On the Benefits of Resource Disaggregation for Virtual Data Centre Provisioning in Optical Data Centres

Albert Pagès(1), Rubén Serrano(1), Jordi Perelló(1), and Salvatore Spadaro(1)

(1) Advanced Broadband Communications Center (CCABA), Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona Spain, e-mail: albertpages@tsc.upc.edu

Abstract: Virtual Data Centre (VDC) allocation requires the provisioning of both computing and network resources. Their joint provisioning allows for an optimal utilization of the physical Data Centre (DC) infrastructure resources. However, traditional DCs can suffer from computing resource underutilization due to the rigid capacity configurations of the server units, resulting in high computing resource fragmentation across the DC servers. To overcome these limitations, the disaggregated DC paradigm has been recently introduced. Thanks to resource disaggregation, it is possible to allocate the exact amount of resources needed to provision a VDC instance. In this paper, we focus on the static planning of a shared optically interconnected disaggregated DC infrastructure to support a known set of VDC instances to be deployed on top. To this end, we provide optimal and sub-optimal techniques to determine the necessary capacity (both in terms of computing and network resources) required to support the expected set of VDC demands. Next, we quantitatively evaluate the benefits yielded by the disaggregated DC paradigm in front of traditional DC architectures, considering various VDC profiles and Data Centre Network (DCN) topologies.

Keywords: Data centres; Resource disaggregation; Virtualization; Optimization; Optical networks.

1. INTRODUCTION

Data Centre (DC) infrastructures are a key element in nowadays' telecom and cloud infrastructures, allowing the access to enormous quantities of information anytime and anywhere. Thanks to the collaborative efforts of the thousands of servers hosted inside their premises, complex Internet and cloud services (e.g., search engines, cloud storage, etc.) can be realized. In traditional DCs, servers are arranged in racks, each one equipped with a Top of the Rack (ToR) switch that interconnects the several servers inside, and allows for the exchange of information between different racks across an intra-DC Network (DCN) fabric. Current DCN architectures are usually build upon commodity electrical switches (e.g., Ethernet), arranged in a multi-layer architecture, which provides several aggregation points and means of redundancy for enhanced utilization of the network resources and Quality of Service (QoS) guarantees [1]. Besides, racks on the DC are usually grouped in different regions, named clusters, to allow for a better scalability and management of the whole DC infrastructure.

However, the constant growth of the Internet traffic and cloud services fostered by bandwidth-hungry applications/paradigms such as Big Data, Internet of Things (IoT) and Video on Demand (VoD), pleads for bigger DC infrastructures in terms of both computing and network capacities, in order to accommodate all applications and workflows. For instance, it is forecast that the global IP traffic managed by DCs will almost double by the year 2019, rising from 5.6 ZB to 10.4 ZB per year, with around 75% of the traffic staying inside their premises [2]. This unprecedented traffic growth is pushing the capabilities of current electrical-based DCN fabrics beyond their limits. For this reason, special attention at improving the performance of intra-DCNs is being put in the development of future DC architectures. In this regard, optical technologies have gained considerable interest due to their superior scalability, bandwidth and latency, as well as reduced power consumption. Hence, lots of efforts are being devoted to integrate them in future DCNs [3], either based on hybrid electrical/optical (e.g., as in [4]) or all-optical (e.g., see [5], [6]) network fabrics for the communication of servers inside the DC.

Despite such efforts on improving the performance of DCNs, current server-centric DCs still face some limitations toward efficient computing resource utilization. In general, services/tasks in DCs are executed on top of Virtual Machines (VMs) that are deployed at servers. Each VM is provisioned with a set of computing resources (i.e., CPU cores, storage and memory) tailored to the computational needs of the applications. These resources are then allocated and dedicated to VMs during their whole lifecycle. A coexistence of VMs inside the same server is possible if the total amount of resources requested by all of them does not exceed the server's total resource capacity. However, the heterogeneous VM computing resource demands can lead to server underutilization. For instance, it may happen that an application/service (i.e., a VM) running on a server employs almost the totality of one resource type (e.g., CPU cores), while imposing almost no requirements to the others (e.g., storage, memory). As a result, it may be impossible to allocate another application in the same server due to the scarcity of that resource type, letting the remainder underutilized. As an example of this phenomenon, Google has recently published data regarding the utilization of their DC infrastructures, disclosing high disparity of storage/memory to CPU usage for their tasks [7]. Furthermore, it becomes even more difficult to dynamically configure the DC resources under an unpredictable traffic profile.

Aside from poor resource utilization, server-centric architectures also suffer from a limited modularity that impacts on the system-wide performance. Traditional servers are usually built by tightly integrating their components (CPU, memory modules, disk, network interface card, etc…) into a single motherboard. This has been the basis of computer manufacturing for many years. However, this tight integration is responsible for the limited improvement possibilities of the overall system performance. This mainly happens because the rate at which the several components scale (in size, speed, etc.) is substantially different. For instance, the rate per year at which CPU performance has increased has been about 60%, while the rate of improvement in DRAM memory performance has merely been around 7% per year. This fact leads to a performance gap between CPU and memory of about 50% per year [8]. Such a disparity on the evolution of the different kinds of server components prevents utilizing the most advanced technology in some cases, since compromise decisions have to be taken in favour of a good system performance.

To overcome these challenges, new DC architectures have to be designed. An interesting approach to this end is the resource disaggregation concept [9], which proposes to disaggregate the computing resource components by physically decoupling and mounting them in separated blades, instead of tightly coupling them in a single integrated system. By physically decoupling the components, it is possible to adopt state-of-the-art technologies for each one of them, thus allowing for system optimization and customization. Such a concept has resulted in the disaggregated DC paradigm [10]-[13], where computing resources are no longer hosted in server units, but spread over standalone hardware blades. Resource blades can be grouped in racks hosting all types of computing resources (see Figure 1, right), or in mono-hardware racks where only a single type of resource is held. Then, resource blades are interconnected through the intra-DCN fabric. To meet the strict latency and bandwidth requirements for communicating the different hardware modules, intra-DCN optical technologies are envisioned [10]-[12].

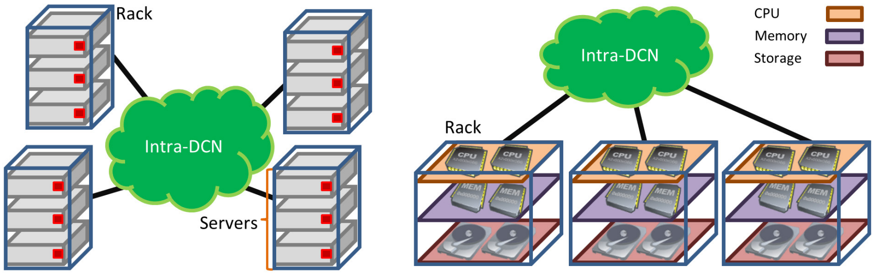

Figure 1. Server-centric (left) vs. disaggregated (right) DC architecture.

Through disaggregation, computing resources can be tightly assigned to VMs according to their needs, requiring fewer resources to satisfy a demand set while reducing the associated CAPEX. Moreover, disaggregation brings modularity to systems, enabling easier hardware upgrades when desired. For these reasons, optically interconnected disaggregated DCs are seen as a solution for future DCs.

Despite the benefits that resource disaggregation paradigm promises to bring, there is little work in the literature on analysing the enhanced computing resource utilization of disaggregated DCs in front of nowadays' server-centric ones. In view of this, in this work we quantify the required computing resources to be equipped at DC infrastructures following the resource disaggregation paradigm when allocating service requests, and compare it to legacy server-centric DC ones. To this end, we focus on a static capacity planning of an optically interconnected disaggregated DC, aiming to give insight into the reduction of computing resource requirements that such paradigm can yield.

Given that Virtual Data Centre (VDC) has been identified as a key service that modern DCs have to offer to be able to efficiently implement multi-tenancy in a cloud environment, we will focus our efforts on analysing the planning of both disaggregated and server-centric DC infrastructures when supporting VDC services. To this goal, the remainder of the paper is structured as follows: section 2 introduces the concept of VDC service and the issues involved. Next, section 3 reviews the related work in the literature regarding disaggregated DCs and VDC provisioning, highlighting the contributions of this work. Section 4 elaborates on the considered DC planning scenario, presenting the optimization problem under consideration, while section 5 details the different solutions proposed to tackle it, both for server-centric and disaggregated DC architectures. Next, section 6 numerically evaluates and compares their computing resource requirements. Finally, section 7 draws up the main conclusions of the work.

2. VIRTUAL DATA CENTRE PROVISIONING

Contemporary DC infrastructures must allocate a plethora of customers, ranging from business/companies or public institutions to individual users, having all of them heterogeneous needs in terms of resource necessities, QoS, degree of control over the employed resources, and so on. In this regard, multi-tenancy becomes a pillar requirement that modern DC infrastructures must provide. However, traditional telecom and cloud infrastructure architectures present some drawbacks compromising the efficient implementation of multi-tenancy, even more in cloud environments, where a high degree of customization and dynamicity is present. Besides, scalability and resource provisioning specific challenges must be solved [14]. Indeed, the service structure is very rigid, with infrastructure owners focusing on the services offered on top of their infrastructures, with limited adaptability to the service to be deployed. This has led to architectures incapable to adapt to dynamic traffic patterns, with high heterogeneity on the characteristics of the services/applications to be deployed.

To overcome these limitations, the concept of Infrastructure as a Service (IaaS) has been introduced [15]. The emergence of IaaS arises from the need to provide telecom and cloud infrastructure owners with means to better exploit and manage their infrastructures, in an environment experiencing frequent changes on user needs and service requirements. IaaS allows offering a portion of the physical infrastructure as a service for exploitation by third party entities, giving them even the possibility to control and manage it as if they were owners of the infrastructure. The key enabling technology behind IaaS is virtualization, which allows abstracting and slicing physical devices into multiple virtual elements. Then, such virtual elements can be composed in virtual infrastructures tailored to the requirements of the renting entities, in terms of resources (network, storage, computation, etc.), management and control.

As a way to implement IaaS, the VDC service has been introduced [16]. It consists of leasing part of a DC operator physical infrastructure to external entities (hereafter referred to as tenants) as a support to develop their own business models. Each VDC is a virtual infrastructure integrating computing capabilities in the form, for example, of VMs, interconnected through a virtual network fabric (i.e., through virtual links) providing the necessary bandwidth between them. These virtual infrastructures are then employed by tenants to deploy applications on top that they can offer as services to end users. Thanks to such VDCs, the coexistence of multiple tenants on top of the same physical infrastructure is achieved, each tenant being completely isolated from the activity of the remainder. At the same time, tenants can also benefit from lower management and maintenance costs, since they do not need to invest on nor further operate their own DC infrastructures.

One of the key problems that the VDC service faces is the *VDC mapping (or embedding) problem*, that is, how the requested virtual resources by a VDC are provisioned over the underlying physical ones. This problem involves two different phases: 1) VM mapping onto physical servers; 2) virtual link mapping onto physical network resources interconnecting these physical servers. Figure 2 depicts an example of such an operation. Ideally, both phases should be tackled jointly or in a coordinated way, so as to enhance the utilization of the physical resources, thus increasing the number of VDC instances that can be deployed on top of the underlying shared physical infrastructure [17], [18]. However, as mentioned before, traditional server-centric DC architectures may lead to poor computing resource utilization when allocating VMs. As a consequence, a DC operator may be forced to overprovision server resources to serve certain VDC instances, increasing the associated CAPEX.

