Virtualization and Edge Compute Evolution in Cable

A Technical Paper prepared for SCTE•ISBE by

Andrii Vladyka
Technical Product Manager, Cable Access
Harmonic Inc.
2590 Orchard Parkway, San Jose, CA 95131
+1 408 542 2559
Andrii.Vladyka@harmonicinc.com

Asaf Matatyaou
Vice President, Solutions and Product Management, Cable Access
Harmonic Inc.
2590 Orchard Parkway, San Jose, CA 95131
+1 408 542 2559
Asaf.Matatyaou@harmonicinc.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

1. Introduction

Traditionally, the CMTS market was dominated by hardware-based CMTS appliances, with no alternatives available. During the last few years, however, the cable industry took a leap forward and started adopting software-based CMTS solutions[4]. Disruptive benefits, long-term potential and deployment scenarios of such software-based CCAP solutions are analyzed in detail in References 1 and 2, while Reference 3 outlines real-world experiences and lessons learned from deploying virtual CMTS in centralized and distributed access architecture (DAA) environments.

This paper analyzes the background and deployment scenarios of a virtualized CMTS (vCMTS) and considers future development of disaggregated vCMTS from the standpoint of leveraging distributed edge compute. We also explore the notion of cluster multitenancy and its potential deployment scenarios in private, public and hybrid cloud.

Here and throughout the paper, we adhere to the following terminology:

- Cluster – a set of compute nodes that can be viewed as a single system. Cluster nodes are connected with each other via a common converged interconnect network (CIN) and managed by specialized platform management software.
- Pod – a group of one or more software containers, orchestrated by Kubernetes[16].
- Infrastructure (infra) pod – contains applications that perform a management role in the system. The resources and functions provided by infra pods are typically shared by multiple data plane pods. The examples of infrastructure pods are pods running databases used for storing internal system states as well as pods providing interfaces for interaction with users (e.g., command-line interface (CLI), web UI) and toward OSS/BSS applications (e.g., via SNMP, IPDR and other protocols).
- vCMTS pod – a collection of containers implementing DOCSIS[14] processing (excluding the PHY layer).
- Data plane pod – a more generic term for the vCMTS pod. A data plane pod contains software components performing data plane traffic processing for an access technology of choice (e.g., DOCSIS, PON, wireless). It should be noted that PHY layer processing is implemented in purpose-built hardware appliances (e.g., DOCSIS 3.1 Remote PHY Device) tailored for communication with the user equipment (e.g., DOCSIS 3.1 cable modem) over specific media type (e.g., coax cable or optical fiber).
- Multitenant cluster – sharing compute resources for infrastructure pods and data plane pods serving different types of applications (e.g., DOCSIS, PON, wireless), while providing required level of resources and components isolation.
- Hybrid cluster deployment – combining local compute resources and compute resources from cloud service provider (CSP) in the same logical cluster.
- Virtualized cloud-native access platform – software platform that follows the cloud-native paradigm[15] and provides facilities for life cycle management of compute resources and software applications running on top of them. This type of platform is agnostic to the type of applications (e.g., data plane pods) deployed on top of it.

In the subsequent sections, we look in more detail at a range of options that become available to operators when migrating to a cloud-native vCMTS.

2. Cloud-Native vCMTS Deployment Evolution

CMTS is not the only product that transformed dramatically over the last few years. Studies[11] show that both public and private cloud adoption grows, with larger enterprises increasing their focus on public cloud. Moving services to the public cloud entails an operations paradigm shift. It is possible to incorporate software-based vCMTS deployment and operations into cloud-centric business processes and automation tools. If done properly, it allows operators to take service agility to the next level, dramatically reducing time to market for rolling out new services and transforming the CMTS from a purpose-built and rigid resource in a headend to an on-demand scalable service.

Due to its disaggregated architecture, cloud-native vCMTS provides full flexibility when selecting deployment type – from on premises to public cloud (Figure 1 - Cloud-Native vCMTS Deployment Scenarios).

