot air hence it is normally located outside the care systems. Cooling efficiency can be best achieved by reducing hot air reticulation. In practice, this is achieved by introducing it to the cold air inlet and in the process, the former (hot air) is effec-

tively isolated. Uninterrupted power supply (UPS) units are also incorporated to cater for power needs when there is a temporary loss of direct grid power feed. These also help to effectively reduce rippling and spikes generally characterizing rectified a.c voltages [19].

Ultimately we can define Power Usage Effectiveness (PUE) as indicative of the degree of effectiveness of a data center. It is the fractional aggregate power utilized by the equipment to the overall supplied power. The PUE of a given data center is improved by the incorporation of UPS and CARC systems [20].

The Server PUE (SPUE) is the proportion of aggregate power delivered to a server to the quantity usefully utilized.

We can distinguish a few types of energy inefficiencies in data center systems. One of these is referred to as Energy Usage Effectiveness (EUE), which is indicative of the structural buildup of the data center building(s). It is normally expressed as the ratio of total energy consumed versus aggregate IT power supply. It has been demonstrated that UPS system, chillers and CARC all affect this factor. Other technics such as fluctuating the overall temperatures in the data center system to increase CARC efficiency, will consequently lead to a reduction of the EUE, [21].

Server PUE (SPUE) is also another metric that is determined as a ratio of total energy delivered to the running servers to aggregate useful energy supplied to the data center.

The energy consumption of the server per computation unit is another measure for quantifying the energy efficiency of a data center.

Overall, as a design feature servers (PCs) can switch to low power. During this mode, it is key that the minimal possible power is utilized by the system. Overall minimizing energy consumption by the HDD, CPU, and memory will significantly improve energy efficiency Techniques such as sleep mode for Disk and CPU components, adaptive emery Management (AEM) , and adaptive power/voltage scaling (AVS) without penalty/ with latency penalty will also enhance energy proportionality of the server.

Power administration in data centers involves the following steps:

- Determining the overall power consumption profiles of the entire system.
- scheduling the job /determining consolidation potentials by VMs.
- Mapping the statistical behavior of the workloads on the energy consumed by servers in place of attempting to serve workloads with an aggregate power less than the required peak power ratings.

An Operating System (OS) is also a key component to optimally managing all server resources as well as providing services to all running applications via system kernels. Directly or indirectly, it, therefore, aids in energy efficiency

The incorporated Resource manager in the OS provides mappings between user jobs and hardware, as well as provisioning task management services. It can also provide SLA-based resource management as per need as well as taking into consideration end user's requests (e.g. require processing speeds, memory, and transmission capacity).

The OS also facilitates message passing, synchronization, as well as data storage at cluster levels.

2.3. Resource Management at Datacenter Level

The resource management system comprises three elements, [22]:

- resource arbiters,
- power managers,
- temperature managers.

Resource arbiter

The resource arbiter is depicted in Figure 2-3.

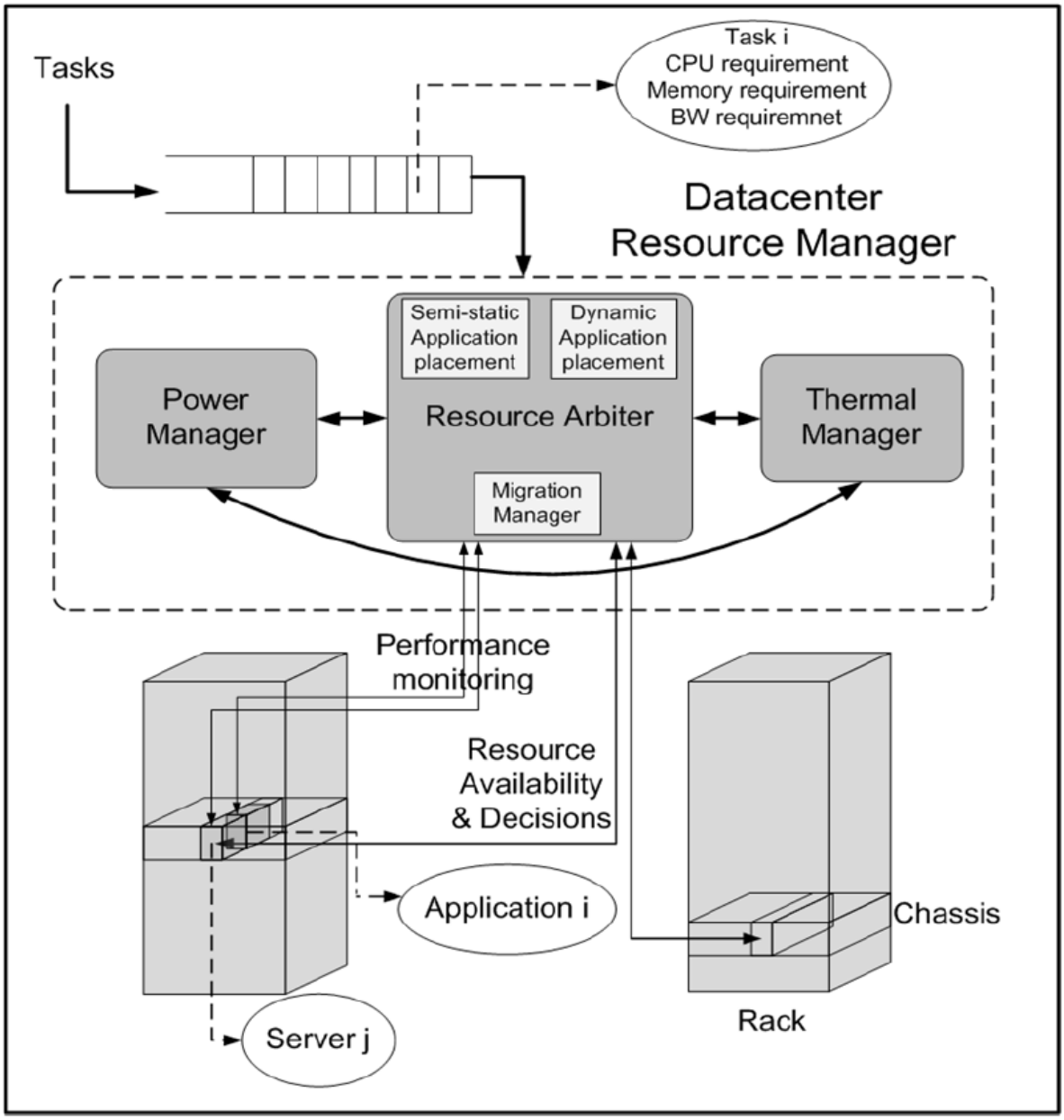

Figure 2.3: Resource Arbiter Block Diagram

Its primary role is to distribute tasks to the available (free) resources in the server. It will also liaise with other entities such as the thermal and power managers. Key to its overall role is in maximizing the number of concurrent tasks on the available resource(s) in the data center. This usually must be at minimal costs incurred in both OPEX and CAPEX [23].

The Power Manager: It primarily deals with power distribution delivery-related issues in the data center system.

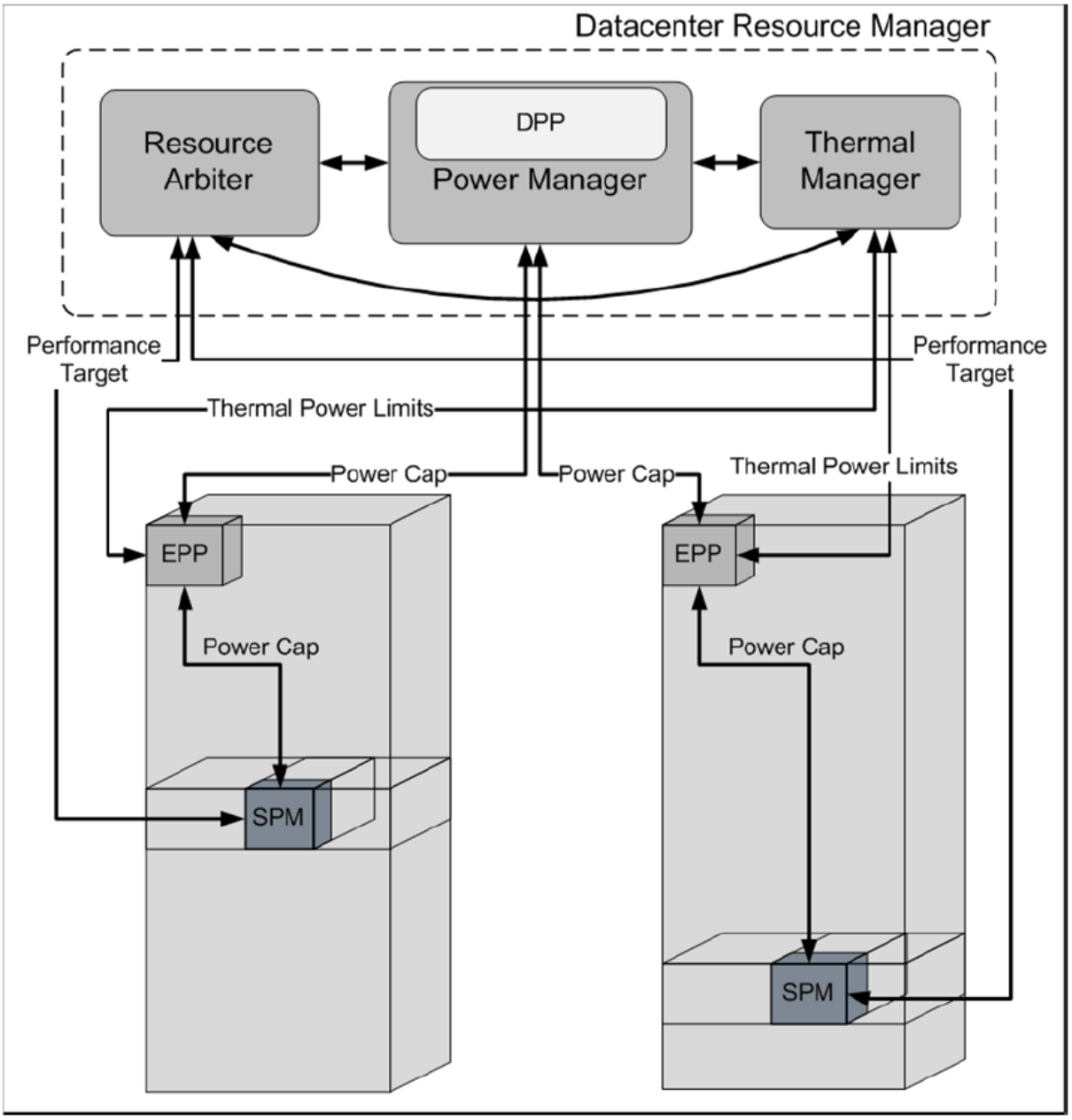

Figure 2.4: Power Management Architecture

As can be seen in Figures 2-4, several elements comprise the Power management architecture. These include data center-level power provisioners, (DPPs), blade enclosure and rack-level (EPPs), as well as server-level power managers (SPMs), [24].

Thermal Management: Most of the electricity bill in a data center (typically 30%) goes towards cooling-related tasks. Research is ongoing to try and reduce this cost by way of introducing new techniques as well as structures. Note that the "hot-aisle/cold-aisle" structure, which is quite prominent of late, is designed to improve the cooling efficiency of data center systems.

In practice, both heat exchange and power distribution networks are containerized before being water chilled. In that effective cooling is achieved. In a way, container-based data centers have higher power efficiencies in comparison to compared to today's typical data centers. They are being viewed as possible candidate solutions for next-generation data center systems.

2.4. Geographical Load Balancing

By definition, a distinction between local versus geographic load balancing is that the earlier is effected within a single data center system (localized) whereas, geographic load balancing utilizes many data centers that are often located in various places. By simple definition, Geographic Server Load Balancing (global server load balancing) abbreviated as GSLB is the distribution of workloads and traffic across servers located on different sites but belonging to the same provider [24], [25].

Geographic Load Balancing can also be viewed as a systematic strategy for reducing the energy cost of data centers that are located in different locations but belong to a single administrative domain. Research has shown that such a scenario can be modeled via a Markov Chain in which jobs will mostly be scheduled or prioritized where renewable energy is currently available. LB further enhances the effective utilization of renewable power.

2.5. Server-Level Energy Efficiency

We commence the section by emphasizing that the key to high-performance data center capabilities is in ensuring the maximization of energy (power) utilization as well as maintaining an optimized overall performance per power budget. This problem is generally tackled at various power hierarchy levels of design in the data center system as follows:-

- Server Level Power Efficient System Design (SLPESD): In this case, the power efficiency is improved by either the service/application itself or on the server level (hardware),
- Higher Level Power Efficient System Design(HLPESD). As energy consumption contributes significantly to the OPEX of a power-efficient data center, research work is also exploring how best to lower the consumption by both servers and related network devices.
- Power Efficient System Designs for Heterogeneous System(PESD-HS): Over-provisioning of power in a data center system results in increased risks of frequent power outages. As a result power scheduling will have to be carefully implemented to try to reduce overall consumption levels. The scheduling itself can be spatial dimensioned (power aware job scheduling) or temporal dimensioned (task resource management), [26].

2.6. Chapter Conclusions

The chapter focused on describing the general infrastructure of the data center system and key design and operational aspects. Virtualization and containerization technology principles were overviewed. Energy consumption and performance-related issues were also dealt with. An overview of key methods implemented to improve both energy and performance efficiency in distributed data center systems is also covered. Note that the power efficiency approaches are addressed at hardware, applications, and resource management levels. We also provide insight into, energy, performance, and cost management issues in distributed systems.

3. Energy-Efficient Scheduling in Data Centers

3.1. Introduction

In this chapter, we discuss scheduling as a key integral part of facilitating (enabling) energy efficiency in a data center. The Operating system scheduling itself is a mechanism of regulating and prioritizing requests dispatched to a given CPU.

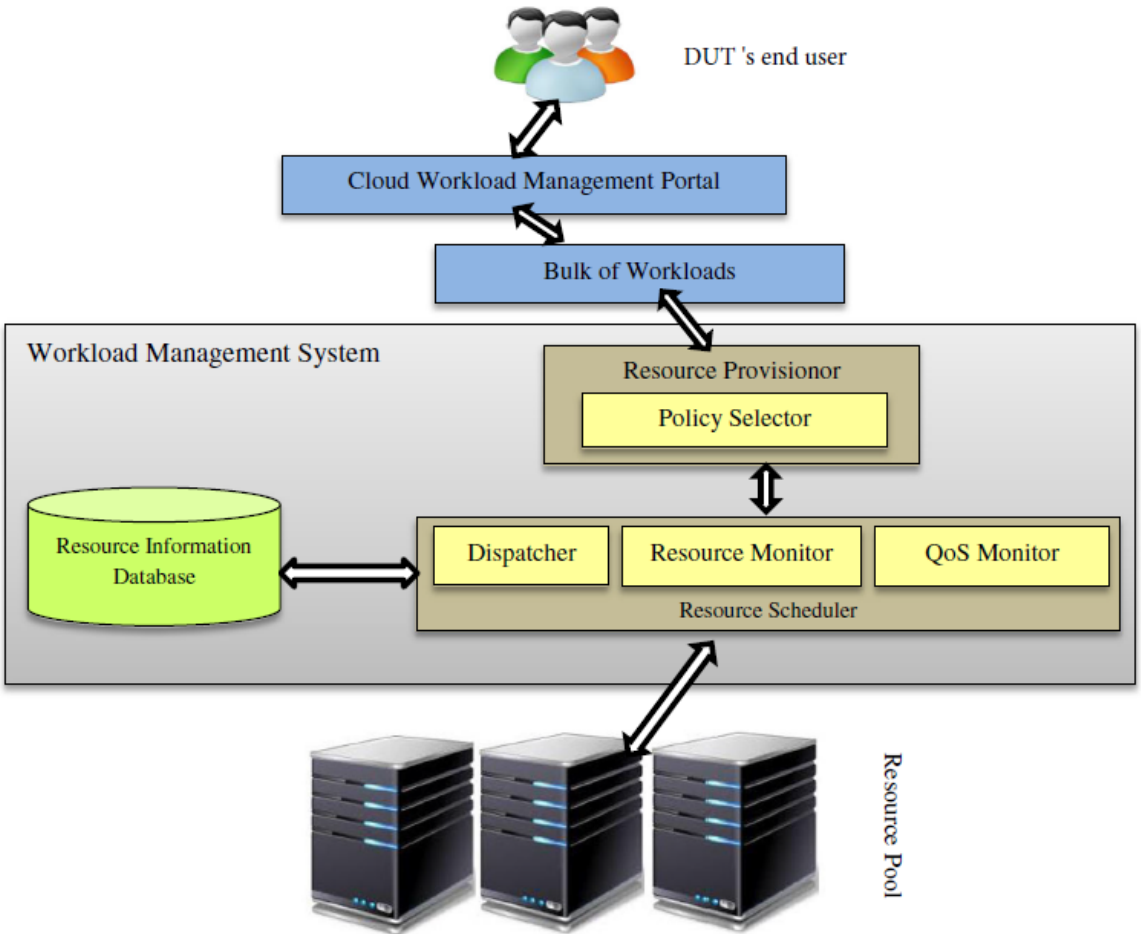

Figure 3.1: Scheduling Principles

The scheduling duties are performed by an OS internal-built routine program called a scheduler shown in Figure 3.1. We differentiate various forms of scheduling, task, job, process etc. E.g. a task scheduling algorithm primarily is a set of rules and policies used to assign tasks to the appropriate computing resources of the data center servers to get the best level possible of performance and resources. utilization. In a nutshell advantages of scheduling include the managing of available cloud data center computing performance and quality of service (QoS).

Typically, a running process involves both I/O and CPU times. As can be recalled in uni-programming power is wasted during the I/O.s waiting time. This is because the CPU can not proceed with the computations as a result, i.e it is temporarily halted as well. However, in a multiprogramming system, the situation is different as one process can await I/O feed whilst the other is in execution (CPU).

In these data centers, clusters of hundreds or thousands of machines run workloads ranging from fault-tolerant, load-balanced web servers to batch data-processing pipelines and distributed storage stacks. We can further elaborate on the differences between tasks, jobs, and processes as follows:

- A process though is usually an isolated entity that's managed by the operating system. A job is often more of an application-level term or just some script that's executed to do a specific set of task (s). A task is often part of a job - sometimes the only part.
- Job is work that needs to be done.
- A task is a piece of work that needs to be done.
- The process is a series of actions that are done for a particular purpose.
- Job and task define the work to be done, whereas process defines the way the work can be done or how the work should be done.

Resource scheduling data centers is a fairly intricate task as the dimensioning in the form of scheduling of appropriate resources to the entire data center cloud is a function of the requested QoS requirements of the services and applications. In such a scenario uncertainty coupled with heterogeneity, as well as decentralized computing resources (distributed data centers) encounters problems of appropriate and optimal dimensioning of available resources, which cannot be effected with current resource allocation approaches. We still are faced with selecting energy-efficient and appropriate resource scheduling algorithms for a specific workload from the pool of proposed resource scheduling algorithms.

In this regard scheduling algorithms for data centers must focus on energy efficiency coupled with maximization of profit margins. We hereby list down what we consider as constraints faced when designing energy-efficient scheduling algorithms for data centers [27].

- Heterogeneity. In data centers, the volatility of workloads, servers, interlinks, and connections is quite pronounced. The load demands vary in various aspects such as duration, magnitude, required resources, arrival patterns, and desired QoS. Similarly, the data center servers are not necessarily identical in the sense that their CPUs differ in processing capacities, and memory types hence speeds also differ, and rated voltage/frequency states vary. Furthermore, interconnecting (network) elements have varying I/O port speeds and network capacity (in terms of bandwidth)constraints This heterogeneity scenario makes the designing of cost-effective and energy-efficient scheduling algorithms quite complex. Thus this poses a key challenge for designing the scheduling algorithms.

- Security and Administration- Ownership. The administration ownership differs with scaling data center services and applications, issues emerge concerning privacy and security, as well as overall administration Aspects such as data center privacy and security namely, availability, identity management, privacy policies, application security, and others must be addressed. All the privacy and security-related issues must be taken into account when designing the scheduling algorithms. Similarly, administration issues involve a strategic automatic planning mechanism.

- Taking into account SLA issues. Note that SLA couples data centers and end-users (subscribers) to enhance QoS capabilities. The design scheduling algorithms therefore should always be in adherence to the desired performance objectives enshrined in the SLA. Unfortunately, this adds to already existing challenges in the design due to the unavoidable and often unpredictable, non-availability durations (downtimes) of data center resources.