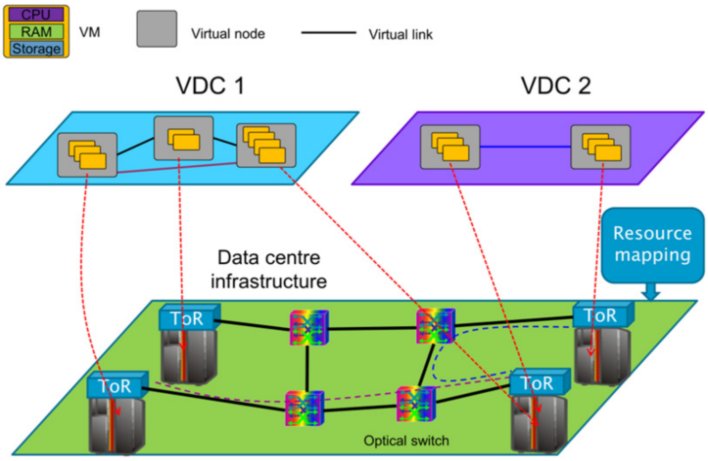

Figure 2. Example of VDC mapping process.

In contrast, virtual servers tailored to the specific needs of the VMs can be built from the pool of available hardware modules at the rack blades in a disaggregated DC. Such a feature is especially beneficial when provisioning complex cloud services, such as VDC instances where an optimal utilization of the computing resources becomes paramount. Thanks to the better resource assignment that disaggregated DCs allow when allocating VMs, a higher number of VDC instances can be deployed on top of a shared physical infrastructure [19]. Additionally, the total amount of computing resources to serve the same VDC request set can be lowered when compared to legacy server-centric architectures. For this reason, in this work we focus on the planning optimization problem of VDC instances on top of a disaggregated DC infrastructure and compare its performance against that of a legacy server-centric DC. Before proceeding to present the problem and scenario under consideration, as well as the proposed mechanisms to address it, next section reviews relevant work in the literature regarding resource disaggregation in DCs and VDC provisioning in optically interconnected DC infrastructures.

3. RELATED WORK

One of the first attempts to introduce resource disaggregation inside DCs came from a Facebook internal initiative, which aimed at improving the networking capacities of their DC infrastructures by distributing the switches functionalities across the whole network, while providing a novel operating system for their control, named Facebook Open Switching System (FBOSS) [20]. As a next step toward its vision of the future DC architecture, Facebook devoted efforts to improve the performance of its server units by stripping out unneeded components in an x86 server, resulting in significant CAPEX and OPEX savings. This spawned the Open Compute Project (OCP) initiative [9], whose goal is to investigate and provide the architecture of future DCs leveraging on the resource disaggregation concept. Among other features, OCP has resulted in a range of initiatives in the design of servers, networking, storage and even the racks for holding it all. This effort is seen as a precursor to the disaggregated DC, but still relying on traditional server designs. The OCP is providing some insights to address the drivers toward disaggregation, with particular focus on solving the massive-scale challenge. Other industrial initiatives focusing on exploiting the concept of disaggregated DC are the Rack Scale Architecture

(RSA) from Intel [13], which aims to disaggregate compute, network and storage across a DC rack, and the High Throughput Computing Data Center (HTC-DC) Architecture from Huawei [21], which, among other features, focuses on a disaggregated DC architecture where blades are interconnected through a high bandwidth optical network fabric.

In all of these initiatives, one of the main identified challenges relates to the realization of the network fabric, which should be flexible enough to allow for any kind of interconnection between blades in order to achieve multiple hardware configurations. Moreover, a critical parameter is the bandwidth needed on the network fabric. For example, typical memory bandwidth stays in the several tens of Gb/s, hence a high performance network fabric is necessary to achieve such high bandwidth communications. This is the main reason motivating optical technologies as prime candidates for the realization of the network fabric in disaggregated DC infrastructures.

The work of authors in [11] was one of the first ones to analyse the problematic of the network connectivity in disaggregated DCs, focusing on the latency and bandwidth requirements of applications deployed on top of the physical infrastructure, and presented optical technologies as the mean to overcome these challenges. In this regard, the authors elaborated on the specific research directions and steps needed to fulfil the connectivity requirements of applications if resource disaggregation comes to fruition, going from simple hardware and device notions to network-wide architecture aspects, finally analysing system-wide implications and requirements. Focusing on the latency and I/O restrictions imposed by applications running on a disaggregated DC, authors in [12] discussed about hardware needs for pure optical data transmission in such DC architectures, and presented Photonic Integrated Circuits (PICs) as the enabling technology to realize an efficient data transmission with low latency. The authors also reviewed different technologies for PICs packaging, analysing both the incurred latencies and the insertion losses. Authors concluded that optical links leveraging on PICs can reduce the overall system complexity and provide communication latencies in the order of very few nanoseconds, satisfying the requirements that a disaggregated DC would impose.

From a system-wide perspective, authors in [22] demonstrated an all-optical architecture for distributed CPU, memory and storage in a disaggregated DC infrastructure, based on Spectrum Selective Switches (SSS) and Wavelength Division Multiplexing (WDM)/Time Division Multiplexing (TDM) granularities, so as to flexibly adapt the transmission capabilities to the interconnection needs between hardware modules. The authors demonstrated intra- and inter-rack transmission latencies in the nanoseconds range, with few microseconds for the inter-cluster communications. As an evolution of this work, the authors presented an updated architecture in [23], where they introduced the presence of hollow-core optical fibres for reduced latency and increased bandwidth for intra- and inter-blade communications. The authors expanded this work in [10], where they analysed the proposed architecture in terms of achieved throughput, latency and Bit Error Rate (BER), and experimentally assessed the on-demand creation of optical paths between hardware blades to accommodate different traffic patterns in a disaggregated DC architecture.

However, all of these works, although providing full designs for optically interconnected disaggregated DC infrastructures and validating them, they do not quantify the benefits that the resource disaggregation paradigm can bring against current DC infrastructures based on traditional server units. For this reason, in our work we will try to answer this question, analysing the planning of disaggregated DCs for supporting VDC services, under heterogeneous requests configurations and network topologies.

Regarding the virtual service provisioning, either computing, networking or both in DC architectures with optical DCN, a plethora of works have analysed the problems involved and presented multiple solutions to solve them. For instance, authors in [24] experimentally assessed the provisioning of VDC instances in Software Defined Networking (SDN)-based Optical Packet Switching (OPS) DCN architectures. The authors proposed a novel control and data plane architecture that allowed the automatic provisioning of VDC slices on top of it aiming to improve the QoS of the overall system. Authors in [25] tackled the energy consumption aspect when provisioning virtual networks in optical DCs. Energy consumption is becoming a very serious challenge in nowadays' DCs and a lot of research efforts are devoted to provide solutions to reduce it. To this goal, the authors proposed an embedding mechanism that minimizes the number of electronic ports used when deploying a virtual network in a hybrid electrical/optical DC infrastructure. Alternatively, authors in [26] studied the added benefits of optical technologies for VM migration operations inside DCs. VM migration is capital in DC infrastructures since it allows for a good resource utilization as well as enhanced protection against failures in dynamic DC environments. In this regard, optical technologies improve the migration time thanks to their superior bandwidth. Nevertheless, all of these works focused on the provisioning of service instances in server-centric architectures, where VMs are allocated in traditional server units. None of them, however, analysed the implications and challenges that the rising resource disaggregation paradigm brings when allocating service instances nor quantified the expected benefits of disaggregated DCs both in terms of better resource utilization and increased service acceptance.

As a previous work, in [19] we analysed the benefits of optically interconnected disaggregated DCs in terms of increased service acceptance when compared to server-centric DC infrastructures. For this purpose, we proposed exact and heuristic mechanisms aiming at maximizing the number of service requests mapped over a shared DC infrastructure. Our analysis revealed that disaggregated DC infrastructures allow 50% more service instances to be deployed on top of the same physical infrastructure compared to legacy server-centric DC architectures. Such results highlighted the expected benefits of the resource disaggregation paradigm. As a follow up, in the present paper we plan to study the resource reduction that disaggregated DCs can allow when supporting a set of service requests, such as VDC instances. This analysis will complement the previous one, providing more insight into the potential adoption of the disaggregated paradigm by DC operators. Next section presents the scenario under consideration.

4. DATA CENTRE PLANNING FOR VDC SERVICE PROVISIONING

4.1. Scenario description

Particularly, we assume an optically interconnected disaggregated DC scenario in which the network topology interconnecting the different hardware blades and racks is already given. An example of the assumed DC architecture is depicted in Figure 3.

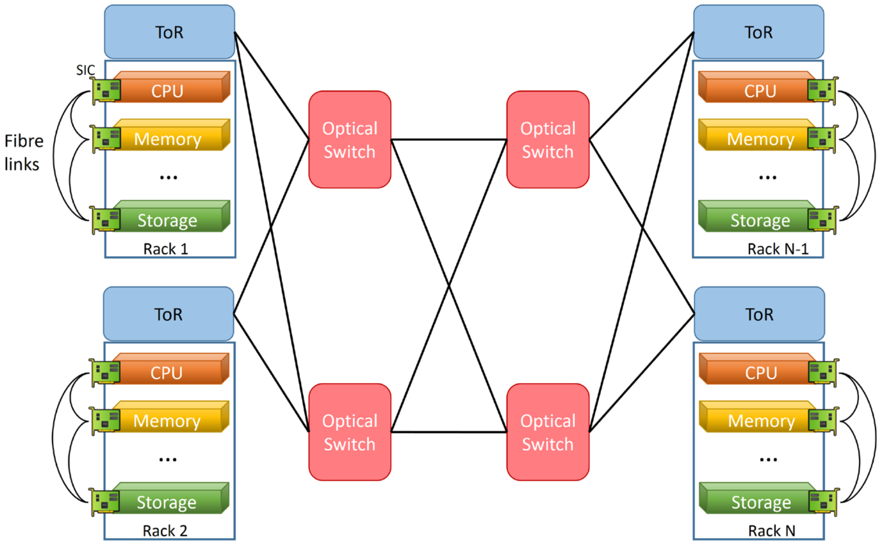

Figure 3. Disaggregated DC architecture assumed in this work.

Specifically, we assume a disaggregated DC infrastructure where computing resources are organized in hardware blades contained in racks. We assume the case where all racks hold blades of all types of computing resources, i.e., CPU cores, storage and memory. Each of the hardware blades has a Switch and Interface Card (SIC) that performs an electrical-to-optical and optical-to-electrical conversion of the signals coming from/going to the hardware modules of the blade, respectively. Moreover, a set of fibre links is employed to interconnect the different hardware blades inside a rack and between racks. Specifically, we assume that dedicated fibre links are set between hardware blades belonging to the same rack, so as to mitigate incurred latencies and bandwidth limitations when communicating hardware modules within the same rack. As for the inter-rack communication, a WDM-based optical network is employed to interconnect the different racks. Each of the racks is equipped with an opto-electronic ToR switch, which interconnects the hardware blades of a rack with other racks across the circuit-switched WDM inter-rack network fabric. The communication between racks is done all-optically through a set of optical switches, assumed to be Colourless, Directionless and Contentionless (CDC). As a result, high bandwidth and ultra-low latency communications can be established among hardware blades along the DC infrastructure [10].

