Disaggregated Servers for Future Energy Efficient Data Centres

Howraa Mehdi Mohammad Ali

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Electronic and Electrical Engineering

Jan 2017

The candidate confirms that the work submitted is her own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapter 3 is based on the work from:

H. M. M. Ali, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient disaggregated servers for future data centers," 20th IEEE European Conference on Networks and Optical Communications (NOC), 2015, pp. 1-6.

Prof. Elmirghani, the supervisor, suggested the study of the energy efficiency of resource provisioning in disaggregated server based data centres. The co-supervisor Dr Lawey, was the co-founder of the idea and revised the paper. The co-supervisor, Dr El-Gorashi, worked with the student on the MILP model development, results analyses and paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote the paper.

And:

H. M. Ali, A. Lawey, T. E. Elgorashi, and J. Elmirghani, "Energy Efficient Resource Provisioning in Disaggregated Data Centres," OSA Asia Communications and Photonics Conference (ACP), 2015, pp. 1-3.

Prof. Elmirghani, the supervisor, contributed the concept of the energy efficiency of resource provisioning in disaggregated server based data centres study. The co-supervisor Dr Lawey, worked with the student on paper preparation, results analyses and revised the paper. The co-supervisor, Dr El-Gorashi, worked with the student on the heuristic development. The PhD student developed the heuristic, obtained and analysed the results, and wrote the paper.

Chapter 4 is based on the work from:

H. M. Ali, T. E. Elgorashi, A. Lawey, and J. Elmirghani, "Future Energy Efficient Data Center with Disaggregated Servers" to be submitted to IEEE Transaction on Networking.

Prof. Elmirghani, the supervisor, suggested the design of a disaggregated server. The co-supervisor, Dr El-Gorashi, worked with the student on the development of the basic concept. The co-supervisor Dr Lawey, worked with the student on the design, building the basic blocks, suggested the disaggregation of the memory controller, and reviewed the design description. The PhD student developed the design, analysed its parts and blocks, checked the functionality, suggested components for practical implementation and wrote the design description and analysis as part of the paper.

Chapter 5 is based on the work from:

H. M. Ali, T. E. Elgorashi, A. Lawey, and J. Elmirghani, "Future Energy Efficient Data Center with Disaggregated Servers" to be submitted to IEEE Transaction on Networking.

Prof. Elmirghani, the supervisor, suggested the concept of considering the communication fabric power consumption in addition to the resource provisioning power consumption in the disaggregated server based data centre design. The co-supervisor, Dr El-Gorashi, worked with the student on the development of the basic concept of the model. The co-supervisor Dr Lawey, worked with the student on the MILP model and heuristic development, results analyses and paper preparation and revision. The PhD student developed the model and the simulation heuristic, obtained and analysed the results, and wrote the paper.

Chapter 6 is based on the work from:

H. Mohammad Ali, A. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server Design," 18th IEEE International Conference on Transparent Optical Networks (ICTON), 2016, pp.1-5.

Prof. Elmirghani, the supervisor, suggested the inclusion of time as a new dimension in the VM requirements idea when implementing VM allocation in disaggregated server. The co-supervisor, Dr El-Gorashi, worked with the student on the development of the basic concept. The co-supervisor Dr Lawey, worked with the student on the heuristic development, results analyses and paper preparation and revision. The PhD student developed the simulation heuristic, obtained and analysed the results, and wrote the paper.

Acknowledgements

I would like to sincerely thank my supervisor, Professor Jaafar Elmirghani for his inestimable assistance and patience throughout my PhD journey. His insights and valuable advice paved the road toward accomplishing the goals of this work.

I am truly grateful to my co-supervisor, Dr. Ahmed Lawey, for the patient guidance, encouragement and immeasurable advice he has provided throughout my time as his student.

I would like to acknowledge Dr. T. H. El-Gorashi, my co-supervisor for her advice and useful discussions.

I would like to acknowledge the Higher Committee for Education Development in Iraq (HCED) for funding my Ph.D.

I am indebted to my two angels, my daughter Zainab, my son Mahdi; and my beloved husband Ali for their everlasting support and love through all stages of my life and especially during my Ph.D.

I wish to express my love and gratitude to all my family, specially my beloved mother, Najat, and my wonderful father, Mahdi, my soul mate sister, Zahraa, my kind brothers Salah and Alaa, my father in law, Mahdi, my mother in law, Rajiha, for their support, patience and understanding during the course of my Ph.D. I would like to express my gratitude and acknowledgment to my darling grandma who passed away during my Ph.D. course for her endless prayers and love.

Finally, I thank all my friends with special thanks to all my colleagues for their friendship, input and valuable discussions.

Abstract

The popularity of the Internet and the demand for 24/7 services uptime is driving system performance and reliability requirements to levels that today's data centres can no longer support. This thesis examines the traditional monolithic conventional server (CS) design and compares it to a new design paradigm known as disaggregated server (DS). The DS design arranges data centres resources in physical pools such as processing, memory and IO module pools; rather than packing each subset in a single server. In this work, we study energy efficient resource provisioning and virtual machine (VM) allocation in the DS based data centres compared to CS based data centres. First, we developed a mixed integer linear programming (MILP) model to optimise VM allocation for DS based data centre. Our results indicate that considering pooled resources yields up to 62% total saving in power consumption compared to the CS approach. Due to the MILP high computational complexity, we developed an energy efficient, fast and scalable resource provisioning heuristic (EERP-DS), based on the MILP insights, with comparable power efficiency to the MILP. Second, we extended the resources provisioning and VM allocation MILP to include the data centre communication fabric power consumption. The results show that the inclusion of the communication fabric still yields considerable power savings compared to the CS approach, up to 48% power saving. Third, we developed an energy efficient resource provisioning for DS with communication fabric heuristic (EERP-DSCF). EERP-DSCF achieved comparable results to the second MILP and with it we can extend the number of served VMs where the MILP scalability for big number of VMs is challenging. Finally, we present our new design for the photonic DS based data centre architecture supplemented with a complete description of the architecture components, communication patterns and some recommendations for the design implementation challenges.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CPUMC	CPU Attached Memory Controller
CS	Conventional Server
DCN	Data Centre Network
DMA	Direct Memory Access
DMC	Disassembled Memory Controller
DS	Disaggregated server
EERP-DS	Energy Efficient Resource Provisioning in DS
EERP-DSCF	Energy Efficient Resource Provisioning in DS with Communication Fabric
EERPVMM-DS	Energy Efficient Resource Provisioning with VM migration in DS
EXC	Electronic Core Packet Switch
GOS	Global Data Centre Operating System
IAT	Inter-arrival Time
ICT	Information and Communication Technologies
IO	Input/Output
IOI	IO Intensive
MEMC	Memory Attached Memory Controller
MEXC	Electronic Memory Switch
MI	Memory Intensive
MILP	Mixed Integer Linear Programming
MMC	Middling Memory Controller
ns	Nanoseconds

OAM	Optically Attached Memory
OS	Operating System
OSPF	Open Shortest Path First
OXC	Optical Cross Connect
PF	Power Factor
PI	Processing Intensive
PUE	Power Usage Effectiveness
SDI	Software Defined Infrastructure
SIC	Switch Interface Card
SiPh	Silicon Photonic
TCO	Total Cost of Ownership
TDM	Time Division Multiplexing
VDC	Virtual Data Center
VM	Virtual Machine
VMM	Virtual Machine Monitor

Chapter 1: Introduction

The data centres are approaching the point of erupting; they are bursting at all seams, including storage, power, traffic, and processing needs [1]. The last few years have witnessed the adoption of clouds in the ICT sector and these have evolved very fast as a recognised, widely deployed, and accepted networking solution. Clouds have been exploited widely by companies, enterprises, and government organisations, as well as personal users; they are expected to be the main factor that will dominate the future Internet service model [2] by offering a ubiquitous access to network-based content and services, delivered to almost anywhere (network) that users wish rather than solely to desktop-based user applications [3]. One of the most significant current discussions in today’s ICT sector is the increased energy consumption due to the massive increase in the number of devices accessing the Internet – with around 40% of the world population having an Internet connection [4] – and the huge amount of generated data. According to [5], more than 2.5 quintillion bytes are being added to the total data traffic on a daily basis. Mobile data traffic is expected to grow at an annual rate of 57%, reaching a throughput of 24.3 exabytes per month by 2019. Data centre power consumption is in the range of 100-130 GWh per year, as measured by the power usage effectiveness (PUE) index, and computer room air conditioning consumes up to half the total power consumed by the data centre [5]. The generated data is massively aggregated and processed at core networks and data centres are the heart of these cores. These trends are accelerating data centre and cloud traffic growth and placing new requirements and demands on data centre and cloud-based infrastructures. Thus, there is currently a need for maximising the data centre performance and minimising its total cost of ownership (TCO) by increasing resource utilisation, reducing hardware acquisition and maintenance, and eventually delivering a better experience to end users.

Virtualised data centres are key services in modern networking. However, in today's traditional rigid architecture of current servers the ratios of CPU to memory to IO are mostly unchangeable inside data centres as they are confined within the boundaries of stand-alone servers [6]. The single box server adds barriers and difficulties including inefficient resource utilisation, prolonged provisioning, difficulties in big data management, and a high risk of blocking when deploying virtual data centre resource instances. Another challenge facing current data centres is the energy consumption of the physical infrastructure that provides resources for the cloud. Thus, energy management is a key challenge for data centres to reduce all their energy-related costs [7, 8].

Significant efforts have been dedicated to optimise the power consumption of conventional data centres including energy-efficient data centre designs [9] [10], energy-efficient inter- and intra-data centre network architectures [11, 12], designing energy-efficient cloud computing services and energy-efficient resource provisioning, and virtual network embedding for cloud systems [13, 14].

To understand the usefulness of the DS concept, consider an example of a conventional ‘server in a box’ (CS), where a processing-intensive task is occupying the processor while the input/output (IO) module is idle. Other servers, due to the current CS’s constrained architecture, cannot access such an idle resource in this case. Similarly, a server running an application that involves intensive IO usage may have a large idle fraction of the CPU processing capability not accessible by other tasks that require access through the bottlenecked IO module. The DS concept removes the barriers of the CS approach and allows virtual machines (VMs) to construct servers on the fly, with the required specifications for a specific duration, and to release these resources at the end of the task, removing many barriers and improving the data centre efficiency significantly.

Silicon photonics [15] is a promising technology that can enable DS, with light as the medium that transfers data from place to place instead of using electronics. With silicon photonics, data can move farther and at lower power consumption than with copper and an incredible amount of data can move over a single strand of fibre or optical waveguide instead of using bulky copper connections. Intel Silicon Photonic connectors, SiPh [16], provide OEO processing for full wavelength conversion at each node. This architecture will appear in more detail in Chapter 4 where our design architecture for the DS will be presented, with a full description for all the components and communications patterns.

For energy efficiency in data centres, this work presents detailed analyses of the process of energy-efficient resource provision and server consolidation, and exemplifies its implication in the DS-based data centre. The approach exploits energy-efficient resource allocation in DS and shows the impact of server disaggregation on increasing the total resources utilisation, which will reduce the overall energy consumption. The benefits were maximised using a mixed integer linear programming (MILP) mathematical optimisation, and a heuristic was developed and used to verify the MILP optimisation. The goal of the optimisation was to ensure that power consumption is minimised, which effectively resulted in the working resources being packed with as many VMs as they can hold before using new resources. The study also investigated the effect of including the communication fabric power consumption on the total power saving by developing a new MILP model to account for the power consumed by the new added networking elements. A new heuristic that mimics the MILP behaviour was developed to validate the MILP outcomes and extend the MILP scope to evaluate a higher number of VM requests.

Regarding the DS architecture design, we proposed a new switch-based communication architecture for the photonic DS-based data centre architecture supplemented with a complete description of the architecture components, communication patterns, and some recommendations for the design implementation challenges.

Finally, we study the idea of considering time as a new dimension associated with each VM and resource reallocation and VM migration concepts. The goal is to use the minimum number of resources by exploiting resource reallocation when a VM finishes its service time duration and leaves the system. In such cases, our approach reuses the released resources to serve new or migrated VMs. The total power savings were considered given different VM inter-arrival time (IAT) patterns.

1.1 Research Objectives

The following primary objectives were set for the work reported in this thesis:

1. To investigate the energy efficiency of DS-based data centres compared to CS design considering VM allocation and resource provisioning by allocating VMs in a resource pool rather than server boxes approach using mathematical modelling. Based on the insights gained through the mathematical modelling, to develop appropriate heuristics that can run in real-time environments as well as expand the data centre size and, accordingly, the number of served VMs.

2. To develop and introduce a new design for the DS architecture with a complete and comprehensive description for all the design components, highlighting the communication patterns and traffic flows between the disaggregated resources and giving some recommendations for the design implementation challenges.

3. To investigate the impact of the communication fabric power consumption on the overall system performance and energy efficiency based on the new DS design architecture.

4. To establish the conditions under which the DS and CS designs result in similar power consumption.

5. To assess the impact of VM migration and resource reallocation on the DS server's power consumption.

1.2 Original Contributions

The main contributions of this thesis are as follows:

1. The development of a MILP that models the energy consumption of VMs' allocation and resource provisioning in DS-based data centre when considering three heterogeneous resource types (processing resources, memory, and IO resources) and three VM types – processing intensive (PI), memory intensive (MI), and IO intensive (IOI). We have shown an average energy saving of up to 49% when considering the DS server design and our set of input parameters for both resources and VMs.

2. A resource provisioning heuristic (EERP-DS) was developed for real-time implementation of the energy-efficient resource provisioning in a DS-based data centre. Comparable power savings and performance were achieved by the heuristic compared to the MILP power consumption.

3. The design of a new DS switch-based server architecture with complete definitions and communication structure.

4. The development of a mathematical optimisation model along with a heuristic (EERP-DSCF) for resource provisioning in a DS-based data centre with communication, based on our switch-based DS design.

5. A heuristic (EERPVMM-DS) was developed to account for the power consumption by the DS-based server considering VM migration and resources release.

1.3 Related Publications

The original contributions in this thesis are supported by the following publications:

- **Journals**

1. Howraa M. Mohammad Ali, Taisir E. H. El-Gorashi, Ahmed Q. Lawey, and Jaafar M. H. Elmirghani, “Future Energy Efficient Data Centres with Disaggregated Servers”, to be submitted to *IEEE/ACM Transactions on Networking*.

- **Conferences**

2. Howraa M. Mohammad Ali, Ahmed Q. Lawey, Taisir E. H. El-Gorashi, and Jaafar M. H. Elmirghani, “Energy efficient disaggregated servers for future data centres” 20th IEEE Networks and Optical Communications(NOC), 2015, pp. 1-6.

3. H. M. Ali, A. Lawey, T. E. Elgorashi, and J. Elmirghani, "Energy Efficient Resource Provisioning in Disaggregated Data Centres," OSA Asia Communications and Photonics Conference (ACP), 2015, pp. 1-3.

4. H. Mohammad Ali, A. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server Design," 18th IEEE International Conference on Transparent Optical Networks (ICTON), 2016, pp.1-5.

1.4 Thesis Structure

Following the introduction in Chapter 1, the rest of the thesis is organised as follows:

Chapter 2 provides an overview of the main topics addressed in this thesis, including energy-efficient data centres and conventional data centres, with a focus on the problems encountered by these architectures. Attention is given to virtualisation and resource provisioning as a vital solution for some problems encountered by current data centres. A special section is dedicated to the DS design description followed by the anticipated benefits to be offered by the DS architecture and ending with a complete description of the technical challenges encountered by the DS design. The chapter reviews the work done by companies and academic researchers on the disaggregated data centre designs covering all the categories and disaggregation levels considered in the literature.

Chapter 3 introduces the energy-efficient resource provisioning DS design MILP model with complete results and analysis. It proposes a heuristic for energy-efficient resource provisioning in the DS server design and compares its performance to the MILP model. It investigates the energy efficiency in the DS by improving resources utilisation as compared to the CS design.

Chapter 4 introduces a new, innovative design for the DS server-based data centre with a comprehensive description of the design components and their functionality and some recommendations for reliable implementation.

Chapter 5 introduces a MILP for energy-efficient resource provisioning in DS design with communication fabric power, and a heuristic (EERP-DSCF) is proposed with a comparable power profile to the MILP.

Chapter 6 introduces a heuristic (EERPVMM-DS) that accounts for the power consumption by the DS-based server considering VM migration and resources reuse.

The thesis concludes in Chapter 7, which summarises this work’s main contributions and gives recommendations for future work.

Chapter 2: Resource Provisioning in Disaggregated Server Based Data Centres

2.1 Introduction

The work presented in this thesis has had as its objective the study of the energy efficiency of the DS-based data centre, characterised by the resource provisioning and VM allocation and migration, and the presentation of new design architecture for the DS-based racks and the communication fabric among the disaggregated resources. This chapter provides an overview of the limitations in current server-based data centres, and reviews virtualisation and DS-based data centres as a promising solution for current server-centric data centre shortcomings. We then survey the benefits, challenges, and enabling technologies for the DS with more detailed descriptions of the silicon photonics and optical interconnects for intra-rack communications. Finally, we review related work on the disaggregated data centre.

2.2 Energy-Efficient Data Centres

Today’s fast-growing number of data centres and the bursting traffic introduce energy consumption problems and motivate the need for “greening” in data centres by applying energy-saving techniques and implementing new data centre designs.

Data centre power consumption can be attributed to the following: 1) processing and storage resources power; 2) communication fabric power; and 3) cooling and heat dissipation [17]. Improving the data centre energy efficiency can be achieved through: 1) energy-aware data centre management; 2) energy-efficient IT equipment; 3) energy-efficient cooling; and 4) renewable energy resources [18]. Fig. 2-1 shows the power dissipation points in a typical data centre [19, 20]. According to Fig. 2-2, IT equipment is the main energy consumer in a data centre. Consequently, introducing energy-efficient servers and networking components can introduce a remarkable reduction in the overall data centre power consumption.

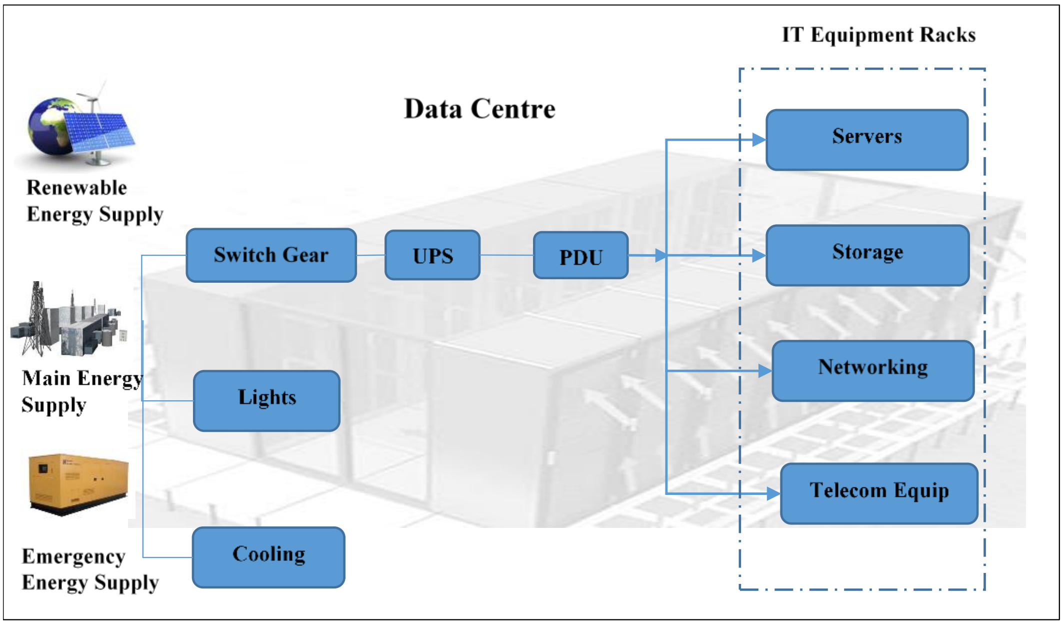


Fig. 2-1: Data centre components [19]

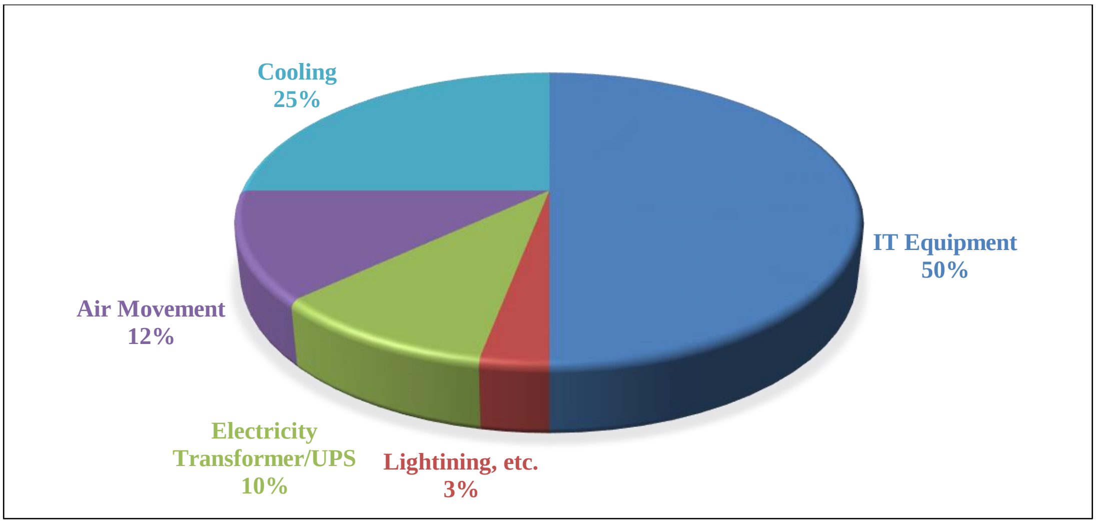


Fig. 2-2: Data centre power expenditure [20]

2.2.1 Green data centre

Energy-efficient data centre management, including energy-aware VM scheduling and consolidation [21-23] can help reduce the overall data centre energy consumption. Secondly, in many science and technology areas, energy-aware ICT solutions are being proposed [24] and low-energy equipment and components are being developed, not only to decrease the energy cost, but also to help save our environment [25]. Building data centres with free cooling systems and utilising the local weather by using outside ambient air for cooling reduces the cost of energy required for cooling. Google and Facebook have begun to incorporate more and more outside cooling in their data centres by building their data centres in cold areas. This will reduce their total OPEX by building data centres without water chillers and using free cooling systems [26, 27]. Data centres use electricity so a data centre is as clean as the electricity supplied to it. Thus, employing renewable energy resources such as solar, hydro, and wind systems to provide the required electricity for powering the data centre can have a significant impact on the data centre's energy efficiency [28, 29].

2.3 Conventional Data Centre

A data centre can be thought of in one of two ways: either as a computational resources provider or as a raw computational resource utility provider. The first view is more preferred than the second due to its simplicity at the data centre level, as it pushes complexity out to the application environment. This is because it considers the data centre provider as the mediator between actual resources (such as servers, light-paths, etc.) and the network users. However, the second model considers the data centre as a services provider (analogous to water or gas services, they give you gas and you can use it for cooking or heating) and this approach provides guarantees on metrics (e.g. bandwidth, utilisation, and service level) and they give you raw resources, such as IaaS, and you can use these resources to deploy your own work, your own applications and management [30].

It is clear that in today's traditional rigid architecture of current servers, the ratios of CPU to memory to IO are mostly unchangeable inside data centres as they are confined within the boundaries of stand-alone servers [31]. The single box server adds barriers and difficulties including inefficient resource utilisation, prolonged provisioning, difficulties in big data management [32], and a high risk of blocking when deploying virtual data centre resource instances. Another challenge facing current data centres is the energy consumption of the physical infrastructure that provides resources for the cloud. Thus, energy management is a key challenge for data centres to reduce all their energy-related costs [32, 33].

Significant efforts have been dedicated to optimising the power consumption of conventional data centres including energy-efficient data centre designs [34], energy-efficient inter-data-centre network architectures [35-37], designing energy-efficient cloud computing services and energy-efficient resource provisioning, and virtual network embedding for cloud systems [6, 38-40].

The above work is limited in that it relies on the single boxed server approach, where the flexible addition and removal of physical resources is very limited. Accordingly, the DS architecture is a potential approach to minimising data centres' power consumption. In this model, servers' resources are separated into discrete pools of resources that are mixed and matched in real time to create differently sized and shaped systems. This technique brings a new server vision for the data centres and motivates a plethora of potential new applications and services [41].

The revolutionary concept of DS will bring radical change to traditional data centres and can simplify the vertical scalability of VMs by decoupling the server components from each other. On the other hand, resources are combined according to their types in a stand-alone and type homogenous "resource rack", constructing resource pools interconnected using an optical backplane. Here, a data centre network directly interconnects all resource racks via a high-bandwidth and low-latency inter-rack switching fabric [32]. Therefore, DS design will bring sharing of CPU, memory, and network components, modularity, and independent allocation of resources, such that a certain resource is no longer tightly coupled to any other resource, which means that resources can be used more efficiently.

2.4 Virtualisation and Resource Provisioning

The term virtualisation refers in general to the abstraction of computer resources for upper-layer applications, leading to the creation of a virtual version of the underlying physical resources for the application that using them. With virtualisation, applications gain access to more resources than the physically installed resources on a single machine. The purpose behind virtualisation is to improve resource utilisation, to improve system security, reliability and availability, reduce costs, and provide greater flexibility. Full virtualisation of all system resources enables service providers to run multiple operating systems (OSs) on a single physical machine while in a non-virtualised system, the whole hardware resources are under the control of a single OS.

With virtualisation, a new software layer is included, with responsibility to monitor, control access, and maintain coherency among the different virtual machines running on top of the hardware resources, so that these resources can be shared among multiple OSs that are “guests” to the underlying physical platform. This software layer is called the hypervisor or the virtual machine monitor (VMM), in general.

From a user’s point of view, virtualisation is non-disturbing, since the user experiences are largely unchanged. However, for administrators, a virtual infrastructure gives them the advantage of flexibility in terms of being more organised and more responsive to dynamic resource management across the enterprise and to better leverage infrastructure investments.

Many approaches are being implemented in order to improve service reliability and continuity in data centres, and one of its main aspects is the resource provisioning and VM consolidation. Another approach is doing regular preventive maintenance, such as replacing the components of the power parts (generators and backup batteries), replacing servers and switches, upgrading software, and fixing security vulnerabilities. According to [42], proper preventive maintenance can prevent 30% to 40% of system outages due to infrastructure hardware failures. In fact, preventive maintenance is a routine task in large data centres, but what has not gained enough attention is the coordination between maintenance and VM resource provisioning [42].

Considering the efficient resource provisioning, and the computing resources that an application needs, the application environment varies over time for a given VM; thus, two types of resource provisioning approaches are being implemented in data centres. One approach is the static resource provisioning [16], which ensures the satisfaction of the computing needs of a particular application by providing enough resources for the expected peak demand and leading to over-provisioned VMs. However, implementing this approach in the allocation of resources will lead to under-utilisation of a data centre's resources since there will be allocated resources that are not needed most of the time. So, the static provisioning cannot fit all applications requests, and will lead to either over-provisioning or under-provisioning. The alternative solution is to be able to dynamically allocate and deallocate resources as needed for an application environment, with the deallocated resources available to be used for other application environments. This is referred to as the dynamic resource provisioning approach [43, 44]. The management system needs to make decisions about resource allocation in a situation where, at any point in time, the resource demands from all application environments exceed the resource supply. Policies are needed in advance as they are the basis for these decisions made by the management system where a policy is defined as any type of formal behavioural guide, and a change in policy should not mean a recompilation of the system [30].

2.5 Disaggregated Server

With the advent of mega trends in information and communication technologies (ICT) such as mobile, social, clouds, and big data, the required processing and services to fulfil these demands are becoming crucial. Considering traditional data centres' infrastructures and services, it is believed that they need serious improvements and developments at both hardware and software levels. Recent architecture research has introduced a new server paradigm and data centre architectural design, the DS-based data centre. The DS has been touted as one of the promising solutions for future data centres [5]. The DS design is producing new building blocks for the data centres, which can be described as resource pools. These pools can be described as a collection of homogenous resources such as processors pools, memory pools, IO module pools, and storage pools. Virtual servers are constructed on the fly using resources from these pools to suit any incoming VM requests [45, 46].