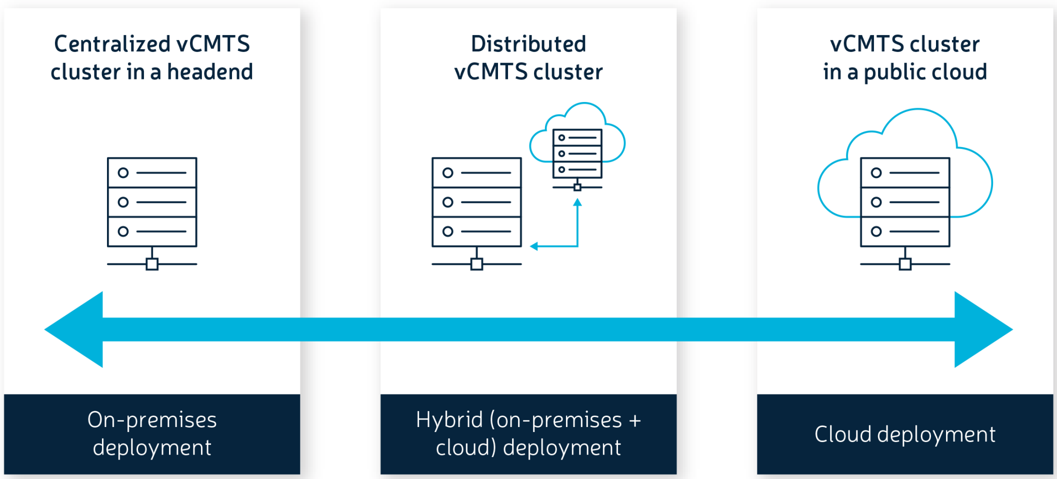

Figure 1 - Cloud-Native vCMTS Deployment Scenarios

Another dimension to look at is types of access technologies that can be deployed on the same cluster and managed in a uniform way: i.e., cluster compute resources that can be shared between different workload types (e.g., vCMTS data plane pods and virtual PON (vPON) data plane pods). It introduces the notion of cluster multitenancy, where the same virtualized cloud-native access platform accommodates data plane pods for different types of access technologies. We will use the more generic "access platform" term throughout the paper when it is required to refer to the workload-agnostic nature of the deployment type.

In the following sections, we look into different scenarios of disaggregated vCMTS deployment, from on-premises to public cloud.

2.1. Edge Deployment

One side of the spectrum of options when deploying cloud-native vCMTS is running it on dedicated compute and management cluster nodes located in the headends and remote hubs, the same way it is done for legacy hardware-based CMTS appliances.