As explained before, a VDC request consists in a set of VMs, each one requesting an amount of computing resources. These VMs are then interconnected by virtual links of the desired bandwidth between VMs. In this work, we assume that VMs are mapped onto physical resources of the same rack, since mapping them over distant racks could lead to unacceptable latencies between its computing resource modules. Moreover, we consider the case where VMs belonging to the same tenant (i.e., VDC) are mapped onto different racks, so as to provide enhanced resilience upon rack failures. Note, however, that VMs belonging to different tenants can be mapped onto resources of the same rack. Regarding the virtual links, without loss of generality, we consider that each one of them requests for a certain bandwidth in terms of wavelength channels. Thus, the virtual link mapping requires solving a Routing and Wavelength Assignment (RWA) problem in a transparent optical network (i.e., the inter-rack network fabric), which implies determining the necessary lightpaths that satisfy the connectivity requirements sated by the virtual links.

Given the direct by-pass links between hardware blades inside the same rack, we assume that there are no latency or bandwidth limitations for the intra-rack communication and only focus on determining the network capacity for the inter-rack segment. Note that such assumption is possible since we are considering that racks keep hardware blades of all types of resources and VMs are mapped over resources of the same rack. In such a situation, the intra-rack communications for a particular VDC instance are only due to the communications between the different hardware modules constituting a single VM. Given that dedicated fibre links are employed to interconnect the different hardware blades, high speed and low latency communications are achieved inside the components of a single VM, thus effectively working as a single composed element, avoiding any kind of bottleneck for intra-rack communications, as evidenced in [10], [22].

Therefore, the virtual link mapping translates into finding the lightpaths between the source-destination ToRs of the racks onto which the endpoints (the VMs) of the virtual link are mapped. This entails finding an end-to-end path between the source and destination, as well as a continuous wavelength channel in all fibre links constituting the path (we remind the reader that an all-optical inter-rack DCN is assumed). Additionally, lightpaths cannot be assigned the same wavelength channel onto the same physical link (wavelength clashing constraint). Finally, optical resources (i.e., the established lightpaths) cannot be shared among different VDCs to keep the isolation between tenants. Different lightpaths can coexist in the same physical links, though, thanks to the multiplexing capabilities of WDM-based optical networks.

4.2. Problem statement

With the aforementioned conditions and scenario, this section states the optimization problem that we are targeting.

Objective: minimize the necessary amount of computing resources (CPU cores, storage, memory) and the number of different wavelength channels per link in the optical DCN to be equipped at the physical DC infrastructure.

Given:

- A transparent WDM-based inter-rack optical DCN represented by the graph $G_n = (N_f, E_f)$, with N_f the set of physical nodes (racks/ToRs and optical switches) and E_f the set of physical fibre links. Additionally, we denote as $N_{OS} \subseteq N_f$ the set of optical switches, with each optical switch having a port count equal to N_{port}.
- An ordered set of wavelength channels per physical link, denoted as W, of enough size to support all virtual links of all VDC requests. Thus, physical links of infinite capacity are initially assumed in practice. The final capacity of the physical links, which may be different for every physical link, will be determined by the optimization procedure.
- A set of racks denoted as R. Each rack r is holding an aggregated amount of CPU cores, storage and memory equal to CPU_r, $HDD_r$and RAM_r (disaggregated case) or a set of servers defined as S_r, each one of them equipped with an amount of CPU cores, storage and memory equal to CPU_s, HDD_s and RAM_s (server-centric case). Note that the resources per rack in the disaggregated scenario or the number of servers in the server-centric scenario are assumed to be enough to allow for the allocation of all VDC

requests. The optimization procedure will determine the minimum necessary to be equipped at the DC infrastructure.

- A set of VDC requests to be served, denoted as D. Each VDC inside the set is characterized by a virtual graph $G_d = (N_v^d, E_v^d)$, being N_v^d the set of VMs and E_v^d the set of virtual links. Each VM $n_v \in N_v^d$ requests for a set of CPU cores, storage and memory equal to CPU_{n_v}, $HDD_{n_v}$and RAM_{n_v}while each virtual link $e_v \in E_v^d$ requests a capacity in wavelengths equal to B_{e_v}.

Find: the mapping of all virtual nodes and links of all VDC requests in D.

Subject to:

1) All VDC requests have to be mapped (no request blocking is considered).
2) Optical resources assigned to a VDC cannot be shared with other VDCs. Additionally, two or more lightpaths employing the same physical link cannot be mapped to the same wavelength channel. Multiple lightpaths can coexist in the same physical link, but employing different wavelength channels thanks to the multiplexing capabilities of WDM-based networks.
3) The wavelength continuity constraint must be ensured along the path onto which a virtual link is mapped (a transparent DCN is considered).
4) A VM can only be mapped to resources of a single rack (disaggregated scenario) or a single server (server-centric scenario).
5) A physical rack/server can only host one VM of a certain VDC request. Nevertheless, racks/servers can host multiple VMs belonging to different VDC requests.
6) The aggregated capacity of the computing resources requested by VMs mapped in a particular rack/server cannot surpass its initial capacity.
7) The number of employed servers per rack cannot surpass the initial server capacity per rack (server-centric case).
8) The total number of active incoming/outgoing wavelengths from/to an optical switch must not surpass its port count.

After stating the targeted optimization problem, we proceed on introducing exact Integer Linear Programming (ILP) formulations to address the problem for both disaggregated and server-centric DC scenarios. Additionally, we also present lightweight heuristic mechanisms for scenarios in which the scalability of the ILP formulations could be compromised. Next section provides the details of the proposed mechanisms.

5. PROPOSED MECHANISMS

5.1. Notation definition

Before going into the details of the proposed mechanisms, Table I summarizes some extra notation and definitions employed through this section. The definition of set P allows us to easily ensure the wavelength continuity constraint across the lightpaths onto which the virtual links are mapped, as wavelength resources are reserved explicitly along end-to-end paths (i.e., they remain the same on all physical links forming the selected path). As for sets $Q_{\delta^+(n_f)}$ and $Q_{\delta^-(n_f)}$, their definition allows us to determine the number of virtual links, hence, optical connections, that are going/coming to/from an optical switch in the DCN, thus, correctly bounding their total value to the port count of the switches defined by N_{port}.

TABLE I

NOTATION DEFINITION

Symbol	Description
P	set of physical paths between ToRs in G_n
P_{e_f}	set of physical paths in P that traverse physical link $e_f \in E_f$
$a(\cdot)$	source endpoint of a virtual link or physical path
$b(\cdot)$	destination endpoint of a virtual link or physical path
$\delta^+(n_f)$	set of outgoing links from physical node $n_f \in N_f$
$\delta^-(n_f)$	set of incoming links to physical node $n_f \in N_f$
$Q_{\delta^+(n_f)}$	set of physical paths in P that traverse an outgoing link of physical node $n_f \in N_f$
$Q_{\delta^-(n_f)}$	set of physical paths in P that traverse an incoming link of physical node $n_f \in N_f$
CPU_s^a	currently available number of CPU cores in server s
RAM_s^a	currently available memory in server s
HDD_s^a	currently available storage capacity in server s

Once all these definitions have been introduced, we proceed with the description of the proposed mechanisms.

5.2. Optimal ILP formulations

5.2.1. Server-centric scenario

To start, we consider a legacy server-centric architecture for benchmark purposes, where computing resources are organized in servers. In such a case, the objective of the optimization problem is still the same as presented in the problem statement: to determine the minimum amount of both computing and network resources to fully allocate the set of VDC instances. Nevertheless, as computing resources are tied to specific servers, the minimization of such resources cannot be addressed without accounting for the server dimension. For this reason, we determine in this case the minimum number of server units required to satisfy the mapping of the VDCs. Note that, due to the rigid computing resource capacities of the servers, it may happen that some of the servers' resources end up underutilized, needing additional server units to allocate all of the VMs and, thus, ending up with a total number of computing resources that is larger than the strictly necessary. We will analyse such differences during Section 6. Aside from this, we consider the same network scenario, where servers are interconnected to a ToR that interconnects all of the servers inside the rack and interconnects different ToRs thanks to an all-optical intra-DCN.

Note that the optimization problem presented has a multi-objective optimization function: on the one hand, the minimization of the computing resources is required; on the other hand, the minimization of the different wavelengths to be equipped per physical link is pursued. To achieve the first objective, a server consolidation approach is required, that is, to fit VMs in servers in a way that the resources of the server are fully occupied, thus minimizing the number of employed servers. This is a form of the bin-packing problem, with servers being multi-dimensional containers and VMs the items to be placed in such containers. As for the second objective, network balancing and route diversity should be encouraged, which allows to re-assign the same wavelength channel to different lightpaths over different parts of the network, thus effectively minimizing the number of different employed wavelengths per link. As can be observed, the two objectives compete with each other, since server consolidation tends to saturate specific racks, concentrating virtual links (i.e., lightpaths) in some physical links, increasing the number of different wavelengths to be equipped at the DCN (we remind the reader that different lightpaths have to be assigned different wavelength channels for isolation purposes). Conversely, in order to better distribute the network load, rack diversity should we encouraged, thus limiting the possibility of server consolidation. Given this situation, the common approach is to define the objective function as the weighted sum of the multiple sub-objectives, which allows a fine tuning of the objectives and optimization goal depending on the weights employed.

With such a discussion, we present an ILP formulation, named ILP-Server-Centric DC Resource Planning (ILP-SCDCRP), which optimizes the presented multi-objective function in a server-centric DC architecture. Table II summarizes the employed problem variables.

