Dynamic Softwarised RAN Function Placement in Optical Data Centre Networks

Nikolaos Gkatzios, Markos Anastasopoulos, Anna Tzanakaki, Dimitra Simeonidou

Dynamic Softwarised RAN function placement in Optical Data Centre Networks

Nikolaos Gkatzios[1[0000-0002-0908-8913]], Markos Anastasopoulos[2[0000-0002-2726-4116]], Anna Tzanakaki[1,2], and Dimitra Simeonidou[2[0000-0002-7046-544X]]

[1] National and Kapodistrian University of Athens, Athens, Greece
[2] HPN Group, University of Bristol, Bristol, United Kingdom
ngkatzio@phys.uoa.gr

Abstract. The ability to assign functions that comprise the BBU processing in centralized softwarised Radio Access Networks into different servers the, have been proven to be beneficial, in terms of energy efficiency. This paper proposes a heuristic suitable for BBU functions allocation and evaluates the impact of dynamic resource management in these environments facilitated through Virtual Machine (VM) live migration. The benefits associated with VM migration are quantified through a series of experiments. Our results show notable improvement in terms of resource and energy efficiency.

Keywords: RAN, BBU, heuristic, compute disaggregation, VM migration.

1 Introduction

The increased bandwidth, connectivity and mobility requirements associated with 5G have led to the densification of wireless access technologies and the introduction of very stringent requirements in the Radio Access Networks (RANs). In these dense environments distributed RAN solutions, where Base Band Units (BBUs) and radio units (RUs) are co-located, suffer several limitations including increased capital and operational expenditures as well as CO2 footprint. To increase infrastructure efficiency and address these challenges Cloud Radio Access Networks (C-RANs) have been proposed. In C-RAN, a centralized approach is adopted according to which distributed Remote Radio Heads (RRHs) are connected to a BBU pool located at the Central Unit (CU). The CU can be hosted in Data Centres (DCs) comprising General Purpose Processors (GPPs) that can be shared efficiently across a set of RRHs. However, although this centralized architecture is more efficient compared to the distributed approach it imposes the need of a high bandwidth transport network to support interconnection of the RRHs with the CU known as fronthaul (FH) [1]. The interface between RUs and CU is standardized through the Common Public Radio Interface (CPRI).

When adopting the centralized architectural model proposed by C-RAN it is very important to identify the optimal allocation of functions comprising the BBU function chain to the appropriate servers hosted by the CU. The concept of compute resource disaggregation approach [2] allows individual allocation of these processing functions, associated with a specific FH service, to different servers depending on the nature and volume of their processing requirements.

In addition, dynamic access and efficient sharing of compute resource for BBU type of processing, through the adoption of Cloud Computing, takes advantage of the notion of virtualisation that has become key technology for Data Center (DC) resource management. In this context, virtualisation can assist in improving performance and reliability as well as operational costs reduction. Virtual Machine (VM) migration is one of the features provided through virtualization. Migration of operating system instances across different servers is a crucial tool, which is used to achieve different performance goals.

To quantify the benefits of live VM migration in a 5G environment employing centralized BBU processing, a heuristic that is able to assign different functions composing the BBU chain to an appropriate set of servers at the CU, was developed. A set of experiments where the developed heuristic was called to allocate the incoming FH traffic to suitable DC resources for the required processing were conducted. Each experiment assumed different initial DC loading conditions referring not only to the absolute processing load, but also to the load distribution to different servers within the DC. Our results show that when we are able to redistribute the load of the compute resources within the DC, considerable benefits can be achieved, in terms of resource as well as energy efficiency and therefore operational cost reduction.

