Anastasopoulos, M. P., Tzanakaki, A., & Simeonidou, D. (2016). Scalable monitoring and optimization techniques for mega-scale data centers. IEEE Journal of Lightwave Technology, 34(8), 1980-1989. DOI: 10.1109/JLT.2016.2522654

Scalable monitoring and optimization techniques for mega-scale data centers (Invited)

Markos P. Anastasopoulos, Anna Tzanakaki, and Dimitra Simeonidou

***Abstract*—This paper focuses on the design of service provisioning schemes suitable for mega data center (DC) infrastructures. A major issue linked with the operation of these infrastructures is scalability caused by the increased number of resources available in mega-size highly-dense DCs and the associated requirements for control and management information. To address this scalability issues, we propose for the first time to monitor and optimize the operation of mega DCs adopting graph factorization combined with compressive sensing theories. This approach takes advantage of the spatial and temporal correlation of compute, and network resource requests, to monitor and optimize metrics, such as delay and energy with reduced control and management information. Our modelling results indicate drastically reduced volume of traffic transferred from the data to control plane and number of optimization process variables.**

***Index Terms*—compressive sensing, graph factorization, mega data centres, optical packet switching, network optimization.**

I. Introduction

Big data, Cloud and Content Delivery are driving the increase of global internet traffic expected to exceed 1.6 zettabytes by 2018. These require to store and process massive amounts of data and drive the need for mega-size Data Canters (DCs) scaling up to hundreds of thousands of server and storage modules interconnected with high speed communications links. The main challenge in mega-DCs involves scaling compute processing, storage and interconnection capacity. In this context, two relevant architectural approaches are considered: the *scale-up* and *scale-out* [1].

According to the scale-up approach, computational intensive tasks are supported by large scale computing platforms (deploying high price servers and routers) offering very high computing power levels in a given system. This approach offers the required high computing power and storage levels in a given relatively simple system, but suffering limitations including increased cost, limited scalability, flexibility, density, availability and lack of modularity. The scale-out concept, on the other hand, accommodates the increasing needs for computational and storage resources in a much more flexible and efficient manner. According to this approach instead of relying on large scale monolithic devices, powerful computing systems are formed deploying a large number of low energy consuming and low cost devices. Connectivity between computing and storage devices is provided through a flat interconnection network collapsing together the Top-of-the-Rack (ToR) and aggregation switches, instead of being supported through a ToR switch in a multi-layer network. These switches are configured in different topologies (e.g. hypercubes, 2D/3D meshes, XD-torus etc.) that enable linear scalability to meet the increasing volume of demands and overcome the hierarchical tree-type network architecture limitations.

However, supporting scalability can be a challenge, due to the increased number of components and the associated control and management requirements. Software Defined Networking (SDN) decoupling the control from the data plane and moving it to a logically centralized controller with a holistic view of the network has been proposed as a key enabling technology [2]. To successfully apply SDN in these environments, novel solutions are needed to measure, predict and optimally respond to dynamically changing traffic workloads in a timely manner and overcome scalability constraints associated with SDN's centralized nature. Beyond a specific volume of collected information, network controllers are limited by insufficient capacity to handle incoming data and processing power to cope with a large number of decision variables and measurements, needed for the network management optimization processes. In response to this, the new trend in network science is to transform this type of optimization problems suffering high computational complexity to a "*practically solvable problem using correlation inferred from data rather than causality*" [3]. A typical example of such a process is presented in [4] where the DC placement problem is addressed by initially analyzing big data to identify possible correlations. Then network coordinate techniques are applied to reduce the size of the problems and identify the optimal matching between clients and servers.

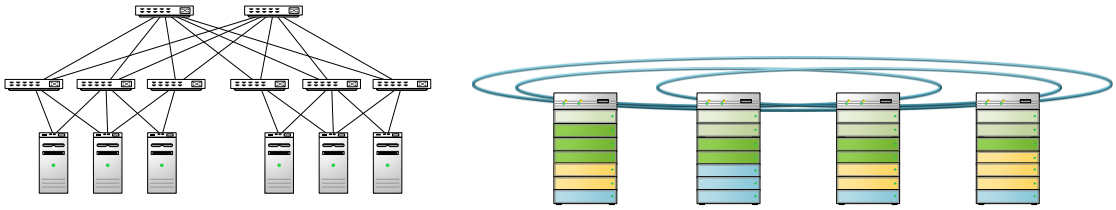
In this study, scalability in mega-DCs associated with monitoring and optimization of management data is addressed by adopting and combining for the first time graph factorization (GF) theory [5] with compressive sensing (CS) techniques [6]-[8], extending our previous work presented in [9]. Through GF, a mega-DC network graph is decomposed into a small number of simple graphs (factors). On top of these simple graphs and taking advantage of the spatial and temporal correlation of inter- and intra-DC traffic

Manuscript received: 26 Oct. 2015, Revised, 6 Jan. 2016.

M. P. Anastasopoulos and D. Simeonidou are with the Electrical & Electronic Engineering Department, University of Bristol, Clifton BS8 1UB, UK (e-mail: m.anastasopoulos, dimitra.simeonidou@bristol.ac.uk).

A. Tzanakaki is with the Electrical & Electronic Engineering Department, University of Bristol, Clifton BS8 1UB and the University of Athens, Department of Physics, Greece (e-mail: anna.tzanakaki@bristol.ac.uk).

A preliminary version of this paper was presented at the European Conference on Optical Communication (ECOC) 2015 [9].



a) b)

Figure 1: a) Traditional hierarchical DCN solution, b) Linear scale out approach with modular racks

characteristics [10]-[11][1], CS is applied to monitor various metrics e.g. resource utilization, using reduced control and management information (low sample number). Once this information is available at the system controller, the optimal resource allocation problem is solved in the compressed space, where the variables involved are significantly reduced (reducing computational complexity), using Integer Linear Programming (ILP). To the best of the authors' knowledge, this is the first time that CS and GF is adopted in cloud computing environments with the aim to analyze the optimal service provisioning problem. Modeling results indicate that applying the proposed approach the volume of information that reaches the controllers together with the number of variables that are involved in the optimization process can be drastically reduced.

The rest of the paper is organized as follows. In Sec. II, a brief description of the related work is provided. The problem description is given in Sec. III, while the proposed hybrid GF/CS joint network monitoring and optimization scheme is presented in Sec. IV. The performance of the proposed scheme is terms of scalability and accuracy is examined in Sec. V. Finally, Sec VI concludes the paper.

II. Related work

A. State of the art in DC network architectures

DCs have become a key element in supporting the new and emerging ubiquitous Internet-based applications and cloud services. Hundreds of thousands of servers are hosted in large-scale DCs, where huge amounts of data (TeraBytes/PetaBytes [12]) are maintained and processed. DC providers have observed an over 70% annual increase in the DC traffic volume [13] and this ever-growing traffic demand is expected to stretch the DC infrastructure requirements. Moreover, in Europe the electricity consumption of DCs is approaching 60TWh at present and is projected to reach 104TWh by 2020 [14]. Therefore, the design and development of future DC infrastructures has attracted significant attention both from academia and industry.