Since the conception of the term DS, and the recognised declaration of the cooperation between Intel and Facebook to "disaggregate" the server, and as they displayed their initial prototype of the "Rack Scale Server" design at the Open Compute Summit in 2013 [32], several researchers and industrial organisations have attempted to provide different definitions of the DS as well as different levels of disaggregation [16] and [47]. In general, "disaggregation" means dividing a completely integrated thing (an "aggregate body") into its component parts. In this context, a modern data centre is already disaggregated when compared to self-contained solutions such as mainframes.

The most common and acceptable definition of DS is based on removing the server box limits from resources and aggregating server resources in their type respective pools. Servers can be constructed dynamically by allocating the specific amount of resources from these pools according to the requirements of workload under consideration. The hypervisor [35] is the first layer of software installed on a virtualised system and has direct access to the hardware resources. Thus, a hypervisor enhances resources manageability by interacting efficiently with the underlying hardware platform and provisions hardware resources to the VM requests, enabling greater scalability, robustness, and performance. In a DS-based data centre, the hypervisor or any other implementation needs to be re-architected to suit the new resource allocations and connectivity, and needs to match resource usage and provisioning to incoming requests.

Fig. 2-3 highlights the main concept of DS. The design in Fig. 2-3 uses a hybrid electro/optical switching fabric in addition to the disaggregated resources pools. Therefore, resources both in the electronic IP layer (packet-switched) and the optical layer (circuit-switched) are needed. The IP switches are to aggregate traffic from resources and each IP switch is connected to an optical switch, which is connected to other optical switches by optical fibre links. Optical fibres provide the large capacity and fast data transmission required to support the communication between the disaggregated resources. Intel Silicon Photonic connectors, SiPh [16], provide OEO processing for full wavelength conversion at each node.

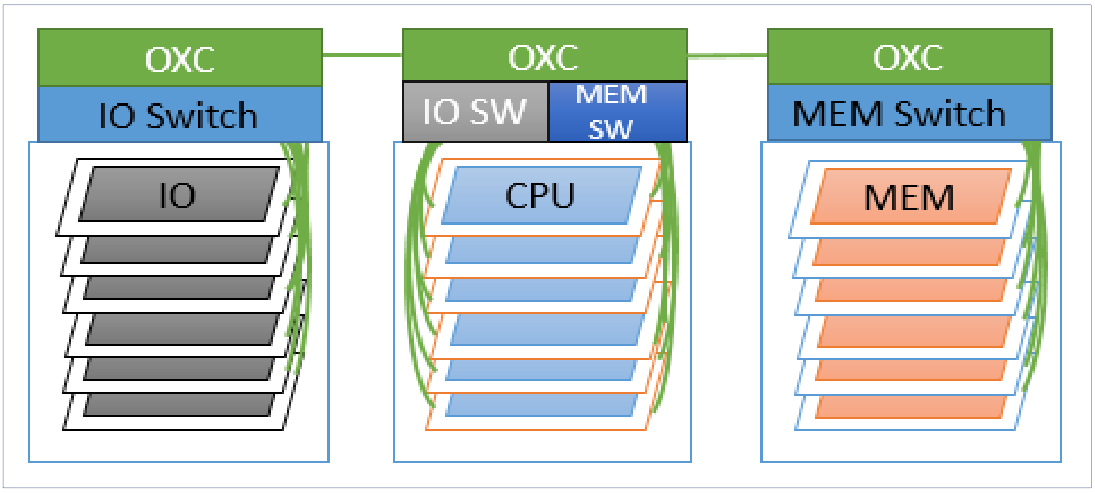


Fig. 2-3: DS architecture

2.5.1 Why disaggregate?

The growth in data centres and cloud workloads are spurring network vendors to rethink both network topologies and more specialised hardware resources; it marks data centre efficiency as a hot topic. Thus, systems' vendors are working on and making frequent announcements on providing new networking equipment such as silicon photonics [16], as well as new architectural designs such as disaggregated systems [32].

The emergence of hyperscale data centres over the last decade has motivated the development of specialised architectures that partition workloads. These workloads can be run on more optimal hardware that suits the workload requirements. The DS-based data centre has shown its potential in reducing capital expenses, by right-sizing compute, storage, and network resources to fit each workload requirement, and also by reducing power consumption and other operational expenses [6].

This architecture motivated the adoption of DS as a new server paradigm to be implemented in the biggest data centres. According to Intel, the goal for designing data centre structure at "rack-scale" is to minimise north-south flow by performing data locality, by managing the placement of compute and storage resources to a row or a small group of racks, and by enabling east-west bandwidth on that local collection of racks [6].

Throughout the literature, the DS design has significant and plentiful advantages that can be of direct impact in solving problems facing traditional data centre and server designs. For example, modularity, higher packaging and cooling efficiencies and higher resource utilisation are among the suggested benefits. Below is an overview of the main benefits that DS brings to data centres' administrators, owners, and users [48]:

Reduces total cost of ownership: DS provides a common platform that is flexible for different applications and services, making it simpler to configure components, in order to optimise performance, regardless of the nature of the demand.

Automation: DS drives continuous delivery of applications and services by composing pools of resources automatically optimised to support specific application and workload demands.

Agility: can be achieved with DS by providing dynamic and rapid provisioning of resources that are specially dedicated for an application and demand workload needs.

High scalability and resource utilisation: the DS design improves performance by matching workloads' demands to resources that best meet the required service quality and service levels and that increase resources utilisation; however, resource aggregation and pooling combines a cost-effective platform with large shared resources, thereby enabling the provisioning of resource-intensive applications.

2.5.2 Disaggregation technical challenges

This section discusses the trade-off between cost and performance in building a DS-based data centre system where resource modules in the data centre are pooled, for example, in memory-only, processor-only, IO-only, and storage-only chassis and racks. Analysis shows that the disaggregated data centre and server design will have a non-trivial performance penalty, and, considering data centre scale, this performance penalty could be serious. Increased latency and increased bandwidth cost are thought to be the main issues that could be a barrier to disaggregation [37].

Considering data centre traffic flows (see Fig. 2-4 [38]), the main flow is within the data centre traffic, which is a combination of both inter- and intra-rack communication flows. The inter-rack traffic is the traffic between racks in the data centre and is handled by the data centre communication fabric, and the intra-rack traffic is the communication traffic within the same rack, which is carried by the rack backplane or rack communication fabric. When disaggregating, all or part of the intra-rack traffic will traverse the data centre fabric adding extra latency, extra bandwidth requirements, and demanding additional or new control and management resources to be used. Thus, it is important to bear in mind that the load on the data centre communication network will increase, and hence consideration has to be given to the data centre communication network in terms of latency and throughput requirements [6].

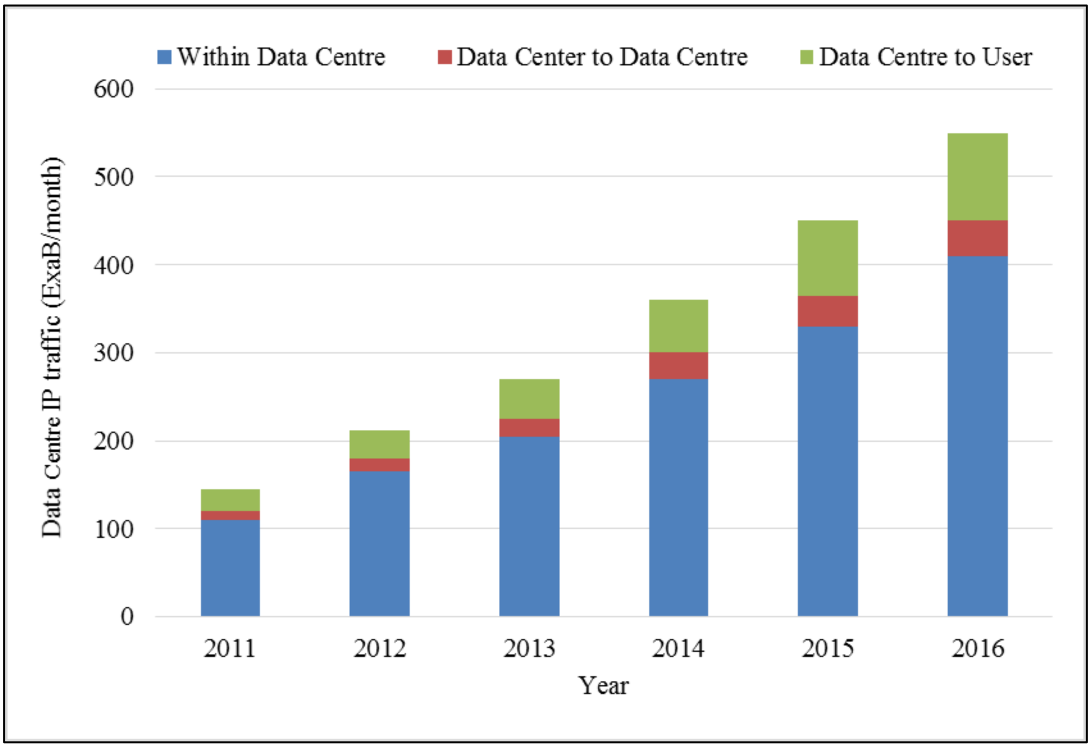


Fig. 2-4: Main data centre traffic flows [38]

Disaggregating the server resources will bring extra challenges regarding the data centre communication network and control and management systems; thus, the new control plane design must identify ways to handle the new resource connections and data centre network hierarchy, which must be reflected when planning the data centre scheduling algorithms and communication protocols [32].

An additional operational cost is exploiting new system components such as optical interconnects, which could be a source of increased cost to provide very fast communication fabric with huge bandwidth capacity [37].

To examine the feasibility and limitations of such a data centre communication network for the DS concept, Table 2-1 reviews the main communication types between resources within a server and a rack alongside their data rates, latency values, and energy consumption specifications [38].

Link	Latency	Data Rate	Energy
CPU-to-CPU Bus	5 ns	320 Gb/s	1 pJ/b
CPU-to-Memory Bus	10-50 ns	800 Gb/s/CPU	25 pJ/b
CPU-to-Peripherals Bus	1000 ns	128 Gb/s/device	35 pJ/b

Table 2-1: Typical requirements of a conventional server components [38]

Examining Table 2-1 clearly shows that the CPU-to-CPU traffic is latency sensitive and has a high throughput demand, which implies that it cannot be delegated to the external data centre network within the current network performance metrics. Therefore, such traffic should be kept within the same rack as much as possible. This problem can be downgraded by reducing CPU-to-CPU traffic by fulfilling each VM in a single CPU, by using enough CPU cores, or by keeping the communication among CPUs within the same CPU rack; however, the data centre scheduler will need to consider this when making scheduling decisions. The CPU-to-memory traffic is slightly less delay-sensitive than the CPU-to-CPU traffic, but demands a high bandwidth. However, these issues can be handled by considering optical interconnects, which provide means for fast and high throughput communications, and by using cache memory of high capacity or by adding more cache layers [39]. For the CPU-to-Peripherals traffic, such as network interfaces and disks, the required latency and bandwidth level are much less than the required values for the previous links, CPU-to-CPU and CPU-to-memory [41], which means that they can be accommodated within a unified network communication fabric, making their separation relatively simpler than the CPU-CPU and CPU-memory links [49]. Below are some metrics to consider for disaggregation [38]:

- Distance of the resource pool, which determines locality of the resources. Given that light travels in fibre at (2/3) of the speed of light in free space [50] and considering the maximum allowed communication latency, the maximum spanning distance between communicating racks should be upper-bounded by a specific threshold.
- Deployment efficiency which calls for detailed understanding of physical / logical partitioning of resources for floor planning within the rack, row, and data centre. Key questions include how and where the design can be improved and if the designs meet the objectives.
- Utilisation efficiency is essential and includes the time to deploy an application and compose a system, resource utilisation, scaling for pools, and scalability of the resource pool.

2.6 DS research and implementation efforts

Data centre disaggregation can fit into any of four main categories: on-board (disaggregating server resources), backplane/intra-rack, intra-data centre/inter-rack, and inter-data centre links [16]. This section reviews the work on the disaggregated data centre with regard to some of these categories.

On-board disaggregation means disaggregating server resources at the bus level as the board contains the server or servers, and connects to the backplane. Thus, this kind of disaggregation leads to pools of resources, such as CPU pools, memory pools, and IO pools, assembled inside a single rack or in separate racks.

Researchers from HP Labs, University of Michigan, and Hewlett-Packard Labs [51] [33, 48, 52], studied the software and systems implications of disaggregated memory. They developed a software-based prototype to emulate disaggregated memory by extending the Xen hypervisor to emulate a disaggregated memory design wherein remote pages are swapped into local memory on-demand upon access. After that, the group designed a new general architectural building block, a memory blade, which enables disaggregated memory across a system ensemble.

A research group from the Polytechnic University of Catalonia, Barcelona [41], presented an Integer Linear Programming (ILP) formulation to optimally allocate virtual data center (VDC) requests on top of an optically interconnected disaggregated data centre infrastructure. It considered the case where different resource blades can be grouped into racks hosting all types of resources where each rack holds the total aggregated computing resources of a rack instead of server boxes – resource pools inside the rack.

Disaggregating the backplane link that connects the racks contents to the top-of-rack switch results in an architecture where each board is a pool of CPU-only or memory-only resources.

In [34, 53-55], the authors presented an FPGA-based switch and interface card (SIC) and its application scenario in an all-optical, programmable disaggregated data centre network (DCN). It has been explained that this SIC card can be plugged into each server directly and it eliminates the need for the electronic top-of-rack switch while enabling direct intra-rack blade-to-blade communication to deliver ultralow chip-to-chip latency.

Disaggregating at the inter-rack link level, which involves links between racks throughout the data centre, is the most common approach declared by Intel’s Rack Scale Architecture [16] when Intel and Facebook declared their Open Compute Project at the 2013 Open Compute Summit. This type exemplifies the disaggregation of computing hardware from storage and networking hardware. They showed Intel’s photonic rack architecture to illustrate the total cost, design, and reliability improvement potential of a disaggregated rack environment of 100Gbps links between compute systems and a remote storage node.

Mellanox Technologies has shown in [49] its InfiniBand switching fabric to disaggregate the IO and storage subsystem from the main computing system. InfiniBand is a switch-based serial IO interconnect architecture operating at a base speed of 2.5 Gb/s or 10 Gb/s in each direction (per port). InfiniBand enables “Bandwidth Out of the Box”, spanning distances up to 17m over ordinary twisted-pair copper wires and it can span distances of several kilometres or more over common fibre cable.

A research group has presented in [56] the design, implementation, and evaluation of a PCIe-based rack area network system called Marlin. Marlin is a memory-based addressing model for both IO device sharing among multiple hosts and inter-host communications, designed to support the communications and resource-sharing needs of disaggregated racks.

In [57] the network traffic in 10 data centres belonging to three different types of organizations, including university, enterprise, and cloud data centres has been assessed to clarify the network-level traffic characteristics of the data centres. The authors have examined the range of applications deployed in these data centres and their placement, the flow-level and packet level transmission properties of these applications, and their impact on network utilization, link utilization, congestion, and packet drops. The observed traffic patterns implication for data centre internal traffic engineering as well as for recently proposed architectures for data center networks have been described as well.

In [58] the authors have reported upon the network traffic observed in some of Facebook’s data centres and their main focus was on the contrasting locality, stability, and predictability of network traffic in Facebook’s data centres, and their implications for network architecture, traffic engineering, and switch design. This work is to address the problem of the limited large-scale workload information available in the literature, in order to help researchers and industry practitioners when designing network fabrics to efficiently interconnect and manage the traffic within large data centres in an efficient fashion to satisfy the requirements of large cloud service providers.

Inter-Data Centre (peripheral bus): The work emphasised that this is not the normal inter-data centre link, but a link purposed specifically to share a resource among data centres, eliminating the need for each data centre to perform the same operation [6].

A group from University of California, Berkeley, Futurewei Technologies, Santa Clara, and ICSI, Berkeley, CA [6], has explored key questions around the data centre network support for the disaggregated server, such as the bandwidth and latency demands due to disaggregation, the key application and hardware parameters that affect these demands and ways to meet these demands in a trial to draw the general headlines for implementing this design. They have concluded that the key enabling or blocking factor will be the network since communication that was previously contained within a single server now traverses the data centre fabric.

The authors in [42] proposed a cloud architecture that disaggregates resources into virtual resource pools to provision virtual machines with the right amount of resources. Their cloud architecture creates a distributed and shared physical resource layer by providing virtual layer and cloud resource aggregation layer between applications and physical servers in real time.

At a workshop at OFC 2015, a group of companies including Facebook, Intel CIAN, IBM, Infenera, MIT, Mellanox, Corning, and Samtec presented their visions, work, and some metrics for the disaggregated data centre [38]. They described their designs and visions for the DS with complete design structures that show the implications of the DS paradigm in future data centres [38].

In [5] a collaboration between Tencent and Intel used a proof-of-concept demonstrator to show that disaggregated data centre and resource pooling, even in the early stages of development, can introduce better performance and reduced power consumption, and can improve the end users experience. They address the motivation for server disaggregation and explain the main challenges and the findings from the Tencent proof-of-concept.

A Group has studied in [37] the trade-off between cost and performance in building a disaggregated memory system. The group constructed a simple cost model that compares the savings expected from a disaggregated memory system to the expected costs, such as latency and bandwidth costs, and then identified the level at which a disaggregated memory system becomes cost competitive with a traditional direct attached memory system.

A software-defined architecture for the next generation data centre, dRedBox has been presented in [59], and a design prototype hardware architecture has been presented too. For the design, SoC-based micro-servers, memory modules, and accelerators are placed in separated modular server trays interconnected via a high-speed, low-latency opto-electronic system fabric, and allocated in arbitrary sets in order remove the limitations of the monolithic common design of servers.

In [36], a group presented preliminary work that shows that optical interconnect technology enables CPUs and local memory to be placed metres away from each other without sacrificing bandwidth. With alternative architecture options for the server memory to CPU interconnect using an optically attached memory (OAM) system, the group showed that, with optical interconnect technology, bandwidth can remain high, even though CPU and memory are separated.

2.7 Summary

This chapter presented a general review of energy-efficient approaches in data centres, and the motivations for the DS design by giving a brief review of the current conventional data centre with its drawbacks. It presented virtualisation and resource consolidation as a solution for these drawbacks and gave a review of the DS design, with its main benefits and promises, followed by the technical challenges facing this emerging data centre architecture. Finally, it gave a complete literature review that summarised the work done previously in the literature on disaggregated data centres, covering all the disaggregation levels and scenarios done previously. Unlike the work done in the literature, our aim is to provide detailed analyses of the impact of the physical disaggregation of the server resources on the overall data centre energy consumption by considering the VM allocation and resource provisioning in this new data centre server paradigm. This will be considered in the remaining chapters in this thesis.

Chapter 3: Energy Efficient Resource Provisioning in Disaggregated Server Design

3.1 Introduction

With the rapid growth of data and processing intensive applications and the shift towards the cloud computing model, serious concerns are raised about the power consumption of data centres. To improve the energy efficiency of data centres, architecture design and hardware design must move in concert. In this chapter we evaluate the energy efficiency of VM placement in the data centre considering the DS concept as a new design of future data centres. This approach can introduce significant improvement for the data centre design where the previously "server dedicated components" are now shared among different servers. Based on the model insights, we develop heuristics to enable the implementation of the model concepts in real time environments. Different VM types have been considered to show the impact on the performance and energy efficiency of DS based data centres. We propose three types of VMs, memory intensive (MI) VMs, processing intensive (PI) VMs and IO intensive (IOI) VMs.

3.2 Resource Provisioning MILP Model

In this section, a data centre comprises of a number of heterogeneous resources chosen from a set of known and well-characterised components (processors, memory and IO cards). Note that here, and for comparison purposes, we calculate the power consumption of only these resources under the DS concept and under the conventional data centre, and leave an in-depth full data centre power consumption evaluation in the upcoming chapters. Given the requested VMs, the MILP model places VMs in the optimal location for minimum power consumption and tries to fully utilise the available resources by packing the resources (processors, memory and IO) with as many VMs as they can hold, in order to minimise the data centre's power consumption by reducing the number of working resources.

The power consumption of a resource is modelled in equation (3-1). Each resource power consumption is composed of two parts, a fixed factor, $XMin_j$, which represents the idle power of the resource X, and a variable power term, ΔX_j, (equation (3-2)) linearly related to the resource utilisation δX_j [60].

$$Power = XMin_j + \Delta X_j \cdot \delta X_j \tag{3-1}$$

$$\Delta X_j = \left(XMax_j - XMin_j \right) \tag{3-2}$$

where $XMax_j$ and $XMin_j$ are the maximum active power and the idle power of the j^{th} resource respectively. In [60] experiments have been conducted on several thousands of nodes under different types of workloads and they have shown that the predicted power by this model accurately predicts the power consumption by server systems with the error below 5% for this linear server power model.

An idle power is defined as the power consumed by the resource when powered, with all links connected (and operating system driver loaded) but without the processing or transmission of any data. In practice it is the least amount of power required to keep the resource functional. Maximum power consumption is obtained by measuring the resource power usage while working at its full capacity. In our work we assume that a resource is turned off instead of staying in the idle state when it is not being utilised by any VM. Below are the parameters and variables used in the model.

Sets:

VM	Set of virtual machines
PR	Set of processors
MR	Set of memories
IOR	Set of IO modules

Parameters:

NPR	Total number of processors
NMR	Total number of memory modules
$NIOR$	Total number of IO modules
NVM	Total number of VMs
ΔP_j	Power delta of processor *j*
ΔM_j	Power delta of memory *j*
ΔIO_j	Power delta of IO module *j*
P_j	The processing capabilities of processor j (GHz)
M_j	Capacity of memory j (GByte)
IO_j	Total bit rate of NIC port j (Gb/s)
VP_i	Processing demands of VM i (GHz)
VM_i	Memory demand of VM i (GByte)
VIO_i	IO demand of VM i (Gb/s)
$PMax_j$	Maximum power consumption of processor j (W)
$PMin_j$	The idle power consumption of processor j (W)

$MMax_j$ Maximum power consumption of memory j (W)

$MMin_j$ The idle power consumption of memory j (W)

$IOMax_j$ Maximum power consumption of NIC port j (W)

$IOMin_j$ The idle power consumption of NIC port j (W)

W Very large number

e Very small number

SLA Agreed percentage of served VMs according to Service Level Agreement (SLA)

Utl The maximum allowed utilisation of each of the resources of the data centre

Variables :

θP_{ij} Portion of the processor j capability allocated to request i

θM_{ij} Portion of the memory j allocated to request i

θIO_{ij} Portion of the the j^{th} IO port module allocated to request i

δP_j The fractional utilisation of processor j

δM_j The fractional utilisation of memory

δIO_j The fractional utilisation of IO port module

YP_{ij} YP_{ij} =1 if processor j hosts request i, otherwise YP_{ij}=0

YM_{ij} YM_{ij} =1 if memory j hosts request i, otherwise YM_{ij}=0

YIO_{ij} YIO_{ij} =1 if port j hosts request i, otherwise YIO_{ij}=0

KP_i KP_i =1 if request i processing requirement is being served, KP_i=0 if it is blocked

KM_i KM_i =1 if request i memory requirement is being served, KM_i =0 if it is blocked

KIO_i KIO_i =1 if request i IO requirement is being served, KIO_i=0 if it is blocked

XP_j XP_j =1 if processor j is active, otherwise, XP_j =0

XM_j XM_j =1 if memory j is active, otherwise, XM_j =0

XIO_j XIO_j =1 if module j is active, otherwise, XIO_j =0

The power consumption of resources in a data centre based on the DS architecture and due to the resource provisioning is composed of:

1) The power consumption of active processors

$$\sum_{j \in PR} ((XP_j \cdot PMin_j) + (\Delta P_j \cdot \delta P_j)) \qquad (3\text{-}3)$$

2) The power consumption of active memories

$$\sum_{j \in MR} ((XM_j \cdot MMin_j) + (\Delta M_j \cdot \delta M_j)) \qquad (3\text{-}4)$$

3) The power consumption of active IO ports

$$\sum_{j \in IOR} ((XIO_j \cdot OMin_j) + \left(\Delta \mathrm{IO}_j \cdot \delta IO_j\right) \qquad (3\text{-}5)$$

Objective: minimise:

$$\sum_{j \in PR} ((XP_j \cdot PMin_j^P) + (\Delta P_j \cdot \delta P_j)) +$$

$$\sum_{j \in MR} ((XM_j \cdot PMin_j^M) + (\Delta M_j \cdot \delta M_j)) +$$

$$\sum_{j \in IOR} ((XIO_j \cdot PMin_j^{IO}) + (\Delta \mathrm{IO}_j \cdot \delta IO_j)) \tag{3-6}$$

Note that objective (3-6) minimises the power consumption by consolidating resources into the minimum possible number of resources (due to the presence of an idle power component), so the number of the functioning resources is minimised.

The model is subject to a number of constraints as follows:

Capacity Constraints:

$$\delta P_j = \sum_{i \in VM} \theta P_{ij} \ \leq UTL \qquad \forall\, j \in PR \tag{3-7}$$

$$P_j \cdot \theta P_{ij} = VP_i \cdot YP_{ij} \qquad \forall\, i \ \in VM, j \ \in PR \tag{3-8}$$

$$\theta P_{ij} \leq W \cdot YP_{ij} \qquad \forall\, i \ \in VM, j \ \in PR \tag{3-9}$$

$$\theta P_{ij} \geq e + YP_{ij} - 1 \qquad \forall\, i \ \in VM, j \ \in PR \tag{3-10}$$

$$\delta M_j = \sum_{i \in VM} \theta M_{ij} \leq Utl \qquad \forall\, j \in MR \tag{3-11}$$

$$M_j \cdot \theta M_{ij} = VM_i \cdot YM_{ij} \qquad \forall\, i \ \in VM, j \ \in MR \tag{3-12}$$

$$\theta M_{ij} \leq W \cdot YM_{ij} \qquad \forall\, i \ \in VM, j \ \in MR \tag{3-13}$$

$$\theta M_{ij} \geq e + YM_{ij} - 1 \qquad \forall\, i \ \in VM, j \ \in MR \tag{3-14}$$

$$\delta IO_j = \sum_{i \in VM} \theta IO_{ij} \leq Utl \qquad \forall\, j \in NIOR \tag{3-15}$$

$$IO_j \cdot \theta IO_{ij} = VIO_i \cdot YIO_{ij} \qquad \forall\, i \ \in VIO, j \ \in IOR \tag{3-16}$$

$$\theta IO_{ij} \leq W \cdot \ YIO_{ij} \qquad \forall\, i \ \in VM, j \ \in IOR \tag{3-17}$$

$$\theta IO_{ij} \geq e + \ YIO_{ij} - 1 \qquad \forall\, i \ \in VM, j \ \in IOR \tag{3-18}$$

Constraint (**3-7**) calculates the total processing utilisation of each processor and ensures it is less than the allowed maximum utilisation, constraint (**3-8**) calculates the CPU utilisation of each processor per allocated VM, and constraints (**3-9**) and (**3-10**) allocate each VM to a certain processor.

Constraints (**3-11**)-(**3-14**) repeat steps (**3-7**)-(**3-10**) but for the memory resources and constraints (**3-15**)-(**3-18**) repeat the same steps but for the IO modules.

Slicing Constraints:

$$\sum_{j \in PR} YP_{ij} \leq 1 \qquad (3\text{-}19)$$

$$\forall i \in VM$$

$$\sum_{j \in MR} YM_{ij} \leq 1 \qquad (3\text{-}20)$$

$$\forall i \in VM$$

$$\sum_{j \in IOR} YIO_{ij} \leq 1 \qquad (3\text{-}21)$$

$$\forall i \in VM$$

Constraints (3-19)-(3-21) ensure that the model serves each VM using only one processor, one memory and one IO port respectively. If multiple VM copies or VM slicing is required, equations (3-19)-(3-21) should be upper bound by an appropriate number greater than 1.

