SCENARIO FOR OPTIMUM SERVICES POSITION WITH CONTROL SYSTEM HUB AND OPEN SOURCE MANO IN EDGE COMPUTING

Andra-Isabela-Elena CIOBANU[1], Eugen BORCOCI[2], Marius VOCHIN[3]

To outcome the higher demand on IT applications, the enterprises and service providers must build and expand the use of edge computing. This can lead to possible difficulties concerning how to place workloads of network services automation and evaluate the costs in an optimal way. This paper's intention is to present a scenario with three different virtual network functions designed with specific restrictions and different compute costs and the Control System Hub with Open Source Mano solution to limit and face those restrictions. To achieve this, the optimal placement algorithm uses the Control System Hub network analytic platform along with virtual charmed factors for control and Quality-of-Service measures.

Keywords: Network Function Virtualization (NFV), edge cloud, latency, cost, workload allocation, optimization, analytics

1. Introduction

Edge computing is a method of optimizing cloud computing systems which makes contact with the physical world. It takes the control of computing applications, data, and services away from some central nodes (the "core") to the other logical extreme (the "edge") of the Internet". In this architecture, data comes in from the physical world via various sensors, and actions are taken to change physical state via various forms of output and actuators. Edge Computing takes advantage of proximity to the physical items of interest also exploiting relationships that those items may have one to each other. [1]

This approach requires leveraging resources that may not be continuously connected to a network such as autonomous vehicles, implanted medical devices, fields of highly distributed sensors, and mobile devices. As the data streams generated from Internet of Things (IoT) devices are transmitted to the remote cloud via Internet, the transferred data may consume a huge amount of bandwidth

[1] PhD student, Faculty of Telecommunications, and Information Technology University POLITEHNICA of Bucharest, Romania, e-mail: andraciobanu90@yahoo.com
[2] Prof., Faculty of Telecommunications, and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: eugen.borcoci@elcom.pub.ro
[3] Reader, Faculty of Telecommunications, and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: Marius.vochin@upb.ro

and energy of the core NFV network. On the other hand, since the remote cloud is far from IoT users which send application requests and await the results generated by the data processing in the remote cloud, the response time of the requests may be too long. [2] Therefore, this paper intention is to provide a feasible vision which can be employed to alleviate the traffic load in the NFV network by optimally placing services and Virtual Network Functions (VNFs) workloads and adjusting the costs. This is achieved with an analytic algorithm and active network monitoring through vCFs, both delivered by Control System Hub (CSH) platform and Open Source Mano (OSM) responsible for lifecycle management.

The structure of paper is the following: Section 2 debates the related work around the topic - challenges and context: workload and limitations considerations in edge computing, the costs of links and employment analytics. Section 3 presents the scenario architecture and the examples related to latency QoS and link service costs and metrics. In section 4 details about the platforms used for the proposed scenario are exposed. Section 5 is allocated for conclusions and future work. Drive-through keys for scenario:

- Restriction samples Network Service Descriptors (NSDs) to be able to grasp the service fulfilment specification
- Assignment of VNF workloads built on latency stipulation
- Assignment of optimal cost pattern to forecast link costs

2. Related work - challenges and context

There are two main types of workloads for developers to consider when it comes to edge computing: out-of-band and inline (or in-band). The more straightforward of the two are out-of-band workloads, which could also be categorized as synchronous or transactional. Within this kind of workload, a client issues a request, and the system issues a block on the response, such as in the case of static file delivery.

Inline workloads are significantly more complex. Moreover thought of as non-concurrent or non-transactional, these sorts of edge inline workloads contain custom rationale to handle handling promptly upon ingestion of information, instead of sending it back to a centralized framework to be prepared. When inline workloads are introduced at the edge, the whole computing show changes. [3]

The state-of-the-art works on edge computing includes platforms and frameworks that aim to provide scalable processing closer to the network border, in order not only to provide low latency results, but also to better utilize resources available on the network. The approaches to solve these problems include predominantly cloud-like deployments at the edge (often described as cloudlets). When the workload of a cloudlet is too heavy, the computing resources available for an application is limited, and thus the response time of the corresponding requests degrades correspondingly [4].

