Easing the Deployment and Management of Cloud Federated Networks Across Virtualised Clusters

Ivan Andrade Castañeda[1], Ignacio Blanquer[1][a] and Carlos de Alfonso[2][b]
[1]*Institute for the Instrumentation for Molecular Imaging - I3M, Universitat Politècnica de València, Valencia, Spain*
[2]*SENESCYT - Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, Ecuador*

Keywords: Cloud Federation, Overlay Networks, Virtual Machine Migration, Federated Virtual Private Network.

Abstract: Cloud federation, in the last years, has grown rapidly in literature because to its multiple advantages to coordinate and re-use different services across multiple sites, different geographic locations or cloud providers. This article presents a federated network architecture and focuses on the multi-tenant overlay networks creation across different sites, as well as, the inter-site migration of virtual machines, introducing a framework that allows us to manage our federate cloud environment in three scenarios: distribution of Virtual Machines, resource management, and network management.

1 INTRODUCTION

Nowadays, big data centers networks provide redundancy and ensure reliability to their customers in case of a site failure, also has to minimize cost while dealing with high peaks of demands. Thus, cloud federation has grown rapidly in the literature and became a powerful paradigm for cloud providers and for scientific computing, as its facilities the interoperability of different cloud computing environments, offering access to different resources that are allocated in different geographic locations or cloud providers.

Despite containers are sometimes treated as virtual machines, they are conceptually different objects. Containers are groups of processes that run on environments isolated at the level of a filesystem, devices, process namespaces or resource allocations. Containers are mainly implemented through operative system calls and therefore do not boot a separate operating system. They offer other advantages as packaging applications dependencies, image preparation and distribution, and memory footprint. However, container isolation is reduced with respect to Virtual Machine environment, especially in a multi-tenancy environment.

The network and resource management requirements are demanding, with the emerging need for a middleware layer to provide a general orchestration of resources, giving the right connectivity between the resources across the federation, as well as a complete view of the entire network from any point of the cloud federated environment.

Such a middleware layer will imply requirements as Network re-programming (such as SDN, which makes possible to separate the Control Plane from the Data Plane (Kaur et al., 2016)), a centralized service that provides a reliable storage data, discovering and configuring services (in our case Consul (Consul, 2019), EC managed (EC-Managed, 2019) or etcd (Coreos, 2019)), coherent IP address assignment supporting WAN live VM migration, user authentication, and authorization, etc.

The framework that we present in this paper, implements a solution that performs resource federation at the network layer, based in Software Defined Networking (SDN) technology, creating virtual networks on demand across multiple data centers including the VM live migration over WAN in a federated environment.

In this article, we are going to test and analyze the benefits of the framework in 3 different scenarios: distribution of VMs, resource management, and network management. The rest of the paper is structured as follows. First, Section II describes an overview of related works in cloud federation, network federation and live migration. In Section III, we define the federated network topology used, based on three different data centers. Section IV describes more in detail the lower-level processes for each scenario that our

[a] https://orcid.org/0000-0003-1692-8922
[b] https://orcid.org/0000-0002-2378-021X

Castañeda, I., Blanquer, I. and de Alfonso, C.
Easing the Deployment and Management of Cloud Federated Networks Across Virtualised Clusters.
DOI: 10.5220/0007877406010608
In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, pages 601-608
ISBN: 978-989-758-365-0

model offers for both, network and resources management. The preliminary results are presented in section V. The article concludes in Section VI, where we discuss the proposed approach, summarizes the work and suggests future works

2 RELATED WORKS

2.1 Cloud Federation

Cloud federation is to interconnect two or more cloud computing environments of the same or different service provider, to interact between different types of typologies that can share resources to optimize the quality of services and reduce provisioning costs (Moreno-Vozmediano et al., 2012). In the literature, cloud federation can be defined as centralized or decentralized (Melhem et al., 2017), where the main different is that in a decentralized federation the clouds communicate and negotiate directly with each other. Meanwhile in a centralized federation, such as Intercloud (Elmroth et al., 2009), Contrail (Carlini et al., 2012), Dynamic Cloud Collaboration (Hassan et al., 2010), and Federated Cloud Management (Marosi et al., 2011); central entities act as intermediates for performing tasks such as resource registration, marketing, searching or allocation.

