GSTF Journal on Computing (JOC)
DOI: 10.5176/2251-3043_4.4.346
ISSN:2251 - 3043 ; Volume 5, Issue 1; 2016 pp. 8-14

Container ecosystem based PaaS solution for Telco Cloud Analysis and Proposal

Vinay Murudi
Technology Manager
Huawei Technologies India Private Limited
Bangalore India
vinaym@huawei.com

Krishna M Kumar
Lead Architect
Huawei Technologies India Private Limited
Bangalore India
krishna.m.kumar@huawei.com

Abstract

Telco over Cloud, Network Function Virtualization and Software Defined Networking are changing the telecommunications industry landscape, more specifically in the Telco Service Providers network infrastructure and systems, by introducing cloud computing, virtualization paradigms and software approaches which are already in use and mature in traditional IT environments.

This paper introduces the current telco cloud landscape and latest developments. It subsequently proposes a container based telco app orchestration mechanism. The shift of the telco cloud landscape towards containers is imperative as the traditional VM based NFV and SDN solutions are running into scalability and performance problems and have an impact on delivery speed and efficient resource utilization.

In the solution we have derived in our lab uses a cluster container orchestration mechanism using Apache Mesos. A custom framework is developed to handle the Telco specific (NFV) capabilities on top of the traditional containers. This novel approach will help telcos to provision tons of containers in a span of short duration adhering to the QoS Requirements of the industry.

I. INTRODUCTION (TELCO CLOUD CASE)

The telecom industry has been undergoing a transformation process for much of the last decade. The threat from new players has marginalized the core communications business and operators have looked to gain traction and grow revenues through the provision of new services in adjacent areas, with one such area being cloud computing.

Cloud computing has ripped through the traditional IT infrastructure model, providing greater flexibility, enabling the pooling of resources and potentially reducing both capex and opex. This new delivery model has led to the development of new services and business models (e.g. 'as-a-service' models), disrupting how individuals consume services and how organizations do business.

Cloud services are emerging as a key strategic imperative for Telcos as revenues from traditional services such as voice, messaging and data come under attack from Over the Top Players, regulators and other Telcos. A majority of these new products are delivered from the Cloud on a "pay for consumption" basis and many business customers are increasingly looking to migrate from traditional in house IT systems to Cloud-based or virtualized services to reduce costs, increase agility and decrease deployment times.

Gartner recently estimated that the Cloud services market would be worth over $200 billion by 2016; roughly double the value of 2012 and with a CAGR of around 17% whereas traditional IT products and services will see just 3% growth.

It is clear that some Telcos have gained a greater understanding of the Cloud market, and based on the action of that understanding, offering increasingly rich Cloud-based products and services, paving the way for Cloud 2.0. But for most Telcos, Cloud services remain secondary to their core business of voice and data delivery. Telcos are wrestling with issues of reduced margin on Cloud and how to stay relevant to their business customers.

This transformation involves the virtualization of the network, embracing software defined-networking (SDN) and network functions virtualization (NFV). As operators harness the power of these new technologies and associated business practices they will develop and implement the infrastructure, software and capabilities to deliver more advanced services through more efficient, automated and programmable networks. Operators in turn will be able to draw on these assets and associated skills to improve how they run and manage their cloud infrastructure.

Furthermore, as the industry develops and implements more advanced networks (i.e. 5G), there exists a potential advantage for telco infrastructure services due to the need for more localized delivery of service. The Next Generation Mobile Networks (NGMN) Alliance highlights that 5G should provide, "much greater throughput, much lower latency, ultra-high reliability, much higher connectivity density, and higher mobility range."

Modern virtualization technologies, such as software-defined networking (SDN), in combination with existing tools for storage and computing, ensure virtualization and abstraction for the entire set of critical resources. The general aim of SDN and NFV is to deliver functions, networks and infrastructure as services rather than as features of vertically integrated systems. This enables operators to offer communication services at the right price points for subscribers, by serving the next generations of terminals like high-end smartphones, tablets, and 3D glasses. In addition, this enables operators to offer low-price connectivity to service providers of M2M communication, by supporting devices like utility meters and health care equipment.