Some general challenges of edge computing are related on: how to discover edge nodes, how to partition and offload tasks, how to use edge nodes in order for Quality-of-Service (QoS) and Experience (QoE) to be achieved.

Quality delivered by the edge nodes can be captured and monitored through their workloads or constraints of their services . One principle that will need to be adopted in edge computing is to not overload nodes with computationally intensive workloads [5].

Regardless of whether and edge node is exploited, the user of an edge device or a data center expects a minimum level of service. For example, when a base station is overloaded, it may affect the service provided to the edge devices that are connected to the base station [6].

A thorough knowledge of the peak hours of usage of edge nodes is required so that tasks can be portioned and scheduled in a flexible manner. The role of a management framework will be desirable but raises issues related to monitoring, scheduling, and re-scheduling at the infrastructure, platform and application levels. [7]

Other specific matters of challenges and limitation in edge computing can intrude, as it can be seen in Fig. 1:

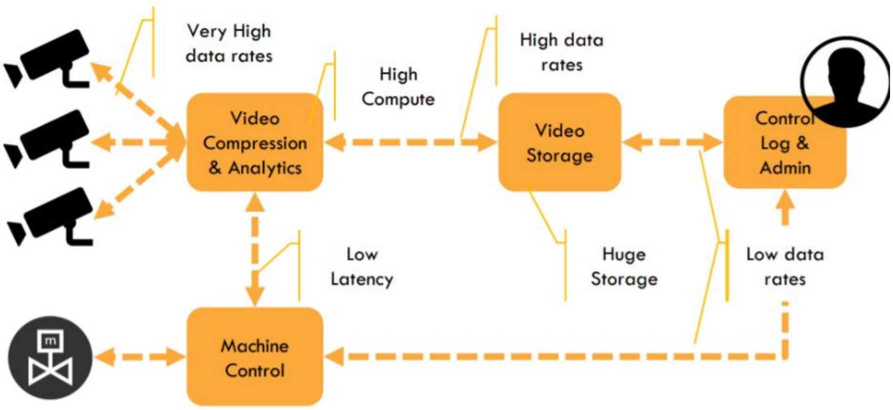


Fig. 1. Challenges in Edge Computing [7]

- several links with particul ar prerequisites
- cost conduit is other for every part/link
- optimum position is different for each software part
- cost of links and compute rely on resource disposition, location etc.
- data relationships between resources and the users of services implies transport capacity, which is another restriction with a cost
- Analytics disposition of workloads in terms of packet loss, security, bandwidth etc.

3. Proposed scenario

The vision around the paper is related to an optimal position of the services in a NFV network in edge computing context, dependent of some restrictions and their workloads or metrics and an active monitor and testing with virtual charmed factors (vCF). (this is achieved through CSH Analytics platform – more details about platforms and software are in section 4).

3.1 Introduction to optimum service position

An optimal position of services doesn't rely only on cost and compute; it also implies network relationships among customers, data sources and destinations, as well as competitor compute in Point of Presence (PoPs) in the network. In the cloud network constant evaluation of infrastructure errors, capacity growth or new service implementations.

These challenges apply to network services (NS), VNFs but also to application services.

There are four types of information in edge computing from the analytic workload point of view [8]:

- Service prerequisites - defined here as restrictions
- Details about the essential infrastructure - evaluation of status and topology
- A cost model in conjunction with the infrastructure
- An analytic workload decision

Edge computing is still in its infancy and a framework to facilitate this is not yet available. Such frameworks will need to satisfy requirements, such as application development to process requests in real-time on edge nodes. Current cloud computing frameworks, such as the Amazon Web Service [9], Microsoft Azure [10] and Google App Engine [11], can support data-intensive applications, but implementing real-time data processing at the edge of the network is still an open research area. Evolving distributed computing environments have resulted in the development of numerous techniques to facilitate partitioning of tasks that can be executed at multiple geographic locations [12], [13]. For example, workflows are partitioned for execution in different locations, as it can be seen also in Fig. 2 [14]. Task portioning is usually expressed explicitly in a language or management tool. However, making use of edge nodes for offloading computations poses the challenge of not simply portioning computational tasks efficiently, but doing this in an automated manner without necessarily requiring to explicitly define the capabilities or location of edge nodes. [15]

