On the Complexity of Configuration and Orchestration for Enabling Disaggregated Server Provisioning in Optical Composable Data Centres

ALBERT PAGÈS[1,*], **FERNANDO AGRAZ**[1], AND **SALVATORE SPADARO**[1]

[1] *Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*
[*] *Corresponding author: albert.pages-cruz@upc.edu*

Compiled April 21, 2023

Due to the limitations of traditional data centre (DC) architectures, the concept of infrastructure disaggregation has been proposed. The DC resources are separated in multiple blades to be exploited independently. As a result, composable DC (CDC) infrastructures are achieved, enhancing the modularity of the resource provisioning. However, disaggregation introduces additional challenges that need to be carefully analysed. One relates to the potential complexity increase on the orchestration and infrastructure configuration that need to be performed when provisioning the resources to support services. Such aspect is highly influenced by the distribution of resources at the physical infrastructure. As such, when analysing the performance of a CDC, it becomes essential to also study the related operational complexity of the resource orchestration and configuration phases. Furthermore, the requirements of several tenant services may impose heterogeneous deployments over the shared physical infrastructure, either in the form of disaggregated single-server or multi-server distributions. The associated orchestration/configuration cost is again highly influenced by the data plane architecture of the CDC. With these aspects in mind, in this paper we provide a methodology for the analysis of the complexity of the resource orchestration for a service deployment and the associated configuration cost in optical CDCs, considering various service deployment set-ups. A selected set of CDC architectures found in the literature is employed to quantitatively illustrate how the data plane design and service deployment strategies affects on the complexity of infrastructure configuration and resource orchestration.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

The rise of virtualization technologies and novel provisioning paradigms (such as Network Function Virtualization (NFV)) is increasing the utilization of data centre (DC) sites. In such situation, DC operators are faced with the challenge of accommodating a plethora of tenant services over the shared resources. These services do not only include customer-grade services and applications, but also enterprise/industrial-grade ones. Furthermore, there is an increasing trend on moving telecom functionalities traditionally implemented by dedicated hardware to commodity servers in DCs by means of Virtual Network Functions (VNFs) thanks to the NFV paradigm. All of this leads to a highly heterogeneous service deployment and resource utilization scenario.

Due to the disparate use of computational resources that these types of services entail, traditional DCs based on integrated server units result in inefficient infrastructure utilization. As an answer to this problem, the resource disaggregation paradigm has been proposed [1–3], giving birth to the so called composable DCs (CDCs) in which the computational resources of server units (CPUs, GPUs, memory, storage, ...) are separated into blades to allow for their independent exploitation. In order to support the required high bandwidth and low latencies between hardware modules for an optimal performance, optical network technologies are envisioned to construct the intra-DC network (DCN) fabric.

Thanks to that, infrastructure customization is highly enhanced, since ad-hoc disaggregated servers can be composed by combining multiple hardware blades. Furthermore, better usage of the physical infrastructure can be achieved as disaggregated servers will tightly fit the resource requirements of the services running in them. Nevertheless, CDCs come with their own set of challenges that need to be carefully analysed when evaluating their potential enhanced performance. Some are linked with the limitations of the data plane technologies (e.g., [4]), while others relate to the configuration, control and orchestration of the infrastructure and deployed services (e.g., [5]).

CDCs lead to a potential increase on the complexity of the service deployment and associated configuration, control and orchestration of the infrastructure resources. This stems from the fact that the number of hardware devices is higher in a CDC than in traditional DCs, and highly dependant on the organization of the hardware elements (i.e., the computational blades) and the design of the DC infrastructure. Especially, it is strongly influenced by the architecture of the intra-DCN fabric interconnecting the blades. Hence, the different CDC architecture designs have to be carefully evaluated to understand their impact on the complexity of the orchestration solutions, and the infrastructure configuration cost when providing services.

Beyond pure resource utilization, service deployment has other dimensions to be considered. In particular, many services running in DCs require to be split across multiple servers due to their internal architecture (e.g., front-end, back-end, data base, etc.), heterogeneous sub-service requirements, resilience and so forth. In addition, these servers must communicate between them to properly execute the operational workflows associated to the services. At the same time, there are services that are better deployed over single powerful server units (e.g., Artificial Intelligence (AI)/Machine Learning (ML) applications). Each kind of service deployment (single- or multi-server) carries different implications to the control and orchestration of the CDC (e.g., different number of hardware resources may need to be managed), which affect to the complexity of the control and management framework. This degree of freedom is highly influenced by the architecture of the CDC, even to the point in which some of the deployments may not be achievable due to the hardware distribution.

Most of the works on optically interconnected CDCs focus on the composition of single disaggregated servers to host any service requested by the tenant (e.g., [1, 3]), without considering the requirements of deploying disaggregated multi-server (MS) infrastructures. To achieve MS provisioning, in addition to the construction of multiple disaggregated servers, inter-server communications must be also provided. This arises the challenge of designing the data plane able to support disaggregated server-to-server data exchanges. The key elements here are the network interface cards (NICs) that have to be composed in the servers, or data plane elements that achieve the same exact functionality. Only by means of such a data plane, MS provisioning can be achieved. However, this may result into an increase in the data plane configuration cost and service mapping complexity.

With all these considerations, in this paper we analyse the complexity of orchestration and configuration operations for enabling disaggregated single-server (SS)/MS provisioning in support of tenant services over optical CDC infrastructures. To this end, we elaborate on a methodology for quantifying the service configuration cost and orchestration complexity over a generic CDC data plane. This methodology is aimed to provide a tool for analysing the control and management procedures associated to different CDC designs and gain insights about the main factors that affect the complexity of said operations, as a complement to the analysis of their performance benefits. The rest of the paper is structured as follows: Section 2 reviews the main related work about CDCs; Section 3 presents the analysis methodology for the orchestration complexity and configuration cost of a CDC infrastructure, detailing the several aspects considered for the analysis; Section 4 presents two case studies of CDC architectures found in the literature so as to provide concrete complexity and cost analysis following the previously presented methodology; Section 5 employs the analysis of the previous case studies in order to quantitatively illustrate the influences of the service and CDC structure into the service configuration cost and mapping complexity; finally, Section 6 draws up the main conclusions of the work.

2. RELATED WORK

Several works have proposed solutions for CDC infrastructures focusing on the description of their architectural design and the evaluation of the data plane (e.g., [1–3, 6, 7]). The distribution of the computational blades and the associated technologies is one of the key elements of the CDC design, for which the several works have proposed the usage of mono-resource blades or specialized hardware that combines the capabilities of multiple resources, for instance, the employment of local memory modules by CPU blades, so as to reduce the CPU-to-memory latencies. The other key element has been the design and evaluation of the network to interconnect the distributed blades. In this regard, optical networks based on Wavelength Division Multiplexing (WDM) are seen as the main enabler for a seamless resource disaggregation. Some of the works have focused solely on the design and evaluation of flexible network fabrics to be exploited within CDC infrastructures (e.g., [6]) while others also focused on their integration with the hardware blades and the performance of the full system. As a result, several optical network designs have been proposed (e.g., [1, 2]), with some works advocating for the use of hybrid networks combining electrical and optical switching technologies (e.g., [8]).

Other studies have focused on quantifying the benefits of the resource disaggregation paradigm. For instance, the work in [9] showcased the potential reduction in needed resources in CDCs when dealing with the provisioning of virtualized services in comparison to traditional DC architectures. The benefits of the disaggregation paradigm in terms of service reliability have been studied in [10]. Resource allocation and provisioning strategies for optimizing the resource utilization or service deployment in CDC infrastructures have been investigated as well (e.g., [11]).

