SCTE·ISBE CABLE-TEC EXPO 2018 | ATLANTA, GA OCTOBER 22-25

2018 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

# Toward Automated Intelligent Resource Optimization for vCMTS Using Machine Learning

A Technical Paper prepared for SCTE•ISBE by

**Kieran Mulqueen**
Research Project Manager
Intel Labs, Intel
Collinstown Industrial Park, Leixlip, Co. Kildare
353 (1) 606 3789
kieran.m.mulqueen@intel.com

**Michael O'Hanlon**
Principal Engineer
Network Products Group, Intel
Dromore House, East Park, Shannon, Co. Clare
353 (61) 777 808
michael.a.ohanlon@intel.com

**Marcin Spoczynski,** Intel Labs, Intel

**Brendan Ryan,** Network Products Group, Intel

**Thijs Metsch,** Intel Labs, Intel

**Leonard Feehan,** Intel Labs, Intel

**Ruth Quinn,** Intel Labs, Intel

# Table of Contents

| Title | Page Number |
|---|---|
| Introduction | 3 |
| Enabling Insight-Driven Management and Orchestration of NFVI Nodes | 3 |
| Utilizing Machine Learning in a vCMTS scenario | 5 |
| Initial Results Demonstrate Key Areas for Optimization | 9 |
| Conclusion | 10 |
| Abbreviations | 11 |
| Bibliography & References | 11 |

## List of Figures

| Title | Page Number |
|---|---|
| Figure 1 - The analytics-driven MANO is supported by long-running observations in a (continuous running) background flow, and a runtime-driven foreground flow in which we deal with fine-grained VNF requests and rebalancing decisions[2] | 4 |
| Figure 2 - Overview of vCMTS scenario configuration in a real-world context | 6 |
| Figure 3 - Testbed setup detailing the network flows between the traffic generation hosts and several instances of the vCMTS data plane | 7 |
| Figure 4 - Sample metrics as collected by the telemetry system | 8 |
| Figure 5 - Showcases variance in energy measurements for several key hardware metrics when two vCMTS VNF instances either share a CPU core (shared) or are pinned to separate cores (exclusive) | 9 |

# Introduction

Virtualization of Cable Modem Termination Systems (CMTS) provides a large range of benefits for operators within the cable industry. Ensuring that this virtualized CTMS architecture can meet required capacity and performance objectives, both current and forthcoming, provides essential assurance to operators in determining whether to adopt this approach. Typically, human-derived policies are being used for the management of the virtualized CMTS (vCMTS), however these are generally static and can result in over- or under-utilized resources and, eventually, in missing service objectives.

The introduction of Network Functions Virtualization (NFV) paves the way toward autonomous management of Virtualized Network Functions (VNF) like vCMTS. Autonomous NFV management using machine learning (ML) shows huge potential to optimize such a system, benefiting both service providers and customers. Machine learning allows for the generation of models that ensure reliability and dependability of the overall system while minimizing running costs and improving service assurance.

To generate analytic models capable of enabling autonomous management capabilities, the NFV Management and Orchestration (MANO) components need to be presented with key behavioral insights of the VNFs and infrastructure resources.

Here we describe an approach that allows derivation of such insights, which facilitates its use in various models that can be used in a Network Functions Virtualization Infrastructure (NFVI). We also show initial proof points demonstrating areas where machine learning can be used for immediate effect.

# Enabling Insight-Driven Management and Orchestration of NFVI Nodes

Software Defined Infrastructures (SDI) as the foundation of NFVIs enable the decoupling of VNFs from the physical infrastructure they run on, which allows for innovations in service delivery: mainly by enabling VNF agility and service assurance.[1] Management and orchestration activities can play a role at various scales. This includes the management of clusters of nodes, as well as the management of individual nodes.

The physical infrastructure is handled as a resource pool, which allows for ease of VNF management. Mapping the VNF instances to the physical infrastructure is the task of the MANO components. By leveraging machine learning-based analytics to, for example, predict resource demands, it is possible to better facilitate various Service Level Agreements (SLAs) of the VNFs while in parallel achieving better resource utilization and reduction of TCO.