TABLE II
ILP-SCDCRP VARIABLE DEFINITION

Variable	Definition
$x_{d,n_v,r,s}$	Binary; 1 if VM n_v of request d is mapped on server s of rack r, 0 otherwise
$y_{d,e_v,p,w}$	Binary; 1 if virtual link e_v of request d is mapped on wavelength w over physical path p, 0 otherwise
z_w	Binary; 1 if wavelength w is being utilized in any link of the optical DCN, 0 otherwise
$s_{r,s}$	Binary; 1 if server s in rack r is being utilized, 0 otherwise

With such introduced variables, the exact details of the ILP formulation are shown below:

$$\textit{minimize} \quad \alpha\left(\sum_{w\in W} z_w + \varepsilon \sum_{d\in D}\sum_{e_v\in E_v^d}\sum_{p\in P}\sum_{w\in W} h_p y_{d,e_v,p,w}\right) + (1-\alpha)\sum_{r\in R}\sum_{s\in S_r} s_{r,s} \tag{1}$$

subject to:

$$\sum_{r\in R}\sum_{s\in S_r} x_{d,n_v,r,s} = 1 \qquad \forall\, d\in D, n_v\in N_v^d \tag{2}$$

$$\sum_{n_v\in N_v^d}\sum_{s\in S_r} x_{d,n_v,r,s} \le 1 \qquad \forall\, d\in D, r\in R \tag{3}$$

$$\sum_{p\in P}\sum_{w\in W} y_{d,e_v,p,w} = B_{e_v} \qquad \forall\, d\in D, e_v\in E_v^d \tag{4}$$

$$y_{d,e_v,p,w} \le \sum_{s\in S_{a(p)}} x_{d,a(e_v),a(p),s} \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \tag{5.a}$$

$$y_{d,e_v,p,w} \le \sum_{s\in S_{b(p)}} x_{d,b(e_v),b(p),s} \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \tag{5.b}$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} CPU_{n_v} x_{d,n_v,r,s} \le CPU_s \qquad \forall\, r\in R, s\in S_r \tag{6.a}$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} RAM_{n_v} x_{d,n_v,r,s} \le RAM_s \qquad \forall\, r\in R, s\in S_r \tag{6.b}$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} HDD_{n_v} x_{d,n_v,r,s} \le HDD_s \qquad \forall\, r\in R, s\in S_r \tag{6.c}$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in P_{e_f}} y_{d,e_v,p,w} \le 1 \qquad \forall\, e_f\in E_f, w\in W \tag{7}$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in Q_{\delta^+(n_f)}}\sum_{w\in W} y_{d,e_v,p,w} \le N_{port} \qquad \forall\, n_f\in N_{OS} \tag{8.a}$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in Q_{\delta^-(n_f)}}\sum_{w\in W} y_{d,e_v,p,w} \le N_{port} \qquad \forall\, n_f\in N_{OS} \tag{8.b}$$

$$y_{d,e_v,p,w} \le z_w \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \tag{9}$$

$$x_{d,n_v,r,s} \le s_{r,s} \qquad \forall\, d\in D, n_v\in N_v^d, r\in R, s\in S_r \tag{10}$$

The optimization goal presented in Eq. (1) is twofold: firstly, it minimizes the number of different wavelengths to be equipped in the network links of the DCN (first term); moreover, it minimizes the total number of necessary servers to support the VMs of the VDC instances (third term). The parameter α is a positive real number in the range [0, 1] that is used to put more emphasis on one of the two sub-objectives depending on the current allocation policy. Additionally, a secondary term is added into the first objective (second term in the summation), with parameter ε being a very small positive number (i.e., $\varepsilon \ll 1$). This term prioritizes solutions that lead to the minimum total number of employed wavelength channels in the network in the case that two or more solutions evaluate to the same number of different wavelengths to be equipped per physical link. As for the constraints, constraint (2) ensures that all VMs of a demand (i.e., a VDC instance) are mapped into physical servers, forcing a particular VM to be mapped to a unique server. Constraint (3) guarantees that the servers of a single rack host at most a VM per VDC instance, that is, the VMs of a VDC are mapped over servers belonging to different racks. This is done in order to encourage some degree of resilience against server and rack failures. Nevertheless, note that VMs of different VDC instances can still be mapped onto the same physical rack or server. Constraint (4) is the link mapping constraint, ensuring that every virtual link is provided with the requested number of lightpaths, while constraints (5) restrict the mapping of the virtual links over physical paths interconnecting the racks into which the remote endpoints of the virtual links have been mapped. Next, constraints (6) ensure that the capacity of a single server unit is not surpassed, that is, the summation of the resources of all VMs mapped onto the server does not exceed the available resources at the server for all types of resources (CPU cores, memory and storage). Constraint (7) is the wavelength clashing constraint, which prevents that two lightpaths share a wavelength channel in the same physical link. Constraints (8) account for the switching limitations at the optical switches, restricting that the number of active incoming/outgoing wavelength channels is lower than their port limit. Finally, constraints (9) and (10) serve to discriminate if a wavelength or a server is being utilized in the DC infrastructure, respectively.

5.2.2. Disaggregated scenario

Similarly to the server-centric scenario, we now present the ILP formulation, named ILP-Disaggregated DCRP (ILP-DDCRP), to optimally decide the VDC mapping on top of a disaggregated DC architecture. The objective of the VDC mapping remains identical: to minimize the number of total computing resources at the racks, as well as the number of different wavelength channels to be equipped per physical link. In this regard, note that a disaggregated DC allows provisioning the exact amount of computing resources that a VM requests, hence, the minimization of the rack resources is implicit. For this reason, in the disaggregated DC scenario we will also pursue the goal to optimize the VM distribution along the different racks, either by minimizing the number of employed racks or the maximum rack load depending on the desired allocation policy. Such considerations are made to give the possibility to the DC operator to tailor the infrastructure planning according to its needs. Like in the server centric case, the resulting optimization goal joins different sub-objectives, one determining the minimum number of wavelengths per physical link and the other the optimal distribution of VMs along the racks. Additionally, the second sub-objective is also formed by two objectives: one that minimizes the number of employed racks and another the maximum rack load. For this, we also defined the objective function as a weighted sum of the different sub-objectives, with several weighting factors to adjust the optimization criteria. With these, Table III introduces the decision variables of the formulation.

TABLE III
ILP-DDCRP VARIABLE DEFINITION

Variable	Definition
$x_{d,n_v,r}$	Binary; 1 if VM n_v of request d is mapped on rack r; 0 otherwise
$y_{d,e_v,p,w}$	Binary; 1 if virtual link e_v of request d is mapped on wavelength w over physical path p; 0 otherwise
z_w	Binary; 1 if wavelength w is being utilized in any link of the optical DCN; 0 otherwise
u_r	Binary; 1 if server rack r is being utilized; 0 otherwise
u_{max}	Real; the maximum average resource utilization among all racks

Next, we proceed on detailing the proposed ILP formulation:

$$\textit{minimize}\ \alpha\left(\sum_{w\in W} z_w + \varepsilon\sum_{d\in D}\sum_{e_v\in E_v^d}\sum_{p\in P}\sum_{w\in W} h_p y_{d,e_v,p,w}\right) + (1-\alpha)(\beta\sum_{r\in R} u_r + (1-\beta)u_{max}) \quad (11)$$

subject to:

$$\sum_{r\in R} x_{d,n_v,r} = 1 \qquad \forall\, d\in D, n\in N_v^d \quad (12)$$

$$\sum_{n_v\in N_v^d} x_{d,n_v,r} \le 1 \qquad \forall\, d\in D, r\in R \quad (13)$$

$$\sum_{p\in P}\sum_{w\in W} y_{d,e_v,p,w} = B_{e_v} \qquad \forall\, d\in D, e_v\in E_v^d \quad (14)$$

$$y_{d,e_v,p,w} \le x_{d,a(e_v),a(p)} \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \quad (15.a)$$

$$y_{d,e_v,p,w} \le x_{d,b(e_v),b(p)} \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \quad (15.b)$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} CPU_{n_v} x_{d,n_v,r} \le CPU_r \qquad \forall\, r\in R \quad (16.a)$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} RAM_{n_v} x_{d,n_v,r} \le RAM_r \qquad \forall\, r\in R \quad (16.b)$$

$$\sum_{d\in D}\sum_{n_v\in N_v^d} HDD_{n_v} x_{d,n_v,r} \le HDD_r \qquad \forall\, r\in R \quad (16.c)$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in P_{e_f}} y_{d,e_v,p,w} \le 1 \qquad \forall\, e_f\in E_f, w\in W \quad (17)$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in Q_{\delta^+(n_f)}}\sum_{w\in W} y_{d,e_v,p,w} \le N_{port} \qquad \forall\, n_f\in N_{OS} \quad (18.a)$$

$$\sum_{d\in D}\sum_{e_v E_v^d}\sum_{p\in Q_{\delta^-(n_f)}}\sum_{w\in W} y_{d,e_v,p,w} \le N_{port} \qquad \forall\, n_f\in N_{OS} \quad (18.b)$$

$$y_{d,e_v,p,w} \le z_w \qquad \forall\, d\in D, e_v\in E_v^d, p\in P, w\in W \quad (19)$$

$$x_{d,n_v,r} \le u_r \qquad \forall\, d\in D, n_v\in N_v^d, r\in R \quad (20)$$

$$\frac{1}{3}\left(\frac{\sum_{d\in D}\sum_{n_v\in N_v^d} CPU_{n_v} x_{d,n_v,r}}{CPU_r} + \frac{\sum_{d\in D}\sum_{n_v\in N_v^d} RAM_{n_v} x_{d,n_v,r}}{RAM_r} + \frac{\sum_{d\in D}\sum_{n_v\in N_v^d} HDD_{n_v} x_{d,n_v,r}}{HDD_r}\right) \le u_{max} \qquad \forall\, r\in R \quad (21)$$

Eq. (11) presents a multi-objective optimization goal: firstly, it minimizes the number of different wavelengths to be equipped in the network links of the DCN (first term); moreover, it minimizes the number of employed racks (third term) and the load of the most loaded rack (fourth term). Parameters α and β are real-valued numbers in the range [0, 1] employed to adapt the optimization goal according to the current needs of the DC operator. In particular, parameter α has the same purpose as in the server-centric DC case, namely, to modulate the emphasis of the objective depending if only computing resources are minimized ($\alpha = 0$), or network resources ($\alpha = 1$), or any situation in between. As for parameter β, its purpose is to better control the distribution of the VMs along the computing resources (i.e., the racks). Since in a disaggregated DC scenario the minimization of the computing resources is implicitly made, as VMs require the allocation of the exact amount of resources they need, an alternative optimization criterion is to choose their distribution in the DC infrastructure. To this end, β parameter is introduced, which allows to minimize the maximum load per rack ($\beta = 0$), the number of employed racks ($\beta = 1$) or a compromise solution in between. Similarly as in the server-centric case, we also introduce a secondary term (second term in the summation) to prioritize solutions leading to the minimum total number of employed wavelength channels in the DCN. As for the constraints, constraints (12)-(19) have the same purpose as their homonyms in the server-centric formulation (Eq. (2)-(9)), accounting for the rack dimension instead for particular servers, while constraint (20) serves to discriminate if a rack is being employed or not. Finally, constraint (21) determines the maximum load of a rack in the DC.

5.3. Heuristic mechanisms

As will be shown in Section 6, the execution times of the presented ILP formulations substantially increase with the size of the problem instance to solve, rendering them unpractical for large scenarios. In fact, VDC embedding or mapping is a particular case of the wider family of Virtual Network Embedding (VNE) problems, which have been shown to be NP-hard [27]. Moreover, for the particular case of WDM-based optical networks, the VNE problem includes an RWA problem to solve the mapping of virtual links (a physical path and a wavelength channel have to be assigned to them), which has also been proven to be NP-hard [28]. For this reason, proven optimal solutions to the problem under study may not be found in reasonable amounts of time for several problem instances. To this end, we also propose two heuristic mechanisms, one for each DC architecture scenario, in order to find near optimal solutions in shorter times. In this regard, let us comment that in the disaggregated DC scenario, the developed ILP formulation can be solved a little bit faster than the one proposed for the server-centric one, as it does not have to account for the server dimension. This mainly happens because the solution space is smaller (a lower number of decision variables and constraints are involved), which tends to reduce the required time to find the optimal solution for a given problem instance. Nevertheless, due to the NP-hard nature of the problem, the vast majority of the problem instances still require a significant time to be solved, increasing with the size of the problem instance.

5.3.1. Server-centric scenario

In this section we present the developed heuristic for the server-centric DC scenario, named Adaptive Greedy Procedure (AGP)-SCDCRP. The details of the proposed solution are depicted in Pseudo-code 1. In essence, the heuristic is an adaptive greedy procedure that maps iteratively all VDC requests in the demand set. At the end, the solution containing the mapping of all VDCs is returned.