As a counterweight, some studies have demonstrated the limitations of CDCs when dealing with the composition of disaggregated servers or the deployment of services. For instance, [12, 13] have studied the penalties introduced by network capacity limits when composing disaggregated servers, showcasing that notorious degradations in service acceptance can be experienced, potentially negating the benefits of CDCs. Nevertheless, there are very few works that deal with the control and orchestration of CDCs, in particular with emphasis on complexity/cost. There is some work related to the performance evaluation of selected management and control aspects of traditional DCs, like [14], which evaluates the performance of abstraction models in Software Defined Networking (SDN)-based optical DCs. However, the study of the complexity of orchestration and control in CDCs remains an almost unexplored subject.

3. SERVICE ORCHESTRATION COMPLEXITY AND CONFIGURATION COST ANALYSIS

The objective of this paper is to analyse the resource orchestration complexity and configuration cost of CDC infrastructures also taking into account the possible service distributions during deployment. To this end, in the following sub-sections we dwell deeper on the details that have to be considered both from the service and the infrastructure perspectives. The last sub-section provides the details of the proposed methodology.

A. Service Considerations

In traditional DCs, virtualized services are usually deployed in the form of Virtual Machines (VMs) fulfilling the computational requirements of the applications (number of CPU cores, memory, storage, etc.). These VMs are then provisioned over server units. Networking resources may be also configured if VM-to-VM communications within a service are required.

This process is more complex in CDCs. First of all, it is required to compose the disaggregated servers that will host the service. This entails the selection of the set of computational blades, linked to the computational capacity required by the service, and their configuration. Moreover, in order to act as a single server unit, these blades will need to exchange data across them, usually from CPU to memory and memory to storage. In integrated servers, such data exchanges are realized through the motherboard, which employs specialized data buses (e.g., PCI-Express). In CDCs, this requires setting up dedicated connections across the blades by configuring the intra-DCN. The deployment of even an SS application entails a higher complexity in the resource orchestration in CDCs as well as a higher configuration cost, since more infrastructure elements need to be properly set-up.

As for the service structure, a significant part of tenant services require to be split across multiple server units/VMs. While the resource provisioning process for this is quite straightforward in traditional DCs, it is more complex in CDCs. First of all, it is necessary to supply the disaggregated servers with networking capabilities. This may take the form of specialized blades (for instance, NIC blades) or equivalent networking resources. In addition, it is required to do the internal connection of the computational blades of the disaggregated server to the networking capabilities. This requires the interconnection of the CPU blades, which are the hardware that will require data exchanges with the other servers, to the networking resources of the server. Lastly, the multiple disaggregated servers need to be interconnected between them by means of additional network connections.

All these extra network connection and hardware provisioning operations can substantially increase the complexity of the control and orchestration in CDCs. Note that in either SS or MS services, the hardware blades constituting a disaggregated server may belong to the same resource group or not. For example, assuming that the blades are organized in racks, it may be possible to select hardware blades belonging to the same rack if the CDC utilization permits it. This option leads to less network resources usage since shorter paths are employed. Nonetheless, if the physical infrastructure is highly occupied or the utilization is fragmented, hardware blades from different racks could be employed. While this increases the chances of successful provisioning of resources, it has associated a higher mapping complexity, since more blade combinations need to be explored across the different racks. It also leads to a higher configuration cost, since longer network paths need to be provisioned and more networking equipment have to be configured.

B. Infrastructure Considerations

The architecture of the CDC infrastructure plays a major role in the service provisioning process and affects to the complexity of the orchestration of resources and the configuration cost. Depending on how the computational resources are distributed across the data plane, and the architecture of the intra-DCN, different resource combinations will need to be explored and ultimately configured to fulfill the service requirements.

First of all, it is important to consider the segmentation levels of the CDC architecture, that is, how the computational resources are grouped and in which hierarchy. In traditional DCs, server units are usually grouped in racks. Then, several racks may be grouped in pods to be finally grouped in clusters. From the physical infrastructure point of view, the server unit is replaced in CDCs by the hardware blade. These hardware blades may be grouped and structured following the same approach as in traditional DCs, or other groupings may be employed. For instance, they may be grouped in computational nodes or cards, which then may be grouped in higher hierarchy elements, such as trays. This, in combination with the possibility that some of the levels found in classical DCs may not be employed, results in that service orchestration may have higher or lower complexity depending on the architecture of the CDC, as different levels of resources have to be explored.

In addition, the composition of the resource groups also plays an important role. The hardware blades can be arranged in different combinations. One approach is to construct groups of resources that are of the same type. For example, in a CDC there may be racks fully composed of CPU blades, while other ones may be composed of memory blades. While this facilitates the segmentation of the DC resources, it may result on having to configure significantly long network connections. Another approach consists on creating heterogeneous groups which combine hardware blades of all types, for example, a rack that hosts blades of CPUs, memory and so on. In this case, the provisioning of services may entail the configuration of less network resources, since near by blades and, thus, shorter paths may be chosen. It can be seen how the composition of the resource groups has a direct impact on the resulting service configuration cost, since more or less networking resources will be required to be configured. The exploration of computational resources during the service mapping process is also affected by this aspect.

Lastly, the architecture of the intra-DCN of the CDC needs to be considered. The network topology of the intra-DCN dictates how many potential routes are present between a pair of hardware blades, which directly affects the orchestration process. In addition, different topologies contribute to a lesser or larger routing complexity depending on the number of network nodes and links of the topology. This also has a direct impact on the configuration cost, as less or more network resources may need to be configured in different implementations of intra-DCN. Besides this, network convergence (or lack of) influences the whole service mapping process, specifically, the part related to intra- and inter-server connectivity establishment. A converged network is the one that employs the same fabric for the traffic between blades of the same disaggregated server and the inter-server traffic. On the other hand, non-converged networks employ parallel fabrics and/or networking devices for separating the two types of traffic, usually by employing different networking technologies to satisfy their heterogeneous requirements. This can result in the presence of more network elements, which need to be considered during the service orchestration and configuration.

C. Analysis Methodology

Following the above-mentioned considerations, we present here the methodology used for analysing the complexity of service provisioning in CDC infrastructures. We focus on both the complexity of the orchestration process and the configuration cost of the resources from a control perspective. In addition, we consider if the service is deployed following an SS or a MS scheme.

Regarding the complexity of the orchestration, we aim to provide a worst case estimation of the resource mapping according to the hardware resources required by the service and the physical data plane. To this end, the computational complexity of the mapping algorithm is studied. We assume here a mapping strategy based on First Fit (FF) mechanisms. We remind the reader that, while we present mapping algorithms for the deployment of services over a CDC infrastructure, the goal of the presented study is not the optimal service mapping or infrastructure utilization, but rather to provide a methodology to analyse how the CDC design affects to the orchestration and configuration complexity/cost of services. As such, it becomes essential to understand the involved operations from a mapping perspective and the involved infrastructure elements. Hence, we opt for simple algorithms that suffice on providing an understanding of the mapping procedure as a function of the physical infrastructure elements and the service deployment strategy.

Next, we present the pseudo-code of the mapping procedure used as baseline for calculating the orchestration complexity for both SS and MS deployment strategies. Besides the structure of the service, the analysis also takes into account if the hardware resources to be provisioned are allowed to belong to different groups, like racks or trays, to better exploit the resource availability across the CDC infrastructure. Such distinction in the mapping procedure can have a significant impact, since employing resources from different groups leads to a more complex routing calculation as well as to longer paths, thus increasing the number of network elements to be configured.

This being said, Algorithm 1 presents the pseudo-code used to analyse the service mapping of an SS deployment, with the criteria that resources need to belong to a single group (SG). In general, service mapping in DC requires two distinguishable phases: node mapping and link mapping. Node mapping entails the choosing of the most suitable servers for allocating the service, while link mapping entails the selection of the most suitable networking resources for server-to-server communications. In CDCs, both phases are still present, with the difference that the node mapping has to select the multiple computational blades that satisfy the service requirements and the link mapping has to find out the networking resources for blade-to-blade (SS and MS) as well as server-to-server communications (MS).