- Hardware/Network Monitoring. Automatic data center resource surveillance and monitoring are necessary to increase overall performance. The acquired data (information) can be to make better-informed decisions.

- Data Availability. If all resources required for executions are available, the scheduling will proceed without delay, and thus the overall computational time is reduced.

- Profit Consideration Cloud schedulers: Clouds with reduced pricing will be preferred.

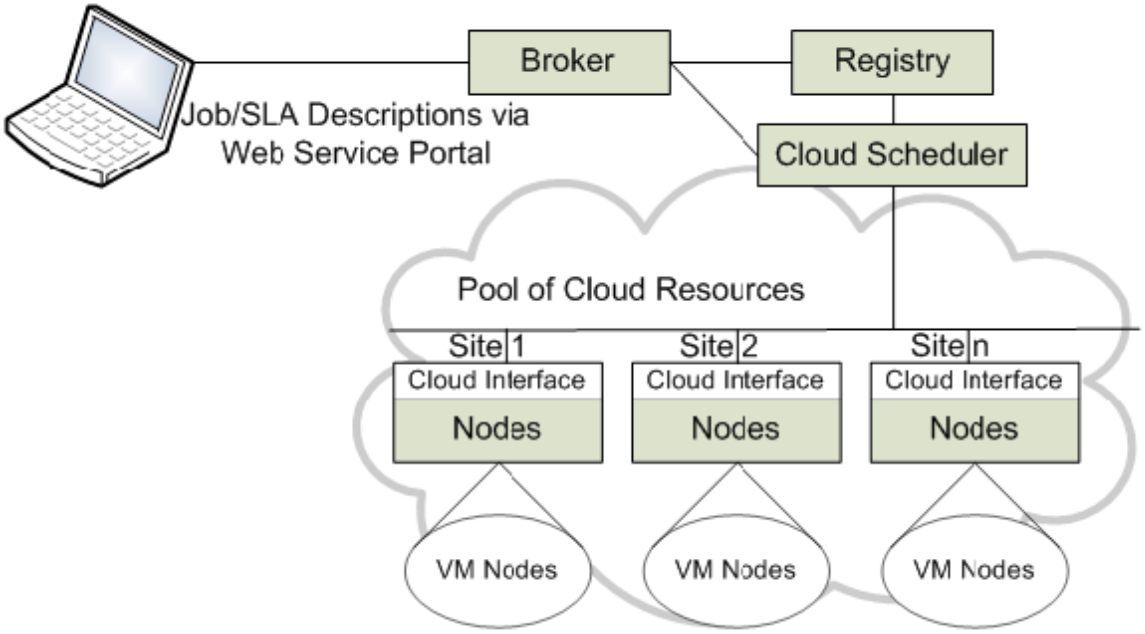

Figure 3.2: Example Scheduling in a Data Center

A cloud scheduler concurrently serves several programs and tasks from end-users. The programs and tasks may be originating from IoT /D2D enabled devices Illustrated in Figure 3.2 is an example scheduler.

To schedule jobs, the following generic steps are followed:

- Jobs are received via a web service portal. Accompanying each request (jobs) is a mandatory SLA.
- Nextly, depending on available resources as indicated in the registry the data center renders resources that are deemed to satisfy the job's characteristics. All this is furnished to the Scheduler so that it can immediately arrange optimum scheduling. E.g the latter engages idle VMs in the incoming jobs based on the objectives and scheduling algorithms.
- The next step is to create VMs that will execute the jobs, following scheduler data.
- Finally, the VMs are dissolved and the registry is accordingly updated so that the data center can be ready for other incoming jobs.

Scheduler characteristics

With regards to energy-aware scheduling, desirable characteristics(features) of each scheduling algorithm running on a scheduler include, [28]:

- Maximization of resource utilization: It must maximize utilization of the data center as this implies efficient and optimized sharing of energy consumption of services and applications.
- Energy Efficient Hardware Incorporation: Strides must be made in only incorporating energy-efficient hardware as that would lead to the promotion of enhanced energy consumption efficiencies overall.
- Heterogeneous Resources: The scheduling algorithm should deal with massively concurrent heterogeneous data center resources. Different resource configurations must also be taken into account.
- Geographical interconnections: We must consider communications costs as the actual job executions may be done at distant servers. As a consequence, long-distance communications will incur costs as well as higher energy consumption.
- Live VM Migration Issues: Schedules and associated scheduling algorithms must schedule jobs taking into consideration live VM migrations which can be expected to happen from time to time in cloud data centers.
- Profit Forecast: Profit forecasting is necessary at the time of selecting a set of resources. This implies the result may be an ultimate loss of profit. This is an important aspect to be considered by both end-users and Data center owners(providers).
- Carbon Gas Emissions: large computations will lead to the overheating of CPUs and other accessories. Cooling will result in more energy consumed and this consequently means an elevation of carbon emissions.
- Multi-Objectives: The entire operation is multiple objectives in nature.

3.2. Energy-Efficient Scheduling Mechanisms

In this section we assume a single data center, However, strictly speaking, the discussions here will also apply to Distributed data centers. The designing of a new generation of scheduling algorithms that are energy efficient was triggered by a rise of data center-associated workloads

These will aim to appropriately dimension the available data center resources subject to certain objectives that include but are not limited to maximizing energy efficiency, maximizing utilization, minimizing live VM migration, preventing QoS degradations of jobs already being served, [29, [30].

In terms of classification, we categorize them into global, local, centralized, distributed or hierarchical, static, and dynamic. We can also classify them according to implementation scenarios, as shown in Figure 3.3, i.e as a function of the number of objectives utilized such as parallel program components, advanced reservations, non-heuristic and heuristic-based algorithms, or slack reclamations we briefly define each of these in the following:

As mentioned earlier, scheduling algorithms target achieving a set of set objectives. In particular, multi-objectivity can be used e.g Pareto Genetic Algorithms.

$$f_{(.)} = \omega * f_1() + \omega * f_2() + ... + \omega * f_n() \quad (1)$$

In practice, most service-based applications are multi-objective in nature. For energy-efficient aware scheduling algorithms, a list of possible objectives would be as follows:

- Energy Efficiency Maximization. This is achieved by minimizing the power consumed by the data center. Achievable when the scheduling algorithm promotes energy-efficient computing mode on all offered jobs and tasks. Furnishing power metrics of compute nodes to schedulers would be a stride in helping to achieve energy efficiencies. DVFS-enabled CPU's will also likely achieve better energy efficiencies Taking into account the energy consumption of peripherals associated with the job being served will also assist.

- Minimizing Completion Time.: The turnaround (completion) times must be kept minimal by the scheduling algorithm.

- Maximizing Job Completion Ratio.: Job Completion Ratio is defined as a ratio of completed jobs and submitted jobs over a defined time, typically 1 second, [28].

$$Job\_Completion\_Ratio = \frac{completed \quad jobs}{Total \ \ submitted \quad jobs} \quad (2)$$

- Security Maximizing. The scheduling algorithm set must aim to maximize the security of the jobs/data in the data center.

- Data Replication or Storage Minimizing. In this case, we thrive to store all associated process data in proximity. In that way, there will be no time-wasting in the acquisition times.
- Communication Overhead. Minimizing high communications overheads if there is data that involves multitudes of inputs and outputs. In that way, more energy will be consumed hence degrading overall efficiency.

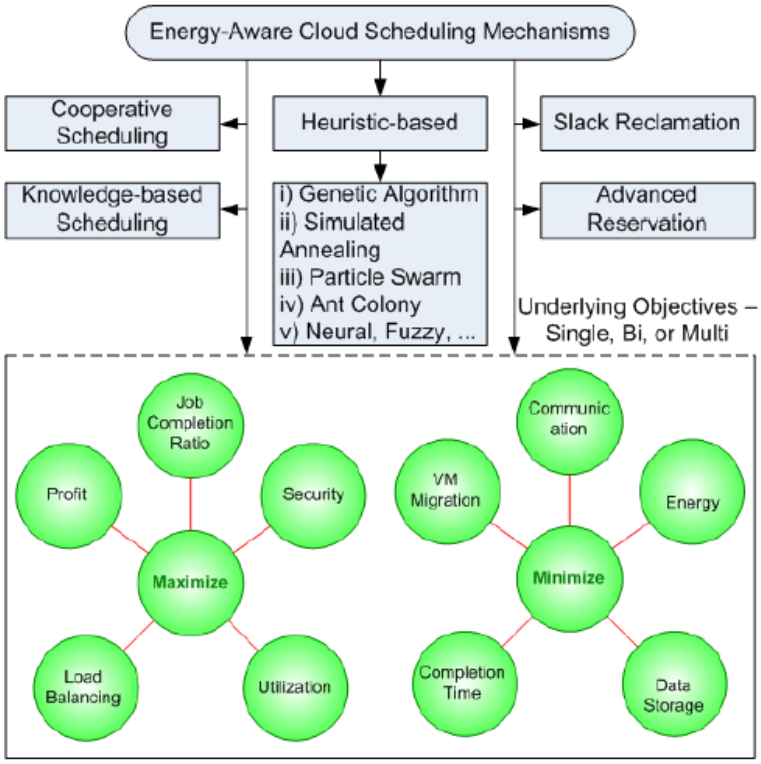

Figure 3.3: Classification of Cloud Scheduling Mechanisms

- Profit Maximizing: Data center-based applications and services must aim at maximizing revenues.
- Load Balance Maximizing: Resources must be used rationally. This implies that the load must be spread evenly across all active (available) resources.
- VM migration Minimizing: The minimization VM migrations should be promoted as much as possible by the data center system.

- Utilization Maximizing: All components constituting the data center must be maximally utilized. These include computing peripherals, and network components, such as; inter and intra-rack switches, L3 and L7 switches etc.

3.3. Cooperative Scheduling

In cooperative scheduling, the operating system thrives not to unnecessarily interrupt a process that is already running, in a bid to initiate a context switch from one process to another. As such all processes must terminate voluntarily unless blocked as a result of resources not being available.

Cooperative scheduling mechanisms can collaborate to come up with an optimum schedule. This is done by way of each agent focusing on a single objective. Iteratively it can eventually find the most ideal solution with regards to the set objective after a period of collaboration. Note, however, that the collaboration maybe with external agents, this implies a scheduling agent that is externally based. A list of example cooperative scheduling mechanisms is as follows, [30], [31]:

- Knowledge-based Scheduling: These require some form of a database for storing the interim as they try iteratively to find the best-fit schedule. Its convergence time tends to be higher in heterogeneous distributed data center environments.

- Lack Reclamation Scheduling: These ensure power saving by lowering CPU's clock frequency for slacked tasks. Slack reclamation scheduling has to maintain a balance between power saving and computational times.

- Advanced Reservation Scheduling: This approach is only effective when the resources are constant and tasks are arriving at regular intervals.

- Heuristic Optimal scheduling Heuristic in the sense that such a scheduling algorithm will not expect to encounter conflicting resource elements but still the scheduler performance can be enhanced further. These types of scheduling algorithms are iterative and thrive to determine optimal solutions swiftly. Various forms include;

    - The Generalised Genetic Algorithm (GGA),
    - Approximated Annealing (AA),

- Optimized Tabu Search (OTS),
- Quantum Particle Swarm Optimization (QPSO),
- Modified Genetic Ant Colony Optimization (MGACO),
- Artificial Intelligence-based Techniques.

We conclude the subsection by reiterating that scheduling algorithms must focus on addressing data center issues, such as scalability security, and administration. Quite some research directions are still available to address typically examples being:

- Lots more heuristics can be studied with varying energy-specific cloud objectives
- Improvements in energy-aware scheduler architectures
- Privacy and security-aware data center schedulers that focus on energy reduction.
- Data center schedulers that couple with data aggregation techniques while considering power consumption issues.

3.4. Joint Resource Provisioning & Scheduling in Distributed Data Centers

As already emphasized, data center computing provides and schedules available resources subject to set constraints and objectives [31], [32].

The fluctuating nature of the number of incidence jobs in the entire data center makes it difficult to allocate optimum resources. Provided in Figure 3-4 provides a taxonomy of resource management in data centers.

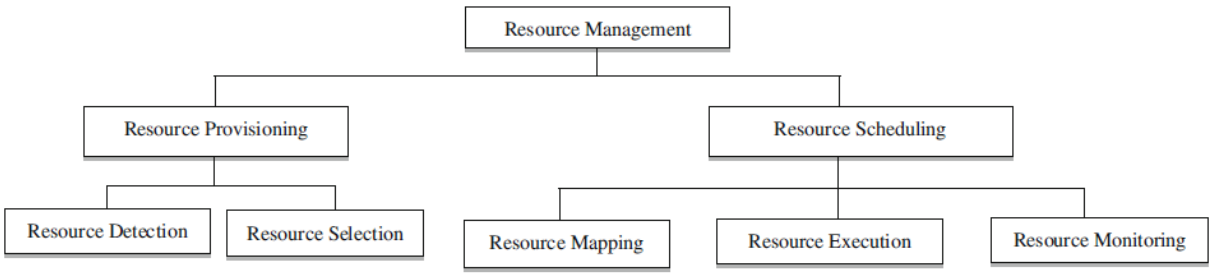

Figure 3.4: Taxonomy of Resource Management

The Cloud Data Center Resource Manager (CDCRM) agent coordinates the resources. This includes managing all incidence loads and resources and mapping available resources such as VMs optimally and efficiently.

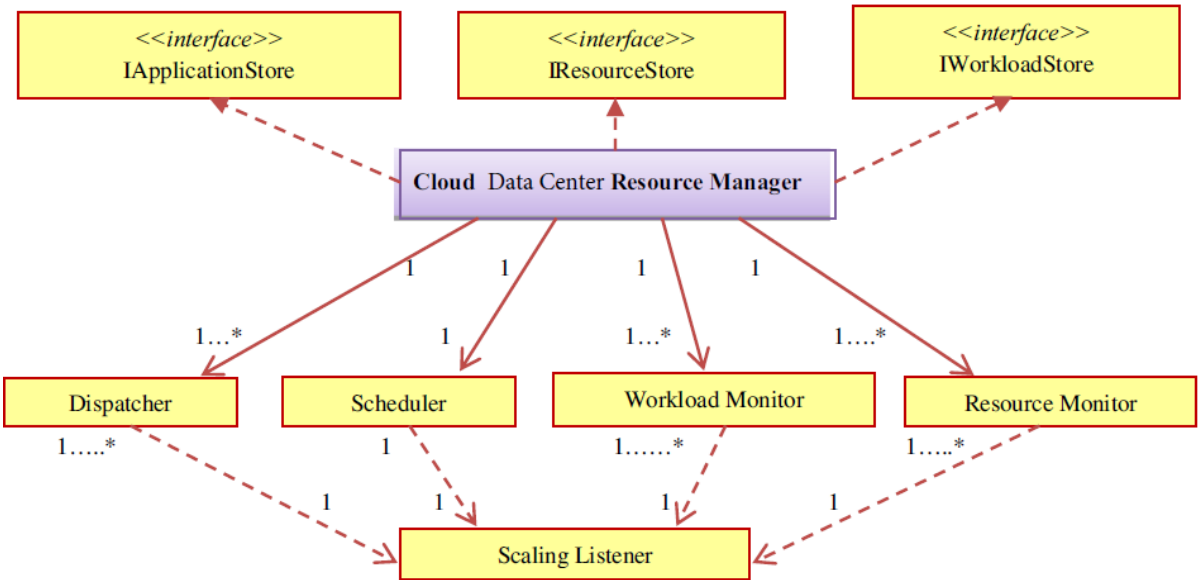

Figure 3.5: Resources Coordination in Distributed Data Centers

QoS requirements dictate how the scaling listener maps the incident workloads based on available resources. The entire arrangement is illustrated in Figure 3.5.

We further briefly discuss resource provisioning versus scheduling in distributed data centers separately.

Resource Provisioning

Figure 3.6 provides a classification of resources in a distributed cloud environment.

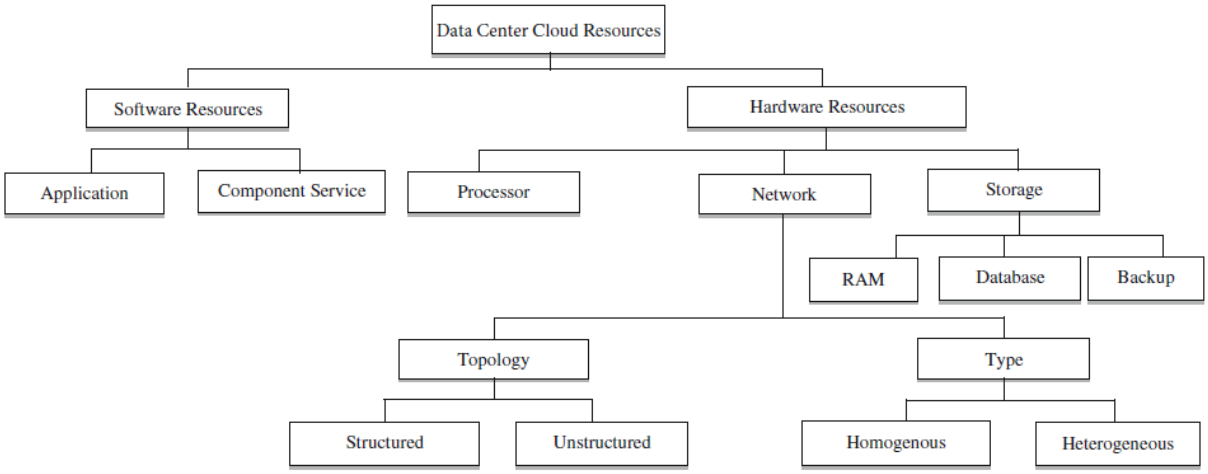

Figure 3.6: Classification of Resources in Distributed Data Centers

By referring to Figure 3.6 which is a depiction of resource provisioning/scheduling modeling in distributed data centers, we note that in the initial phase (I), a user will submit his/her intended workload for initial evaluation. The user also submits attributes such as the desired QoS which will form part of the SLA. The Resource Provisioning Agent (RPA), liaises with the Resource Information Centre (RIC) to check on the availability of resources. At this point, the required resources are provisionally reserved. If the negotiation is unsuccessful, the RPA may have to suggest a compromise QoS, and the SLA will be amended accordingly.

Once the negotiations for resources have succeeded, the workload will be furnished to the next scheduler. It will in turn seek the final provisioning of resources for its own operations. This is a phase also referred to as. resource discovery or resource detection. Finally, resource selection, which is a process of selecting the best resource set from a list availed will be carried out.

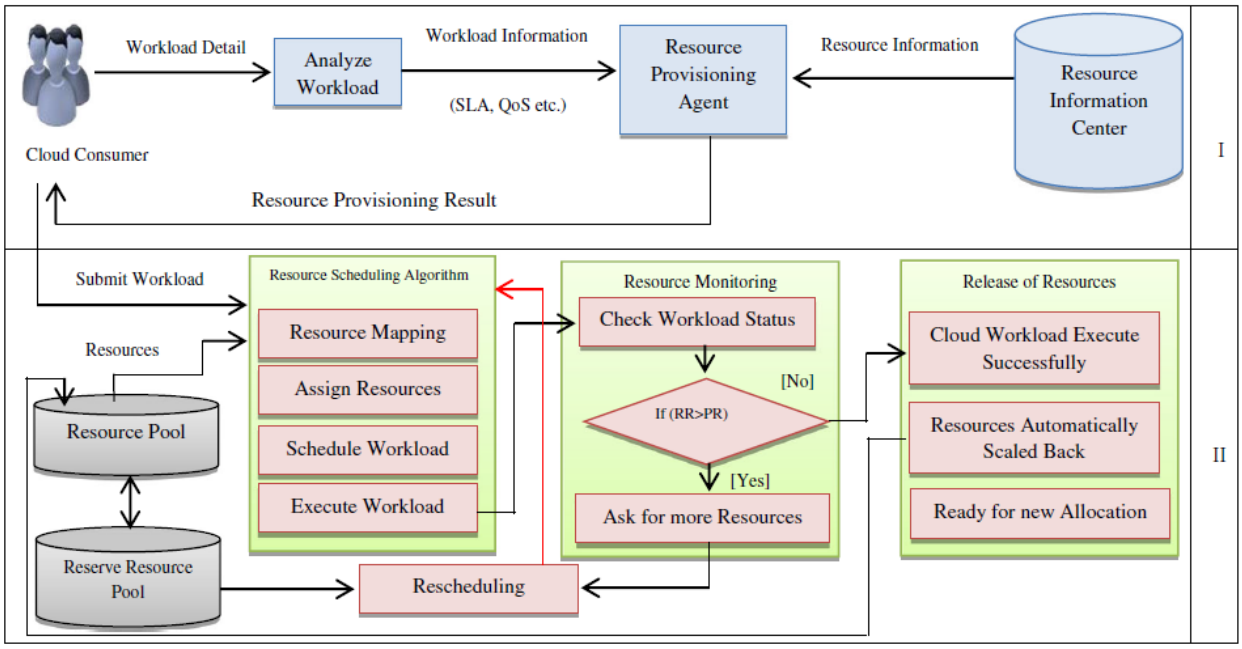

Figure 3.7: Resource Provisioning and Resource Scheduling in Distributed Data Centers

Resource Scheduling

Concerning resource scheduling, in distributed data center scenarios, we note the challenges of dispersion uncertainties as well as the heterogeneity of resources. Otherwise, by referring to Figure 3.7 we note that the process itself comprises three functions namely, [34], [35]:

- Resource Mapping,
- Resource Execution
- Resource Monitoring.