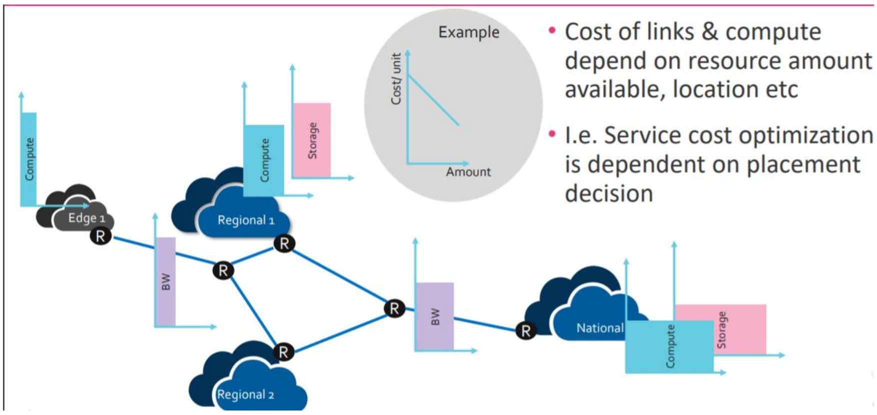


Fig. 2. Workflows and costs in edge locations [15]

3.2 Proposed architecture

Based on what has been presented in subsection 3.1, the proposed architecture contains three main parts, from a high level perspective:

- An analytic optimal solution based on software from the analytic server from CSH
- Same CSH platform offers live monitoring of different QoS (bandwidth, packet loss, voice and video quality etc.) through software virtual factor charms
- Open Source Mano used for lifecycle management part: besides the design and configuration of Virtual Network Functions Descriptors (VNFDs) and Network Service Descriptors (NSDs), instantiation is also provided.

The other functional used tools are:

- OpenStack as Virtual Infrastructure Management (VIM) where four different data centers are developed as instances
- Juju and Python
- VMware
- The overall scenario from Fig. 3 presents three services which have been experimentally created. Also, specific restrictions have been allocated. (these can be seen at left part in OSM, regarding resources allocation for the slices). A bandwidth service (which can be for example a web service), an IoT service (video one) and an industry service (which can be anything related to manufacture). The service chain contains three VNFs and the attached vCF, the innovation vCF which is basically an active VNF and acts like a virtual test agent, it can generate traffic, it can do real service requests in order to measure through the entire service chain and make it sure it works configured also as a VNF.