Based on the type of services supported and the available equipment information exchange, various intra-DC communication architectures have been proposed to date. These architectures are organized into three major classes based primarily on network topology. These include direct networks (also known as server-only), indirect networks (switch-only), and hybrid networks (hybrid server and switch DC) architectures [15]. Direct network architectures comprise a set of nodes (e.g., servers), each one being directly connected to other nodes. In these architectures, each server apart from executing regular applications, it also participates in packet relaying [16]. Although significant work has focused on analyzing the performance of various server-only interconnection architectures, only a limited subset of these have been actually implemented. Most of the implemented networks use an orthogonal topology in which the servers are arranged in an *n-dimensional* space. Orthogonal topologies are further classified into strictly orthogonal and weakly orthogonal [17]. The main advantage of the direct architectures is that they scale very well to a large number of servers. However, they suffer the following limitations: a) they require significant processing resources for packet forwarding, and b) servers are interconnected using a large number of links and network interface cards.

In indirect or switch-based networks on the other hand, connectivity between any two nodes is carried out through switches. Multiple layers of switches are then interconnected forming a hierarchical networking model. Switches may be organized either using simple tree topologies [18] (usually two-tier or three-tier [19]) or interconnected in a more sophisticated manner e.g. using fat trees [20], [21]. The hierarchical model consists of the core, the aggregation and the access layers. Typically, the access layer consists of 20-40 servers per rack, each connected to a ToR switch through a 1 or 10Gbps link. Other switching solutions for server networking today include end-of-row (EoR) as well as integrated switching. Connectivity between layers is achieved using the IEEE 802.1Q family of Ethernet protocols that enables synchronization of physical and virtual network configurations. It is reported in the literature [21], that this type of DC architectures suffers: a) limited DC-to-DC capacity, b) fragmentation of resources, c) poor reliability and utilization, and d) high latency. These limitations could be overcome by the use a single large scale N × N switch, however, the cost of such switch is still prohibitive for large DCs. Figure 1 illustrates the hierarchical scale-up and the flat distributed scale-out DC architectures.

Assessing the benefits and limitations of these solutions in the present study the hybrid switch-server approach ([22], [23]) is adopted in a flexible and dynamic fashion. As such the proposed DC network relies on interconnecting compute and

[11] As discussed in [10], Web services, email, video and messaging present correlation patterns in terms of the interplay of data between different services. These include the use of a common data set or exchange of information produced by the interaction with the user. Furthermore, DC traffic exhibits diurnal and clear weekend/weekday variation [11].

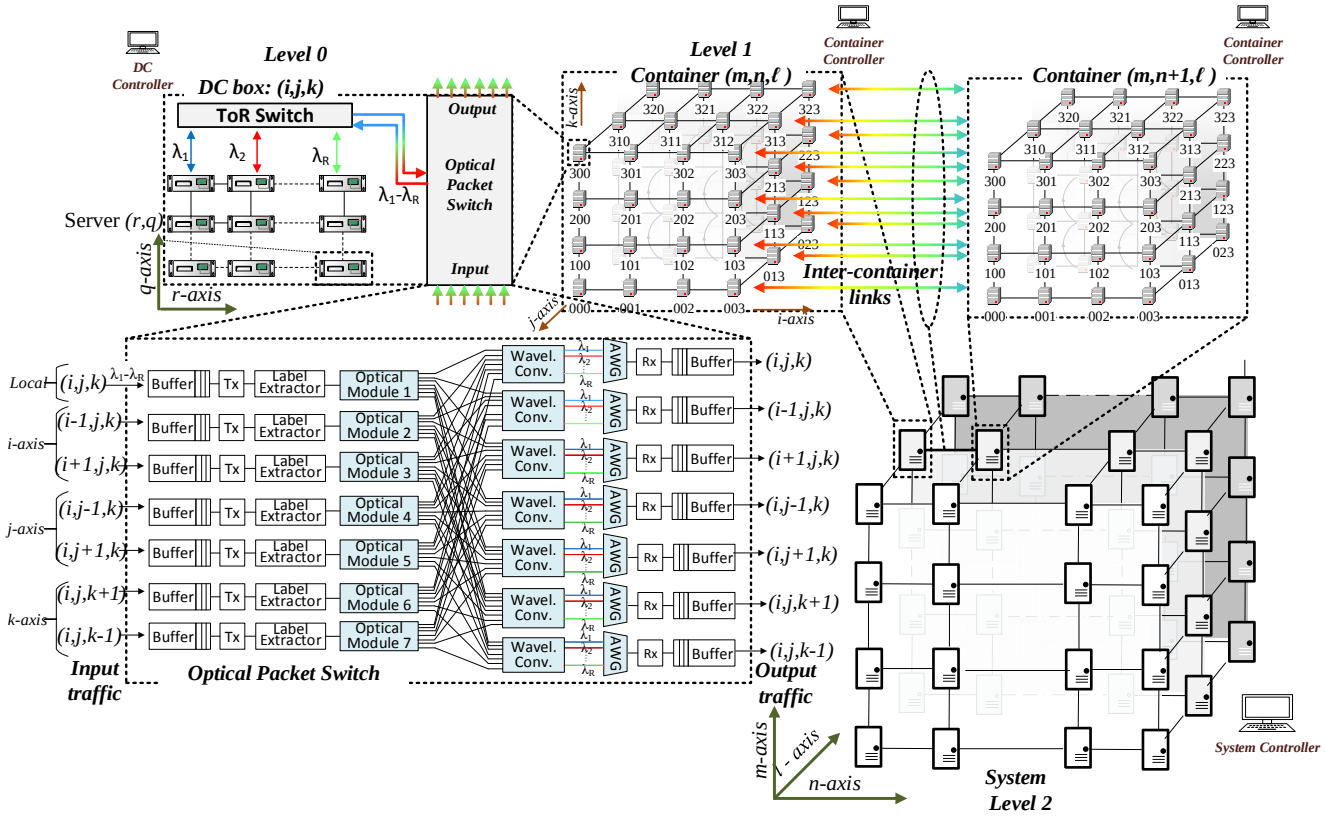


Figure 2: Example of a hierarchical mega DC network architecture with 2d/3d mesh connectivity. Level 0: Small scale servers are combined to form DC boxes. Connectivity between DC boxes is achieved through an 7x7 OPS. Level 1: DC boxes are grouped to form containers and finally, several containers are combined to form the mega-DC (level 2).

storage modules through a combination of server-to-server and distributed switch type of connectivity based on optical packets switches (OPS) [24]. This solution can provide significant benefits in terms of scalability, resource and energy efficiency and effectively improved system performance in terms of metrics such as latency.

B. *State of the art in Compressive Sensing*

So far, CS has been successfully applied to solve a variety of problems ranging from data gathering in multi-hop wireless sensor networks (WSNs) (see e.g., [25]-[27]) and network traffic estimation ([28]-[29]) to network tomography [30]. For example, in [25] the authors investigated the performance in terms of capacity and delay of data aggregation employing CS for a scenario where n sensor nodes are randomly deployed in a region. In [26], the authors applied CS in data collection to "*efficiently reduce communication cost and prolong network lifetime for large scale monitoring sensor networks*". [27] addressed the data aggregation problem in WSNs by jointly considering routing and CS to transport random projections of the monitored data, whereas in [28], [29] the authors proposed optimization approaches to estimate the normal and anomalous traffic, using a small subset of measurements. Finally, in [30] the authors formulate the minimum path selection problem that aims at estimating link delays using a small number of end-to-end delay measurements.

Despite its great potential, efficient implementation of CS in mega-DC environments can be quite challenging as for large number of components, the storage space requirements for the measurement matrix and the computational cost required to recover the original information is high. To address these issues, we adopt GF and combine it with CS in order to decompose the original problem into a set of separable sub-problems with reduced computational complexity. To the best of the authors' knowledge this is the first time that CS is combined with GF to address scalability issues in mega-DC infrastructures.