2 Problem Statement

A generic 5G C-RAN is considered, where compute resources located at the CU support the processing requirements of a group of RRHs. The compute resources at the CU comprise a set General Purpose Processors (GPPs). These servers are arranged in a simple tree topology shown in **Fig. 1**. The required compute capacity to support FH service provisioning, is supplied by these servers. These employ an implementation of softwarised BBU processing that executes the baseband signal processing to support the operation of RRHs. The requirements, in terms of compute resources, for the baseband processing of an RU, can be estimated as the sum of all compute elements performing the BBU functions. These functions include: Single Carrier - Frequency Division Multiple Access (SC-FDMA) Demodulation, Subcarrier Demapping, Frequency Domain Equalizer, Transform Decoding, Constellation Demapper, Descrambling, Rate Matching and Turbo Decoding. It should be noted, that as shown in **Fig. 1**, these functions need to be performed in a specific order.

The main objectives of this work are:

- to identify, in real time, optimal allocation of the BBU functions to the DC servers in order to minimize the total power consumption at the DC, while, at the same time, complying with the stringent delay constrains imposed by the CPRI protocol.

- to investigate the benefits, in terms of power consumption, which arise from the live migration of VMs in a DC.

To achieve the first objective, we are taking into consideration the processing requirements, in terms of Instructions, for each baseband processing function [2]. Then, a heuristic algorithm, with low computation complexity, is developed that tries to match the BBU Service Chain functions to the best suited servers, in terms of power consumption, inside a DC.

For the second objective the first step is to quantify the benefits of live VM migration. The services that are already running at the DC are utilising a percentage of the total available resources. We will refer to these resources as initial DC load. The VMs that correspond to already existing services can either remain allocated to the original set of servers or can be reallocated to a set of different servers with the aim to improve a specific metric in our case utilisation. Utilisation here is defined as the ratio of the servers that are switched-on compared to the total number of servers available in the DC. A set of experiments that aim to allocate DC resources is conducted adopting this heuristic assuming different initial loading conditions and load distributions across the set of servers that are being used.

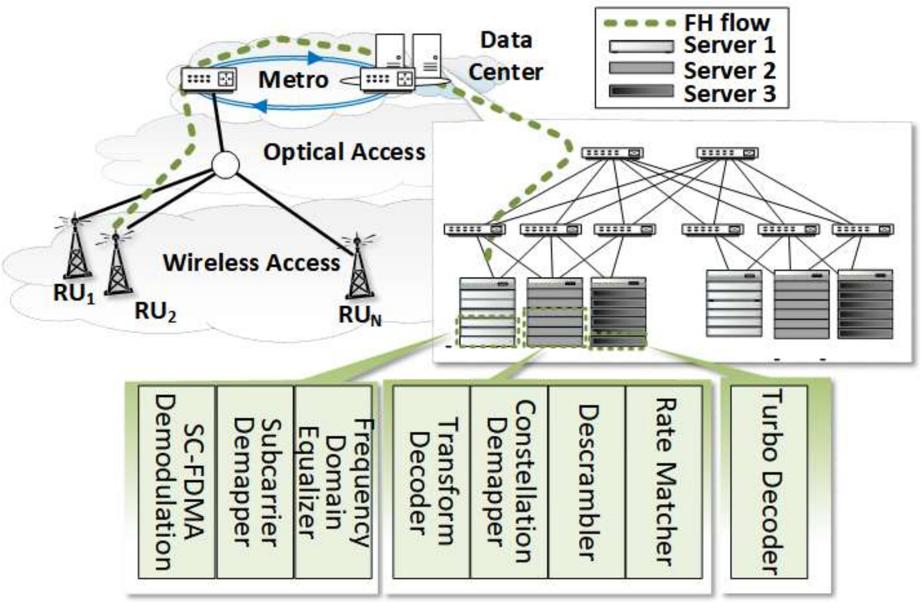

Fig. 1. Centralized processing of softwarised-RAN functions on a data center hosting different type of servers

3 Framework

3.1 Heuristic Design

As shown in our previous work [2], a multi-stage Integer Linear Programming (ILP) model can be effectively used to identify the optimal placement of BBU functions within the datacenter, but it suffers high computational complexity thus making it unsuitable for real time system deployments. To address this issue, a heuristic algorithm with low computational complexity is proposed that tries to identify the optimal compute resources required to support the most energy efficient processing of the BBU Service Chain within the DC.