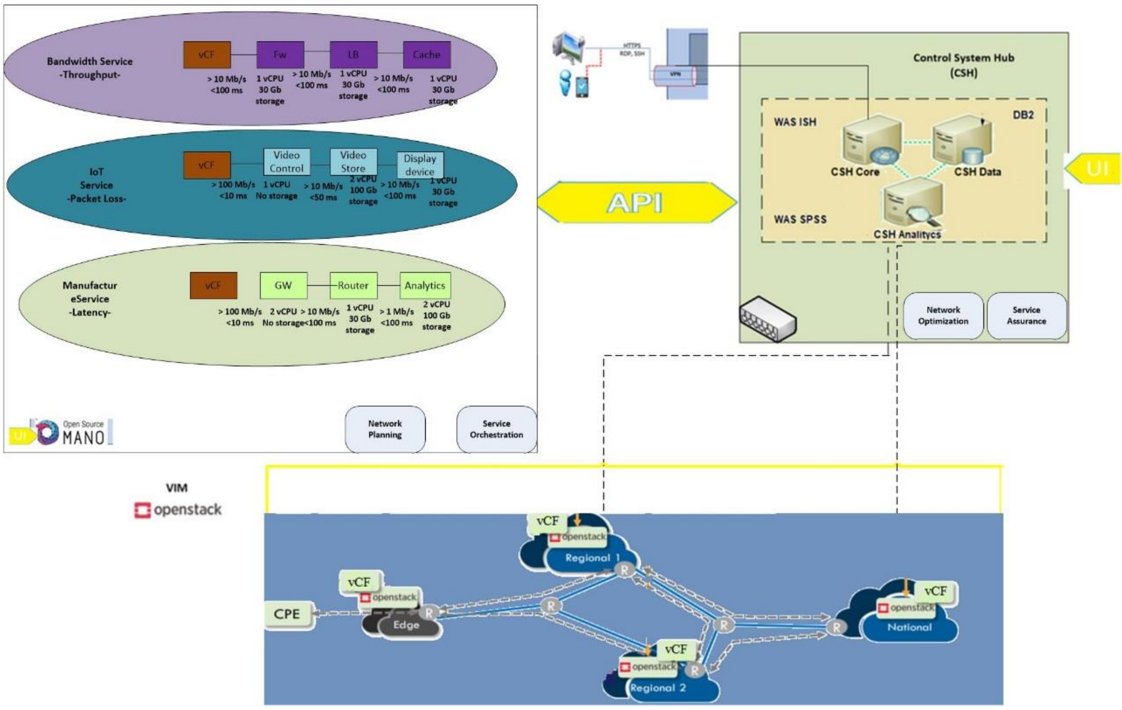


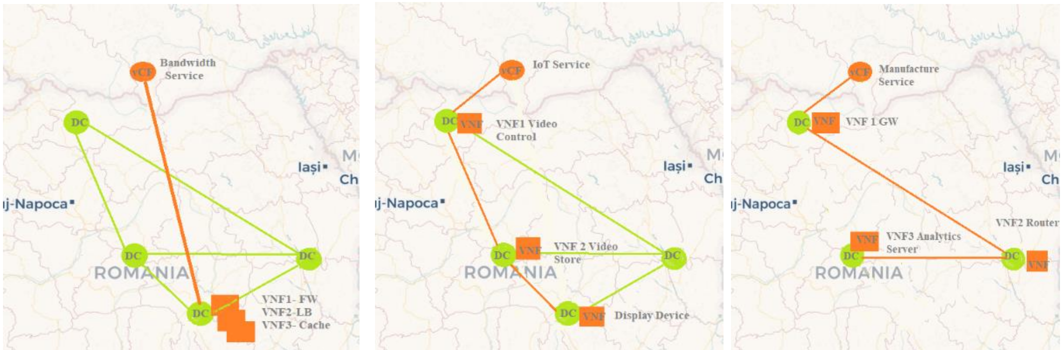
Fig. 3. Edge Computing Scenario optimum services workload position

The VIM infrastructure is designed as four data centers (DC). vCF agents are also attached in each datacenter in order to capture active latency metrics for all DCs. The steps of the entire workflow:

a. Blueprint/Pattern items - VNFDs and NSDs (first of all, it starts with the elaboration of VNFDs in OSM UI, more specifically in the OSM catalog)
b. Blueprint activation test templates for three different restriction models which are going to be presented below. This part is achieved in CSH hub where specific parameters and maps are defined.
c. Instantiation of VNFs and services
d. Start the VNFs and test agents vCFs
e. Capture network properties.

The scenario has been thought in **three main parts** (see Fig. 4):

1. No-latency or mild latency service in DC with lowest cost;
2. Latency-critical service. In this case, some VNFs need to be allocated near to the customer;
3. DC failover - the position analytic software from CSH can evaluate the redeployment of the VNF/service in case that a DC fails, and the back-up DC takes over the functionality. In this case, the cost will increase.



- 1 - - 2 - - 3 -

Fig. 4. DCs services workload positions

4. Platforms used for the proposed architecture

NFVI Storage Hardware- IBM Power Systems AC922 servers have been used with dual Intel Xeon CPUs E5-2620. Operating System is Linux.