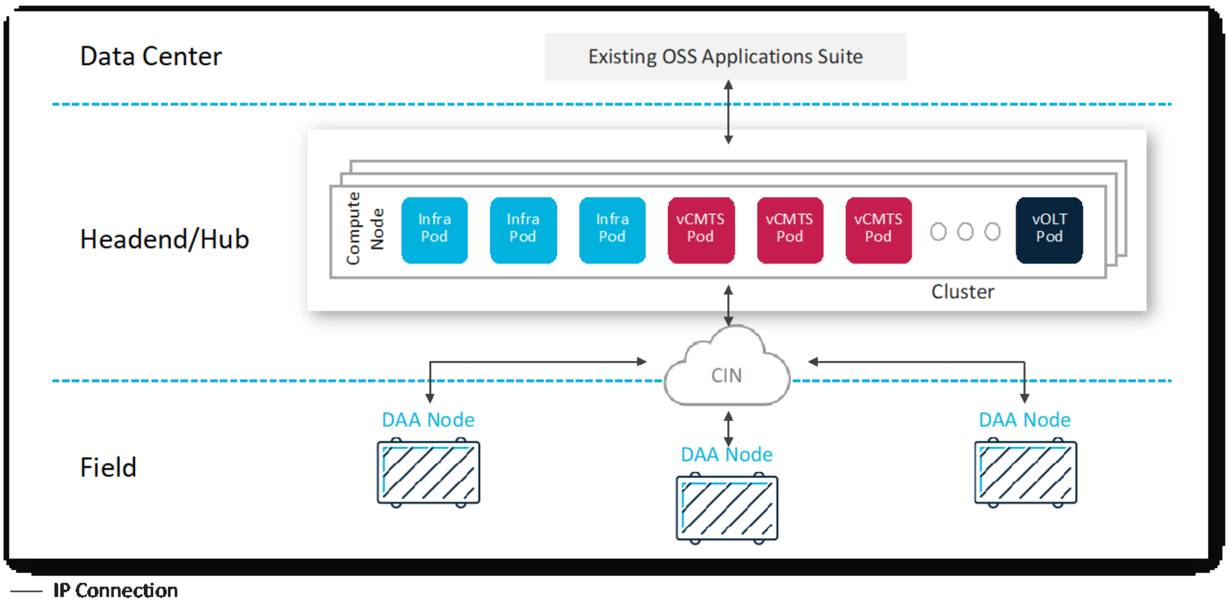

Figure 2 - Disaggregated Virtualized Cloud-Native Access Platform as Stand-Alone Edge Compute Cluster

In this scenario, individual clusters work independently of each other, and while cluster scale (e.g., number of compute nodes) can vary from site to site, every cluster acts as a single deployment unit. All resources required for CMTS functionality (e.g., data plane workloads, generic control plane (GCP) Core, interfaces to OSS applications) are unique per cluster. While at first sight it looks like yet another CMTS, there are clear benefits of migrating to a cloud-native architecture, even in a stand-alone edge deployment:

1. Increase reliability by leveraging cloud-native best practices and tools for high availability (HA).
2. Reduce operating domain to one or several service groups by isolating vCMTS resources with Docker containers.
3. Solve scaling issues by simple addition of compute nodes, with growth in bandwidth as well as the number of subscribers and/or service groups.
4. Significantly reduce power and cooling requirements, space footprint and wiring complexity in the headend.
5. Ability to share cluster compute resources between different types of workloads (data plane pods). As such, the same cluster can accommodate, for example, vCMTS and vPON pods, while new emerging types of data plane pods might be added to a cluster in the future.

Horizontal scaling in stand-alone edge deployment is achieved by scaling the number of vCMTS pods (or other data plane pods) on existing compute nodes, as well as by adding more compute nodes proportionally to the number of connected devices. The contributing factors into cluster scaling are:

1. The number of connected Remote PHY Devices (RPDs).
2. The number of connected Cable Modems (CMs) and Customer Premises Equipment (CPE) units.
3. Peak traffic load.

Cluster scaling implies increasing the number of data plane pods, while the number of infra pods remains constant. Keeping the amount of infra resources constant with the growing number of data plane pods further improves the efficiency of a virtualized cloud-native access platform.

Such a deployment scenario is most suitable for cable operators who would like to follow the traditional way of operating CMTSs and do not plan to take steps toward building network function virtualization (NFV) infrastructure and/or network automation or run some of the vCMTS components in the hybrid cloud.

2.2. CMTS in a CSP Infrastructure

Another potential scenario of cloud-native vCMTS deployment is to completely decouple it from hardware resources and run it in a public cloud on compute resources provided by a CSP.