The analysis of the LTE uplink application, provided by the WiBench suite [3], showed that different construction elements of the BBU chain have different requirements in terms of processing. From the 8 different functions, Turbo Decoder was proven to be the one with the dominant contribution to the total BBU service chain instruction requirements, as shown in **Fig. 2** and **Fig. 3**.

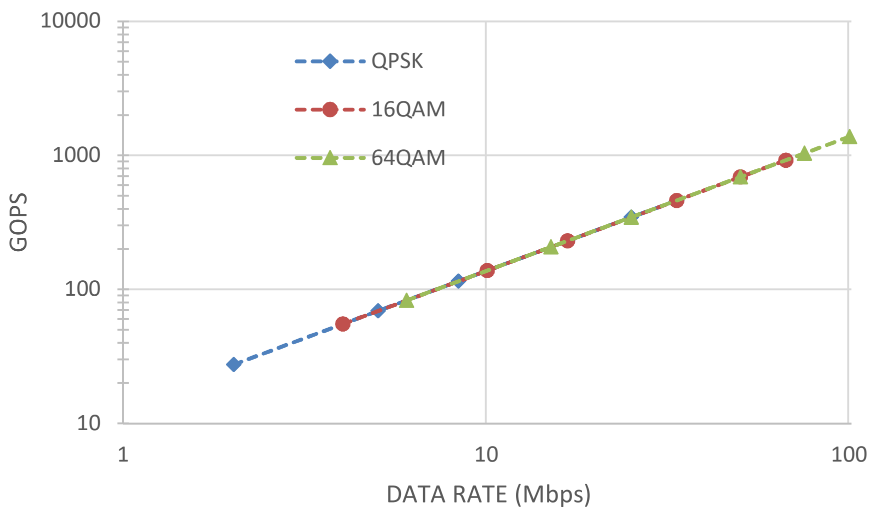

Fig. 2. Operations per second, measured in GOPS, under various data rates for the Turbo Decoder

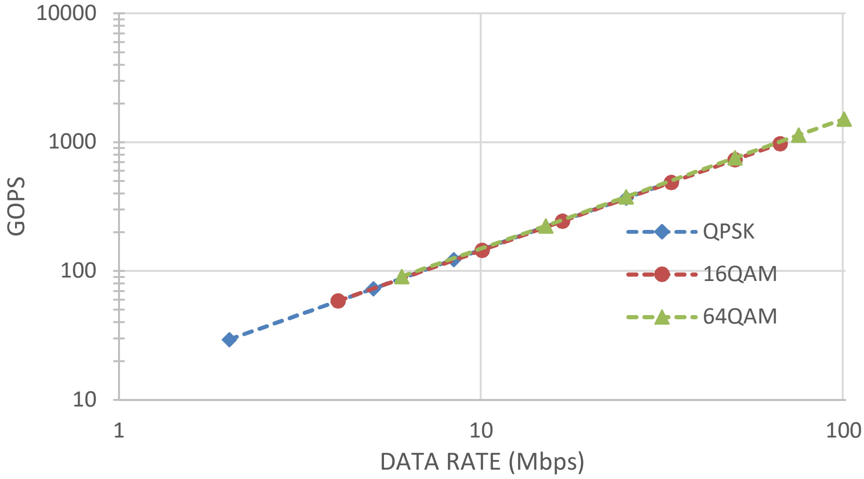

Fig. 3. Operations per second, measured in GOPS, under various data rates for the Total BBU Service Chain

Taking this information into consideration, in order to limit the complexity of the heuristic, we divide the 8 different BBU processing functions into two sub-sets of functions (1st and 2nd sub-set, as shown in **Fig. 4**. To satisfy the requirements of the BBU Service Chain, the order of these processing functions is always maintained within and across the 2 different sets of functions defined. The first sub-set includes the SC-FDMA, Sub-carrier Demapper, Frequency Domain Equalizer, Transform Decoder and Constellation Demapper functions, while the second sub-set comprises Descramble, Rate Matcher and Turbo Decoder functions. As shown in [4], the proposed grouping policy has been selected as it requires a relatively small amount of network resources for the interconnection of the first (1st) with the second (2nd) sub-set of functions while the computational requirements of the 2nd sub-set is still very high.