SLA Constraints:

$$\sum_{i \in VM} KP_i \geq NVM \cdot SLA \qquad (3\text{-}22)$$

$$KP_i \leq \sum_{j \in PR} YP_{ij} \tag{3-23}$$

$$\forall\, i \in VM$$

$$W \cdot KP_i \geq \sum_{j \in PR} YP_{ij} \tag{3-24}$$

$$\forall\, i \in VM$$

$$\sum_{i \in VM} KM_i \geq NVM \cdot SLA \tag{3-25}$$

$$KM_i \leq \sum_{j \in MR} YM_{ij} \tag{3-26}$$

$$\forall\, i \in VM$$

$$W \cdot KM_i \geq \sum_{j \in MR} YM_{ij} \tag{3-27}$$

$$\forall\, i \in VM$$

$$\sum_{i \in VM} KIO_i \geq NVM \cdot SLA \tag{3-28}$$

$$KIO_i \leq \sum_{j \in IOR} YIO_{ij} \tag{3-29}$$

$$\forall\, i \in VM$$

$$W \cdot KIO_i \geq \sum_{j \in IOR} YIO_{ij} \tag{3-30}$$

$$\forall\, i \in VM$$

Constraints (3-22)-(3-24) guarantee that the number of served VMs is greater than a pre-specified value according to SLA, and constraints (3-25)-(3-27) provide similar guarantees as in (3-22)-(3-24) but for the memory resources, while constraints (3-28)-(3-30) are concerned in a similar fashion with the IO resources.

Active Resources:

$$XP_j \leq W \cdot \ \delta P_j \qquad (3\text{-}31)$$

$$\forall\, j \in PR$$

$$W \cdot XP_j \geq \delta P_j \qquad (3\text{-}32)$$

$$\forall\, j \in PR$$

$$XM_j \leq W \cdot \delta M_j \qquad (3\text{-}33)$$

$$\forall\, j \in MR$$

$$W \cdot XM_j \geq \delta M_j \qquad (3\text{-}34)$$

$$\forall\, j \in MR$$

$$XIO_j \leq W \cdot \delta IO_j \qquad (3\text{-}35)$$

$$\forall\, j \in IOR$$

$$W \cdot XIO_j \geq \delta IO_j \qquad (3\text{-}36)$$

$$\forall\, j \in IOR$$

Constraints (3-31)-(3-36) find the active resources, if the resource utilisation δx is larger than zero then the indicator X is 1, otherwise X is zero.

3.3 Energy Efficient Resource Provisioning Heuristic

In this section we develop a heuristic that mimics, in real time, the behaviour of the MILP as heuristics typically allow larger problems (here placing VMs in DS data centre) to be handled for a given amount of computational resources compared to the MILP, due to their lower computational complexity. The energy efficient resource provisioning with DS (EERP-DS) mechanism allows the data centre to serve the incoming VMs assuming SLA value of 100%. The flowchart in Fig. 3-1: EERP-DS heuristic shows the heuristic which aims to pack incoming VM requests into the minimum number of resources so that minimal power is consumed through packing and by powering off un-utilised resources.

Here we define the power factor (PF) which captures the energy efficiency of the different disaggregated resources. The PF is the resource ΔP_j divided by the resource capacity C_j. Therefore, low PF values reflect high energy efficiency. Note that if the idle power is very close to the maximum power for a given resource, then ΔP_j can be very small while in reality the resource power consumption may be high. Deceptively in this case the power factor can be low while the resource is not energy efficient. In our case, the practical power consumption values we used did not lead to this situation arising, but it is a condition that has to be checked. The heuristic first creates sorted lists for each resource type in an ascending order according to the values of their PF as resources with lower PF values are preferred. In a case where two resources have the same PF value, the resource with the highest capacity is listed first. At the end of this stage, there will be three sorted lists: processor list, memory list and IO module list.

For each VM, starting from the top of the sorted lists, the heuristic then picks one resource from each sorted list and checks the chosen resource to determine if there is enough capacity on that resource type to serve the current VM request. If any of the resources (processors, memory, or IO modules) cannot serve any more VM requests, then the heuristic will proceed to the next resource in the corresponding sorted list and test it.

First the chosen processor from the processor list is tested. If the current tested processor does not have enough capacity then the heuristic will pick up the next processor from the processor sorted list, and if it has the ability to host the VM, then the heuristic will test the selected memory from the memory list. Again, if the memory does not have enough capacity to serve the VM under consideration, the next memory must be retrieved from the memory list; otherwise the chosen IO module must be tested. Finally, if the IO module can accommodate the network traffic requirements of the VM under consideration, it will be used directly; otherwise the next IO module must be retrieved from the IO list. The heuristic then allocates the selected resources to the current VM and updates these resources' remaining capacities. The heuristic proceeds by reading the next VM to be served and repeats the same steps to serve all the incoming VMs.

3.4 Evaluation and Results

In this section we evaluate the MILP model and the EERP-DS heuristic and compare their performances to the CS approach considering the resource provisioning when presented with the same set of VMs.

We built a DS architecture for evaluation by disaggregating the IBM system X3650 M3 server [61]. The IBM X3650 M3 server supports 11 processor types with different number of cores, and power characteristics. Table 3-1 shows the 11 processor types with their maximum power draws. The IBM X3650 M3 server comes also with three standard memory bandwidth rates. The memory is a DDR3 SDRAM with three bandwidth values, where the evaluation in [60] for DDR3, gives the memory power consumption, see Table 3-2.

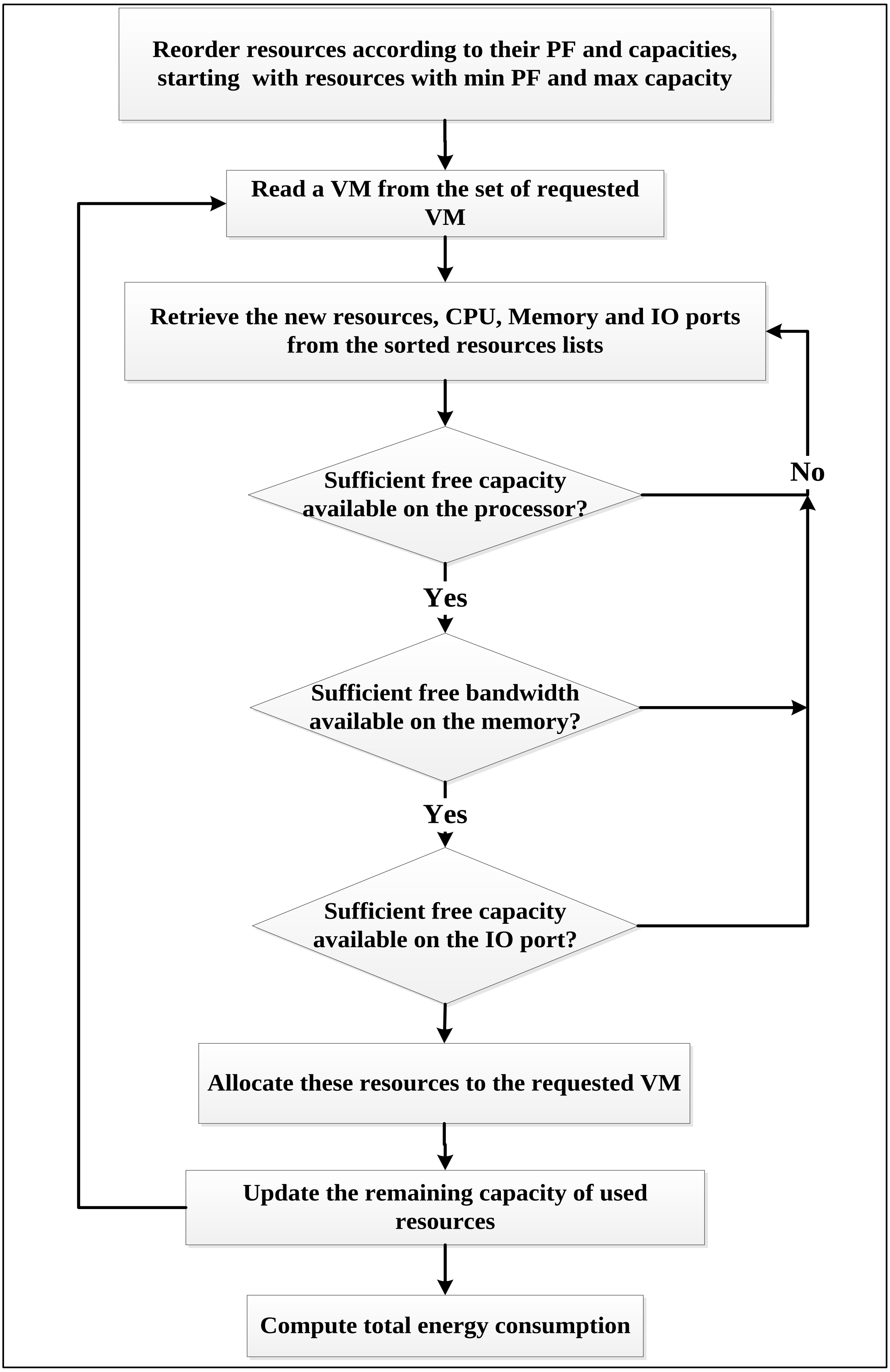

Reorder resources according to their PF and capacities, starting with resources with min PF and max capacity
Read a VM from the set of requested VM
Retrieve the new resources, CPU, Memory and IO ports from the sorted resources lists
Sufficient free capacity available on the processor?
No
Yes
Sufficient free bandwidth available on the memory?
Yes
Sufficient free capacity available on the IO port?
Allocate these resources to the requested VM
Update the remaining capacity of used resources
Compute total energy consumption

Fig. 3-1: EERP-DS heuristic

In practice, and although the data centre may be fully disaggregated into resources (processors, memory, IO), these resources are in practice not a single fully reconfigurable block. Instead, to ease the resource handling, to ease the mounting of resources in racks and to ease the communication requirements, pools of resources can be defined. Note that a pool that has a single processor, a single IO card and memory is a conventional server. We consider in this chapter a medium size pool of processors made of 6 processors of each of the 11 types in Table 3-1. We plan to optimise the size of the pool in future work. The active power consumption of the processors is set depending on the system described in [61]. The processor idle power consumption is set to 0.7 of its power consumption when fully utilised [60].

A pool of memory is made of 3 memory types, see Table 3-2. It contains a total of 66 memory units where 36% of the used memories operate at 4 GB/s, and 28% operate at 8 GB/s, while the remaining 36% operate at 24 GB/s. The system is completed with a pool of 66 NIC ports where half of the ports support a rate of 1 Gb/s and the remaining are 10 Gbps, see Table 3-3. The numbers of the three used resources are equal in order to have a fair comparison with the CS data centres.

We considered two types of IO ports in the evaluation, 1 Gb/s and 10 Gb/s data rates, and their power consumption is based on the work in [62]. As a conservative case, we consider the situation where an idle port consumes 0.7 of the power consumed when it is fully utilised. The maximum power of each port type is given in Table 3-3. Some of the input parameters of the model are given in Table 3-4.

A VM is characterised by three main requirements, CPU requirement $\mathrm{CP_i}$, memory requirement $\mathrm{CM_i}$, and IO requirement $\mathrm{CIO_i}$. In view of the available resources capacity, the request types under consideration and the SLA violation avoidance needs, we have estimated the amount of resources that each VM type needs and proposed three types of VMs, MI VMs, PI VMs and IOI VMs. The details of the resource requirements for each VM type are shown in Table 3-5.

Processors capacities (GHz)	**Processors Max. power consumption (W)**
3.46	130
3.6	130
2.93	130
2.66	95
3.2	95
2.4	80
2.53	80
2.13	80
2.26	60
2.13	40
1.86	40

Table 3-1: Power consumption and capacity of IBM X3650 M3 server

Memory Data Rate (GB/s)	**Memory Max. Power Draw (W)**
4	5.12
8	10.24
24	30.72

Table 3-2: Memory data rate and power consumption

The results in Fig. 3-2 are based on our MILP optimisation and compare the power consumption of memory intensive virtual machines if these VMs are implemented using a conventional data centre design CS and if they are implemented using a DS design. Fig. 3-2 shows that VMs implemented in a data centre using the DS approach achieve a best average power saving of 49% (given our set of parameters) when all the VM requests are MI.

NIC Port Rate (Gb/s)	NIC Port Max. Power Draw (W)
1	1.9
10	21.4

Table 3-3: Network interface cards data rates and power consumption

Number of processors	66
Number of memory Chips	66
Number of NIC ports	66
Number of processors types	11
Number of memory types	3
Number of NIC port types	2
Utl value	0.9
SLA value	100%

Table 3-4: Input parameters used in the optimisation model

Request Type	Processing (G CPU Cyclesps)	Memory (GB)	IO (Gbps)
Processing Intensive	1-3.3	0.05-0.2	0.05-1
Memory Intensive	0.2-1	1-4	0.05-1
IO Intensive	0.2-1	0.1-0.5	1-4

Table 3-5: Resources required by each VM type

Considering the MI VMs, the memory requirements are the highest and therefore in these VMs memory is used to a much larger extent compared to other processing and IO resources utilisation, see Table 3-5. Thus, in the “traditional single box” server, we call here CS, the memory requirement of the VM may cause a whole server to be dedicated to a single VM, as the memory of the server cannot accommodate more VMs. This comes at the expense of losing free space in the processor and the IO port, thus most of the servers will host only one VM. However, in DS, the memory, the processor and the IO port are limited by the server box boundaries, thus the spare processor and the IO ports capacities can be accessed allowing additional VMs to be accommodated and leading to improved resource utilisation.

It is clear that processors consume the most power and memory resources consume the least power, while IO ports power consumption lies between the two. Thus, with MI VMs, the number of working processors and IO ports in the DS are much less than CS because they can be used efficiently in the DS architecture, which in turn will result in high power saving.

We optimised the DS infrastructure under processing intensive VM requests, and here we achieve about 11% power saving as shown in Fig. 3-3. With the PI VM requests the processing requirements are higher than memory and IO requirements, thus a large number of processors, which consume the most power, will be used in both CS and DS. Thus, the power saving in this case will come from the memories and the IO ports, which explains why we observe a smaller power saving.

For IOI VM requests about 24% of the power consumed by CS will be saved when implementing DS approach, see Fig. 3-4. With IOI VM type, the bottle neck is usually the IO requirements, thus IO ports will not be used efficiently, which affects the use of the whole server in the CS. With IOI VM type, the bottle neck is usually the IO requirements, thus IO ports will not be used efficiently, which affects the use of the whole server in the CS. However, with the DS the number of working processors and memories is less than the number of working processors and memories in CS, which results in a good power saving. Nonetheless, this will be less than the power saving achieved when serving MI VMs, because the power saving will come from the latter’s efficient use of processors and memories. Thus, serving MI requests will be the less efficient scenario with the CS, and this will lead to the maximum amount of power saved with the DS architecture. The IOI scenario is an intermediate case and serving PI requests will result in the minimum power saving.

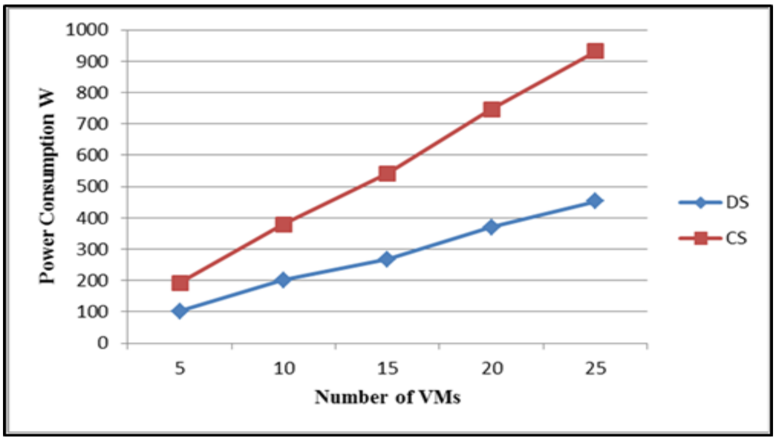


Fig. 3-2: Power consumption of MI VMs

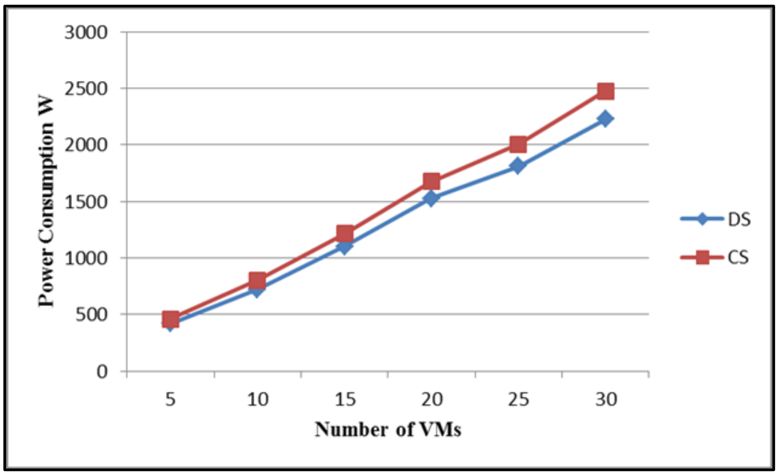


Fig. 3-3: Power consumption of PI VMs

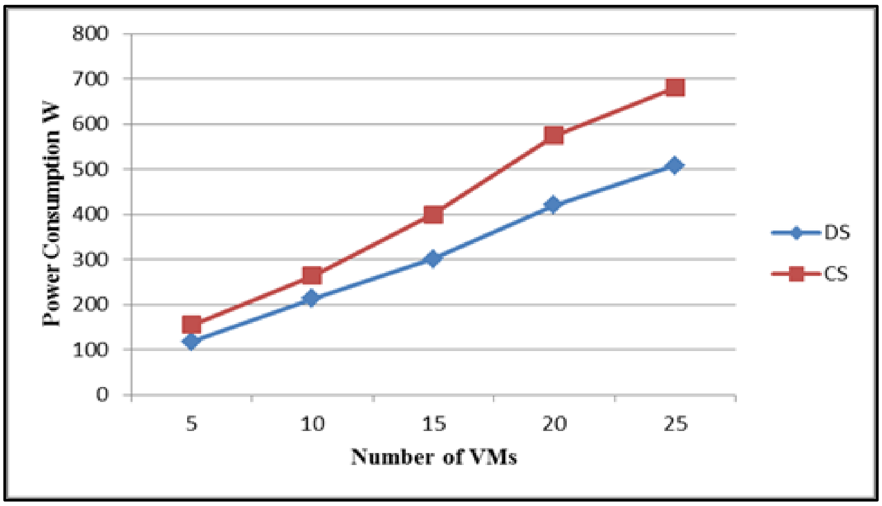
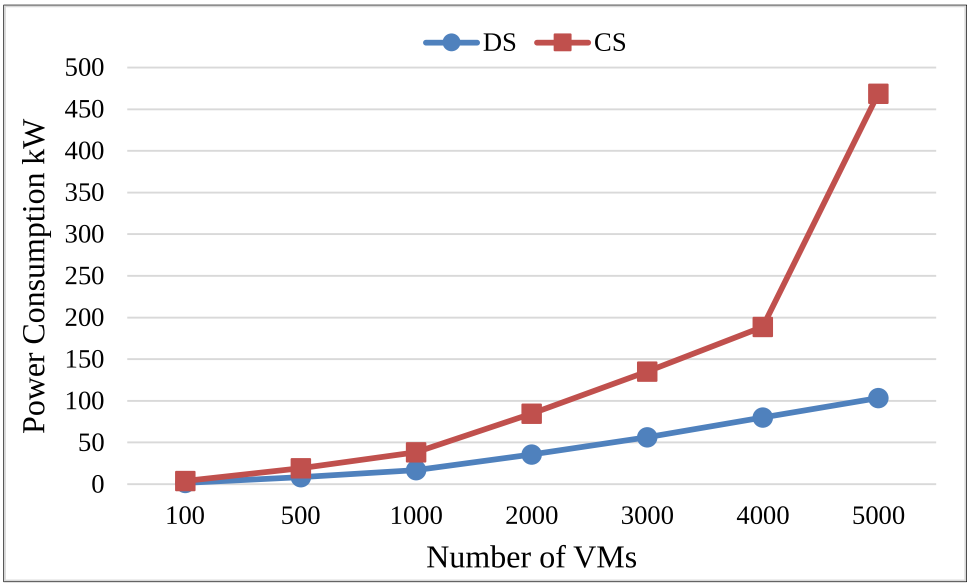
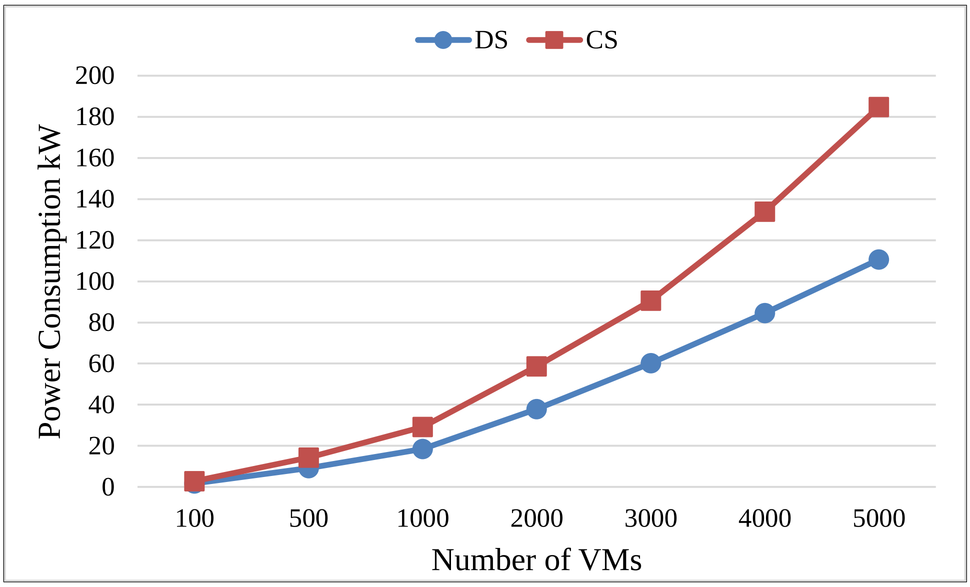


Fig. 3-4: Power consumption of IOI VMs

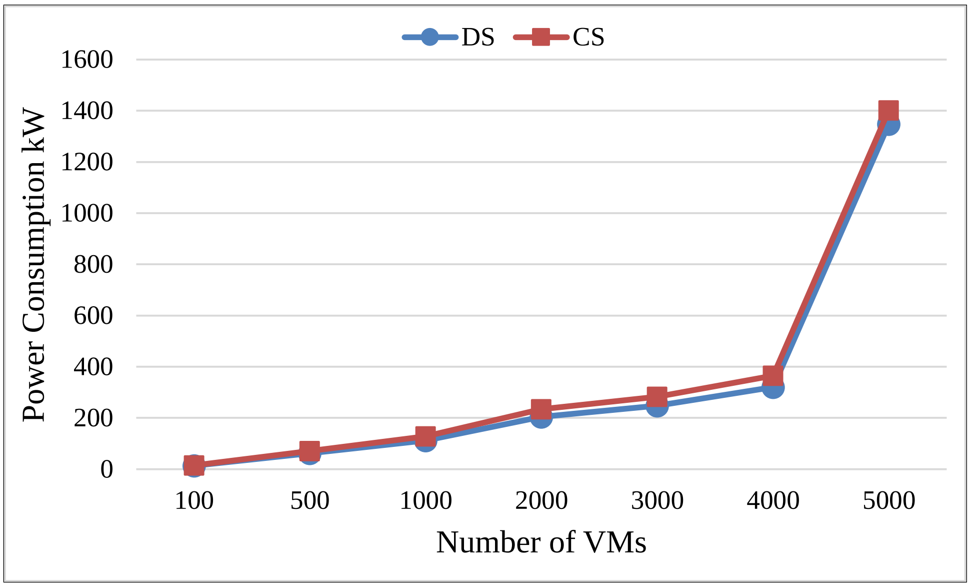
For the heuristic we use the same parameters used in the evaluation of the MILP model but with higher numbers of resources to expand the size of the resource pools under consideration and show the DS potential when implemented in big data centre. Our EERP-DS heuristic shows a very comparable results to the MILP power consumption such that MILP power consumption is only 9% less than heuristic power consumption. Reducing the cooling power consumption is implicitly considered through limiting the resource's maximum utilisation.



(a) MI requests power consumption



(b) IOI requests power consumption



(c) PI requests power consumption

Fig. 3-5: Heuristics power consumption for CS and DS

Considering the MI VMs in CS, the memory requirements are the highest and may cause a whole server to be dedicated to a single VM as the memory of the server cannot host more VMs. To host more MI VMs, more servers need to be powered on with high memory utilisation and low CPU and IO port utilisation. The spare capacity in the CPU and IO ports of those servers cannot be made available to other VMs as these resources are physically bound inside servers that have already consumed their memory resource. However, in DS, memories, processors and IO ports are not limited by the server box boundaries, thus the spare processors and IO port capacities can be accessed by additional VMs to be accommodated, thereby leading to improved resource utilisation. Therefore, with MI VMs, the number of working processors and IO ports in the DS based data centre are much lower than in CS, which in turn results in a high average power saving of 60% for DS compared to CS (given our set of parameters), see Fig. 3-5.a.

In the same manner, considering IOI VMs in CS results in inefficient use of processing and memory resources, which is not the case in DS. Thus, with the DS the number of working processors and memories is less than the number of working processors and memories in CS, which results in an average power saving of 36%, see Fig. 3-5.b. Nonetheless, this will be less than the power saving achieved when serving MI VMs due to the low power consumption of the efficiently utilised memories compared to the IO ports' power consumption. Similarly, considering the PI VMs results in an average power saving of 11% as shown in Fig. 3-5.c. The power saved in this case comes from the efficient usage of low power consuming memories and IO ports, which explains why we observe a smaller power saving.

3.5 Summary

In this chapter, we have investigated the energy efficiency of VM placement in data centres based on the DS approach and evaluated the power saving of this new server paradigm. The approach considered enables the separation of the computing, memory, storage and network resources of the server leading to better resource utilisation by "composing on the fly" servers with the exact required processing, memory and IO capabilities to accommodate the virtual machines or tasks of interest. We have developed a MILP optimisation model, which optimally places VMs in the disaggregated data centre with the objective of minimising the power consumption. We have compared a data centre with DS architecture to a data centre using the normal rack of server units considering the VM placement and resource provisioning operations. To gain a good view for the operation of the proposed approach, we have considered three types of VMs: PI, MI and IOI in the model. The results show that with MI applications, the DS approach achieves the maximum power saving. When serving MI requests the MILP achieves (in DS versus CS) an average power saving of 49% and for IOI requests the average power saving is 24%, while serving PI request results in 11% average power saving under the set of typical parameters and conditions we considered. For real time implementation, a simple heuristic is developed based on the model insights with comparable power values with the MILP model. The heuristic achieved power savings of 38% (MILP 49%), 18% (MILP 24%), and 10% (MILP 11%) for the MI, IOI, and PI respectively. These results considered 20 VM requests (in DS versus CS) due to the high computational complexity of MILP. Furthermore, EERP-DS heuristic results showed that for *extended* data center size and serving big numbers of VMs (5000 VMs) the average power savings were 60% when serving MI requests, 36% for IOI requests and 11% when serving PI requests (under the set of typical parameters and conditions we considered).

Chapter 4: Disaggregated Server Design

4.1 Introduction

In this chapter we present our photonically enabled design, which uses Intel's new photonic interconnect approach [16]. The design shares the memory and IO modules among multiple processors to form resource pools connected through a distributed switching fabric. The concept of distributed switch functionality and modular architecture design supports high granular resource deployment approaches which allow for greater resilience, upgradability, and scaling up a VM can be done directly and seamlessly with this modular architecture. This architecture can potentially enable re-partitioning of the resources in such a way that system resources can be better shared between different compute elements.

Based on the ideas and guidelines given in [16] and [38], we have designed our modular architecture for the disaggregated server. We propose a new interconnect topology to support the communication between the disaggregated server blocks. Given a data centre system, the main communication components are inter and intra rack communications. Considering the inter rack communications, the communicating units (e.g. servers or disaggregated devices) are located in different racks, while, for the intra rack communication, these communicating units are located within the same rack. Thus, for DS the communication that were confined inside single servers are now an inter-rack traffic and traverse the whole data centre communication fabric.

To show the functionality of the suggested architecture and clarify its performance, we will define each type of these communications while we describe our design. Also we will show how each part of the architecture will perform its assumed function to support these communications. The following sections detail all the distributed components, focusing on each part of the architecture and the interconnecting components.