Although there are approaches to tackle both phases in an efficient and coordinated way (e.g., [15, 16]), a simple approach to analyse the mapping complexity is to consider the execution of the phases sequentially. In this regard, we assume an FF strategy for both the node and link mapping phases. For the former, multiple computational blades need to be explored to find out which ones are suitable for server composition. Without loss of generality, we assume that a service requires to be assigned a number of CPU, memory and storage hardware units, indicated by C_s, M_s and S_s, respectively. Each of these units is of the size of a single blade at the CDC physical infrastructure. To find the blades, starting from the larger physical grouping, that is, racks, trays or similar ones, it is required to iteratively explore each group level up to the blade level. Typical strategies also entail the computation of some form of load metric, potentially combining computational and associated networking resources incoming/outgoing from the element, to sort the groups according to their current usage status (lines 2-6) to select the most suited one. Once sorted, if a free blade is found, then it would be assigned to the service, until all required resources are assigned or the current highest group (i.e., rack/tray) has been fully explored and not all resource requirements have been met. In that case, the resource mapping is reset and the next group is explored (lines 7-11).

Algorithm 1. Orchestration of SS-SG service deployment

```
1:  Node mapping
2:  resource_groups = [highest_tier, ..., CPU blades, Memory blades, Storage blades]
3:  for All group in resource_group do
4:      for All element in group.elements() do
5:          compute element metric
6:      sort group
7:  for All group in resource_group do
8:      for All element in group.elements() do
9:          if element is CPU, memory or storage blade, and is free and resources are still required, then allocate
10:     if group == highest_tier then
11:         if service requirements not satisfied, reset mapping
12: Link mapping
13: for All assigned memory blades do
14:     for All assigned CPU blades do
15:         compute paths between CPU and memory blades
16:         for All paths do
17:             for All wavelengths do
18:                 check continuity and available capacity
19:                 if continuous and enough capacity, allocate connection and no further options are explored
20:     for All assigned storage blades do
21:         compute paths between storage and memory blades
22:         for All paths do
23:             for All wavelengths do
24:                 check continuity and available capacity
25:                 if continuous and enough capacity, allocate connection and no further options are explored
```

In general, the link mapping (lines 12-25) in CDCs entails both the intra- and inter-server connection mapping. However, SS-based services only require blade-to-blade (i.e., intra-server) communications. To this end, connections between all pairs of CPU/memory blades and memory/storage blades need to be assigned. For each pair, physical paths need to be computed to find the candidate ones between endpoints. These paths computations can be restricted to the part of the physical infrastructure of interest, thus, not requiring to explore the full CDC graph. For instance, for the case of allocating all the hardware blades within a single group (rack/tray), it is not needed to explore the network beyond of the graph representation of the single group. With the paths calculated, it is required to find the most suitable optical channels (wavelengths) to be used. Although several types of optical networks may be employed depending on the CDC architecture, most of the works found in the literature propose transparent optical networks. Also, due to the wavelength continuity constraint, the resource assignment in this type of networks is more complex compared to non-transparent ones. For this, in order to give an upper bound of the service mapping complexity, we assume that transparent lightpaths need to be established. As such, all the candidate paths are iteratively explored and, in each one of them, the wavelength channels that are continuous and can meet the capacity required by the connection are determined. If a free and continuous wavelength is found, the combination of path and wavelength is selected.

The described mapping algorithm provides a generic skeleton that indicates the main operations that an orchestrator needs to execute when dealing with an SS-SG request mapping. The exact volume of operations is a function of the number of infrastructure resources, which depends on the CDC architecture. Nevertheless, it can be applied to estimate the mapping complexity of a given CDC architecture. It is worth noting here that, although operations inside conditional clauses have an impact on the execution time, we consider that such clauses do not contain additional loops or operations whose computational cost can be considered greater than $\mathcal{O}(1)$, hence, conditionals can be considered cost-less operations. The same procedure can be applied to estimate the complexity in the case of SS service deployment if multiple groups are allowed to be employed for assigning hardware blades to it (SS-MG). The only difference would be at lines 10-11, which are not required if an SS-MG strategy is followed, since all resources can be explored without any restriction until a valid mapping is found.

The orchestration complexity for an MS service deployment can be estimated following the same philosophy. In this case, though, the service can be modelled as requiring a set number of server units to be composed H_s, with each one requiring a set number of CPU, memory and storage blades, $C_{h,s}$, $M_{h,s}$ and $S_{h,s}$, respectively. In addition, each server is required to be assigned with networking capabilities, which can take the form of a NIC blade or hardware with the same functionality. Hence, during the node mapping, server dimension plus the networking resources at the CDC physical infrastructure need to be accounted, explored and assigned to the disaggregated servers. For the link mapping, an MS deployment also requires to allocate connections between all the CPU blades of a disaggregated server to their corresponding assigned networking hardware, as well as interconnecting all the networking capabilities of each server between them for server-to-server communications. With these considerations, Algorithm 2 depicts the pseudo-code of the mapping algorithm for an MS-SG service deployment, only highlighting the differences with respect to the SS case. Likewise, the MS services can also be deployed following an MG strategy for better exploiting the CDC infrastructure resources. In such a case, an MG strategy can have different approaches: 1) distribute the multiple servers across the full infrastructure, but contain a composed server within a single group; or 2) distribute the resources of the multiple servers across the full infrastructure without any kind of restriction. For the analysis at hand, we focus our efforts in the first case. In this scenario, the mapping of an MS-MG deployment follows the same approach as in an MS-SG mapping but removing the limitation of having all the servers in the same group and adding the restriction that a composed server has to be contained within a single group.

Algorithm 2. Orchestration of MS-SG service deployment

```
Node mapping
Same as in SS-SG, accounting for the server dimension and assignment of networking hardware capabilities
Link mapping
for All composed servers do
    for All assigned memory blades do
        for All assigned CPU blades do
            compute paths between CPU and memory blades
            for All paths do
                for All wavelengths do
                    check continuity and available capacity
                    if continuous and enough capacity, allocate connection and no further options are explored
            compute paths between CPU blade and server networking capabilities
            for All paths do
                for All wavelengths do
                    check continuity and available capacity
                    if continuous and enough capacity, allocate connection and no further options are explored
        for All assigned storage blades do
            compute paths between storage and memory blades
            for All paths do
                for All wavelengths do
                    check continuity and available capacity
                    if continuous and enough capacity, allocate connection and no further options are explored
    for All composed servers do
        if Not the same server then
            compute paths between servers networking hardware
            for All paths do
                for All wavelengths do
                    check continuity and available capacity
                    if continuous and enough capacity, allocate connection and no further options are explored
```

Once the mapping of a service has been decided, the infrastructure elements that will host the service need to be configured by the control framework of the CDC. This entails configuring the hardware blades assigned to the service, as well as all the involved network resources for all the connectivity requirements. Different CDC architectures can imply different configuration costs, namely, due to the distribution of the physical infrastructure elements and the optical intra-DCN, the service deployment strategy, as well as the different technologies of the hardware elements (e.g. circuit-based switches, packet-based switches). Additionally, the technology of the control and management layers and the employed protocols may influence on the configuration performance, as analysed in [17]. In this regard, we aim at providing an estimation of the configuration cost for a service deployment taking into account the hardware elements that need to be configured and their associated technology-dependant cost, deriving the asymptotic configuration cost as a function of the service and data plane structure, independently of the control/management technology employed. A simple approach can be to count how many different hardware elements need to be configured per type of device and then multiply these numbers by their associated technology-dependant cost, which indicates for every type of device which are the high level configurations that need to be applied within the hardware block. For example, considering a transparent Optical Circuit Switching (OCS)-based CDC data plane, from a configuration perspective, to establish connectivity across a single intermediate cross-connect, it is required to modify its switching matrix. In this regard, the technology-dependant cost is equal to one. In another case, for instance in a source/destination cross-connect, in which client flows are aggregated to a WDM channel, the technology-dependant cost would be three, since aside from the modification of the switching matrix, it would also be needed to configure the transponders. This provides a rough estimation on how costly a CDC architecture can be from a configuration perspective. With this in mind, the methodology for analysing the configuration cost is summarized in the following steps:

1. Sum the number of hardware elements by type required by the service. This provides the number of the computational blades per type that need to be configured for the service.