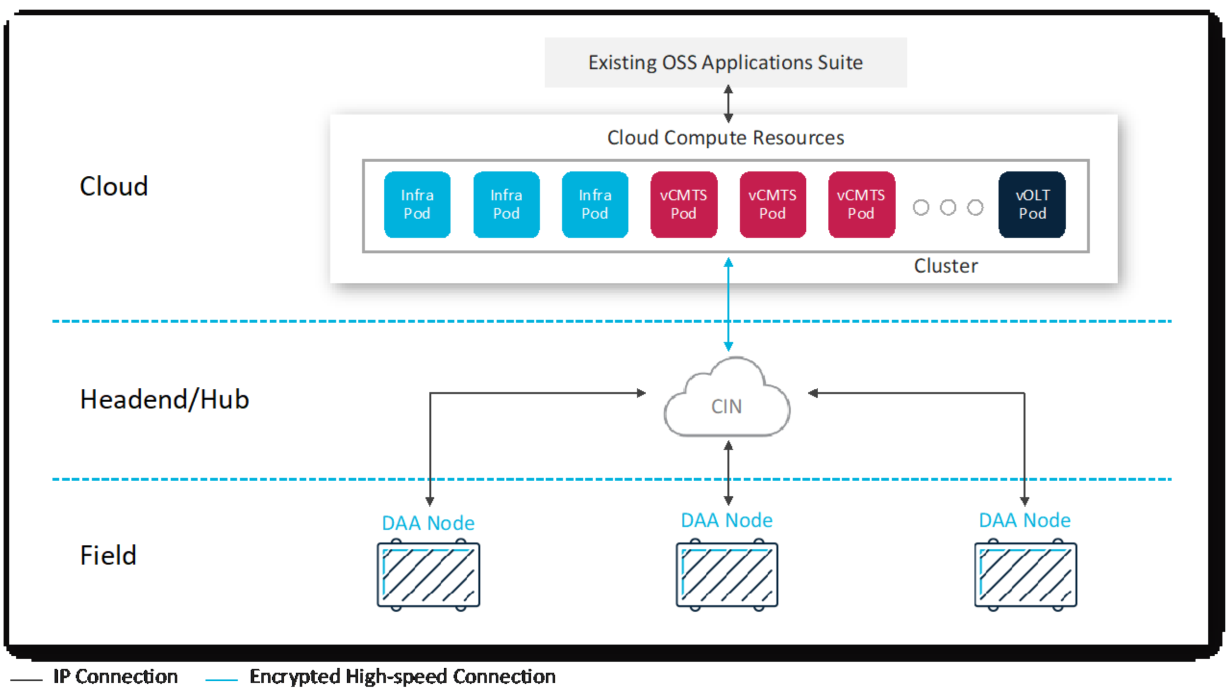

Figure 3 - Disaggregated Virtualized Cloud-Native Access Platform in Public Cloud

This type of deployment implies that all control plane and data plane vCMTS components run on compute resources in the public cloud. Remote PHY protocol (e.g., downstream external PHY interface (DEPI), upstream external PHY interface (UEPI)) tunnels are established between the vCMTS data plane workloads and R-PHY nodes, and all data processing is performed in the public cloud. The operations support systems (OSS) applications suite may remain in the operator's data center, or it can be deployed in the public cloud as well. The factors that contribute to the decision-making process are:

1. Network total cost of ownership (TCO) optimization.
2. Migration path for legacy OSS applications.
3. Regulations compliance.

Such a solution requires reliable and secure connections between the R-PHY nodes and vCMTS components running in the cloud. It is achieved in the following ways:

1. Using facilities provided by CSPs, such as encryption gateways that enable secure high-speed data transfer between the operator's data center and public cloud infrastructure[5].
2. By leveraging a suite of security features defined for RPD and CCAP Core Mutual Authentication[6] and DOCSIS Baseline Privacy[7].
3. Additional security measures might be applicable to other types of access technologies deployed on a multitenant cluster.

Moving all data and control plane resources to a public cloud can be considered by operators pursuing the following goals:

1. Minimize power consumption and space footprint in operator's facilities. In this scenario, only routing/switching equipment remains in the headend/hub.
2. Accelerate and streamline service deployment and nodes splits. vCMTS resources can scale on demand in a matter of minutes/hours in response to business needs.
3. Move to advanced high availability (HA) scenarios with geo-redundancy, with different components of disaggregated vCMTS deployed in different locations. For example, it should be possible to deploy critical system components in geographically distributed availability zones in public cloud and build redundant network paths between hub/headend and CSP infrastructure.
4. Unify vCMTS resource provisioning with existing business processes, such as service provisioning, and leverage cloud-native architecture for achieving CMTS-as-a-service deployment scenarios.