To achieve this level of autonomous management of VNFs and infrastructure resources in an NFVI, several components are required:

1. VNF and infrastructure resource aware MANO components, such as the NFV Orchestrator (NFVO); VNF Manager (VNFM) and Virtual Infrastructure Manager (VIM);
2. Background flow analytics enabling observation of VNFs and infrastructure resources over longer time frames;

3. An Information Core which captures insights on application and resource behavior;
4. Foreground flow analytics utilizing the Information Core to deal with VNF requests and rebalancing decisions

Using these concepts, autonomous MANO of VNF instances and the NFVI can be achieved as described in Figure 1.

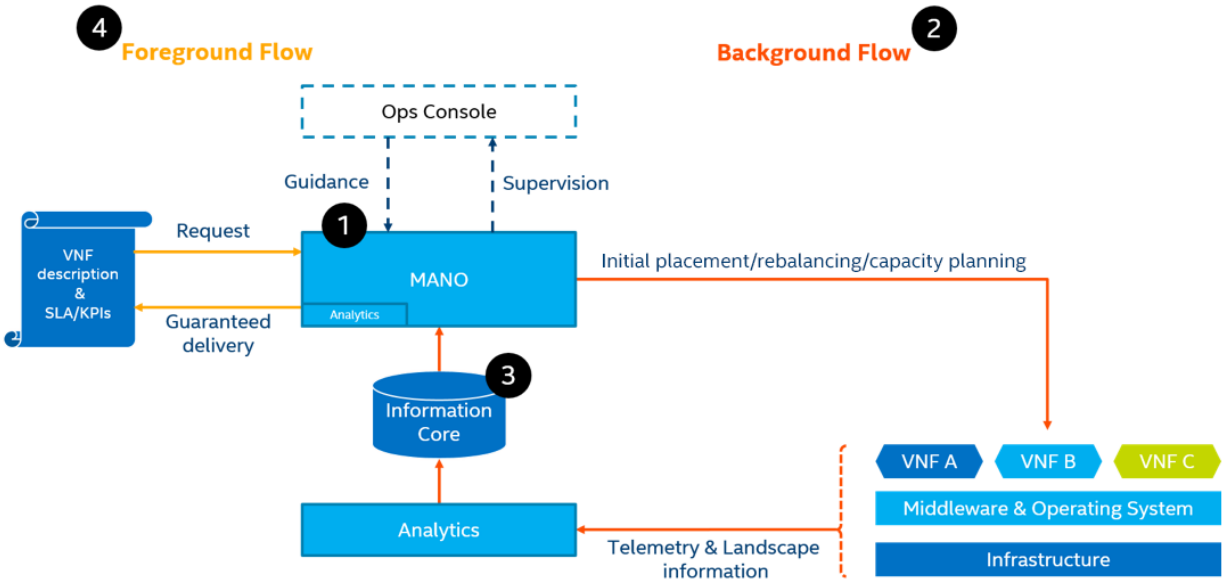


**Figure 1 - The analytics-driven MANO is supported by long-running observations in a (continuous running) background flow, and a runtime-driven foreground flow in which we deal with fine-grained VNF requests and rebalancing decisions[2]**

Within the MANO-enabling components, machine learning derived insights can play several roles for various activities:

a. Determining appropriate **initial placements** of the VNF components allows for tighter packaging. This can be carried out using classification of the VNFs based on available data, allowing for the prediction of future behavior when placed on given resources. This can also predict interactive and interfering behavior resulting from placement with VNF instances sharing resources.
b. Periodic **rebalancing** of the VNF to resource mapping can be triggered only when the benefit derived from migrating, scaling, or tuning the VNF configurations would outweigh the costs of the rebalancing activity.
c. **Capacity planning** allows the forecasting of infrastructure resource capacities.

This system will enable the autonomous management of NFVI: to ensure that an autonomous system can still be supervised and guided it is proposed that it connect to the appropriate operations console.

Machine learning-based analytics that are embedded in the background flow allow for creation of insights in the form of models that capture behavioral patterns of the VNFs and the infrastructure.[3] The models are eventually stored in the Information Core. As this background flow is enabled to run continuously, the models present in the environment can be regularly updated and adapt to changes.

Since various machine learning algorithms exist, the resulting models also have various forms. This includes, but is not limited to, decision trees, regression models, or neural network (NN)-based models. In addition to this, machine learning can facilitate the automated testing of alternate model types, which given the increased amount of data collected over time, might be more suitable than the original model architectures provided to the system. For example, various deployed model outputs can be traded off against each other and, if threshold levels of accuracy are reached, replace those currently in action. Also, continuous exploration of the original model's hyperparameters can be achieved through the mechanism described earlier.

