White Paper

Energy Efficient vCMTS Deployments Using 4th Gen Intel® Xeon® Scalable Processor Power Management Features

Realize up to 26% CPU power savings and 18% platform power savings over 24 hours in a vCMTS deployment with the latest Intel® Xeon® Scalable processor power management features.

Authors

Rory Sexton
Senior Software Engineer
Intel

David Coyle
Senior Software Engineer
Intel

Subhiksha Ravisundar
Network Software Engineer
Intel

Introduction

The increased popularity of virtualized cable modem termination system (vCMTS) deployments as the Data Over Cable Service Interface Specification (DOCSIS) MAC solution of choice in the cable industry has been heavily influenced by numerous factors. The flexible nature of servers using Intel® Xeon® Scalable processors allows for many opportunities to optimize deployments in terms of performance, utilization, management, resiliency, and workload convergence.

While outright performance is and remains crucial [1] in all decisions, power consumption and performance per watt are quickly becoming more critical when it comes to key performance indicators (KPIs). The global trend across society and industry to reduce the carbon footprint has accelerated the need for reduced energy consumption, extending to the cable access network elements, including vCMTS. Power consumption at the headend is lowered significantly by moving from a legacy big-iron CMTS solution to a vCMTS deployment [2]. However, when the correct techniques are used, further power optimizations are achievable on a vCMTS.

vCMTS implementations are generally based on the Data Plane Development Kit (DPDK) [3] to ensure the data plane elements providing the DOCSIS MAC functionality are optimized for performance. While using highly efficient DPDK libraries benefits throughput, it creates a challenge regarding efficient power utilization.

Hardware features present on servers with Intel Xeon Scalable processors, including P-states and C-states [4], are traditionally controlled either by the operating system (OS) or autonomously by the hardware itself. Power consumption is reduced by entering power-efficient states when the OS or hardware detects that a CPU is not used to its full capacity. However, due to the 'always on' nature of the poll mode drivers (PMDs) used by DPDK, the OS and hardware cannot distinguish between the varying levels of CPU utilization generated by the network load of the vCMTS DOCSIS MAC application. As a result, to take advantage of the underlying hardware power management features on servers using Intel Xeon Scalable processors, further techniques are required to accurately compute the true CPU load and determine when power-saving states can be used.

The varying nature of data usage on cable access networks over a daily and weekly basis lends itself to many opportunities where vCMTS servers are under-utilized. Each moment where CPU cores are processing less traffic than they are capable of handling is an opportunity to use the underlying hardware power management features to increase the energy efficiency of the overall solution. For example, view a typical 24-hour network traffic profile, as shown in Figure 1.

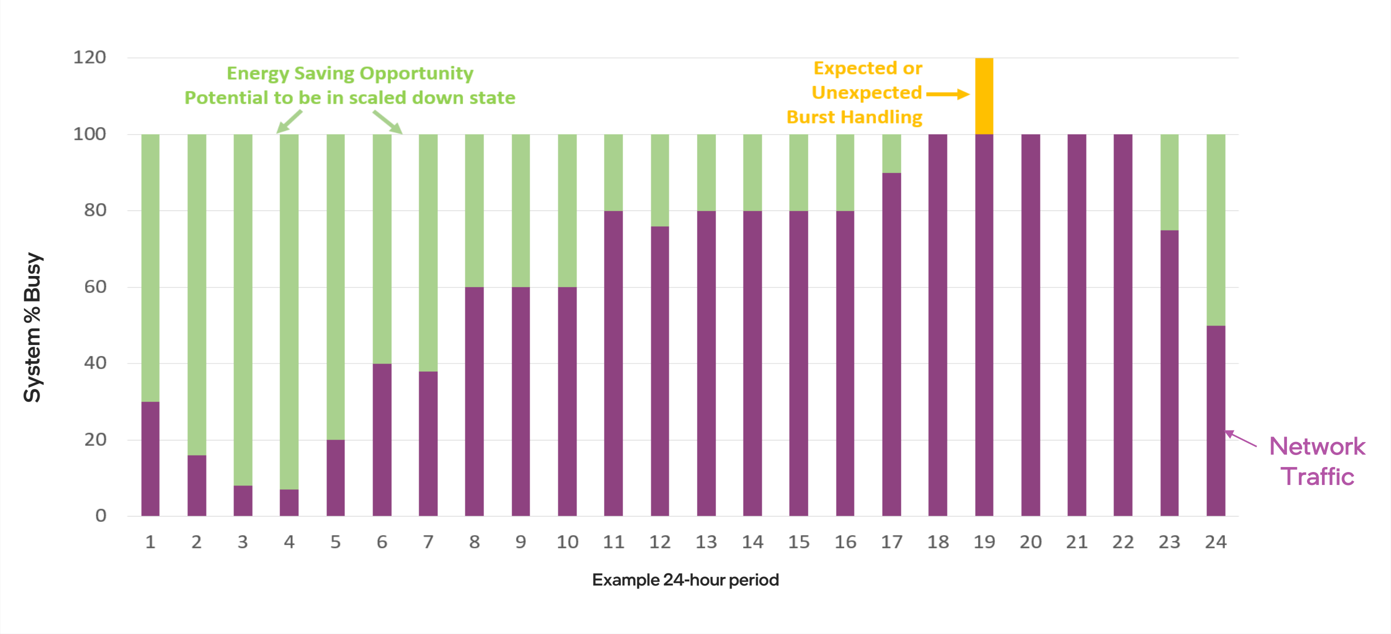


Figure 1. Example 24-hour traffic profile.