2. Using as input the described orchestration complexity analysis methodology, depending on the deployment strategy (SS-SG, SS-MG, MS-SG, MS-MG) and the CDC architecture, devise the worst case mapping result for the endpoints of the network connections. That entails considering the situations in which the endpoints are as far as possible in the physical infrastructure, taking into account the restrictions of the mapping strategy.

3. With the positioning of the end-points, consider the networking path to interconnect each pair of them as required by the mapping strategy following a shortest path approach. This marks the networking devices that need to be configured for a single connection.

4. Once shortest paths between worst-case end-point selection are determined for all of the endpoint pairs, determine the set of networking resources (e.g., optical switches) that need to be configured. Note that in this step, if a resource is employed by two or more different connections, the resource is counted only once. Although the multiple connections may entail different configurations at the resource hardware level, this highly depends on the resource itself and the configuration framework employed. For simplicity, we assume here that the configurations of the multiple connections can be addressed by a single action, e.g. in an OCS switch, the modification of the switching matrix accounts for all the input/output pair cross-connections that need to be configured.

5. Finally, the total configuration cost can be understood as the summation of all hardware devices computed during step 1 and the networking resources determined during step 4, each one multiplied by its technology-dependant cost, which indicates the differences on hardware configurations according to the specific technology of the element.

The obtained numbers provide a cost function that showcases the configuration cost dependence with the service to be deployed as well as the architecture of the CDC.

4. CASE STUDIES

In this section, we apply the proposed methodology to derive the orchestration and control complexity and cost functions associated to the deployment of services over two representative CDC design proposals. Note that while the orchestration and complexity associated to service provisioning is an important part in the operational expenditures (OPEX) evaluation of the CDC architecture, it is not the sole contributor to it. Other important factors are related to quality assurance (QA) operations to ensure the optimal performance of the deployed services/physical hardware, or the energy/power consumption of the physical data plane (blades, network, cooling, etc.). The analysis of such contributors requires thorough modelling of the operations and involved elements, which merit their own studies. In this paper we put the focus on the analysis of the orchestration complexity and configuration cost implications during provisioning operations.

A brief overview of the CDC infrastructure is given for the selected use cases, highlighting their main distinguishable architectural characteristics. Once the particularities of each of the data planes are understood, the functions are derived for all of the four provisioning strategies analysed in the previous section. Before proceeding with the use cases, let us discuss that, although the presented methodology for the analysis is generic and can be applied to any architecture regardless of the implementation of the service mapping algorithm, there are two aspects that still are dependant on the actual employed procedures of the mapping operation: the sorting of elements and the path calculation. Different sorting and path calculation algorithms have different associated complexities. In this regard, we assume that the Merge Sort algorithm [18] is employed, since it is one of most effective and most used sorting algorithms in data sets. This algorithm has a worst case complexity of $\mathcal{O}(N \cdot log_2(N))$, with N the number of elements to be sorted. In regards to the presented service mapping algorithms, Merge Sort is employed at line 6 for the computational hardware sorting and at lines 15 and 21 as a part of the path calculation between blades in algorithm 1, as well as at lines 2, 7, 12, 18 and 25 in algorithm 2 for the same stated purposes. As for the path calculation, the Breadth-First Search (BFS) algorithm [19] is assumed to be employed as one of the leading path calculation algorithms in optical networks in cases in which all the candidate paths between a pair of source-destination nodes have to be computed. The computational complexity of the BFS algorithm is equal to $\mathcal{O}(|V| + |E|)$, being V and E the set of nodes and edges of the graph to be traversed. The BFS algorithm is used for all the path calculations between hardware elements (computational blades, networking devices) assigned to a service deployment. Specifically, it is used at lines 15 and 21 in algorithm 1, and lines 7, 12, 18 and 25 in algorithm 2.

Having said that, before particularizing the methodology for the case studies, we elaborate on the generic complexity of the presented algorithms to facilitate the later understanding of the derived complexities per use case. To this end, we focus on algorithm 1. A first set of operations is dedicated to the exploration and sorting of resources groups, with a complexity equal to $G_1 \cdot (G_2 \cdot (...G_n \cdot (C + C \cdot log_2(C) + M + M \cdot log_2(M) + S + S \cdot \cdot log_2(S)) + G_n \cdot log_2(G_n))... + G_2 \cdot log_2(G_2)) + G_1 \cdot log_2(G_1)$, with $G_1, ..., G_n$ being the number of elements within resource group of tier n down to the one before the blade level, and C, M and S the number of CPU, memory and storage blades at the lowest tier of resource groups (i.e. the blades). The next set of operations is dedicated to the exploration and assignment of resources, with a complexity equal to $G_1 \cdot G_2 \cdot ... \cdot G_n \cdot (C + M + S)$. The last block of operations relates to the link mapping, which entails the connection allocation for every pair of assigned memory-cpu and memory-storage blades. As explained, the BFS algorithm is used here. Once the paths are calculated, these are sorted (by means of the Merge Sort algorithm) from shortest to longest. The operation of verifying the suitability of the candidate paths according to the required capacity between endpoints requires the iteration over them and, for each one, the iteration over its physical links and wavelength channels to determine if the combination of path-wavelength can support the communication. Hence, the complexity of the link mapping phase is equal to $M_s \cdot C_s \cdot (|V| + |E| + P \cdot log_2(P) + W \cdot \sum_{p=1}^{p=|P|} E_p) +$ $+ M_s \cdot S_s \cdot (|V| + |E| + P \cdot log_2(P) + W \cdot \sum_{p=1}^{p=|P|} E_p)$, being V and E the set of nodes and links of the graph when executing the BFS algorithm, P the set of candidate paths, E_p the number of links of the candidate path and W the number of wavelength channels per fiber link. The complexity of the rest of deployment strategies can be derived following a similar reasoning. This being said, the following sub-sections focus on the case studies.

A. DACON

The first case of study is the DACON CDC architecture presented in [2]. In DACON, computational resources are arranged in racks, with R being the rack set, with each rack having computational blades of each type of resources, here noted as C_r, M_r and S_r for the CPU, memory and storage blades, respectively. A Fast Optical Switching (FOS) network is employed for blade-to-blade communications, with several nano-second optical switches (NOS) enabling the communications within the racks (RNOS) and across them (CNOS, MNOS, SNOS), employing W optical wavelength channels per fiber link. Figure 1 depicts a high-level schematic of the DACON architecture. Aside from the use of FOS, the main characteristic of DACON is the use of a flat optical network, with only a tier of switches, meaning that blade-to-blade communications can be achieved employing one or two switches at most. The original DACON architecture did not consider the possibility of MS service deployment, thus, no networking blades were present at the data plane. The work presented in [5] extended the DACON architecture with the inclusion of NIC blades at each of the racks, with each rack having N_r of these blades, to serve as networking capabilities for servers in an MS deployment, as well as a parallel optical network for the NIC-to-NIC communications (bottom network fabric in Figure 1). The extended DACON architecture presents a non-converged CDC architecture, that is, different network fabrics are used for the intra- and inter-server communications, thus meaning that two different graphs need to be explored during the service mapping.