It must be noted here that the background flow can either be enabled on single nodes on the NFVI or be offloaded (if needed, based on available compute capacity). This would mean that models trained elsewhere simply need to be embedded on the nodes for scoring and interference.

When the VNFs are deployed, the VNF manager, as well as the NFV Orchestrator, can use several other machine learning based analytics techniques within the foreground flow. This mainly includes notions of enabling anomaly detection. If the infrastructure itself is predicted to move to an undesirable state an intervention can be made prior to its deterioration. Given the past behavior of classified VNFs and the infrastructure resources at play, the future behavior of VNFs can be predicted and compared to the actual behavior to determine any possible mitigation actions. These mitigations include pausing, killing, or relocation of the VNFs or tuning configurations, such as altering the clock frequency of the CPU or activating available offloading capability. Further, an excessively anomalous result during the foreground analytics points toward suboptimal models being embedded in the MANO stack, and requires further mitigation of the same.

Overall, the integration of machine learned models is key to enabling a responsive, adaptable system. Through the foreground and background flow, enabled analytics models could be created to generate more precise models (e.g., for use in capacity planning activities), rather than ones based entirely on historical data.

# Utilizing Machine Learning in a vCMTS scenario

Introducing the insights derived from machine learning can provide solutions to the numerous activities described earlier, such as optimized initial placement, rebalancing, and capacity planning. While the vCMTS VNFs can be relatively static in terms of infrastructure requests, differences in user activity, channel configurations, subscriber density, and modem encryption types can impact the required infrastructure resources. Further, it is likely that vCMTS VNFs will be executed alongside other VNFs, making appropriate placement a key problem to address. To deal with such a mix of VNFs and their requirements, the available physical infrastructure can be considered a resource pool, providing opportunity for optimization on several fronts, while maintaining adherence to specific performance Key Performance Indicators (KPIs).

Figure 2 gives a high-level overview on how a set of vCMTS VNF instances are deployed on physical infrastructure using Kubernetes as a VIM.