Only during the peak evening hours is there enough network load to fully stress the system. For most of the day, particularly in the early morning hours, load is significantly lower, leaving huge potential to save power during these periods. Furthermore, platforms are typically provisioned in a way that leaves additional overhead for supporting unexpected surges in peak network demand. This additional overhead provides further chances to reduce power consumption during times when the server is operating within its expected load.

This paper will introduce the power management features available on 4th Gen Intel Xeon Scalable processors before detailing how they can be applied to vCMTS workloads to ensure that power consumption matches the network load. It will discuss how CPU and platform power reductions of up to 33% and 25%, respectively, are achievable during non-peak periods using a combination of C-states and P-states, averaging out to CPU and platform power savings of 26% and 18% over a typical 24-hour period[1]. Such examples of significant power reductions are key in the path towards a more energy-efficient cable access network, with improved performance per watt and reduced operational expenditure (OpEx).

Table of Contents

Figures

Tables

Power Management Features on Intel® Xeon® Scalable Processors

Intel Xeon Scalable processors provide two primary power management features: C-states and P-states. As previously mentioned, control of these features has traditionally been left to the OS and hardware. However, recent advancements by Intel mean these can now be controlled directly by user-space applications.

Power-Optimized C-states on 4th Gen Intel® Xeon® Scalable Processors

Technology Overview

C-states are a hardware power management feature that enable power savings by turning off specific functions of a CPU core. Only upon exit of the power saving state does the execution of instructions recommence. There is a latency associated with exiting a C-state, as various functions of the CPU core must be re-activated. This exit latency, coupled with the OS and hardware exclusively controlling their usage, limited the suitability of C-states for power management on the 1st-3rd generations of Intel Xeon Scalable processors to cores that did not process data plane traffic (such as cores used for control plane or high availability).

4th Gen Intel Xeon Scalable processors introduce new power-optimized C-states, which are considered substates of C0 and known as C0.1 and C0.2. Like the deeper C-states (C1, C1E, C6), instruction execution stops once a CPU core enters one of these new C-states. However, the new C-states have near-negligible exit latencies and are much better suited to data plane workloads.

These new C-states can effectively reduce power consumption for DPDK-based "100% polling" data plane workloads, such as a vCMTS DOCSIS MAC data plane. Key to their suitability is the new User Wait (or WAITPKG) instruction set [5]. This instruction set can be used to place CPU cores into the new C-states, for example, when network load is below peak rates. Because they do not require root privilege, the instructions can be called directly from a user-space application.

The recommended instruction from this set is TPAUSE, which instructs the core to stop instruction execution for a defined period. During this period, the core can enter the C0.1 or C0.2 states or switch to a hyper-thread sibling. The TPAUSE instruction is recommended as it can be used in scenarios that necessitate multiple receive interfaces to be polled or where critical time-based processing is required. Both above scenarios may be valid in a typical vCMTS application:

- A vCMTS thread may need to poll both a NIC port/queue interface for IP packets and a ring interface for control plane or MAC management messages.
- A vCMTS thread may need to poll a NIC port/queue interface while also handling critical time-based operations, such as DOCSIS Upstream Bandwidth Allocation Map messages, at a defined fixed interval.

Enablement in DPDK

The Ethernet PMD Power Management [6] feature was added to DPDK v21.02 in the form of new APIs, allowing applications to enable and disable power savings on a per NIC port/queue basis. The API enables/disables an algorithm that tracks the rate at which packets are received when the NIC interface is polled and instructs an associated CPU core, through the User Wait instruction set, to enter the C0.1 or C0.2 states if they are available.