This work is focused on a decentralized federation environment, such as, The European Grid Infrastructure (EGI) Federated Cloud (Newhouse and Brewer, 2012) that is a federation of publicly funded computing, storage, and data resource centers across Europe to provide access to high-throughput computing resources through the EGI Core Infrastructure Platform.

The idea behind it is to offer a cloud infrastructure consisting of a pool of resources and services provided by both public and private partners and presented as a single system. The problem is that the federated resources demanded by the tenants are implemented at the level of the cloud management middleware, making it more complicated to federate clouds that run non-supported middleware or even particular versions of supported middleware that are not compatible with the EGI model.

Another model of cloud federation is Reservoir (Rochwerger et al., 2009), a European project interconnecting computing grids, creating a peer-to-peer(P2P) cloud federation to lease its resources to other members of the federation.

OpenStack (OpenStack, 2019), one of the most used and popular platforms to build private clouds, where a cloud installation can be configured as a service provider and as an identity provider.

- Used as a service provider, a local cloud will trust users that present proof of authentication from a trusted identity provider; such as SAML or OpenID to give authentication and authorization to local clouds.
- When is used as an entity provider, the local cloud can be used to emit authentication tokens that can be taken to a trusted remote service provider.

Fogbow (Brasileiro et al., 2016) is a middleware for cloud federation that has been developed in the context of the EUBrazilCC project, co-funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC). It has the same concept as EGI Federated Cloud, but implements federation at a higher level, being more flexible to interact with cloud providers that are not compatible with the EGI model.

In summary, the actual model of cloud federation are mature in Authentication and Authorization Infrastructures (AAI), virtual machine repositories federation, IaaS interfaces or use of distributed entities for directories or brokering, as we presented in our previous work (Andrade et al., 2017) . But, in Federation of cloud networks is still an issue in cloud computing.

2.2 Networking Federation

Software-Defined Networking (SDN) makes possible to separate the Control Plane from the Data Plane giving us more control to re-programming the network on demand. There are various solutions that provide a tool for cloud network management, such as, Onos, Nox/Pox, OpenDaylight, Beacon, RYU, Floodlight and many more (Khondoker et al., 2014) allowing the traffic orchestration through the OpenFlow protocol (McKeown et al., 2008).

Network Functions Virtualisation (NFV) enable the virtualization of dedicated network components, like routers, firewalls, load balancers, and other components that need dedicated hardware equipment and give them to users as services (Rodriguez and Guillemin, 2016). Also NFV is considerate as a key concept in container-based computing to create overlay network, such as, Flannel (Flannel, 2019), Weave (Weave, 2019), Calico (Calico, 2019), Romana (Romana, 2019); used in well-known platforms for virtualization and resource management,such as Mesos (Mesos, 2019) and Kubernetes (Kubernetes, 2019).

These two technologies allow cloud providers to provide the necessary federated networking capabilities, having better control and performance over the networks, also allow the automation of resource management to improve scalability, quality of services and reduce costs.

2.3 Live Migration

The main advantage of live migration is that give us flexibility, allowing us to move one virtual machine from one physical host to another without disrupting its normal operation, where memory, storage, and network connectivity are transferred. In a WAN network, the live migration faces more challenges than in a LAN-based live VM migration, such as the latency, limited bandwidth, having a shared storage system between the sites that can be a bottleneck, correct IP addresses allocation after the migration.

Solutions such as Silvera et al. (Silvera et al., 2009), they propose to have central agents on both sites to avoid changing the virtual machine IP address in each migration. These agents are responsible to ensure the connectivity using Proxy-ARP and IP-in-IP tunnels to forward the traffic between them.

LayerMover (Zhang et al., 2018) use a duplication technique in three-layer image structure, in order to optimize the performance.