4.2 Design Description

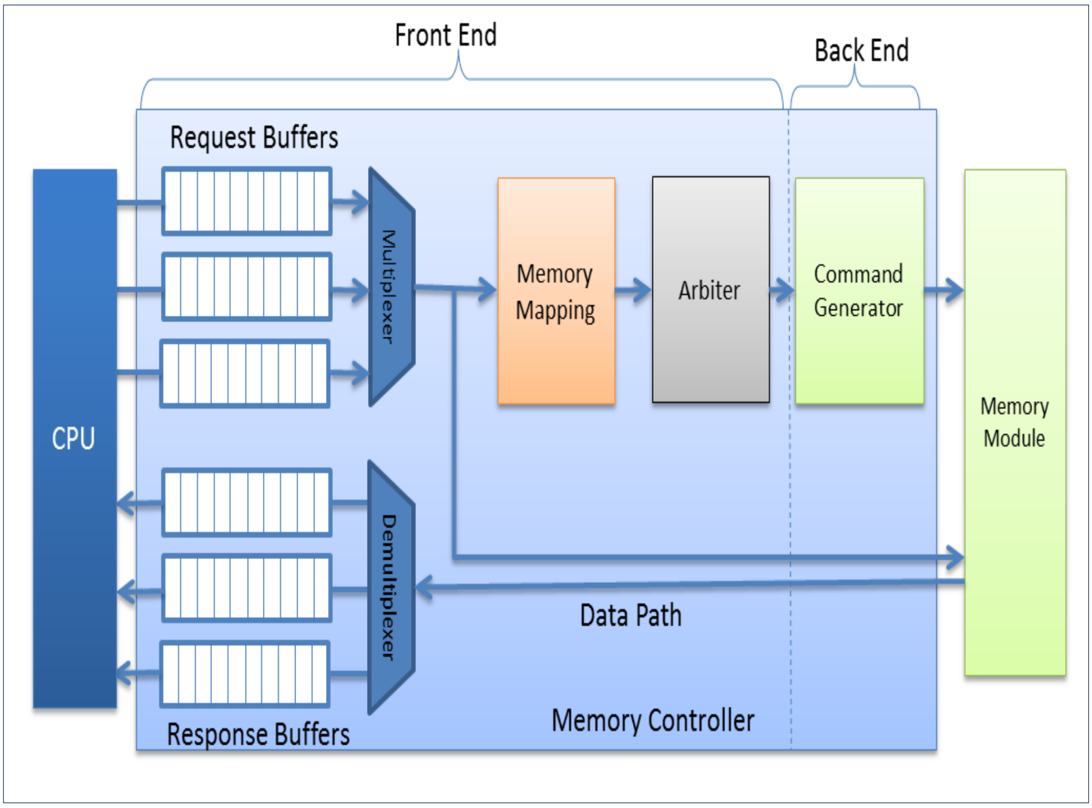
In this section we highlight the main components required to establish an end to end connection and guarantee fast and durable communication path from source to destination based on our novel design for the DS architecture. In the literature, very few have considered full server disaggregation, the most common work is by disaggregating the IO and storage only rather than memory disaggregation, or by considering virtual resource sharing as a mean for disaggregation. We are the first to disaggregate without changing the CPU interface as we are considering the splitting of the same memory controller and letting the CPU, memory and IO to see the same old interfaces. The first component to consider is the memory controller [63]. In this design, we split the memory controller into three functional blocks. The first block is attached to the CPU itself, named CPU attached memory controller (CPUMC), and the second block is general to the whole memory rack, named middling memory controller (MMC), while the last block is attached to the memory module directly and is the memory attached memory controller (MEMC). Before we present our new disassembled memory controller (DMC), we need to examine the current classical memory controller. Fig. 2-1.a displays the complete architecture of the current memory controller [63]. It is mainly composed of two segments, the front end and the back end. While the front end is independent of the memory module type and provides an interface to the back end segment of the memory controller, the back end is memory type dependent. It translates requests from the front end to the target memory.

Functions such as buffering and instruction mapping and sequencing are performed in the front end part. This consists of buffers to store memory requests and responses. The buffers are attached to multiplexers/demultiplexers to send/receive one data word at a time [64]. The memory mapping decodes the memory address from the CPU address view to the memory address view (virtual memory to physical memory) and the arbiter decides the sequence in which requests from the CPU can access the memory modules. Thus, memory access requests are queued in the arbiter. The back end command generator generates the commands for the target memory. It is memory type dependent, thus we will keep it attached to the memory, and it is customised to handle different timings so that different components having different clock rates can access the same memory module.

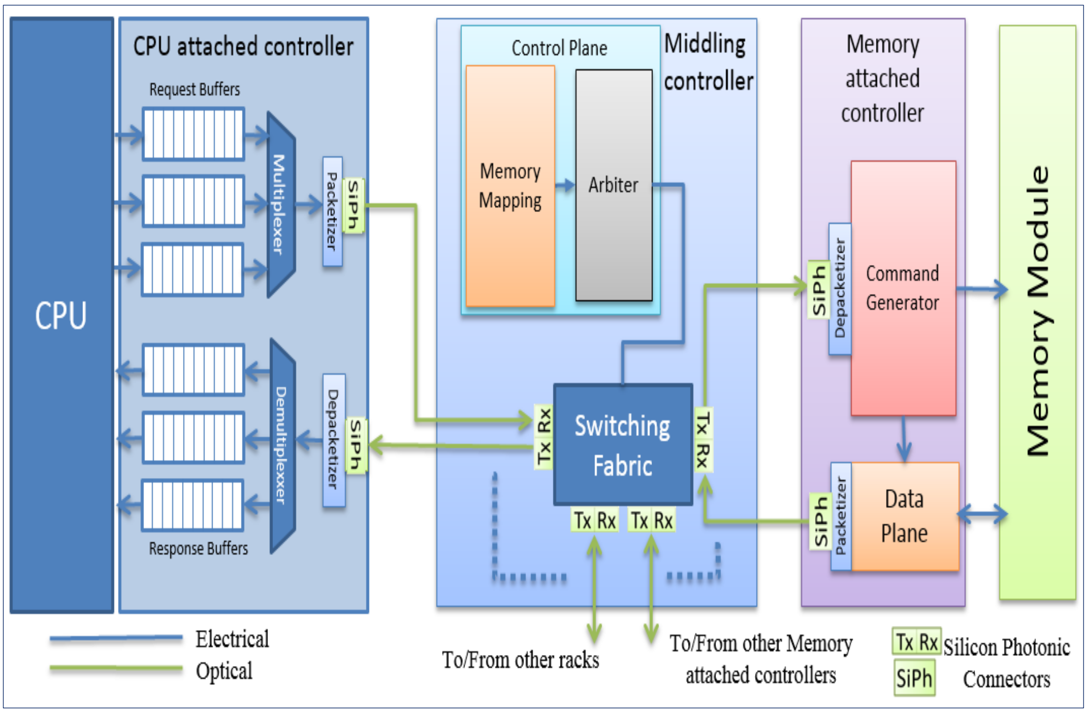
When disassembling the memory controller we construct the three functional blocks shown in Fig. 4-1.b. The first block of Fig. 4-1.b, starting from the left is the CPU directly attached to the CPUMC as the CPU needs to see the same old interface. Buffers from the memory controller are attached to the CPU directly and data are being selected from these buffers to be sent to their destination memory rack. In this block we have added a packetiser [65], after multiplexing the incoming memory access requests from the CPU. The packetiser’s role is to packetise the memory controller data to be switched between the CPU rack and the memory rack.

On the other hand, the depacketiser puts the responses from the memory in normal data form to be read by the CPU.

The block in the middle is the MMC where the memory mapping and the arbiter functional blocks are integrated with the top of memory rack switch. The memory mapping and the switch arbiter form the control plane of the switch. When receiving memory access requests, in packets forms, the control plane of the memory controller reads the header of the packets and according to the ID of the destination memory module, a path is established to the intended memory module. Regarding the memory management, we assume its functionality will be added to the functionality of the control plane of the MMC and any changes to the management and control system can be manage through the control plane of the MMC.



(a) Classical Memory Controller



(b) Disassembled Memory Controller

Fig. 4-1: Memory controller

Finally, the command generator is attached directly to the memory modules to form the MEMC as shown in Fig. 4-1.b. It generates commands to read from/write to the memory through the control path for control signalling, and through the data path for receive and send data.

4.2.1 Racks Interconnect Topology

We have designed a modular software defined architecture that can replace the traditional single rack of servers, with three racks, namely CPU rack, memory rack and IO rack. These racks are connected and communicate using the new communication fabric described. In this architecture our DS design is built up by disaggregating the server into its main components where the switching between the racks is accomplished in a distributed manner through the use of the previously mentioned components in Table 4-1.

Optical Connectors (SiPh)	Intel Silicon Photonic interconnects [16]

MEXC	Electronic switch that grooms different CPUs’ traffic to access RAM racks
IOXC	Electronic switch that grooms different CPUs’ traffic to access IO racks
IO Packet Engine	IO adapter [66]
IO CTRL	IO controller
OXC Switch	TOR optical switching units [52]
DMC Blocks	Disassembled memory controller blocks

Table 4-1: Main components in our DS design

Starting with the CPU rack, in this implementation, the new photonic interconnects and fibre cables are used to connect the CPUs throughout the rack via a point-to-point to a top of rack electronic memory switch (MEXC). These intra-rack connections are all optical, i.e. different wavelengths are used for the set of computing trays in each rack.

In this design the computing systems have been configured in trays, each tray contains a single CPU die and its associated cache memory and control. The control consists of CPUMC and PCIe interface connecting the CPU with the IO packet engine. Thus both PCI and Ethernet networking protocols can be implemented in the same rack system, all enabled by the functionality of the MEXC and IOEXC switches, using light as the transmission medium over fibre channels.

Two IO packet engines are used in this design, one for each side of the CPU-IO link. This serial interface is configured to transfer the data, address and control information, required to communicate with external IO modules such as hard disks and Ethernet ports using a serial packetised protocol. The CPU side IO packet engine provides an interface to the CPU and supports the IO switch IOEXC on top of the CPU rack, by packetising/depacketising the IO control/data signals, to be sent to their intended destination. The IO side IO packet engine provides an interface to peripheral devices such as IO cards to support the communication between the disaggregated resources. This relies on the design idea given in [49], where the IO modules are disaggregated from the rest of the server box.

Due to differences between the memory and IO packet formats two separate switches have to be implemented, one for the CPU-memory and the other for the CPU-IO communications. Another reason for separating the TOR Ethernet switches is that the CPU-Memory communication is latency intolerant. Here application specific switches have to be used, which are normally expensive but are high performance, in contrast to the CPU-IO traffic which is latency tolerant and commodity switches can be used to transfer such a traffic. Furthermore, the separation of the two forms of traffic reduces the load on the bottleneck MEXC and results in fast communication. These switches are very important for traffic grooming to collect traffic from different CPU cards to optimise the number of wavelengths used in the optical layer. These switches can be programmed to assign all traffic associated with a particular CPU to a specified port. The switch is programmable to allow software based implementation of the protocols used for communications at any particular port. The output from these switches are fed into an OXC switch which is the gateway for the rack to connect it with its neighbouring racks.

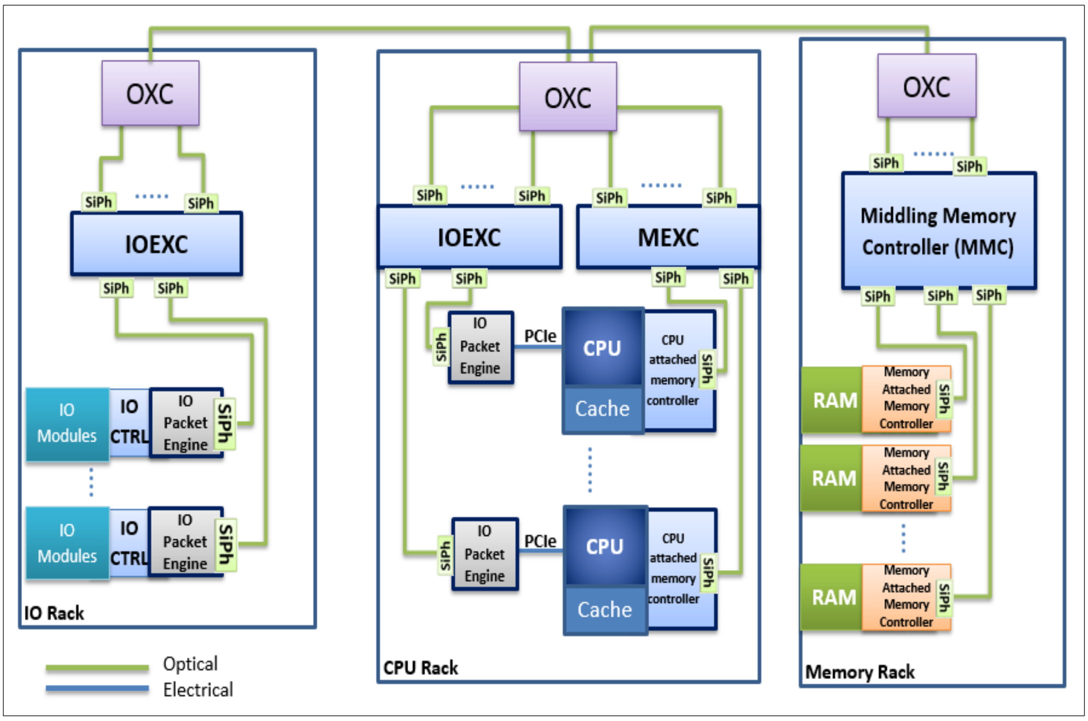


Fig. 4-2: DS architecture

The connecting inter-rack links, linking the OXCs, are all optical to achieve high bandwidth, low latency data transmission and simplicity of wiring by using fewer number of cables/fibres which is an essential issue for certain dense applications. The number of output ports of each WDM OXC switch depends on the number of neighbours of the rack where the switch resides, where these outputs are connected to its neighbouring OXCs.

In the memory rack, starting from the top, the OXC is connected to the middling memory controller via the fibres and silicon photonic interconnects. The middling memory controller, in turn, provides a path to the selected memory module. The middling memory controller combines both the switching and the MMC functionalities. After switching, the data are sent to the MEMC attached to the required memory module, optically.

Additionally, the design supports direct memory access (DMA) such that the memory rack can communicate with the IO rack directly without interrupting the busy CPU. This is because the memory and IO racks are interconnected through their optical switches, either by directly bypassing the electronic switches of the intermediate racks via a cut-through lightpath in the bypass scenario [67], or through the intermediate electronic switches, in a non-bypass scenario, i.e., all the data carried by the lightpaths is processed and forwarded by electronic switch [67].

The IO rack structure is relatively similar to the memory rack and it is disaggregated in a similar way to what is done in [49], with the use of the IOEXC to support the OXC. All the communication links here are optical to achieve fast and high bandwidth transmission. In this rack, the WDM OXC on the very top of the IO rack is connected to the electronic switch on top of the IO modules, IOEXC, and the IOEXC is connected optically to the different IO modules, which reside in the IO rack, through their packet engines and passing their IO controllers.

Communication integrity, control and management are provided by a global data centre operating system (GOS). This operating system is a general control layer that has an inclusive view for the whole disaggregated racks with their connectivity in order to be able to provide fluency in communication and manage the connectivity.

A hypervisor which is a software layer that runs on top of the hardware resources and provides virtual partitioning capabilities to higher-level virtualisation services can be coupled with the GOS. The Hypervisor enables the GOS to supervise and multiplex multiple operating systems to maintain and control the entire resources at all times and enable different operating systems to operate cooperatively.

CPU-to-CPU communication is managed by the top of memory rack electronic switch, MMC, and the MEXC, as communicating CPUs will interconnect through the remote memory modules, shared memory, they are using [68]. CPU-Memory rack communication is performed by mutual functionality between the OXCs, DMC blocks and the MEXC. The CPU-IO communication is facilitated by the functionality of IO packet engines on both racks to support the switching fabric implemented by the IOEXCs and the OXCs.

In brief, in the CPU rack, there are CPU trays whose traffic is aggregated using an electronic switch and is forwarded to the destined rack through optical layer switching using the OXC switch.

4.3 Disaggregated Server Implementation Technologies and Limitations

This section provides a comprehensive description of the design components highlighting the communication patterns and traffic flows between the disaggregated resources with some implementation suggestions.

In our new design, packets have special packet formats. The CPUMC encapsulates the memory address and control information like Read, Write, number of successive bytes etc, as an Ethernet packet for communication between the processor and memory modules that are located in different racks. For example, a packet sent from a CPU contains an address part (header) and data (payload). The address contains, the IP of the destination rack and the ID of the specific module memory or IO, which the CPU wants to access. These are provided by the data centre global operating system. The rack IP is used by the CPU rack MEXC or

IOEXC switch, to forward the packet to the destination rack. On receiving the packet at the destination rack, the top of rack, MMC or IOEXC, reads the specific module ID and forwards the packet to its right destination module.

Recall that our design is a packet switch based communication fabric where all the communication passes through the electronic (Ethernet) TOR switches. For high performance computing, such as the DS data centre, low latency switching is a key element to enable upper layer applications to get their job done as quickly as possible, thus, switch latency is becoming a very critical factor. Ethernet switching latency is defined as the time it takes for a switch to forward a packet from its ingress port to its egress port. Thus choosing the right switch for this job needs to be done with extensive care considering the main factors related to latency mentioned earlier. Many factors can affect the switch latency such as [50]:

1) Switching method: cut-through or store-and-forward: With Store-and-forward the switch stores the received data in memory at the ingress of port, upon receiving the entire packet frame the switch then transmits the data frame using the appropriate egress port(s). This switching method introduces a relatively high latency which is proportional to the size of the frame being transmitted and inversely proportional to the bit rate. In contrast, with cut-through the switch starts to send the packet as soon as the destination becomes known, normally using the first 6 bytes, eliminating the need for the whole packet to be read into the switch. Obviously cut-through switching can achieve much faster performance and provides lower latency.
2) Wireline Latency: fibre transmits data at about ⅔ of the speed of light [50]. Thus travelling long links makes this delay more significant. Note that for the distances

involved in our DS design which could traverse a maximum of few meters, this delay becomes trivial compared with other contributors to latency.

3) Traffic patterns and packet size: different traffic patterns from full-mesh to port pair can affect the load on the switch and eventually the switch latency. The difference in latency between the most complicated configuration, mesh, and the simplest configuration, port pair, can be huge (300-500% or even higher) [69]. On the other hand, the packet size can affect the latency, long packets take more time in switching and may delay other packets.
4) Traffic rate: describes the change from the physical line rate of the switch port to the potentially significantly lower throughput of the data flowing through the switch ports. Reducing the traffic flow through the switch reduces the switching latency.
5) Number of utilised switch ports: change from the fully loaded configuration where all ports are used, to only few working ports affects the latency. Having a lower number of working ports means having less load which in turn reduces the switching latency.

In addition, the internal switching fabric of the switch (the switch fabric consists of silicon that implements the store and forward engine, MAC address table, and VLAN, among other functions) will participate in the total latency introduced by the switch. Thus incorporating high performance and application specific switching components introduces a favourable impact on the total data centre performance [50].

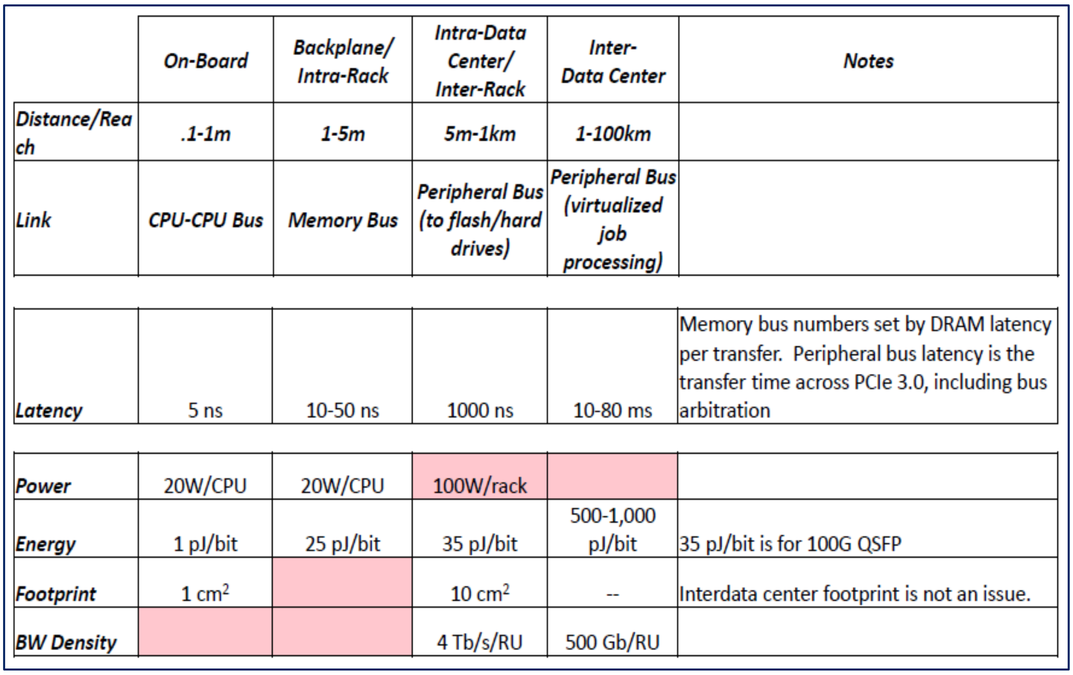
In relation to the implementation of the design, we describe here some proposals and visions that can further reduce the total latency in our design: (i) we suggest the use of reduced switching protocol overhead and simple packet format, due to the topology and data nature, which will jointly help in reducing the total system latency; (ii) our switches could be designed specifically for certain packets formats, like the CPU-MEM and CPU-IO and MEM-IO, instead of generic IP switches (iii) we propose the use of flexible protocol formats to handle different applications that have different latency restrictions. Thus for latency tolerant applications we allow the use of implicit circuit switching by establishing dedicated channels for a given time for this specific application; (iv) the use of MPLS as a simple switching technique or implicit circuit switching with time division multiplexing (TDM); (v) implementing optical switching (circuit, packet, and burst switching), as fast and reliable switching technique, which will also eliminate the need for some of the optical transceivers which perform optical to electrical to optical conversion when electronic switches are used. The elimination of the packetiser / depacketiser is also attractive; (vi) finally by looking at the latency reduction trends in the last several years in Ethernet switches, attributed to new advanced switching architecture design and improved silicon technology, the Ethernet switch latency is decreasing from double-digit milliseconds to sub-1 microsecond [69]. With this trend it is highly likely that the Ethernet switch latency will decrease in future to the point that fits the DS requirements.

In response to the metrics given in Fig. 4-3, we suggest here several low latency switches showing that switches with the required performance exist. These switches include for example:

The Cisco Nexus 3064 switch: Cisco has introduced an ultra-low latency of less than a microsecond, low-power, dense switch, the Cisco Nexus 3064 switch, part of the unified fabric family [70].

The Cisco SFS 7000P: the Cisco SFS 7000P is a new class of data centre switches that delivers scalable, high-performance, low-latency server switching with less than 200 nanoseconds (ns) of port-to-port latency for high-performance server clusters of all sizes [71].

Mellanox M4001Q 40Gb/s Infiniband Switch: the SwitchX® M4001Q QDR and M4001F FDR InfiniBand blade switches for Dell PowerEdge M-series chassis delivers superior performance for applications that demand the highest bandwidth and lowest latency, a port-to-port latency of 170ns [72].



	On-Board	*Backplane/ Intra-Rack*	*Intra-Data Center/ Inter-Rack*	*Inter-Data Center*	*Notes*
Distance/Reach	*.1-1m*	*1-5m*	*5m-1km*	*1-100km*	
Link	*CPU-CPU Bus*	*Memory Bus*	*Peripheral Bus (to flash/hard drives)*	*Peripheral Bus (virtualized job processing)*	
Latency	5 ns	10-50 ns	1000 ns	10-80 ms	Memory bus numbers set by DRAM latency per transfer. Peripheral bus latency is the transfer time across PCIe 3.0, including bus arbitration
Power	20W/CPU	20W/CPU	100W/rack		
Energy	1 pJ/bit	25 pJ/bit	35 pJ/bit	500-1,000 pJ/bit	35 pJ/bit is for 100G QSFP
Footprint	1 cm^2		10 cm^2	--	Interdata center footprint is not an issue.
BW Density			4 Tb/s/RU	500 Gb/RU	

Fig. 4-3: Key metrics for disaggregation [38]

In the following, we discuss the possibility of replacing the electronic switches in our design by optical switches, i.e. full optically switched disaggregated data centre. One of the suggestions is to use full optical switching in our design. Implementing full optical switching can reduce the problem of switching latency given that optical switches can have very low latency. Additionally, implementing optical switching can eliminate the need for electrical to optical transceivers at some points. This can reduce the total transmission latency and power consumption. However, moving to optical switching has its own drawbacks such as the potential high cost associated with the new technologies used. Below are some optical switching techniques with some real world implemented switches that can be useful in the implementation of our design.

Optical circuit switching: is a mature technology where data are transmitted in the form of optical signals and the transmission paths are point-to-point connections such that a dedicated path is established between communicating ends for the communication period. Optical circuit switching can be fast, however the communication patterns between the disaggregated resources are of high diversity throughout the day or over different days. Thus, this technique is useful for VMs that are maintained over long time periods, as for short time VMs, the path establishment frequency increases which might lead to performance degradation due to the associated latency and path set up time. Current switches such as the S320 optical circuit switch [73], by CALIENT, delivers sub-60 nanosecond (ns) packet-streaming latency performance, according to recent tests. Being inherently point to point technique, this approach might limit the number of memory modules assigned to each CPU.

Optical packet switching: has been researched over the last two decades, yet it has not been deployed in commercial networks. The main obstacle facing optical packet switching is the lack of optical buffers, however recirculating optical buffers [74], electronic buffers with O/E/O conversion [75] and slow light technologies [76] are promising techniques. There have been some implementations of optical packet switching with latencies of 15.3 ns [77], and 25 ns [78]. The EpiPhotonics' unique

PLZT waveguide technology [79] enables a new generation of efficient and ultra-fast photonics with potent advantages such as ultra-fast switching (< 5~10 ns) making it a potential base for implementing an optical packet switch [80].

Optical label switching: is a specific implementation of optical packet switching where each packet is given a label and the switching decision is made after the examination of the label assigned to each packet. The switching occurs at the data link layer rather than at the network layer, thus switching is much faster. An optical label switch for WDM optical packet switching with latency of 105 ns has been developed [81].

Optical burst switching: is implemented by aggregating the data packets into data bursts at the edge of the network to form the data payload. These bursts are transmitted optically after extracting the routing control signals from the data packet. The routing control signals are transmitted optically on a special control channel while the payload data is transmitted on a different channel(s). The communication channel is established for the duration of the burst and is subsequently released. The signalling stream is checked at each intermediate node while the data burst can cut through intermediate nodes. This form of switching can be useful for memory-IO communication which is typically bursty in nature (e.g. file downloading) unlike the CPU-MEM communications which is not usually bursty, i.e. is typically a continuous flow of read/write commands [82].

In conclusion, the best approach is to choose the one that best fits the application requirements and network restrictions, from the sets of implementation recommendations listed in the previous sections. Having large L1, L2 or even L3 caches reduces the impact of the main memory latency since the CPU can retrieve and execute new data easily from its caches.

For the memory modules, we propose the use of high performance components to overcome the latency and communication delay bottlenecks. DDR4 [83] is the latest version of RAM technology, offering a range of improvements over its predecessor, DDR3 [84], such as greater range of available clock speeds and timings, lower power consumption. Having fast memory can speed up the whole system operation given that latency is a crucial point in DS design. Concerning energy efficiency, DDR4 reduced energy consumption makes it a good candidate for our design implementation.

4.4 Summary

This chapter examined the traditional monolithic CS design and compared it to a new design paradigm, the DS data centre design, and investigated the advantages of the DS design over traditional CS design. The DS design arranges data centres resources in physical pools such as processing, memory and IO module pools; rather than packing each subset of such resources in a single server box. We presented our new design for the photonic DS based data centre architecture supplemented with a complete description of the architecture components and communication patterns. Our new DS architecture has been built using some new functional components in addition to some conventional components that can facilitate the communication and connectivity among the disaggregated resources. Finally, some recommendations for the design and implementation have been suggested focusing on the requirements, the capabilities of different switching and implementation technologies and the challenges that can face this architecture.