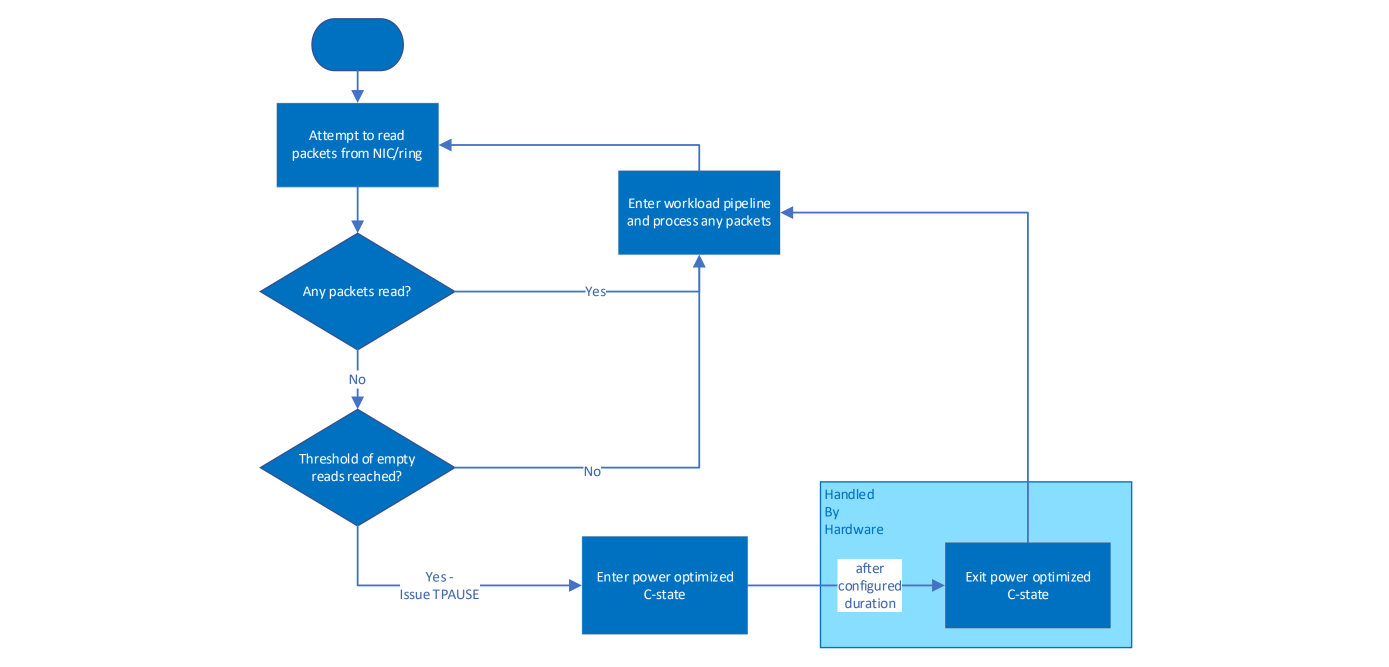

Attempt to read packets from NIC/ring
Enter workload pipeline and process any packets
Any packets read?
Yes
No
Threshold of empty reads reached?
No
Yes - Issue TPAUSE
Enter power optimized C-state
Handled By Hardware
after configured duration
Exit power optimized C-state

Figure 2. User Wait 'Pause' algorithm implemented by DPDK.

This API introduced several modes of operation, but the recommended mode for a vCMTS data plane is the 'Pause' mode. When this mode is configured at application initialization time, the algorithm shown in Figure 2 is enabled within the DPDK 'ethdev' and 'power' libraries and within the DPDK PMD responsible for polling the NIC queue. The algorithm tracks the number of packets received on every queue read. Once a configured threshold of empty reads is reached, the PMD issues the TPAUSE instruction to place the core into the power-optimized C-state for a configured number of microseconds. The core is re-activated (i.e., it exits the C-state) after the configured duration has elapsed.

Not all application threads, however, are based on receiving packets from an Ethernet NIC interface. It is common to have threads that read packets from a DPDK ring interface. Such threads can also benefit from the User Wait power management techniques provided by the PMD Power Management feature.

A patch is available on the DPDK patchwork site [7], which adds support for the 'Pause' mode of the PMD Power Management feature, and, therefore, for the TPAUSE instruction, to DPDK rings. It follows the same algorithm shown in Figure 2.

Because this algorithm is fully contained within the DPDK libraries and PMDs, an application must only enable and configure the feature during application initialization. The underlying algorithm remains completely transparent to the application once enabled. A detailed deployment guide is available [8], describing how to enable and configure the PMD Power Management feature in a vCMTS application.

P-states and Intel® Infrastructure Power Manager

Technology Overview

P-states refer to the specific frequency and voltage a core operates at while executing instructions. They offer power savings by reducing the voltage and frequency of CPU cores while they run. Unlike with C-states, the execution of instructions continues as P-states are altered on the CPU core, albeit at an adjusted rate. This has made them an effective mechanism for reducing power consumption across all generations of Intel Xeon Scalable processors. P-states benefit vCMTS deployments as reducing the operating voltage and frequency of the CPU cores that vCMTS data plane threads are running on results in a corresponding reduction in power consumed by the CPU. Their suitability to vCMTS deployments is further strengthened by the significant reductions in P-state transition latency seen on both 3rd and 4th Gen Intel Xeon Scalable processors [9].