The CMTS scaling factors described for the on-premises edge deployments are also valid for a public cloud deployment scenario. Additional benefits in the latter case include:

1. Improved flexibility for horizontal and vertical node scaling.
2. Reduced time to market when adding more compute nodes to the cluster.
3. Transfer of compute resources life cycle management and associated cost from the operator's domain of responsibility to a CSP.

It should be noted that distance and latency between provider edge and cloud-based facilities should be considered when planning vCMTS deployment scenarios in the public cloud. Another important point is that measures to achieve advanced HA capabilities are not limited to protecting components deployed in public cloud and should also include necessary steps to protect system components in headend/hub and in the field (e.g., CIN redundancy, RPD network connectivity redundancy).

2.3. Hybrid Deployments of Virtualized Cloud-Native Access Platform

In a broad sense, hybrid cloud platforms from almost all the major infrastructure vendors not only manage clusters running on premises and in their own cloud platforms, but any cluster including those that are deployed in other cloud environments[8]. Due to its cloud-native packaging, disaggregated vCMTS provides enough flexibility to deploy it in hybrid scenarios, where data plane pods run on local edge compute nodes, while infra pods are deployed in private or public cloud. It allows operators to fill the gap between on-premises deployment and cloud-based deployment with other options that combine the benefits of both.

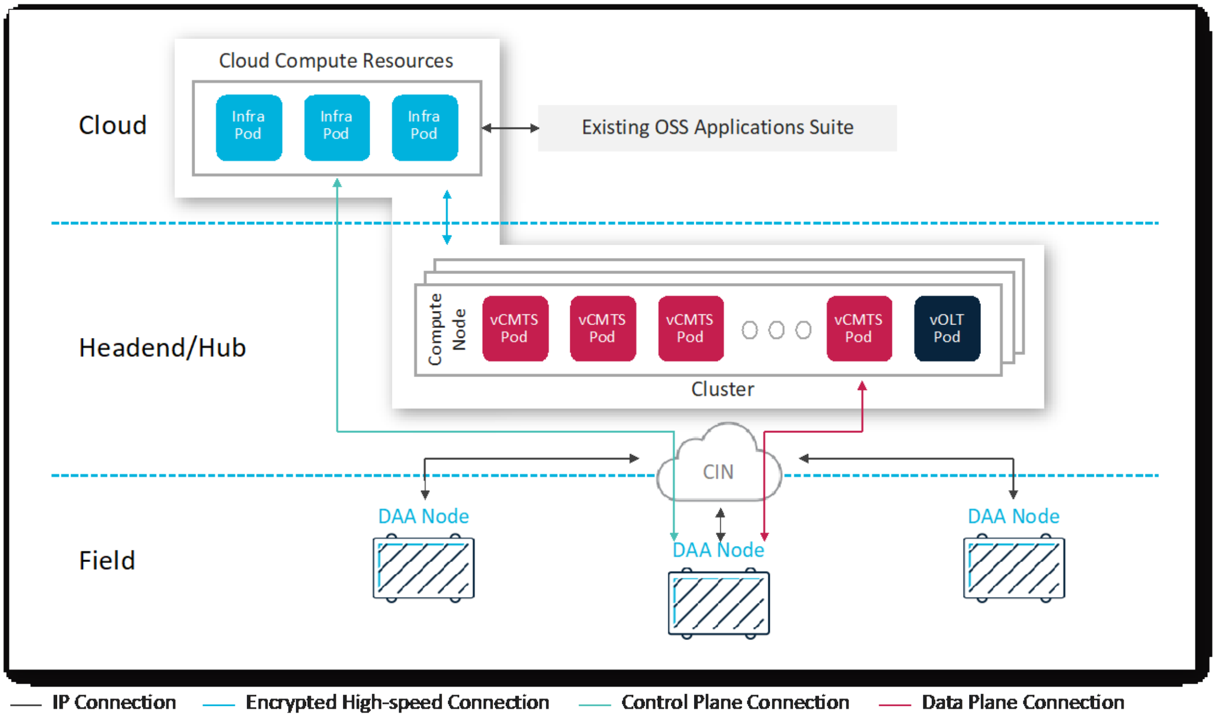

Figure 4 - Example of Hybrid Deployment of Virtualized Cloud-Native Access Platform