Chapter 5: Energy Efficient Resource Provisioning in DS Server with Communication Fabric

5.1 Introduction

In the literature a number of energy efficient inter data centre communication networks and architectures have been proposed and studied [85-87], however, data centre energy management is still a hot topic for both industry and academia. We believe that implementing the DS based data centres architecture can bring a variety of benefits considering different prospects including improved energy efficiency. In this chapter we focus on the energy efficiency gains of resource provisioning and VM allocation in a DS based data centre. Data centres are large computing facilities built for applications that have very diverse resource requirements and are supposed to last for 15 to 20 years. Some applications are network intensive, such as video streaming applications, and others are latency sensitive and/or CPU intensive, such as web search. The loads on a data centre vary throughout the day and are related to our daily life events. This in turn creates challenges in attempting to reduce power consumption while maintaining the data centre's performance. Precise resource provisioning and management directly influence the overall data centre energy efficiency, and are of extreme importance in data centre design. Under provisioning of data centre resources means that resources will be the bottleneck, while over provisioning data centre resources means a loss in power and capital. Thus, accurate provisioning is of high importance and motivates data centre efficient design. Our vision is that implementing DS to provide a solution for the problem of good resource provisioning can result in pleasing outcomes.

Most previous work in the area of resource provisioning in data centres focused on dealing with the VM itself, such as slicing [21], queuing, and migration [88], and multiple VM multiplexing [44]. In this chapter and due to the limitations of current server design, we study the DS approach which can improve the data centre resource provisioning and resource utilisation. Thus, the aim is an efficient data centre in terms of power consumption and performance.

In the following paragraphs, we describe the type of the data centre we considered and how we can account for the power consumption associated with a requested VM running in the data centre. We present details of the assumptions and system configuration for the resource provisioning and VM placement using disaggregated resources. Each VM request is identified by a unique id, denoted by index i, and each CPU, memory and IO module in the data centre is similarly identified by a unique id, denoted by index j. Throughout the rest of the work we use VM and VM requests interchangeably to refer to requested resources by a VM.

With the optimisation of the VM placement in DS based data centres, consideration has to be given also to the inter rack communication power consumption considering the new DS design structure. In the CS data centre, resource utilisation may not be as efficient as in the DS data centre, however the traffic which used to be contained within the same server or the same rack in CS data centre, now typically navigates through several racks spanning part of the data centre fabric [6].

5.2 Resource Provisioning MILP model with Communication Fabric

As explained in Fig. 2-3 and in Chapter 4, each processing resource rack is served by two electrical switches, one for CPU-Mem communication and the other for CPU-IO communication; and on the top of the rack there is an optical switch. The memory rack and IO rack are each served by a single electrical switch and a top of rack optical switch. All optical switches on top of the racks are connected in a semi mesh connection. Inside each rack the transceivers [89] shown in Fig. 4-2, SiPh, support each port in each electronic switch. Each link is supported by transceivers and packetisers (packet engines for communications with IO modules) [90] at each end, one next to the source resource and one next to the destination resource. In addition an optical Mux/Demux [91] is added after the transceivers at the link ends near the resources. As each transceiver operates at 100 Gb/s in our design and a single resource traffic could exceed this 100 Gb/s, more transceivers can be used by a single resource, imposing the need to add multiplexing units. For the added functionalities of the memory mapping and arbiter to the MMC, we consider an additional 5 W to each working MMC to account for the power consumption of these units.

VMs demand resources in both the IP layer and the optical layer, in addition to the underlying DS resources. For evaluation, we define the following sets, parameters and variables:

Sets:

NR Set of all racks

PR	Set of CPU racks
MR	Set of memory racks
IOR	Set of IO racks
N_a	Set of neighbour racks of rack a
VM	Set of VMs to be served
NP	Set of CPUs in each CPU rack
NM	Set of memories in each memory rack
NIO	Set of IOs in each IO rack

Parameters:

$a\ and$ b	Denote end points of a physical fibre link in the optical layer
NVM	Total number of VMs
PRO	The CPUs processing capabilities (GHz)
MEM	Memory capacity of each memory module (GB)
IO	Total transmission rate of each IO module (Gbps)
VP_i	Processing demand of VM i (GHz)
VPM_i	CPU-Memory traffic demand of VM i (GBps)
VM_i	Memory demand of VM i (GB)
$VMIO_i$	Memory-IO traffic demand of VM i (Gbps)
VIO_i	IO demand of VM i (Gbps)
$VPIO_i$	CPU-IO traffic demand of VM i (Gbps)
D_{ab}	Distance between rack pair (a, b) (m)

ΔP	Power Factor of the CPU (W), $\Delta P = Pmax - Pmin$
ΔM	Power Factor of the memory (W), $\Delta M = Mmax - Mmin$
ΔIO	Power Factor of the IO module (W), $\Delta IO = IOmax - IOmin$
$Pmax$	Power consumption of fully utilised CPU (W)
$Pmin$	The idle power consumption of the CPU (W)
$Mmax$	Power consumption of fully utilised memory (W)
$Mmin$	The idle power consumption of the memory (W)
$IOmax$	Power consumption of fully utilised IO module (W)
$IOmin$	The idle power consumption of an IO module (W)
PRS	Electrical switch port power for source nodes (W)
PRI	Electrical switch port power for intermediate nodes (W)
PO	Optical switch power (W)
B	Wavelength rate (Gbps)

Variables:

$\theta P_{i,j}^{p}$	Portion of the processing capacity of processor j in processors rack p allocated to request i
$\theta M_{i,j}^{m}$	Portion of the memory j in memory rack m allocated to request i
$\theta IO_{i,j}^{io}$	Portion of the transmission rate of port j in IO rack io allocated to request i
δP_{j}^{p}	Total utilisation of processor j in CPU rack p

δM_j^m Total utilisation of memory j in memory rack m

δIO_j^{io} Total utilisation of IO module j in IO rack io

$YP_{i,j}^p$ $YP_{i,j}^p = 1$ if processor j in processors rack p hosts request i, otherwise $YP_{i,j}^p = 0$

$YM_{i,j}^m$ $YM_{i,j}^m = 1$ if memory j in memory rack m hosts request i, otherwise $YM_{i,j}^m = 0$

$YIO_{i,j}^{io}$ $YIO_{i,j}^{io} = 1$ if port j in IO rack io hosts request i, otherwise $YIO_{i,j}^{io} = 0$

K_i K_i =1 if request i is served, K_i=0 if it is blocked

K_{ip} K_{ip} =1 if request i processor requirements are served, K_{ip}=0 if request i is blocked

K_{im} K_{im} =1 if request i memory requirements are served, K_{im}=0 if request i is blocked

K_{iio} K_{iio} =1 if request i IO requirements are served, K_{iio}=0 if request i is blocked

XP_j^p XP =1 indicates that processor j in processors rack p is active, otherwise, XP =0

XM_j^m XM_j =1 indicates that memory j in memory rack m is active, otherwise, XM_j =0

XIO_j^{io} XIO_j^{io} =1 indicates that port j in IO rack io is used, otherwise, XIO_j^{io}

	=0
TVM	Total number of served VMs
QPM_p	Number of aggregation ports of the CPU-MEM electrical switch at processor rack p
$QPIO_p$	Number of aggregation ports of the CPU-IO electrical switch at processor rack p
$QMIO_m$	Number of aggregation ports of the electrical switch at memory rack m
$WPM_{a,b}$	Number of wavelengths that carry the CPU-MEM traffic in physical link (a, b)
$WPIO_{a,b}$	Number of wavelengths that carry the CPU-IO traffic in physical link (a, b)
$WMIO_{a,b}$	Number of wavelengths that carry the MEM-IO traffic in physical link (a, b)
$WPM_{a,b}^{p,m}$	The number of wavelengths of lightpath (p, m) passing through a physical link (a, b);
$WPIO_{a,b}^{p,io}$	The number of wavelengths of lightpath (p, io) passing through a physical link (a, b);
$WMIO_{a,b}^{m,io}$	The number of wavelengths of lightpath (m, io) passing through a physical link (a, b);
$PM_i^{p,m}$	Indicator to connect the *ith* VM CPU-MEM traffic to the relevant

CPU and MEM racks

$ZPM_i^{p,m}$ Binary variable, index to the source-destination of the *ith* VM CPU-MEM traffic

$PIO_i^{p,io}$ Indicator to connect the *ith* VM CPU-IO traffic to the relevant CPU and IO racks

$ZPIO_i^{p,io}$ Binary variable, index to the source-destination of the *ith* VM CPU-IO traffic

$MIO_i^{m,io}$ Indicator to connect the *ith* VM MEM-IO traffic to the relevant MEM and IO racks

$MIO_i^{m,io}$ Binary variable, index to the source-destination of the *ith* VM MEM-IO traffic

$TPM_{p,m}$ Total CPU-MEM traffic (Gbps)

$TPIO_{p,io}$ Total CPU-IO traffic (Gbps)

$TMIO_{m,io}$ Total MEM-IO traffic (Gbps)

The power consumption of a data centre based on the DS architecture is composed of two parts, the first part is the power consumed by active resources:

1) The power consumption of active processors

$$\sum_{p \in PR} \sum_{j \in NP} ((XP_j^p \cdot Pmin) + (PF \cdot \delta P_j^p))$$

2) The power consumption of active memories

$$\sum_{m \in MR} \sum_{j \in NM} ((XM_j^m \cdot Mmin) + (MF \cdot \delta M_j^m))$$

3) The power consumption of active IO modules

$$\sum_{io \in IOR} \sum_{j \in NIO} ((XIO_j^{io} \cdot IOmin) + \left(IOF \cdot \delta IO_j^{io}\right)$$

The second part is the power consumed by networking elements:

1) Power consumption due to CPU-Memory traffic, which in turn is composed of:

a) The power consumed by the electrical switches

$$\sum_{p \, \in \, PR} PRS \cdot QPM_p + \sum_{a \in NR} \sum_{b \in N_a} PRI \cdot WPM_{a,b}$$

b) The power consumed by the optical switch

$$\sum_{a \in NR} PO$$

2) Power consumption due to CPU-IO traffic, which is composed of:

a) The power consumed by the electrical switch

$$\sum_{p \in PR} PRS \cdot QPIO_p + \sum_{a \in NR} \sum_{b \in N_a} PRI \cdot WPIO_{a,b}$$

b) The power consumed by the optical switch

$$\sum_{a \in NR} PO$$

3) Power consumption due to Memory-IO traffic, which consists of:

a) The power consumed by the electrical switch

$$\sum_{m \in MR} PRS \cdot QMIO_m + \sum_{a \in NR} \sum_{b \in N_a} PRI \cdot WMIO_{a,b}$$

b) The power consumed by the optical switch

$$\sum_{a \in NR} PO$$

The model is defined as follows:

Objective: minimise:

$$\sum_{p\in PR}\sum_{j\in NP}((XP_j^p \cdot Pmin) + (PF \cdot \delta P_j^p)) +$$

$$\sum_{m\in MR}\sum_{j\in NM}((XM_j^m \cdot Mmin) + (MF$$

$$\cdot \delta M_j^m)) +$$

$$\sum_{io\in IOR}\sum_{j\in NIO}((XIO_j^{io} \cdot IOmin)$$

$$+ \left(IOF \cdot \delta O_j^{io}\right) +$$

$$\sum_{p\in PR} PRS \cdot QPM_p +$$

$$\sum_{a\in NR}\sum_{b\in Na} PRI \cdot WPM_{a,b} \ +$$

$$\sum_{p\in PR} PRS \cdot QPIO_p +$$

$$\sum_{a\in NR}\sum_{b\in N_a} PRI \cdot WPIO_{a,b} \ +$$

$$\sum_{m\in MR} PRS \cdot QMIO_m +$$

$$\sum_{a\in NR}\sum_{b\in N_a} PRI \cdot WMIO_{a,b} \ +$$

$$\sum_{a\in NR} PO \qquad (5\text{-}1)$$

Equation (5-1) gives the model objective which is to minimise the resource provisioning power consumption and the communication fabric power consumption.

For simplicity and due to their small power consumption, we assume that the optical switches are always on. Note that *NR* unite all of *PR*, *MR*, and *IOR*, thus *PO* is being summed once over *NR*. *PRS* and *PRI* will explained later in detail to show the difference between their values.

Subject to :

1) Resource Allocation Constraints

   Capacity Constraints.

$$\delta P_j^p = \sum_{i \in VM} \theta P_{i,j}^p \leq Utl \tag{5-2}$$

$$\forall\, j \in NP,\ p \in PR$$

$$\sum_{p \in PR} \sum_{j \in NP} \theta P_{i,j}^p = V_i^p / PRO \tag{5-3}$$

$$\forall\, i \in VM$$

$$\theta P_{i,j}^p \leq W \cdot YP_{i,j}^p \tag{5-4}$$

$$\forall\, i \in VM, j \in NP, p \in PR$$

$$\theta P_{i,j}^p \geq e + YP_{i,j}^p - 1 \tag{5-5}$$

$$\forall\, i \in VM, j \in NP, p \in PR$$

$$\delta M_j^m = \sum_{i \in VM} \theta M_{i,j}^m \leq Utl \tag{5-6}$$

$$\forall\, j \in NM, m \in MR$$

$$\sum_{m \in MR} \sum_{j \in NM} \theta M_{i,j}^m = V_i^m / MEM \tag{5-7}$$

$$\forall\, i \in VM$$

$$\theta M_{i,j}^{m} \leq W \cdot YM_{i,j}^{m} \tag{5-8}$$

$$\forall\, i\ \in VM, j\ \in\ NM, m \in MR$$

$$\theta M_{i,j}^{m} \geq e + YM_{i,j}^{m} - 1 \tag{5-9}$$

$$\forall\, i\ \in VM, j\ \in\ NM, m \in MR$$

$$\delta IO_{j}^{io} = \sum_{i \in VM} \theta IO_{i,j}^{io} \leq Utl \tag{5-10}$$

$$\forall\, j\ \in\ NIO, io \in IOR$$

$$\sum_{io \in IOR} \sum_{j \in NIO} \theta IO_{i,j}^{io} = V_{i}^{io}/IO \tag{5-11}$$

$$\forall\, i\ \in VM$$

$$\theta IO_{i,j}^{io} \leq W \cdot YIO_{i,j}^{io} \tag{5-12}$$

$$\forall\, i\ \in VM, j\ \in\ NIO, io \in IOR$$

$$\theta IO_{i,j}^{io} \geq e + YIO_{i,j}^{io} - 1 \tag{5-13}$$

$$\forall\, i\ \in VM, j\ \in\ NIO, io \in IOR$$

Constraint (5-2) calculates the total processing utilisation of each processor and ensures that it is less than the maximum allowed utilisation. Constraint (5-3) calculates the utilisation of each processor per allocated VM, and constraints (5-4) and (5-5) allocate each VM to a certain processor in a certain CPU rack.

Constraints (5-6)-(5-9) and (5-10)-(5-13) repeat the same steps of constraints (5-2)-(5-5) but for the memory and IO modules, respectively.

2) Service Level Constraints:

$$\sum_{i \in VM} K_i \geq NVM \cdot SLA \tag{5-14}$$

$$K_{ip} \leq \sum_{p \in PR} \sum_{j \in NP} YP_{i,j}^{p} \tag{5-15}$$

$$\forall\, i \in VM$$

$$W \cdot K_{ip} \geq \sum_{p \in PR} \sum_{j \in NP} YP_{i,j}^{p} \tag{5-16}$$

$$\forall\, i \in VM$$

$$K_{im} \leq \sum_{m \in MR} \sum_{j \in NM} YM_{i,j}^{m} \tag{5-17}$$

$$\forall\, i \in VM$$

$$W \cdot K_{im} \geq \sum_{m \in MR} \sum_{j \in NM} YM_{i,j}^{m} \tag{5-18}$$

$$\forall\, i \in VM$$

$$K_{iio} \leq \sum_{io \in IOR} \sum_{j \in NIO} YIO_{i,j}^{io} \tag{5-19}$$

$$\forall\, i \in VM$$

$$W \cdot K_{iio} \geq \sum_{io \in IOR} \sum_{j \in NIO} YIO_{i,j}^{io} \tag{5-20}$$

$$\forall\, i \in VM$$

$$K_i = K_{ip} = K_{im} = K_{iio} \tag{5-21}$$

$$\forall\, i \in VM$$

Constraint (5-14) ensures that the total number of served VMs is within an acceptable predefined percentage of the incoming VMs requests. The K_i value depends on the outcomes from constraints (5-15)-(5-21) collectively. If all these constraints yield a value of 1, then K_i is 1, otherwise K_i is 0.

3) Slicing Constraints:

$$\sum_{p\in PR}\sum_{j\in NP} YP_{i,j}^{p} \leq 1 \qquad \forall\, i\ \in VM \qquad (5\text{-}22)$$

$$\sum_{m\in MR}\sum_{j\in NM} YM_{i,j}^{m} \leq 1 \qquad \forall\, i\ \in VM \qquad (5\text{-}23)$$

$$\sum_{io\in IOR}\sum_{j\in NIO} YIO_{i,j}^{io} \leq 1 \qquad \forall\, i\ \in VM \qquad (5\text{-}24)$$

Constraint (**5-22**) ensures that a VM i processing requirement is served by only one CPU, i.e. this constraint prevents VM slicing. Constraints (**5-23**) and (**5-24**) repeat constraint (**5-22**) for the memory and IO requirements. If multiple VM copies or VM slicing is required, equations (**5-22**)-(**5-24**) should be upper bound by an appropriate number greater than 1.

4) Active resources constraints:

Active processors

$$XP_j^p \leq W \cdot \delta P_j^p \qquad (5\text{-}25)$$

$$\forall\, p \in PR, j\ \in NP$$

$$W \cdot XP_j^p \geq \delta P_j^p \qquad (5\text{-}26)$$

$$\forall\, p \in PR, j\ \in NP$$

Active memory modules

$$XM_j^m \leq W \cdot \delta M_j^m \qquad (5\text{-}27)$$

$$\forall\, m \in MR, j \in NM$$

$$W \cdot XM_j^m \geq \delta M_j^m \tag{5-28}$$

$$\forall\, m \in MR, j \in NM$$

Active IO ports

$$XIO_j^{io} \leq W \cdot\ \delta IO_j^{io} \tag{5-29}$$

$$\forall\, io \in IOR, j \in NIO$$

$$W \cdot XIO_j^{io} \geq \delta IO_j^{io} \tag{5-30}$$

$$\forall\, io \in IOR, j \in NIO$$

Constraints (5-25) and (5-26) jointly find the active processors by checking the utilisation δP_j^p. Constraints (5-27) and (5-28) together check the active memory modules and constraints (5-29) and (5-30) repeat same steps but for the IO modules.

5) Communication constraints:

Generating the index matrix for the CPU-Memory traffic

$$PM_i^{p,m} \cdot 2 = \sum_{j \in NP} YP_{i,j}^p + \sum_{j \in NM} YM_{i,j}^m \tag{5-31}$$

$$\forall\, i \in VM, p \in PR, m \in MR$$

$$ZPM_i^{p,m}\ \leq\ PM_i^{p,m} \tag{5-32}$$

$$\forall\, i \in VM, p \in PR, m \in R$$

$$ZPM_i^{p,m} \geq\ PM_i^{p,m} - 0.5 \tag{5-33}$$

$$\forall\, i \in VM, p \in PR, m \in MR$$

Generating the index matrix for the CPU-IO traffic

$$PIO_i^{p,io} \cdot 2 = \sum_{j \in NP} YP_{i,j}^p + \sum_{j \in Nio} YIO_{i,j}^{io} \tag{5-34}$$

$$\forall\, i \in VM, p \in PR, io \in IOR$$

$$ZPIO_i^{p,io}\ \leq\ PIO_i^{p,io} \tag{5-35}$$

$$\forall\, i \in VM, p \in PR, io \in IOR$$

$$ZPIO_i^{p,io} \geq \, PIO_i^{p,io} - 0.5 \tag{5-36}$$

$$\forall\, i \in VM, p \in \, PR, io \in IOR$$

Generating the index matrix for the Memory-IO traffic

$$MIO_i^{m,io} \cdot 2 = \sum_{j \in NP} YM_{i,j}^{m} + \sum_{j \in NIO} YIO_{i,j}^{io} \tag{5-37}$$

$$\forall\, i \in VM, m \in MR, io \in IOR$$

$$ZMIO_i^{m,io} \, \leq \, MIO_i^{m,io} \tag{5-38}$$

$$\forall\, i \in VM, m \in MR, io \in IOR$$

$$ZMIO_i^{m,io} \geq \, MIO_i^{m,io} - 0.5 \tag{5-39}$$

$$\forall\, i \in VM, m \in MR, io \in IOR$$

Constraint (5-31) connects each source CPU rack to its destination memory rack. Constraints (5-32) and (5-33) collectively generate source-destination index matrix for all CPU-Memory traffic depending on constraint (5-31). Constraints (5-34)-(5-36) and constraints (5-37)-(5-39) repeat the same steps of (5-31)-(5-33) but for the CPU-IO traffic and Memory–IO traffic, respectively.

Generating the traffic demand matrix:

$$TPM_{p,m} = \sum_{i \in VM} VPM_i \cdot ZPM_i^{p,m} \tag{5-40}$$

$$\forall\, p \in PR, m \in MR$$

$$TPIO_{p,io} = \sum_{i \in VM} VPIO_i \cdot ZPIO_i^{p,io} \tag{5-41}$$

$$\forall\, p \in PR, io \in IOR$$

$$TMIO_{m,io} = \sum_{i \in VM} VMIO_i \cdot ZMIO_i^{m,io} \qquad (5\text{-}42)$$

$$\forall\, m \in MR, io \in IOR$$

Constraint (5-40) generates the CPU-Memory traffic matrix based on the index matrix ZPM calculated previously in constraints (5-32) and (5-33), and constraints (5-41) and (5-42) generate the CPU-IO and Memory-IO traffic matrices, respectively.

Traffic flow conservation:

$$\sum_{b \in N_a} WPM_{a,b}^{p,m} - \sum_{b \in N_a} WPM_{b,a}^{p,m} = \begin{cases} (TPM_{p,m}\ /B) & a = p \\ -(TPM_{p,m}\ /B) & a = m \\ 0 \quad otherwise & \end{cases} \qquad (5\text{-}43)$$

$$\forall p\ \in PR, m \in MR, a \in NR$$

$$\sum_{b \in N_a} WPIO_{a,b}^{p,io} - \sum_{b \in N_a} WPIO_{b,a}^{p,io} = \begin{cases} (TPIO_{p,io}\ /B) & a = p \\ -(TPIO_{p,io}\ /B) & a = io \\ 0 \quad otherwise & \end{cases} \qquad (5\text{-}44)$$

$$\forall p\ \in PR, io \in IOR, a \in NR$$

$$\sum_{b \in N_a} WMIO_{a,b}^{m,io} - \sum_{d \in N_a} WMIO_{b,a}^{m,io} = \begin{cases} (TMIO_{m,io}\ /B) & a = m \\ -(TMIO_{m,io}\ /B) & a = io \\ 0 \quad otherwise & \end{cases} \qquad (5\text{-}45)$$

$$\forall m\ \in MR, io \in IOR, a \in NR$$

Constraints (5-43)-(5-45) are the flow conservation constraints for the CPU-Memory, CPU-IO and Memory-IO traffic, respectively, in the networking elements switches. They ensure that the total incoming traffic is equal to the total outgoing traffic for all racks except for the source and destination racks.

Wavelengths capacity constraints:

$$\sum_{p\in PR}\sum_{m\in MR} WPM_{a,b}^{p,m} \leq WPM_{a,b} \tag{5-46}$$

$$\forall\, a \in NP, b \in N_a$$

$$\sum_{p\in PR}\sum_{io\in IOR} WPIO_{a,b}^{p,io} \leq WPIO_{a,b} \tag{5-47}$$

$$\forall\, a \in NP, b \in N_a$$

$$\sum_{m\in MR}\sum_{io\in IOR} WMIO_{a,b}^{m,io} \leq WMIO_{a,b} \tag{5-48}$$

$$\forall\, a \in NP, b \in N_b$$

Constraints (5-46)-(5-48) ensure that the summation of the wavelengths traversing a physical link in the optical layer do not exceed the total number of wavelengths in that link for the CPU-Memory, CPU-IO and Memory-IO traffics, respectively.

Number of aggregations ports

$$QPM_p = \frac{1}{B} \cdot \sum_{m\in MR} TPM_{p,m} \tag{5-49}$$

$$\forall\, p \in PR$$

$$QPIO_p = \frac{1}{B} \cdot \sum_{io\in IOR} TPIO_{p,io} \tag{5-50}$$

$$\forall\, p \in PR$$

$$QMIO_m = \frac{1}{B} \cdot \sum_{io\in IOR} TMIO_{m,io} \tag{5-51}$$

$$\forall\, m \in MR$$

Constraints (5-49)-(5-51) find the total number of aggregation ports utilised by the CPU-Memory, CPU-IO and Memory-IO traffics, respectively, in each rack.

5.3 Resource Provisioning in CS with Communication Power MILP Model

In the CS MILP evaluation we consider a pool of servers, rather than a pool of resources, the same resources used in the DS MILP are combined to form servers such that a CPU has its associated memory and IO resources in a closed server box and one resource utilization will affect the other two resources. For example CPU#1 is associated with memory#1 and IO port#1 and if CPU#1 is fully utilized by VM#1 then memory#1 and IO port#1 cannot be used by another VM even if they have enough capacity for the second VM. This is to be compared to the DS design. Thus, the number of servers is the same as the number of one type of the resources, such as total number of CPU resource. To account for the server power, we consider a fully loaded server power of 300 W [92]. The power consumption of each resource (eg. CPU, memory, IO card) in this server was comparable to the values we used in the DS. The power consumption of each resource was subsequently set exactly equal to the values used in DS to facilitate comparison. Out of the 300W, the idle power was then calculated as 150W which is typical for this type of server and is in agreement with experimental measurements in our lab.

For the CS approach, each VM is allocated to the server that has enough CPU, memory and IO modules to accommodate the VM, otherwise a new server is powered on to host the requesting VM.

In addition to the parameters and variable defined in Section 5.2, we define the following:

Sets:

NS Set of all servers

VM Set of VMs to be served

Variables:

δP_j Total processor utilisation of server *j*

δM_j Total memory utilisation of server *j*

δIO_j Total IO utilisation of server *j*

X_j Indicates if server j is active, $X_j = 1$ otherwise $X_j = 0$

θP_{ij} Portion of the processing capacity of server j allocated to request i

θM_{ij} Portion of the memory capacity of server j allocated to request i

θIO_{ij} Portion of the IO capacity of server j allocated to request i

Y_{ij} $Y_{ij} = 1$ if server *j* hosts request i, otherwise $Y_{ij} = 0$

NOS Number of working servers

The resource provisioning in CS based data centre MILP model is:

Objective: minimise:

$$\sum_{j \in NS} ((X_j \cdot Pmin) + (\Delta P \cdot \delta P_j)) +$$

$$\sum_{j \in NS} ((X_j \cdot Mmin) + (\Delta M \cdot \delta M_j)) +$$

$$\sum_{j \in NS} ((X_j \cdot IOmin) + (\Delta \mathrm{IO} \cdot \delta IO_j))$$

$$+NOS \cdot 150 \tag{5-52}$$

The objective (5-52) aims to minimise the total power (by consolidating VMs in the minimum number of working servers).

Capacity Constraints:

$$\delta P_j = \sum_{i \in VM} \theta P_{ij} \leq Utl \qquad \forall\, j \in NS \qquad (5\text{-}53)$$

$$P_j \cdot \theta P_{ij} = VP_i \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}54)$$

$$\theta P_{ij} \leq W \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}55)$$

$$\theta P_{ij} \geq e + Y_{ij} - 1 \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}56)$$

$$\delta M_j = \sum_{i \in VM} \theta M_{ij} \leq Utl \qquad \forall\, j \in NS \qquad (5\text{-}57)$$

$$M_j \cdot \theta M_{ij} = VM_i \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}58)$$

$$\theta M_{ij} \leq W \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}59)$$

$$\theta M_{ij} \geq e + Y_{ij} - 1 \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}60)$$

$$\delta IO_j = \sum_{i \in VM} \theta IO_{ij} \leq Utl \qquad \forall\, j \in NS \qquad (5\text{-}61)$$

$$IO_j \cdot \theta IO_{ij} = VIO_i \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}62)$$

$$\theta IO_{ij} \leq W \cdot Y_{ij} \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}63)$$

$$\theta IO_{ij} \geq e + Y_{ij} - 1 \qquad \forall\, i \in VM, j \in NS \qquad (5\text{-}64)$$

Constraint (5-53) calculates the total processing utilisation of each processor in each server and ensures that it is less than the maximum allowed utilisation. Constraint (5-54) calculates the utilisation of each processor per allocated VM, and constraints (5-55) and (5-56) allocate each VM to a certain processor in a certain server. Constraints (5-57)-(5-60) and (5-61)-(5-64) repeat the same steps of constraints (5-53)-(5-56) but for the memory and IO modules, respectively.