VIM

Openstack has been used to be the cloud computing part of the architecture, as well as the VIM part.

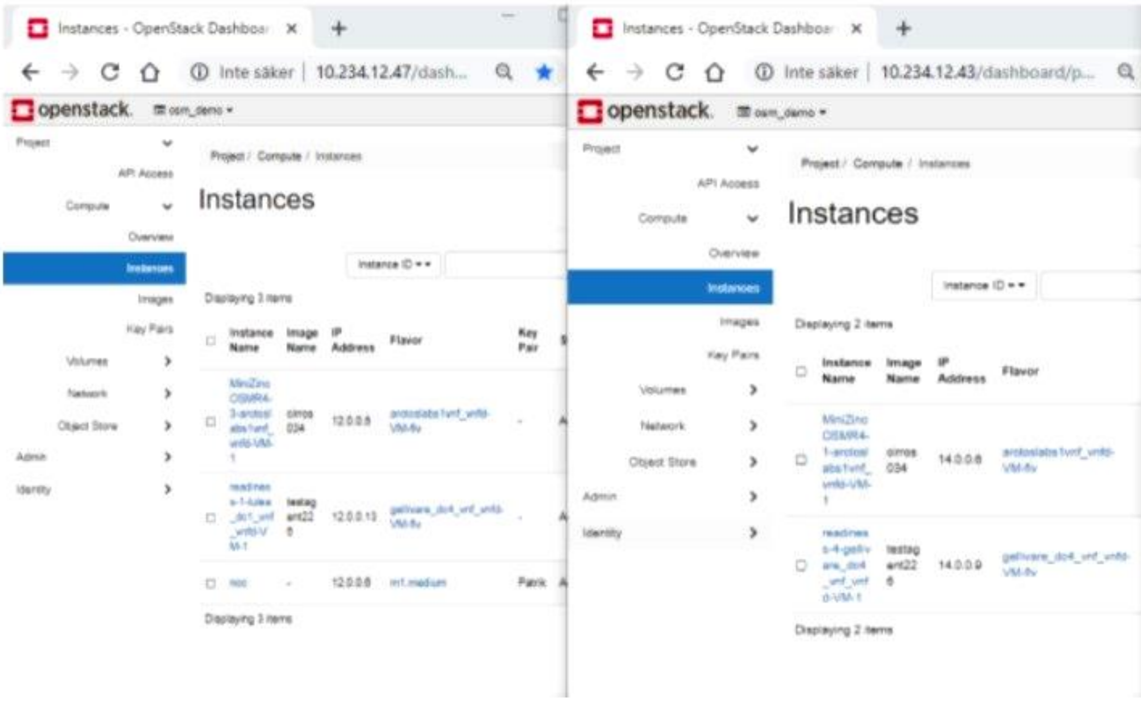


Fig. 5. Openstack Dashboard

OSM

OSM is an ETSI-hosted open source community delivering a production-quality MANO stack for NFV, capable of consuming openly published information models, available to everyone, suitable for all VNFs, operationally significant and VIM independent. OSM is aligned to NFV Industry Specification Group (ISG) information models while providing first-hand feedback based on its implementation experience [16].