Pseudo-code 1: AGP-SCDCRP heuristic

```
Input: $G_n$, $D$, $K$
Output: $Sol$    //Solution

$P$ ← K-Shortest Paths (SPs) between all pairs of ToRs in $G_n$
Sort demands in $D$ in descending order according to their most restrictive VM in terms of server occupation
For each $d$ in $D$ do
```

```
4:  Sort VMs in $N_v^d$in descending order according to their most restrictive resources in terms of server occupation
5:  $R_d \leftarrow \emptyset$   //Racks employed to map demand $d$
6:  For each $n_v$in $N_v^d$ do
7:    mapped←false   //Boolean to indicate if the VM has been mapped
8:    For each $r$ in $R$ do
9:      minServer← ∞   //Index of the first unemployed server in the least loaded rack
10:     If not mapped and $r$ not in $R_d$ then
11:       $s$ ←Find server already in use with enough room to map $n_v$ minimizing $\Delta(n_v, s)$
12:       If found then
13:         Map $n_v$ in $s$
14:         Add $r$ to $R_d$
15:         mapped←true
16:       Else
17:         $s$ ← index of first unemployed server
18:         If $s < minServer$ then
19:           minServer← $s$
20:   If not mapped then
21:     Map $n_v$ in minServer
22:     Add $r$ of minServer to $R_d$
23:   Update server status
24: For each $e_v$in $E_v^d$ do
25:   For 1 to $B_{e_v}$ do
26:     Select least congested candidate path from $P$. If two paths are equally occupied, select SP among them
27:     Select first available wavelength channel with continuity in the selected path
28:     Update network status
29: Add mapping of $d$ to $Sol$

Return $Sol$
```

In more detail, the algorithm firstly calculates the set of candidate paths in the DCN employing a K-SP routing mechanism, using the length of the paths in hops as the metric (line 1). A Depth First Search (DFS) procedure is employed to determine the routes. Next, it sorts the demands in the demand set in descending order considering the VM that has the most restrictive resource in terms of server occupation (line 2). To exemplify such an ordering, let us consider the following scenario. Let d_1 and d_2 be two VDCs in the demand set, with d_1 having two VMs requesting (CPU cores, storage, memory) server capacities equal to (20, 40, 10)% and (20, 20, 60)%, respectively, while d_2 has also two VMs requesting server capacities equal to (5, 5, 70)% and (30, 20, 45)%, respectively. In this situation, d_2 would precede d_1, since it has the VM with the most restrictive resource (i.e., amount of memory required). Once the demands are ordered, it iterates through them to find an appropriate mapping satisfying their resource requirements (lines 3-29). As a first step, the VMs inside a demand are ordered following the same criterion employed for the demand ordering, giving preference to the VMs with the most restrictive resource in terms of server utilization (line 4). The well-known Timsort sorting algorithm is employed for the ordering of the elements inside a particular collection. Then, the algorithm proceeds to the VM mapping, that is, the selection of the server to deploy the VM (lines 5-23). The strategy followed in this regard consists on finding an already occupied server that has enough capacity to map the VM under consideration and minimizes the following metric:

$$\Delta(n_v s) = \frac{1}{3}\left(\frac{CPU_s^a - CPU_{n_v}}{CPU_s} + \frac{RAM_s^a - RAM_{n_v}}{RAM_s} + \frac{HDD_s^a - HDD_{n_v}}{HDD_s}\right) \qquad (22)$$

In this regard, the algorithm minimizes the number of occupied servers while searching for the server that provides the tightest fit for it. If such a server is not found, the VM is mapped onto the lowest indexed server that is free in any of the DC racks. In this process, the algorithm avoids mapping two VMs belonging to the same VDC onto servers of the same rack (line 10). In more detail, looking at the pseudo-code, the algorithm firstly initializes a set named R_d (line 5), whose purpose is to track the different racks employed to map the VDC instance, thus, avoiding their repetition (we remind the reader that VMs of a VDC must be mapped over different racks). Next, it iterates over all VMs of the VDC instance. For each one of them, it iterates over the entire set of available racks in aims to find a server already in use that has enough resources to host the VM and minimizes the metric introduced in (22). If such a server is found, the VM is mapped to this server and the employed rack is added to R_d. If such a server is not found in the rack, the algorithm keeps track of the lowest indexed server that is completely free at the rack (lines 16-19). Once all racks have been examined, if the VM has not been yet mapped, the algorithm maps it to the server with the lowest index among all the lowest indexed free servers in the racks (lines 20-22). To finalize this procedure, the servers' status are updated according to the decided mapping for the VM (line 23).

Once all VMs have been mapped, i.e., physical server resources have been reserved for all of them, it proceeds to the virtual link mapping. For this, it iterates over all virtual links of the VDC instance aiming to find enough lightpaths to satisfy their bandwidth requirements (lines 24-28). For a particular virtual link, it seeks the candidate path set P for the physical paths that connect the racks onto which the endpoints of the virtual links (i.e. the VMs) have been mapped. Then, for each of these paths it calculates the number of continuous free wavelengths along the path. This is done to ensure that wavelength continuity is maintained for the selected wavelength channels (a transparent DCN is assumed). To do so, it checks the wavelength status in all physical links along the path and determines which wavelength channels are free in all of them. If two or more paths have the same number of occupied wavelengths, the path whose hop count is lower is selected (line 26). Then, the wavelength selection is performed following a First Fit (FF) criterion (line 27). If enough continuous lightpaths are found to satisfy the bandwidth requirements of the virtual link, the virtual link is considered as mapped, reserving the selected optical resources (i.e., wavelengths) in the physical network. Once all virtual links are mapped, a satisfactory mapping of the VDC under study has been found and its details are included into the partial solution found so far (line 29). Next, the heuristic proceeds on finding the mapping of the following VDC in the demand set following the procedure explained above. At the end of the whole procedure, the total problem solution is returned, which includes the full details of the mapping of all VDC instances in the demand set.

In what follows, we will provide a time complexity analysis of the proposed heuristic, considering the internal operations and mechanisms employed. As a first step, the algorithm computes all the candidate physical routes between rack pairs in the DC. This translates to having $\frac{|R|\cdot(|R|-1)}{2}$ different source-destination pairs for which route calculations are needed. Considering that a DFS procedure is employed for the route calculation, for which the average time complexity is equal to $\mathcal{O}(|N_f| + |E_f|)$, the time complexity of the route calculation between rack pairs can be approximated to $\mathcal{O}\left(\frac{|R|\cdot(|R|-1)}{2}\cdot\left(|N_f| + |E_f|\right)\right)$. Next, the algorithm proceeds on sorting the demands, employing the well-known Timsort procedure, thus the time complexity of this step can be considered equal to $\mathcal{O}\ (|D|\cdot log|D|)$. For the next step, the algorithm sequentially calculates the mapping of all the VDC demands, hence having to repeat the mapping procedure a number equal to $|D|$ times. Focusing on the mapping procedure, this is essentially structured in two big steps: the VM mapping and the virtual link mapping. For the VM mapping, the algorithms sorts the VMs according to the most restrictive resource that they are requesting, as explained before, employing the same Timsort procedure. Next, the algorithm essentially iterates among the different VM, rack and server combinations to find a suitable server to deploy the VM. Thus, the time complexity of the VM mapping can be approximated as $\mathcal{O}\left(\left|\overline{N_v^d}\right| \cdot log\left|\overline{N_v^d}\right| \cdot \left|\overline{N_v^d}\right| \cdot |R| \cdot |S_r|\right)$, with $\left|\overline{N_v^d}\right|$ being the average number of VMs per VDC. As for the virtual link mapping, the algorithm iterates through all virtual links to find a suitable candidate physical path and available wavelength channels to fulfil their bandwidth requirements. Assuming that K candidate paths are considered per source-destination rack pairs, the algorithm has to repeat this operation a number of times equal to $K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right|$, with $\left|\overline{E_v^d}\right|$ being the average number of virtual links per VDC while $\left|\overline{B_{e_v}}\right|$ denotes the average number of lightpaths requested per virtual link. Then, for each of the candidate paths, the algorithm has to check for the available wavelength channels at the path, which translates on having to check which wavelengths are free for all physical links in the path, requiring at most $\overline{h_p} \cdot |W|$ operations, where $\overline{h_p}$ denotes the average length in hops for a path between racks in the physical DCN. Thus, the complexity of the virtual link mapping phase can be approximated to $\mathcal{O}\left(K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right| \cdot \overline{h_p} \cdot |W|\right)$. Taking into account all the steps involved, the time complexity of the proposed heuristic can be approximated as $\mathcal{O}\left(\frac{|R|\cdot(|R|-1)}{2} \cdot \left(\left|N_f\right| + \left|E_f\right|\right) + |D| \cdot log|D| + |D| \cdot \left(\left|\overline{N_v^d}\right| \cdot log\left|\overline{N_v^d}\right| \cdot \left|\overline{N_v^d}\right| \cdot |R| \cdot |S_r| + K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right| \cdot \overline{h_p} \cdot |W|\right)\right) \approx \mathcal{O}\left(\frac{|R|^2}{2} \cdot \left(\left|N_f\right| + \left|E_f\right|\right) + |D| \cdot \left(log|D| + \left|\overline{N_v^d}\right|^2 \cdot |R| \cdot |S_r| \cdot log\left|\overline{N_v^d}\right| + K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right| \cdot \overline{h_p} \cdot |W|\right)\right)$. It can be observed that the performance of the proposed heuristic is polynomial and is tightly related to both the size of the physical infrastructure (with special emphasis on the number of racks) and the average size of the VDC requests. Moreover, note that, although the number of assumed servers per rack and the wavelength channels per physical link (S_r and W, respectively) contribute to the time complexity of the heuristic, as mentioned before during the problem statement, such numbers are assumed to be enough to allow for the mapping of all VDC requests in the demand set. Hence, if the initial value for both parameters is chosen carefully, the required time to run the heuristic could be lowered.

5.3.2. Disaggregated scenario

In this section we present the developed heuristic for the disaggregated DC scenario, named AGP-DDCRP. The details of the heuristic are depicted in Pseudo-code 2. Similarly to the server-centric case, the proposed heuristic is based on an adaptive greedy mechanism that iteratively maps all VDCs in the demand set. At the end, it returns a solution containing the mapping of all VDC requests.