Control via Intel Infrastructure Power Manager

As discussed previously, the main challenge is identifying opportunities where network load is lower than the maximum load for which the platform has been provisioned, and the P-state of cores can subsequently be lowered to optimize power consumption. It is also necessary to identify sudden increases in network load so that the P-state can be increased in a sufficiently quick manner to ensure that no packets are dropped. Traditional P-state controls within both the OS and hardware itself cannot distinguish between low and high utilization of CPU cores which are processing packets using DPDK PMDs, which always show 100% CPU usage.

A basic "time-of-day" approach to solving the problem relies on configuring pre-determined core frequencies capable of handling the expected network load over a typical 24-hour period. While such a technique can provide power savings, it requires a solution that reacts quickly to a change in the load conditions and adjusts frequency appropriately. Any increases in network load atypical of a normal 24-hour period either requires implementation of an automation entity to reconfigure the core frequencies or may result in a performance degradation, including potential packet loss. The lack of real-time metrics used in this P-state control technique gives it clear limitations.

Intel has recently launched the Intel® Infrastructure Power Manager (referred to as IPM in this paper) [10] to overcome the above challenges. IPM determines a "true" busyness metric for CPU cores and uses this to dynamically match CPU performance and power consumption with real-time network traffic levels, thereby providing a significant power saving automation opportunity versus the basic "time-of-day" approach. IPM does all this while maintaining existing throughput, latency, and packet drop KPIs under real network conditions (i.e., handling packet bursts). While this paper focuses primarily on power management on 4th Gen Intel Xeon Scalable processors, it must be noted that IPM is also supported on the previous 3rd Gen Intel Xeon processors.

Once installed, IPM connects to the DPDK telemetry socket of the applications (e.g., vCMTS) under its control and closely monitors their reported busyness telemetry. IPM can automate several power management technologies in the CPU hardware to maximize power efficiencies. However, this paper used IPM solely for run-time per-core P-state control. With millisecond-level granularity, it continually re-computes the target P-states of CPU cores, setting higher or lower core frequencies as needed, with a configurable bias toward slow reductions and fast increases. The only requirement is that a patch be applied to DPDK [11] to expose the required busyness telemetry and that the vCMTS application be compiled against the patched DPDK. There are no other updates required to the vCMTS application itself.

The self-contained, workload-agnostic, and platform-feature-agnostic nature of IPM make it a very attractive option for multiple system operators (MSOs) and independent software vendors (ISVs) who do not wish to modify their vCMTS software at all but still want to realize significant power savings. The new power-optimized C-states described previously, on the other hand, moves the power management into the vCMTS application itself, requires some vCMTS software changes to enable/configure their usage and requires the CPU to provide these new C-states and the User-Wait TPAUSE instruction.

For more information about IPM and details on installation and use, please contact your Intel representative. For more information about P-states in general and some other tools that can be used to control them, please see the power management deployment guide for a vCMTS [8].

Intel® vCMTS Reference Dataplane

The Intel® vCMTS Reference Dataplane [12] is a reference implementation of a vCMTS DOCSIS MAC data plane compliant with DOCSIS 3.1 specifications and optimized for high-performance packet processing on Intel Xeon Scalable processors. The reference implementation accurately reflects a typical vCMTS DOCSIS MAC data plane, allowing for accurate characterization of the expected performance, power consumption, and total cost of ownership (TCO) analysis on Intel Xeon Scalable processor-based servers. The implemented upstream and downstream pipelines, as shown in Figure 3, are described in detail in Appendix A.