Slicing Constraint:

$$\sum_{j \in NS} Y_{ij} = 1 \qquad \forall\, i \in VM \qquad (5\text{-}65)$$

Constraint (5-65) ensures that each VM will be served by one server. This constraint will force service quality equal to 100% SLA.

Active Resources Constraint:

$$X_j \leq W \cdot \delta P_j \qquad \forall\, j \in NS \qquad (5\text{-}66)$$

$$W \cdot X_j \geq \delta P_j \qquad \forall\, j \in NS \qquad (5\text{-}67)$$

$$NOS = \sum_{j \in NS} X_j \qquad (5\text{-}68)$$

Constraints (5-66) and (5-67) find the working servers and constraint (5-68) uses their results to calculate the total number of working servers.

5.4 Energy Efficient Resource Provisioning in Disaggregated Servers with Communication Fabric (EERP-DSCF) Heuristic

This section presents our EERP-DSCF heuristic that mimics the MILP model behaviour and expands the scope of the MILP model by providing lower complexity algorithms that enable real time operation of the DS data centre and enable the evaluation of relatively large size DS data centre clusters.

The flow chart of the heuristic is shown in Fig. 5-1. In this study we use homogenous resources, therefore, sorting resources according to their PF is not necessary. Thus, the heuristic picks the first CPU from the first CPU rack in the cluster and uses it for serving the first VM request. Then the heuristic decides the VM's memory and IO racks allocation based on a joint criteria involving the resource availability and rack distance from the chosen CPU rack. Thus both packing and "open shortest path first (OSPF)" algorithms are applied together.

As shown in the flow chart, the heuristic picks the first VM and allocates the first CPU in the first CPU rack in the cluster. The heuristic then organises the memory and IO racks in a list according to their distances from the chosen CPU rack in an ascending order. Subsequently, the heuristic checks the resources availability in the newly organised lists. If the first memory rack has enough capacity to accommodate the VM under consideration, the heuristic uses it, otherwise, the next rack is tested. In the same fashion the heuristic checks the first IO rack in the list and allocates the chosen resources for the VM under consideration, otherwise, the next nearest IO rack is tested, and so on, till an available IO module is found. The heuristic tries to fill partially used resources and racks as much as possible before moving to next racks. After allocating resources to the first VM request, the heuristic loops for the rest of the VM requests until all VMs allocations are done. Finally, EERP-DSCF grooms the traffic from each rack according to their destinations and routes them among racks and calculates the total consumed power.

5.5 Evaluation Scenarios for the MILP and Heuristic

To evaluate the performance of the proposed model and heuristic, we consider the example data centre shown in Fig. 5-2. It consists of 72 racks (24 racks of each resource type) organised in an 8×9 matrix and 127 links as shown in Fig. 5-2 which represents the top view of the DS data centre considered.

For the model, a smaller version of the data centre cluster shown in Fig. 5-3 is used due to MILP computational complexity. It comprises 9 racks (3 racks of each resource type) organised in a 3×3 matrix and 12 links, as shown in Fig. 5-3. It follows the same structure and racks sequencing of the data centre cluster in Fig. 5-2. The first column consists of three IO racks, the second consists of three CPU racks and the last column consists of three memory racks. We consider a scenario in which each rack contains 8 resources of its own type, each CPU rack contains 8 processors, each IO rack contains 8 IO modules and each memory rack contains 8 memory modules.

For the heuristic, and due to its lower computational complexity, we evaluated the full 8×9 data centre shown in Fig. 5-2. Each column is of one type of resource racks and each rack of each type contains 42 resources of its type. Starting from the far left, the first column contains the IO racks, followed by the CPU racks, then the Memory racks, and this sequence is repeated for the next 6 columns. Note that each rack is only connected to the nearest neighbour racks using optical fibres. As for the heuristic, the distance between adjacent racks is set to 1m [93].

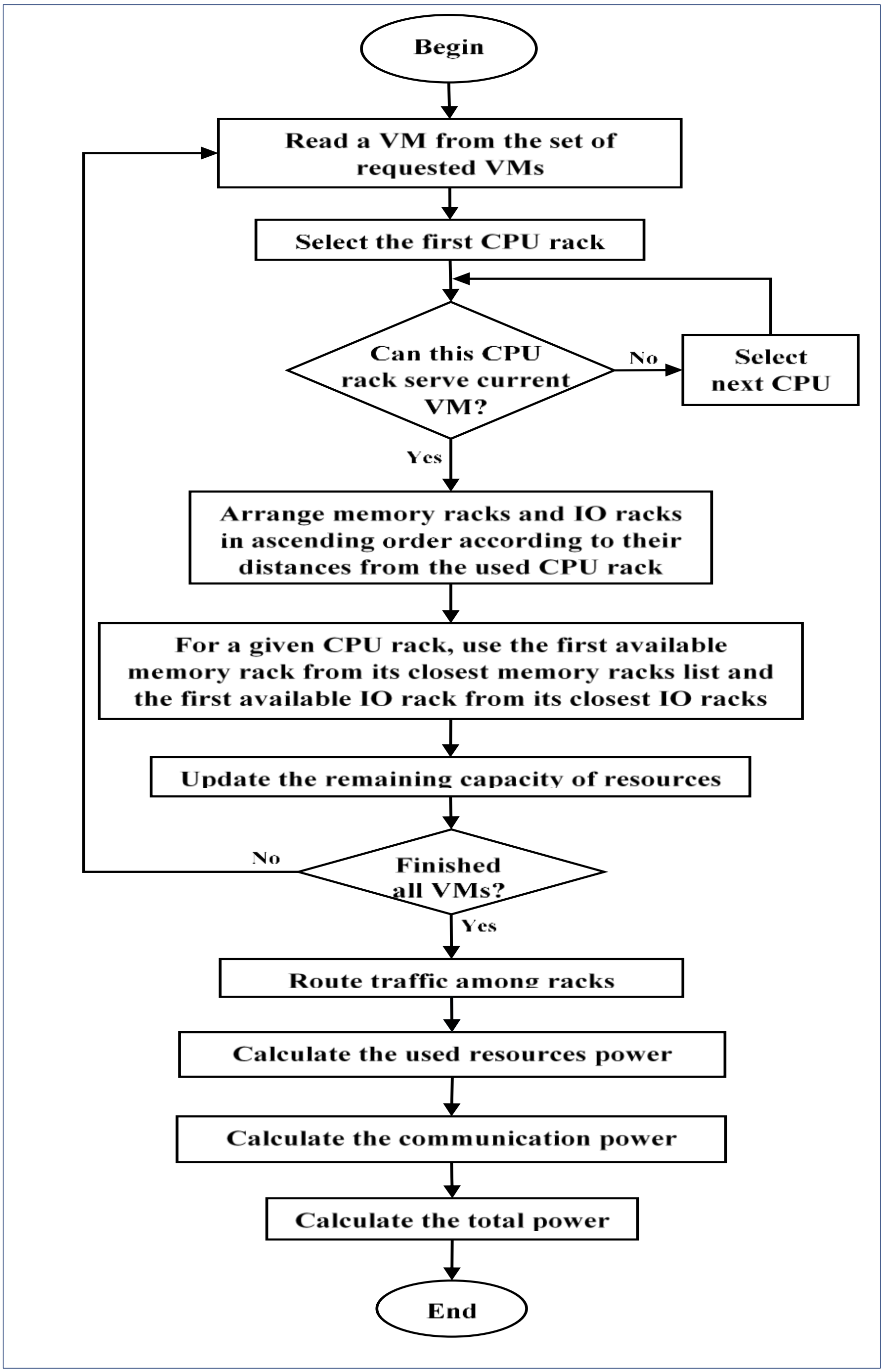

Begin
Read a VM from the set of requested VMs
Select the first CPU rack
Can this CPU rack serve current VM?
No
Select next CPU
Yes
Arrange memory racks and IO racks in ascending order according to their distances from the used CPU rack
For a given CPU rack, use the first available memory rack from its closest memory racks list and the first available IO rack from its closest IO racks
Update the remaining capacity of resources
Finished all VMs?
No
Yes
Route traffic among racks
Calculate the used resources power
Calculate the communication power
Calculate the total power
End

Fig. 5-1: EERP-DSCF heuristic flow chart

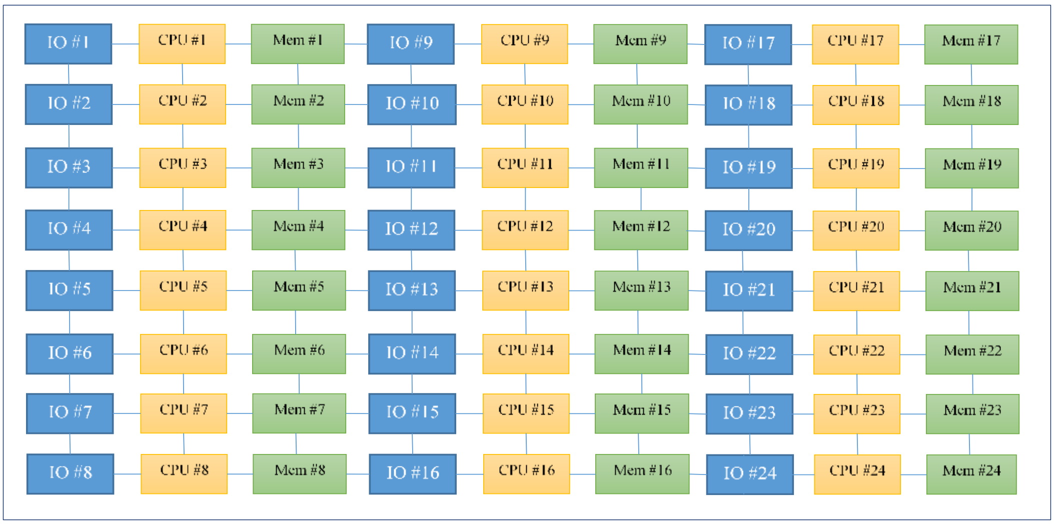


Fig. 5-2: DS based data centre structure under consideration

For both the model and the heuristic, each rack has its own intra rack communication fabric and electrical and optical switches to facilitate the communication among racks and inter rack communication, as shown in Fig. 2-3 and exemplified in Fig. 4-2.

Table 5-1 shows the parameters used for both model and heuristic. The power consumption of the resources we have used is consistent with our previous work in [94, 95], and for the network devices we use the values in Table 5-1 below.

Power consumption of electrical switch port for source nodes *PRS*	70.5 W
Power consumption of electrical switch port for intermediate nodes *PRI*	43.5 W
Power consumption of electrical 100 Gbps switch port (Pr)	40 W [96]
Power consumption of an optical switch	85 W [97]
Bit rate of each wavelength	100 Gbps
CPU capacity	3.6 GHz [61]

CPU maximum power consumption	130 W [61]
RAM capacity	8 GB [61]
Memory maximum power draw	10.24 W [98]
IO module rate	10 Gbps [61]
IO module maximum power draw	21.4 W [39]
100 Gbps Optical transceiver power consumption	3.5 W [89]
100 GHz Multiplexer power (W)	4 W [91]
100 Gbps Packet engine (packetiser or packet engine) power	20 W [90]

Table 5-1: Input Parameters for the model and simulation heuristic

Regarding some of the power values given in Table **5-1** it is worth noting that for the electrical switch port power and the packetiser power, we have scaled up the values given in [96] for the switch port and [90] for the channel adapter, linearly to account for 100 Gbps port power. For the switch port power and according to [96] a 10 Gbps port consumes 4 W, therefore, a switch port power is calculated as 4×10W (equivalent to 100 Gbps port). For the 100 Gbps packetiser, in [90] the 40 Gbps packetiser power consumption is 7 W, thus to scale up for 100 Gbps the total power is calculated as 7×2.5=17.5 W and an extra 2.5 W was added, i.e. 20 W in total in order to account for other possible functionalities such as buffering and arbitration, to be more conservative.

We evaluate PRS and PRI by considering the scenario explained in Section 5.2 where a transceiver power is added to each end of each link, and a packetiser power is considered for each source-destination pair.

$$PRS = 40\ W(electrical\ switch\ port) + 3\times 3.5\ W\ (transcivers\ power) + 20\ W\ (packetizer\ power)$$

Here we consider 3 transceivers, the first is for the source resource, the second is for the destination resource, and the third is for the source electrical switch port.

For each intermediate node we consider a switch port power plus a transceiver power, which yields:

$$PRI = 40\ W(eletrical\ switch\ port) + 3.5\ W\ (transciver\ power)$$

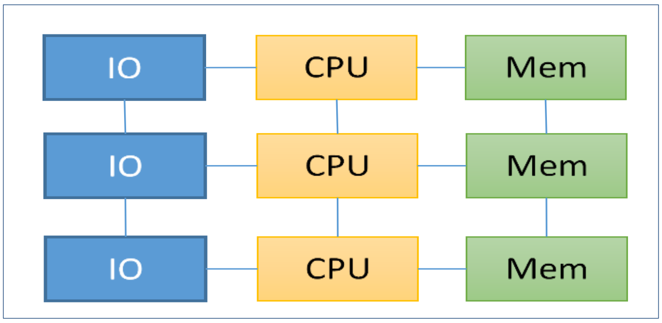


Fig. 5-3: Substrate data centre for the MILP model

In this section we use different values for the VM resources requirements and traffic demands which are assumed with respect to the three different VM types, PI, MI, and IOI. Table 5-2 lists the input parameters for the VMs requirements. Comparing the IO resources requirements to the actual traffic reveals that delay or may be blocking situations can occur on the egress ports. However, we have not considered its effects in our analysis presented in this work.

Using a computer with a 3.3 GHz CPU and 8 GB memory, our heuristic produced the results in less than one minute considering 1000 IOI VMs. This is a remarkable improvement over the MILP model which requires about 2 days to produce the results for only 20 IOI VMs using the same computer.

Demands \ VM Type	PI	MI	IOI
CPU (GHz)	2-3.6	0.1-0.3	0.1-0.3
Mem (GB)	0.1-0.3	6-8	1-4
IO (Gbps)	0.5-1	0.5-1	6-10
CPU-M Traffic (Gbps)	10-100	10-50	5-20
CPU-IO Traffic (Gbps)	1-3	1-3	1-3
M-IO Traffic (Gbps)	1-3	1-5	6-10

Table 5-2: Input parameters for the VMs requirements

5.6 Model and Heuristic Results

Fig. 5-4 compares the power consumption results of the MILP model for the DS and CS designs and the DS heuristic for 20 VM requests and considers the three VM types PI, MI and IOI. It shows clearly that the DS heuristic results and the DS MILP results are comparable and the heuristic follows the MILP closely.

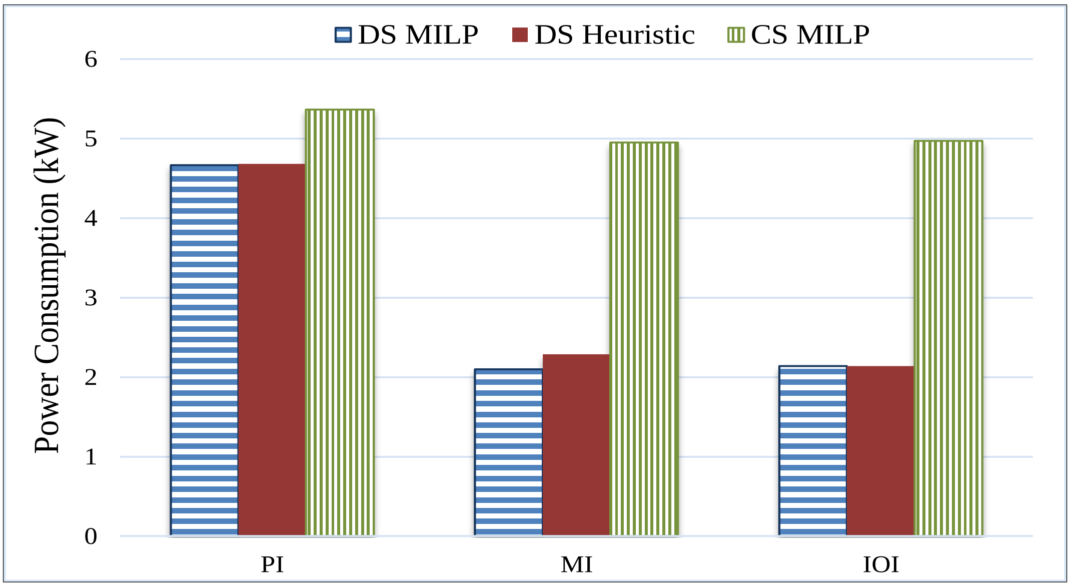
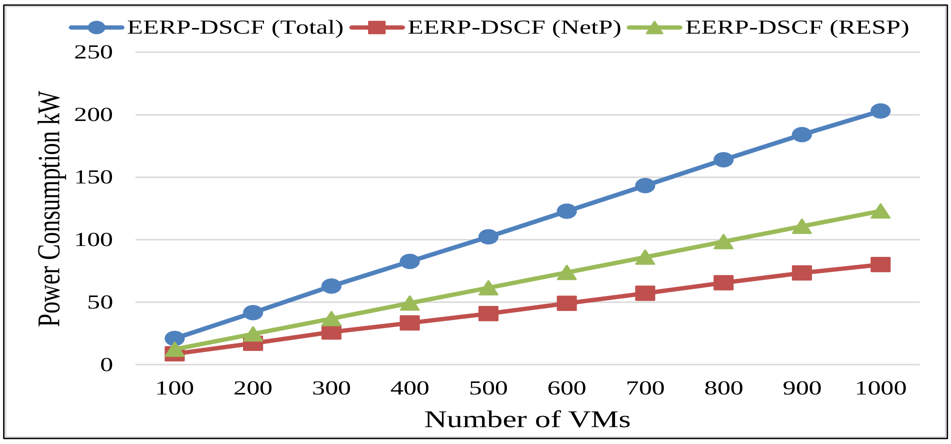


Fig. 5-4: Power consumption comparison of the DS MILP model, DS heuristic and CS MILP with communication fabric for 20 VM requests

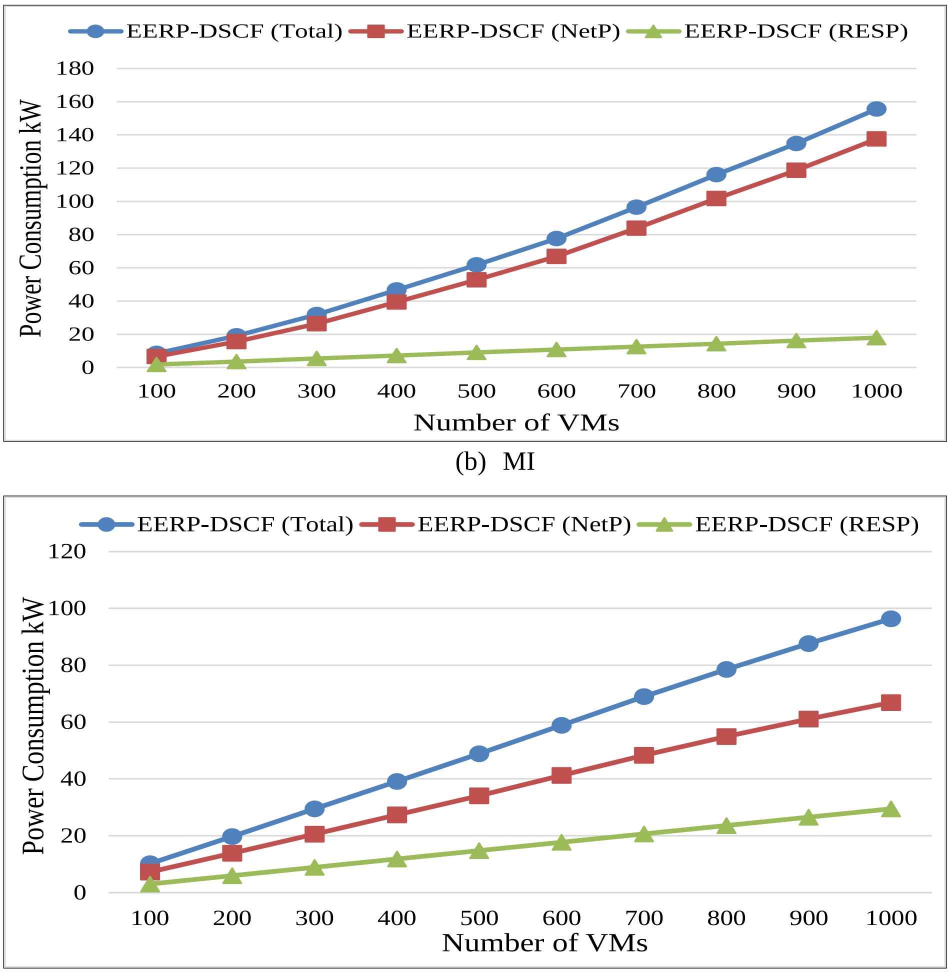
Examining the results in Fig. 5-4 and comparing the DS MILP to the CS MILP shows that the PI VMs are the highest power consuming demands which leads to minimum power saving, about 3% while MI and IOI have comparable power consumptions and they have comparable power savings, about 42%.

Fig. 5-5 shows the DS heuristic results for the power consumption of the networking resources, the resource provisioning power consumption and the total power consumption. The results can be explained by considering the cluster topology and size under consideration plus the number of served VMs, and the inputs, in particular the resources specifications and VM requirements.

Fig. 5-5.a shows the PI VM requests results. Note that the resource power is higher than the networking power and it has higher impact on the total power consumption. Given the parameters in Tables 5.1 and 5.2 for the resources power consumption and VM demand values, the average CPU demand per VM is 3 GHz for the PI VMs type, thus huge number of CPUs will be used for VM allocation. As the CPU has high power consumption values, the resource power consumption is the highest and exceeds the network power consumption.



(a) PI



(b) MI

(c) IOI

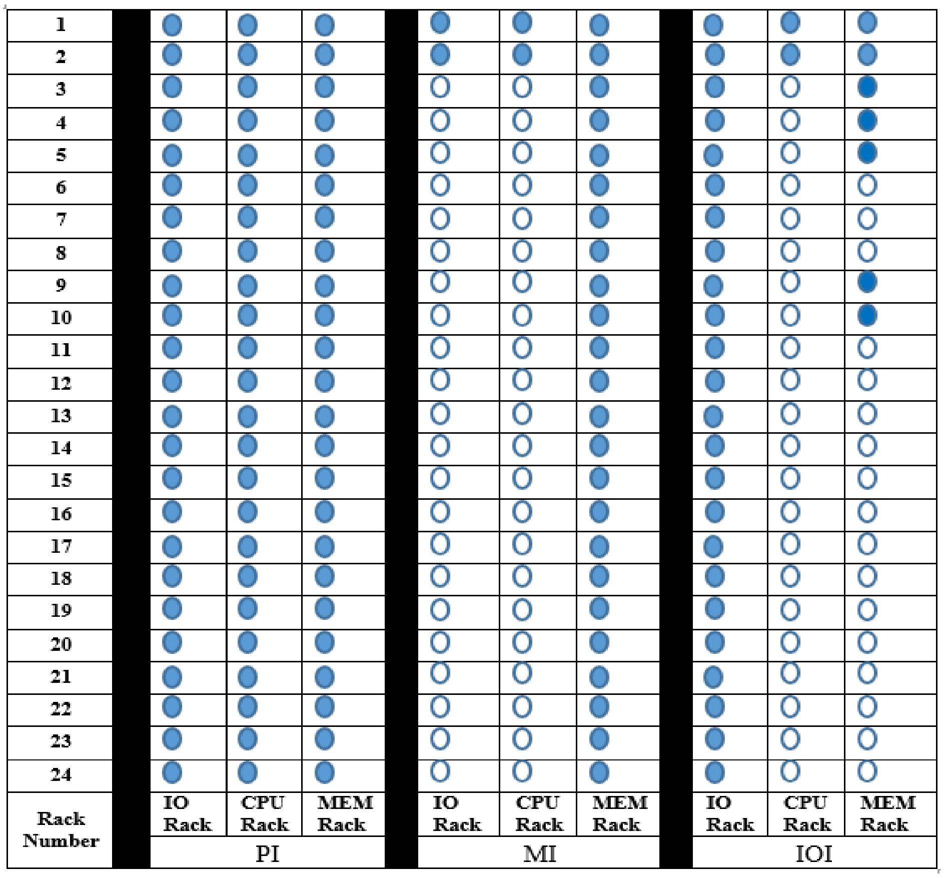
Fig. 5-5: Power consumption of the EERP-DSCF with large number of VMs

In relation to the networking power, Fig. 5-6 shows the active racks of each type when serving 1000 VM requests for the three VM types, PI, MI and IOI. Examining Fig. 5-6 reveals that considering 1000 PI VM requests results in a case where all the racks in the cluster are activated, regardless of their utilisation, where an active rack means there is an outgoing/incoming traffic. Clearly each CPU rack has an active memory and IO racks among its neighbours, enforced by the heuristic, thus all traffic in this case will be a single hop traffic resulting in about 80 kW networking power which is less than the power consumed by the resources.

Fig. 5-5.b shows the evaluation for the MI VM requests and compares the two power components, the network and resources powers. As in PI, the network power and resource power consumption increase with increasing number of VMs. The increase in resource power is far less however than the increase in network power consumption as memories consume few watts and only two CPU racks are enough to accommodate the processing requirements. Inspecting Fig. 5-6, and the MI results specifically, it can be seen that all the memory racks are used but only two CPU racks and two IO racks are used. This is due to the approach followed by our heuristic when performing the resource allocation and packing. The heuristic first allocates the best available CPU in order to reduce the number of working CPUs, resulting in only two working CPU racks, whereas the memory and IO allocations follow the CPU allocation by choosing the closest memory and IO racks to the used CPU rack that have enough capacity to accommodate the VM under consideration. Thus the traffic due to these CPU racks destined to the memory racks, which typically has moderate values, has to travel through long paths passing a significant number of multi-hop links, further increasing the network power consumption. In the same manner the traffic from the memory racks to the IO racks traverses almost the whole cluster, in some cases, to reach its destination. This leads to about 138 kW networking power consumption following resource packing which reduces the number of active resources (processing resources especially) which results in the minimum resources power consumption.

Fig. 5-5.c shows the power consumption of the DS heuristic for IOI VMs. As the IOI VM type demands are the lowest among the other VM types on average in terms of resource requirements, and due to both good resource packing for this energy efficient heuristic, and traffic routing by the heuristic, this scenario resulted in the minimum network power consumption and minimum resources power consumption. Fig. 5-6 reveals that all IO racks are working in addition to only two power consuming CPU racks and seven memory racks. Again this can be explained by observing the way the heuristic works. The heuristic first priority is to allocate the VMs in the smallest number of power hungry CPUs. After that, the memory and IO modules allocation follows the CPU allocation by choosing the closest available resources to the used CPUs. It can clearly be seen in Fig. 5-6 that the heuristic preferred memory racks # 9 and #10 instead of #6, #7 and #8. Examining Fig. 5-2 shows that CPU racks 1 and 2 are closer to memory racks 9 and 10 compared to racks 6, 7 and 8. To show the effect of the inclusion of the communication on the overall power saving, Fig. 5-7 compares the total power consumption of the CS data centre design to the power consumption of the DS based data centre design when the latter includes the power consumption attributed to communications. The CS approach is implemented as a heuristic where the total number of coupled resources required to form server units to serve incoming VMs, are determined, then 150 W is added to the power consumption of the resources of each active server to account for the internal communication overhead.