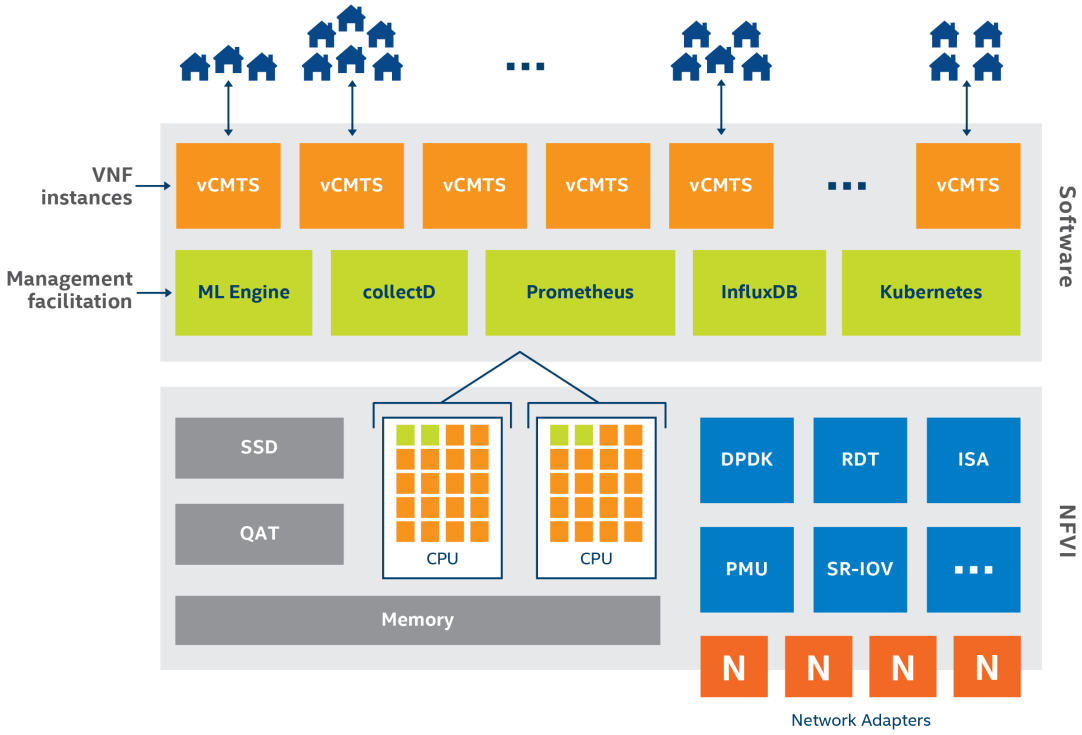


**Figure 2 - Overview of vCMTS scenario configuration in a real-world context**

For preliminary investigation, we have set up the following environment to demonstrate how an ideal initial placement of the VNF instances on single host NFVI can be achieved. Better initial placement can have a huge benefit for the infrastructure resources provider in terms of efficiency and performance and, therefore, profit.

Our baseline VNF is an Intel-developed vCMTS data plane pipeline. This was previously used to investigate performance on an Intel® Xeon® Scalable processor-based platform[4] and contains a DOCSIS MAC* data-plane running on IA heavily utilizing the Data Plane Development Kit (DPDK).[5] Each VNF instance is defined as a service group handling a collection of end users and is pinned to a core of the CPU.

Our testbed is based on Intel's latest architecture in a two-socket configuration, with two Intel® Xeon® Gold 6148 processors. Each processor contains 20 cores with a default frequency of 2.4 GHz. Furthermore, two Intel® QuickAssist Technology (Intel® QAT) adapters are installed, providing hardware acceleration for offloading of crypto- and compression-based tasks. Four Intel® Ethernet Network Adapters are installed and are connected to a separate traffic generation host allowing for high bandwidth testing. Each Network Adapter provides two ports, and hence up to eight physical network connections.

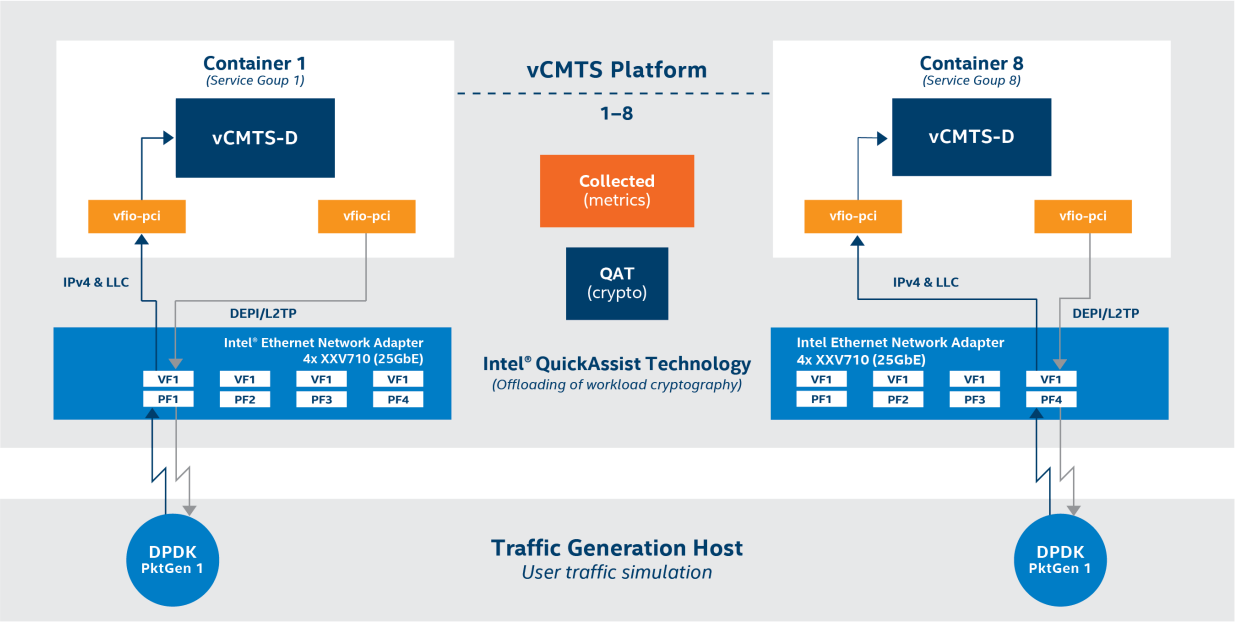


**Figure 3 - Testbed setup detailing the network flows between the traffic generation host and several instances of the vCMTS data plane**

Figure 3 shows the network flows from the traffic generation hosts to several vCMTS data plane instances. This setup leverages platform differentiating features, such as Intel Ethernet Network Adapters technology and Intel QAT.