This is a process preceded by resource provisioning which we have already discussed in the previous subsections.

The scheduling is accomplished in steps as follows:

- Initially, the end-user submits his/her job for execution.
- This is followed by appropriate mappings of the submitted jobs and required resources. This must be in adherence to the QoS specified in the SLA. Concerning QoS, metrics such as memory and CPU utilization are taken into consideration. The resource execution will utilize appropriate resources to accomplish the workload as soonest. In that way, the services and applications are utilizing the availed resources efficiently. Consequently, the result is that it is energy efficient. In cases where Required Resources (RRs) exceed actual availed or provided Resources (PRs), more can be allocated. In

this case, they will be acquired from a Reserve resource pool. The resources are released and availed to the next workloads immediately after job completion.

SDN-based Network Resource Scheduling

In the advent of SDN implementation, we briefly look at Network Resource Scheduling theory when SDN is implemented in distributed Cloud data centers. It is generally noted that SDN has a global view of the entire distributed network hence we can take advantage of that to fully optimize both resource allocation and scheduling. In that way, we will be able to achieve v full utilization and desired multi-objectives Provided in Figure 3.8 is a scheduling reference architecture, [36], [37].

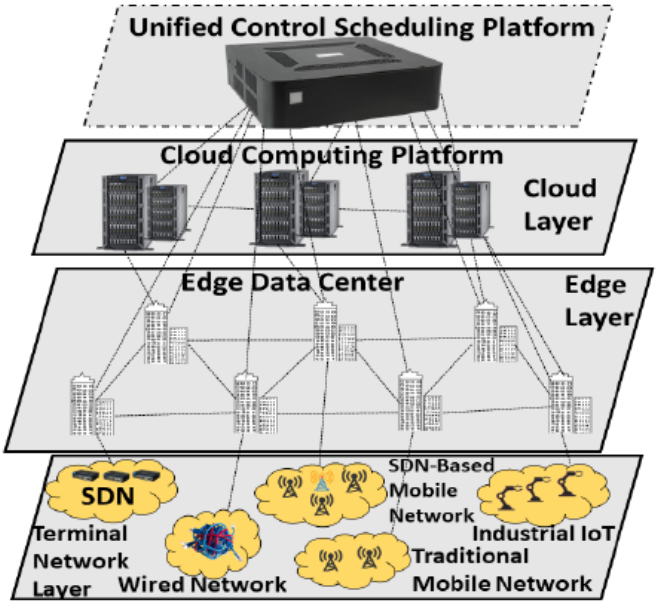

Figure 3.8: Hierarchical Universal Scheduling Theory Architecture

More details regarding the model can be found in [67]. In a nutshell, there are plans to adopt a hierarchical cloud edge differentiated resource scheduling for different cloud services and edge services. In this regard, a few steps are to be adopted and specifies as follows:

- to perform service awareness and determine the service coverage;
- to be in line with different business scopes, to perform network resource scheduling at the level of the service coverage along with the following levels, to be friendlier to the end-user, without occupying too many network resources outside the business scope.

- to identify the type of network service and the type of data to be forwarded, and to perform automatic hierarchical routing according to the data type.

Resource Scheduling Evolution

As discussed previously, Resource scheduling in distributed data centers is a fairly intricate task as the dimensioning in the form of scheduling of appropriate resources to the entire data center cloud is a function of the requested QoS requirements of the services and applications. In such a scenario uncertainty coupled with heterogeneity, as well as decentralized computing resources (distributed data centers) encounters problems of appropriate and optimal dimensioning of available resources, which cannot be effected with current resource allocation approaches.

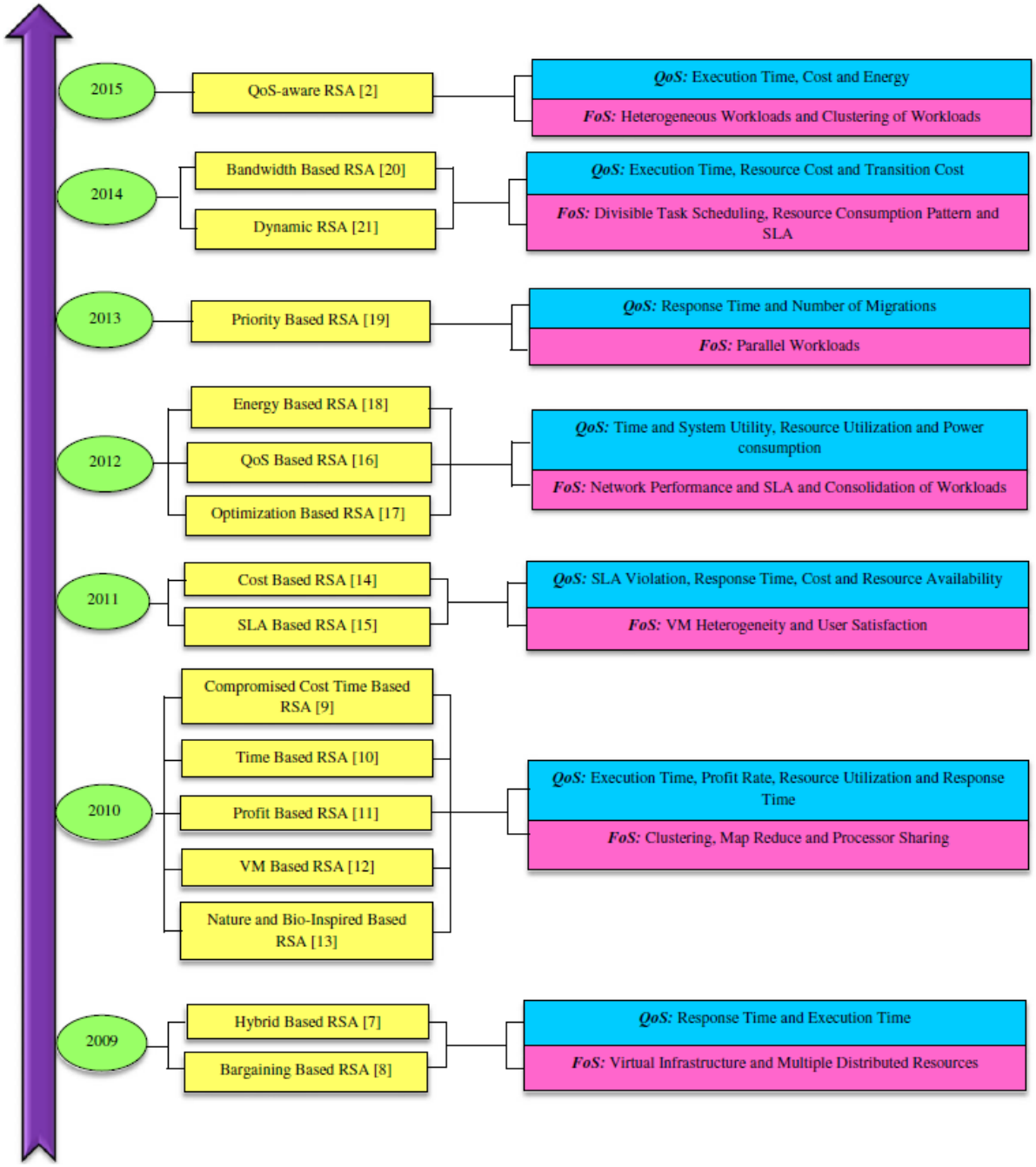

Figure 3.9: Evolutionary Resource Scheduling Path

So the inevitable ever-changing resource scheduling path will focus on describing the QoS parameters. A summary of the evolutionary path based on extensive research is provided in Figure 3.9.

3.5. Chapter Summary

The chapter is devoted to scheduling given that as a key integral part of facilitating (enabling) energy efficiency in data centers. We surveyed energy-efficient scheduling algorithms. It is noted that the designing of a new generation of scheduling algorithms that are energy efficient was triggered by a rise of data center-associated workloads These will aim to appropriately dimension the available data center resources subject to certain objectives that include but are not limited to maximizing energy efficiency, maximizing utilization, minimizing live VM migration, preventing QoS degradations of jobs already being served. In terms of classification, we categorize them into global, local, centralized, distributed or hierarchical, static, and dynamic.

4. Data Center Paradigms

4.1. Overview

In this chapter, we focus on the operation of an energy-efficient data center system. We will partly mitigate the choice of configuration and operation procedures. We also will present an Edge-Fog cloud architecture, which we believe would be ideal for attaining the goals of overall cost-effectiveness and energy efficiency[38].

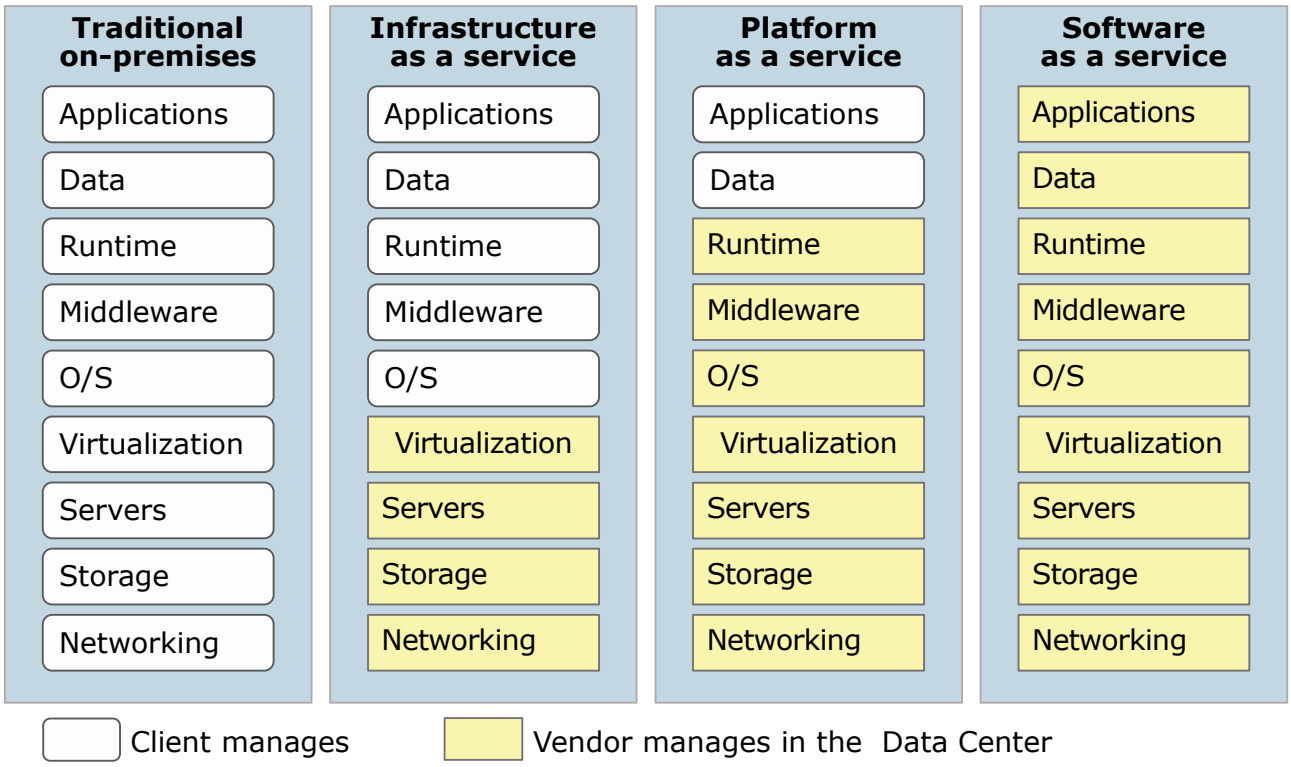

Figure 4.1: Split of Responsibilities: Provider-Side and Consumer

The implementation of appropriate resource allocation Fig 4.1, as well as scheduling algorithms, would also be dealt with taking into consideration the multi-objective goals that have to be met. We further discuss and provide insights pertaining to features and characteristics that influence its overall performance.

4.2. Architectural Design of Various Paradigms

Both design and operation trends in Datacenters architectures are evolving rapidly and thus constantly changing. The main drivers for the demand for data center-based computing include, [39, [40]:

- The emergence, as well as surging numbers of IoT, end terminals (devices).
- Demands for lowered latencies and jitter for Services and applications.
- Need for faster turnaround times hence minimized transmission delays (bandwidth issues).
- Constrained nature (in terms of computing capabilities and power) of IoT and other end-user devices.
- Energy savings at the network's ingress sections in comparison to the cloud.
- Security and privacy threats for centralized data centers.
- The emergence of the 5g network.

With regard to an appropriate data center architecture, it is first necessary to explore the various existing paradigms. Ultimately we will propose an architecture to which we will apply both resource allocation and scheduling algorithms (schemes).

4.3. Edge Cloud Paradigm Scenarios

We devote this section to discussing various architectural paradigms, which will help in improving overall QoS. In particular, the QoS metrics of interest are e.g end-to-end delays(latency), delay variations (jitter), and transition delays, [38].

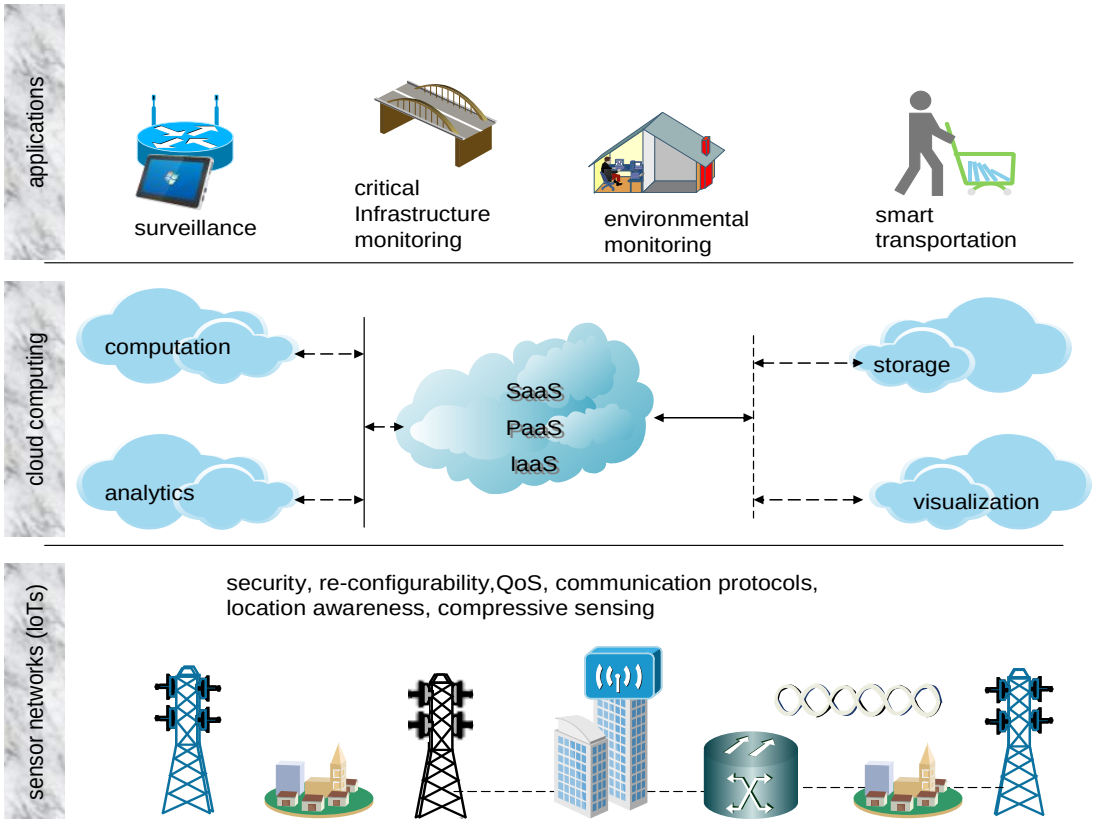

Figure 4.2: Cloud as Middleware in an IoT Paradigm

Typical cloud computing characteristics are: -

- *On-Demand self-service:* i.e, the ability to render users instantaneous access, to computing resource requirements (e.g. CPU time, storage space, network access etc.) without requiring any human interaction with the provider of those resources.

- *Network Access:* Such requested resources are deliverable through the IoT-enabled network and accessible to several clients as well as client applications with diverse platforms requiring standard protocols and mechanisms to access them.
- *Resource Pooling*: The available resources are pooled together to serve many customers concurrently utilizing various dynamically assigned physical and virtual resources to satisfy customers' QoS expectations. This "multitenancy" model relies on the use of virtualization and in that way, IT resources can be dynamically assigned and reassigned, according to demands.

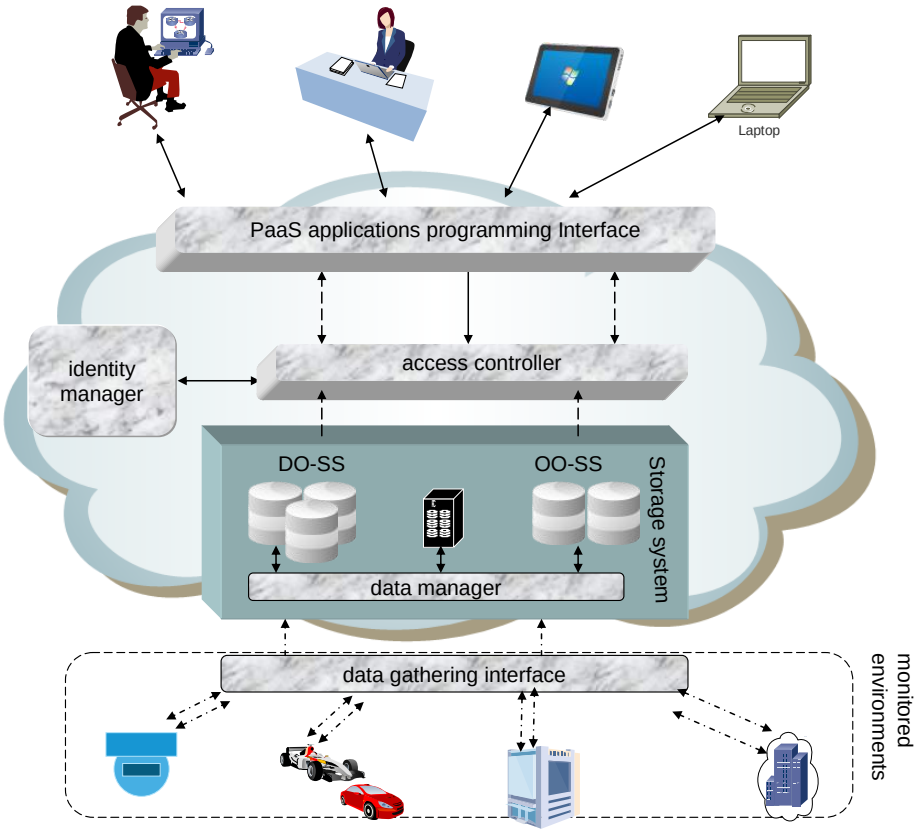

Figure 4.3: Cloud Storage System

- *Rapid Elasticity*: The service provisioned by a cloud provider is elastically deployed, assigned, released, or scaled as per demand.