Telco's have following advantages and motivations to embrace cloud

- Opportunity to leverage existing sales relationships with enterprise customers: Many telcos have already been providing services such as e-mail, managed hosting and storage solutions to their enterprise customers. The evolution to cloud computing is expected to be a natural progression for them. In addition, as they already have long-term relationships with enterprise customers, they are at an advantage over other service providers
- Ability to offer end-to-end service-level agreements to customers: As telcos control the networks through which cloud services are offered, it enables them to offer end-to-end service-level agreements (SLAs) on application performance and availability. With their combination of service assurance and data center and network redundancy, telcos are also better positioned to offer secure cloud services to enterprise customers
- In-house usage-based billing capabilities: Cloud computing is characterized by usage-based billing as users only pay for services they actually use. Most cloud service providers find it difficult to run their billing infrastructure in-house, due to technical and cost challenges involved in building and running a pay-per-use billing infrastructure. Telcos are at an advantage, as they already have the charging and billing capabilities that are required for usage-based billing in a cloud environment. In addition, telcos can utilize their experience in pay-per-use billing to provide cloud billing solutions for other cloud vendors.

II. CURRENT CLOUD COMPUTING TYPICAL ARCHITECTURE

Following figure presents an overview of the NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing. The diagram depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

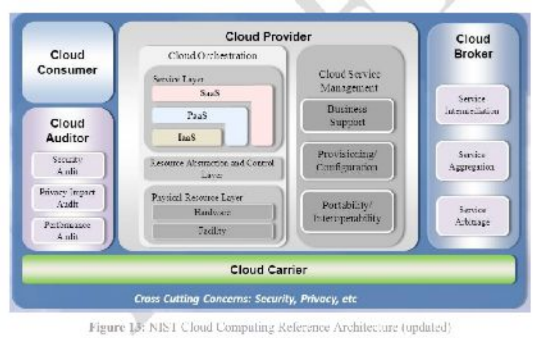

Figure 1. NIST cloud computing architecture

One of the examples of evolution of telco platforms towards leverating cloud platforms would be SDP over PaaS, the discussion of this example would set the context for driving further points of this paper.

Telecommunication SDP is the open, horizontal, standards based platform that enables service providers to create and deliver new, innovative services. To attract and serve an expanded customer base, telecommunication service providers are moving from voice-centered legacy network service to services that target the "long tail" across industries and communities. Services will be more targeted, focused, and personalized. This is a challenge for business models and technologies to develop and consume such value added services. Telecommunication networks and the Internet have evolved separately with regard to software applications and application development technologies.

For SDP, the cloud, with low cost for value added services and Platform-as-a-Service (PaaS), is the best way to host solutions. The cloud provides the open development community and business models for telecommunication services. A new range of services and applications are delivered to customers through several business models, though the PaaS model is the most prominent.

The PaaS model is a novel approach for software suppliers that want to focus primarily on the software development cycle and the monetization of new applications. They can bypass the investment in, and maintenance of, the underlying infrastructure and services for application design, development, testing, deployment, and hosting. PaaS creates a virtual platform for application development and deployment. With PaaS, the system's provider makes most of the choices that determine how the application infrastructure operates. Users build their applications with the provider's on-demand tools and collaborative development environment.

PaaS enables a centralized cloud-computing model where different roles in the ecosystem are magnetized around value added services into a centralized hosting environment. Roles include:

- Telecommunication operator
- Partner for development of value added services

- Enterprises using telecommunication related applications
- Individual customers

PaaS fits a double-sided market model where, in a platform, business is mediated between the demand and supply side. It creates value for upstream developers and downstream customers by extracting revenues from both sides. Telecommunication service providers will benefit from this new business model by achieving: low cost, openness, faster time to market, rich applications, and rich approaches for increasing revenue. Since the telecommunication operator provides the environment for the entire software cycle, developers can reduce the cost of provisioning and managing their own IT infrastructure. Runtime fees will be their primary costs, based on the actual computing resources used by applications or the number of users who are using them.

However it is also noteworthy that current PaaS ecosystem is largely IT industry driven lately momentum is gaining for telco apps to leverage PaaS ecosystem. This poses a challenge for defining and adopting telco apps to be able to seamlessly integrate into current PaaS solutions – for example enabling telco apps to be PaaS ready on the lines of 12 factor apps.