Pseudo-code 2: AGP-DDCRP heuristic

```
Input: $G_n, D, K$
Output: $Sol$   //Solution

1:  $P$ ← K-Shortest Paths (SPs) between all pairs of ToRs in $G_n$
2:  Sort demands in $D$ in descending order according to their most restrictive VM in terms of server occupation
3:  For each $d$ in $D$ do
4:    Sort VMs in $N_v^d$in descending order according to their most restrictive resources in terms of server occupation
5:    Sort racks in $R$ in descending order according to their average load
6:    Map first VM in $N_v^d$ onto the least loaded rack
7:    Map the rest of the VMs onto the consecutive indexed racks from the one selected in the previous step
8:    Update rack status
9:    For each $e_v$in $E_v^d$ do
10:     For 1 to $B_{e_v}$ do
11:       Select least congested candidate path from $P$. If two paths are equally occupied, select SP among them
12:       Select first available wavelength channel with continuity in the selected path
13:       Update network status
14:   Add mapping of $d$ to $Sol$

Return $Sol$
```

As in the previous heuristic, the algorithm firstly calculates the set of candidate paths between pairs of ToRs in the DCN employing a K-SP routing mechanism considering the length of the path in hops as the metric based on a DFS procedure (line 1). Next, it starts with the mapping of the VDC instance (lines 2-14). For this, it firstly sorts the VDCs in the demand set in descending order considering the VM that has the most restrictive resource in terms of server occupation, as already explained before (line 2). Then, for each of the VDCs, it sorts the VMs in descending order, also following the same criterion (line 4). Once sorted, the algorithm sorts the racks in the DC in descending order according to their average rack load, computed as the average utilization between CPU cores, storage and memory (line 5). Like before, for all sorting operations the well-known Timsort sorting algorithm is used. Once the racks and the VMs are sorted, the first VM in the VDC is mapped onto the least loaded rack (line 6), while the following VMs are mapped in the next indexed racks from the one selected previously, reserving the necessary computing resources to satisfy their needs (lines 7-8). For instance, if the first VM is mapped onto the rack with index 7, the next VMs will be mapped onto the racks 8, 9, 10, etc. In this way, the algorithm pursues the minimization of the employed racks, balances the load of the racks and avoids mapping the VMs onto racks that may belong to different regions (i.e., clusters) across the DC, which could lead to the utilization of fairly long multi-tier paths to connect them depending on the DCN topology, increasing both the necessary wavelengths per physical link, as well as the total number of employed wavelength channels in the DCN. Once all VMs are mapped, it proceeds to the virtual link mapping, in the same fashion as in the server-centric case (lines 9-13). When all virtual links are mapped, i.e., satisfactory lightpaths meeting their requirements are found, the details of the mapping of the VDC instance are added to the solution found so for (line 14). At the end, the total solution is returned, which includes all the details of the mapping of all VDC in the demand set.

Like for the server-centric case, we will proceed on providing a time complexity analysis of the proposed heuristic. Because the overall structure and operations of the heuristic are the same as in the server-centric case, with the difference that servers are not present, we will only detail the main differences and proceed to depict the final time complexity of the proposed heuristic. After the route calculation and the demand sorting, the algorithm start with the main loop to find the mapping of the demands, one at a time. In this loop, the algorithm first sorts the VM according to their resources. Then, it sorts the racks according to their average load. Once sorted, the ordered VMs are mapped sequentially onto the racks. Such process, considering the two ordering operations and the sequential mapping of the VMs constitutes an average time complexity equal to $\mathcal{O}\left(\left|\overline{N_v^d}\right| \cdot log\left|\overline{N_v^d}\right| \cdot |R| \cdot log|R| \cdot \left|\overline{N_v^d}\right|\right)$. The virtual link mapping procedure is exactly the same as in the server-centric case, thus having the same average complexity. With this, the time complexity of the proposed heuristic can be approximated to $\mathcal{O}\left(\frac{|R|\cdot(|R|-1)}{2} \cdot \left(\left|N_f\right| + \left|E_f\right|\right) + |D| \cdot log|D| + |D| \cdot \left(\left|\overline{N_v^d}\right| \cdot log\left|\overline{N_v^d}\right| \cdot |R| \cdot log|R| \cdot \left|\overline{N_v^d}\right| + K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right| \cdot \overline{h_p} \cdot |W|\right)\right) \approx \mathcal{O}\left(\frac{|R|^2}{2} \cdot \left(\left|N_f\right| + \left|E_f\right|\right) + |D| \cdot \left(log|D| + \left|\overline{N_v^d}\right|^2 \cdot |R| \cdot \log|R| \cdot log\left|\overline{N_v^d}\right| + K \cdot \left|\overline{E_v^d}\right| \cdot \left|\overline{B_{e_v}}\right| \cdot \overline{h_p} \cdot |W|\right)\right)$, with the same definitions introduced for the server-centric case. It can be appreciated in the disaggregated scenario that the performance of the heuristic is also polynomial and highly tied to the size of the physical infrastructure and the VDC requests. A significant difference compared to the server-centric case is the lack of the multiplicative term related to the size of the server set per rack, since in the disaggregated scenario computing resources are directly organized in racks and not in individual server units. For this reason, the time complexity of the heuristic for the disaggregated DC scenario is slightly lower than the one proposed for the legacy server-centric DC case.

6. PERFORMANCE COMPARISON

In this section, we will evaluate the potential reduction on the amount of computing resources needed to allocate a set of VDC requests on top of a disaggregated DC against a traditional server-centric DC. For this purpose, we assume that each rack in the server-centric DC case is equipped with a set of servers. Conversely, in the disaggregated DC, each rack is equipped with a number of aggregated computing resources equal to the sum of the computing resources of all servers in the rack in the server-centric case. This will allow us to perform a fair comparison between the two scenarios. Note that such values will be enough to allow for the mapping of all VDC requests, as we are attempting a DC planning where the optimal computing resource capacity is determined.

Regarding the offered demand set, VDCs are randomly generated following a 2-step procedure: firstly, between 3 or 4 VMs are generated with equal probability. Then, these VMs are interconnected with virtual links with the same probability, avoiding the generation of non-connected virtual graphs. For simplicity, we assume that all virtual links request one wavelength, since our main focus is to compare the computing resource requirements of both DC architectures. To this end, we consider four different VM profiles, each one requesting an amount of computing resources in the form of (number of CPU cores, memory, storage):

- General Purpose (GP) = (8, 48, 800)
- Computing Oriented (CO) = (22, 16, 200)
- Memory Oriented (MO) = (3, 116, 200)
- Storage Oriented (SO) = (3, 16, 1800)

The amount of resources per server is equal to (24, 128, 2000). These values are inspired in the Amazon Elastic Compute Cloud (EC2) service [29]. Following these VM profiles, we consider three offered VDC demand distributions: T1 = (25% GP, 25% CO, 25% MO, 25% SO), T2 = (70% GP, 10% CO, 10% MO, 10% SO) and T3 = (10% GP, 30% CO, 30% MO, 30% SO).

With these assumptions, we start evaluating the impact of the different terms of the multi-objective optimization function in both server-centric and disaggregated DC scenarios. In other words, we evaluate the impact of the α and β parameters in the presented ILP formulations. To this end, we analyse how the main components of the objective function change according to variable values of the weighting parameters. The scenario considered consists of 6 racks interconnected to a central optical switch, with an arbitrary large port count, describing a tree structure. All racks are equipped with 20 servers per rack in the server-centric case. We focus our analysis on the T1 generation profile. Moreover, we focus our studies on two sizes of the demand set, namely, 10 and 20. Additionally, we fix the value of β to 0.5 for the disaggregated DC scenario while varying the value of α. The obtained results are depicted in Tables IV and V, where every data value is obtained by averaging 20 random instances. All executions in this section have been run in PCs with i7-4770 at 3.4GHz CPUs and 16GB of memory, employing CPLEX v12.5 optimization software [30].

TABLE IV

EVOLUTION OF THE OBJECTIVE FUNCTION AS A FUNCTION OF α (SERVER-CENTRIC SCENARIO)

	$\alpha = 0$		$\alpha = 0.25$		$\alpha = 0.5$		$\alpha = 0.75$		$\alpha = 1$	
$\|D\|$	Total # of servers	# of W/link	Total # of servers	# of W/link	Total # of servers	# of W/link	Total # of servers	# of W/link	Total # of servers	# of W/link
10	29.6	23.8	29.6	10.3	29.6	10	29.8	10	33.8	9.8
20	59.8	39.3	59.8	19.3	59.8	18.5	61.1	18.5	66.3	18.2

TABLE V

EVOLUTION OF THE OBJECTIVE FUNCTION AS A FUNCTION OF α (DISAGGREGATED SCENARIO)

	$\alpha = 0$			$\alpha = 0.25$			$\alpha = 0.5$			$\alpha = 0.75$			$\alpha = 1$		
$\|D\|$	# of racks	max rack load	# of W/link	# of racks	max rack load	# of W/link	# of racks	Max rack load	# of W/link	# of racks	max rack load	# of W/link	# of racks	max rack load	# of W/link
10	4	18.6	22.4	5.1	12.4	9.8	5.7	12.5	9.6	6	13.2	9.6	6	13.2	9.5
20	4	37.1	44	5.5	24.9	18.4	5.8	24.6	18.2	6	27.3	18.2	6	27.2	18

It can be appreciated that for $\alpha = 0$ (i.e., when only the computing resources are minimized) the resulting number of employed servers/racks is the minimum, while the employed number of different wavelengths per link is substantially high, both for server-centric and disaggregated scenarios. This is due to the server/rack consolidation discussed in previous sections, which tends to concentrate VMs in a small sub-set of racks, causing the outgoing physical links to become fairly saturated, thus increasing the number of different wavelengths to be equipped per physical link. Conversely, for $\alpha = 1$ it happens the opposite, that is, the number of wavelengths per physical links is minimum while the number of employed servers/racks increases. Indeed, to encourage the reutilization of already employed wavelength channels, rack diversity takes priority consequently increasing the number of employed servers/racks. Any situation in between is a compromise solution between the two competing main sub-objectives. Given these results, we will adopt $\alpha = 0.5$ for the rest of results of this section, since it offers a fair trade-off between the minimization of the computing resources and wavelength channels. Nevertheless, the exact value to be employed can change depending on the scenario under consideration and is left to the DC operator's discretion.

For the next step, we evaluate the impact of the β parameter present in the ILP formulation for the disaggregated DC scenario, which controls the distribution of the VMs along the racks in the physical infrastructure. To this end, we will perform the same study as in the previous analysis, focusing on the same scenario and VM profile. Table VI depicts the obtained results for varying values of β. All data points have been obtained averaging the results of 20 random problem instances.

TABLE VI
EVOLUTION OF THE OBJECTIVE FUNCTION AS A FUNCTION OF β (DISAGGREGATED SCENARIO)

	$\beta = 0$		$\beta = 0.25$		$\beta = 0.5$		$\beta = 0.75$		$\beta = 1$	
$\lvert D \rvert$	# of racks	max rack load	# of racks	max rack load	# of racks	max rack load	# of racks	max rack load	# of racks	max rack load
10	6	13.2	5.8	13.2	5.8	13.3	5.8	13.3	5.6	13.5
20	6	24.6	5.8	24.6	5.7	24.9	5.7	25.7	5.6	25.8

It can be appreciated that, although the differences are relatively small, higher β values imply the utilization of less racks, since the objective function term related to their minimization takes preference as the maximum load per rack increases. The opposite happens for lower values of β, due to the fact that rack balancing plays a more important role, thus lowering the maximum load per rack and, at the same time, increasing the number of employed racks. Any value between the two extremes offers a compromise working point for the two sub-objectives. Given these results, and looking at the table, we will adopt $\beta = 0.25$ for the rest of the executions in this section. Note, as before, that the most suitable value for β may change depending on the scenario under consideration and the intended optimization policy.

After having determined the most suitable values for the α and β parameters, we proceed to evaluate the performance of the proposed heuristics against the ILP formulations. The scenario considered is the same as before. We also focus our analysis on the T1 generation profile. Then, we explore how the value of the objective function evolves when increasing the size of the demand set. The obtained results are depicted in Table VII, where every data value is obtained by averaging 20 random instances.