- *Measured Service*: The ability of the cloud service to monitor and measure actual individual usage and charge fairly. In terms of infrastructural deployment within the IoT context, four models exist, and these are [19]:
- *Private Clouds*: This infrastructure is provisioned to an individual organization so that it restricts access and usage of the services it avails to employees.
- *Community Cloud*: This is an infrastructure of a community that shares a common goal *Public Cloud*: Such an infrastructure's services are provisioned for open use on a pay-per-use model.
- *Hybrid Cloud*: In this case, the infrastructure blends two or more distinct infrastructure deployment models.
- *Inter-Clouds (Cloud Federations*): This is a relatively newer cloud provisioning model that offers more flexibility, as well as improved reliability and geographic distribution.

Depending on cloud services that are renderable by cloud providers, three service models are specified. These differ in control granted to requested resources by a user as well as, the general functionalities and the architectural layer offered.

- Software-as-a-Service (SaaS): In this case, the users rent out their applications via a service provider.

- *Platform-as-a-Service (PaaS):* This is primarily a development platform that is provisioned to customers to develop their proper applications or services.

Infrastructure-as-a-Service (IaaS): The users are allowed direct usage of the IoT infrastructure. This includes processing, storage, and network resources. In practice, this is implementable through virtualization techniques.

4.4. Criteria for Comparisons

Given that selection of one paradigm over the other requires careful consideration, we thus list down a set of criteria that designers can base on in choosing a particular paradigm. Such criteria include similarities versus differences, [38, [41].
The similarities are:

- End-to-end delays (latencies)
- Topology design (architectural design)
- Support for mobility
- Degree of availability
- Scalability and flexibility

The differences include:

- Adminstrive ownership
- hardware
- privacy and security
- proximity to end users.

4.5. Example Data Center Paradigms

Cloudlets paradigm

By definition, a cloudlet is a trusted, mini data center or computing facility, that is connected to the IP network and readily available for use by GSM devices in proximity (within range) Figure 4.4[38].

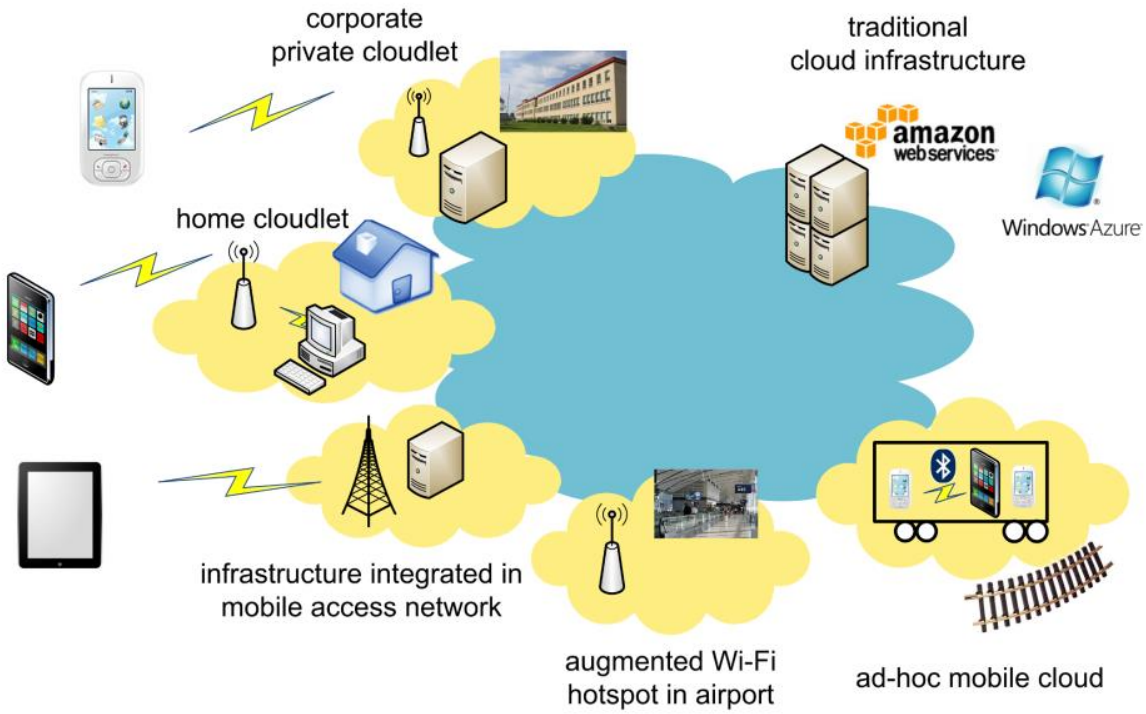

Figure 4.4: Example Cloudlest

The Cloudlest network is normally connectable via single-hop. It is normally self-manageable. There exists multi-hop and mobile Cloudlest as well.

Edge-Centric Data Center

Edge data centers are relatively small and located close to the periphery of a network. Typically, they are housed in a smaller area. Key characteristics include proximity, intelligence, trust, control, and human-centric design as shown in Figure 4.5[41].

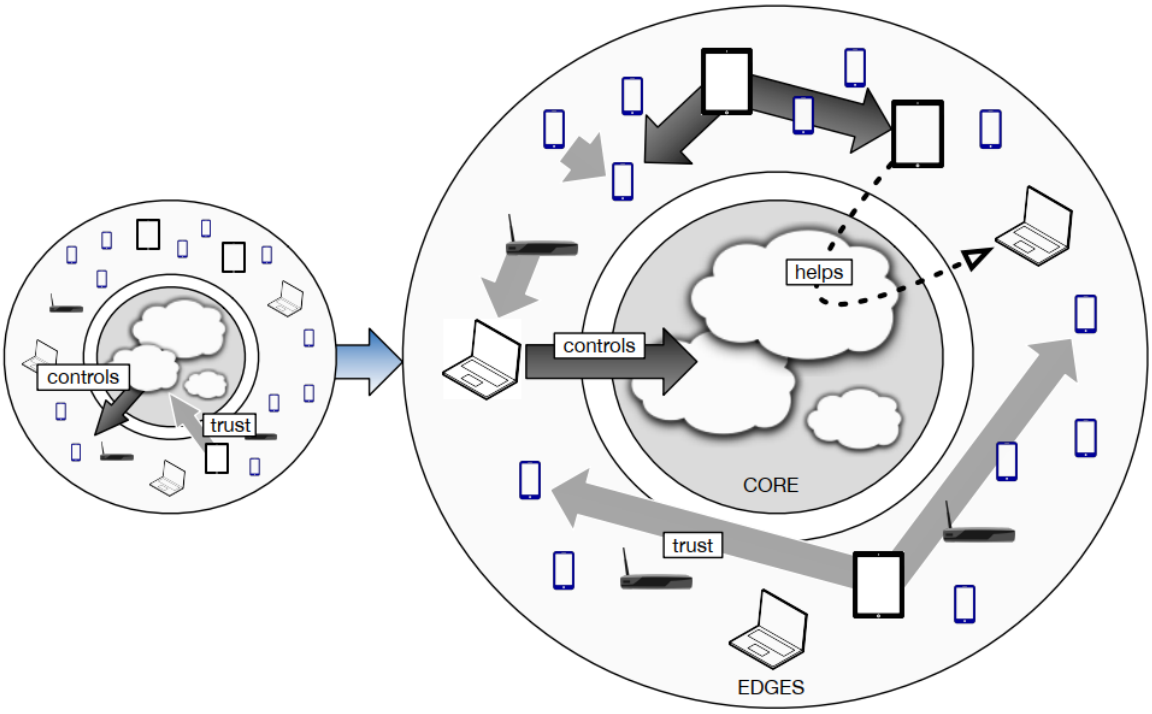

Figure 4.5: Comparing Cloud Computing Versus Edge-Centric Computing

Mobile Edge Data Center Paradigm

Mobile-edge Data centers primarily provisions cloud-based computing services or rather capabilities of cloud computing within the locality of a RAN. In that way, the end users are in proximity. Figure 4.6 shows the key components include GSM mobile and IoT-enabled devices. Its key characteristics are as follows, [38, [41]:

- Proximity. It has support for D2D communication. It has high proximity this is attributed to the excellent wide spread of servers, coupled with good connectivity.
- Geographical Distribution. Its servers are geographically distributed as dictated by the ETSI Standards. It readily avails a programming platform and thus makes it quite popular with application developers.
- Low end-to-end delays (Latency.): It has much lower latencies in comparison to other cloud paradigms. This is because its servers are closely placed

- Location Awareness. Its servers are situated next to or within base stations. It becomes easier to locate the developers. User location information can be used to design better resource offloading and mobility management algorithms. The servers easily map current and future mobility patterns. In that way, it becomes relatively easy to predict the future mobility patterns of the end-users.
- Network Context Information. Its servers store network context information.
- Distributed content delivery and caching. It has the advantage of facilitating good video streaming data distribution. Typically the video requires high bandwidth for downloading to end-users. Storing the video in the servers thus saves on bandwidth because of the latter's proximity to the end-users.

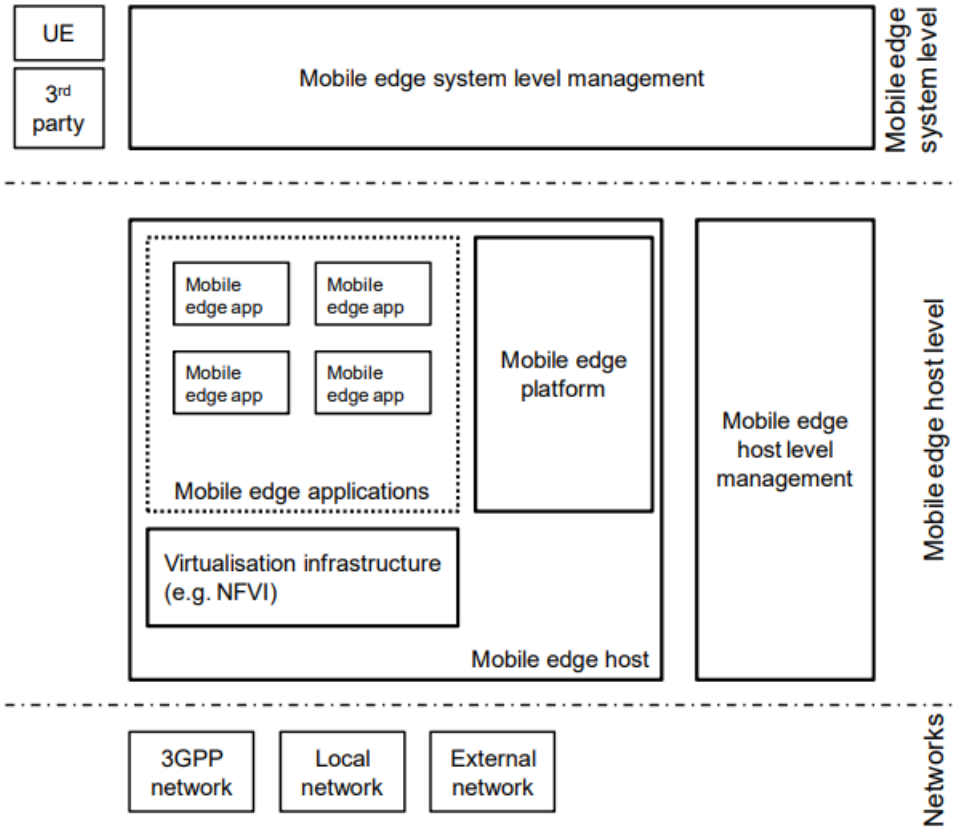

Figure 4.6: MEC Framework

- Web performance enhancement: The proximity of the servers means web surfing is relatively swift.
- IoT and Big Data applications: Can partially data requests for IoT-enabled devices

Fog Data Center Paradigm.

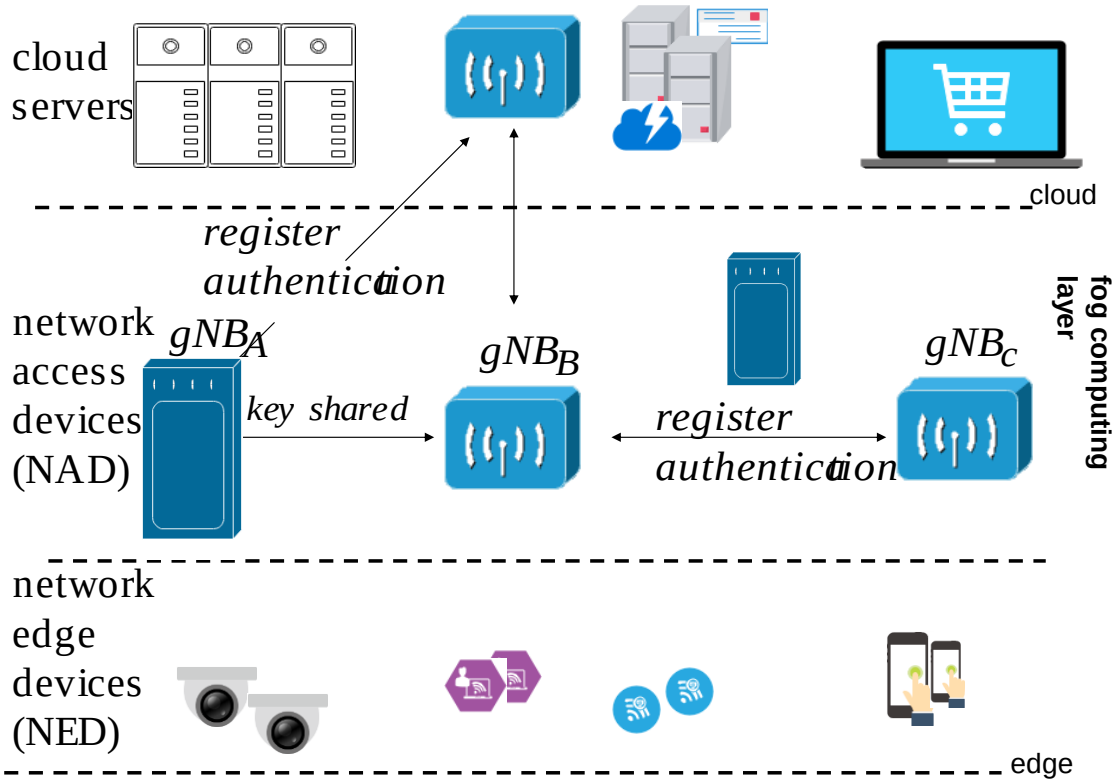

Figure 4.7: Fog Data Center Paradigm Alternative

Despite the distances involved between end users and the central cloud servers, the elasticity nature of the resources offered by this paradigm ensures high QoS . Figure 4.7 illustrates such as structure., whereas Figure 4.8 exemplifies the fact that certain duties such as security and authentication can be delegated to the Fog layer, which itself is in proximity to the end user. In that way, a user would not have to wait long to be confirmed access to the network resources.

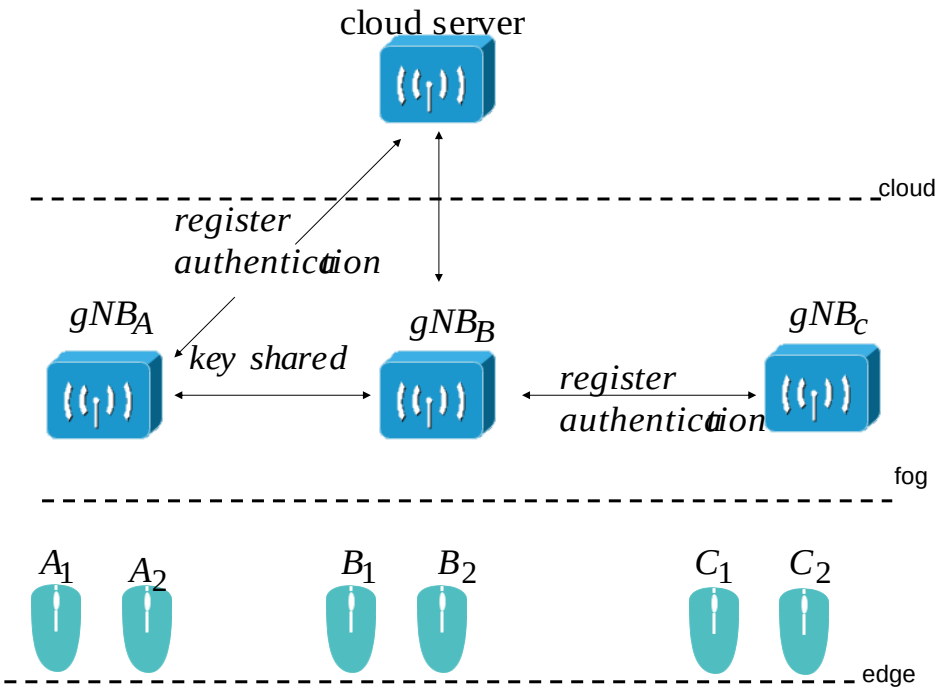

Figure 4.8: Authentication Delegation at Fog Layer

Besides privacy and security checks closing the user level means fewer risks to attackers by adversaries. E.g it is always desirable that the authentication process be done within the confines of the customer premises, i.e without having to involve long distances as this otherwise would unnecessarily increase that attack surface.

Edge-Fog Data Center

Ege-centric cloud and fog cloud paradigm models can be merged to form an Edge-fog Data center. Edge-fog cloud is a hybrid of e. Figure 4.9 illustrates the edge-fog cloud. It consists of the following layers

- edge level layer
- fog level layer,
- Datastore level layer.

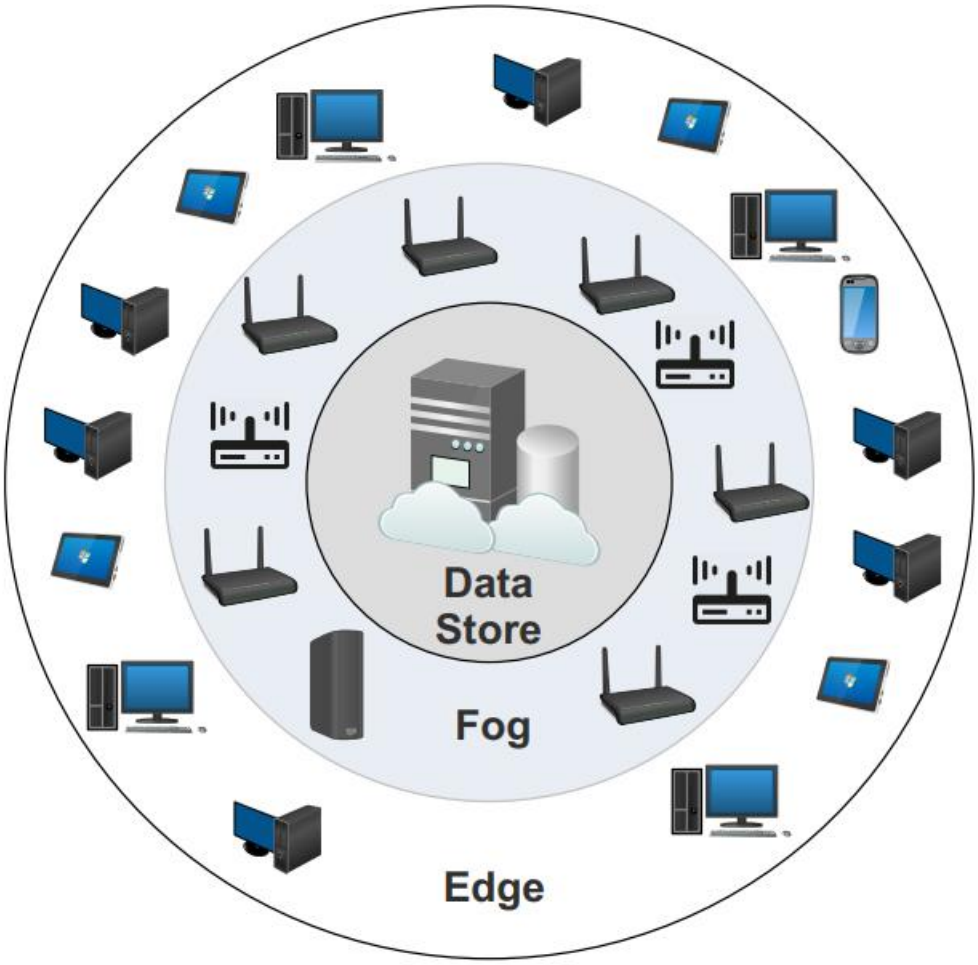

Figure 4.9: Edge-Fog-Cloud Data Center

Edge Layer. It comprises multitudes of mainly nano data centers, and other end devices. IoT-enabled terminals can also connect to the edge and thus creating a single or two-hop type network. In that way, the requirement for the proximity of edge resources to devices s maintained.