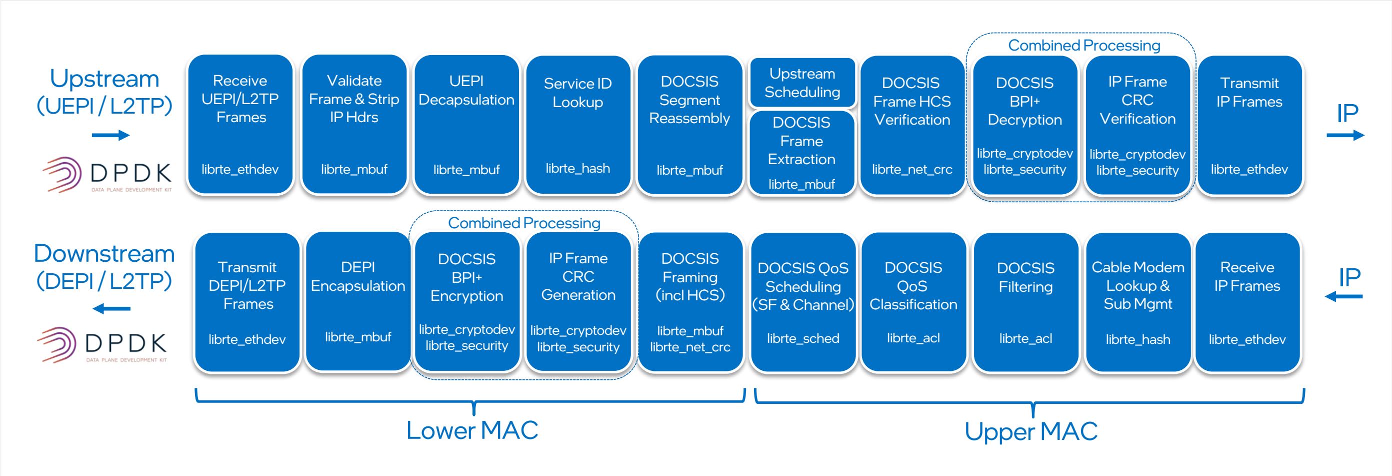


Figure 3. Intel® vCMTS Reference Dataplane.

Each instantiation of the Intel vCMTS Reference Dataplane application requires two CPU cores, as illustrated in Figure 4. One core is used for downstream processing, with the upper and lower MAC functionality split across the two hyper-thread siblings[2]. The other core is used for upstream processing and management, again split across the two hyper-thread siblings of the core.

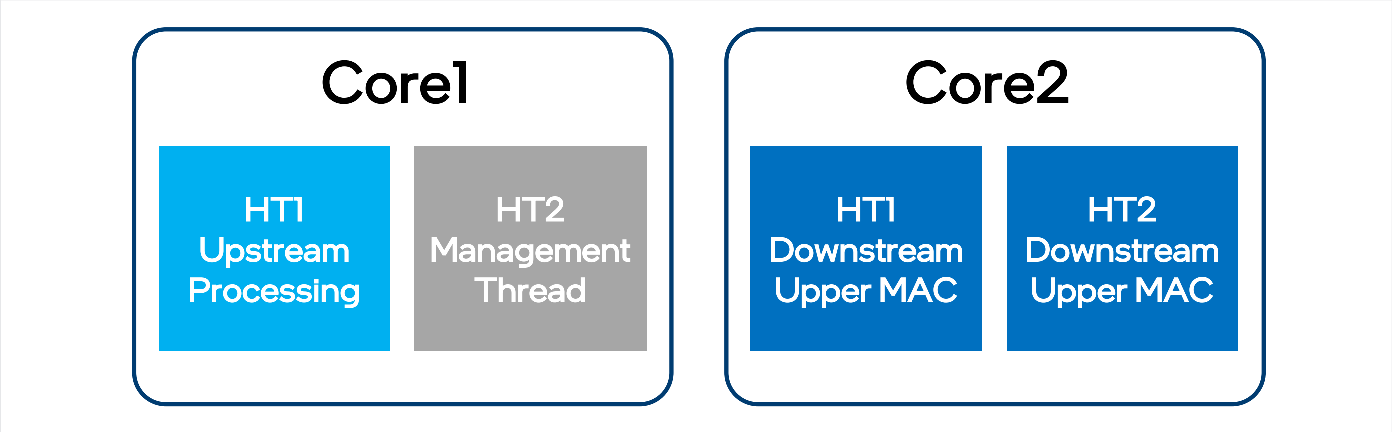


Figure 4. vCMTS thread placement on CPU cores.

As with most performant vCMTS data plane implementations, the application is built on top of DPDK, with its highly optimized libraries and PMDs. As a result, utilization of the CPU cores used by the Intel vCMTS Reference Dataplane always appears as 100%, regardless of the level of actual DOCSIS traffic being processed. This prevents traditional hardware and OS-based power management from achieving energy efficiencies. As a result, support for previously described power management features, namely power-optimized C-states via User Wait TPAUSE instruction and P-state control via IPM, have been added to the application. The results described in the remainder of this paper are based on benchmarks run with the Intel vCMTS Reference Dataplane with one or both power management features enabled on the data plane cores.