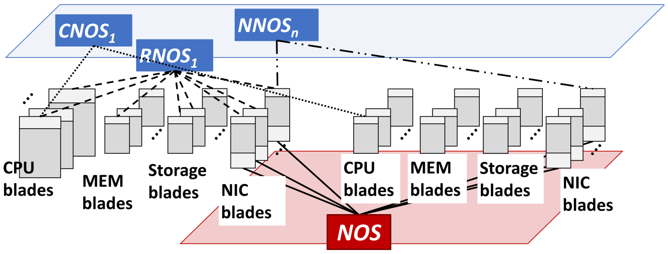

Fig. 1. High-level view of the DACON CDC architecture.

Starting with the SS-SG service deployment strategy, we particularize the presented methodology for the DACON architecture. The main consideration for this case is that the path calculation can be confined within a rack from the BFS perspective, since all hardware blades belong to the same rack, leading to having only one candidate path with a maximum number of hops equal to two. Additionally, the DACON architecture only has two tiers of resources groups: the racks and the blades. Finally, the NIC blades and the parallel optical network to interconnect them are not required to be considered, since only a server is being composed. As a result, the orchestration complexity can be modelled with the following expression:

$$\begin{aligned} &R \cdot (log_2(R) + C_r \cdot (log_2(C_r) + 2) + M_r \cdot (log_2(M_r) + 2) + \\ &+ S_r \cdot (log_2(S_r) + 2)) + M_s \cdot (C_s + S_s) \cdot (2 \cdot (C_r + M_r + \\ &+ S_r + W) + 1) \end{aligned} \tag{1}$$

There are two distinguishable parts in the expression, as it will be in the rest of the mapping complexity expressions across this section, one for the node mapping and another for the link mapping. It can be seen how the mapping complexity exhibits a quadratic growth with the infrastructure resources as well as with the resources assigned to the service. Following the analysis of the SS-SG strategy, the worst case configuration cost entails the configuration of as many as hardware blades as required for the composition of the server plus one NOS to enable the communications within the rack, thus the cost is modelled as the following, considering Γ_{comp}, Γ_{mem}, Γ_{stor} and Γ_{nos} the technology-dependant costs for an individual CPU, memory and storage blade, and NOS, respectively:

$$Cost = C_s \cdot \Gamma_{comp} + M_s \cdot \Gamma_{mem} + S_s \cdot \Gamma_{stor} + \Gamma_{nos} \tag{2}$$

In the SS-MG case, since resources can be mapped across different racks, the full graph of the CDC, leaving aside the NIC blades and the related parallel network, needs to be considered for the path computation operations. This results in having two candidate paths per source-destination pair, with a maximum of four hops per path. The mapping complexity in this case is:

$$\begin{aligned} &R \cdot (log_2(R) + C_r \cdot (log_2(C_r) + 2) + M_r \cdot (log_2(M_r) + 2) + \\ &+ S_r \cdot (log_2(S_r) + 2)) + M_s \cdot (C_s + S_s) \cdot (R + (2R + 3) \cdot \\ &\cdot (C_r + M_r + S_r) + 8W + 2) \end{aligned} \tag{3}$$

The main difference is in the link mapping component, since the path calculation is more complex due to the larger graph considered. The configuration cost for the SS-MG is represented by the expression 4 below. The difference in cost is due to that computational blades may be distributed across different racks, hence the rack-to-rack network may need to be configured.

$$\begin{aligned} &Cost = C_s \cdot \Gamma_{comp} + M_s \cdot \Gamma_{mem} + S_s \cdot \Gamma_{stor} + min(2R; 2M_s \cdot \\ &\cdot (C_s + S_s)) \cdot \Gamma_{nos} \end{aligned} \tag{4}$$

Next, expressions 5 and 6 represent the mapping complexity and configuration cost for the MS-SG service deployment strategy. It can be noted that in this case, aside from accounting for the server dimension, the part related to the network configuration and link mapping has to account for the interconnection of CPU blades to the corresponding NIC blades and the interconnecting between NIC blades, being Γ_{nic} the technology-dependant configuration cost of a NIC blade.

$$\begin{aligned} &R \cdot (log_2(R) + C_r \cdot (log_2(C_r) + 2) + M_r \cdot (log_2(M_r) + 2) + \\ &+ S_r \cdot (log_2(S_r) + 2) + N_r \cdot (log_2(N_r) + 2)) + H_s \cdot M_{h,s} \cdot \\ &\cdot (C_{h,s} + S_{h,s}) \cdot (2 \cdot (C_r + M_r + S_r + N_r + W) + 1) + H_s \cdot \\ &\cdot C_{h,s} \cdot (2 \cdot (C_r + M_r + S_r + N_r + W) + 1) + H_s \cdot (H_s - 1) \cdot \\ &\cdot (2 \cdot (N_r + W) + 1) \end{aligned} \tag{5}$$

$$\begin{aligned} &Cost = H_s \cdot (C_{h,s} \cdot \Gamma_{comp} + M_{h,s} \cdot \Gamma_{mem} + S_{h,s} \cdot \Gamma_{stor} + \\ &+ \Gamma_{nic}) + 2 \cdot \Gamma_{nos} \end{aligned} \tag{6}$$

Lastly, expression 7 represents the configuration cost for the MS-MG case. Servers are distributed across racks, hence the configuration cost increases. As for the mapping complexity, it remains the same as in expression 5. This is due to the fact that DACON employs a non-converged parallel network for NIC-to-NIC communications, which needs to be considered in both the MS-SG and MS-MG strategies, thus resulting in the same mapping complexity.

$$Cost = H_s \cdot (C_{h,s} \cdot \Gamma_{comp} + M_{h,s} \cdot \Gamma_{mem} + S_{h,s} \cdot \Gamma_{stor} + \\ + \Gamma_{nic}) + (H_s + 1) \cdot \Gamma_{nos} \tag{7}$$

B. NetCoD

The other case of study is the NetCoD architecture presented in [8]. As in the DACON architecture, the CDC resources are organized in heterogeneous racks hosting all types of computational resources. The main difference steams from the fact that blades are organized in compute nodes, with each rack hosting a V_r number of them and each node hosting C_v, M_v and S_v CPU, memory and storage blades/units, respectively. At the same time, racks are organized in clusters, resulting in a more hierarchical CDC architecture. Figure 2 depicts a high level view of the NetCoD CDC architecture. The other main characteristic of the NetCoD is the use of a leaf-spine-based converged optical network to intercommunicate the multiple blades. Since it is a converged network, the same fabric is employed for intra- and inter- disaggregated server communications, with NetCoD considering the possibility to interconnect several composed servers. At each node, a Node Hub Controller (NHC) element acts as the ingress/egress of the nodes, having a set of optical interfaces. The NHC is connected to a Semiconductor Optical Amplifier (SOA) switch, which, in turn is connected to a passive optical backplane that connects all the nodes in the rack. This element is connected to a Top of the Rack (ToR) optical switch for rack-to-rack communications. The ToR at each rack is connected to a pair of Top of the Cluster (ToC) optical switches in a leaf-spine fashion in order to provide path diversity. Finally, all the ToCs are connected in a star fashion to a DC gateway (DC Gw).

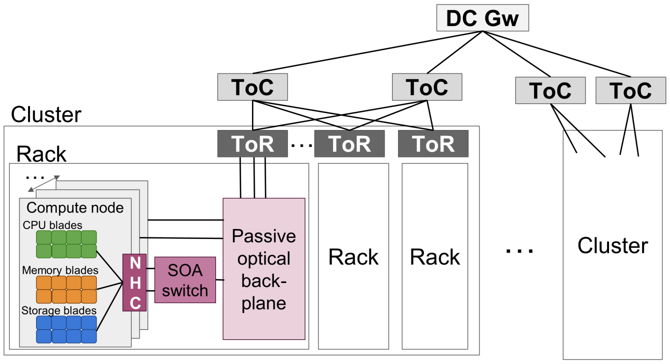

Fig. 2. High-level view of the NetCoD CDC architecture.