Further despite the appeal of cloud environments and the opportunities enabled by PaaS, many service providers are reluctant to take the first step because application migration and deployment in the cloud bring challenges that are unique to carrier environments. Most PaaS tools available today address the basic operational challenges associated with building, deploying and maintaining applications, but lack the carrier-grade attributes that service providers require.

III. Challenges in Migrating Telco Apps to Cloud and Telco Adoption of Cloud

Although telecoms operators have often talked a good game when it comes to offering enterprise cloud services, most have found it challenging to compete with the major dedicated and Internet-focused cloud providers like Rackspace, Google, Microsoft, and most of all, Amazon Web Services. Smaller altnets and challenger mobile operators – and even smaller incumbents – have struggled to find enough scale, while even huge operators like Telefonica or Verizon have largely failed to differentiate themselves from the competition. Further, the success of the software and Internet services cloud providers in building hyperscale infrastructure has highlighted a skills gap between telcos and these competitors in the data centre. Although telcos are meant to be infrastructure businesses, their showing on this has largely been rather poor.

There is an argument that the more telcos bought into pre-packaged technology solutions from vendors like VMWare, the less control over the future development path of their software they would have, and the more difficult it would be for them to differentiate effectively

Relying heavily on third-party proprietary technology solutions for cloud would give telcos a structural disadvantage relative to the major non-telco cloud players, who either develop their own, or contribute to fast-evolving open-source projects.

IV. Telcom App Current Scenario – PaaS

Any PaaS based solution for Carrier should be a solution which is focused on both productivity and control. It should combine cloud advantages with the service providers' own distributed network footprint and knowhow to address the unique challenges and requirements service providers face in building and deploying network applications and appliances in a distributed, multi-tenant and automated environment.

A carrier PaaS solution must also offers service providers five main advantages over enterprise grade PaaS tools:

- Application and infrastructure neutrality
- Automation and scalability
- Security and resiliency
- SLA compliance
- Regulatory and business rules compliance

Among the most commonly prevalent deployments currently, an operator network can be described as a set of solutions, such as EPC or IMS/ VoLTE, split over multiple organizational domains. In turn, these domains consist of sets of telco applications that interconnect using site infrastructure, such as switches, routers, firewalls and transport networks.

When a network is deployed virtually, organizational domains within the operator becomes a tenant of the shared cloud infrastructure, which provides virtual data centers (VDCs) that serve as resource containers and zones of isolation. Operator solutions are deployed in VDCs and telco applications and nodes become telco virtual network functions (VNFs) – implemented as portable containers for multiple virtual machines (VMs).

V. Growing Traction of Containers in PaaS and Telco Cloud

PaaS generally provide mechanisms for deploying applications, designing applications for the cloud, pushing applications to their deployment environment, using services, migrating databases, mapping custom domains, IDE plugins, or a build integration tool. PaaS have features like built farms, routing layers, or schedulers that dispatch workloads to VMs.

A container solution supports these problems through interoperable, lightweight and virtualised packaging. Containers for application building, deployment and management (through a runtime) provide interoperability. Containers produced outside a PaaS can be moved in the container encapsulates the application. Existing PaaS have embraced the momentum caused by containerization and standardized application packaging driven by Docker. Many PaaS have a container foundation for running platform tools

Currently the evolution of PaaS is moving towards container based, interoperable PaaS.

- The first generation was made up of classical fixed proprietary platforms such as Azure or Heroku.
- The second generation was built around opensource solutions such as Cloud Foundry or OpenShift that allow users to run their own PaaS (on premise or in the cloud), already built around containers. Openshift moves now from its own container model to the Docker container model, as does Cloud Foundry through its internal Diego solution.
- The current third generation includes platforms like Dawn, Deis, Flynn, Octohost and Tsuru, which are built on Docker from scratch and are deployable on own servers or on public IaaS clouds.

Open PaaS like Cloud Foundry and OpenShift treat containers differently, though. While Cloud Foundry supports stateless applications through containers, stateful services run in VMs where as Openshift does not distinguish these

Further Container technology is becoming increasingly important to major telecom service providers as they move to be more applications-focused and to maximize the benefits of virtualization by moving more overhead into the cloud.