The main objective of the proposed heuristic is to allocate an input BBU service chain to the most energy efficient servers that have sufficient capacity to process it. The input service can be split and allocated to a set of servers, in case that splitting the service across servers is a more energy efficient option. A more detailed description regarding the server allocation process is provided in **Fig. 5**.

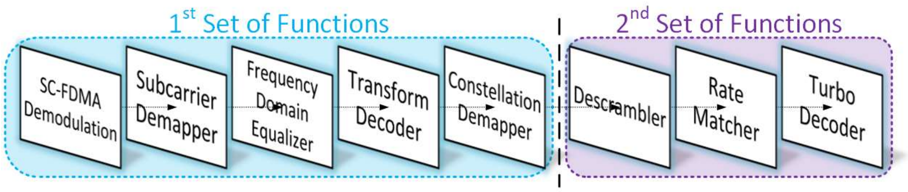

Fig. 4. Reduction of complexity through grouping of BBU functions into 2 sets

```
for each RU:
  if a server that is already used can process it in time:
    assign it to that server
    update that server's capacity
  else:
    assign it to the most efficient server type (search
    between types 1-3) which can process it
    update that server's capacity
    if only the least energy efficient server type (type 4) can
    process it:
      check the thresholds to decide if and which split
      option should be enabled
      if split is enabled:
        create the 2 sets of functions
        find the closest appropriate servers (Dijkstra)
        assign the sets to the appropriate servers
        update the servers' capacity
      else:
        assign it to a least energy efficient server (type 4)
        update the server's capacity
```

Fig. 5. The Heuristic developed for BBU assignment problem

In our analysis, we were aiming at always serving the input traffic, independent of the volume of incoming data to be processed, satisfying at the same time, the time constraints associated with the service. Therefore, we are considering the ratio of the number of instructions required for the 2nd sub-set of functions to be performed, over the number of instructions of the 1st sub-set of functions.

The scenario considered assumes the DC topology illustrated in **Fig. 6**. This topology includes 6 racks. **Fig. 6a** depicts the interconnection of the Top-of-the-Rack (ToR) switches. Each rack incorporates 48 servers, as shown in **Fig. 6b**. The connectivity between the racks is assumed to be provided by an optical switching solution described in [5]. For the numerical calculations, four different server types were randomly placed within each rack. The technical specifications of the servers assumed are provided in **Table 1**. These servers can be classified according to their performance in terms of energy efficiency, with type 1 server being the most energy efficient, while type 4 the least energy efficient server.

Considering this assumption, we calculate the ratios of the capacity of the larger type of server (type 4 server, least energy efficient) over the capacities of the rest of the servers (type 1, type 2 and type 3). Based on these ratios and the time constraints associated with the LTE upload service (in total <1ms per sub-frame), we define a set of thresholds that can be used to identify whether an incoming service chain (including a set of functions) can be split between the larger server type and any other of the smaller available type of servers.

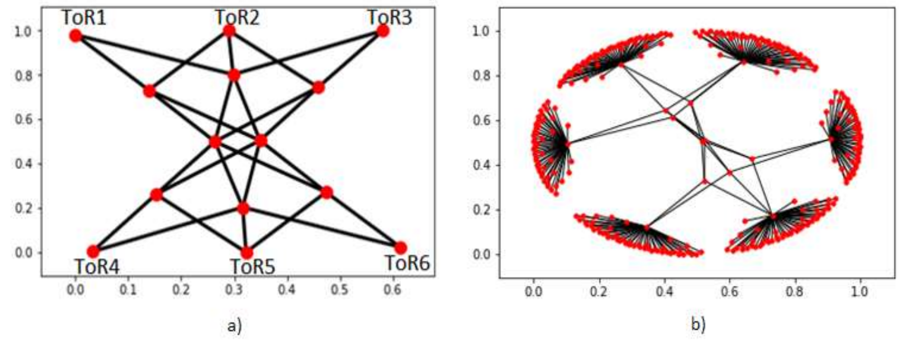

Fig. 6. The DC Topology used for the experiments. In a) is presented the interconnection of the ToRs in the DC and in b) is displayed the total intra-DC network, including the servers under each ToR