The rest of the section is devoted to apply the analysis methodology to the presented NetCoD data plane architecture. In this case, for the exploration and ordering of resources, three tiers of groups are considered: racks, nodes and blades. With this consideration, the mapping complexity for the SS-SG service deployment strategy is represented by the expression below, taking into account that there are two candidate paths between blades within a rack, with a worst hop count equal to six.

$$R \cdot (log_2(R) + V_r \cdot (log_2(V_r) + C_v \cdot (log_2(C_v) + 2) + M_v \cdot \\ \cdot (log_2(M_v) + 2) + S_v \cdot (log_2(S_v) + 2))) + M_s \cdot (C_s + S_s) \cdot \\ \cdot (11V_r + 12W + 3) \tag{8}$$

As in the previous case study, the complexity exhibits a quadratic growth with respect to the infrastructure and service elements. The difference resides in the slop of such growth, which depends on how the computational resources are organized as well as the topology of the intra-DCN. The cost configuration of the SS-SG is modelled by the following expression:

$$Cost = C_s \cdot \Gamma_{comp} + M_s \cdot \Gamma_{mem} + S_s \cdot \Gamma_{stor} + min(V_r \cdot \\ \cdot (\Gamma_{hnc} + \Gamma_{soa}); (C_s + M_s + S_s) \cdot (\Gamma_{hnc} + \Gamma_{soa})) \tag{9}$$

Similarly to the DACON case, the cost configuration has a component related to the number of blades assigned to the service and another related to the different network elements that need to be configured. In this case, intra-rack communications entail the use of multiple devices, namely, NHC and SOA switches (Γ_{hnc} and Γ_{soa} are their technology-dependant configuration costs, respectively). The use of a more hierarchical intra-DCN can lead to a more rapid growth, due to the extended number of network devices, especially when compared to flatter intra-DCN design approaches, such as the DACON case, which only requires one device to be configured (NOS). This difference is accentuated in the MG deployment strategies, since paths crossing several racks or clusters may be needed. With this in mind, expressions 10 and 11 represent the mapping complexity and configuration cost for the SS-MG service deployment strategy, with Γ_{tor}, Γ_{toc} and Γ_{gw} as the technology-dependant configuration costs for a ToR, ToC and Gw, respectively.

$$R \cdot (log_2(R) + V_r \cdot (log_2(V_r) + C_v \cdot (log_2(C_v) + 2) + M_v \cdot \\ \cdot (log_2(M_v) + 2) + S_v \cdot (log_2(S_v) + 2))) + M_s \cdot (C_s + S_s) \cdot \\ \cdot (R \cdot (11V_r + E_{tor} + 4) + 4C + 1 + 8E_{tor} \cdot (12W + \\ + log_2(E_{tor}) + 3)) \tag{10}$$

$$Cost = C_s \cdot \Gamma_{comp} + M_s \cdot \Gamma_{mem} + S_s \cdot \Gamma_{stor} + min(R \cdot (V_r \cdot \\ \cdot (\Gamma_{nhc} + \Gamma_{soa}) + \Gamma_{tor}) + 2C \cdot \Gamma_{toc} + \Gamma_{gw}; (C_s + M_s + S_s) \\ \cdot (\Gamma_{nhc} + \Gamma_{soa} + \Gamma_{tor} + \Gamma_{toc}) + \Gamma_{gw}) \tag{11}$$

NetCoD considers that multiple links may exist between the passive optical back plane and the ToR switch. This determines how many routes there are between blades of different racks or clusters, hence, it appears as a component of the complexity expression as E_{tor}. While having more available paths favors the service acceptance, it usually entails a higher mapping complexity, as it is reflected in Equation 10.

Next, we analyse the MS service deployment strategies. NetCoD employs a converged network, where the HNC modules are used to implement intra- and inter-server communications. From a configuration cost perspective, this means that no extra number of devices needs to be configured. Nevertheless, from a mapping complexity perspective, it entails to select one of the HNC modules employed for blade-to-blade communications of the disaggregated server as the networking hardware, like in the NIC blade selection in DACON. Then, suitable optical connections need to be established from the CPUs to the HNC modules and between HNC modules of different servers, as explained in Section 3C. This being said, expressions 12 and 13 represent the mapping complexity and configuration cost for the MS-SG case.

$$\begin{aligned}&R \cdot (log_2(R) + V_r \cdot (log_2(V_r) + C_v \cdot (log_2(C_v) + 2) + M_v \cdot \\ &\cdot (log_2(M_v) + 2) + S_v \cdot (log_2(S_v) + 2))) + H_s \cdot M_{h,s} \cdot \\ &\cdot (C_{h,s} + S_{h,s}) \cdot (11V_r + 12W + 3) + H_s \cdot (V_r \cdot log_2(V_r) + \\ &+ (C_{h,s} - 1) \cdot (5V_r + 10W + 9) + (H_s - 1) \cdot (5V_r + 4W + 3))\end{aligned} \tag{12}$$

$$\begin{aligned}&Cost = H_s \cdot (C_{h,s} \cdot \Gamma_{comp} + M_{h,s} \cdot \Gamma_{mem} + S_{h,s} \cdot \Gamma_{stor}) + min(V_r \cdot \\ &\cdot (\Gamma_{hnc} + \Gamma_{soa}); H_s \cdot (C_{h,s} + M_{h,s} + S_{h,s}) \cdot (\Gamma_{hnc} + \Gamma_{soa}))\end{aligned} \tag{13}$$

It can be seen how the configuration cost function remains equal to the SS-SG case, while the mapping complexity is larger due to the higher number of optical connections that need to be orchestrated. To finish the analysis of the NetCoD architecture, expressions 14 and 15 focus on the mapping complexity and configuration cost for the MS-MG case, respectively.

$$\begin{aligned}&R \cdot (log_2(R) + V_r \cdot (log_2(V_r) + C_v \cdot (log_2(C_v) + 2) + M_v \cdot \\ &\cdot (log_2(M_v) + 2) + S_v \cdot (log_2(S_v) + 2))) + H_s \cdot M_{h,s} \cdot \\ &\cdot (C_{h,s} + S_{h,s}) \cdot (11V_r + 12W + 3) + H_s \cdot (V_r \cdot log_2(V_r) + \\ &+ (C_{h,s} - 1) \cdot (11V_r + 12W + 3) + (H_s - 1) \cdot (R \cdot (11V_r + \\ &+ E_{tor} + 4) + 4C + 1 + 8E_{tor} \cdot (12W + log_2(E_{tor}) + 3)))\end{aligned} \tag{14}$$

$$\begin{aligned}&Cost = H_s \cdot (C_{h,s} \cdot \Gamma_{comp} + M_{h,s} \cdot \Gamma_{mem} + S_{h,s} \cdot \Gamma_{stor}) + min(V_r \cdot \\ &\cdot (\Gamma_{hnc} + \Gamma_{soa}); H_s \cdot (C_{h,s} + M_{h,s} + S_{h,s}) \cdot (\Gamma_{hnc} + \Gamma_{soa})) + \\ &+ min(R \cdot \Gamma_{tor} + 2C \cdot \Gamma_{toc} + \Gamma gw; H_s \cdot (\Gamma_{tor} + \Gamma_{toc}) + \Gamma_{gw})\end{aligned} \tag{15}$$

Note how the component related to the node mapping complexity remains the same as well as the component related to the cost of the blade and intra-server network configurations. However, due to the need to establish server-to-server communications, the orchestration and configuration of CPU-to-NHC and HNC-to-NHC add extra components in the expressions.

5. NUMERICAL COMPARISON

To better illustrate the implications of CDC architecture design and service deployment strategy in the complexity and cost of the orchestration/control layers, here we present a graphical comparison applied to the presented case studies. To do so, we numerically evaluate the derived expressions as a function of the data plane and service characteristics. Since the expressions have a dependence on multiple factors, the evaluation will be done by fixing the value of some of the variables in order to showcase the growth respect to others.