Fog Layer. This is a layer above the edge layer. Normally comprises network elements such as L3 switches computing resources. It assists the upper layer (cloud) in performing some of the computations or security-related checks. It also functions as a middleware or backbone in an edge-fog cloud model.

Data Store. This is a data storage facility for the overall cloud. e data store layer stores the data in the cloud. Overall it's a common data repository for edge and cloud

Mobile Cloud Computing

With this paradigm we have GSM operators/base stations forming a mobile network layer> the operators handle privacy-related issues such as authorization, as well as accounting services. Hence directly interface with end-users. Typically this service is handled by a Home Agent (HA)[42].

Access to cloud services is the responsibility of the cloud service provider layer (an upper layer). This includes granting access to infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS). The main difference between the cloud and mobile cloud data center paradigms is that in the former, the servers are placed nearer the GSM mobile users. In that way, the computing resource-constrained mobile terminals rather offload the computations to the nearby servers instead.

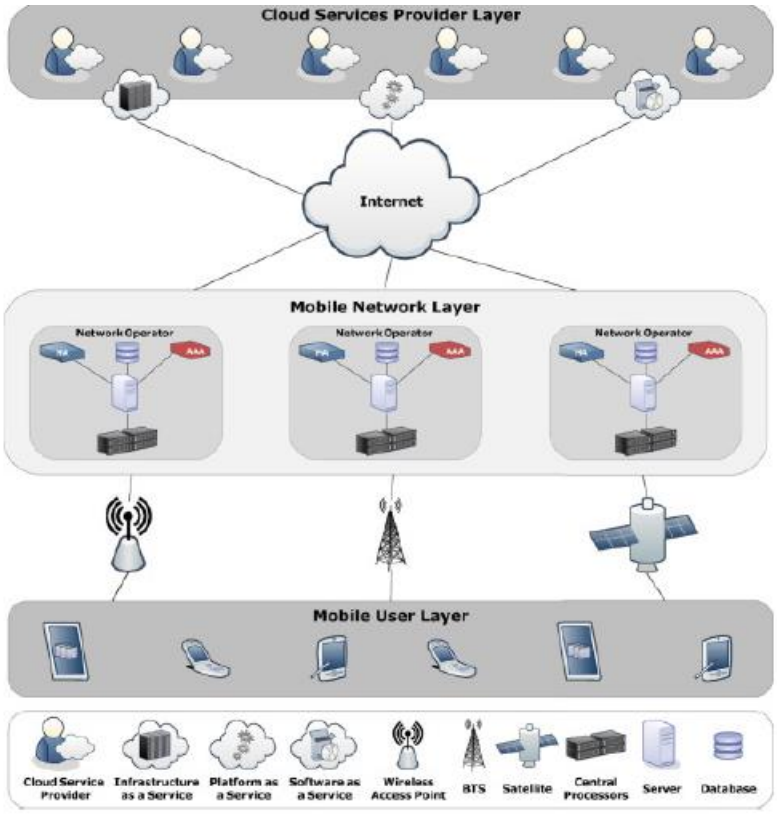

Figure 4.10: Mobile Cloud Computing Data Center Paradigm

- Overall the mobile computing paradigm has advantages such as [38, [40, [42]:

- Battery life extensions, as the servers are in proximity hence communication ranges are quite short.
- improved processing power as well as data storage capacity for the end terminals (GSM mobile devices).
- Improved robustness as well as reliability.
- dynamic provisioning of resources capabilities.
- scalability,
- Improved degrees of integration
- multitenancy.

A few challenges with this paradigm are still being addressed. These include:

- abilities to offload in a mobile environment.
- costing analysis and benefits thereof.
- issues around mobile movements (mobility) and their management.
- trust, privacy, and security.
- overall data storage, processing, and administrative management.

Superfluid Cloud Data Center Paradigm

This is a paradigm in which multi-tenant, virtualized software-based services run on common, shared commodity hardware infrastructure deployed throughout the network. The concept is illustrated in Figure 4-11.

The rationale is for the creation of a relatively low-cost virtualized platform mainly for leasing to third parties. In that way, leased resources act as microdata centers run mainly by Telecommunication Operators, [43], [44], 45].

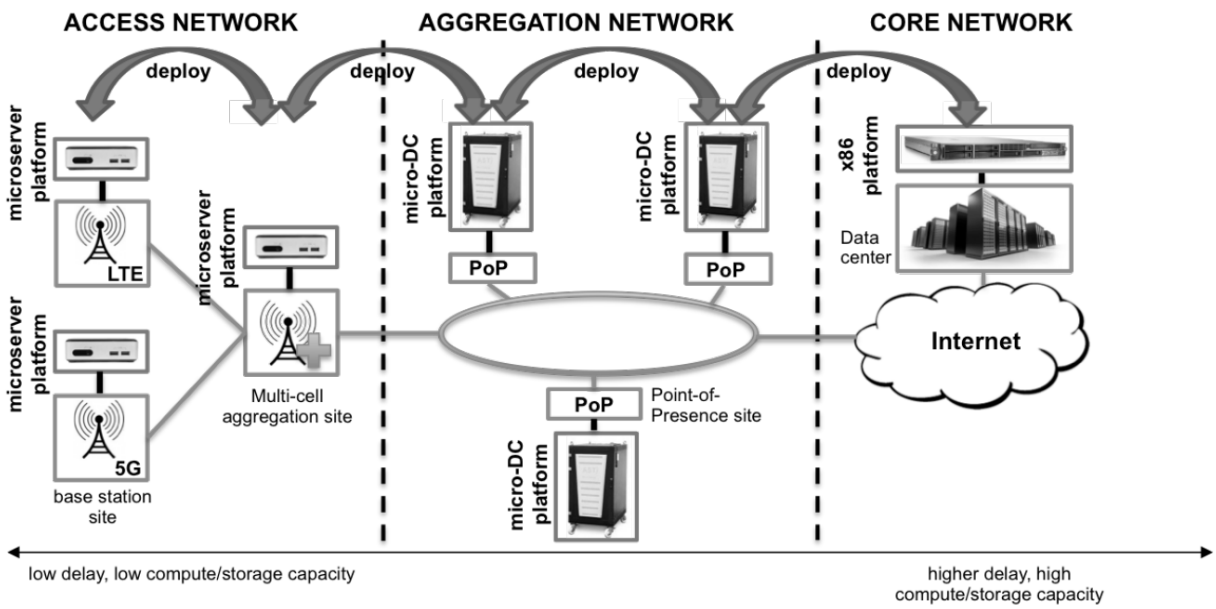

Figure 4.11: Superfluid Cloud Architecture Paradigm

As can be further noted from Figure 4.11, the architecture comprises three network sections namely the access, aggregation, and core. The access network section is characterized by low latencies as well as limited computing and storage capacities. However, this works well for real-time services and applications. As opposed to the access layer, the core network has considerably higher delays but unlimited computing as well as storage capacities, [45].

4.5. Chapter Summary

The chapter focused on the operation of energy-efficient data center systems. We also discuss the choice of configurations and operation procedures. Various paradigm architectures are overviewed. The focus is on mitigating an architecture that would best promote overall cost-effectiveness and energy efficiency.

The implementation of appropriate resource allocation, as well as scheduling algorithms are further discussed taking into consideration the multi-objective goals that have to be met. Insights pertaining to features and characteristics that influence its overall performance are provided.

Given that selection of one paradigm over the other requires careful consideration, we thus list down a set of criteria that designers can base on in choosing a particular paradigm.

5. Resource Provisioning and Scheduling in Fog Cloud Data Centers

The chapter devotes to discussing key drivers for achieving energy efficiency. In this regard, we assume a distributed Fog Cloud data center paradigm. The key to achieving energy efficiency in its operations would be sound load balancing among the active servers as well as appropriate scheduling. The latter is necessary to ensure that we maximize of the completion jobs in the shortest time possible using the minimum possible resources. In that way, much less power will be consumed.

5.1. Introduction

The emergency of IoT and associated services has resulted in more requests coming through the Fog layer. As the Fog layer is in proximity, it readily supports real-time interactions for several applications and services. As the number of requests further ballon, so is the VMs at this layer becoming clogged. This necessitates a redistribution of jobs. The load balancing mechanism can distribute load among all the VMs in equal proportions. Figure 5.1 shows the exponential growth of devices and objects for the next decade, [90].

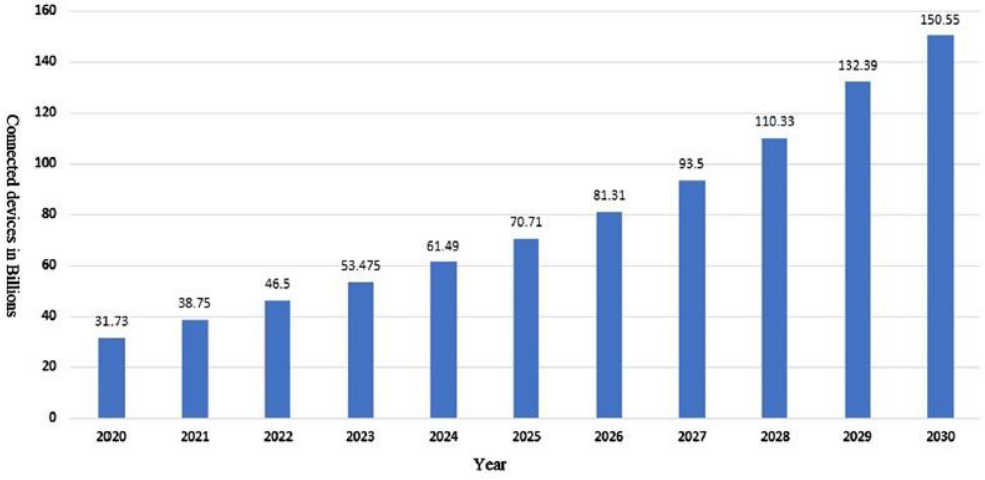

Figure 5.1: Exponential Growth of IoT Enabled Objects and Devices, [90]

Power consumption by these devices likewise also increasing. Most of their data is ultimately processed and stored in Cloud data centers. Notably, the Fog layer supports low latencies, but the Cloud data centers are not. Hence the Fog–Cloud Datacenter paradigm will be ideal to support both real-time and non-real-time services and applications. All received jobs have to be distributed rationally throughout. The load balancing can be fixed, dynamic, or adaptive. Note that appropriately balancing the load would result in enhanced resource utilization as well as minimized energy conversation.

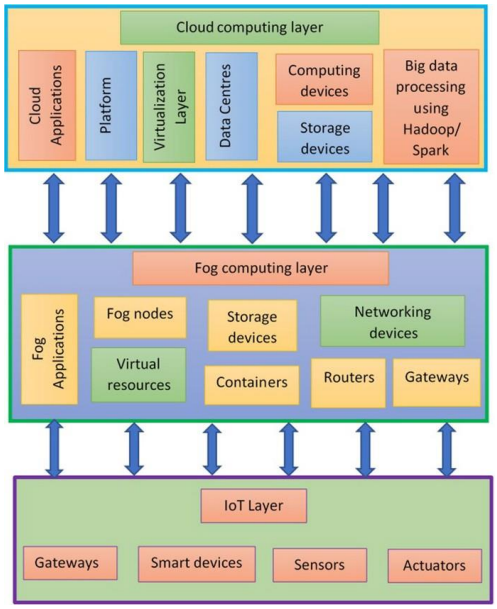

Figure 5.2 : Summarised Fog-Cloud Data Center Architecture

Figure 5.2 illustrates the main layers of a Fog-Cloud Data center, [91]. Key issues to be addressed in such an architecture include data management, security as well as resource provisioning. Resource provisioning also incorporates scheduling.

- **Resource management:** The system has to take into account the requested QoS specified in the SLA and ensure that adequate resources are allocated. QoS consistency is mandatory or else end users would migrate or subscribe to rival data centers instead. The resource allocation itself faces challenges of heterogeneity, failures, and dynamism. A typical; resource management system will provision resources statically or dynamically. In that way, overall system performance will be enhanced at the same time while maintaining energy efficiency.

- **Resource scheduling:** Resource scheduling is also key to improving resource utilization and the overall performance of the Fog-Cloud data center.
- **Load balancing:** This is necessitated by the need to distribute the workloads evenly. Note that should servers in one part of the system malfunction, their load will be relayed to any other available servers. In so doing the load balancing is executed such that it does not congest the active servers. The load balancing guards against over-utilization or underutilization. Overall it assists in achieving overall maximized resource utilization as well as power efficiency.
- **Data management:** This is mainly for data administration, i.e data has to be validated at the Fog layers before it is relayed to Cloud data center servers for further processing.
- **Security:** This is to ensure privacy data integrity and confidentiality as the data is transferred (transmitted) from one entity to another.

5.2. Dynamic Load Balancing Technique

Load balancing applies to both physical servers and VMs. Associated algorithms can either be initiation process-based (sender or receiver-initiated) or current state-based (static or dynamic) as shown in Figure 5.3. The first type is further divided into three types, i.e. sender initiation based, [92].

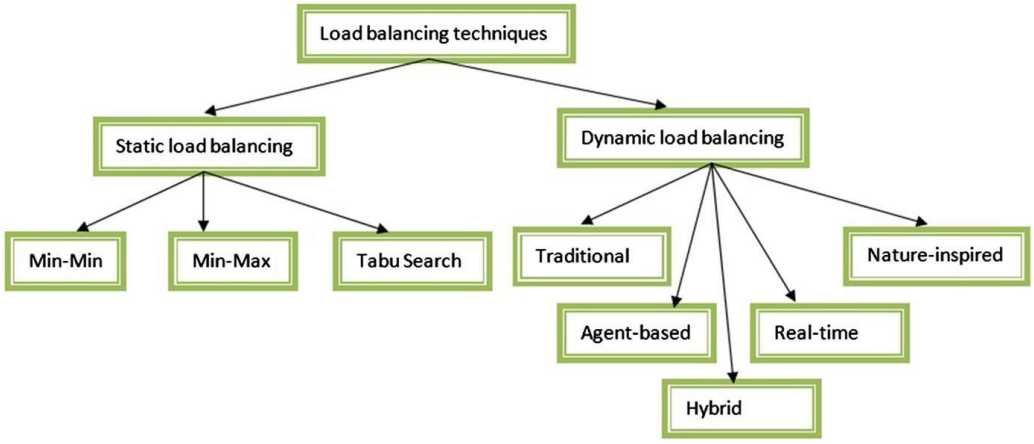

Figure 5.3: Load Balancing

Static load balancing has since been proved to be quite inefficient and hence we will only review dynamic load balancing, Various categories of dynamic and adaptive load balancing techniques have been explored. These include traditional, nature-inspired, agent-based, real-time as well as adaptive/hybrid load balancing. We define each of these as follows:

- **Traditional load balancing:** This encompasses various techniques. Examples include the breadth-first search (BFS) algorithm which operates mainly at the Fog layers. Another example of techniques in this category is dynamic resource allocation (DRA) which provides equilibrium in load balancing among servers running at varying speeds. In short, the algorithm will prefer a faster-running server when allocating tasks. In that way, load balancing is achieved. A real-time efficient scheduling algorithm (RTES) was also proposed to balance loads by way of executing all real jobs before their time outs. In that way, the turnaround times experienced by end-users would be very minimal. The throttled algorithm is another example of traditional load balancing that assigns tasks to VMs based on the latter's capabilities and size. All active VMs are indexed as either busy or idle. When a new job arrives an assigned manager will search the index table to determine which VMs are available and allocate assign the jobs to them, [92].
- **Nature-inspired load balancing:** An example is the ant colony algorithm, which generally brings about a reduction of the makespan and thus consequently balances the load. The honey bee behavior-based load balancing (HBB-LB, external optimization (EO), and ant colony optimization (ACO) are also example techniques in this category.
- **Agent-based approaches for load balancing**: This operates on a client-server style basis in which a load balancer is fed with load levels of each active server via agents. The agents themselves are incorporated into each server and relay the information in real-time to the load balance. The load balancer utilizes this information when assigning workloads to servers. Example algorithms in this category include agent-based automated service composition (A2SC).
- **Real-time load balancing**: This is a category of algorithms that gear themselves towards minimizing latency and execution times. Examples include the real-time efficient (RTES) algorithm which tends to force the Fog layer ends to give priority to real-time tasks so that

they can be served and completed before timeouts. Other example algorithms in this category include round-robin (RR first come first serve (FCFS).

- **Adaptive/ Hybrid balancing:** These will generally incorporate one or more of the categories already discussed in a bid to achieve good load balancing, [93].

5.3. Energy Efficient Load Balancing Framework

In this section, we propose and analyze a framework for load balancing in a Fog-Cloud Data center. In so doing we assume that the data center architecture is 3-layered ,i.e it comprises the end user, Fog, and Cloud layers. As alluded to before the Fog layer is dominated by resource-constrained terminals and as such provides a processing platform for real-time jobs only. Otherwise, the rest of the tasks are relayed to the Cloud layer. Because the arrival volumes as well as patterns are unpredictable, it becomes necessary to address load imbalances as soon as the need arises. An example is that at each layer, should some VMs be overwhelmed, then some of the load is diverted instead to the less loaded VMs. In formulating our problem we take into cognizance that:

- The end-user layer comprises millions of IoT-enabled and other terminals that frequently generate huge volumes of data that require processing either in real or non-real-time frames.
- Nodes at both layers (Fog and Cloud) of the data center will utilize lesser power during periods of idleness in comparison to periods of activeness.
- A desire thus arises for a need to design an energy-efficient framework that will drastically reduce power consumption within the data center, i.e at all nodes in both the cloud and Fog layers.
- That frequently the volumes of requests from end users may be overwhelming such that the aggregated Data Center resources also become overwhelmed hence overall QoS degrades.

- The intermittency nature of request arrival together with the heterogeneity nature of the workloads thus provides a case for designing an energy-aware load balancing mechanism that will serve both layers.

Figure 5.4 is representative of such a framework. As can be seen from the same figure, three layers are distinguishable. These are the end-user, Fog, and cloud.

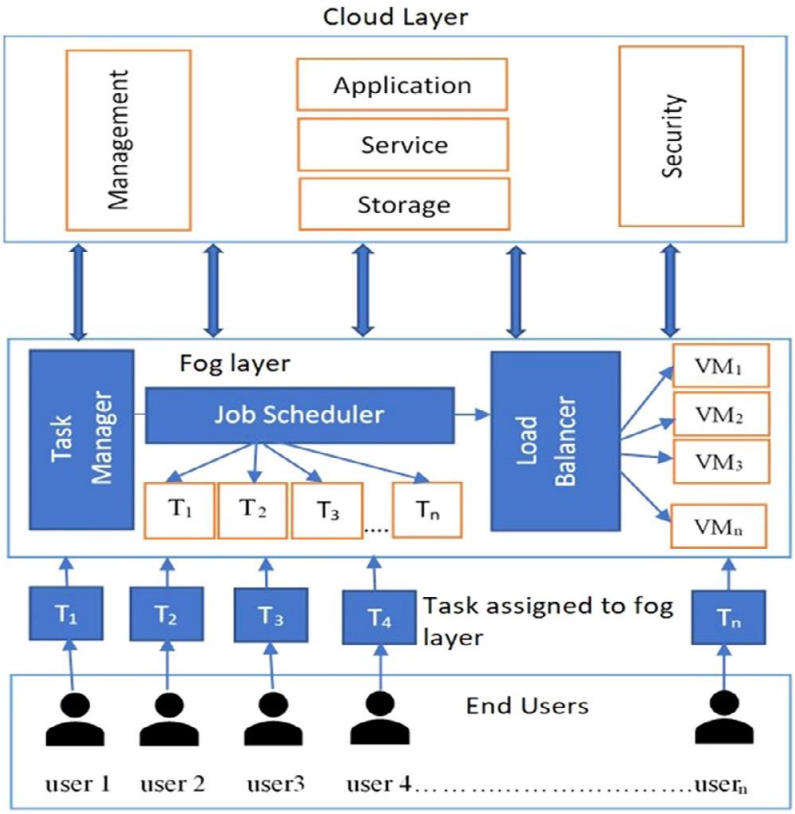

Figure 5.4: Load Balancing Framework for Fog Computing.