Fig. 5-7 shows that the average power saving for the PI DSCF is about 10% compared to the PI CSCF. This is due to the use of the power hungry processing resources in both DS and CS designs to high extent compared to the number of used memory and IO resources in the DS. However, due to the DS ability to pack higher number of VMs in fewer resources (i.e. memory and IO in this case), DS managed to save a significant fraction of the power compared to CS. Regarding the networking power consumption, as discussed earlier in this section, the allocation of VMs in the DS racks led to a communication pattern such that all traffic passes single hop paths leading to moderate network power consumption in spite of having high traffic values associated with the PI VM demands, leading to overall total power saving compared to CS.



Rack Number	PI			MI			IOI		
	IO Rack	CPU Rack	MEM Rack	IO Rack	CPU Rack	MEM Rack	IO Rack	CPU Rack	MEM Rack
1	●	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●	●
3	●	●	●	○	○	●	●	○	●
4	●	●	●	○	○	●	●	○	●
5	●	●	●	○	○	●	●	○	●
6	●	●	●	○	○	●	●	○	○
7	●	●	●	○	○	●	●	○	○
8	●	●	●	○	○	●	●	○	○
9	●	●	●	○	○	●	●	○	●
10	●	●	●	○	○	●	●	○	●
11	●	●	●	○	○	●	●	○	○
12	●	●	●	○	○	●	●	○	○
13	●	●	●	○	○	●	●	○	○
14	●	●	●	○	○	●	●	○	○
15	●	●	●	○	○	●	●	○	○
16	●	●	●	○	○	●	●	○	○
17	●	●	●	○	○	●	●	○	○
18	●	●	●	○	○	●	●	○	○
19	●	●	●	○	○	●	●	○	○
20	●	●	●	○	○	●	●	○	○
21	●	●	●	○	○	●	●	○	○
22	●	●	●	○	○	●	●	○	○
23	●	●	●	○	○	●	●	○	○
24	●	●	●	○	○	●	●	○	○

Fig. 5-6: Active racks considering 1000 VM requests

Fig. 5-7 shows that the total power saving for the MI DSCF design is about 53% compared to the MI CSCF design. The higher saving percentage for the MI results compared to the previous PI results is due to the efficient utilisation of the power intensive CPU resources in the DS compared to the CS which could power on large number of servers due to the congestion on the memory modules.

In relation to the networking power consumption, serving MI requests increases the network power consumption in DS as compared to PI and IOI demands. Having a large number of working memory racks and a lower number of working CPU and IO racks increases the traffic among these racks and many traffic flows pass multi-hop paths. However, the DS total power consumption (resources plus networking) is still lower than the CS total power consumption as CS operates large number of servers, roughly the same as the number of powered-on memory modules in the DS, which maintain the higher CS power consumption compared to DS.

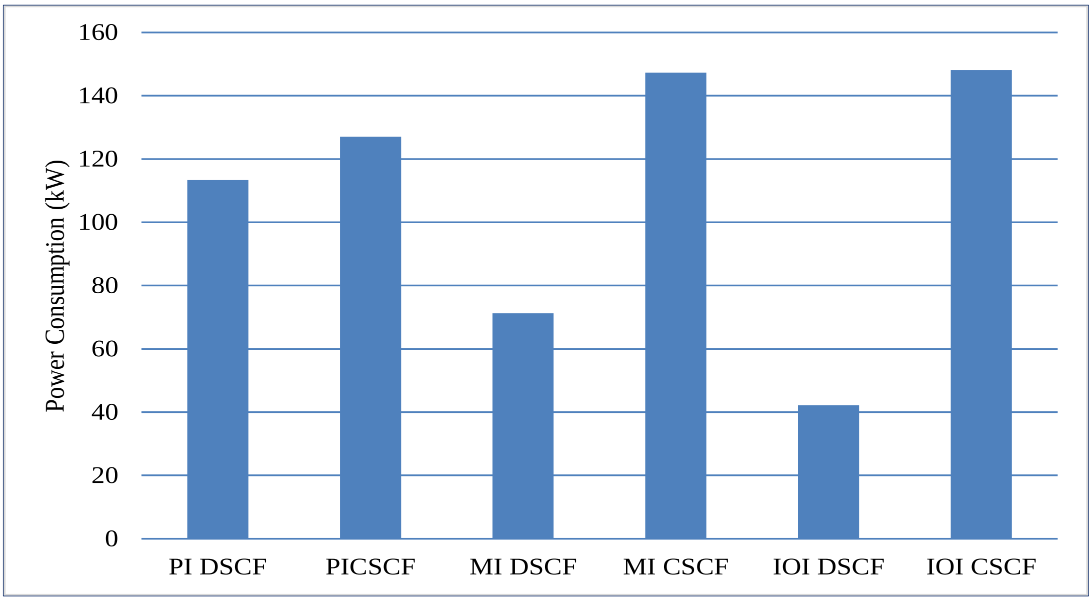


Fig. 5-7: Average power consumption of EEPR-DSCF compared to CS with communication fabric for large number of VMs ranging from 100 to 1000 VMs and considering the PI, MI and IOI VM types.

Similarly, the IOI VMs scenario resulted in the highest power saving, with an average power saving of about 63% compared to CS. For the resource power, the IOI VMs consume the least power compared to the PI and MI scenarios. According to Fig. 5-6, the communication power, as explained earlier, is the smallest among other VM types, thus, the IOI total power is the smallest among other VM types, which yielded the highest power saving.

Finally, we investigate the other extreme scenarios represented by mixed VM demands such as PI+MI VMs or PI+MI+IOI VMs. Table 5-3 captures the requirements for these mixed VM types. Note that we selected the maximum demand values in each set of VM combinations to establish the power savings limits.

The VMs requirements, resources and traffic values, have been chosen to cover the extreme values for the considered types to cover a variety of VM categories.

Fig. 5-8 shows the total power consumption of the four different scenarios mentioned in Table 5-3, for both DS and CS servers.

VMType Demands	PI+MI	PI+IOI	MI+IOI	PI+MI+IOI
CPU (GHz)	2-3.6	2-3.6	0.1-0.3	2-3.6
Mem (GB)	6-8	1-4	6-8	6-8
IO (Gbps)	0.5-1	6-10	6-10	6-10
CPU-M Traffic (Gbps)	10-100	10-100	10-50	10-100
CPU-IO Traffic (Gbps)	1-3	1-3	1-3	1-3
M-IO Traffic (Gbps)	1-5	6-10	6-10	6-10

Table 5-3: Input mixed VMs resources and traffic requirements

The MI+IOI and PI+MI scenarios have the highest power savings, about 28% and 27% respectively. For the first scenario, MI+IOI, due to the low CPU demands and relatively low M-IO and CPU-IO traffic values; and low CPU-M traffic values, compared to the other VM types, this scenario resulted in the best power profile followed by the PI+MI scenario. The mixed PI+MI scenario has higher CPU demand compared to the MI+IOI VMs scenario but lower IO demand and M-IO traffic compared to the other scenarios, PI+IOI and PI+IOI+MI. Thus it has higher power savings than both but lower than MI+IOI due to its high CPU demand and CPU-M traffic. The last two scenarios, PI+IOI and PI+MI+IOI resulted in the minimum power savings 17% and 15% respectively. This is consistent with the resources demands and traffic values required by these VM types.

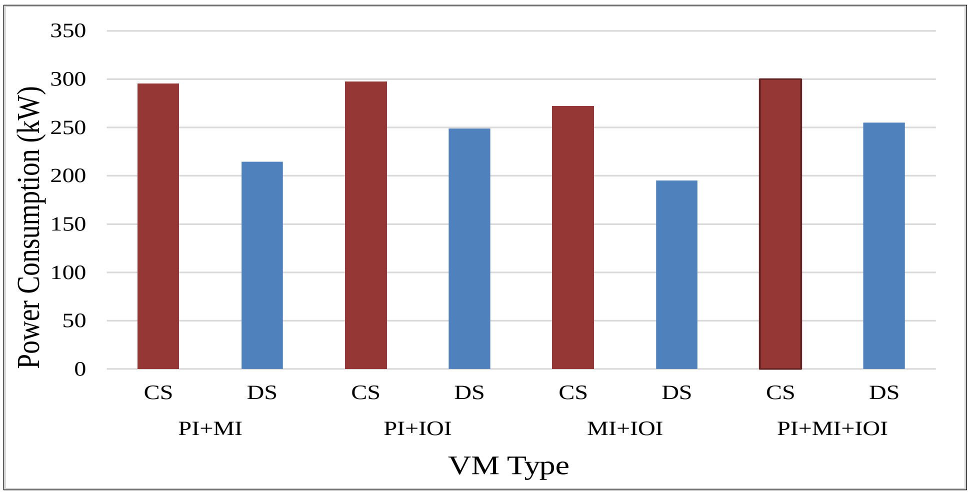


Fig. 5-8: Power consumption of DS and CS heuristics considering variety of 1000 VM requests

Regarding the traffic values, Fig. 5-9 is a case study which shows the effect of having high CPU-M traffic on the total power saving. This case study investigates the highest power saving scenario, IOI VMs, and it shows clearly that increasing the high CPU-M traffic increases the DS power consumption until it reaches the point where both designs have the same power profiles, around 225 Gbps traffic rate.

With increase in the traffic, the CS design gives better power profile than the DS design. However, the evaluations conducted in this chapter are based on current technologies with very conservative assumptions. With future improved communications technologies, the DS architecture is expected to be a promising choice over several dimensions and especially the energy saving dimension.

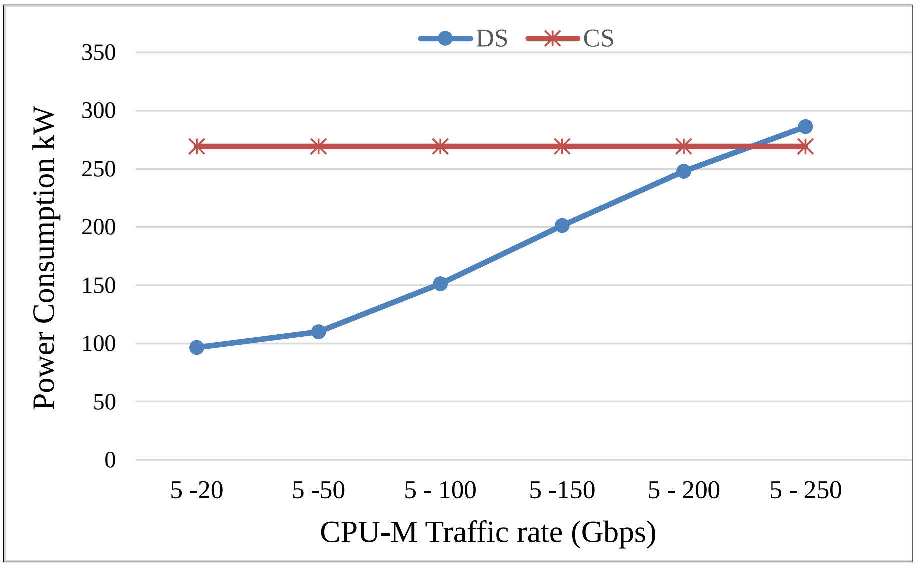


Fig. 5-9: Heuristic results showing the effects of increasing the CPU-M traffic on the total power consumption considering 1000 IOI VMs.

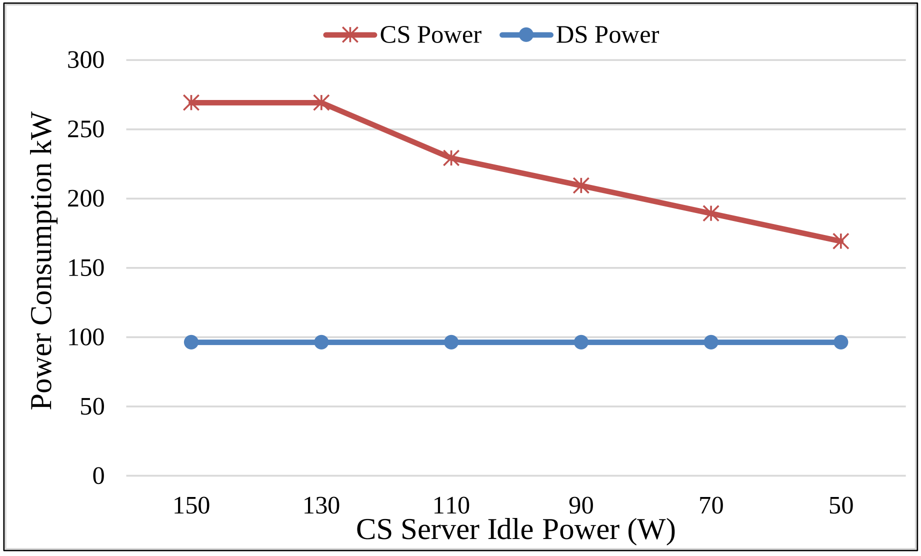


Fig. 5-10: Heuristic results showing the effect of decreasing the CS server idle power on the power savings considering 1000 IOI VMs.

Another issue is the CS server idle power, in current work and as mentioned in Section 5.3, the CS server idle power was set to 150 W. For a full and comprehensive evaluation, variety of idle power values have been considered and the CS power was compared to the DS server power. The heuristic considered 1000 IOI VMs. The results of these evaluations are shown in Fig. 5-10. In Fig. 5-10, the CS idle power was reduced from 150 W to 50 W and in each step the CS power was compared to the DS power. Fig. 5-10 shows that the CS power consumption is reduced with reduction in the idle power but even at 50 W idle power, the DS server is still more efficient than the CS design.

5.7 Summary

In this chapter, we have investigated the energy efficiency of resource provisioning and VM allocation in the DS based data centres compared to CS based data centres with consideration for the communication power by including the communication fabric components in the evaluation. A MILP optimisation was developed for the purpose to optimise VM allocation for DS based data centre considering the communication fabric power consumption. The results show that considering pooled resources yields considerable power savings compared to the CS approach. Both MI VMs and IOI VMs scenarios have shown comparable power profiles and up to 42% total power saving was achieved based on the MILP optimised system and about 3% power saving was achieved for the PI scenario. For real time implementation, we have developed an energy efficient resource provisioning heuristic for DS (EERP-DSCF) based on the model insights with comparable power efficiency to the MILP. With heuristic, and due to its fast nature, we extended the size and number of served VMs. Up to 63% average power saving was achieved when serving IOI VMs, 53% when serving MI VMs, and 10% when serving PI VMs.

Chapter 6: Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server(EERPVMM-DS)

6.1 Introduction

This chapter introduces an energy efficient heuristic that performs energy efficient resource provisioning and VM migration in the DS (EERPVMM-DS) schema [99]. We examined 1000 VM requests that demand various processing, memory and IO requirements. Requests have exponentially distributed inter arrival time and uniformly distributed service duration periods. Resources occupied by a certain VM are released when the VM finishes its service duration. The heuristic optimises VM allocations and dynamically migrates existing VMs to occupy newly released energy efficient resources. We assess the energy efficiency of the heuristic by applying increasing service duration periods. The results of the numerical simulation indicate that our power savings can reach up to 55% when compared to our previous study where VM service duration is infinite and resources are not released.

6.2 Virtualisation and VM Migration

Virtualisation and as mentioned in Chapter 2 is a key technology for cloud computing as it allows several VM instances to be embedded on the same physical machine by running them on top of a software layer known as a hypervisor. The hypervisor enhances resources manageability by simulating the underlying hardware platform and provisions hardware resources to the VM requests, and consequently consents virtualisation which brings set of new services and applications to the data centres [88]. In addition to enabling higher utilisation of hardware resources, by packing VMs in minimum underlying physical resources, virtualisation facilitates VM movement from one host to another which is an important feature considering server consolidation, power consumption and data centre manageability.

The migration procedure is a heuristic approach. VMs are instated by hosting them in the optimal available server that meets some criteria such as: resource availability, energy efficiency and latency bounds. If a VM finishes its service duration and releases its occupied resources in a particular server, the following qualifications must be guaranteed before performing VM migration to that server: (i) the target server has enough capacity to host the migrated VM, (ii) the migration will save some resources, (iii) the migration will not increase the migrated VM duration. If all these conditions are satisfied, then the migration can be done [100]. Thus, recently there has been a huge shift toward virtualised data centres in order to address traditional data centres limitations, improve the data centre efficiency, improve resource utilisation, simplify resource management and reduce power consumption.

6.2.1 EERPVMM-DS Heuristic Approach

In this work we develop a heuristic that completes our work which appeared in Chapter 3 and in [95] by considering VMs having finite service duration. The heuristic in Chapter 3 and in [95] assumed that VMs have infinite service durations, therefore, VMs are assumed to arrive all at once and occupy the resources permanently. This might be a reasonable assumption for certain classes of VMs, such as IaaS [101], however, many other VM requests are established for a limited service duration, such as Amazon AWS [102]. We consider VM arrivals that follow a Poisson distribution, i.e. associated with exponentially distributed IAT. The IAT reflects how frequently jobs (i.e. VMs) are being submitted to the data centre while the service duration is the total period a VM needs to finish its processing task and then leaves the server.

The EERPVMM-DS heuristic, Fig. 6-1, aims to pack incoming VMs in the minimum number of resources with minimal power to be consumed. According to equation (6-1), resource power consumption is proportional to its Power Factor (PF, equation (6-2)), and its utilisation (δ). Therefore, resources with small PF are the optimum candidates to host a VM.

$$Power = PMin_j^x + PF_j^x \cdot \delta x_j \tag{6-1}$$

$$PF_j^x = \left(\ PMax_j^x - \ PMin_j^x \right) \tag{6-2}$$

where $PMax$ and $PMin$ are the maximum active power, idle power, respectively. In this study the resource type can be processor, memory or IO port. The heuristic first assesses the resources according to their PF and capacity. The heuristic then constructs sorted lists of the resources (i.e. one list per resource type) by organising them in an ascending order starting with resources having smallest PF, and moving on to resources with higher PF, and if two resources of similar type have the same PF, then the resource with the highest capacity is favoured. The DS is presented with a set of requested VMs. All VMs information such as service durations and IATs are known for the heuristic at the beginning of the run time and from this information, the heuristic finds the total required time slots to serve all the requested VMs. The VMs to be served in each time slot are obtained by knowing the IAT of each VM and its service duration. A time slot is a unit of time and it could be in seconds, minutes or hours. Each VM duration is represented by the number of time slots it needs. Each VM is expected to occupy some resources for at least one time slot.

Then the heuristic creates a list of the number of VMs in each time slot by arranging the VMs in the time slots according to their arrival time and service duration. For example, if VM1 arrives at the first time slot and needs 3 time slots then it will appear in time slots 1, 2 and 3. Similarly, if VM2 arrives at the third time slot and needs 7 time slots then it will appear in time slots 3-9. Subsequently, the heuristic must loop for all the time slots and serve the requested VMs in each time slot.

For each time slot, and for each VM in this time slot, the heuristic selects one of the resources from the top of each sorted list; recall that there are three lists, processors list, memories list and IO ports list. Similar to our work in [95], the heuristic checks the chosen resources to find out if there is enough capacity on each resource type to serve the VM request. First the processor is tested, if it does not have enough capacity then the heuristic picks up a new processor from the processor sorted list, otherwise, the heuristic tests the selected memory. Again, if the memory does not have enough capacity to serve the VM under consideration, a new memory must be retrieved from the memory list; otherwise the selected IO port must be tested. If the IO port can accommodate the network traffic requirements of the VM under consideration, it is used directly; otherwise a new IO port must be retrieved from the IO list.

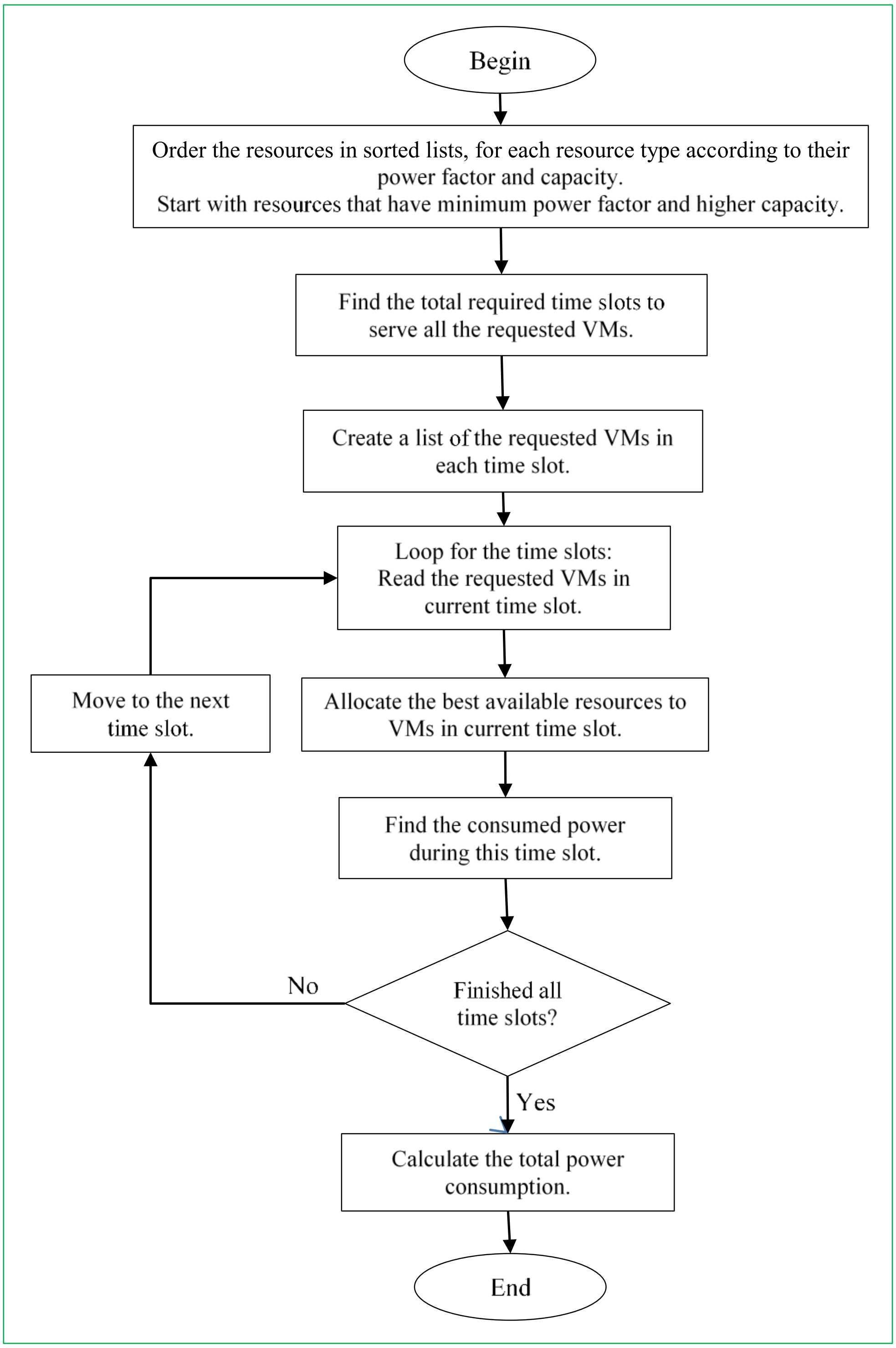

Begin
Order the resources in sorted lists, for each resource type according to their power factor and capacity.
Start with resources that have minimum power factor and higher capacity.
Find the total required time slots to serve all the requested VMs.
Create a list of the requested VMs in each time slot.
Loop for the time slots:
Read the requested VMs in current time slot.
Allocate the best available resources to VMs in current time slot.
Find the consumed power during this time slot.
Finished all time slots?
No
Move to the next time slot.
Yes
Calculate the total power consumption.
End

Fig. 6-1: EERPVMM-DS flowchart

After choosing all three resources that can fulfil the current VM demands, the heuristic allocates these resources to the VM under consideration and updates the used resources’ remaining capacity. In each time slot, the heuristic tries to occupy the remaining capacity of the already utilised resources by packing them with as many requests as they can serve and if any one of them cannot serve more VM requests then the heuristic proceeds to the next resource in the sorted list and uses it. For each time slot and after serving all the VMs in the current time slot, the heuristic calculates the total power consumed in that time slot.

After that the heuristic moves on to the next time slot where the resources occupied by VMs that finished in that time slot are to be released. The loop continues by repeating the above steps until all VMs are served. Finally, the heuristic calculates the average power consumption of the DS due to serving the current VMs set.

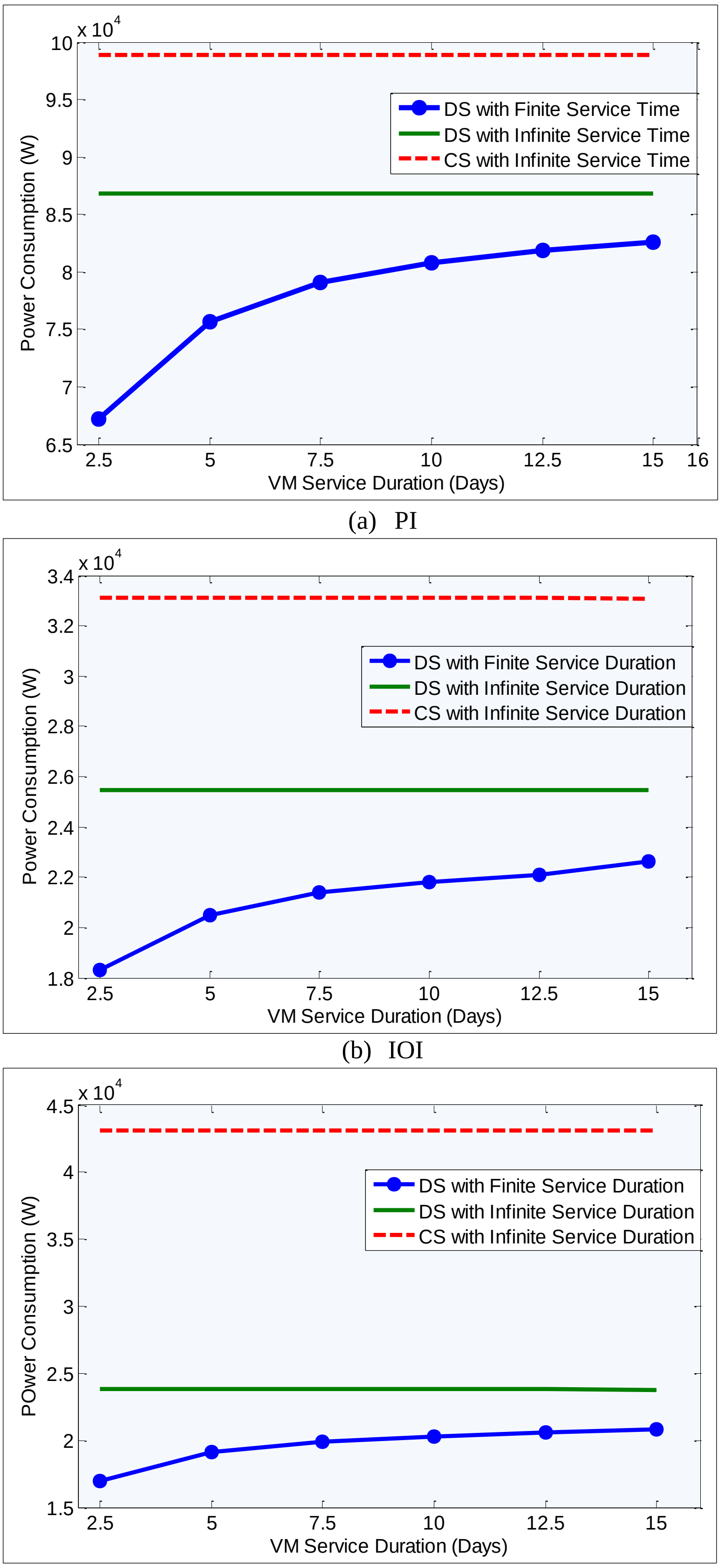
6.2.2 Results

In this section we compare our current EERPVMM-DS heuristic and our previous heuristic described in Chapter 3, to show how considering VMs with finite serving duration affects the DS power consumption. We assume that the VMs IAT is exponentially distributed with mean of 1 minute, considering that we are dealing with data centre of average access rates[103, 104]. The IAT spans from few seconds to a maximum value of 5 minutes, and the serving durations are generated using the uniform distribution. We consider the same three types of VMs as in Chapte 3: (i) Processing Intensive Requests (PI), (ii) Memory Intensive Requests (MI) and (iii) IO Intensive Requests (IOI).