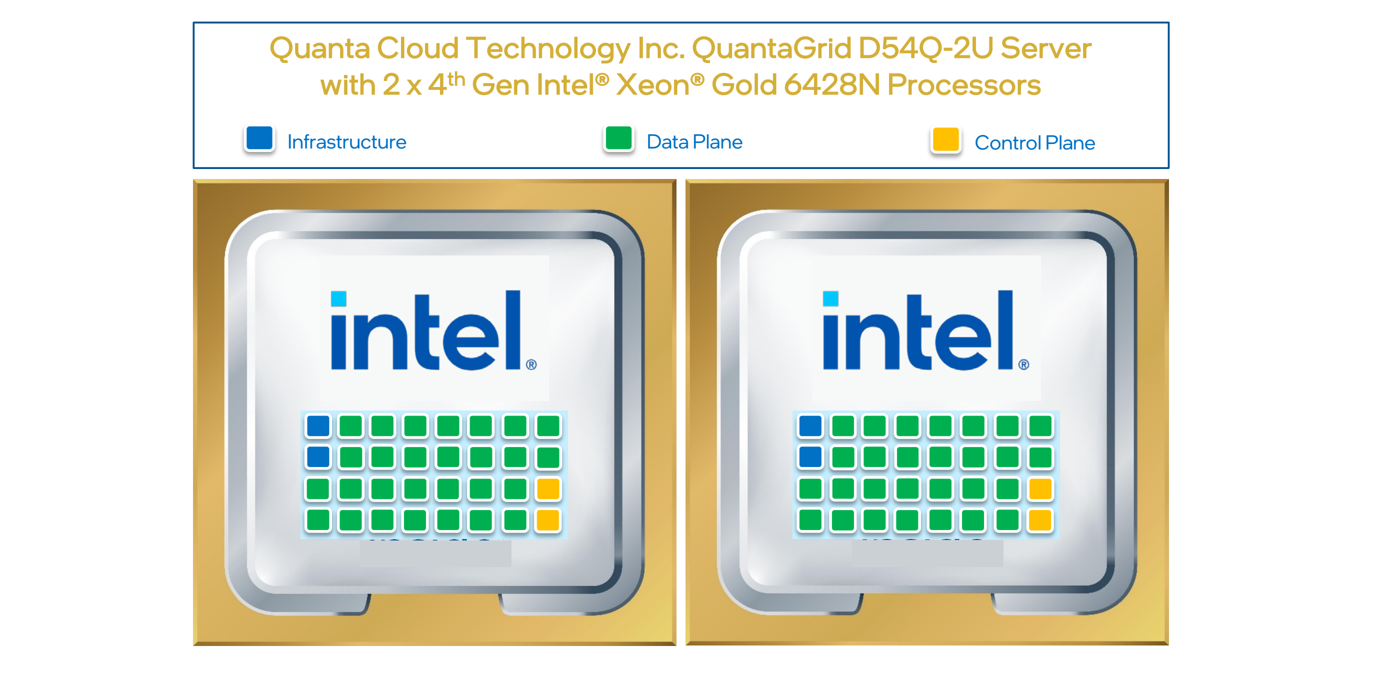

Quanta Cloud Technology Inc. QuantaGrid D54Q-2U Server
with 2 x 4th Gen Intel® Xeon® Gold 6428N Processors
Infrastructure
Data Plane
Control Plane
intel.
intel.

Figure 5. Intel® vCMTS Reference Dataplane CPU core usage.

Testing

Methodology

To understand the scope of power savings that can be achieved by using the new power-optimized C-states and/or by controlling P-states with IPM in a typical vCMTS deployment, several tests were run on a Quanta Cloud Technology Inc. QuantaGrid D54Q-2U server containing two 4th Gen Intel Xeon Scalable processors and using the Intel vCMTS Reference Dataplane[1]. The server was equipped with two Intel Xeon Gold 6428N (4th Gen) CPUs. Each CPU has a thermal design power (TDP) of 185W and has 32 physical cores; with hyper-threading enabled, this gives 64 logical cores per CPU. For all tests, 28 instances of the Intel vCMTS Reference Dataplane were deployed on the server, 14 instances per CPU. In this configuration, the 56 physical cores running the data plane pipeline, as shown in Figure 5, can process a maximum of 287 Gbps of bi-directional DOCSIS network traffic. However, because a server in a live deployment would typically be provisioned to allow additional headroom to handle unexpected surges in network traffic, the test server is only loaded to a maximum of 90% of this capacity (258 Gbps) for all test cases. The remaining cores of the CPUs are reserved for infrastructure and control plane components.

Three separate test cases were executed to gain a full picture of what various power management features can offer in terms of power savings:

1. One test to measure the maximum power savings possible in a vCMTS deployment and to understand the latency implications, if any, of using the power management features.
2. A second test to ensure that the monitoring algorithms can react to rapid increases in network traffic and that the power saving state can be exited quickly enough not to cause packet loss in the network.
3. The third and final test used a typical 24-hour cable traffic profile to determine the expected power savings of the various features over a typical 24-hour period.

For the first test, the network load applied to each vCMTS instance was gradually increased from idle to 258 Gbps (i.e., 90% of the maximum possible network load). At each increment, the aggregate DOCSIS throughput, CPU power draw, platform power draw, vCMTS pipeline latency, and C-state and P-state residency (as applicable) were recorded.

For the second test, the network load to the CPU was rapidly changed from idle to 258 Gbps of aggregate DOCSIS throughput and packet loss statistics were recorded.

For the final test, a typical 24-hour cable network traffic profile was replayed over a condensed time, with a peak traffic rate of 258 Gbps. The aggregate DOCSIS throughput, CPU power draw, platform power draw, vCMTS pipeline latency, and C-state and P-state residency (as applicable) were recorded at regular intervals during the test.