III. Problem Description

A multi-tier mega-DC network where computing modules are interconnected, based on a hybrid switch-server approach is considered (Figure 2). At the lower level (level 0), a DC box system comprising servers and network switches is used. DC boxes are also equipped with local controllers. The various modules of the DC box system are interconnected forming a 2d mesh topology combining server-to-server and distributed switching connectivity. To achieve low latency, high bandwidth and energy efficient connectivity between compute modules, the DC boxes deploy an OPS solution based on [24], [36]. Each non-blocking optical packet switch comprises 7 input and 7 output ports, and supports *R* wavelengths per port. In the next level (level 1), the various DC boxes are grouped in a 3d mesh topology to form containers. Connectivity between neighboring DC boxes is achieved through the OPSs. Across each dimension, two ports of the OPS are used for the ingress and two ports for the egress traffic, respectively, whereas one port is used to provide local connectivity between

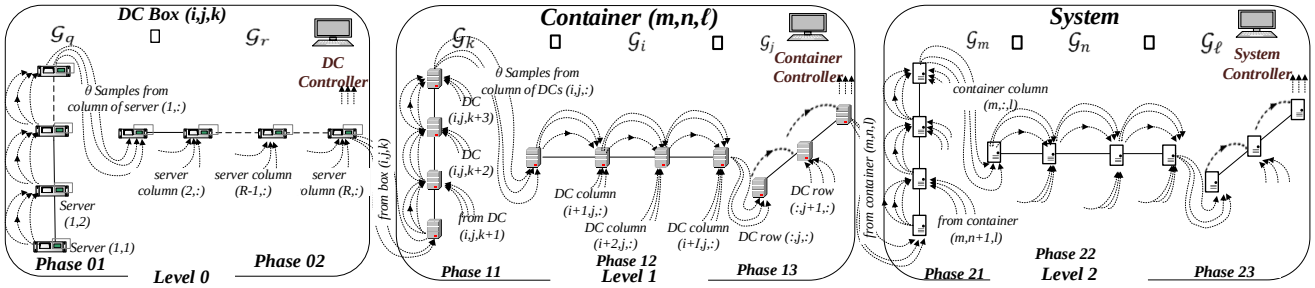


Figure 3: CS-based network monitoring with graph factorization

servers. Therefore, depending on its position every DC box connects with a number of neighboring DC boxes that varies between 3 and 6. It is clear that DC boxes located at the sides of each container will be connected with up to 5 neighboring nodes thus, leaving some ports of the OPS switches unused. However, these ports can be used, in the final stage, to interconnect the containers in a 3d mesh manner and form the mega-DC system (level 2).

In mega-DCs, optimal resource allocation aims at determining in a timely manner the network and computational resources required to satisfy a set of demands $d \in D$ with volume h_d. Traditionally, resource allocation problems in DCs are solved by centralized controllers applying an overall optimization criterion through ILP and Mixed ILP techniques. Although ILP-based optimization schemes can be easily formulated and implemented, they suffer disadvantages such as: i) requirement of full and accurate information of all parameters involved, ii) exponential scaling of computational complexity with the network size, making it unsuitable for mega-DCs. To cope with the increasing computational complexity inherent in ILP formulations, dimensionality reduction based on Lagrangian Relaxation [30], clustering [31] and heuristic techniques [32] have been proposed. In the present study, a different approach is adopted and a hybrid GF/CS scheme is employed to reduce the global amount of traffic transferred from the data to the control plane and the number of variables involved in the optimization process.

IV. Mathematical Modeling Preliminaries

Let $\boldsymbol{x}$ be an $N\times 1$ dimensional signal vector with elements $x_i, i=1,2,...,N$. Any signal $\boldsymbol{x}$ can be represented using as a basis a $N\times N$ matrix $\boldsymbol{\psi}$ with elements ψ_{ij}, as follows [34]:

$$\boldsymbol{x}=\boldsymbol{\psi}\boldsymbol{s} \Leftrightarrow \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \psi_{1N} & \cdots & \psi_{1N} \\ \vdots & \ddots & \vdots \\ \psi_{N1} & \cdots & \psi_{NN} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix} \quad (1)$$

where $\boldsymbol{s}=\left[s_1,...,s_N\right]^{\mathrm{T}}$ is vector with weighting coefficients s_i. Signal $\boldsymbol{x}$ is said to be K-sparse in domain $\boldsymbol{\psi}$ if in (1) there are K non-zero elements in vector $\boldsymbol{s}$. CS theory states that the K-sparse vector $\boldsymbol{x}$ can be efficiently reconstructed based on a set of M measurement, captured through the vector $\boldsymbol{y}=\left[y_1,...,y_M\right]^{\mathrm{T}}$ with $M \ll N$, using an $M\times N$ random measurement matrix $\boldsymbol{\varphi}$. Mathematically, this process can be written in the following form:

$$\boldsymbol{y}=\boldsymbol{\varphi}\boldsymbol{x} \Leftrightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \varphi_{1N} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots \\ \varphi_{M1} & \cdots & \varphi_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad (2)$$

At this point it should be noted that the minimum number of measurements M required to reconstruct the K-sparse signal $\boldsymbol{x}$ is given by [35]:

$$M \geq c\mu^2\left(\boldsymbol{\varphi},\boldsymbol{\psi}\right)K\,log\left(N\right) \quad (3)$$

where c is a positive constant number and $\mu\left(\boldsymbol{\varphi},\boldsymbol{\psi}\right)$, known as *mutual coherence*, is the largest correlation between any two elements of $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$. The mutual coherence is bound by $1\leq\mu\leq\sqrt{N}$ [35]. Once the set of measurements $\boldsymbol{y}$ has been collected, the original signal $\boldsymbol{x}$ can be recovered solving the following ℓ_1 - minimization problem:

$$\min_{\boldsymbol{s}\in\mathbb{R}^N}\left\|\boldsymbol{s}\right\|_{\ell_1} \quad \text{subject to} \quad \boldsymbol{y}=\boldsymbol{\varphi}\boldsymbol{x},\ \boldsymbol{x}=\boldsymbol{\psi}\boldsymbol{s} \quad (4)$$

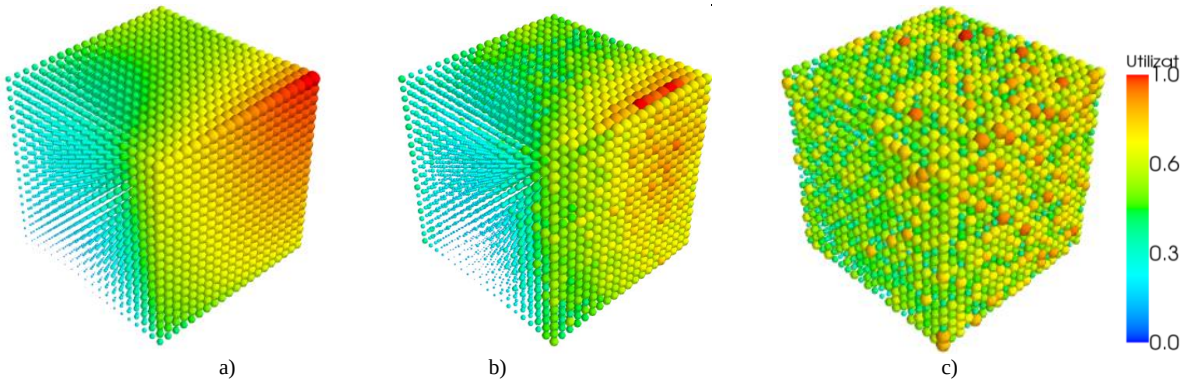
The output of (4), namely $\hat{s}$, is then used as input to (1) in order to reconstruct an approximation $\hat{\boldsymbol{x}}$ of the original signal. At this point, it should be mentioned that a necessary condition for (4) to efficiently reconstruct the original signal is the matrix $\boldsymbol{\theta}=\boldsymbol{\varphi}\boldsymbol{\psi}$ to satisfy the *restricted isometry property* (RIP) [35]. As discussed in [35], this can be achieved with high probability simply by selecting the elements of $\boldsymbol{\varphi}$ at random.