Hybrid deployment of the disaggregated virtualized cloud-native access platform is suitable for use cases wherein data plane processing is kept in provider edge facilities, orchestrated by virtualized control plane resources from public or private cloud. The benefits of such deployment type include but are not limited to:

1. Minimized latency between end-user equipment and data plane pods.
2. Possibility to deploy infrastructure and data plane resources in existing k8s clusters, with unified orchestration of vCMTS workloads and third-party workloads.
3. Ability to handle high-availability scenarios, possibly with geo-redundancy provided by a CSP, for control plane components and data plane components in a uniform way. From a pod orchestration standpoint, cloud compute resources and on-premises compute resources are considered to be part of the same cluster.

From a horizontal scaling point of view, the hybrid deployment scenario combines the benefits of on-premises and public cloud deployment scenarios.

There are, of course, other options for deploying disaggregated vCMTS in an operator's edge facilities as well as in a public and private cloud environment. In the following section, we'll look at how the concept of cloud-native disaggregated access platform allows operators to leverage emerging deployment scenarios.

2.4. Cloud-Native Virtualized Access Platform and Deep Edge Compute

So far, we discussed centralized compute resources available to cable operators in a public and private cloud, as well as edge compute resources distributed across headends and hubs. As equipment deployed on the outside plant becomes more intelligent and provides more processing power, we can leverage its capabilities and extend the range of the compute resources available to cable operators to so-called "deep edge" compute. Therefore, deployment concepts we discussed so far can be considered as a superset of CableLabs' Flexible MAC architecture (FMA)[9] that specifies interoperable management, control and data plane interfaces on the digital fiber between the core components and a series of devices, regardless of compute locality.

A virtualized cloud-native access platform allows operators to control the following aspects of FMA deployments:

1. Locality of certain vCMTS components, based on the distance/latency requirements and constraints. Data plane pods can be placed on the compute resources located in the outside plant (so-called "deep edge" deployment), while infrastructure components can be deployed in a centralized fashion.
2. Flexible separation of individual vCMTS components (microservices) and their subsequent placement on the compute resources in the data center, headend/hub or outside plant.
3. Backward compatibility with existing OSS applications suite.
4. Fast time to market and ability to add new services and applications on top of the cluster that consists of general-purpose compute resources.

When deployed as part of a cloud-native virtualized access platform, a group of FMA remote MAC core units (RMC) and/or virtual core (vCore) units in effect becomes a pool of general-purpose compute nodes orchestrated by Kubernetes.