The layers are summarily defined as follows:

- End-User layer: This is the main origin of all requests. Note that an end-user request can also be in the form of requests coming from a neighboring data center. The end user layer connects to the cloud layer via the Fog. Depending on the resource requirements as well as the time-sensitive nature of a given request from this layer, it can be processed at the Fog layer or relayed to the Cloud.
- **Fog Layer**: The layer is mostly defined by Fog nodes designed to receive, and authenticate, as well as processes originating from the user-end layer. Depending on the

resource and QoS requirements, this layer may immediately process and post back the results or may refer the request(s) to the upper (cloud) layer. It is at this same layer that scientific workflow tasks are mapped directs to its nodes. The incorporated task manager module coordinates the handling of tasks based on priority such as on a FCFS basis. Likewise, the scheduling will be implemented on a priority basis by the Scheduler module. The scheduler will help to set the priority of execution of the tasks. To prevent some VMs from being overwhelmed whilst others are idling, a load balancer carefully manages the process by routinely evenly spreading the newly arriving jobs (tasks). This also consequently reduces both power consumption as well as CAPEX and OPEX.

- **Cloud layer**: The layer is equipped with high computing and storage capabilities. All data requiring long-term storage may utilize this layer. In addition services such as PaaS. SaaS and IaaS together with security are provisioned at this same layer.

5.3.1. The Balancing Algorithm

We assume the Data center is distributed and hence has many interconnected processing nodes (PNs), such that the entire Fog-Cloud Data center system is defined by:

$$PN = \langle PN_1, PN_2, ...PN_n \rangle \tag{5.1}$$

Likewise, each PN comprises several parallel servings VMs;

$$VM = \langle VM_1, VM_2, ...VM_n \rangle \tag{5.2}$$

The user layer is an originator of so many requests (Rs), i.e defined by;

$$R = \langle R_1, R_2, ...R_y \rangle \tag{5.3}$$

The aggregate number of requests within a given measured interval would be;

$$R_{aggregate} = \sum_{i=0}^{y} R_y \tag{5.4}$$

To sustain the workload, the associated costs are determined by summing the cost of each VM, hence we have;

$$C_{total} = C_{\cos t_{VM}} + C_{\cos t_{DT}} + C_{\cos t_{storage}} \tag{5.5}$$

where in the last equation;

$C_{cost_{VM}}$ -is the cost of running VMs. Note that the physical servers are also incorporated as the earlier is implemented in a physical server.

$C_{cost_{DM}}$ -costs associated with data transfer. This can be locally or between the Fog and cloud layers.

$C_{cos_{storage}}$ - data storage-related costs.

The aggregate workload (WL) is determined from;

$$WL_{total} = \sum_{i=0}^{n} r * \left\langle \sum T_{VM} \right\rangle^{*} f \tag{5.6}$$

where , T_{VM} is the duration of task completion per VM at frequency (f).

The aggregated power consumption is approximated from;

$$E_{k_{flow}} = E_{b_{flow\_}} N_{bit_k} \tag{5.7}$$

In which case the variables contained in the last equation are defined as follows;

$E_{b_{flow\_}}$ -is the network's energy's consumption per bit.

N_{bit_k} -denotes total number of bits exchanged among the k services.

A summary of the load balancing approach (algorithm) at both the Fog and Could layer VMs is summarised in Figure 5.5. The algorithm overall ensures that all the resources at both layers are efficiently, rationally, and maximally utilized. In that way, maximum energy savings is achieved. As can be observed from the same figure, it is noted that as far as the workload assignment is concerned, nodes are initially analyzed for their readiness in terms of resource availability and power consumption. Given that not all nodes are active at the same time, then the algorithm will avoid any inactive ones if one or more of the currently active ones can. In this way both power and running costs are optimized.

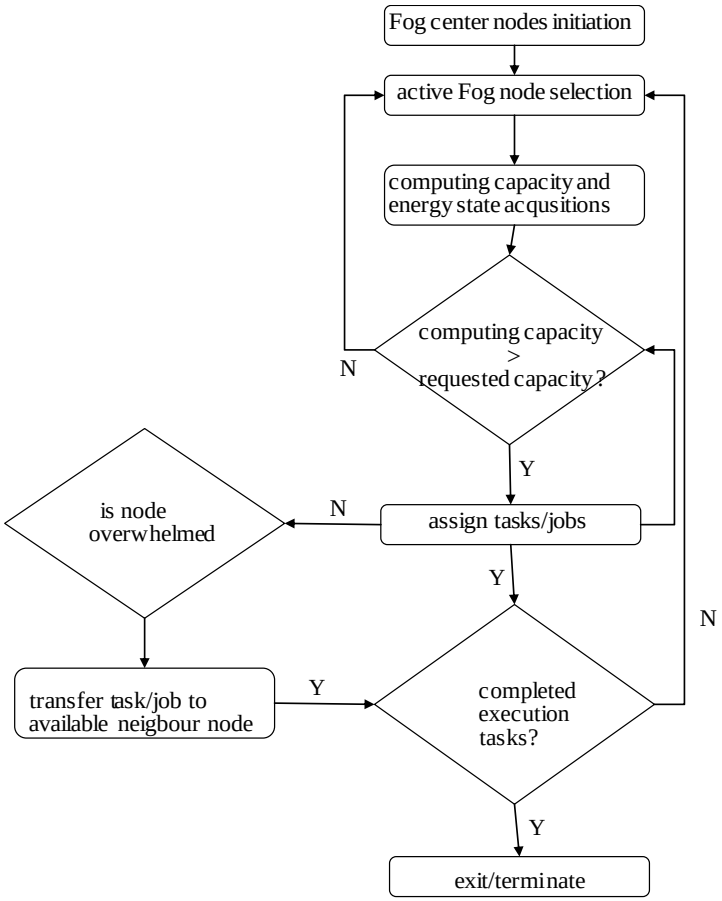

Figure 5.5: Load Balancing Summarised Algorithm

5.3.2. VM Capacity

For each active VM, its capacity versus its rated workload are also compared. We assume that there are three threshold levels of caparisons, high (WL_H), moderate (WL_M) and low(WL_L). If we assume the number of workloads served to a single VM to be $P_{WL_{number}}$ and $P_{WL_{speed}}$ is the number of executable instructions (in millions). per second , then the VM capacity is ultimately determined from;

$$C_{VM} = P_{WL_{number}} \times P_{WL_{speed}} \tag{5.8}$$

By further assuming that independence of the workloads; i.e they are independent of each other are independent;

$$\langle WL_1, WL_2, ...WL_n \rangle \tag{5.9}$$

The VM load is expressed as;

$$VM_{loading} = \frac{\sum_{j=1}^{n} WLL_j}{n} \tag{5.10}$$

where L_j is the workload duration (length).

The VM loadings can thus be compared is phases as follows;

Case I :

$$VM_{loading} < C_{VM} \times WLL \tag{5.11}$$

The above infers that the VM is underloaded and thus can accept for workloads.

Case II :

$$VM_{loading} > C_{VM} \div TL_L \quad \&\& \quad VM_{loading} \leq C_{VM} \times WL_L \tag{5.12}$$

In the case the VM is running in optimum state.

Case III :

$$VM_{loading} > C_{VM} \div WL_L \quad \&\& \quad VM_{loading} \leq C_{VM} \times WLH \tag{5. 13}$$

This is a m fairly highly balanced state.

Case IV :

$$VM_{loading} > C_{VM} \times WL_H \tag{5.14}$$

This is a state signalling that the VM is overloaded.

We can also compute the expected workload completion time

$$E[T] = \frac{WLL}{C_{VM}} \tag{5. 15}$$

The above metric is also referred to as the execution time.

Consequently, the estimated completion time is approximated according to the following equation;

$$ECT = E[T] + VM_{loading} \tag{5.16}$$

Overall, we conclude that in order for a VM to be selected, the following equation must be satisfied;

$$VM_{loading} = VM_{loading} + WLL \tag{5.17}$$

5.3.3. Performance Metrics of Interest

This set of metrics discussed in this section directly dictates the overall performance of the Fog-Cloud data center.

Makespan: This is the total duration (in seconds) required to execute to completion of all workloads in the VM 's existing queue;

$$Makespan = \max\langle CT_i \rangle \quad i \in VMs \tag{5.18}$$

Average Makespan: Note that on average the makespan fluctuates such that the average value would be more practical. This is determined from;

$$\overline{Makespan}[Av] = \frac{\sum_{k=1}^{m} Makespan}{m} \tag{5.19}$$

where m denotes the number of averaged values.

Response times(ART): This is the aggregate time to make the necessary searches as well as execute the workload.

$$RT_i = FT_i - SB_i \tag{5.20}$$

In the above equation FT_i is the completion (finish) time of a given workload and SB_i is its actual submission time to the VM .

The value tends to fluctuate hence Its overall average can be determined;

$$E[RT] \div \frac{\sum_{i=1}^{m} E[RT_i]}{m} \tag{5.21}$$

Execution Times: This is the time-lapse between workload completion time to its executing start time (EST).

$$ET_i = FT_i - EST_i \quad (5.22)$$

5.4. Performance Evaluation

The proposed framework's algorithm is implemented in Cloud Sim, [94], [95], [96], [97]. We first briefly introduce this package next.

5.4.1 Cloud Sim Overview

CloudSim is a Cloud simulator that is extensively used by researchers worldwide. It accomplishes the simulation via its classes and packages. Currently (in Cloudsim 4.0) there are 14 defined packages comprising several classes. E.g the org.cloudbus.cloudsim package is quite popular with researchers evaluating Cloud data centers. The same package defines numerous resource allocation and scheduling policies. Classes defined within the package are categorized into an entity or associated classes The list of core entities includes Cloudlet, Host, Datacenter and VM. The associated classes shown in Figure 5.6 can further be categorized as classes that define policies(VmScheduler, VmAllocationPolicy, CloudletScheduler, UtilizationModel, DatacenterBroker) and classes that specify certain computational requirements(Pe, NetworkTopology, SanStorage, Log, File, HarddriveStorage), [96].

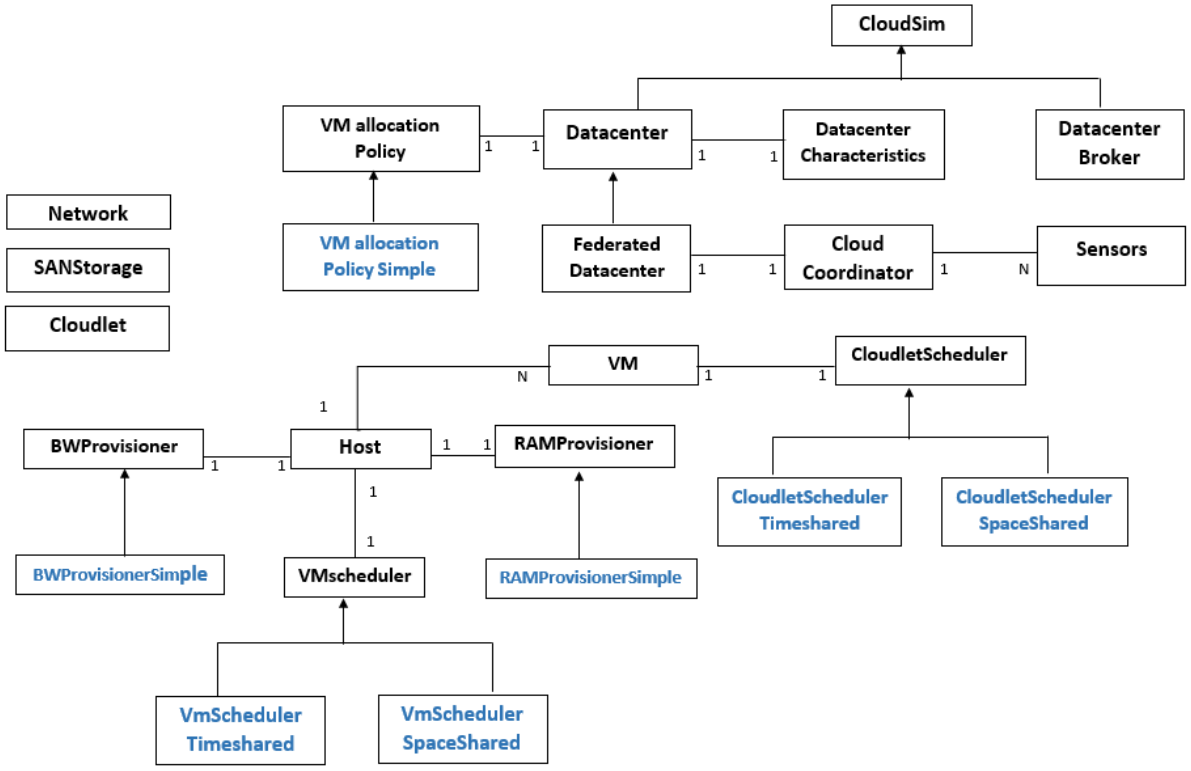

Figure 5.6: Classes in CloudSim

Example other packages in CloudSim include, [96]:

- org.cloudbus.cloudsim.core, which enables the simulation of user end, Fog and cloud layers. Defines FutureQueue , DeferredQueue, , SimEvenet and SimEntity.
- org.cloudbus.cloudsim.predicate- for matching events to entities.
- org.cloudbus.cloudsim.distribution- defines various distributions.
- org.cloudbus.cloudsim.lists –defines various resource related operations lists.
- org.cloudbus.cloudsim.network- for defining network topologies.
- org.cloudbus.cloudsim.network.datacenter- for defining a networking elements within a data center.
- org.cloudbus.cloudsim.power- for fasciliattaing power aware simulations.

- org.cloudbus.cloudsim.power.lists- for providing power VMs to be used in simulations..
- org.cloudbus.cloudsim.power.models- for defining power models
- org.cloudbus.cloudsim.provisioners- used as a bandwidth descriptor
- org.cloudbus.cloudsim.util- provides a list of classes for various specific mathematical functions.

Load balancing Algorithm

initialise

set n ( )

set m( )

setting VM set;
{

$VM = [VM_1, VM_2, ...VM_n]$

initialisi ng workload: $WL = \langle wl, wl_2, ...wl_m \rangle$

while
{

VM = active but not maximally loaded

do

assign work load from list

do

VM = active but not loaded

shift wl to VM_j

next state = desired goal_state?? = YES

return

end if
end while
end while
}

In our evaluation, we make use of two types of VMs each with RAM size of 4 G, HDD sise of 500 G and supported by a bandwidth of up to 1 GBps. For the various workloads we assume that the lengths are variable.. The file sizes are however fixed at 500 MB. Energy consumptions for the two types of servers are tabulated (see Table 5.1)

Table 5.1: Power Consumption for Two Different Servers at Various Loading Levels

server	*idle*	15%	25%	35%	45%	55% 65%	75%	85%	95%
I	88	90.1	93.4	96.02	98	103.2	105.7	109.9	115.6
II	95.7	97.4	100.3	104.1	109.3	114	119.4	125.2	131.5

We also provide additional simulation parameters in Table 5.2 Note that in this case we set our cloud data to be distributed in nature, i.e with about 5 processing centers. Distributed in the sense that they are scattered in three different regions, but fully interconnected hence operating as a single distributed Fog Cloud data center .

Table 5.2: Additional Parameters

center	*region*	*user* request rate (per/hr)	*request* size (in bytes)	*peak times*	*average peak* users	*off – peak users*
A		300	200	5-8 am	1500	150
B		1500	150	7-11 am	1500	150
C		2500	180	6-11 am	1500	150
D		370	190	5-10 am	1500	150
E		555	220	7-9 am	1500	150

5.4.2. Results and Discussion

In this subsection, we investigate the performance of the Fog-Cloud data center, but initially restricting ourselves to the end user and Fog layer. Key performance indicators of ineterst will be costs, energy efficiency as well as end to end delays (latencies). We note that to enhance the overall performance of the Fog layer, proper load balancing must be carried out so that VMs are not overwhelmed. As explained before, the framework's algorithm will search for less-loaded active VMs when assigning new loads. We compare our proposed algorithm to

that of the Tabu search [82]. Note that the Tabu search restricts the feasible neighborhood by neighbors that are excluded.[82].

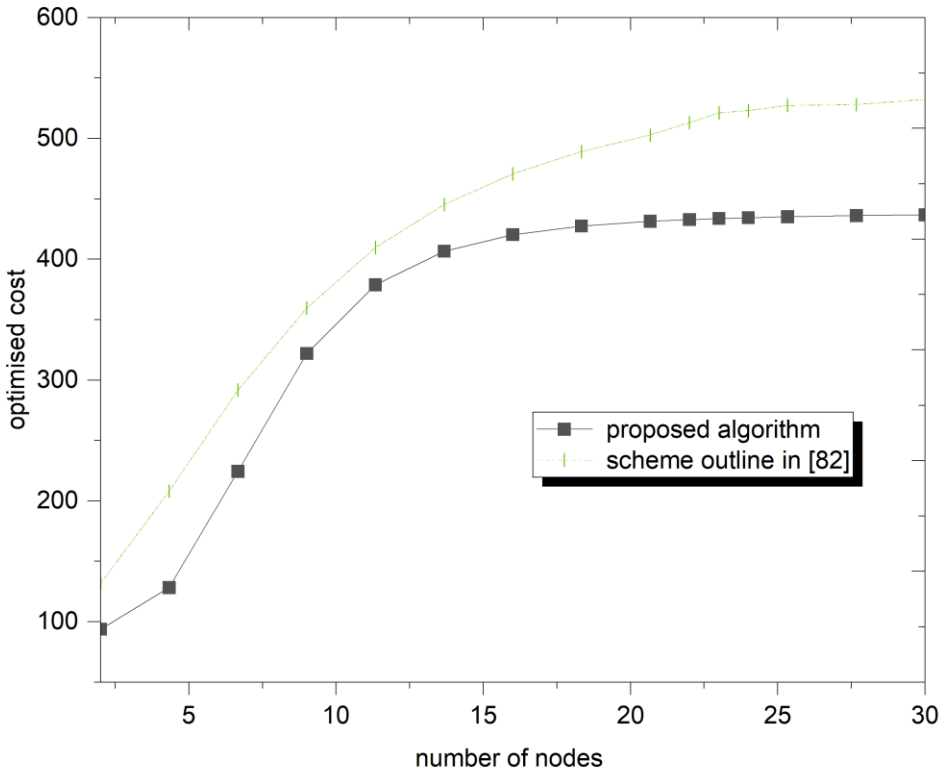

Figure 5.7: Optimised Costs Comparisons.

We observe from Fig 5.7 that as we increase the number of processing nodes, the proposed algorithm does not drastically increase the cost metrics. Partly this is because the algorithm focuses on active VMs for assigning new workloads. Only when all the VMs are critically loaded will it turn to idle VMs. Hence costs associated with cooling are minimised. The Tabu search algorithm will activate any neighbourhood node / VM irregardless of state.

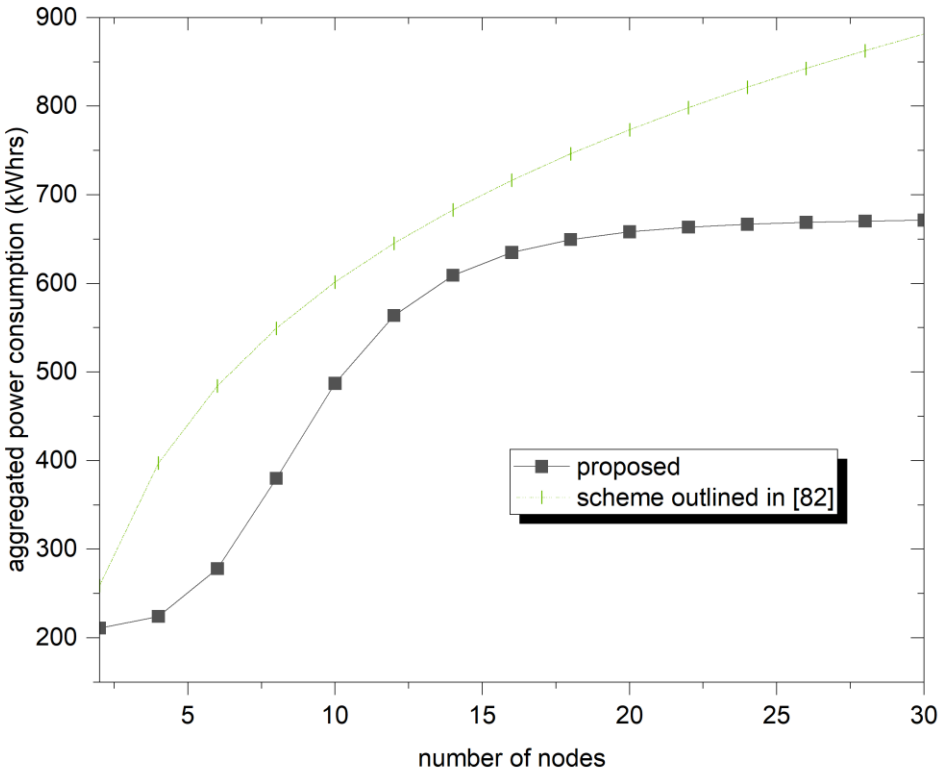

Figure 5.8: Power Consumption Comparisons.