V. Hybrid GF/CS-based service provisioning

The hybrid GF/CS scheme can achieve improved scalability if spatio-temporal correlated performance DC-related metrics are transported to the system controller over the factorized graphs and are processed jointly. The joint monitoring and network optimization framework comprises the following steps:

1) Network topology decomposition:

The mega-DC network topology is decomposed into multiple simple graphs based on GF theory. Assuming that a DC box is modelled as an undirected 2D mesh graph, $\mathcal{G}_{Box}$, with size $Q\times R$, $\mathcal{G}_{Box}$ can be rewritten in a decomposed form as $\mathcal{G}_{Box}=\mathcal{G}_q\square\mathcal{G}_r$ where $\mathcal{G}_q$, $\mathcal{G}_r$ are simple linearly connected



a) b) c)

Figure 4: Numerical example: a) Actual average utilization per container, Reconstruction of the original data with 8% samples using b) Compressive Sensing (CS) with error 11%, c) Least Squares Error (LSE) analysis (System with 20^3 containers) with error 25%

graphs comprising Q and R server nodes, respectively and $\square$ denotes the Cartesian product operator. Each container with connectivity graph $\mathcal{G}_{Con}$ can be decomposed into a set of sub-graphs. Assuming that $\mathcal{G}_{Con}$ follows a 3D mesh pattern can be written as $\mathcal{G}_{Con}=\mathcal{G}_k \square \mathcal{G}_i \square \mathcal{G}_j$ where $\mathcal{G}_k$, $\mathcal{G}_i$ and $\mathcal{G}_j$ are linear graphs with K, I and J nodes respectively (Figure 3). The same rationale can be extended at the system level where the 3D mesh graph of the system, $\mathcal{G}_{sys}$, can be expressed as: $\mathcal{G}_{sys}=\mathcal{G}_m \square \mathcal{G}_n \square \mathcal{G}_\ell$ where $\mathcal{G}_m$, $\mathcal{G}_n$ and $\mathcal{G}_\ell$ are simple line graphs with M, N and $\mathcal{L}$ containers, respectively.

2) Network parameters compression

The details of the DC are then abstracted and transmitted to the system controller. Starting from level 0 (phase 01 in Figure 3), each server with coordinates (r,q) $(r=1,\ldots,R,\ q=1,\ldots,Q)$ multiplies the parameters of interest, say u_{rq}, with a random coefficient e.g., $q_{\vartheta,(rq)}$, and transmits the product to its adjacent node. In Figure 3, server $(1,1)$ transmits the products $q_{1,(11)}u_{11}$ up to $q_{\Theta,(11)}u_{11}$ containing a set of measurements Θ to its adjacent server $(1,2)$. Once server $(1,2)$ has received these messages, it calculates the random products $q_{\vartheta,(12)}u_{12}$, $\vartheta=1,2,...,\Theta$ and sends the weighted averages of the measurements generated at server $(1,1)$ and $(1,2)$, $q_{\vartheta,(11)}u_{11}+q_{\vartheta,(12)}u_{12}$ to server $(1,3)$. Each intermediate server $(1,q)$ adds to the incoming messages its product $q_{\vartheta,(1q)}u_{1q}$ and forwards the weighted average to the next server. Through this process, the top servers in each graph $\mathcal{G}_q$ will receive Θ packets containing the weighted averages of the random measurements performed by all servers in each $\mathcal{G}_q$. This is given by:

$$y_{\vartheta r}=\sum_{q=1}^{Q} q_{\vartheta,(rq)}u_{rq}=\boldsymbol{q}_{\vartheta,(r,:)}\boldsymbol{u}_r^{\mathrm{T}},\ r\in R,\vartheta\in\Theta \quad (5)$$

where $\boldsymbol{q}_{\vartheta,(r,:)}=\left[q_{\vartheta,(r1)},...,q_{\vartheta,(rQ)}\right]$, $\boldsymbol{u}_r=\left[u_{r1},...,u_{rQ}\right]$ and $[\cdot]^{\mathrm{T}}$ is the transpose operator. Now let $\boldsymbol{\Phi}_q$ be the random measurement matrix over the factorized graphs $\mathcal{G}_q$ defined through:

$$\boldsymbol{\Phi}_q=\begin{bmatrix}\boldsymbol{q}_{1,(1,:)} & \cdots & \boldsymbol{q}_{1,(R,:)}\\ \vdots & \ddots & \vdots\\ \boldsymbol{q}_{\Theta,(1,:)} & \cdots & \boldsymbol{q}_{\Theta,(R,:)}\end{bmatrix} \quad (6)$$

Based on (6), equation (5) can be written in compact form as follows:

$$\boldsymbol{y}=\boldsymbol{\Phi}_q\boldsymbol{u}^{\mathrm{T}} \quad (7)$$

where $\boldsymbol{u}=\left[\boldsymbol{u}_1^{\mathrm{T}},\boldsymbol{u}_2^{\mathrm{T}},\ \ldots,\boldsymbol{u}_R^{\mathrm{T}}\right]$ stacks the original set of measurement into a vector, and $\boldsymbol{y}=\left[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_\Theta\right]^{\mathrm{T}}$ is a column vector with elements $\boldsymbol{y}_\vartheta=\left[y_{\vartheta 1},\ldots,y_{\vartheta r}\right]^{\mathrm{T}}$. In phase 02, the data collected from all top servers are multiplied with the random coefficients $\varrho_{\vartheta'\vartheta r}, r\in R,\vartheta\in\Theta,\vartheta'\in\Theta'(\Theta'\prec\Theta)$ and relayed across the decomposed graph $\mathcal{G}_r$. The last server in graph $\mathcal{G}_r$ (server (R,Q)) of DC box (i,j,k)) will get ϑ' packets containing weighted averages of all random measurements for all servers within each DC box, that is:

$$\boldsymbol{z}_{\vartheta',(ijk)}=\sum_{\vartheta=1}^{\Theta}\boldsymbol{\varrho}_{\vartheta',(\vartheta,:)}\boldsymbol{y}_\vartheta=\boldsymbol{\varrho}_{\vartheta'}\boldsymbol{y}^T,\ \vartheta'\in\Theta' \quad (8)$$