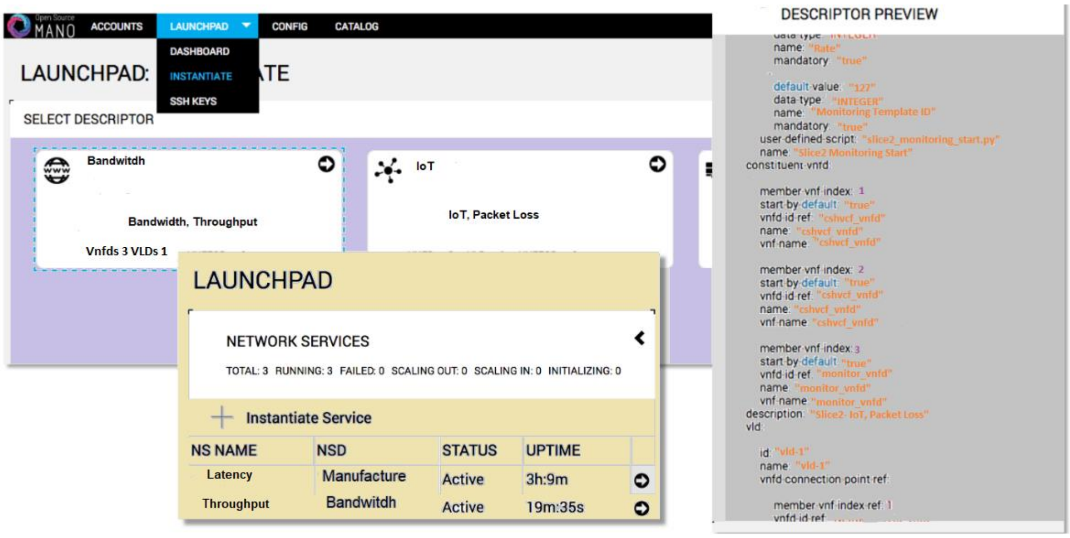


Fig. 6. OSM Dashboard

In Figs. 5 and 6 the instantiation of DCs, services, VNFs and vCFs is presented. In OSM User Interface (UI), go to NSD which takes the actual creation of the NS by bringing the VNF from the catalogue. The constitute information of NSD captures not only the essence of the service itself but also the componence of the service primitives, packet monitoring etc. When Slice 2 becomes active state, orchestration has completed its part. OSM will instantiate also the vCFs. Using Openstak as VIM, OSM will deploy vCF on strategic locations as part of the network service chain.

CSH

CSH platform contains tipically three servers: the Data, Core, and Analytics components, which are hosted on separate servers to provide higher capacity and quality of service that a production environment needs.

At the Core part, the main component consists of IBM® WebSphere® Application Server ISH (Service hub) which serves applications from the front office to the back office - whether mobile or web applications, whether mission critical or not, whether hosted in the cloud or on premises. It has HTTP server and plugins for API.

CSH Analitycs is the place where traffic for vCFs is generated. It has IBM SPSS Analytics client software for in-depth data exploration, reporting and modeling. CSH Data contains information about templates of vCFs and also it stores the reports of monitoring which can be send to customers. CSH communicates with OSM via API and it also coordinates the vCFs.

A main characteristic of this solution is the potential of quantify with the latency on links in the patterns. (see Fig. 7). This is reached in CSH with the use of vCFs. Particular tests in terms of metrics and QoS can be measured (latency, jitter, loss, throughput etc).