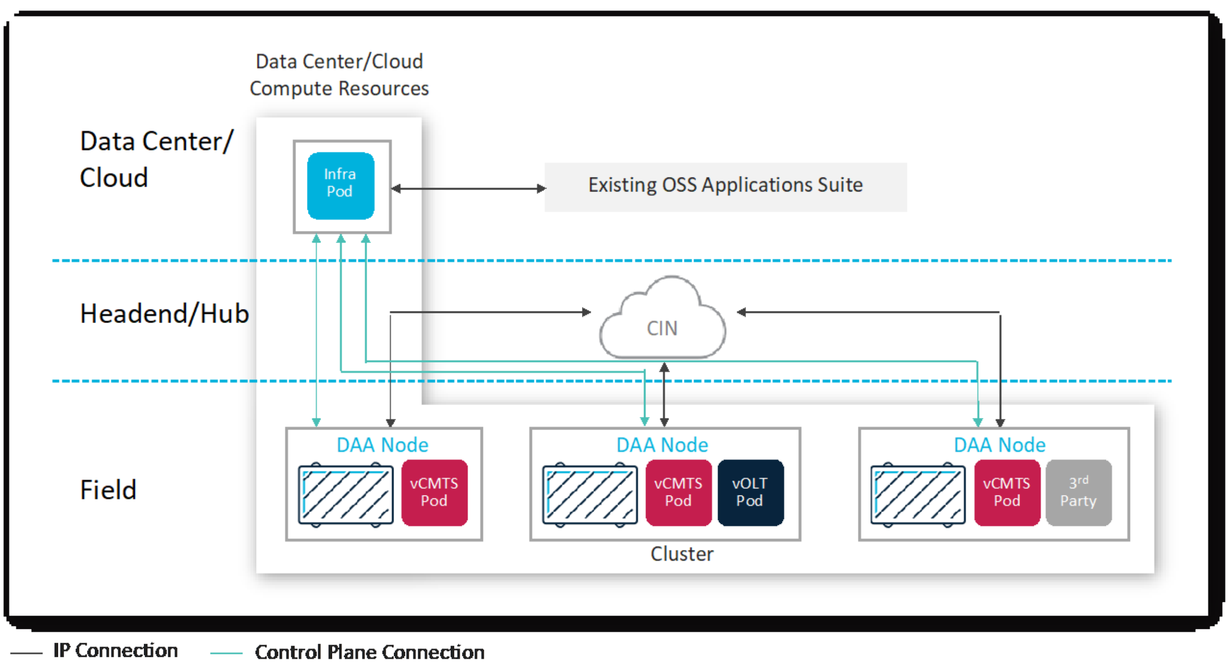

Figure 5 - Disaggregated Virtualized Access Platform and Deep Edge Compute

RMD/RMC/vCore can be considered as a generic compute resource that can be utilized, in particular, for DOCSIS MAC layer processing combined with purpose-built hardware that implements, for example, the DOCSIS PHY layer. In such a case, it is possible to push vCMTS pods to the very network edge for vCMTS software-based packet processing, while leveraging best industry practices for cloud-native software deployment and life-cycle management. It also opens the door for utilizing RMD compute resources for other types of workloads, such as:

1. IoT edge processing.
2. Sharing common generic compute resources in RMD for different types of data plane pods (e.g., perform DOCSIS and wireless data plane processing on the same compute resources in RMD).
3. Hosting applications that can benefit from deep edge deployment: caching servers, speed test applications, VR-specific applications, machine-to-machine communication applications.

Horizontal scaling and infra resource-sharing concepts described earlier are also valid for multitenant clusters. Potential benefits from placing DOCSIS and third-party workloads on deep edge compute nodes include:

1. Reduced end-to-end network latency, which is inherited from the fact that compute resources are located in physical proximity to the customer end devices.
2. Maximized resource utilization that emerges from using generic edge compute resources for DOCSIS and non-DOCSIS applications.
3. Network traffic reduction and savings on bandwidth between deep edge devices and the central cloud.

3. Conclusion

Compute resources available for cable operators span from centralized compute in public and private cloud to the edge compute in headend/hub and deep edge compute in an outside plant. Cloud-native virtualization enables operators to leverage distributed compute resources in a uniform way.

A virtualized cloud-native and disaggregated vCMTS provides a wide range of options for deployment:

1. Traditional on-premises edge deployments in headend/hub.
2. Cloud-based deployments in private cloud and in CSP infrastructure.
3. Hybrid deployments, wherein certain vCMTS components remain in operator’s facilities, while other components get pushed to a CSP infrastructure.
4. Deep edge deployments, where some of the cluster compute resources are located in the outside plant.

In all deployment scenarios, cluster resource and application life cycle management are performed in a uniform way:

1. Compute resources are considered to be general-purpose.
2. Heterogeneous workloads (different types of infrastructure, data plane and third-party pods) are placed on general-purpose compute resources, taking into account application locality preferences (e.g., cloud/data center, network edge or deep network edge).
3. Horizontal scaling is achieved by increasing the number of workloads (data plane pods) and by adding more compute resources (in the cloud or on premises).