We start with the service configuration cost. This value has a dependence with both the number of available blades and networking hardware elements at the physical infrastructure and the number of resources required to deploy a service, i.e., the number of blades per type. We start by illustrating the variation of the cost focusing on the service resource variation, that is, the number of resource blades per type required by the service, assuming that an equal number is required for all of them. To this end, the physical infrastructure needs to be fixed. We take as a baseline a traditional integrated DC of 12800 servers, which represents a medium sized DC infrastructure. A typical rack in a DC has 40 servers per rack. Then, a cluster has usually around 10 racks per cluster. With these numbers, the equivalent DACON data plane has 160 racks with 80 blades per rack; for the NetCoD, assuming that each node has 4 computational blades/units, this translates on having 3200 computational nodes, 80 racks with 40 nodes per rack and 8 clusters. Figure 3 depicts the configuration cost for both architectures and the four service deployment strategies as a function of the requested resources per type per server, considering 3 servers per service in the MS deployments. Regarding the technology-dependant configuration costs per hardware element, note that in all the studied architectures and deployment scenarios the configuration of the blades assigned to service is always required. As a result, all cost functions, when compared, have the same offset, which is precisely the cost associated to blade configurations, and only differ in the part related to the DCN configuration. For this reason, we decided to focus the analysis on the networking aspects of the architectures, thus fixing $\Gamma_{comp} = \Gamma_{mem} = \Gamma_{stor} = 1$ as the blades do not play a relevant role in the comparison. As for the rest, in the DACON architecture, the main configuration that needs to be done in all NOS nodes is the update of their lookup table, hence $\Gamma_{nos} = 1$, while for the NIC blades, it is required to modify the configuration of the switching matrix plus the input/output ports due to the change of network fabric, resulting in $\Gamma_{nic} = 3$. In the NetCoD architecture, the configuration of the SOA switches, ToRs, ToCs and GW, basically entails the modification of their switching matrix as OCS technology is employed, thus resulting in $\Gamma_{soa} = \Gamma_{tor} = \Gamma_{toc} = \Gamma_{gw} = 1$, while the configuration of the HCN, due to its opto-electronic nature, entails the configuration of the electrical client port coming from a blade, the configuration of the outgoing optical port and the configuration of the switching matrix, resulting in $\Gamma_{nhc} = 3$.

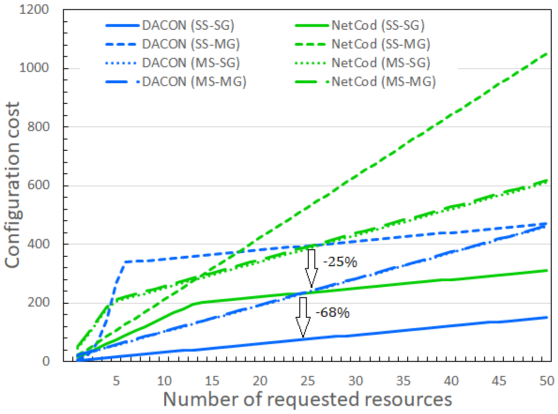

Fig. 3. Configuration cost as a function of the requested resources (blades) per type.

Interestingly, despite employing a non-converged network, DACON presents a lower configuration cost than NetCoD because its design based on a flattened network infrastructure reduces the length of the paths and, thus, the number of network elements to be configured. This sheds an important insight which is that, although, in principle, converged networks require less devices to be configured, it is also important to design a

CDC architecture with a flatter DCN, with less network devices required for establishing connections. For instance, when comparing SS-SG deployments, a flatter network results in around 30-70% lesser configuration cost with respect to more hierarchical networks, while differences in the range of 25-50% are observed when comparing MS-SG deployments. Besides this, in general, MS approaches lead to greater configuration costs since more infrastructure elements need to be configured, both computational blades and networking devices. For instance, when comparing SG deployments, an MS type of service increases the configuration cost by around three fold, due to the larger number of networking hardware involved for the network connections. Paradoxically, this rationale is confirmed by the SS-MG cases, due to the assumptions done in the design of the approaches, which are detailed in section 3C. More specifically, for the sake of simplicity in the analysis, in the MS-MG scheme we assumed that, although the service can be distributed in multiple groups, a single server is only distributed in a single group (e.g., a rack), in opposition to SG-MG where we assume complete freedom on the selection of hardware blades to deploy the server, thus leading to this apparently contradictory result, as a consequence of the longer network paths. Similar conclusions can be extracted from Figure 4, which showcases the cost as a function of the total number of available blades per type at the CDC, assuming the grouping employed as before, as well as fixing the service characteristics to 3 blades per type and 3 servers (only MS cases). It can be seen that a flatter CDC architecture leads to lesser configuration costs, since less networking elements are required. Thus, network convergence and a flat distribution of the fabric should be pursued together in CDC design, as it becomes essential to optimize the distribution of the intra-DCN from a scale perspective. Lastly, as commented during the expressions derivation, note how the cost for MS strategies within a single architecture are very close to each other. This again is a consequence of the assumptions made for the allocation schemes, making MS-SG and MS-MG strategies differ only in the cost of the server-to-server connection configurations.

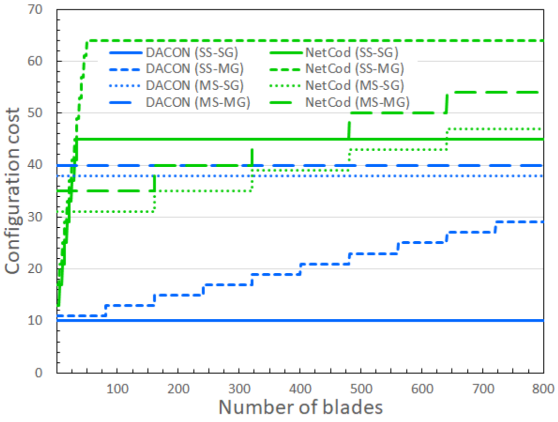

Fig. 4. Configuration cost as a function of the total number of available blades per type at the data plane.

Next, we evaluate the service mapping complexity. We start by depicting the evolution of the mapping complexity as a function of the service characteristics. To this end, we fix the infrastructure resources to the same values as before, with the addition of fixing W to 8 for all the scenarios and E_{tor} to 8 links in the NetCoD architecture. Figure 5 depicts the evolution of the mapping complexity, using the same approach as in Figure 3 for the fixed parameters.

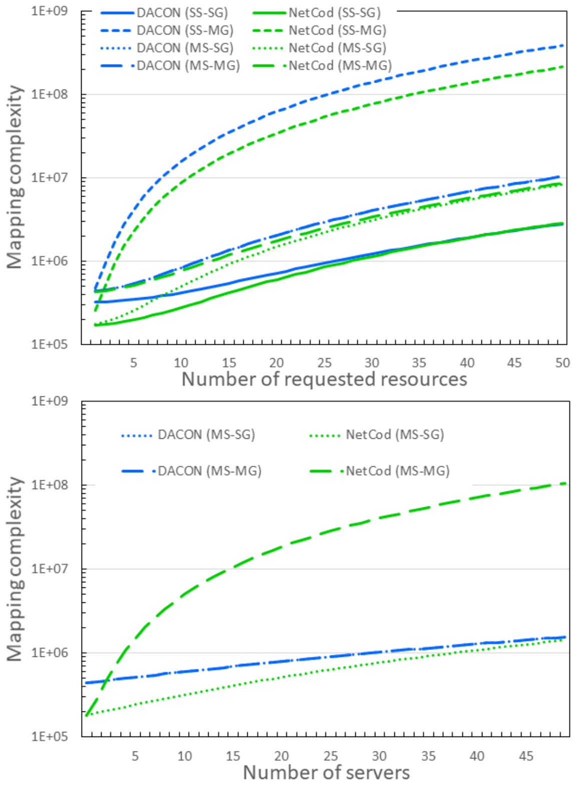

Fig. 5. Mapping complexity as a function of the requested resources (blades) per type (top) and servers (bottom).