CollectD* is used for gathering telemetry data from the infrastructure, the middleware, and VNF instances. Intel® Resource Director Technology (Intel® RDT) and Intel® Performance Counter Monitor (Intel® PCM) allow for detailed instrumentation and management of the platform. These technologies allow gathering of detailed hardware-level telemetry data, such as statistics about the usage of the individual cores and caches within the processors. The open source metric monitoring tool Prometheus* is used to consume the output of CollectD, aggregate the data, and perform derivation of our custom set of additional metrics.

Telemetry data is a key input for understanding the behavior of the VNF instances, as well as the infrastructure resources.[6] Metrics have various levels of granularity (for example, per socket, per core, or per thread). As an example, high levels of utilization of the CPU, the associated caches, and the number of Instructions per Cycle (IPC) can indicate when computation itself has become a limiting factor.[7,8] Furthermore, metrics on the usage of memory and I/O provide insights into the VNF's impact on other subsystems of the platform. Next to this, a landscape makes the relationship between entities in the system explicit. For example, the metadata describing which VNF instances were associated with which cores in the system can be used by the data pipeline in the analytics component.

Preselection of key useful metrics allows for more lightweight machine learned models to run efficiently, facilitating the foreground analytics flow. Detailed metrics—or statistical information on the same—are stored to the Information Core for later processing and future model generation. Derived metrics can also be generated, which provides a set of more complex data that captures less obvious component interactions. For example, the interplay between and impact on performance of the L1 and L2 caches can be rolled up into a single, more useful, metric.

 

Figure 4 shows a subset of metrics as gathered by the telemetry system, which can potentially be used to train machine learned models. The initial focus for this paper is on metrics such as the energy usage of the CPU (package energy) and memory (RAM energy). The IPC metric indicates how many instructions per cycle the CPU is performing and is a good indicator of overall system performance. The closer the IPC metric is to the theoretical maximum, the more efficient the system is performing (similar to miles per gallon). A drastic reduction of the IPC metric would indicate that the CPU is most likely waiting on memory I/O and cannot process faster.