The combination of the available compute resources and the flexibility of cloud-native access platform allows operators to deploy different types of access networks in a uniform and operations efficient way. Implementing as many network functions as possible in software and minimizing the number of purpose-built hardware appliances on the network provides a bridge to the future that was never more unknown than now.

Abbreviations

CCAP	converged cable access platform

CIN	converged interconnect network
CLI	command-line interface
CM	cable modem
CMTS	cable modem termination system
CPE	customer premises equipment
CSP	cloud service provider
DAA	distributed access architecture
DEPI	downstream external PHY interface
DOCSIS	Data Over Cable Service Interface Specification
FMA	Flexible MAC Architecture
GCP	generic control plane
HA	high availability
HFC	hybrid fiber-coax
IoT	Internet of things
IPDR	Internet Protocol Detail Record
MAC	media access control
NFV	network function virtualization
OLT	optical line terminal
OSS	operations support systems
PHY	physical layer
PON	passive optical network
RMC	remote MAC core
RPD	remote PHY device
SCTE	Society of Cable Telecommunications Engineers
SNMP	Simple Network Management Protocol
TCO	total cost of ownership
UEPI	upstream external PHY interface
UI	user interface
vCMTS	virtual CMTS
vCore	virtual Core
vPON	virtual PON

Bibliography & References

[1] Matatyaou, Asaf. Transforming the HFC Access Network with a Software-Based CCAP. Publication. San Jose: Harmonic, 2015. Web.

[2] Matatyaou, Asaf. Real-World Deployment of a Virtual Cable Hub. Publication. San Jose: Harmonic, 2017. Web.

[3] Matatyaou, Asaf. Practical Lessons of a DAA Deployment with a Virtualized CMTS. Publication. San Jose: Harmonic, 2017. Web.

[4] Robuck, Mike. Comcast executives on lessons learned from deploying vCMTS. October 29, 2018. Web. https://www.fiercetelecom.com/telecom/comcast-execs-lessons-learned-from-deploying-vcmts

[5] Barr, Jeff. New – Use an AWS Transit Gateway to Simplify Your Network Architecture. November 26, 2018. Web. https://aws.amazon.com/blogs/aws/new-use-an-aws-transit-gateway-to-simplify-your-network-architecture/

[6] Remote PHY Specification, CM-SP-R-PHY-I14-200323, March 3, 2020, Cable Television Laboratories, Inc.

[7] Baseline Privacy Plus Interface Specification, CM-SP-BPI+-C01-081104, March 11, 2008, Cable Television Laboratories, Inc.
[8] MSV, Janakiram. How Kubernetes Has Changed The Face Of Hybrid Cloud. December 16, 2019. Web. https://www.forbes.com/sites/janakirammsv/2019/12/16/how-kubernetes-has-changed-the-face-of-hybrid-cloud
[9] Flexible MAC Architecture (FMA) System Specification, CM-SP-FMA-SYS-D04-200218, February 18, 2020, Cable Television Laboratories, Inc.
[10] Milinkovich, Mike. K8s at the Edge – Some Context on the New Kubernetes IoT Working Group. September 26, 2018. Web. https://eclipse-foundation.blog/2018/09/26/k8s-at-the-edge/
[11] Weins, Kim. Cloud Computing Trends: 2018 State of the Cloud Survey. February 13, 2018. Web. https://www.flexera.com/blog/cloud/2018/02/cloud-computing-trends-2018-state-of-the-cloud-survey/
[12] Computer cluster. https://en.wikipedia.org/, April 15, 2020. Web. https://en.wikipedia.org/wiki/Computer_cluster
[13] What Is NFV Infrastructure (NFVI)? Definition. https://www.sdxcentral.com/, April 27, 2016. Web. https://www.sdxcentral.com/networking/nfv/definitions/nfv-infrastructure-nfvi-definition/
[14] MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I20-200407, April 6, 2020, Cable Television Laboratories, Inc.
[15] CNCF Cloud Native Definition v1.0. Web. https://github.com/cncf/toc/blob/master/DEFINITION.md
[16] Kubernetes. Web. https://github.com/kubernetes/kubernetes