Bradford et al. (Bradford et al., 2007) propose a solution based on DNS-resolutions through IP tunneling. Improving the network performance because the VM machine maintains its canonical names, and the new IP address is registered with the named host.

A unified virtual network to provide a complete view of the LAN keeping a single IP address, for instance, is presented by wood et al. (Wood et al., 2015). Where the idea is to combine virtual Private networks (VPNs), layer 3, and virtual private LANs service (VPLS), layer 2, to provide end-to-end routing across multiple networks and bridge LANs at different locations.

TCP connection before the migration to reduce the total size of the dataset is proposed by Kuribayashi et al. (Kuribayashi, 2013). Eliminating the redundancy on block level and more coarse-grained than compression.

3 ARCHITECTURE

The architecture that has been deployed to recreate a federated network environment is shown in Figure 1. Where we have some resources deployed across three different sites reaching each other through a private overlay network, having two different simulated tenants with full isolation between them and with the underlying network with the same IP segment address (172.16.0.0/16).

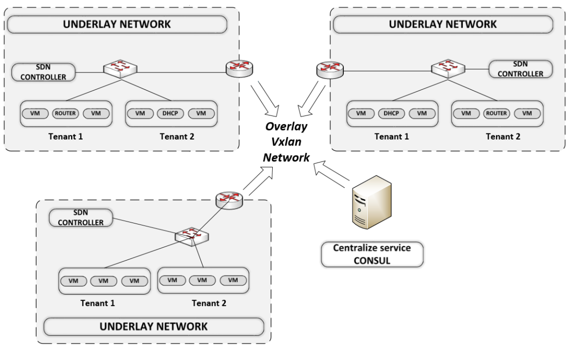

Figure 1: Multi-site Federated Network.

This model of cloud federation architecture enables the access to resources that are not allocated in the same site, such as the Router as a Service - RaaS (to reach to the outside world), the dynamic IP allocation (via DHCPaaS) or the IPfloater tool to serve floating IPs.

It includes a centralized service-based for dynamic infrastructure, CONSUL (Consul, 2019). It is a distributed service mesh to connect, secure, and configure services across any runtime platform and public or private cloud.

Features such as service and node discovery mechanisms to incorporate new Cloud servers, ensure that services are always working and scalable when facing peak demand, health checks with the purpose of providing detailed monitoring and control of services and nodes. Therefore, we can build a sophisticated level of awareness into your applications and services.

The network and resource management are also managed by OpenVswitch among with libvirtd, where the first one is a multilayer software switch, that provides the connectivity among the nodes; as a virtual boundary switch. Enabling massive network automation through programmatic extension and providing logical end-points to the resources (OpenVswitch, 2019). On the other hand, libvirt is used as a wrapper to handle VM and LXC deployed across the federation (Libvirt-project, 2019).

Floodlight (Floodlight-Project, 2019) is an SDN controller designed to work with the growing number of switches, routers, virtual switches, and access points that support the OpenFlow standard; which is an open standard to orchestrate traffic flows in an SDN environment.

To manage the WAN-based live VM migration and reduce problems such as network performance, data migration performance, the continued connectivity, and IP address reassignment; we decide to use NFS (NFS, 2019) as file sharing protocol in combination with OFS (OFS, 2019), also Known as Offline FileSytem. It is is a tool that makes it possible to extend every filesystem with offline capabilities. In other words, we are able to work in offline mode avoiding the RPCs calls made by NFS that can be deadly to performance in WAN network.

In summary, the framework uses these technologies to grant elasticity, reliability, isolation, scalability, monitoring, deployment, releasing and control of services and nodes in our federated cloud network.

4 USES CASES

According to the cloud architecture presented in section 3, we have defined three main uses cases: distributions of VMs, resource management, and network management. Inside of these three scenarios, we derive several uses cases that describe lower-level processes for each scenario which are described in this section briefly.

4.1 Network Management

Combining Libvirt, Consul, OpenVswitch and the Floodlight controller, we can manage our federate network built across the sites. It allows us to create overlay networks, assign the right VLANs for each tenant, manage the complete process of creating and deleting end-points for each tenant (Linux Bridges), define firewall rules and modifies network characteristics such as routing, NAT capacity, firewall rules, and IP assignment.