Containers essentially create a package around a piece of software that includes everything it needs to run, including its code, runtime and systems tools and libraries. The software can then run the same way in different operating environments. That fits neatly into the telecom operators' movement from an infrastructure focus toward an application one in response to customer demand. The telco vendors are looking for an unprecedented visibility and level of control; secondly they are looking for elasticity or scalability not so much at the server level but scalability that aligns with their business functions. Container technology enables the telecom operators to support those applications, and virtual network functions, in software packages that can run where they need to.

In 2016, enterprises will embrace containers and start shifting production workloads onto them. They'll also start more projects to "containerize" their legacy apps with similar enthusiasm as when they visualized their legacy apps from bare metal.

Furthermore, most of the development effort on software managing containers in 2016 will be spent on two things:

- Security of containers so they can be audited and verified that they have not been in no way altered or tampered
- Deployment tools which will integrate into IDE making deployment of code into a container as simple as one click

The traction towards containers is natural as Container technology allows software to be distributed across a network or data-center in small packages—the "container" part—and share operating system resources, rather than requiring regular software updates or a virtual machine for each instance of the software. The container and microservices paradigm can be thought of as small software packages that can travel around the network and talk to each other on a more real-time basis.

This can help with scalability, especially with network functions virtualization (NFV), wherein thousands of virtual machines (VMs) might have to be created to scale up a service. Container-based software and microservices can be distributed to run on top of a shared operating system, without requiring a virtual machine.

In theory there is so much potential to leverage the power and scalability of containers. For instance, one can set quality of service for different tenants, or scale bandwidth for different tenants

Containers might first gain traction in upper layers of the telecom software—for example, operations and support systems (OSS) or services orchestration—because they are better for dynamic software upgrades and real-time data exchange

Typically, VNFs are deployed in virtual machines that sit on top of a hypervisor, but Linux containers – which can coexist with virtual machines on some virtualisation platforms – use a different structure that puts isolated Linux systems (containers) on a single Linux control host, and allows more granular resource isolation and greater elasticity.

Telcos and cloud service providers are keen to exploit NFV because it has the potential to reduce reliance on expensive proprietary networking gear and increase network elasticity

VI. A Container Based NFV Orchestration

As we have explored, the containers are gaining momentum in PaaS and as well as telcom cloud scenarios – one obvious question need to be addressed is Why Containers for NFV. Some of the obvious advantages are

- Instant booting and Deployment: Very quick deployment and Rapidly deploy applications with minimal run-time requirements
- Low latency
- Low overhead : No virtualization overhead and Lower use of system resources (CPU, memory, etc.) by eliminating hypervisor & guest OS overhead
- Lightweight footprint: Very small images with API-based control to automate the management of services
- Established provisioning and management tools
- Updates: Depending on requirements, updates, failures or scaling apps can be achieved by scaling containers up/down

Following diagram depicts typical Container stack for NFV as compared to VM.

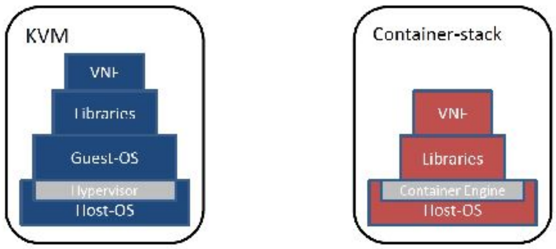

Figure 2. Conatainer Based NFV

As Docker being the most popular container technology using currently, we will project a NFV Orchestration with Docker. In this model each Function typically runs on a Container, which could be modeled on micro services also. Micro services can be deployed in containers and seamlessly communicate each other across various functions. This way thousands of end points must be able to access by several customers. As and when new tasks are needed, it will be able to provision tasks in new containers and execute fast to finish the job taking care data locality, I/O speed, real time processing, etc.

While we have explored various container orchestration technologies like CoreOS rkt with tectonic, Docker Swarm, Google Kubernetes, Cloud Foundry, etc. but Apache Mesos seems to be more adaptable and easy to handle multiple tasks for our own telecom software stack implementation. Mesos not only supports resource scheduling efficiently but also can handle all kinds of application orchestration with various kinds of underlying hardware stacks. There are company specific platforms like Ericson's APcera & MetaSwitch Calico which are highly customized for their needs.

We have used Mesos as our orchestration engine to provision containers. By the power of Mesos Resource Scheduling, large pool of containers can be provisioning in seconds.