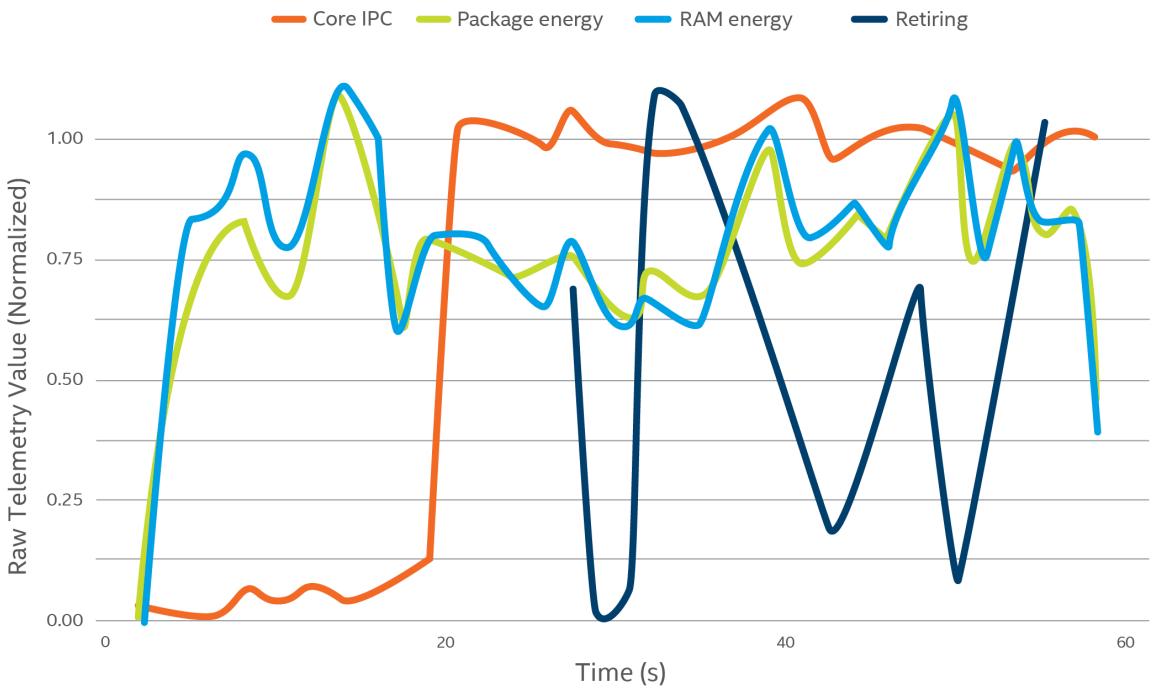


**Figure 4 - Sample metrics as collected by the telemetry system**
Energy usage of the sockets, as well as memory, are shown in relation to metrics about the usage of the CPU (Instructions Per Cycle and Retiring)

Even for a single compute host NFVI about 200 different metrics can be collected from various subsystems. A small time window is shown in the diagram with a few data points; in a production system, large volumes of data points can be expected. It is important to be able to select those metrics which are key for the scenario at hand. This can be partly automated by a feature selection process that determines those metrics which are significant enough or show a high enough probability to be useful to predict a certain outcome.

# Initial Results Demonstrate Key Areas for Optimization

To investigate the potential benefits of using machine learning to optimize initial placement of VNFs on a single host NFVI, experiments were run that compared the behavior of two vCMTS instances. In one case, the instances were pinned to two separate cores and can exclusively make use of them, while in the other case, two instances share a single core. The latter case is a simple example of dense packing and ultimately of a better return on investment as the processor cores are more optimally utilized. Note that although we seek to achieve dense packing of VNF instances, it must not come at the cost of a reduction in the performance of the vCMTS instances and hence an impact on the end user. Being able to apply machine learning optimization by determining the most beneficial number of VNF instances sharing a resource is hence the driving goal.

As a baseline measurement of the two scenarios, we investigated performance by configuring the system to have 500 users subscribed to each VNF instance, exclusively using AES encryption and one OFDM channel. The total throughput for each VNF instance was 5 Gbps.

Behavioral differences were seen when comparing two different usage scenarios. As Figure 5 shows, the average package energy measured by the system was ~161 Joules/s when two VNF instances were pinned to individual cores (exclusive), while when pinned to a single core (shared) the overall power usage was reduced by 32 percent. The IPC metric on the other hand shows less of a reduction, which demonstrates that the system is not yet saturated. This means that although the vCMTS VNF instances are more tightly packaged, the system is not under-provisioned. This is also confirmed through the experiments (regardless of whether the VNF instances are sharing CPU cores or not) which show a similar throughput.

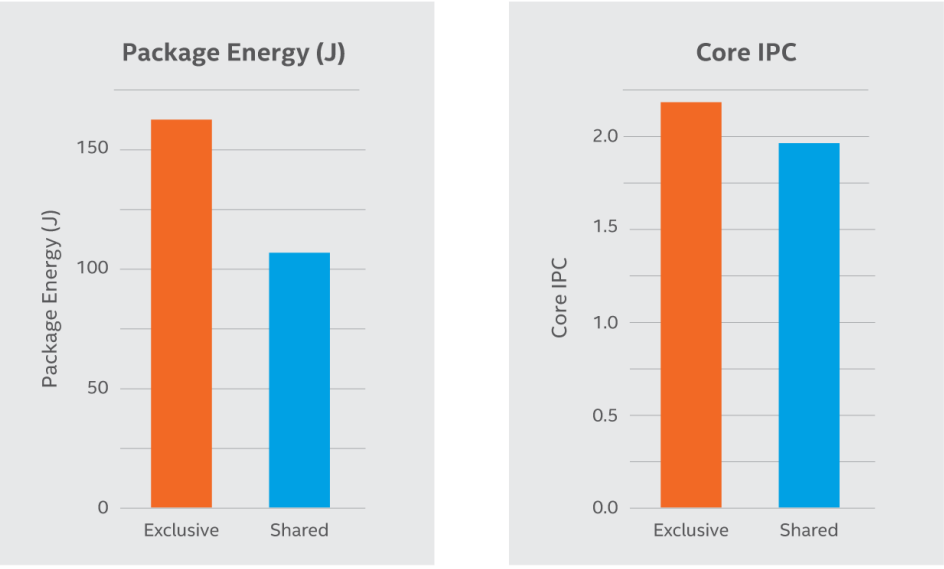


**Figure 5 - Showcases variance in energy measurements for two key metrics when two vCMTS VNF instances either share a CPU core (shared) or are pinned to separate cores (exclusive)**