The set of network management calls that we have defined is the next:

- **Fed_net_create:** It creates a new federated network from scratch from either of the sites, it defines the network configuration (name, UUIDs, VLANs, end-points for the resources, boundary switches creation, overlay network connection, IPs assignment, assigns the corresponding SDN controller and applies the flow rules from fed_net_sdn_controller use case), avoiding the duplication between tenants. The network definition, in most of the process, is carried out by configuration files and replicated in all the sites thanks to the use of key-value storage based on Consul.
- **Fed_net_delete:** Performs the opposite action of the previous use case. It deletes the federated network from all the sites. It also ensures that no more resources are still deployed in the tenant network before and proceed to remove both the entire network configuration and the definition files in our centralized service.
- **Fed_net_list:** Give us a global view of all the resources in the federation using the Libvirt API to retrieve the networks deployed at each site of the federated cloud.
- **Fed_net_sdn_controller:** This function call updates the flow entries, previously defined in a configuration file. It enables to program and reprogram the network on demand.

4.2 Resource Manegement

Libvirt as wrappers to handle Kernel-based Virtual Machine (KVM, 2019), Linux containers (LXC, 2019) and Consul as distributed service, give us complete resource administration across the federated cloud, from the resources definition to the resources removal in an automatic way.

In conjunction with the live migration use case, we can define the deployment, edit, delete and perform WAN live migration across the different sites in the federation. Guaranteeing high availability, elasticity, scalability of the resources and avoiding duplication because it gives us a global view of the resources that are deployed across the federation.

- **Fed_vm_create:** We can deploy resources on demand and verify if the resource has not been created before avoiding duplication. We used predefined templates that define the features of each resource, the packages required according to VM role (Rass, DHCPaaS, DNSaaS, LBaaS), user, password, hostname, IP tables, ssh keys, IPs segments and other information which is updated and contextualized on deployment time.
- **Fed_vm_delete:** This function is the opposite of the previous use case and considers the same criteria as before.
- **Fed_vm_list:** From a single point of our federated cloud, we can have a complete view of the resources deployed in the federation, as an extended LAN network.

4.3 WAN-based Virtual Machine live migration

Having talked about the issues as network and data migration performance, the continued connectivity and IP address reassignment in section 2. We decided to use a file sharing protocol NFS and OFS (Offline FileSytem tool) to avoid the RRPCs calls made by NFS.

- **Fed_vm_migration:** It performs the live migration, from any point of the federation, of a VM across the cloud sites using both tools mentioned before, NFS and OFS.

5 TEST AND RESULTS

As we describe in section 3, the federation has three different sites simulating a WAN network environment, where our WAN will be the Campus network. Accessing the different resources deployed across the federation via a private overlay network.

For the purpose of the test we carried it on two of the three hosts with different characteristics, as we can see in table 1, gauging the time performance in seconds and executing it five times for each lower-level use cases using four different flavors, as shown in table 2, mainly for the resource management use case.

Table 1: Host features.

	Processor	cores	Frequency - (GHz)	RAM
Host A	intel i5-3470	4	3.20	7.7
Host B	intel core 2 Quad - Q9300	4	2.50	7.8

Table 2: Virtual Machine Flavors Description.

Flavor	VCPUs	RAM(MB)	Disk(GB)
Tiny (t)	1	512	2
Small (s)	1	2048	20
Medium (m)	2	4096	40
Large (l)	4	8192	80

5.1 Network Management

Table 3: Average Performance Time - Network Management.

	Host A	Host B	Percentage difference
Network Deployment	4.001	4.186	5 %
Network Removal	1.711	1.953	14 %
Network List	0.658	0.689	5 %
Network Reprogramming	0,366	0.348	5 %

Table 3, shows four lower-level processes for the complete network administration: network deployment (fed_net_create), network removal (fed_net_delete), network list(fed_net_list) and network reprogramming (fed_net_sdn_controller). Revealing that different features of each host did not affect the performance of the framework, because the percentage difference for each use case is between 5% to 14%, which means, a minimum increase of only milliseconds between the two host.