The goal of Apache Mesos is to provide an abstraction layer for compute resources (CPU, RAM, Disk, Ports, etc.). This allows applications and developers to request arbitrary units of compute power without an IT provider having to worry about how this translates to bare-metal or VMs. Mesos democratizes computational resources.

The Mesos process runs on all machines and one machine is defined as a master, which has the responsibility of managing the cluster. All other machines are slaves which work on a particular task using a set number of resources. Each slave utilizes linux's cgroups to ensure that processes are isolated and are only allowed to consume a set amount of resources. The combination of the master and a set of slaves create a Mesos cluster.

It is noteworthy that Containerization is a concept at the heart of Linux's cgroups. Containers provide a clean, packaged, decoupled, cohesive abstraction that yields many positives. And Mesos has rich support and ecosystem built around Containers.

Following are the major advantages of Mesos as compared to other orchestrators:

- Testability: By representing the application as a framework, it becomes possible to test the system as a whole. Previously it has been difficult to test clustered systems because of lack of standards and requirement for a cluster.
- Scalability: Many applications can benefit from accessing more detailed information about the state of a task. Horizontal and vertical scaling is now possible on a cluster of machines.
- Resiliency: If a task is lost due to a failure or a crash, Mesos is able to reliably inform the framework of the inconsistent state and the framework can take action as necessary. Details about the task allow the optimal placement of tasks.
- Democracy: By using Mesos authorization and roles, it is possible to reserve and control how resources are used. Priorities could be attached to allow important jobs priority access to limited resources. The cluster would be under developer control and the infrastructure can be left to IT or removed entirely.

More on Mesos can be found at here http://mesos.apache.org/.

A sample Mesos Architecture is shown below for a general understanding of the subsequent sections (source internet site).

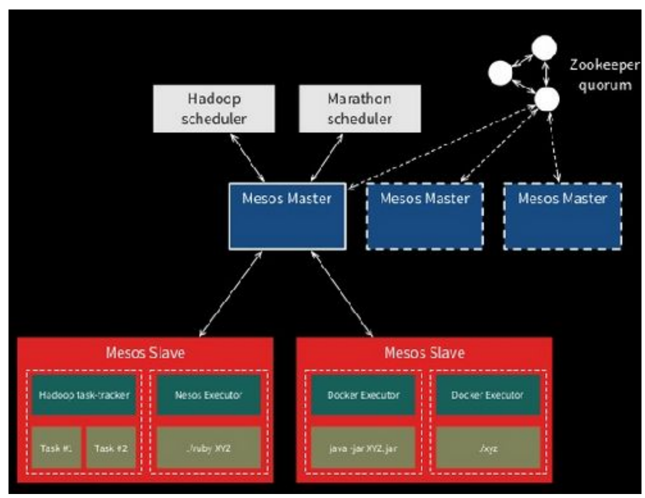

Figure 3. Mesos Architecture

Working across hosts in a clustered environment with across the communication on multiple containers, Docker native networking is the simplest way, though it has lots of challenges. Though beyond the scope of Mesos to try to address the concerns of networking setup, topology, performance, etc. Mesos can ease integrations with existing networking solutions and enable features, like IP per container, task-granular task isolation and service discovery.

The networking support is enabled via a Mesos module and thus the Mesos master and agents are completely oblivious of it. It is completely up to the networking module to provide the desired support.

Next, the IP requests are provided on a best effort manner. Thus, the framework should be willing to handle ignored (in cases where the module(s) are not present) or declined (the IPs can't be assigned due to various reasons) requests.

As Containers are created and deleted much faster pace, traditional multi-layer network may not be efficient (like Openstack VM). So interconnect IP endpoint and put policy abstraction on it so that applications can be connected through micro services and the work load can be run in any place.

Mesos Marathon framework support Docker container provision at a rapid pace. We will make use of this feature to exploit the NFV orchestration. Like any telco app, Mesos with marathon framework can provision hot/new/cluster as needed with application versioning, authentic scaling and automatic testing.

Marathon can be used to start other Mesos frameworks, and it can also launch any process that can be started in the regular shell. As it is designed for long-running applications, it will ensure that applications it has launched will continue running, even if the slave node(s) they are running on fails. More on Marathon is available at github as shown below https://github.com/mesosphere/marathon.