TABLE VII
HEURISTIC COMPARISON AGAINST ILP FORMULATIONS

		ILP		Heuristic		
Scenario	$\lvert D \rvert$	Objective	Time	Objective	Time (ms)	Error (%)
Server-centric	10	22.5	>12h	23.8	29.2	5.7
	15	30.5	>12h	31.5	33.4	3.3
	20	40.1	>12h	40.7	38.1	1.5
	25	49.6	>12h	50.5	40.2	1.8
	30	62.7	>12h	64.5	48.5	2.9
Disaggregated	10	6.02	>12h	6.62	23.1	10.1
	15	8.54	>12h	9.04	26.7	5.8
	20	11.56	>12h	12.55	30.4	8.5
	25	12.57	>12h	13.57	32.3	7.9
	30	17.1	>12h	18.4	38.2	7.6

It can be appreciated that the proposed heuristics obtain good results when compared to the optimal one of the ILP formulations (less than a 10% relative error observed in almost all scenarios). Additionally, the execution times needed by the heuristics to find the solutions are reduced by several orders of magnitude when compared to the optimal ILPs. Therefore, we can conclude that the proposed heuristics succeed on finding near optimal solutions in much lower execution times when compared to the ILP formulations, motivating their use for solving larger problem instances.

Once validated the goodness of the heuristic mechanisms, we analyse the amount of computing resources to be equipped at the DC racks as a function of the VM profiles of the VDC instances and the size of the demand set. To this end, we consider a DC scenario composed of 64 racks and 48 servers per rack connected in a tree topology, like the one depicted in Figure 4 (top). Figure 5 presents the comparison of the needed resources (CPU cores, storage, memory) for the three considered VM profiles (T1, T2, T3) and the two DC architectures (server-centric and disaggregated). All results presented hereafter are obtained using the proposed heuristic mechanisms, averaging 100 random instances per data point.

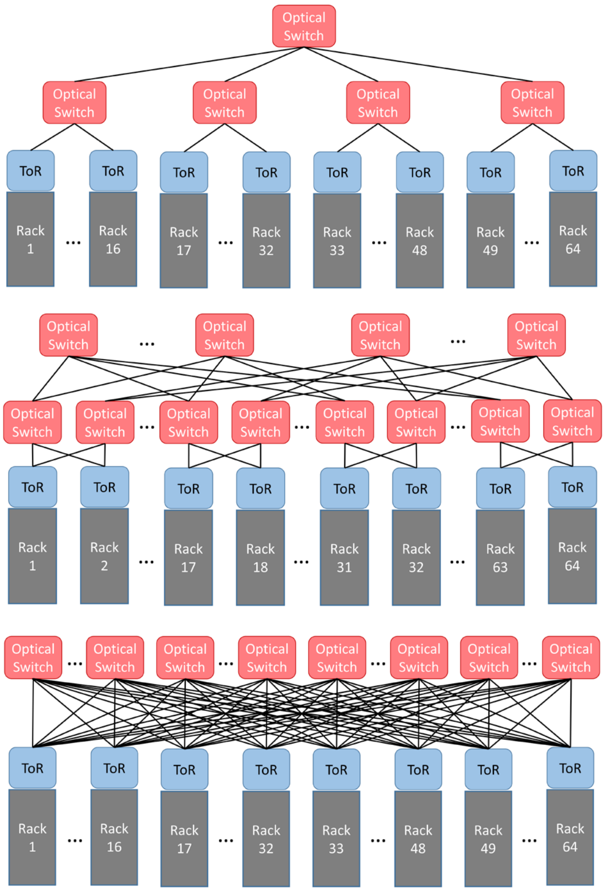

Figure 4. Employed DCN topology: tree (top), fat-tree (middle) and leaf-spine (bottom)

It can be seen that a disaggregated DC architecture allows for a substantial reduction in terms of needed computing resources. Particularly, reductions around 46% can be appreciated for all computing resources and sizes of the demand set with VMs not requiring high amounts of resources (T2). Moreover, these reductions increase up to around 60% with the presence of fairly demanding VMs (e.g., under T3). This is due to the fact that, when allocating a VM, a computing resource in a server may be almost fully utilized, while the utilization of the remainder may stay fairly low. Hence, it may not be possible to allocate a new VM in that server, thus requiring another one. Such a phenomenon requires equipping a large number of server units, even if remaining underutilized, which eventually increases the amount of computing resources required. On the other hand, in a disaggregated DC, as resources can be tightly allocated to match the exact needs of the VMs, it is only necessary to equip the exact amount of computing resources to satisfy all VDC requests. This is the main reason behind the observed computing resource reduction. Additionally, it can be observed a similar performance for all considered VM profiles (T1, T2 and T3) in regards of the needed resources to be equipped at the physical infrastructure. The main reason for this behaviour lays on the average resource utilization of the VMs in all traffic profiles. For instance, focusing on the memory component of the VMs, profiles T1, T2 and T3 result in an average requested memory of 49, 48.4 and 49.2 GB, respectively. Since in a disaggregated DC only the strictly needed computing resources are provisioned, due to the higher flexibility and modularity of the resource assignment, this results in having an overall amount of computing resources equipped at the physical infrastructure very similar across all VM profiles. Nevertheless, different behaviours could be experienced with other VM profiles. On the other hand, in a server centric DC, because computing resource assignment is constrained to server units, although the average resource utilization may be similar, the presence of substantially different VMs greatly impacts on the free resources available at the servers, leading to underutilization and the need to overprovision in regards of the number of needed servers.

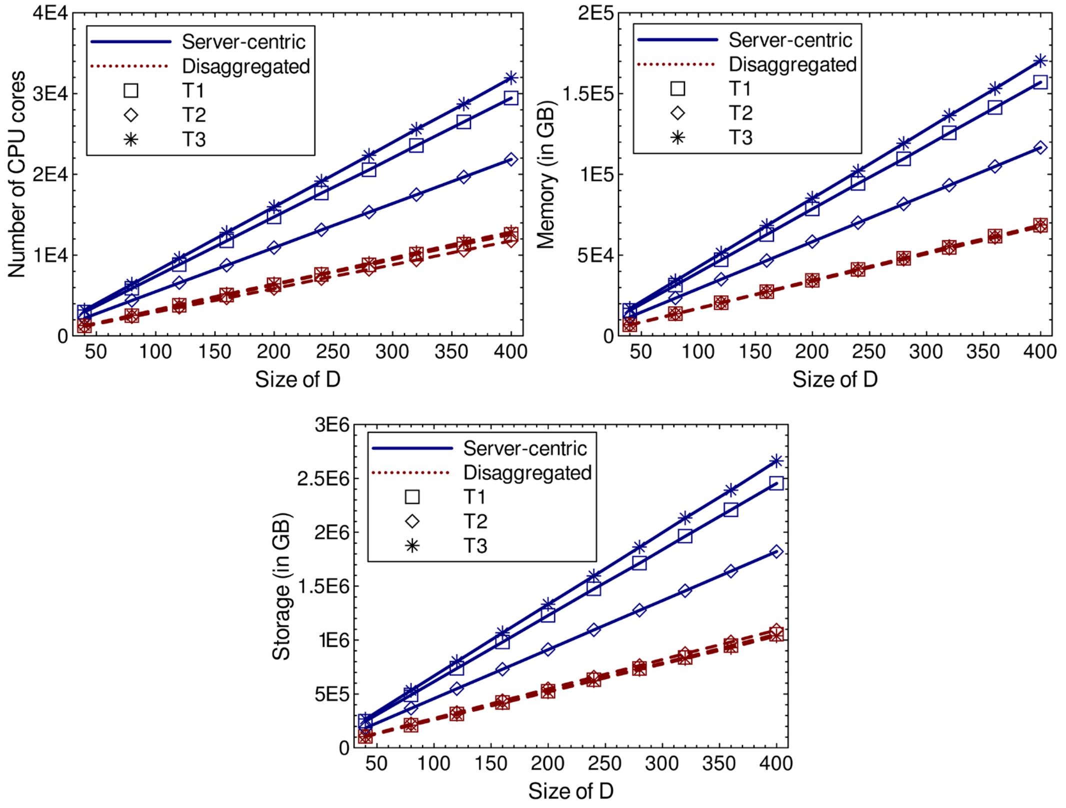

Figure 5. Comparison between server-centric and disaggregated DC architectures in terms of needed computing resources: CPU cores (left), memory (right) and storage (bottom)

To complete our study, we also analyse the influence of the DC topology on the required resources to satisfy a given set of VDC requests. To this end, besides the tree topology employed before, we also contemplate the fat-tree and leaf-spine topologies depicted in Figure 4 (middle) and Figure 4 (bottom). The rest of the DC parameters, as well as the VDC parameters, remain the same. After evaluating a substantially large number of problem instances, we have found that the assumed DCN topology has a very small effect on the number of computing resources needed for the VM mapping. Besides, the number of wavelengths per physical link is not affected by the VM profile, as it is independent of the network utilization. However, we have observed that a disaggregated DC architecture may require fewer wavelengths per physical link when compared to the server-centric scenario depending on the employed DCN topology. To illustrate these results, Figure 6 depicts the number of wavelengths per physical link as a function of the demand set size in the three DCN topologies shown in Figure 4, assuming either a server-centric (left) or a disaggregated (right) DC architecture. To get these results, the T2 VM profile has been considered.

It can be appreciated that for the fat-tree and leaf-spine topologies, the number of employed wavelengths per link in both architectures remains the same. However, in the tree topology, the disaggregated DC architecture requires up to 10 wavelengths per physical link less than the server-centric architecture. This mainly happens as an influence of the mapping of the VMs onto computing resources. In more detail, in the server-centric case, aiming to reduce the number of equipped computing resources, server re-utilization is encouraged to minimize the needed number of server units. However, this may lead to outgoing links from particular racks, thus ToRs, to be slightly more saturated, increasing the number of wavelengths per physical link. This is especially critical in sparsely meshed DCN topologies, as the number of candidate paths is fairly low, thus an increase of the load of a particular link can affect a significant number of source-destination pairs in the DCN. On the other hand, in the disaggregated DC architecture, since VMs are assigned with the exact number of computing resources that they require, there exists a higher degree of freedom on selecting the rack onto which the VM will be mapped, hence, a better network optimization can be achieved while still pursuing the optimization of the computing resource allocation.