Table 4: Average Performance Time - Resources Management.

	Flavor	Host A	Host B	Percentage difference
Resource Deployment	t	7.280	13.130	80 %
	s	7.828	20.390	160 %
	m	7.986	20.927	162 %
	l	8.249	21.118	156 %
Resource Releasing	t	0.572	1.920	236 %
	s	0.577	2.342	306 %
	m	0.561	2.483	343 %
	l	0.592	2.663	350 %
Full boot	t	108.186	365.306	216 %
	s	119.122	347.542	218 %
	m	111.116	351.260	216 %
	l	142.536	350.180	164 %
Resource List	-	0.658	0.681	1 %

5.2 Resource Management

This use case has three lower-level processes for complete management: resource deployment (fed_vm_create), resource removal (fed_vm_removal) and resource list (fed_vm_list).

The resource deployed for the test was a DHCP server, using Ubuntu Cloud Images (Ubuntu-Cloud-images, 2019). Therefore, it needs to update its repository at the booting time and install some packages for its correct performance, giving us the possibility to gauge the deployment time made by the framework and the time it takes to a full boot, as shown in table 4.

Also, we have to consider the fact that we are deploying a resource that has to reach the gateway in Host B to get access to the internet. Therefore, we have to reduce the MTU slightly in the gateway interface because the traffic has to to go through the tunnel Vxlan and will add an extra header to every packet; as we presented in our previous work (Andrade et al., 2017).

- Resource Deployment: Shows an increase in the performance from 80% to 156%, this means between 6 to 13 seconds more in host B than host A.
- Resource Removal: The performance increase was even greater than in the first use case, with an increase from 256% to 350%. However, if we look at table 4, the increase was just 0.5 to 2.6 seconds respectively.
- Resource List: This use case presented an increase of only 1%, which means nothing relevant, just milliseconds.
- Full boot: This use case shows a big increase in the performance around 230 seconds more than host A.

The test reveals that an increase in three of the four use case, where we can determine that one of the most relevant reasons for this increase is the different feature of each host. Because if we compare each flavor in each host, there is not much increase, just around 0.5 - 1 second, even though each flavor has different characteristics.

Where we can see a consistent increase is the full boot test, due to the slight reduction of the MTU that we had to do at the interface of our gateway. We can conclude that the performance time will concern the WAN network congestion (UPV network) than to the characteristics of each host.

5.3 VMs Distribution

Migration time depends on the resource activity, so if the resource is highly active, the memory transfer process will take longer to complete. For the present work, we have focused only on VM migration, leaving container migration for further work.

We followed the same procedure as (ZHAW, 2014) to analyze the performance of our framework in live migration on WAN. Using a stress tool we could generate a stable memory consumption of around 75%, during migration.

Giving us the opportunity to measure the migration process time, Virtual machine downtime duration and the amount of data transferred across the network during the live migration process between a loaded against unloaded VM.

Table 5: WAN live migration Performance.

	Flavor	Unloades VM	Loades VM	Percentage difference
Migration Time	t	40.298	52.637	31 %
	s	43.509	183	321 %
	m	52.228	196.5	276 %
	l	60.12	207	244 %
VM Down time	t	1	31	3000 %
	s	1	159.5	15850 %
	m	9.6	190.5	1884 %
	l	12	588.5	4804 %
Total amount of data transferred	t	408	514.5	26 %
	s	440	1547.5	252 %
	m	544	2028.5	273 %
	l	623	2040	227 %

On table 5, we can see the three tests to gauge the live migration performance:

- Migration time: Total migration time, in seconds, of a resource from one physical host to another while continuously without disrupting its normal operation.
- VM Downtime: the period during which VM does not reply to ICMP echo request from the remote host. Total downtime is counted by summing of all lost packets multiplied by requests interval. This interval was set up between 2 consecutive ICMP packets to 200 ms.
- Total amount of data transferred: Amount of data transferred from source to destination host during migration process via the network measured in MB. We used the iftop tool (IFTOP-Tool, 2019) to measure transferred data during the migration process.