Figure 5.8 directly compares power consumption. Once again the proposed algorithm tends to outperform. However, as the number of nodes increases, so is the power consumption levels. As can be recalled, one advantage of Fog layer computing is that there are fewer delays incurred. Partly this is because the users are in proximity. As can be noted from Figure 5.9, the proposed algorithm does not delay the jobs that much

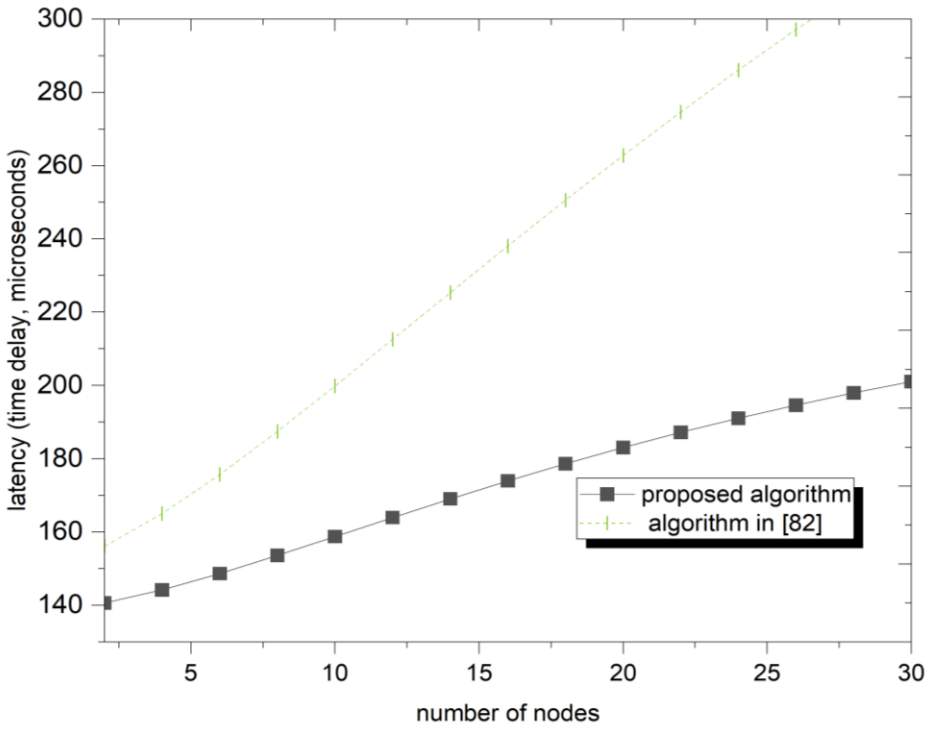

Figure 5.9: Latency Delays for Delay-Sensitive Workloads

5.4.3. Overall Fog- Cloud Data Center Performance

We further evaluate the overall performance of the Fog-Cloud data center, in this case we take into consideration that a significant proportion of the user-generated workloads are ultimately processed at the Cloud layer. We thus make a brief performance evaluation comparison of our framework's algorithm versus well-known similar algorithms, namely the Max-Min, SJF and Round Robin algorithms.

Table 5.3: Aggregate Execution Times

Trial	$makespan[Av]$	$E[RT]$	AET
150 Jobs in 10 VMs	371	282	**31 352**
200 Jobs in 13 VMs	435	305	**51 325**
250 Jobs in 16 VMs	441	311	**71 346**
300 Jobs in 18 VMs	456	319	**100 324**
350 Jobs in 21 VMs	**463**	**331**	**121 897**

Table 5.3 provides the average makespan, response time and average execution times of the proposed framework's algorithm. As expected, as offered workloads increase so would be these parameters.

Table 5.4: Makespan Comparisons

Trial	*makespan*[*Av*]			
	proposed	*Max – Min*	*RR*	*SJF*
150 Jobs in 10 VMs	400	419	453	**457**
200 Jobs in 13 VMs	451	478	488	**492**
250 Jobs in 16 VMs	478	501	531	**536**
300 Jobs in 18 VMs	483	520	561	**562**
350 Jobs in 21 VMs	**499**	**547**	**589**	**590**

Both Table 5.4 and Figure 5.10 demonstrate that the algorithm comparably reduces the makespan.

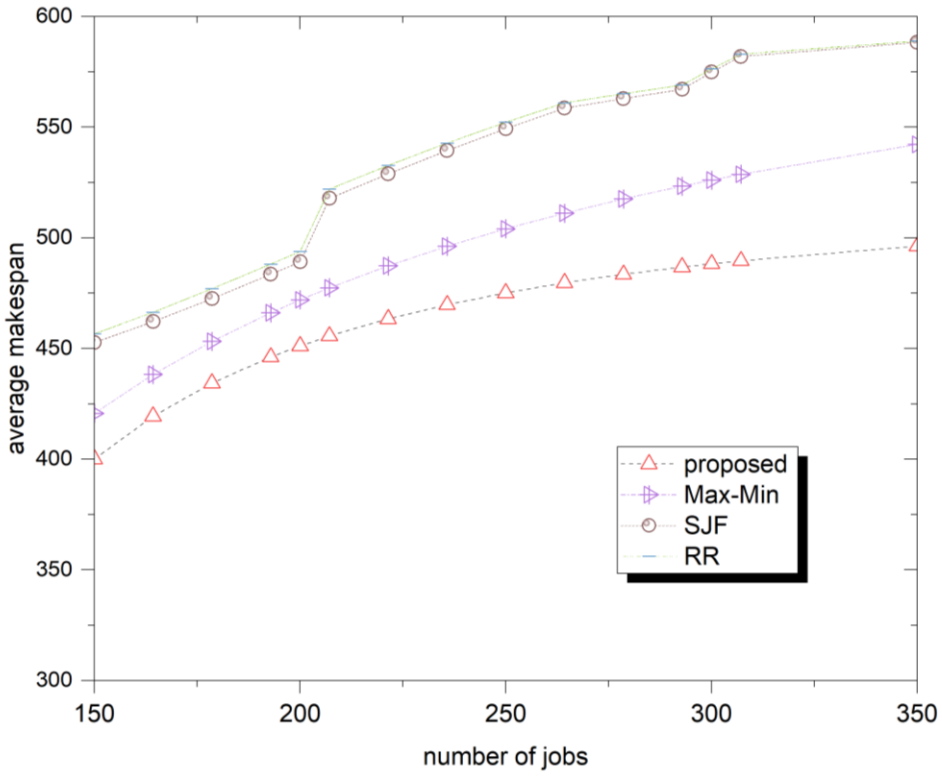

Figure 5.10. Average Makespan

Similarly, it is observed from both Table 5.5 as well as Figure 5.11 that the algorithm reduces the response time

Table 5.5: Average Response Times

Trial	$E[RT]$			
	proposed	*Max – Min*	*RR*	*SJF*
150 Jobs in 10 VMs	287	310	350	**359**
200 Jobs in 13 VMs	318	366	407	**424**
250 Jobs in 16 VMs	330	384	436	**510**
300 Jobs in 18 VMs	346	385	434	**446**
350 Jobs in 21 VMs	**348**	**472**	**457**	**406**

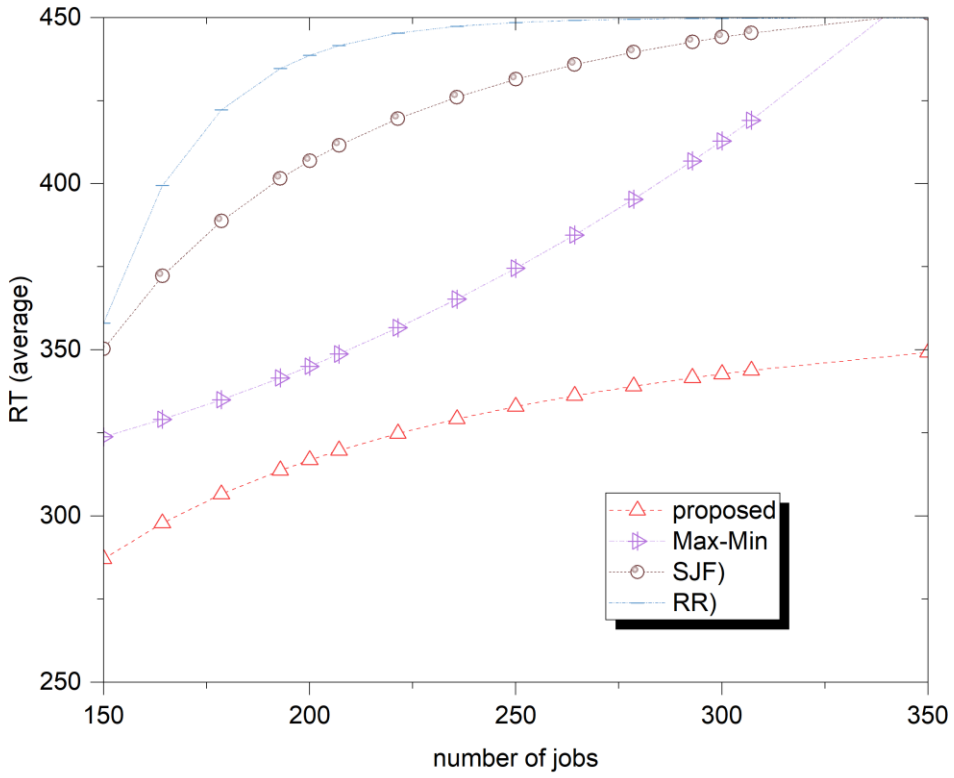

Figure 5.11: Average Response Times

Finally, we also compared the total execution times in Table 5.6 and Figure 5.12.

Table 5.6: Total Execution Time Comparisons

Trial	$E[RT]$			
	proposed	*Max – Min*	*RR*	*SJF*
150 Jobs in 10 VMs	23564.8	24085.6	27379.2	**28120**
200 Jobs in 13 VMs	40546.4	42803.2	47687.2	**49699.2**
250 Jobs in 16 VMs	57340.8	60246.4	68243.2	**80241.6**
300 Jobs in 18 VMs	73606.4	75300.8	85134.4	**87525.6**
350 Jobs in 21 VMs	**90364.8**	**95656**	**107308**	**111028**

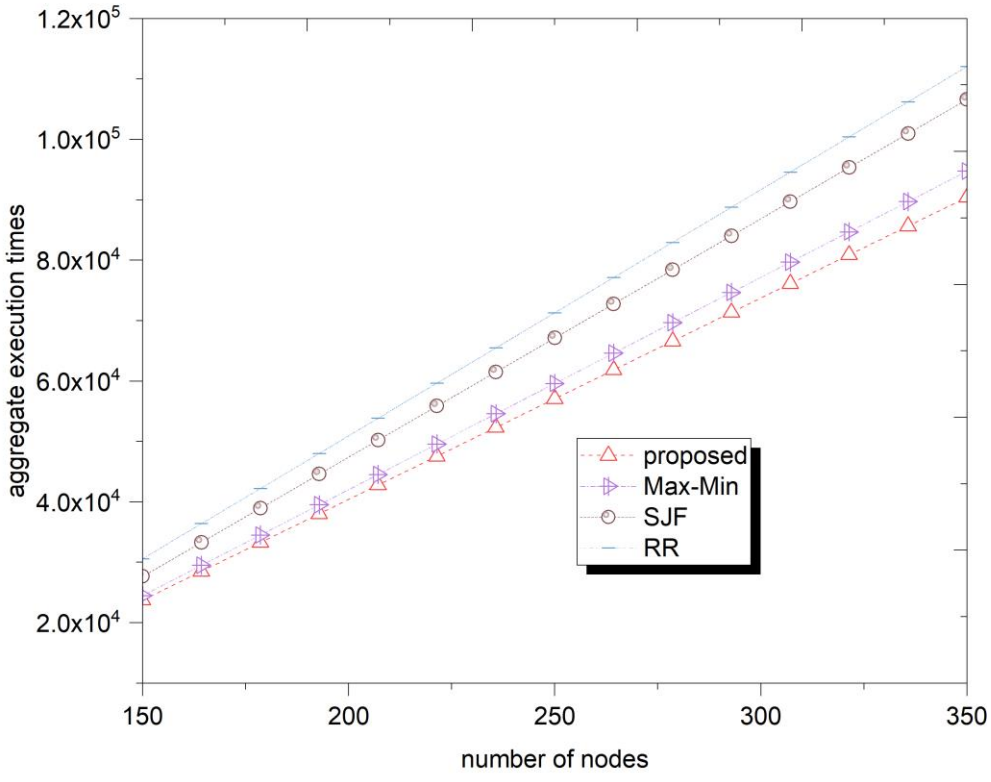

Figure 5.12: Total Execution Times

5.5. Chapter Conclusions

In this chapter, our efforts devote mostly to discussing key drivers for achieving energy efficiency. In particular, we referred to a Fog Cloud data center paradigm. The choice for such a paradigm is that the Fog servers are normally placed in proximity to end users and as such QoS related issues such as latencies for critical mission services and other applications can easily be overcome. We also note that the key to achieving energy efficiency in its operations would be sound load balancing among the active servers as well as appropriate scheduling. The latter is necessary to ensure that we maximize of the completion jobs in the shortest time possible using the minimum possible resources. In that way, much less power will be consumed.

The chapter extensively reviewed a typical Fog Cloud architecture. We also carried out an in-depth classification of load-balancing approaches. A load balancing framework was proposed and a blanching algorithm was proposed and analyzed in respect of the key QoS metrics that would affect both the users' satisfaction as well as overall energy efficiency A comparative performance evaluation of the algorithm was carried out at Fog as well as at Fog-Cloud data center levels. Overall it is concluded that load balancing at VMs level coupled with sound scheduling is key to achieving energy efficiency.

6. Conclusions

6.1. Achievements

As is known, the continuing global surge in various cloud services, IoT, and Edge (Fog) computing has led to a sudden increase in the demand for Datacenters. By definition, a data center is **a** physical facility that corporations/organizations use to house their critical applications and data. A data canter's design is based on a network of computing and storage resources that enable the delivery of shared applications and data.

Notable advantages of Data Centers include but are not limited, to their ability to provide services to end-users based on affordable rates in various plans as per contractual agreements. They also offer a robust hardware ecosystem as well as software. In operational terms, data centers offer reliable and enhanced system performance by way of carefully distributing the traffic loads uniformly across the cluster nodes.

A notable drawback of Datacenters is the high power consumption which up both CAPEX and OPEX costs. E.g it is prohibitively costly to erect robust cooling systems for a large-scale data center. The same cooling system ought to be scalable to accommodate future expansions of the data centers in terms of new services that may require new hardware to be incorporated. Thus scalability of energy supply capacity is quite a challenge. Thus, how to maximize power utilization and optimizing the performance per power budget is critical for data centers to deliver enough computation ability. Overall the operational costs of Data centers directly link the resource management algorithms implemented to assign virtual machines (VMs) to actual hardware servers and degrees of flexibility to relocate them elsewhere in case of emergencies usually associated with power losses of excessive heating of system elements. Thus this thesis focused on proposing a resource allocation and optimization scheme, that considers constraints such as energy consumption and cooling-related energy consumption in addition to the scalability issue. We also incorporate a load-balancing algorithm to minimize the operational costs of the proposed distributed data center cloud system.

Thus in the first chapter, we defined the various data center types as well as factors to be considered in their design. Challenges are also explored Overall we note that notable advantages of Data Centers include but are not limited, to their ability to provide services to end-users based on affordable rates in various plans as per contractual agreements. They also offer a ro-

bust hardware ecosystem as well as software. In operational terms, data centers offer reliable and enhanced system performance by way of carefully distributing the traffic loads uniformly across the cluster nodes. End users are excused from maintenance responsibilities. Data centers also afford instant scalability based on changing capacity demands by users. To enhance the fail-safe abilities of data centers, backup systems are incorporated. The objectives and aims of the thesis are also spelled out. The next chapter will mostly center on energy consumption and performance-related issues. An overview of key methods implemented to improve both energy and performance efficiency in distributed data center systems will be discussed. The power efficiency approaches will be addressed at various levels, namely hardware, applications, and; resource management. A taxonomy of the various performance management approaches will also be overviewed together with insights into energy, performance, and cost management issues in distributed systems.

In the following chapter, (Chapter 2), the focus was on describing the general infrastructure of the data center system and key design and operational aspects. Virtualization and containerization technology principles were overviewed. Energy consumption and performance-related issues were also dealt with. An overview of key methods implemented to improve both energy and performance efficiency in distributed data center systems is also covered. Note that the power efficiency approaches are addressed at hardware, applications, and resource management levels. We also provide insight into, energy, performance, and cost management issues in distributed systems.

Chapter 3 devotes to scheduling given that as a key integral part of facilitating (enabling) energy efficiency in data centers. We surveyed energy-efficient scheduling algorithms. It is noted that the designing of a new generation of scheduling algorithms that are energy efficient was triggered by a rise of data center-associated workloads These will aim to appropriately dimension the available data center resources subject to certain objectives that include but are not limited to maximizing energy efficiency, maximizing utilization, minimizing live VM migration, preventing QoS degradations of jobs already being served. In terms of classification, we categorize them into global, local, centralized, distributed or hierarchical, static, and dynamic.

Chapter Four centers on the operation of energy-efficient data center systems. The same chapter also discusses the choice of configurations and operation procedures. Various paradigm ar-

chitectures were overviewed. The focus is on mitigating an architecture that would best promote overall cost-effectiveness and energy efficiency.

The implementation of appropriate resource allocation, as well as scheduling algorithms is further discussed taking into consideration the multi-objective goals that have to be met. Insights pertaining to features and characteristics that influence its overall performance are provided.

Given that selection of one paradigm over the other requires careful consideration, we thus list down a set of criteria that designers can base on in choosing a particular paradigm.

The final chapter devotes mostly to discussing key drivers for achieving energy efficiency. In particular we referred to a Fog Cloud data center paradigm. The choice for such a paradigm is that the Fog servers are normally placed in proximity to end users and as such QoS related issues such as latencies for critical mission services and other applications can easily be overcome. We also note that the key to achieving energy efficiency in its operations would be sound load balancing among the active servers as well as appropriate scheduling. The latter is necessary to ensure that we maximize of the completion jobs at the shortest time possible using the minimum possible resources. In that way, much less power will be consumed.

The chapter extensively reviewed a typical Fog Cloud architecture. We also carried out an in-depth classification of load-balancing approaches. A load balancing framework was proposed and a balancing algorithm was proposed and analyzed in respect of the key QoS metrics that would affect both the users' satisfaction as well as overall energy efficiency A comparative performance evaluation of the algorithm was carried out at Fog as well as at Fog-Cloud data center levels. Overall it is concluded that load balancing at VMs level is coupled with sound scheduling is key to achieving energy efficiency.

Overall the work done herein has shown that load ad balancing significantly reduces the operational costs of the overall cloud data center system hence making it energy efficient.

6.2. Future Directions

Based on the outcomes of the research, we believe that in the future, the focus shift will be on the design of energy-efficient scheduling algorithms in highly heterogeneous, dynamic, and complex Edge and Fog computing environments at scale. Incorporating artificial in-

telligence (neural networks) in the load balancing at VM and host server levels would also be studied.

References

[1] M. Y. uddin and S. Ahmad, "A Review on Edge to Cloud: Paradigm Shift from Large Data Centers to Small Centers of Data Everywhere," 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 318-322, doi: 10.1109/ICICT48043.2020.9112457.C. Lee and A. Fumagalli, "Internet of Things Security - Multilayered Method For End to End Data Communications Over Cellular Networks," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 24-28, doi: 10.1109/WF-IoT.2019.8767227.

[2] M. Y. uddin and S. Ahmad, "A Review on Edge to Cloud: Paradigm Shift from Large Data Centers to Small Centers of Data Everywhere," 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 318-322, doi: 10.1109/ICICT48043.2020.9112457.

[3] Y. Zhao, W. Zhang, M. Yang and H. Shi, "Network Resource Scheduling For Cloud/Edge Data Centers," 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), 2020, pp. 1-4, doi: 10.1109/IPCCC50635.2020.9391529.

[4] Y. Yu, G. Cheng and X. Qiang, "Data centre transformation: Integrated business model framework of cloud computing oriented data centre," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1848-1855, doi: 10.1109/IMCEC.2016.7867538.