For each test above, four different configurations were benchmarked:

- A baseline with no power management features enabled on any of the vCMTS data plane cores.
- New power-optimized C-states enabled on vCMTS data plane cores, triggered by the User Wait TPAUSE instruction.
- P-states of vCMTS data plane cores controlled by IPM.
- New power-optimized C-states enabled on vCMTS data plane cores, triggered by the User Wait TPAUSE instruction, and P-states of the cores controlled by IPM.

Results

Maximum Possible Power Savings

To visualize the maximum possible power savings, the total CPU power draw from the two CPUs of the server is graphed in Figure 6. The X-axis shows the DOCSIS throughput handled by the server increasing from 0 to 258 Gbps, while the Y-axis shows the total CPU power draw of the two CPUs. Data series for the baseline, C-states (enabled via User Wait TPAUSE), P-states (manipulated via IPM), and combined C-states + P-states are plotted, showing the CPU power consumption at each rate of DOCSIS throughput.

The maximum power savings are achieved when the DOCSIS throughput is at its lowest, shown by the leftmost point in Figure 6, where the DOCSIS throughput processed by the server is below 10 Gbps. At this point:

- C-states triggered by the User Wait TPAUSE instruction result in a total CPU power reduction of up to 62W, or 24%.
- IPM managing the P-states of the vCMTS CPU cores gives a total CPU power reduction of up to 71W, or 27.5%.
- Combining both techniques above gives a total CPU power reduction of up to 90W, or 34.6%.

As DOCSIS throughput increases, the power savings are reduced, as there are fewer opportunities for the C-states to be entered or the P-states to be lowered. At the rightmost point in Figure 6, power savings are still being achieved, albeit much less. This power saving at 258 Gbps of DOCSIS throughput is primarily due to the asymmetric nature of cable access network traffic, where upstream rates typically top out at 10%-20% of the downstream rate. This lower rate of upstream throughput allows some headroom on the upstream CPU cores to enter C-states or lower P-states, even at these higher overall network loads.

It is worth noting that the C-states can only provide additional savings in the combined scenario (orange series) at the lowest traffic rates. At these rates, IPM has already set the core frequencies to the lowest possible frequency, and it is only at this point that the User Wait TPAUSE algorithm can provide some extra power savings. At higher rates, IPM has set the core frequencies to values that can handle those particular throughput levels. However, at these frequencies, there are fewer opportunities for the cores to enter the C-states via the User Wait TPAUSE algorithm.

The power savings achievable with the three power management scenarios are further visualized in Figure 7, with the Y-axis showing the percentage by which the CPU power draw is reduced at each rate of DOCSIS throughput.