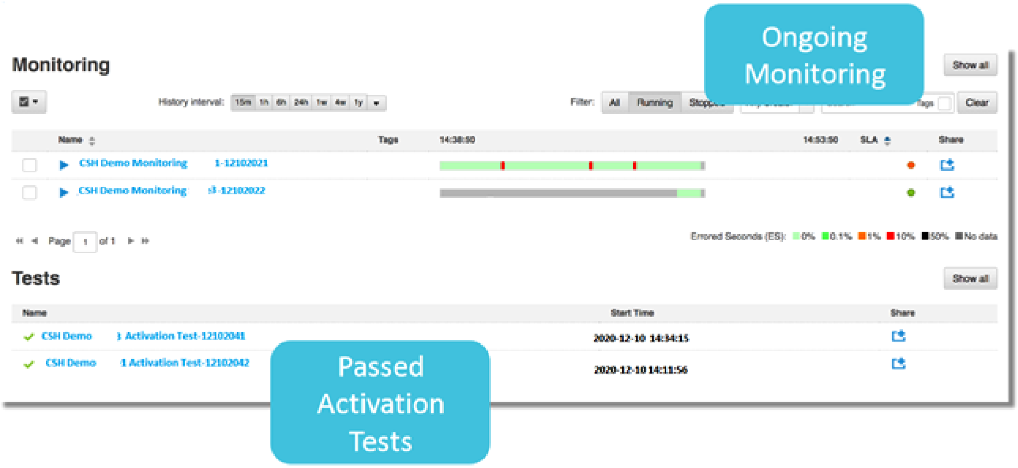


Fig. 7. CSH Dashboard

5. Conclusions

Questions like: is the Service Level Agreement (SLA) valid at roll-out point? is the SLA valid before transfer to the customer? what about the transfer of birth certificates? are more and more present in NFV industry. Life and active tests are a proper answer for this, with the integration of vCFs. They can detect earlier cases related on how to get insight of quality assurance from end-users' view and appreciate which parts of the network are affected. They can create traffic and monitor it at different levels of the service chain/network slice, in order to avoid potential issues prior to the last traffic sent to the final customers. Demonstration tests can be performed based on firewall rules, latency, throughput, QoS priority etc. The vCFs have the capability to be versatile, they can be placed at the basic OS levels, hypervisors, VNFD, NSD designs or the entire. This paper brings out the main scientific contributions based on how to test and monitor the quality of service (QoS) of network services or service chains in multidomain edge network slices. A proposed arhitecture based on OSM and another trobleshoot platform is made. The software simulations are accomplished with charms and primitives and things which have been already presented in previous chapters.

As future objectives, reaserch on 5G network slicing, edge computing, NFV and SDN is going to be done.

Acknowledgment

The research leading to these results has received funding from the NO Grants 2014-2021, under Project contract no. 42/2021, RO-NO-2019-0499 - "A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services" - SOLID-B5G.

R E F E R E N C E S

[1]. https://madhavuniversity.edu.in/edge-computing.html
[2]. L. Wang and R. Ranjan, "Processing distributed internet of things data in clouds," IEEE Cloud Computing, vol. 2, no. 1, pp. 76–80, 2015.
[3]. https://www.section.io/blog/edge-workload-considerations/.
[4]. *N. Ansari and X. Sun*, "Mobile edge computing empowers internet of things," IEICE Trans. on Comm., vol. E101-B, no. 3, pp. 604–619, 2018.
[5]. *M. T. Beck and M. Maier*, "Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms," in Proceedings of the International Conference on Advanced Engineering Computing and Applications in Sciences, 2014, pp. 65–70.
[6]. *P. Simoens, L. Van Herzeele, F. Vandeputte, and L. Vermoesen*, "Chal-lenges for Orchestration and Instance Selection of Composite Services in Distributed Edge Clouds," in Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, 2015, pp. 1196–1201
[7]. *G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra*, "MADAM: A Multi-level Anomaly Detector for Android Malware," in Proceedings of the International Conference on Mathematical Methods, Models and Architectures for Computer Network Security: Computer Network Security, 2012, pp. 240–253
[8]. *A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel*, "The Cost of a Cloud: Research Problems in Data Center Networks," SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 68–73, Dec. 2008
[9]. https://aws.amazon.com
[10]. https://azure/microsoft.com/en-gb/
[11]. https://appengine.google.com/
[12]. *D. Yin and T. Kosar*, "A Data-aware Workflow Scheduling Algorithm for Heterogeneous Distributed Systems," in International Conference on High Performance Computing and Simulation, 2011, pp. 114–120
[13]. *T. Ghafariana and B. Javadi*, "Cloud-aware Data Intensive Workflow Scheduling on Volunteer Computing Systems," Future Generation Computer Systems, vol. 51, pp. 87–97, 2015
[14]. *W. Chen and E. Deelman*, "Integration of Workflow Partitioning and Resource Provisioning," in Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012, pp. 764–768
[15]. *W. Tang, J. Jenkins, F. Meyer, R. Ross, R. Kettimuthu, L. Winkler, X. Yang, T. Lehman, and N. Desai*, "Data-Aware Resource Scheduling for Multicloud Workflows: A Fine-Grained Simulation Approach," in Proceedings of the IEEE International Conference on Cloud Computing Technology and Science, 2014, pp. 887–892
[16]. OSM Release FIVE, [Online]. Available from: https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE 2019.12.11