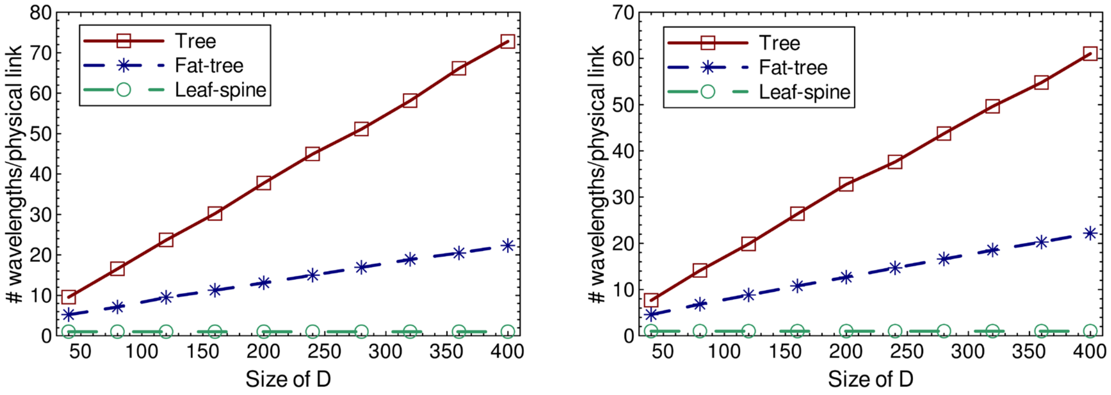

Figure 6. Comparison of the considered DCN topologies regarding number of number of wavelengths used per physical link in server-centric (left) and disaggregated (right) DC scenarios

Additionally, we have observed that a high dependency between the number of wavelengths to be equipped per physical link and the DCN topology exists in both server-centric and disaggregated architectures. In this regard, we can appreciate that the tree topology is the one requiring more wavelengths per physical link, up to three times more than a fat-tree topology and more than one order of magnitude than the leaf-spine topology. This is because the latter has more alternative paths in the DCN, which leads to physical links being less congested. As a result, the number of different wavelengths equipped per physical link remains lower. Special mention requires the case of the spine-leaf topology. We can see that very few wavelength channels are required. This is due to the sheer amount of alternative paths between pairs of ToRs in the network, thus, making it easy to re-utilize a wavelength that has been already employed in another link of the network. In this regard, we can conclude that densely meshed DCNs allow for a better distribution of the traffic inside the DC, as well as an increased resilience against network failures. Such feature is especially valuable in the presence of electrical equipment, for example, when applying Equal Cost Multi-Path (ECMP) routing in Multipath TCP (MPTCP)-based data transmissions, as it allows to substantially increase the network bandwidth [31]. Nevertheless, note that a significant number of network nodes and links are needed to achieve such purposes, which may increase the total network cost, depending on the unitary cost of the optical switches and the links. Thus, a DC operator has to carefully decide in which term of the total DC cost has more interest on minimizing when serving VDC requests. The optimization of the total network cost, taking into account the optimal topology and number of network nodes and links is out of the scope of this paper and left for future work.

7. CONCLUSIONS

Virtual Data Centre (VDC) instances require the joint allocation of computing and network resources. However, traditional server-centric data centre (DC) architectures can lead to the necessity of overprovisioning server units to satisfy the mapping of a set of VDC requests, thus increasing the associated CAPEX. In this paper, we proposed exact Integer Linear Programming (ILP) formulations as well as heuristic-based mechanism to determine the minimum number of computing resources needed to satisfy a known VDC request set on top of a shared DC infrastructure. Through the proposed solutions, we have shown how disaggregated DC architectures can help on reducing the amount of computing resources needed when allocating VDC instances on top of a shared physical infrastructure. In particular, around 46% average reductions can be experimented for fairly balanced Virtual Machine (VM) profiles while the reductions can increase up to 60% in the case of highly specialized VMs. Such results indicate that disaggregated DCs can overcome the limitations of current architectures in regards of efficient computing resource utilization, pleading for their adoption in future DC architectures.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Government through project SUNSET (TEC2014-59583-C2-1-R) with FEDER contribution.

REFERENCES

[1]. T. Wang, Z. Su, Y. Xia and M. Hamdi, “Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing”, IEEE Access, vol. 2, pp. 1481-1496, December 2014.

[2]. Cisco, “Cisco Global Cloud Index: Forecast and Methodology, 2014–2019”, [Available online]: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html, accessed May 2016.

[3]. C. Kachris and I. Tomkos, “A Survey on Optical Interconnects for Data Centers”, IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 1021-1036, Fourth Quarter 2012.

[4]. N. Farrington, G. Porter, S. Radhakrishnan, H. Hajabdolali Bazzaz, V. Subramanya, Y. Fainman, G. Papen and A. Vahdat, “Helios: a hybrid electrical/optical switch architecture for modular data centers”, Proceedings of ACM SIGCOMM 2010, 2010, pp. 339–350.

[5]. J. Perelló, S. Spadaro, S. Ricciardi, D. Careglio, S. Peng, R. Nejabati, G. Zervas, D. Simeonidou, A. Predieri, M. Biancani, H. J. S. Dorren, S. D. Lucente, J. Luo, N. Calabretta, G. Bernini, N.

Ciulli, J. C. Sancho, S. Iordache, M. Farreras, Y. Becerra, C. Liou, I. Hussain, Y. Yin, L. Liu and R. Proietti, "All-Optical Packet/Circuit Switching-based Data Center Network for Enhanced Scalability, Latency and Throughput", IEEE Network, vol. 27, no. 6, pp. 14-22, December 2013.

[6]. G. Saridis, S. Peng, Y. Yan, A. Aguado, B. Guo, M. Arslan, C. Jackson, W. Miao, N. Calabretta, F. Agraz, S. Spadaro, G. Bernini, N. Ciulli, G. Zervas, R. Nejabati and D. Simeonidou, "Lightness: A Function-Virtualizable Software Defined Data Center Network With All-Optical Circuit/Packet Switching", IEEE Journal on Lightwave Technology, vol. 34, no. 7, pp. 1618-1627, 2016.

[7]. C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema", Google Technical Report, 2012, [Available Online] http://code.google.com/p/googleclusterdata/wiki/TraceVersion2, accessed May 2016.

[8]. M. Byrne, "Memory Is Holding Up the Moore's Law Progression of Processing Power", 2014, [Available Online] http://motherboard.vice.com/read/memory-is-holding-up-the-moores-law-progression-of-processing-power, Accessed May 2016.

[9]. Open Compute Project, [Available online] http://www.opencompute.org/, Accessed May 2016.

[10]. Y. Yan, G. Saridis, Y. Shu, B. R. Rofoee, S. Yan, M. Arslan, T. Bradley, N. V. Wheeler, N. H. L. Wong, F. Poletti, M. N. Petrovich, D. J. Richardson, S. Poole, G. Zervas and D. Simeonidou, "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card", IEEE/OSA Journal of Lightwave Technology, vol. 34, no. 8, pp. 1925-1932, April 2016.

[11]. S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi and S. Shenker, "Network Support for Resource Disaggregation in Next-Generation Datacenters", Proceedings of 12th ACM Workshop on Hot Topics in Networks (HotNets 2013), November 2013.

[12]. J. Weiss, R. Dangel, J. Hofrichter, F. Horst, D. Jubin, N. Meier, A. L. Porta and B. J. Offrein, "Optical interconnects for disaggregated resources in future datacenters", Proceedings of 40th European Conference and Exhibition on Optical Communications (ECOC 2014), September 2014.

[13]. J. Kyathsandra and X. Zhou, "Rack Scale Architecture: Designing the Data Center of the Future", Intel Developers Forum 2014 (IDF 2014), September 2014.

[14]. B. P. Rimal and M. Maier, "Workflow Scheduling in Multi-Tenant Cloud Computing Environments", IEEE Transactions on Parallel and Distributed Systems, Early Access, 2016. Doi: 10.1109/TPDS.2016.2556668.

[15]. K.-K. Nguyen, M. Cheriet and M. Lemay, "Enabling infrastructure as a service (IaaS) on IP networks: from distributed to virtualized control plane", IEEE Communications Magazine, vol. 51, no. 1, pp. 136-144, January 2013.

[16]. Interoute VDC service, [Available Online] https://cloudstore.interoute.com/, Accessed May 2016.

[17]. A. Pagès, M. P. Sanchís, S. Peng, J. Perelló, D. Simeonidou and S. Spadaro, "Optimal Virtual Slice Composition Toward Multi-Tenancy Over Hybrid OCS/OPS Data Center Networks", IEEE/OSA Journal of Optical Communications and Networking, vol. 7, no. 10, pp. 974-986, October 2015.

[18]. M. G. Rabbani, R. P. Esteves, M. Podlesny, G. Simon, L. Z. Granville and R. Boutaba, "On tackling virtual data center embedding problem", 2013 IFIP/IEEE International Symposium

on Integrated Network Management (IM 2013), pp. 177-184, May 2013.

[19]. A. Pagès, J. Perelló, F. Agraz and S. Spadaro, “Optimal VDC Service Provisioning in Optically Interconnected Disaggregated Data Centers”, IEEE Communications Letters, vol. 20, no. 7, pp. 1353-1356, July 2016.

[20]. Facebook, “Facebook Open Switching System ("FBOSS") and Wedge in the open”, [Available Online] https://code.facebook.com/posts/843620439027582/, Accessed October 2016.

[21]. Huawei technical white paper, “High Throughput Computing Data Center Architecture”, 2014, [Available Online] http://www.huawei.com/ilink/en/download/HW_349607&usg=AFQjCNE0m-KD71dxJeRf1cJSkNaJbpNgnw&cad=rja

[22]. G. M. Saridis, E. Hugues-Salas, Y. Yan, S. Yan, S. Poole, G. Zervas and D. Simeonidou, “DORIOS: Demonstration of an all-optical distributed CPU, memory, storage intra DCN interconnect”, Proceedings of Optical Fiber Communications Conference and Exhibition 2015 (OFC 2015), March 2015.

[23]. G. M. Saridis, Y. Yan, Y. Shu, S. Yan, M. Arslan, T. Bradley, N. V. Wheeler, N. H. L. Wong, F. Poletti, M. N. Petrovich, D. J. Richardson, S. Poole, G. Zervas and D. Simeonidou, “EVROS: All-optical programmable disaggregated data centre interconnect utilizing hollow-core bandgap fibre”, Proceedings of 41st European Conference and Exhibition on Optical Communications (ECOC 2015), September 2015.

[24]. W. Miao, F. Agraz, S. Peng, S. Spadaro, G. Bernini, J. Perelló, G. Zervas, R. Nejabati, N. Ciulli, D. Simeonidou, H. Dorren and N. Calabretta, “SDN-enabled OPS with QoS guarantee for reconfigurable virtual data center networks”, IEEE/OSA Journal of Optical Communications and Networking, vol. 7, no. 7, pp. 634-643, July 2015.

[25]. Y. Tarutani, Y. Ohsita and M. Murata, “Virtual network reconfiguration for reducing energy consumption in optical data centers”, IEEE/OSA Journal of Optical Communications and Networking, vol. 6, no. 10, pp. 925-942, October 2014.

[26]. P. Samadi, J. Xu and K. Bergman, “Experimental demonstration of one-to-many virtual machine migration by reliable optical multicast”, Proceedings of 41st European Conference and Exhibition on Optical Communications (ECOC 2015), September 2015.

[27]. I. Houidi, W. Louati, W. Ben Ameur and Djamal Zeghlache, “Virtual network provisioning across multiple substrate networks”, Elsevier Computer Networks, vol. 55, no. 4, pp. 1011-1023, March 2011.

[28]. M. Andrews and L. Zhang, “Complexity of Wavelength Assignment in Optical Network Optimization”, IEEE/ACM Transactions on Networking, vol. 17, no. 2, pp. 646-657, April 2009.

[29]. Amazon Elastic Compute Cloud service, [Available Online] https://aws.amazon.com/ec2/, Accessed May 2016.

[30]. IBM CPLEX Optimizer, [Available Online] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, Accessed May 2016.

[31]. H. Han, S. Shakkottai, C. V. Hollot, R. Srikant and D. Towsley, “Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet”, IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1260-1271, December 2006.