[5] I. Arapakis et al., "Towards Specification of a Software Architecture for Cross-Sectoral Big Data Applications," 2019 IEEE World Congress on Services (SERVICES), 2019, pp. 394-395, doi: 10.1109/SERVICES.2019.00120.

[6] K. M. U. Ahmed, M. H. J. Bollen and M. Alvarez, "A Review of Data Centers Energy Consumption and Reliability Modeling," in IEEE Access, vol. 9, pp. 152536-152563, 2021, doi: 10.1109/ACCESS.2021.3125092.

[7] C. Z. Rădulescu and D. M. Rădulescu, "A performance and power consumption analysis based on processor power models," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2020, pp. 1-4, doi: 10.1109/ECAI50035.2020.9223124.

[8] "IEEE/ASHRAE Draft Guide for the Ventilation and Thermal Management of Batteries for Stationary Applications," in IEEE P1635/D22 /ASHRAE Guideline 21, March 2022 , vol., no., pp.1-141, 4 March 2022.

[9] D. Chen, Q. Li and L. Kong, "Process Customization Framework in SaaS Applications," 2013 10th Web Information System and Application Conference, 2013, pp. 471-474, doi: 10.1109/WISA.2013.94.

[10] Y. Ding, K. Li and Z. Meng, "CPS Optimal Control for Interconnected Power Grid Based on Model Predictive Control," 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), 2018, pp. 1-9, doi: 10.1109/EI2.2018.8582528.

[11] Z. Qian et al., "Workload-Aware Scheduling for Data Analytics upon Heterogeneous Storage," 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 580-587, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00088.

[12] P. Kaushik, A. Kothawale, R. DelValle, A. Jain and M. Govindaraju, "Analysis of Dynamically Switching Energy-Aware Scheduling Policies for Varying Workloads," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, pp. 130-137, doi: 10.1109/CLOUD.2018.00024.

[13] L. Yu, T. Jiang, Y. Cao and Q. Qi, "Joint Workload and Battery Scheduling with Heterogeneous Service Delay Guaranteesfor Data Center Energy Cost Minimization," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 7, pp. 1937-1947, 1 July 2015, doi: 10.1109/TPDS.2014.2329491.

[14] L. Duan, D. Zhan and J. Hohnerlein, "Optimizing Cloud Data Center Energy Efficiency via Dynamic Prediction of CPU Idle Intervals," 2015 IEEE 8th International Conference on Cloud Computing, 2015, pp. 985-988, doi: 10.1109/CLOUD.2015.133.

[15] S. Aslam, S. Aslam, H. Herodotou, S. M. Mohsin and K. Aurangzeb, "Towards Energy Efficiency and Power Trading Exploiting Renewable Energy in Cloud Data Centers," 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), 2020, pp. 1-6, doi: 10.1109/AECT47998.2020.9194169.

[16] Y. Goyal, M. S. Arya and S. Nagpal, "Energy efficient hybrid policy in green cloud computing," 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015, pp. 1065-1069, doi: 10.1109/ICGCIoT.2015.7380621.

[17] A. A. Abbasi, A. Abbasi, S. Shamshirband, A. T. Chronopoulos, V. Persico and A. Pescapè, "Software-Defined Cloud Computing: A Systematic Review on Latest Trends and Developments," in IEEE Access, vol. 7, pp. 93294-93314, 2019, doi: 10.1109/ACCESS.2019.2927822.

[18] S. Vobugari, M. K. Srinivasan and D. V. L. N. Somayajulu, "Practitioner's guide for building effective complex enterprise architecture in digital transformation: An experience-based industry best practices summary," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017, pp. 338-346, doi: 10.1109/ICATCCT.2017.8389158.

[19] J. Shin and E. Seo, "Impact of Data Center Cooling Technology to Effectiveness of Turbo-Mode," 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 1691-1693, doi: 10.1109/ICTC49870.2020.9289200.

[20] M. Vuckovic and N. Depret, "Impacts of local cooling technologies on air cooled data center server performance: Test data analysis of Heatsink, Direct Liquid Cooling and passive 2-Phase Enhanced Air Cooling based on Loop Heat Pipe," 2016 32nd Thermal Measurement, Modeling & Management Symposium (SEMI-THERM), 2016, pp. 71-80, doi: 10.1109/SEMI-THERM.2016.7458448.

[21] Y. M. Manaserh, M. I. Tradat, G. Mohsenian, B. G. Sammakia and M. J. Seymour, "General Guidelines for Commercialization a Small-Scale In-Row Cooled Data Center: A Case Study," 2020 36th Semiconductor Thermal Measurement, Modeling & Management Symposium (SEMI-THERM), 2020, pp. 48-55, doi: 10.23919/SEMI-THERM50369.2020.9142847.

[22] J. Guo et al., "Who Limits the Resource Efficiency of My Datacenter: An Analysis of Alibaba Datacenter Traces," 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), 2019, pp. 1-10, doi: 10.1145/3326285.3329074.

[23] K. M. Nwe, M. K. Oo and M. M. Htay, "Efficient Resource Management for Virtual Machine Allocation in Cloud Data Centers," 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 2018, pp. 419-420, doi: 10.1109/GCCE.2018.8574804.

[24] M. Balaji and C. A. Kumar, "Inter-Application Based Resource Management Approach for Cloud Infrastructure," 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2018, pp. 1-6, doi: 10.1109/NTMS.2018.8328728.

[25] D. Saxena, A. K. Singh and R. Buyya, "OP-MLB: An Online VM Prediction based Multi-objective Load Balancing Framework for Resource Management at Cloud Datacenter," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2021.3059096.

[26] K. D. Rathod and M. R. Desai, "A Novel Approach for Resource Provisioning in Cloud Using Load Comfort Index and VM Demand: A Preview," 2018 3rd International Conference for Convergence in Technology (I2CT), 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529783.

[27] K. Kaur, S. Garg, G. Kaddoum, E. Bou-Harb and K. R. Choo, "A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers in IoT Setups," in IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2687-2697, April 2020, doi: 10.1109/TII.2019.2939573.

[28] H. Saadatfar, H. Deldari and M. Naghibzadeh, "Improving the Scheduler's Energy Saving Capability by Noting both Job and Resource Characteristics," in The Computer Journal, vol. 58, no. 6, pp. 1482-1493, June 2015, doi: 10.1093/comjnl/bxu098.

[29] J. Han and S. Lee, "Performance Improvement of Linux CPU Scheduler Using Policy Gradient Reinforcement Learning for Android Smartphones," in IEEE Access, vol. 8, pp. 11031-11045, 2020, doi: 10.1109/ACCESS.2020.2965548.

[30] Z. Li and H. Shen, "Co-Scheduler: A Coflow-Aware Data-Parallel Job Scheduler in Hybrid Electrical/Optical Datacenter Networks," in IEEE/ACM Transactions on Networking, doi: 10.1109/TNET.2022.3143232.

[31] M. S. Iqbal, S. A. Abbas Kazmi, Y. Sadi and S. Coleri, "Scheduling and Relay Selection for Full-Duplex Wireless Powered Cooperative Communication Networks," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 188-193, doi: 10.1109/ICCC51575.2020.9345137.

[32] T. He, H. Khamfroush, S. Wang, T. La Porta and S. Stein, "It's Hard to Share: Joint Service Placement and Request Scheduling in Edge Clouds with Sharable and Non-Sharable Resources," 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018, pp. 365-375, doi: 10.1109/ICDCS.2018.00044.

[33] T. Wu, X. Fan, Y. Qu and P. Yang, "MobiEdge: Mobile Service Provisioning for Edge Clouds with Time-varying Service Demands," 2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS), 2021, pp. 490-497, doi: 10.1109/ICPADS53394.2021.00067.

[34] K. Wang, Q. Zhou, S. Guo and J. Luo, "Cluster Frameworks for Efficient Scheduling and Resource Allocation in Data Center Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3560-3580, Fourthquarter 2018, doi: 10.1109/COMST.2018.2857922.

[35] H. T. Minh and M. Samejima, "An Evaluation of Job Scheduling Based on Distributed Energy Generation in Decentralized Data Centers," 2015 IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 1172-1177, doi: 10.1109/SMC.2015.210.

[36] T. Feng, J. Bi and K. Wang, "Allocation and scheduling of network resource for multiple control applications in SDN," in China Communications, vol. 12, no. 6, pp. 85-95, June 2015, doi: 10.1109/CC.2015.7122483.

[37] Z. Guo-Hong, "Network Resource Scheduling Mechanism of Cloud Computing Based on SDN," 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2016, pp. 332-336, doi: 10.1109/ICITBS.2016.60.

[38] M. Y. uddin and S. Ahmad, "A Review on Edge to Cloud: Paradigm Shift from Large Data Centers to Small Centers of Data Everywhere," 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 318-322, doi: 10.1109/ICICT48043.2020.9112457.

[39] I. Hewapathirana and T. Silva, "A big Data Analytics Framework for the Integration of Heterogeneous Federated Data Centers," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 650-657, doi: 10.1109/ICICT50816.2021.9358503.

[40] M. K. Jackson Ramphela, P. A. Owolawi, T. Mapayi and G. Aiyetoro, "Internet of Things (IoT) Integrated Data Center Infrastructure Monitoring System," 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), 2020, pp. 1-6, doi: 10.1109/icABCD49160.2020.9183873.

[41] M. Muniswamaiah, T. Agerwala and C. C. Tappert, "A Survey on Cloudlets, Mobile Edge, and Fog Computing," 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2021, pp. 139-142, doi: 10.1109/CSCloud-EdgeCom52276.2021.00034.

[42] W. Jiang, Y. Wang, Y. Jiang, J. Chen, Y. Xu and L. Tan, "Research on mobile Internet mobile agent system dynamic trust model for cloud computing," in China Communications, vol. 16, no. 7, pp. 174-194, July 2019, doi: 10.23919/JCC.2019.07.014.

[43] F. Yang, H. Liu and T. Bi, "A Distributed Data Center for Full-view Synchronous Measurement System Based on OpenPDC/ OpenHistorian," 2021 IEEE 4th International Electrical and Energy Conference (CIEEC), 2021, pp. 1-5, doi: 10.1109/CIEEC50170.2021.9511078.

[44] B. Cao et al., "Multiobjective 3-D Topology Optimization of Next-Generation Wireless Data Center Network," in IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3597-3605, May 2020, doi: 10.1109/TII.2019.2952565.

[45] S. R. Talpur, S. Abdalla and T. Kechadi, "Towards middleware security framework for next generation data centers connectivity," 2015 Science and Information Conference (SAI), 2015, pp. 1277-1283, doi: 10.1109/SAI.2015.7237308.

[43] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. ACM SIGCOMM Computer Communication Review, 39(4):39–50, 2009.

[44] G. Wang, D. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in ACM SIGCOMM Computer Communication Review, vol. 40, no. 4. ACM, 2010, pp. 327–338.

[45] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, TS Ng, M. Kozuch, and M. Ryan. c-Through: Part-time optics in data centers. In ACM SIGCOMM Computer Communication Review, volume 40, pages 327–338. ACM, 2010.

[46] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: a hybrid electrical/optical switch architecture for modular data centers," in ACM SIGCOMM Computer Communication Review, vol. 40, no. 4. ACM, 2010, pp. 339–350

[47] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, "Osa: An optical switching architecture for data center networks with unprecedented flexibility," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 18–18.

[48] Server Clusters. "In Power-Aware Computer Systems", pages 179–197. Springer, 2003.

[49] K. Bondcap, "Internet Trend 2019," June 11, 2019.

[50] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A scalable and fault-tolerant network structure for data centers. ACM SIGCOMM Computer Communication Review, 38(4):75–86, 2008.

[51] Financier Worldwide, "The emergence of edge computing," Dec 2019.

[52] C. Sharma," The Edge Internet Economy Forecast to be Worth Over $4.1 Trillion," August 2019

[53] N. H. Motlagh, M. Bagaa, and T. Taleb, ''UAV-based IoT platform: A crowd surveillance use case,'' IEEE Commun. Mag., vol. 55, no. 2, pp. 128–134, Feb. 2017.

[54] M. Peng, S. Yan, K. Zhang, and C. Wang. ''Fog-computing-based radio access networks: Issues and challenges,'' IEEE Netw., vol. 30, no. 4, pp. 46–53, Jul. 2016

[55] J. Valenzuela, J. Wang, and N. Bissinger, ''Real-time intrusion detection in power system operations,'' IEEE Trans. Power Syst., vol. 28, no. 2, pp. 1052–1062, Nov. 2013.

[56] S. Anwar et al., ''from intrusion detection to an intrusion response system Fundamentals, requirements, and future directions,'' MDPI Algorithms, vol. 10, no. 2, pp. 1–24, Mar. 2017.

[57] C. C. Lee, C. H. Liu, and M. S. Hwang, ''Guessing attacks on strong password authentication protocol,'' Int. J. Netw. Security, vol. 15, no. 1, pp. 64–67, 2013.

[58] T. Cruz et al., ''A cyber security detection framework for supervisory control and data acquisition systems," IEEE Trans. Ind. Informat., vol. 12, no. 6, pp. 2236-2246,Aug. 2016

[59] R. Marty, ''Cloud application logging for forensics,'' in Proc. ACM Symp. Appl Comput. (SAC), Mar. 2011, pp. 178-184.

[60] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, ‘‘Toward intelligent machine-to-machine communications in smart grid,’’ IEEE Commun. Mag., vol. 49, no. 4, pp. 60–65, Apr. 2011.

[61] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 0471417432.

[62] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, A. D. Keromytis, “Detecting Tar geted Attacks using Shadow Honeypots”, In SSYM’05: Proceedings of the 14th conference on USENIX Security Symposium, pages 9–9, Berkeley, CA, USA, 2005. USENIX Association.

[63] Xuxian Jiang, Dongyan Xu, “Collapsar: a VM-based Architecture for Network Attack Detention Center”, In SSYM’04: Proceedings of the 13th conference on USENIX Security Symposium, pages 2–2, Berkeley, CA, USA, 2004, USENIX Association.

[64] Tal Garfinkel, Mendel Rosenblum, “A Virtual Machine Introspection based Architecture for Intrusion Detection”, In In Proc. Network and Distributed Systems Security Symposium, pages 191–206, 2003.

[65] Dolezal, Ondrej. An HTTP Proxy Server. [s.l.] : UTB in Zlin, 2012.

[66] W. Li, T. Yang, F.C. Delicato, P.F. Pires, Z. Tari, S.U. Khan, A.Y. Zomaya, On enabling sustainable edge computing with renewable energy resources, IEEE Commun. Mag. 56 (5) (2018) 94–101

[67] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. “VL2: a scalable and flexible data center network”. SIGCOMM Comput. Commun. Rev., 39(4):51–62, 2009.

[68] M. Al-Fares, A. Loukissas, and A. Vahdat. “A scalable, commodity data center network architecture”. In ACM SIGCOMM Computer Communication Review, volume 38, pages 63–74. ACM, 2008.

[69] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A scalable and fault-tolerant network structure for data centers. ACM SIGCOMM Computer Communication Review, 38(4):75–86, 2008.

[70] S. Saha, J. S. Deogun, and L. Xu. “Hyscale: A hybrid optical network based scalable, switch-centric architecture for data centers”. In Communications (ICC), 2012 IEEE International Conference on, pages 2934–2938. IEEE, 2012.

[71] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. “Bcube: a high performance, server-centric network architecture for modular data centers”. ACM SIGCOMM Computer Communication Review, 39(4):63–74, 2009.

[72] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. ACM SIGCOMM Computer Communication Review, 39(4):39–50, 2009.

[73] G. Wang, D. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, “c-through: Part-time optics in data centers,” in ACM SIGCOMM Computer Communication Review, vol. 40, no. 4. ACM, 2010, pp. 327–338.

[74] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, TS Ng, M. Kozuch, and M. Ryan. c-Through: Part-time optics in data centers. In ACM SIGCOMM Computer Communication Review, volume 40, pages 327–338. ACM, 2010.

[75] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in ACM SIGCOMM Computer Communication Review, vol. 40, no. 4. ACM, 2010, pp. 339–350

[76] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, Osa: An optical switching architecture for data center networks with unprecedented flexibility,” in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 18–18.

[77] Server Clusters. “In Power-Aware Computer Systems”, pages 179–197. Springer, 2003.

[78] K. Bondcap, “Internet Trend 2019,” June 11, 2019.

[79] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A scalable and fault-tolerant network structure for data centers. ACM SIGCOMM Computer Communication Review, 38(4):75–86, 2008.

[80] Financier Worldwide, “The emergence of edge computing,” Dec 2019.

[81] C. Sharma,” The Edge Internet Economy Forecast to be Worth Over $4.1 Trillion,” August 2019

[82]. N. Siasi, A. Jaesim and N. Ghani, "Tabu Search for Efficient Service Function Chain Provisioning in Fog Networks," 2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC), 2019, pp. 145-150.

[83] M. Peng, S. Yan, K. Zhang, and C. Wang. ‘‘Fog-computing-based radio access networks: Issues and challenges,’’ IEEE Netw., vol. 30, no. 4, pp. 46–53, Jul. 2016

[84] J. Valenzuela, J. Wang, and N. Bissinger, ‘‘Real-time intrusion detection in power system operations,’’ IEEE Trans. Power Syst., vol. 28, no. 2, pp. 1052–1062, Nov. 2013.

[85] S. Anwar et al., ‘‘from intrusion detection to an intrusion response system Fundamentals, requirements, and future directions,’’ MDPI Algorithms, vol. 10, no. 2, pp. 1–24, Mar. 2017.

[86] C. C. Lee, C. H. Liu, and M. S. Hwang, ‘‘Guessing attacks on strong password authentication protocol,’’ Int. J. Netw. Security, vol. 15, no. 1, pp. 64–67, 2013.

[87] T. Cruz et al., ‘‘A cyber security detection framework for supervisory control and data acquisition systems,” IEEE Trans. Ind. Informat., vol. 12, no. 6, pp. 2236-2246,Aug. 2016

[88] R. Marty, ‘‘Cloud application logging for forensics,’’ in Proc. ACM Symp. Appl Comput. (SAC), Mar. 2011, pp. 178-184.

[89] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, ‘‘Toward intelligent machine-to-machine communications in smart grid,’’ IEEE Commun. Mag., vol. 49, no. 4, pp. 60–65, Apr. 2011.

[90] V. S. Kireev et al., "Predictive Repair and Support of Engineering Systems Based on Distributed Data Processing Model within an IoT Concept," 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2018, pp. 84-89, doi: 10.1109/W-FiCloud.2018.00019.

[91] T. Islam and M. M. A. Hashem, "Task Scheduling for Big Data Management in Fog Infrastructure," 2018 21st International Conference of Computer and Information Technology (ICCIT), 2018, pp. 1-6, doi: 10.1109/ICCITECHN.2018.8631959.

[92] P. Geetha and C. R. R. Robin, "A comparative-study of load-cloud balancing algorithms in cloud environments," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 806-810, doi: 10.1109/ICECDS.2017.8389549.

[93] A. Agarwal, G. Manisha, R. N. Milind and S. S. Shylaja, "Performance analysis of cloud based load balancing techniques," 2014 International Conference on Parallel, Distributed and Grid Computing, 2014, pp. 49-52, doi: 10.1109/PDGC.2014.7030714.

[94] R. R. Kumar, S. K. Jha, D. Garg and S. Vaishnav, "Evaluation of Load Balancing Algorithm Using Cloudsim," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2018, pp. 78-81, doi: 10.1109/ICICT43934.2018.9034367.

[95] H. Xu, G. Wang, L. Luo and M. Lei, "The Design of Reliability Simulation of Cloud System in the Cloudsim," 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2018, pp. 215-219, doi: 10.1109/ICCWAMTIP.2018.8632572.

[96] P. Humane and J. N. Varshapriya, "Simulation of cloud infrastructure using CloudSim simulator: A practical approach for researchers," 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015, pp. 207-211, doi: 10.1109/ICSTM.2015.7225415.

[97] S. Santra and K. Mali, "A new approach to survey on load balancing in VM in cloud computing: Using CloudSim," 2015 International Conference on Computer, Communication and Control (IC4), 2015, pp. 1-5, doi: 10.1109/IC4.2015.7375671.

[98] [M. Haghi Kashani and E. Mahdipour, "Load Balancing Algorithms in Fog Computing: A Systematic Review," in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2022.3174475.

[99] J. Yan, J. Wu, Y. Wu, L. Chen and S. Liu, "Task Offloading Algorithms for Novel Load Balancing in Homogeneous Fog Network," 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2021, pp. 79-84, doi: 10.1109/CSCWD49262.2021.9437748.