For the specific functional split and the set of servers considered in this study, the numerical values of the thresholds we have identified are: a) 68% of type 4 server processing capacity if the 1st sub-set of functions is allocated to server type 1, b) 69% of type 4 server processing capacity if the 1st sub-set of functions is allocated to server type 2 and c) 70% of type 4 server processing capacity if the 1st sub-set of functions is allocated to server type 3. It should be noted that in our calculations additional processing margins of the order of 2% have been allowed.

Table 1. Technical specifications of the servers used in the numerical evaluations

COMPUTER / DEVICE	SERVERS	CHIPS	CORES	THREADS	GOPS	POWER (Watt)	GOPS / Watt	IDLE (Watt)
SuperMicro X11DPi-N(T) SMC X11	2x Intel Xeon Platinum 8160	2	48	96	1071.37	360	2.976	53.4
SuperMicro X11DPG-QT	2x Intel Xeon Gold 6140	2	36	72	888.52	336	2.644	52.4
SuperMicro X10Dai SMC X10	2x Intel Xeon E5-2683 v4	2	32	64	700.94	288	2.434	81
Sugon I908-G20	8x Intel Xeon E7-8860 v3	8	128	256	2510.56	1344	1.868	269

3.2 Numerical Results

To quantify the benefits of the VM migration, the simple DC network topology of **Fig. 6** is considered. This comprises 6 racks, each one composed of 48 servers. Four different types of servers were randomly placed inside each rack.

We experimented for various initial DC loads (10%, 15%, 20% and 25%), for 2 different scenarios. In the first scenario, the initial load was distributed on a low number of servers (80 servers). This scenario will be referred to as “compact”. In the second scenario, the initial load was allocated to a large number of servers (180 servers). This scenario will be referred to as “spread”. In both scenarios the heuristic that was developed was used to assign the incoming traffic to the remaining DC resources.

Through our experiments we calculated the total power consumption of the DC, the average CPU utilisation of the switched-on servers, as well as the total DC utilisation. The total DC utilisation is defined as the percentage of the switched-on servers compared to the total number of DC servers. All the experiments showed that the scenario with the “compact” initial load had much better utilisation of the DC resources and much lower power consumption than the “spread” scenario.

Fig. 7 presents the benefits, in terms of power consumption, in the case where live VM migration is applied. It also shows that by migrating the initial DC load, a great reduction of the number of switched-on servers can be achieved. In **Fig. 8** we observe that the average CPU utilisation of the switched-on servers is much higher in case of the “compact” scenario compared to the “spread” scenario. It can be also noted that the CPU utilisation reaches a maximum value introduced due to live VM migration. In order to migrate a VM in real time, some of its resources have to be used to support the migration. To address this requirement, a threshold of 75% of CPU usage was adopted, as proposed by [6].

The difference in the power consumption of the schemes under evaluation observed in **Fig. 7**, is due to the minimization of the power achieved when setting servers to the idle state. The idle state consumes less power but none the less it still adds a fair amount to the total power consumption. In the “spread” scenario most of the switched-on servers had low CPU utilization, which means that they spend a considerable amount of time idle. On the other hand, in the “compact” scenario the number of switched-on servers were significantly reduced, which led to the high CPU utilisation of these servers and the reduction of their time spent in the idle state.