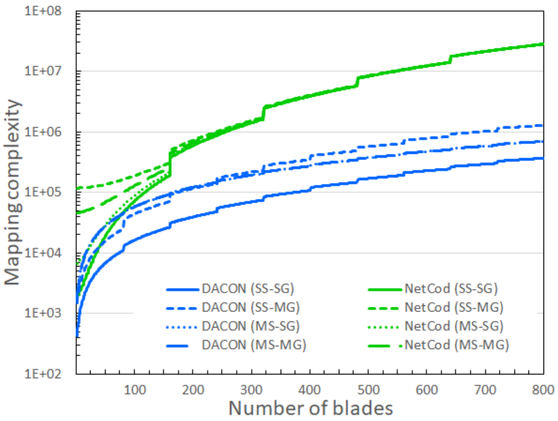

Fig. 6. Mapping complexity as a function of the total number of available blades per type at the data plane.

With the already discussed exception of the SS-MG, as expected, MS deployments result in a higher complexity, due to the larger number of blades and network paths that need to be orchestrated. The same applies to MG with respect to their SG counterparts. In addition, note how, although the complexity of the DACON architecture is higher for a low number of resources/servers, the complexity of NetCoD grows with the number of servers. This is due to the fact that more resource groups need to be explored and sorted during the node mapping as well as a larger network graph in the MS scenarios. This highlights again that the scale and organization of the CDC plays a key role on the potential impact to the orchestration framework. This can be further observed in Figure 6, which depicts the mapping complexity as a function of the total number of vailable blades per type at the CDC. Again, MS and MG strategies result in higher complexities, with the addition that more hierarchical CDC architectures and dense intra-DCNs generally result in higher complexities for the same reasons as before.

6. CONCLUSIONS

Resource disaggregation promises to bring a higher flexibility in terms of computational resources utilization, which, at its turn, would enhance the provisioning of services in a shared DC infrastructure. Nevertheless, the increased complexity of the data plane of CDC infrastructures carries its associated set of challenges. Due to the larger number of computational resources and network elements, the operations related to service orchestration and infrastructure configuration can be more complex/costly. In order to provide insights about this, in this paper, we have detailed which are the main factors on the design of CDC architectures as well as on the service deployment strategies that can contribute to the operational expenditures of a CDC when dealing with service provisioning and infrastructure composition. Additionally, we proposed a methodology to quantitatively assess the service mapping complexity and configuration cost of a generic CDC architecture. The methodology can be then particularized for any specific CDC architecture and serve as tool to understand the potential advantages and drawbacks of a CDC architecture design. Through the use of two case studies, we illustrated how the characteristics of the service and the physical infrastructure influence on the overall orchestration and control complexity for service provisioning operations. In this regard, it has been shown that a CDC architecture with a flatter intra-DCN results in less configuration cost and mapping complexity when compared to more hierarchical architectures (around 30-57% less configuration cost). This highlights that the design of the intra-DCN network has a notorious impact on the cost and orchestration complexity. As such, it becomes an essential aspect that should be optimized during CDC design.

FUNDING

This work has been supported by the Spanish Government through project TRAINER-B (PID2020-118011GB-C22) with FEDER contribution.

REFERENCES

1. V. Mishra, J. L. Benjamin and G. Zervas, "MONet: heterogeneous Memory over Optical Network for large-scale data center resource disaggregation," Journal of Optical Communications and Networking, vol. 13, no. 5, pp. 126-139, May 2021.
2. X. Guo, et al., "DACON: a reconfigurable application-centric optical network for disaggregated data center infrastructures," Journal of Optical Communications and Networking, vol. 14, no. 1, pp. A69-A80, Jan. 2022.
3. J. Gonzalez, et al., "Optically Connected Memory for Disaggregated Data Centers," 2020 Proceedings of IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Sept. 2020, Porto (Portugal).
4. R. Lin, et al., "Disaggregated Data Centers: Challenges and Tradeoffs," IEEE Communications Magazine, vol. 58, no. 2, pp. 20-26, Feb. 2020.
5. F. Agraz, A. Pagès and S. Spadaro, "Data Plane Technology-agnostic Control and Orchestration Architecture for Optical Disaggregated Data Centers," 2022 Optical Fiber Communications Conference and Exhibition (OFC), Mar. 2022, san Diego (USA).
6. X. Xiao et al., "Silicon Photonic Flex-LIONS for Bandwidth-Reconfigurable Optical Interconnects," IEEE Journal of Selected Topics in Quantum Electronics, vol. 26, no. 2, pp. 1-10, Mar.-Apr. 2020.
7. H. M. Mohammad Ali, T. E. H. El-Gorashi, A. Q. Lawey and J. M. H. Elmirghani, "Future Energy Efficient Data Centers With Disaggregated Servers," in Journal of Lightwave Technology, vol. 35, no. 24, pp. 5361-5380, Dec. 2017.
8. O. O. Ajibola, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Network Topologies for Composable Data Centers," in IEEE Access, vol. 9, pp. 120955-120984, Aug. 2021.
9. A. Pagès, R. Serrano, J. Perelló and S. Spadaro, "On the benefits of resource disaggregation for virtual data centre provisioning in optical data centres", Elsevier Computer Communications, vol. 107, pp. 60-74, Jul. 2017.
10. C. Guo, et al., "Exploring the Benefits of Resource Disaggregation for Service Reliability in Data Centers," IEEE Transactions on Cloud Computing, Early Access, Feb. 2022.
11. A. Lin, C. Li, W. Liao and H. Franke, "Capacity Optimization for Resource Pooling in Virtualized Data Centers with Composable Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 2, pp. 324-337, Feb. 2018.
12. Y. Cheng, M. De Andrade, L. Wosinska and J. Chen, "Resource Disaggregation Versus Integrated Servers in Data Centers: Impact of Internal Transmission Capacity Limitation," 2018 European Conference on Optical Communication (ECOC), Sept. 2018, Rome (Italy).
13. A. Pagès, F. Agraz and S. Spadaro, "Analysis of Service Blocking Reduction Strategies in Capacity-Limited Disaggregated Datacenters," 2020 Optical Fiber Communications Conference and Exhibition (OFC), May 2020, San Diego (USA).
14. M. Licciardello, et al., "Performance evaluation of abstraction models for orchestration of distributed data center networks," 19th International Conference on Transparent Optical Networks (ICTON), Jul. 2017, Girona (Spain).
15. G. Sun, et al., "Toward SLAs Guaranteed Scalable VDC Provisioning in Cloud Data Centers," IEEE Access, vol. 7, pp. 80219-80232, Jun. 2019.
16. K. M. Nwe and Y. M. Zaw, "Efficient Mapping for VM Allocation Scheme in Cloud Data Center," 2020 IEEE Conference on Computer Applications(ICCA), Mar. 2020, Yangon (Myanmar).
17. R. Vilalta et al., "Experimental evaluation of control and monitoring protocols for optical SDN networks and equipment [Invited Tutorial]," in Journal of Optical Communications and Networking, vol. 13, no. 8, pp. D1-D12, Aug. 2021.
18. J. Lobo and S. Kuwelkar, "Performance Analysis of Merge Sort Algorithms," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Jul. 2020, Coimbatore (India)
19. G. Lopes Andrade and D. Hoffmann Thomas, "An Optimized Breadth-First Search Algorithm for Routing in Optical Access Networks," IEEE Latin America Transactions, vol. 17, no. 07, pp. 1088-1095, Jul. 2019.