On top of Marathon, we will have a custom framework to enable the NFV layers per Container. This gives good flexibility in container to container networking. Framework can request IP address at task launch time.

Marathon Framework based Docker containers orchestration can bring a standard image of the Docker with NFV already configured to the network function. This is an easy way using the orchestration. However, Custom framework can be used for further fine tuning of the network functions as desired. This not only helps bring the containers as needed but also adds specific characteristics for the NFV operation.

Virtual Network Functions are very complex – many tiers, firewalls, static networks, Geographic location, SLAs, etc. Hence on the fly if such a VNF has to be implemented, then it is better to have a custom Framework to handle this job.

One of the typical use case involve VNF in a container connect to the other container using specific network and subnet, or in addition some of the Open source technologies like Openstack, ODL used in order to orchestrate the NFVs.

Container to container direct networking with special functions can be orchestrated with the complementary technologies like OpenStack Kuryr. Also one could combine these with Container/Docker technologies for NFV or SDN for fine tuned network stack.

Further there is a provision to write custom network isolator module. The module is loaded as a dynamic shared library in to the Mesos Agent and gets hooked up in the container launch sequence. A network isolator may communicate with external IPAM and network virtualizer tools to fulfill framework requirements.

In addition the custom Framework the Policy Engine and Configuration Scripts can be used to control the containers characteristics. The highly available custom Framework, Mesos by native support the HA, which will handle telecom requirements for managing containers 24 x 7 and other availability requirements. As more and more network functions becomes software, micro services managing large number of containers are taken care efficiently by the Mesos setup.

Applications can discover each other with Service discovery. There are two types of service discovery in Mesos:

- DNS Based (mesos-dns)
- HA Proxy based (marathon-lb).

The above model shows the DNS based approach. Services like IoT, Video Streaming, Mobile apps and wireless network functions, etc. can be hosted on it and discovered through the service discovery.

The illustrated Diagram below shows a Docker provisioned Model with marathon and Custom Framework. The top figure shows the Mesos networking and the bottom figure shows the Framework mapping for custom provisioning.