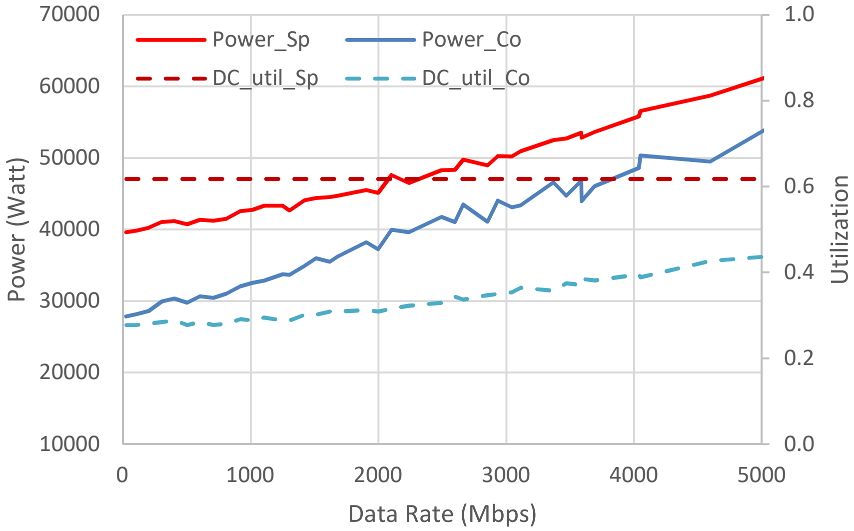

Fig. 7. Power consumption and DC utilisation for 15% initial DC load for various Data Rates. Dotted lines correspond to the DC's utilisation while solid lines to the Power consumption. Red corresponds to the "spread" scenario while blue to the "compact"

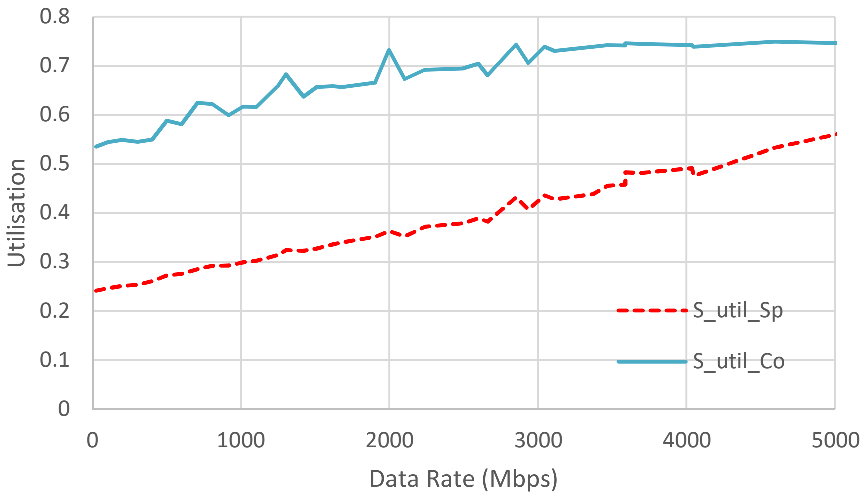

Fig. 8. Switched-on servers' utilisation for 15% initial DC load for various Data Rates. The red dotted line corresponds to the "spread" scenario while the blue solid line to the "compact"

4 Discussion

Although VM migration comes with great benefits for the DCs, it also contains the risk of service disruption and as such Service Level Agreement (SLA) violations [6-10]. In order to migrate a VM, one must contemplate the total time during which the services, supported by this VM, would be unavailable and try to maintain this below an acceptable threshold. A second aspect to consider is the total migration time in which both machines are synchronizing their states and therefore reliability might be affected. Finally, one must ensure that the active services running on the DC would not be disrupted due to the resources allocated to the VM migration.

The duration between the initiation of the live VM migration and the moment during which the original VM can be discarded is defined as the total migration time and can be divided in three phases as shown in **Fig. 9**. The first phase referred to as "image-copy" phase, is the phase where all the memory pages from the source VM are copied to the destination VM. During this process, a number of memory pages may change. At the pre-copy phase, the memory pages that have changed are re-copied. At the stop-copy phase, also known as downtime, the source VM is suspended and the remaining dirty pages are transferred to the source VM which spins up.

The total migration time is vastly affected by the volume of the VM and the network capacity, since migration involves the transfer of the entire VM volume from one physical server to another. While this transfer takes place, the rest of the services already running at the DC will have to be supported without disruption, thus maintaining the relevant resources.