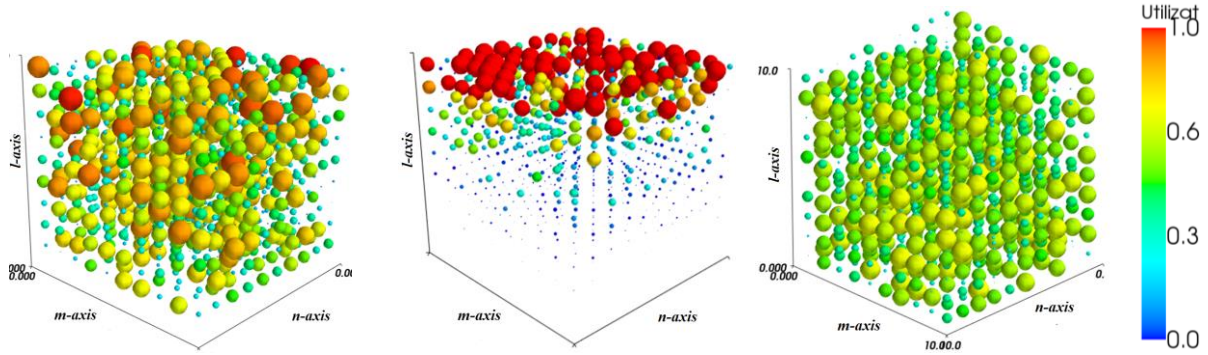
where $\boldsymbol{\varrho}_{\vartheta',(\vartheta,:)}=\left[\varrho_{\vartheta',(\vartheta 1)},...,\varrho_{\vartheta',(\vartheta R)}\right]$, $\boldsymbol{\varrho}_{\vartheta'}=\left[\boldsymbol{\varrho}_{\vartheta',(\vartheta,:)},...,\boldsymbol{\varrho}_{\vartheta',(\Theta,:)}\right]$. Assuming that $\boldsymbol{\Phi}_\varrho$ is the random measurement matrix over the factorized graphs $\mathcal{G}_r$, defined through:

$$\boldsymbol{\Phi}_\varrho=\begin{bmatrix}\boldsymbol{\varrho}_{1,(1,:)} & \cdots & \boldsymbol{\varrho}_{1,(\Theta,:)}\\ \vdots & \ddots & \vdots\\ \boldsymbol{\varrho}_{\Theta',(1,:)} & \cdots & \boldsymbol{\varrho}_{\Theta',,(\Theta,:)}\end{bmatrix} \quad (9)$$

then, (8) can be written as follows:

$$\boldsymbol{z}_{ijk}=\boldsymbol{\Phi}_\varrho\boldsymbol{y}=\boldsymbol{\Phi}_\varrho\boldsymbol{\Phi}_q\boldsymbol{u}^{\mathrm{T}} \quad (10)$$

where $\boldsymbol{z}_{ijk}$ is an $1\times\Theta'$ column vector defined as $\boldsymbol{z}_{ijk}=\left[z_{1,(ijk)},...,z_{\Theta',(ijk)}\right]^{\mathrm{T}}$. Packets $\boldsymbol{z}_{ijk}$ will be then used as input to level 1 containers. Within each container, these



a) b) …. c)

Figure 5: Snapshot of the average utilization per container (system with 10^3 containers) a) before system optimization, b) after optimizing for maximum performance per watt per space, b) after optimizing for load balancing (the axes represent the coordinates of the containers)

packets will be relayed across the graphs $\mathcal{G}_k$, $\mathcal{G}_i$ and $\mathcal{G}_j$. The output of the last DC box in $\mathcal{G}_j$ (phase 13 in Figure 3) of container (m,n,ℓ), namely $\boldsymbol{w}_{mn\ell}=\left[\boldsymbol{w}_{1,(mn\ell)},...,\boldsymbol{w}_{\Theta,(mn\ell)}\right]^{\mathrm{T}}$, will be used as input to level 2. Following the same rationale $\boldsymbol{w}_{mn\ell}$ will be equal to:

$$\boldsymbol{w}_{mn\ell}=\left(\boldsymbol{\Phi}_{\varkappa}\boldsymbol{\Phi}_i\boldsymbol{\Phi}_j\right)\boldsymbol{z}^T \quad (11)$$

where $\boldsymbol{\Phi}_{\varkappa}$, $\boldsymbol{\Phi}_i$, $\boldsymbol{\Phi}_j$ are the sampling matrices across $\mathcal{G}_k$, $\mathcal{G}_i$ and $\mathcal{G}_j$, respectively, and $\boldsymbol{z}$ is a vector that stacks all DC box measurements. The same process is repeated for all containers, until the collected information, namely $\boldsymbol{g}$, reaches the system controller. $\boldsymbol{g}$ can be estimated through the following equation:

$$\boldsymbol{g}=\left(\boldsymbol{\Phi}_m\boldsymbol{\Phi}_n\boldsymbol{\Phi}_\ell\right)\boldsymbol{w}^T \quad (12)$$

where $\boldsymbol{w}$ is a stacked vector with elements $\boldsymbol{w}_{mn\ell}$.

3) Reconstruction of the information at the system controller

Once the abstracted information $\boldsymbol{g}$ reaches the system controller, the optimal inter-container resource allocation strategies need to be identified. To achieve this, the compressed parameters $\boldsymbol{g}$, together with the random coefficients $\boldsymbol{\Phi}_m$, $\boldsymbol{\Phi}_n$ and $\boldsymbol{\Phi}_\ell$ are used to recover vector $\boldsymbol{w}$, the elements of which contain information on the utilization of each container. Now let $\boldsymbol{\Psi}_i$ be a compressibility basis for $\boldsymbol{w}$ with $\boldsymbol{\Psi}_i, i=\{m,n,\ell\}$, being a basis for the graph $\mathcal{G}_i$. $\boldsymbol{w}$ can be recovered solving a set of ℓ_1- minimization problems using the following steps:

i) In the first step (phase 23), the elements $\boldsymbol{w}_\ell=\left[w_1,...,w_{\mathcal{L}}\right]$ of graph $\mathcal{G}_\ell$ are recovered from $\boldsymbol{g}$ through the solution of the following problem:

$$\min_{s\in\mathbb{R}^{\Theta\times\mathcal{L}}}\|\boldsymbol{s}\|_{\ell_1} \quad \text{subject to} \quad \boldsymbol{g}=\boldsymbol{\Phi}_\ell\boldsymbol{w}_\ell,\ \boldsymbol{w}_\ell=\boldsymbol{\Psi}_\ell\boldsymbol{s} \quad (13)$$

ii) Once $\boldsymbol{w}_\ell$ has been estimated, in the second step (phase 22), the elements $\boldsymbol{w}_{n\ell}$ of graph $\mathcal{G}_n$ that is connected with the element ℓ of $\mathcal{G}_\ell$ are estimated through:

$$\min_{s\in\mathbb{R}^{\Theta\times N}}\|\boldsymbol{s}\|_{\ell_1} \quad \text{subject to} \quad \boldsymbol{w}_\ell=\boldsymbol{\Phi}_n\boldsymbol{w}_{n\ell},\ \boldsymbol{w}_{n\ell}=\boldsymbol{\Psi}_n\boldsymbol{s},\ \ell\in\mathcal{L} \quad (14)$$

iii) In the final step, $\boldsymbol{w}_{mn\ell}$ is recovered by the solution of the following problem:

$$\min_{s\in\mathbb{R}^{\Theta\times M}}\|\boldsymbol{s}\|_{\ell_1} \text{ subject to } \boldsymbol{w}_{n\ell}=\boldsymbol{\Phi}_m\boldsymbol{w}_{mn\ell},\ \boldsymbol{w}_{mn\ell}=\boldsymbol{\Psi}_m\boldsymbol{s}, n\in N,\ell\in\mathcal{L} \quad (15)$$

Problems (13)-(15) can be solved in polynomial time over the factorized graphs using interior point methods that have $\mathcal{O}\left(N'^3\right)$ computational complexity, where N' is the number of components that need to be monitored. For example, the complexity for recovering information per container at the system level without GF is $\mathcal{O}\left(N^3\times M^3\times\mathcal{L}^3\right)$. However, when GF is adopted, the original problem is decomposed into a set of much smaller sub-problems (with size equal to the size of the factorized graphs) leading to significant computational complexity reduction.