# Conclusion

Tighter packaging of VNF instances on shared resources of an NFVI is beneficial to make the best use of the available capacity. However, it is crucial to understand the behavior of VNF instances in such scenarios to ensure that their service level performance is not impacted. Such insights on what the optimal configuration is can be derived using machine learning insights. These insights can then be embedded in the MANO decision-making process while allowing for an autonomous adaptive system. This is especially necessary if the overall system needs to adapt to context changes, such as changes in user activity, channel configurations, and subscriber density.

Overall, the benefits of using machine learning approaches for derivation of insights that can be used in a MANO stack are the following:

1. An adaptive system that can automatically adapt to the addition of new VNF types or customer behaviors, reducing the need to manually change set placement policies.
2. Higher quality of service, due to multiple layers of anomaly detection and fault checks in service behavior data.
3. Reduction of OpEx through tighter packaging of VNFs, ensuring power savings when VNFs share infrastructure resources.
4. Coordination of multiple VNF types, enabling a vCMTS to not only execute on the same server infrastructure as other VNFs, such as augmented reality (AR), IoT, etc., but to leverage the differences between these VNFs to improve collective execution and save on OpEx across multiple fields.

vCMTS combined with insight-driven MANO allows for improved utilization of the resources, while offering consistent levels of Quality of Service (QoS). As a result, cable service providers can achieve new levels of optimization of the NFVI in order to deliver better service to their customers, while reducing TCO.

# Abbreviations

| | |
|---|---|
| CMTS | Cable Modem Termination System |
| CPU | Central Processing Unit |
| IPC | Instructions per Cycle |
| I/O | Input/Output |
| KPI | Key Performance Indicator |
| MANO | Management and Orchestration |
| ML | Machine Learning |
| NFV | Network Functions Virtualization Infrastructure |
| NFVI | Network Functions Virtualization Infrastructure |
| NFVO | NFV Orchestrator |
| SDI | Software-Defined Infrastructure |
| SLA | Service Level Agreements |
| TCO | Total Cost of Ownership |
| vCMTS | Virtualized Cable Modem Termination System |
| VIM | Virtualized Infrastructure Manager |
| VNF | Virtualized Network Function |
| VNFM | VNF Manager |

# Bibliography & References

1. S. Krishnapura, et al., *How Software-Defined Infrastructure Is Evolving at Intel*, Intel Corporation, 2015, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/how-software-defined-infrastructure-is-evolving-at-intel-paper.pdf
2. T. Metsch et al., *Apex Lake: A Framework for Enabling Smart Orchestration*, Middleware Industry '15, Proceedings of the Industrial Track of the 16th International Middleware Conference, https://dl.acm.org/citation.cfm?id=2830016
3. R. Khanna R. et al., *Autonomic Characterization of Workloads Using Workload Fingerprinting*, https://ieeexplore.ieee.org/document/7015482/.
4. Maximizing the Performance of DOCSIS 3.0/3.1 Processing on Intel® Xeon® Processors https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/vcmts-docsis-architecture-study.pdf
5. H. Wippel, *DPDK-based implementation of application-tailored networks on end user nodes*, 2014 International Conference and Workshop on the Network of the Future (NOF), Paris, 2014, pp. 1-5., https://ieeexplore.ieee.org/document/7119762/
6. A. Yasin, *A Top-Down Method for Performance Analysis and Counters Architecture*, 2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), https://ieeexplore.ieee.org/document/6844459/
7. A. Herdrich et al., *Cache QoS: From Concept to Reality in the Intel® Xeon® Processor E5-2600 v3 Product Family*, https://ieeexplore.ieee.org/document/7446102/
8. M. Schwarzkopf, *Operating System Support for Warehouse-Scale Computing*, PhD thesis, https://people.csail.mit.edu/malte/pub/dissertations/phd-final.pdf

 

Estimated results reported above may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information about benchmarks and performance test results, go to www.intel.com/benchmarks.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.