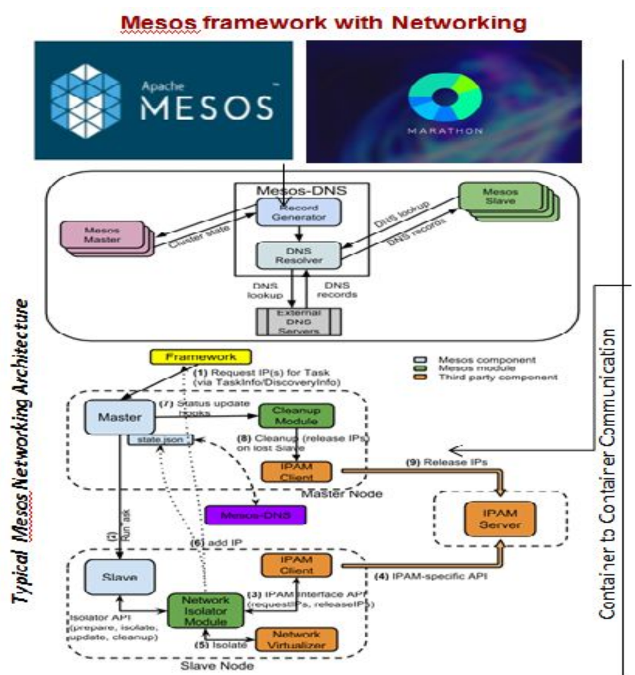

Figure 4. Mesos Framework with Networking

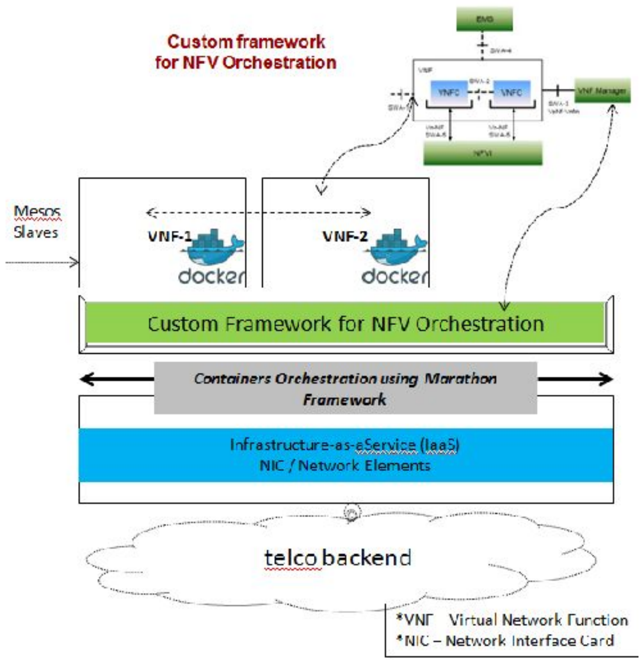

Figure 5. Mesos Custom Framework for NFV Orchestration

We have modeled the standard NFV orchestration within a container as highlighted in the diagram above. NFV is the software implementation of the Network Function. The functions are accessed through end points and there could be nested services in the case of multiple functions case.

Typical operations include instantiation, update, scaling, query, monitoring, fault diagnostics, healing, and termination. Mesos by native supports several of these functions and on top of that custom Framework can manipulate to handle all the necessary items.

For the multi cluster large scale deployment the higher layer management tool composed of VNF Manager and Orchestration Manager will help to monitor and understand the health of the deployment.

Telecom industry needs predictable performance, Reliability and security as some of the key components of deployment. Typical NFV use case is virtualization of CDN (all network functionality will be virtualized). There are three ways the things could be accomplished:

- IaaS – Container based infrastructure for NFV
- PaaS - Virtual Network Platform
- SaaS – Network function access

The model we mention above could falls in to any of the above three; but more inclined towards the the PaaS layer as standard APIs between modules can be used to manage the entire application flow. Also the custom framework could be enhanced further to handle much more sophistication like QoS, etc. as needed.

In addition to the custom framework mentioned above, we also have created a framework for caching service (Redis framework for Mesos) to speed up the provisioning of large clusters for high speed data retrieval. Enabling caching with Mesos, we can seamlessly and efficiently manage the cloud resources required by a globally scalable NoSQL database, along with all kinds of telecom apps like IoT enablers, Mobile wireless, etc.

OpenNFV has done some good work in terms of creating overall NFV framework by building NFV infrastructure (NFVI) and Virtualized Infrastructure Management (VIM). NFV & SDN are complementary technologies, the way things are described here are keeping in NFV focus. Similar approach can be used to implement SDN solutions in containers with some customization as needed which is supported by our Framework. SDN on containers are under exploration in our lab and further study will be published later.

CONCLUSION

Spearheaded by the IT industry, a shift toward providing everything as a service is taking place. Adopting a similar approach for network functions is one of the ways for telco operators to expand the infrastructure and offered services.

This paper presents a model based on container technologies which shall enable telcos to adapt to cloud ecosystem well and leverage on its advantage, hence meet ever growing demands of innovative services by subscribers and customers.

The approach we mentioned is a novel idea using the Apache Mesos Framework to provision customized NFV containers to scale the system for telcos.

In this approach, we can make use most of the standard technology available with Mesos and need only some minor specific customization to meet our demand. Further scope of the project is been explored in our lab including integrated SDN/NFV solutions at containers ranging 10000 and above.

REFERENCES

[1] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf " NIST Cloud Computing Reference Architecture"

[2] Benjamin Hindman, Andy Konwinski, Matei Zaharia,A li Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica "Mesos: A Platform for Fine-Grained Resource Sharing inthe Data Center"

[3] D. Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes," IEEE Cloud Computing,vol.1, no. 3, pp. 81-84, 2014.

[4] N. Kratzke, "A Lightweight Virtualization Cluster Reference Architecture Derived from Open Source PaaS Platforms", Open Journal of Mobile Computing and Cloud Computing vol. 1, no. 2, 2014

[5] Telco and Cloud Service Provider Predictions for 2016 -https:// www.flexiant.com/2016/01/07/5-telco-and-cloud-service-provider-predictions-for-2016/

[6] Mesos – http://mesos.apache.org/.

[7] Mesosphere - https://mesosphere.com/

[8] SDN - https://www.opennetworking.org/

[9] NFV – http://opennfv.org

[10] Docker – http://www.docker.com/

[11] Telco Cloud - http://www.telco2research.com