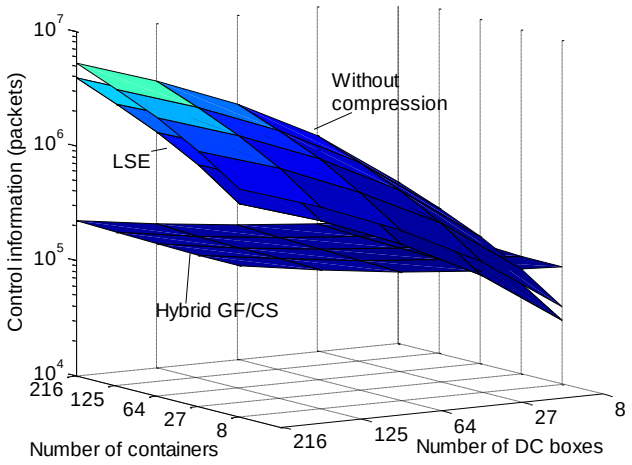


Figure 6: Volume of information vs number of servers for the following schemes: without compression, with statistical sampling and information reconstruction based on LSE and, with hybrid GF/CS (Optimization error 8%)

4) Optimization in the compressed space

In the following step, the recovered information $w_{mn\ell}$ indicating the average actual usage of all servers and DC boxes belonging to (m,n,ℓ) is used to formulate an optimization problem in the compressed space. Through aggregation of the resources' details, the number of parameters and decision variables involved in the optimization phase can be drastically reduced. Now let d $(d \in D)$ be the demands that need to be allocated to containers (m,n,ℓ) $m \in M, n \in N, \ell \in \mathcal{L}$. The volume of the traffic demand d is denoted by h_d. The objective is to identify optimal resource allocation strategies maximizing performance per space:

$$\min \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}} \Delta_{mn\ell}\mathcal{A}_{mn\ell}\left(\mathcal{W}_{mnl} - w_{mn\ell} - w'_{mn\ell}\right)^3 \quad (16.1)$$

Subject to

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}}\sum_{q=1}^{Q_{mn\ell}} \alpha_{d,(mn\ell)} z_{dq} = h_d, d \in D \quad (16.2)$$

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}} \alpha_{d,(mn\ell)} = 1, d \in D \quad (16.3)$$

$$\sum_{d=1}^{D}\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}}\sum_{q=1}^{Q_{mn\ell}} \delta_{dqe} z_{dq} \le \mathcal{C}_e \quad (16.4)$$

$$\sum_{d=1}^{D} \alpha_{d,(mn\ell)} h_d \le w'_{mn\ell},\ m \in M, n \in N, \ell \in \mathcal{L} \quad (16.5)$$

$$w'_{mn\ell} \le \mathcal{A}_{mn\ell}\left(\mathcal{W}_{mnl} - w_{mn\ell}\right)\ m \in M, n \in N, \ell \in \mathcal{L} \quad (16.6)$$

where $q \in Q_{mn\ell}$ is the candidate path list at the system required to support demand d at container (m,n,ℓ). This can be pre-computed using the k-shortest path algorithm. $\Delta_{mn\ell}$ denotes the distance in terms of number of hops of (m,n,ℓ) from the system controller, z_{dq} is the capacity allocated to path q for demand d, $\mathcal{C}_e$ is the link e capacity. $\alpha_{d,(mn\ell)}$ is a binary coefficient taking value equal to 1 if demand d is assigned to container (m,n,ℓ). δ_{dqe} is a binary coefficient that equals 1 if link e belongs to path q realizing demand d at container (m,n,ℓ), $\mathcal{W}_{mnl}$ is the capacity of container (m,n,ℓ) and $\mathcal{A}_{mn\ell}$ is a binary parameter taking values equal to 1 when container (m,n,ℓ) is active $\left(\sum_{d=1}^{D}\alpha_{d,(mn\ell)} \ge 1\right)$; 0 otherwise.

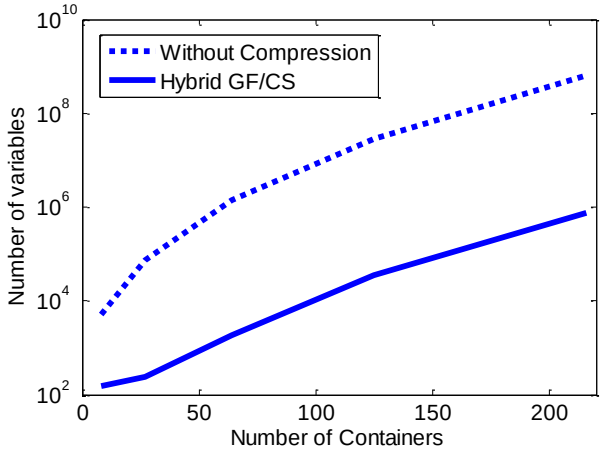


Figure 7: Number of decision variables as a function of the number of DCs with and without GF/CS.

In the above formulation, constraint (16.2) (known as the demand constraints) assures that the volume h_d of demand d will be realized through flows z_{dq} at the container (m,n,ℓ). (16.3) assures that each demand will be assigned at a single container whereas (16.4) denotes the network capacity constraints. The necessary processing capacity $w'_{mn\ell}$ required to support demands d at container (m,n,ℓ) is captured through (16.5). Finally, the available capacity at each container $(\mathcal{W}_{mnl} - w_{mn\ell})$ should be adequate to support the requested services (16.6). An interesting observation is that for the objective function a cubic deviation cost has been adopted that aims at maximizing performance per watt per space by packing as many demands as possible at a single container. To achieve this, in case where a container is active, $w'_{mn\ell}$ takes values very close to the available capacity $\mathcal{W}_{mnl} - w_{mn\ell}$, minimizing the deviation cost $\left(\mathcal{W}_{mnl} - w_{mn\ell} - w'_{mn\ell}\right)^3$. Note that the cubic cost adopted in the objective function aims at magnifying the penalty that is introduced when a container is underutilized. This gives an incentive to the system to transfer tasks from low to high utilized containers and introduces a high penalty when containers remain underutilized.

5) Optimal Intra-container resource allocation

Once the containers, where the demands are processed, have been defined, the optimal intra-container routing strategies are determined at each container controller. For each container, the compressed parameters $w_{mn\ell}$ together with the random

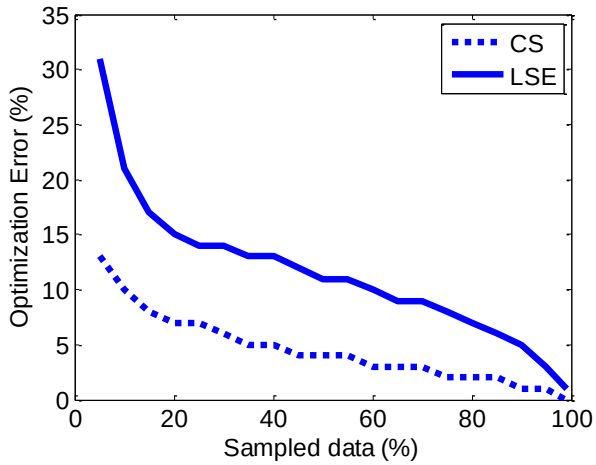


Figure 8: Impact of level of compression on the accuracy of the obtained results when the original information is reconstructed using CS and LSE.