This implies that the DC network capacity that can be used for migration purposes is only the one remaining unused by the rest of the running services. In addition, for scenarios in which more than one VMs need to migrate in a short-time window, there is a clear need for identifying suitable migration scheduling schemes to avoid DC network congestion situations [10,11].

Furthermore, the additional CPU overhead during migration may cause service disruption. This leads to the introduction of a threshold to the CPU utilization, in order to be able to manage the CPU overhead caused by migration.

Fig. 9. Live VM migration phases

The main element that affects the downtime is the dirty page rate. The rate at which the memory of the source VM is being written, during the migration process, is called dirty page rate, measured in pages per second, or dirty rate, measured in MBps and in the case in which the dirty rate is similar with the transfer rate, the result is the increase of the downtime.

5 Conclusions

This paper focused firstly on the design of a heuristic, which assigns the BBU functions to the DC servers in order to minimize the total power consumption at the DC and secondly on the investigation of the benefits of VM migration inside a DC. Experimental results have shown significant power savings and substantial reduction of the total DC resources utilisation, in a heterogeneous DC, when live VM migration is applied. A discussion on how live migration can be performed in these environments is also provided.

Acknowledgements
This work has been financially supported by the EU Horizon 2020 project 5G-PICTURE under grant agreement No 762057 and the EU Horizon 2020 project IN2DREAMS under grant agreement No 777596.

References

1. A. Tzanakaki *et al*., "Wireless-Optical Network Convergence: Enabling the 5G Architecture to Support Operational and End-User Services," in *IEEE Communications Magazine*, vol. 55, no. 10, pp. 184-192, October 2017.
2. N. Gkatzios, M. Anastasopoulos, A. Tzanakaki and D. Simeonidou, "Compute Resource Disaggregation: An Enabler for Efficient 5G RAN Softwarisation," *2018 European Conference on Networks and Communications (EuCNC)*, Ljubljana, Slovenia, 2018, pp. 1-5.
3. Q. Zheng, Y. Chen, R. Dreslinski, C. Chakrabarti, A. Anastasopoulos, S. Mahlke, and T. Mudge, "WiBench: An Open Source Kernel Suite for Benchmarking Wireless Systems", 2013 IEEE International Symposium workload Characterization (IISWC), Portland, OR, 2013, pp.123-132.
4. A. Tzanakaki, M. Anastasopoulos, and D. Simeonidou, "Converged Access/Metro Infastructures for 5G services," in Optical Fiber Communication Conference, OSA Technical Digest (online) (Optical Society of America, 2018), paper M.
5. J. Perelló *et al*., "All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput," in *IEEE Network*, vol. 27, no. 6, pp. 14-22, November-December 2013.
6. Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif, "Black-box and gray-box strategies for virtual machine migration." In Proceedings of the 4th USENIX conference on Networked systems design & implementation (NSDI'07). USENIX Association, Berkeley, CA, USA, 2007, 17-17.

7. Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield, “Live migration of virtual machines.” In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2 (NSDI'05), Vol. 2. USENIX Association, Berkeley, CA, USA, 2005, 273-286.
8. William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya, “Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation.” In Proceedings of the 1st International Conference on Cloud Computing (CloudCom '09), Martin Gilje Jaatun, Gansen Zhao, and Chunming Rong (Eds.). Springer-Verlag, Berlin, Heidelberg, 2009, 254-265.
9. Akshat Verma, Puneet Ahuja, and Anindya Neogi, “pMapper: power and migration cost aware application placement in virtualized systems.” In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware '08), Valérie Issarny and Rick Schantz (Eds.). Springer-Verlag New York, Inc., New York, NY, USA, 2008, 243-264.
10. M. F. Bari, M. F. Zhani, Q. Zhang, R. Ahmed and R. Boutaba, "CQNCR: Optimal VM migration planning in cloud data centers," *2014 IFIP Networking Conference*, Trondheim, 2014, pp. 1-9.
11. Alexander Stage and Thomas Setzer, “Network-aware migration control and scheduling of differentiated virtual machine workloads.” In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09). IEEE Computer Society, Washington, DC, USA, 2009, 9-14.