The results show us a considerable percentage difference for each lower-level use case between a VM with a high CPU and memory usage (loaded virtual machine) and a unload virtual machine. This is because the amount of data to be transferred and the number of messages to synchronize at the moment of the migration are higher in a loaded resource than an unloaded one.

Also, as the previous results for the resource and management use cases, the difference between the smaller and larger images increased; but in a reduced rate concerning the percentage reported in loaded against unloaded resources.

6 CONCLUSION AND FURTHER WORK

According to the rapid growth of the cloud federation paradigm, network federation has become more important and necessary in a federated networks environment. The capability of having NFV services on federated infrastructures as cloud services facilitate the implementation of new use cases, such as geographically-wide workload migration, the definition of priority network paths, confinement of resources to cloud regions due to legal constraints or increased multitenancy in network management.
Current solutions focus mainly on user-based and centralized solutions (such as the set up of VPN servers), the experiments performed in this article show the possibility to have a flexible middleware tool that allows us a complete network federation and at the same time supports VM live migration in a WAN network. Further updates to the framework will allow, first, support to container networks and combining them with OpenVswitch to create overlay WAN networks in a containers environment and second test the framework in an intercontinental scenario integrating the model with the fogbow federation system (Brasileiro et al., 2016).

ACKNOWLEDGEMENTS

This work is supported by SENESCYT (Secretaría de Educación Superior, Ciencia, Tecnología e Innovación - Ecuador) and partially funded by the ATMOSPHERE Project (funded by the European Commission under the Cooperation Programme, Horizon 2020 grant agreement No 777154).

REFERENCES

Andrade, I., de Alfonso, C., and Blanquer, I. (2017). Defining and testing virtual federated networks across multiple cloud sites. In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 107–112.

Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H. (2007). Live wide-area migration of virtual machines including local persistent state. In *Proceedings of the 3rd International Conference on Virtual Execution Environments*, VEE '07, pages 169–179, New York, NY, USA. ACM.

Brasileiro, F., Vivas, J. L., d. Silva, G. F., Lezzi, D., Diaz, C., Badia, R. M., Caballer, M., and Blanquer, I. (2016). Flexible federation of cloud providers: The eubrazil cloud connect approach. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 165–170.

Calico (2019). Calico official website. https://www.projectcalico.org/. Accessed: 2019-01-12.

Carlini, E., Coppola, M., Dazzi, P., Ricci, L., and Righetti, G. (2012). Cloud federations in contrail. In Alexander, M., D'Ambra, P., Belloum, A., Bosilca, G., Cannataro, M., Danelutto, M., Di Martino, B., Gerndt, M., Jeannot, E., Namyst, R., Roman, J., Scott, S. L., Traff, J. L., Vallée, G., and Weidendorfer, J., editors, *Euro-Par 2011: Parallel Processing Workshops*, pages 159–168, Berlin, Heidelberg. Springer Berlin Heidelberg.

Consul (2019). Consul official website. http://www.consul.io. Accessed: 2019-01-10.

Coreos (2019). Coreos official website. https://www.coreos.com/etcd/. Accessed: 2019-01-10.

EC-Managed (2019). Ec managed official website. https://www.ecmanaged.com/. Accessed: 2019-01-10.

Elmroth, E., Marquez, F. G., Henriksson, D., and Ferrera, D. P. (2009). Accounting and billing for federated cloud infrastructures. In *2009 Eighth International Conference on Grid and Cooperative Computing*, pages 268–275.

Flannel (2019). Flannel official website. https://www.coreos.com/etcd/. Accessed: 2019-01-10.

Floodlight-Project (2019). Floodlight project official website. http://www.projectfloodlight.org/floodlight. Accessed: 2019-01-13.

Hassan, M. M., Song, B., and Huh, E.-N. (2010). A market-oriented dynamic collaborative cloud services platform. *annals of telecommunications - annales des télécommunications*, 65(11):669–688.