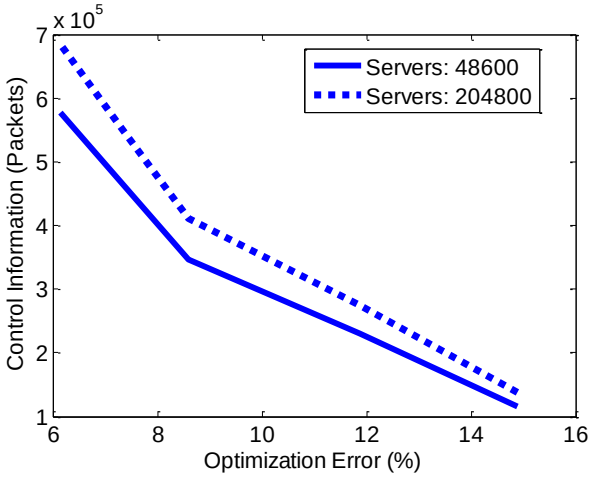


Figure 9: Volume of information reaching the controller under different levels of optimization error and number of servers for the proposed hybrid GF/CS scheme.

coefficients $\mathbf{\Phi}_{\mathcal{K}}$, $\mathbf{\Phi}_i$, $\mathbf{\Phi}_j$ are used to recover vector $\boldsymbol{z}$, the elements of which contain information on the utilization of the DC boxes. Following the derivation of $\boldsymbol{z}$, an optimization problem similar to that presented in (16) is formulated at a container level that determines the DC boxes where demands are processed. Based on $\boldsymbol{z}_{ijk}$, a set of ℓ_1- minimization problems at a DC box level are formulated estimating the utilization per server within the DC boxes (u_{rq}). Based on u_{rq} and solving a problem similar to (1)-(5) the optimal demand allocation at a server level is determined.

VI. Numerical Results

The hybrid GF/CS optimization approach is evaluated for the topology of Figure 2 with 20x10 servers per DC box and cubic sized containers (where K=I=J). Both the system controller and the network controllers are placed at the top side of the 3d mesh topologies. Traffic statistics have been generated by appropriately modifying [37] assuming 58.88% usage for inter-container links, 73.77% for intra-container and 57.52% for server-to-server communication links. The size of packets generated follow a bimodal distribution with peaks around 40B and 1500B and an average packet size of 850B. Generated traffic exhibits an ON/OFF pattern with duration of the ON/OFF period following the lognormal distribution. The packet inter-arrival times within ON periods (in milliseconds scale) follow the lognormal distribution with parameters (6.14, 1.56). The same also holds for the length of OFF-periods and ON-periods that follow the lognormal distribution with parameters (10.29, 0.39) and (2.55, 0.81), respectively. The performance of the proposed hybrid GF/CS scheme is compared to the following baseline approaches:

i) "*Without compression*": This corresponds to the case where information is gathered from all hardware elements in the system. This functionality is supported by the majority of the existing operating systems i.e., Junos OS 15.1 [38]

ii) "*LSE*": This corresponds to the case where statistical sampling is performed (one packet is randomly selected in an interval of n packets i.e., CISCO NetFlow [39]). The original information is then reconstructed at the SDN controller using regression analysis techniques, such as, Least Squares Error (analysis) [40]. In both schemes, once information has been collected network optimization is performed.

Initially, the efficiency of the proposed hybrid GF/CS information reconstruction scheme is examined. In Figure 4 (a), a snapshot of the actual average utilization per container is provided for a system with 20^3 containers. Figure 4(b) shows that this information can be successfully reconstructed with error 11% using the proposed CS-based scheme even when a very low number of samples is used (8% samples). However, when the LSE analysis is applied over the same number of samples the reconstruction error is (Figure 4 (c)) in the order of 25%. It is also observed that the prevailing trend for the CS approach is to underestimate the utilization of the containers. This is explained by the low sampling rate (8%)) and the small number of highly utilized containers. On the other hand, in a scenario where a large number of highly utilized containers exists, an overestimation of underutilized containers is expected.

In the next step, once the necessary information has been retrieved, an optimization problem that tries to maximize performance per watt per space is solved at the system controller. A snapshot of the average utilization per container before applying the proposed optimization scheme is illustrated in Figure 5 (a). Once the system has been optimized for maximum performance per watt per space, it is seen that the majority of the containers have been switched off to save energy and tasks have been consolidated to a small number of highly utilized containers (Figure 5 (b)). It is also observed that containers located at the top of the system are almost fully utilized whereas containers located at the bottom are inactive. This is explained by the fact that the proposed objective function allocates tasks to containers that are located close to the system controller. Through this approach, the performance of the system can be improved through the reduction in the container-to-control plane delays. A different task allocation policy that is also examined tries to equally distribute tasks among containers (i.e., this can be achieved by maximizing the Jain's fairness index $\left(\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}} x_{mn\ell}\right)^2$ /

$\left(mnl\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{\ell=1}^{\mathcal{L}}x^{2}{}_{mn\ell}\right)$ with $x_{mn\ell}=\mathcal{W}_{mnl}-w_{mn\ell}-w'_{mn\ell}$ [41]). The output of this process using the same starting point is illustrated in Figure 5 (c) when it is seen that containers are almost equally utilized.

In Figure 6, we compare the performance of the proposed hybrid GF/CS scheme in terms of the volume of information that reaches the system's controller with the traditional scheme where compression is not applied and the statistical sampling scheme. Our results show that the hybrid GF/CS scheme drastically reduces the volume of control data compared to existing approaches, thus reducing the variables involved in the ILP formulation and the associated computational complexity. It is also observed that the volume of information increases almost linearly with the number of servers (Figure 7). When the proposed scheme is adopted, instead of all servers sending their status to the controller directly, information is multiplexed. This allows a constant number of packets, containing the weighted averages of the measurements performed, to be relayed across the factorized graphs. It is also observed that the benefits of the proposed hybrid GF/CS scheme increases with the size of the DC systems. However, for small scale DCs traditional schemes report lower amounts of packets compared to that for the GF/CS. As already mentioned, in order for the CS scheme to be effective, the number of measurements should be much lower than the number of monitored data. Hence, for small scale DCs, the number of packets that are relayed containing the weighted averages of the random measurements, is higher than the number of servers leading to suboptimal performance of the proposed scheme.

Figure 8 shows the impact of the number of samples on the optimization error when the original information is reconstructed using the CS and the LSE approach. The optimization error is defined as the gap between the result of each one approach (i.e., the hybrid GF/CS and the LSE) and the original information. As expected, for lower number of samples, the estimation error increases. This may lead to an overestimation or an underestimation of the available capacity per server causing suboptimal operation of the entire system. For example, underestimation of the actually used resources (i.e., underestimation of $w_{mn\ell}$ indicating the average usage per container) may lead to an inability for the system to satisfy resource requests (especially if the system operates close to its capacity limit i.e., $w_{mn\ell}$ takes values close to $\mathcal{W}_{mn\ell}$). Overestimation of the actually used resources on the other hand may lead to increased operational expenditures since additional servers will be activated to cover the same traffic demands. However, the CS scheme requires a much lower number of samples compared to the LSE scheme to achieve the same level of accuracy.