For the simulation and evaluation of the DS architecture we are considering a set of heterogeneous resources by disaggregating the IBM X3650 M3 server system [61] and using the same parameters in Chapter 3 and [94]. Note that, given our set of resources, processors are the most intensive power consumers and memory resources are the least power consumers, while the IO ports' power consumption lies between the two.

The results in Fig. 6-2 show the power consumption considering the DS and CS power consumption with infinite service durations, which appeared in Chapter 3, and DS power consumption with finite VM service durations. The graphs show clearly that considering VM with finite service durations with dynamic resource allocation has a positive impact on the total data centre power efficiency.

In our simulation, we evaluated the power consumption of the different data centres designs considering 1000 VM requests, where each request has different resources requirements. We assessed the DS with a range of VMs serving durations to show the effects of increasing the VM average service duration on the DS power consumption. Our findings show that when dealing with low service durations, resources can be used more efficiently and freused as the number of VMs that finish their serving duration and leave the system are considerably high. Thus the resources can be reused to serve new incoming VMs, which eliminates or reduces the need to turn-on new resources. This is to be compared to our previous analysis in Chapter 3 where VMs were considered to have an infinite service duration and all VMs arrive at once.



(a) PI

(b) IOI

(c) MI

Fig. 6-2: Power consumption evaluation considering CS and DS with infinite service duration and DS with finite service duration

By increasing the VM service duration, the number of VMs that leave the data centre decreases, resulting in continuous increase in the number of working resources, and consequently increase in the total power consumption. As limited number of VMs are leaving the system, new arriving VM requests will probably not be able to fit in any of the already working resources, thus new resources will be needed to be turned on, leading to an increase in the number of working resources.

Investigating Fig. 6-2 reveals that when handling VMs with average service duration around 15 days, which could be any cloud service such as SaaS or PaaS [101], our new heuristic will approach in behaviour our old heuristic in [94], which indicates that almost all the incoming VM requests are staying in the system as if they have infinite durations, and the used resources remain under the occupancy of the already arrived VMs.

From Fig. 6-2, it is apparent that serving VM requests with finite service duration reduces the power consumption of the DS compared to the case when VMs were assumed to have infinite service duration, and the average power savings for the considered range of VM service durations are 10% for the PI, 17% for IOI, and 18% for the MI. This is due to our given input parameters for the different resource types. Note that PI VMs scenario has the least power saving as the saving will be from the efficient use of the lower power consuming resources, i.e. the memory and IO ports, as compared to the high power consumption of processors. When considering the other two scenarios, i.e. MI and IOI VMs, the higher power saving will be achieved through the efficient use of the most power hungry processing resources, thus these scenarios will attain a higher power savings compared to the PI scenario.

What is interesting in the data given by Fig. 6-2 is that, comparing the power consumption of the EERPVMM-DS heuristic to the old CS scenario unveils massive power savings. About 55% of the consumed power in the old CS scenario can be saved when considering time for the MI VMs type, 36% for the IOI VMs type, and 21% for the PI VMs type in average.

This behaviour asserts the fact that considering VMs having finite service time, the resources will not be loaded with all VMs at the same time. Rather, after a VM finishes its serving duration, the used resources will be vacated from this VM, and therefore we can perform VM migration by moving VMs that still need further processing to vacated resources. Thus the most efficient resources, from the top of the resources' lists will be used more likely than other resources as they will be reused after being vacated from finished VMs. This alongside the better resource packing are the main factors that achieve this improved power efficiency.

In addition, the VM service durations are exponentially distributed which reduces the power consumption which means that the average service duration that is needed to produce the same performance as our old heuristic with infinite service duration is more than 15 days. Also increasing the mean IAT value to more than 1 minute reduces the total power consumption values.

6.3 Summary

In this chapter, we introduced our new energy efficient EERPVMM-DS heuristic which performs resource provisioning and VM migration in the DS paradigm with finite VM service time. For simulation, we considered 1000 VM requests which have different processing, memory, IO, and serving duration demands, and considered a range of heterogeneous resources to be used for performing resource provisioning. The VMs are assumed to arrive with exponentially distributed inter arrival times and request uniformly distributed service durations. The heuristic optimises VMs allocation and dynamically migrates VMs to occupy newly released energy efficient resources. The heuristic results showed that the power consumption of the data centre has been reduced remarkably as compared to our work which appeared in Chapter 3 that compares CS with infinite service duration and DS with infinite service duration. Respectively, compared to the CS and DS with infinite service duration, EERPVMM-DS power savings are 55% and 18% when considering MI VM requests, 36% and 17% for the IOI VM requests and 21% and 10% when considering PI VM requests, on average.

Chapter 7: Conclusions and Future Work

In this chapter, the work presented in the thesis is summarised and the original contributions are specified. For the future, potential new directions for research that can be conducted as a result of work in this thesis are suggested.

7.1 Summary of Contributions

In this thesis, an investigation is reported considering the problems of optimising the energy consumption of resource provisioning in DS-based data centres considering three scenarios, resource provisioning in DS-based data centres considering power consumption of the processing resources, resource provisioning in DS-based data centres considering both resources and communication power, and resource provisioning in DS-based data centres with VM migration, and resources reallocation considering resources power only. These different research problems have been investigated and the first two problems have been formulated as a MILP model, and then a heuristic, whereas the third problem has been formulated only as a heuristic. A new and innovative DS modular architecture was developed and introduced as a promising server paradigm for future data centres.

In Chapter 3 the energy efficiency of VM placement in a disaggregated data centre approach has been investigated and the power savings of the new approach has been evaluated. The approach considered enables the separation of the computing, memory, storage, and network resources of the server leading to better resource utilisation by "composing on the fly" servers with the exact required processing, memory, and IO capabilities to accommodate the virtual machines or tasks of interest. A MILP optimisation model, which optimally places VMs in the disaggregated data centre with the objective of minimising the power consumption, has been developed. It was compared to a data centre using the normal racks of server units. Here consideration was given to the VM placement and resource provisioning operations. To gain a good view of the operation of the proposed approach, three types of VMs – PI, MI, and IOI – have been considered in the model. The results show that, given the set of input parameters used, a DS approach can reduce the energy consumption of resource provisioning by 11%, 49%, and 24% compared to the CS server considering the PI, MI, and IOI VMs, respectively. An EERP-DS heuristic has been developed and the results showed that the average power savings were 60% when serving MI requests, 36% for IOI requests, and 11% when serving PI requests (under the set of typical parameters and conditions we considered).

In Chapter 4 we introduced a new photonic communication fabric for the DS-based data centre. Our innovative modular and switch-based DS design was discussed with proposed techniques for intra- and inter-racks communication. Communication challenges such as latency and switching speed have to be addressed by this design, including the specifications of the equipment used such as power, speed, and reliability, to guarantee a reliable and fast communication among disaggregated resources. Some recommendations for the design implementation have been presented in consistency with some defined metrics for server disaggregation, with a comprehensive description of the difficulties and challenges that could face this architecture.

In Chapter 5 we studied the problem of VM placement in a DS-based data centre by analysing the energy efficiency of resource provisioning and VM allocation in DS server design, with considerations for the communication fabric power consumption, and compared it to the CS approach. An MILP optimisation was developed for the purpose of optimising VM allocation for the DS-based data centre while considering the communication fabric power. The results show that considering pooled resources with the communication power yields considerable power savings compared to the CS approach, and up to 42% total power saving was achieved form the MILP model. For real-time implementation, EERP-DSCF heuristic was developed based on the model insights with comparable power efficiency to the MILP.

In Chapter 6 we introduced our new energy efficient EERPVMM-DS heuristic that performs resource provisioning and VM migration in DS server paradigm. The EERPVMM-DS heuristic was used to study the impact of adding service duration as a new dimension to the VM requirements. For simulation, we considered 1000 VM requests that have various processing, memory, IO, and serving duration demands, and considered a range of heterogeneous resources to be used for performing resource provisioning. The VMs are assumed to arrive with exponentially distributed inter-arrival times and request uniformly distributed service durations. The heuristic optimises VMs allocation and dynamically migrates VMs to occupy newly released energy-efficient resources. The heuristic results showed that the power consumption of the data centre has been reduced remarkably as compared to our old work in Chapter 3 that compares CS with infinite service duration and DS with infinite service duration. Respectively, compared to the CS and DS with infinite service duration, EERPVMM-DS power savings are 55% and 18% when considering MI VM requests, 36% and 17% for the IOI VM requests, and 21% and 10% when considering PI VM requests, on average.

The results show that the DS based data center was able to achieve higher energy savings while fewer number of resources are required to serve the same set of incoming VMs compared to the CS based data center. Considering some extreme circumstances, the CS and DS server designs will give the same results. In terms of other factors that affect the performance of the data center, DS can achieve better resources modularity and fine grained levels of resource allocation.

7.2 Future Work

In this section several future directions for the topic of energy-efficient DS are proposed.

7.2.1 Extensions based on stated limitations

The first possible future work is to address the limitations of this thesis mentioned above. One conceivable direction is to validate the findings of this work that were investigated using MILP mathematical modelling by using experimental evaluation. Another possible direction is to consider other metrics in the objective function such as delay, reliability, blocking rate, and locality of used resources by setting limits on spanning distance among resources used by the same VM, etc. In addition, experimental evaluation can be conducted by benchmarking DS-based data centres.

7.2.2 Integrating queuing theory with MILP for resource provisioning and VM allocation in DS-based data centres

Another extension to this work can be conducted by considering VM scheduling. This can reformulate the problem in a wider context where resource placement can be jointly optimised with considerations for VM scheduling. This is better, in terms of optimisation scope and energy-efficiency gains, as each VM will be given a specific time slot and, upon finishing, it will be removed from the system and the vacated resources will be used to serve new or already available migrated VMs.

7.2.3 Considering geo-distributed DS-based data centre

Based on the IP/WDM network, applying the DS-based data centres and considering communication bandwidth among these distributed data centres will add a new prospect to the area better than isolating the data centres and optimising it separately from the rest of the core networks.

7.2.4 Applying virtualisation at higher levels

This can be implemented by having different operators owning or managing big boxes of disaggregated resources and those operators can sell their resources to end users. Different operators can rent resources from each other and use them as if they were their own resources.

7.2.5 Mixing DS approach with PON data centres

Another interesting topic is to consider DS servers for PON data centres revising the MILP models and heuristics to reflect the new interconnection topology among the racks of disaggregated resources. However, some challenges will need to be tackled when considering the new DS server with PON data centres, such as latency, resource locality, and blocking.

References

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, 2008, pp. 5-13.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," *INC, IMS and IDC*, pp. 44-51, 2009.

[3] L. Chiaraviglio and I. Matta, "An energy-aware distributed approach for content and network management," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, 2011, pp. 337-342.

[4] Internetlivestats, "Internet Live Stats - Internet Usage & Social Media Statistics". [Online]. Available: http://www.internetlivestats.com. [Accessed: 12- Jan- 2017]."

[5] Intel and Tancent, "Tencent Explores Datacenter Resource Pooling Using Intel® Rack Scale Architecture (Intel® RSA)". [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/rsa-tencent-paper.pdf. [Accessed: 01- Dec- 2016]."

[6] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, "Network support for resource disaggregation in next-generation datacenters," in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, 2013, p. 10.

[7] J. Baliga, R. W. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, pp. 149-167, 2011.

[8] "L. Barroso and U. Hölzle The case for energy-proportional computing computer," *IEEE Computer*, vol. 40, pp. 33-37, 2007.

[9] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, 2008, pp. 63-74.

[10] A. Hammadi, T. E. El-Gorashi, and J. M. Elmirghani, "Energy-efficient software-defined AWGR-based PON data center network," in *Transparent Optical Networks (ICTON), 2016 18th International Conference on*, 2016, pp. 1-5.

[11] B. Kantarci, L. Foschini, A. Corradi, and H. T. Mouftah, "Inter-and-intra data center VM-placement for energy-efficient large-scale cloud systems," in *2012 IEEE Globecom Workshops*, 2012, pp. 708-713.

[12] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review,* vol. 39, pp. 68-73, 2008.

[13] A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Distributed energy efficient clouds over core networks," *Journal of Lightwave Technology,* vol. 32, pp. 1261-1281, 2014.

[14] L. Nonde, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *Journal of Lightwave Technology,* vol. 33, pp. 1828-1849, 2015.

[15] R. Soref, "The past, present, and future of silicon photonics," *IEEE Journal of selected topics in quantum electronics,* vol. 12, pp. 1678-1687, 2006.

[16] Intel, "Design Guide for Photonic Architecture". [Online]. Available: http://opencompute.org/assets/Uploads/Open_Compute_Project_Open_Rack_Optical_Interconnect_Design_Guide_v0.5.pdf. [Accessed: 01- Dec- 2016]."

[17] A. Kulseitova and A. T. Fong, "A survey of energy-efficient techniques in cloud data centers," in *ICT for Smart Society (ICISS), 2013 International Conference on*, 2013, pp. 1-5.

[18] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "GreenCloud: a new architecture for green data center," in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, 2009, pp. 29-38.

[19] G. Koutitas and P. Demestichas, "Challenges for energy efficiency in local and regional data centers," *Journal of Green Engineering,* vol. 1, pp. 1-32, 2010.

[20] B. Weihl, E. Teetzel, J. Clidaras, C. Malone, J. Kava, and M. Ryan, "Sustainable data centers," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, pp. 8-12, 2011.

[21] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 750-757.

[22] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega: flexible, scalable schedulers for large compute clusters," in *Proceedings of the 8th ACM European Conference on Computer Systems*, 2013, pp. 351-364.

[23] I. Gog, M. Schwarzkopf, A. Gleave, R. N. Watson, and S. Hand, "Firmament: fast, centralized cluster scheduling at scale," in *Proceedings of OSDI'16: 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, p. 99.

[24] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications surveys & tutorials*, vol. 13, pp. 524-540, 2011.

[25] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "IP over WDM networks employing renewable energy sources," *Lightwave Technology, Journal of*, vol. 29, pp. 3-14, 2011.

[26] J. Dai, M. M. Ohadi, D. Das, and M. G. Pecht, *Optimum cooling of data centers*: Springer, 2014.

[27] E. Ariwa, *Green Technology Applications for Enterprise and Academic Innovation*: IGI Global, 2014.

[28] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, 2011, pp. 143-164.

[29] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green IP over WDM networks with data centers," *Journal of Lightwave Technology*, vol. 29, pp. 1861-1880, 2011.

[30] B. Simmons, A. McCloskey, and H. Lutfiyya, "Dynamic provisioning of resources in data centers," in *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*, 2007, pp. 40-40.

[31] C. G. Sheridan, K. A. Ellis, E. G. Castro-Leon, and C. P. Fowler, "Green Data Centres," *Harnessing Green IT: Principles and Practices*, p. 85, 2012.

[32] Intel, "The Case for Rack Scale Architecture". [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/rsa-introduction-paper.html. [Accessed: 01- Dec- 2016]."

[33] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," in *ACM SIGARCH Computer Architecture News*, 2009, pp. 267-278.

[34] Y. Yan, Y. Shu, G. M. Saridis, B. R. Rofoee, G. Zervas, and D. Simeonidou, "FPGA-based optical programmable switch and interface card for disaggregated OPS/OCS data centre networks," in *Optical Communication (ECOC), 2015 European Conference on*, 2015, pp. 1-3.

[35] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, 2010, pp. 222-226.

[36] Y. Katayama and A. Okazaki, "Optical interconnect opportunities for future server memory systems," in *2007 IEEE 13th International Symposium on High Performance Computer Architecture*, 2007, pp. 46-50.

[37] B. Abali, R. J. Eickemeyer, H. Franke, C.-S. Li, and M. A. Taubenblatt, "Disaggregated and optically interconnected memory: when will it be cost effective?," *arXiv preprint arXiv:1503.01416*, 2015.

[38] OSA, "Photonics for Disaggregated Data Centers Workshop," 2015.

[39] K. T. Malladi, B. C. Lee, F. A. Nothaft, C. Kozyrakis, K. Periyathambi, and M. Horowitz, "Towards energy-proportional datacenter memory with mobile DRAM," in *ACM SIGARCH Computer Architecture News*, 2012, pp. 37-48.

[40] G. F. Pfister, "An introduction to the infiniband architecture," *High Performance Mass Storage and Parallel I/O*, vol. 42, pp. 617-632, 2001.

[41] A. Pagès, J. Perelló, F. Agraz, and S. Spadaro, "Optimal VDC Service Provisioning in Optically Interconnected Disaggregated Data Centers," *IEEE Communications Letters*, vol. 20, pp. 1353-1356, 2016.

[42] P. Svärd, B. Hudzia, J. Tordsson, and E. Elmroth, "Hecatonchire: enabling multi-host virtual machines by resource aggregation and pooling," *Digitala Vetenskaplika Arkivet*, 2014.

[43] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, 2014, pp. 433-441.

[44] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient resource provisioning in compute clouds via vm multiplexing," in *Proceedings of the 7th international conference on Autonomic computing*, 2010, pp. 11-20.

[45] P. Costa, "Bridging the gap between applications and networks in data centers," *ACM SIGOPS Operating Systems Review*, vol. 47, pp. 3-8, 2013.

[46] Microsoft, "First International Workshop on Rack-scale Computing (WRSC 2014)". [Online]. Available:https://www.microsoft.com/en-us/research/event/first-international-workshop-on-rack-scale-computing-wrsc-2014/?from=http%3A%2F%2Fresearch.microsoft.com%2Fwrsc2014.[Accessed: 10- Jun- 2017]."

[47] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Towards a next generation data center architecture: scalability and commoditization," in *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, 2008, pp. 57-62.

[48] K. Lim, Y. Turner, J. Chang, J. R. Santos, and P. Ranganathan, "Disaggregated Memory Benefits for Server Consolidation"." *HP Laboratories*, 2011.

[49] R. Buyya, T. Cortes, and H. Jin, "An Introduction to the InfiniBand Architecture". [Online]. Available: http://buyya.com/superstorage/chap42.pdf. [Accessed: 10-Jun-2017]."

[50] Cisco, "Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments". [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-5020-switch/white_paper_c11-465436.pdf. [Accessed: 01- Dec- 2016]."

[51] K. Lim, Y. Turner, J. R. Santos, A. AuYoung, J. Chang, P. Ranganathan, and T. F. Wenisch, "System-level implications of disaggregated memory," in *IEEE International Symposium on High-Performance Comp Architecture*, 2012, pp. 1-12.

[52] K. T.-M. Lim, "Disaggregated memory architectures for blade servers". [Online]. Availabe: http://web.eecs.umich.edu/~tnm/trev_test/dissertationsPDF/kevinL.pdf. [Accessed: 10-Jun-2017]," Hewlett-Packard Labs, 2010.

[53] Y. Yan, G. M. Saridis, Y. Shu, B. R. Rofoee, S. Yan, M. Arslan, T. Bradley, N. V. Wheeler, N. H.-L. Wong, and F. Poletti, "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card," *Journal of Lightwave Technology*, vol. 34, pp. 1925-1932, 2016.

[54] G. Saridis, E. Hugues-Salas, Y. Yan, S. Y. Yan, S. Poole, G. S. Zervas, and D. E. Simeonidou, "DORIOS: Demonstration of an all-optical distributed CPU, memory, storage intra DCN interconnect," in *Optical Fiber Communication Conference*, 2015, p. W1D. 2.

[55] G. Saridis, Y. Yan, Y. Shu, S. Yan, M. Arslan, T. Bradley, N. Wheeler, N. Wong, F. Poletti, and M. Petrovich, "EVROS: All-optical programmable disaggregated data centre interconnect utilizing hollow-core bandgap fibre," in *Optical Communication (ECOC), 2015 European Conference on*, 2015, pp. 1-3.

[56] C.-C. Tu, C.-t. Lee, and T.-c. Chiueh, "Marlin: a memory-based rack area network," in *Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems*, 2014, pp. 125-136.

[57] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 267-280.

[58] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *ACM SIGCOMM Computer Communication Review*, 2015, pp. 123-137.

[59] K. Katrinis, D. Syrivelis, D. Pnevmatikatos, G. Zervas, D. Theodoropoulos, I. Koutsopoulos, K. Hasharoni, D. Raho, C. Pinto, and F. Espina, "Rack-scale disaggregated cloud data centers: The dReDBox project vision," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 690-695.

[60] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in computers*, vol. 82, pp. 47-111, 2011.

[61] IBM, "Product Guide IBM System x3650 M3". [Online]. Available: https://lenovopress.com/tips0805.pdf. [Accessed: 10-Jun-2017]," 2011.

[62] R. Sohan, A. C. Rice, A. W. Moore, and K. Mansley, "Characterizing 10 Gbps network interface energy consumption," in *IEEE Local Computer Networks LCN*, 2010, pp. 268-271.

[63] B. Akesson, K. Goossens, and M. Ringhofer, "Predator: a predictable SDRAM memory controller," in *Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, 2007, pp. 251-256.

[64] J. P. Hayes, *Computer architecture and organization*: McGraw-Hill, Inc., 2002.

[65] V. D. Suite, "RFC4175 Packetizer v1.0 LogiCORE IP Product Guide," 2016.

[66] Y. Zhang and N. Ansari, "HERO: Hierarchical energy optimization for data center networks," *IEEE Systems Journal*, vol. 9, pp. 406-415, 2015.

[67] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *Journal of Optical Communications and Networking*, vol. 1, pp. 176-186, 2009.

[68] K. Gharachorloo, D. Lenoski, J. Laudon, P. Gibbons, A. Gupta, and J. Hennessy, *Memory consistency and event ordering in scalable shared-memory multiprocessors* vol. 18: ACM, 1990.

[69] E. Vonnahme, S. Ruping, and U. Ruckert, "Measurements in switched Ethernet networks used for automation systems," in *Factory Communication Systems, 2000. Proceedings. 2000 IEEE International Workshop on*, 2000, pp. 231-238.

[70] Cisco, "Cisco Nexus 3064-X, 3064-T, and 3064-32T Switches".[Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/data_sheet_c78-651097.pdf. [Accessed: 01- Dec-2016]."

[71] Cisco, "Cisco SFS 7000P Infiniband Server Switch". [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/sfs-7000p-infiniband-server-switch/prod_bulletin0900aecd80337b11.pdf .[Accessed: 01- Dec- 2016]."

[72] M. Technologies, "M4001 16-port 40 and 56Gb/s InfiniBand Blade Switches". [Online]. Available: http://www.mellanox.com/related-docs/oem/dell/Dell_M4001.pdf. [Accessed: 01- Dec- 2016]."

[73] Calient. *CALIENT Optical Circuit Switch Brings Sub-60ns Latency to Data Centers ". [Online]. Available: http://www.calient.net/2012/09/calient-optical-circuit-switch-brings-sub-60ns-latency-to-data-center-financial-networks/. [Accessed: 01- Dec- 2016].*

[74] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *Journal of Lightwave Technology,* vol. 16, pp. 2081-2094, 1998.

[75] T. S. El-Bawab and J.-D. Shin, "Optical packet switching in core networks: between vision and reality," *IEEE Communications Magazine,* vol. 40, pp. 60-65, 2002.

[76] R. S. Tucker, P.-C. Ku, and C. J. Chang-Hasnain, "Slow-light optical buffers: capabilities and fundamental limitations," *Journal of lightwave technology,* vol. 23, pp. 4046-4066, 2005.

[77] B. A. Small, A. Shacham, and K. Bergman, "Ultra-low latency optical packet switching node," *IEEE photonics technology letters,* vol. 17, pp. 1564-1566, 2005.

[78] J. Luo, S. D. Lucente, J. Ramirez, H. J. Dorren, and N. Calabretta, "Low latency and large port count optical packet switch with highly distributed control," in *Optical Fiber Communication Conference*, 2012, p. OW3J. 2.

[79] EpiPhotonics, "High-Speed PLZT Optical Switches". [Online]. Available: http://epiphotonics.com/products.html. [Accessed: 04- Jan- 2017]."

[80] K. Nashimoto, N. Tanaka, M. LaBuda, D. Ritums, J. Dawley, M. Raj, D. Kudzuma, and T. Vo, "High-speed PLZT optical switches for burst and packet switching," in *2nd International Conference on Broadband Networks, 2005.*, 2005, pp. 1118-1123.

[81] B. Meagher, G. Chang, G. Ellinas, Y. Lin, W. Xin, T. Chen, X. Yang, A. Chowdhury, J. Young, and S. Yoo, "Design and implementation of ultra-low latency optical label switching for packet-switched WDM networks," *Journal of lightwave technology,* vol. 18, p. 1978, 2000.

[82] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," *IEEE network,* vol. 18, pp. 16-23, 2004.

[83] K.-W. Lee, J.-H. Cho, B.-J. Choi, G.-I. Lee, H.-D. Jung, W.-Y. Lee, K.-C. Park, Y.-S. Joo, J.-H. Cha, and Y.-J. Choi, "A 1.5-V 3.2 Gb/s/pin Graphic

DDR4 SDRAM with dual-clock system, four-phase input strobing, and low-jitter fully analog DLL," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 2369-2377, 2007.

[84] C. Park, H. Chung, Y.-S. Lee, J. Kim, J. Lee, M.-S. Chae, D.-H. Jung, S.-H. Choi, S.-y. Seo, and T.-S. Park, "A 512-mb DDR3 SDRAM prototype with C IO minimization and self-calibration techniques," *IEEE journal of solid-state circuits*, vol. 41, pp. 831-838, 2006.

[85] J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif, "On the use of fuzzy modeling in virtualized data center management," in *Fourth International Conference on Autonomic Computing (ICAC'07)*, 2007, pp. 25-25.

[86] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," in *ACM SIGARCH Computer Architecture News*, 2008, pp. 48-59.

[87] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *arXiv preprint arXiv:1006.0308*, 2010.

[88] P. Svärd, "Live VM Migration: Principles and Performance". [Online]. Available: http://www.diva-portal.org/smash/get/diva2:707793/FULLTEXT02. [Accessed: 10-Jun-2017]," 2012.

[89] ProLabs, "100G QSFP28 Optical Transceiver". [Online]. Available: http://www.prolabs.com/products/datasheets/msa_standard/QSFP28-100G-SR4-NC.pdf. [Accessed: 04- Jan- 2017]."

[90] Mellanox, "Power Saving Features in Mellanox Products". [Online]. Available: http://www.mellanox.com/pdf/whitepapers/WP_ECONET.pdf. [Accessed: 04- Jan- 2017]."

[91] Enablence, "100GHz WAVELENGTH DIVISION MULTIPLEXER/DEMULTIPLEXER". [Online]. Available: http://www.enablence.com/media/mediamanager/pdf/18-enablence-datasheet-ocsd-awg-standard-100ghzmultidemulti.pdf. Accessed: 04- Jan-2017]."

[92] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," 2006.

[93] J. Niemann, "Best practices for designing data centers with the infrastruxure inrow RC," *Application note of American Power Conversion*, 2006.

[94] H. M. M. Ali, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient disaggregated servers for future data centers," in *Networks and Optical Communications, 2015. NOC 20th European Conference on*, 2015, pp. 1-6.

[95] H. M. Ali, A. Lawey, T. E. Elgorashi, and J. Elmirghani, "Energy Efficient Resource Provisioning in Disaggregated Data Centres," in *Asia Communications and Photonics Conference*, 2015, p. AM1H. 1.

[96] ARISTA, "ARISTA 7500 Data Center Switch". [Online]. Available: https://www.arista.com/assets/data/pdf/Datasheets/7500_Datasheet.pdf. [Accessed: 04- Jan- 2017]."

[97] GreenTouch, "GreenTouch Final Results from Green Meter Research Study Reducing the Net Energy Consumption in Communications Networks by up to 98% by 2020," *A GreenTouch White Paper*, vol. Version 1, 15 August 2015.

[98] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: a survey," in *Approximation algorithms for NP-hard problems*, 1996, pp. 46-93.

[99] H. M. M. Ali, A. M. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient resource provisioning with VM migration heuristic for Disaggregated Server design," *18th International Conference on Transparent Optical Networks (ICTON),* pp. 1-5, 2016.

[100] K. Li, H. Zheng, and J. Wu, "Migration-based virtual machine placement in cloud systems," in *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*, 2013, pp. 83-90.

[101] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (IAAS)," *International Journal of engineering and information Technology,* vol. 2, pp. 60-63, 2010.

[102] A. w. services, "How AWS Pricing Works," *white paper*, 2015.

[103] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp. 202-208.

[104] J. F. C. Kingman, *Poisson processes*: Wiley Online Library, 1993.