IFTOP-Tool (2019). Iftop tool. http://www.ex-parrot.com/pdw/iftop/. Accessed: 2018-01-28.

Kaur, K., Kaur, S., and Gupta, V. (2016). Software defined networking based routing firewall. In *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, pages 267–269.

Khondoker, R., Zaalouk, A., Marx, R., and Bayarou, K. (2014). Feature-based comparison and selection of software defined networking (sdn) controllers. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–7.

Kubernetes (2019). Kubernetes official website. https://kubernetes.io/. Accessed: 2019-01-13.

Kuribayashi, S. (2013). Improving quality of service and reducing power consumption with wan accelerator in cloud computing environments. *CoRR*, abs/1302.1921.

KVM (2019). Kvm official website. https://www.linux-kvm.org/page/Main_Page. Accessed: 2019-01-16.

Libvirt-project (2019). Libvirt official website. https://libvirt.org/. Accessed: 2018-01-24.

LXC (2019). Linux container official website. https://linuxcontainers.org. Accessed: 2019-01-15.

Marosi, A., Kecskemeti, G., Kertész, A., and Kacsuk, P. (2011). Fcm: an architecture for integrating iaas cloud systems. pages 7–12.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

Melhem, S. B., Agarwal, A., Daraghmeh, M., Goel, N., and Zaman, M. (2017). Live vm migration across cloud data centers. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 654–659.

Mesos (2019). Mesos official website. http://mesos.apache.org/. Accessed: 2019-01-13.

Moreno-Vozmediano, R., Montero, R. S., and Llorente, I. M. (2012). Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures. *Computer*, 45(12):65–72.

Newhouse, S. J. and Brewer, S. (2012). Egi: An open e-infrastructure ecosystem for the digital european research area and the humanities. In Ioannides, M., Fritsch, D., Leissner, J., Davies, R., Remondino, F., and Caffo, R., editors, *Progress in Cultural Heritage Preservation*, pages 849–856, Berlin, Heidelberg. Springer Berlin Heidelberg.

NFS (2019). Network file system official website. http://linux-nfs.org/wiki/index.php/Main_Page//. Accessed: 2019-01-14.

OFS (2019). Offline file system official website. http://offlinefs.sourceforge.net/wiki/. Accessed: 2019-01-20.

OpenStack (2019). Openstack official website. http://www.openstack.org. Accessed: 2019-01-11.

OpenVswitch (2019). Openvswitch official website. https://www.openvswitch.org/. Accessed: 2019-01-22.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., and Galan, F. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4:1–4:11.

Rodriguez, V. K. Q. and Guillemin, F. (2016). Performance analysis of resource pooling for network function virtualization. In *2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks)*, pages 158–163.

Romana (2019). Romana official website. ttp://romana.io/. Accessed: 2019-01-12.

Silvera, E., Sharaby, G., Lorenz, D., and Shapira, I. (2009). Ip mobility to support live migration of virtual machines across subnets. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, SYSTOR '09, pages 13:1–13:10, New York, NY, USA. ACM.

Ubuntu-Cloud-images (2019). Ubuntu official website. lhttps://cloud-images.ubuntu.com/. Accessed: 2018-01-24.

Weave (2019). Weave official website. https://www.weave.works/. Accessed: 2019-01-11.

Wood, T., Ramakrishnan, K. K., Shenoy, P., der Merwe, J. V., Hwang, J., Liu, G., and Chaufournier, L. (2015). Cloudnet: Dynamic pooling of cloud resources by live wan migration of virtual machines. *IEEE/ACM Transactions on Networking*, 23(5):1568–1583.

Zhang, F., Fu, X., and Yahyapour, R. (2018). Layermover: Fast virtual machine migration over wan with three-layer image structure. *Future Generation Computer Systems*, 83:37 – 49.

ZHAW (2014). An analysis of the performance of block live migration in openstack. https://blog.zhaw.ch/icclab/an-analysis-of-the-performance-of-block-live-migration-in-openstack/.