Finally, Figure 9 illustrates the volume of information that reaches the system controller under different levels of optimization error and number of servers for the proposed hybrid GF/CS scheme. As expected, system controllers that are able to handle higher volumes of control information can process more complex optimization tasks leading to improved system performance and lower levels of optimization error. Furthermore, it is observed that the proposed scheme is not affected by the increase in the DC size since a four factor growth in the number of servers increases the data volume by less than 20% with a 6% optimization error

VII. Conclusions

This paper focused on the design of service provisioning schemes suitable for mega-DC infrastructures. To address the scalability issues of these infrastructures, introduced by the increased number of resources available in these and the associated requirements for control and management information, we propose for the first time to combine graph factorization with the recently reported compressive sensing theories to monitor and optimize their operation. This approach takes advantage of the spatial and temporal correlation of compute, and network resource requests, to monitor and optimize metrics, such as server utilization with reduced control and management information. Our modelling results indicate drastically reduced amounts of traffic transferred from the data to control plane and number of optimization process variables.

Acknowledgements

The work was supported by the EPSRC grant EP/L020009/1: TOUCAN and the Horizon 2020 project IN2RAIL. The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] A. Vahdat et al., Scale-out networking in data center, *IEEE Micro*, vol. 30, no. 4, 29-41, 2010
[2] M. Channegowda et al., Software-defined optical networks technology and infrastructure, *J. Opt. Commun. Netw*., nol. 5, no. 10, 2013.
[3] H. Yin et al., "Big data: transforming the design philosophy of future internet," *IEEE Network*, vol.28, no.4, pp.14,19, July-August 2014.
[4] Y. Hao, Z. Xu, Z. Tongyu, Z. Ying, M. Geyong, D. O. Wu, "NetClust: A Framework for Scalable and Pareto-Optimal Media Server Placement," *IEEE Transactions on Multimedia*, vol.15, no.8, pp.2114-2124, Dec. 2013
[5] R. Hammack, Handbook of product graphs, CRC, 2011
[6] D. Donoho, Compressed Sensing, *IEEE Trans. on Information Theory*, vol. 52, no. 4, 2006.
[7] M. Anastasopoulos et al., Scalable Service Provisioning in Converged Optical/Wireless, *in proc. of OFC*, 2015
[8] H. Zheng et al., "Capacity and Delay Analysis for Data Gathering with CS in Wireless Sensor Networks", *IEEE Trans. Wirel. Commun.*, vol, 12, no. 2, 2013.
[9] M.P. Anastasopoulos, A. Tzanakaki, D. Simeonidou, "Enabling mega-DCs through scalable monitoring and optimization", *in proc. of ECOC* Sept. 27 2015-Oct. 1 2015
[10] C. Yingying; S. Jain, V.K Adhikari, Z. Zhi-Li; X. Kuai, "A first look at inter-data center traffic characteristics via Yahoo! datasets," *in proc. of INFOCOM, 2011*, pp.1620-1628, 10-15 April 2011
[11] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild", *in proc. of ACM IMC*, pp. 267-280, 2010.
[12] Pattern-Based Strategy: Getting Value From Big Data, June 2011.
[13] R. Villars, et al, "From IDC Predications 2013: The new data centre dynamic", Dec. 2012.
[14] [Online] DCD Industry census 2014: Data Centre Power. http://tinyurl.com/zgnah2f
[15] L. Popa et. al., "A cost comparison of datacenter network architectures'.", *in Proc. of Co-NEXT*, no. 16, 2010
[16] H. Abu-Libdeh et al., "Symbiotic Routing in Future Data Centers", *In proc. of ACM SIGCOMM*, 2010.
[17] W. Dally and B. Towles. 2003. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers

[18] Cisco Systems, “Data Center Design – IP Network Infrastructure,” http://www.cisco.com/en/ US/docs/solutions/Enterprise/Data Center/DC 3 0/DC-3 0 IPInfra.pdf, Oct. 2009
[19] http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_1.html
[20] M. Al-Fares et al., “A scalable, commodity data center network architecture,” in proc. of SIGCOMM, pp. 63–74, 2008
[21] A. Greenberg et al., VL2: a Scalable and Flexible Data Center Network,” in proc. of SIGCOMM, pp. 51–62, 2009
[22] C. Guo et al., “BCube: A High Performance, Server-centric Network Architecture for Modular DCs”, *in proc. of SIGCOMM*, vol. 39 no. 4, pp. 63-74, 2009.
[23] C. Guo et al., “Dcell: A Scalable and Fault-tolerant Network Structure for Data Centers”, *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 75-68, Oct. 2008
[24] S. Lucente et al., FPGA Controlled Integrated Optical Cross-Connect Module for High Port-Density Optical Packet Switch, *in proc. of ECOC* 2012, Tu.3.A.3.
[25] H. Zheng et al., “Capacity and Delay Analysis for Data Gathering with Compressive Sensing in Wireless Sensor Networks,” *IEEE Trans. Wireless Commun.* vol.12, no.2, pp.917-927, Feb. 2013
[26] C. Luo et al., “Efficient Measurement Generation and Pervasive Sparsity for Compressive Data Gathering,” *IEEE Trans. Wireless Commun.*, vol.9, no.12, pp.3728-3738, Dec. 2010
[27] G. Quer et al., “On the interplay between routing and signal representation for Compressive Sensing in wireless sensor networks," *In Proc. of ITA 2009*, pp.206-215, 8-13 Feb. 2009
[28] M. Mardani, G. B. Giannakis, “Robust network traffic estimation via sparsity and low rank,” *in Proc. of IEEE ICASSP*., pp.4529-4533, 2013
[29] Y. Zhang, M. Roughan, W. Willinger, L. Qiu. “Spatio-temporal compressive sensing and internet traffic matrices,” *SIGCOMM Comput. Commun. Rev.* Vol. 39, no. 4, 267-278, August 2009.
[30] M P. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Stochastic Planning of Dependable Virtual Infrastructures Over Optical Datacenter Networks," *J. Opt. Commun. Netw*. Vol. 5, pp. 968-979, 2013
[31] K. Chen et al., “Generic and automatic address configuration for data center networks”, *in proc. of SIGCOMM '10*, pp. 39-50, 2010
[32] X. Li, G. Shen, "Optimal Content Caching based on Content Popularity for Content Delivery Networks," *in proc. of APC* 2015, AS4G.4.
[33] M.H. Firooz, S. Roy, “Network Tomography via Compressed Sensing,” *in proc. of IEEE GLOBECOM 2010*, pp.1,5, 6-10 Dec. 2010
[34] R.G. Baraniuk, “Compressive Sensing [Lecture Notes],” *Signal IEEE Processing Magazine*, vol.24, no.4, pp.118,121, July 2007
[35] E. Candès and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Prob.* , vol. 23, no. 3, pp. 969–985, 2007.
[36] N. Calabretta et al.,, "Flow controlled scalable optical packet switch for low latency flat data center network," in *Proc. of ICTON, 2013* June 2013
[37] T. Benson et al., Understanding data center traffic characteristics, ACM *SIGCOMM CCR*, vol. 40, no. 1, pp. 92-99, Jan. 2010
[38] [Online] Configuring Traffic Sampling - Juniper Networks” http://tinyurl.com/zc7h7lj
[39] [Online] CISCO NetFlow Sampling, http://tinyurl.com/j9as2r6
[40] D. Longfei, Y. Wenguo, S Gao; X. Yinben, Z. Mingming, Zhigang Ji, "EMD-Based Multi-Model Prediction for Network Traffic in Software-Defined Networks," *in proc. of IEEE MASS*, pp.539-544, 28-30 Oct. 2014
[41] T. Taleb, N. Nasser, M.P. Anastasopoulos, "An Auction-Based Pareto-Optimal Strategy for Dynamic and Fair Allotment of Resources in Wireless Mobile Networks," *IEEE Trans. Vehicular Technology*, vol.60, no.9, pp.4587-4597, Nov. 2011