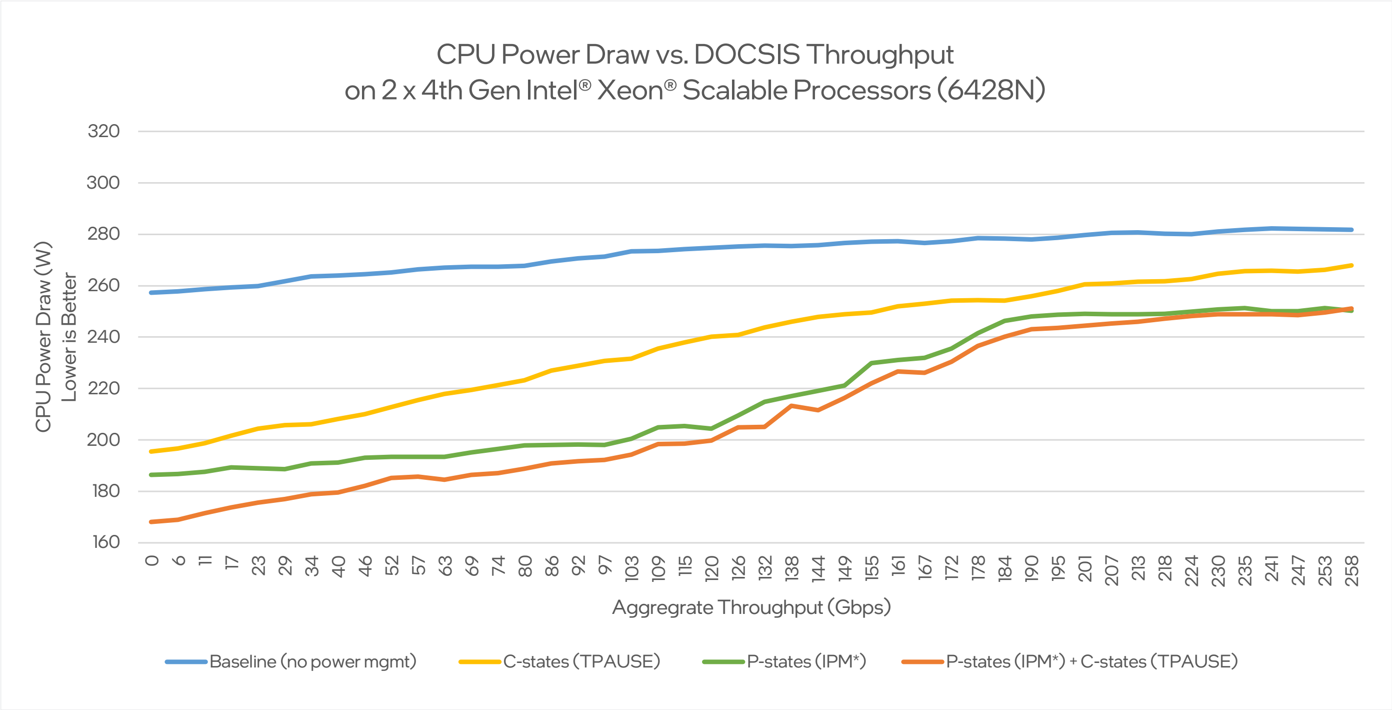


Figure 6. CPU power draw vs. DOCSIS throughput.

* IPM = Intel® Infrastructure Power Manager

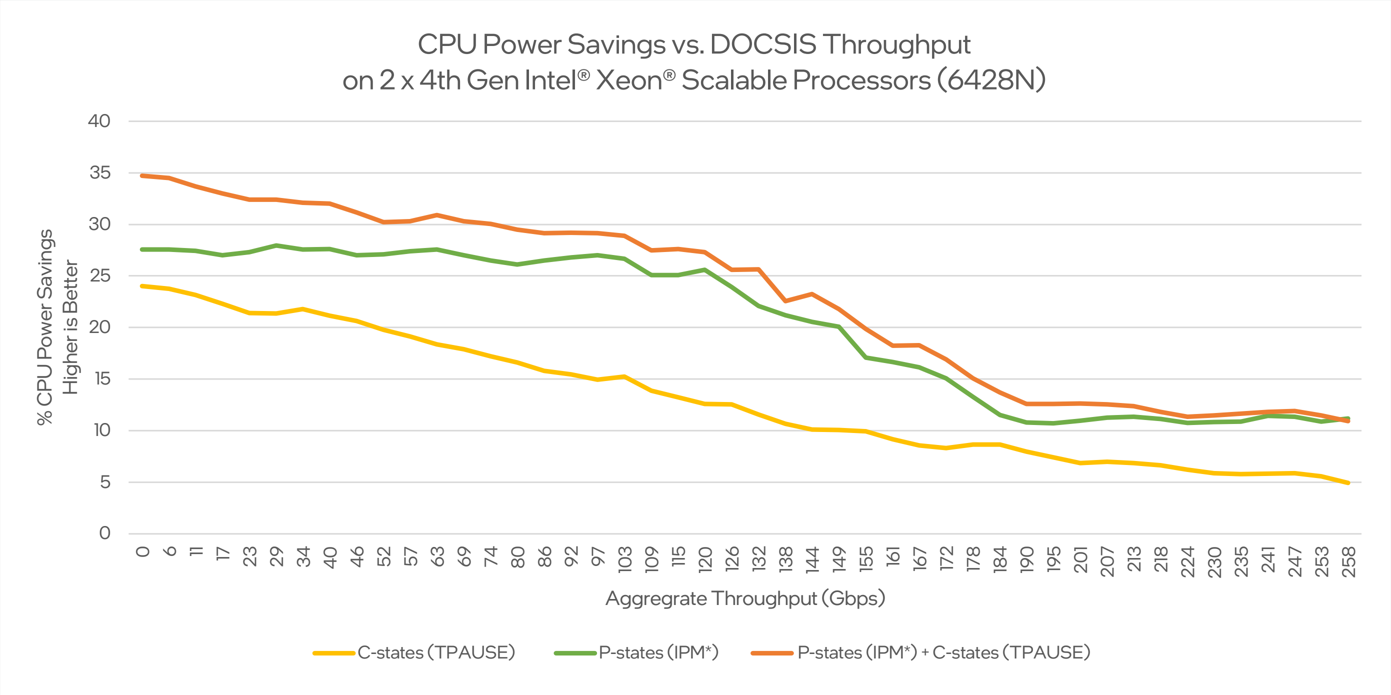


Figure 7. CPU power savings vs. DOCSIS throughput.

* IPM = Intel® Infrastructure Power Manager

While CPU power savings are important, most ISVs and MSOs are interested in the overall platform power savings. Figure 8 shows the total power draw of the Quanta Cloud Technology Inc. QuantaGrid D54Q-2U server for the baseline and three power management scenarios. It should be noted that the power management of other platform components (e.g., uncore, memory, etc.) are out of scope for this paper, with the sole focus on CPU power management and its impact on the power draw of the platform as a whole.

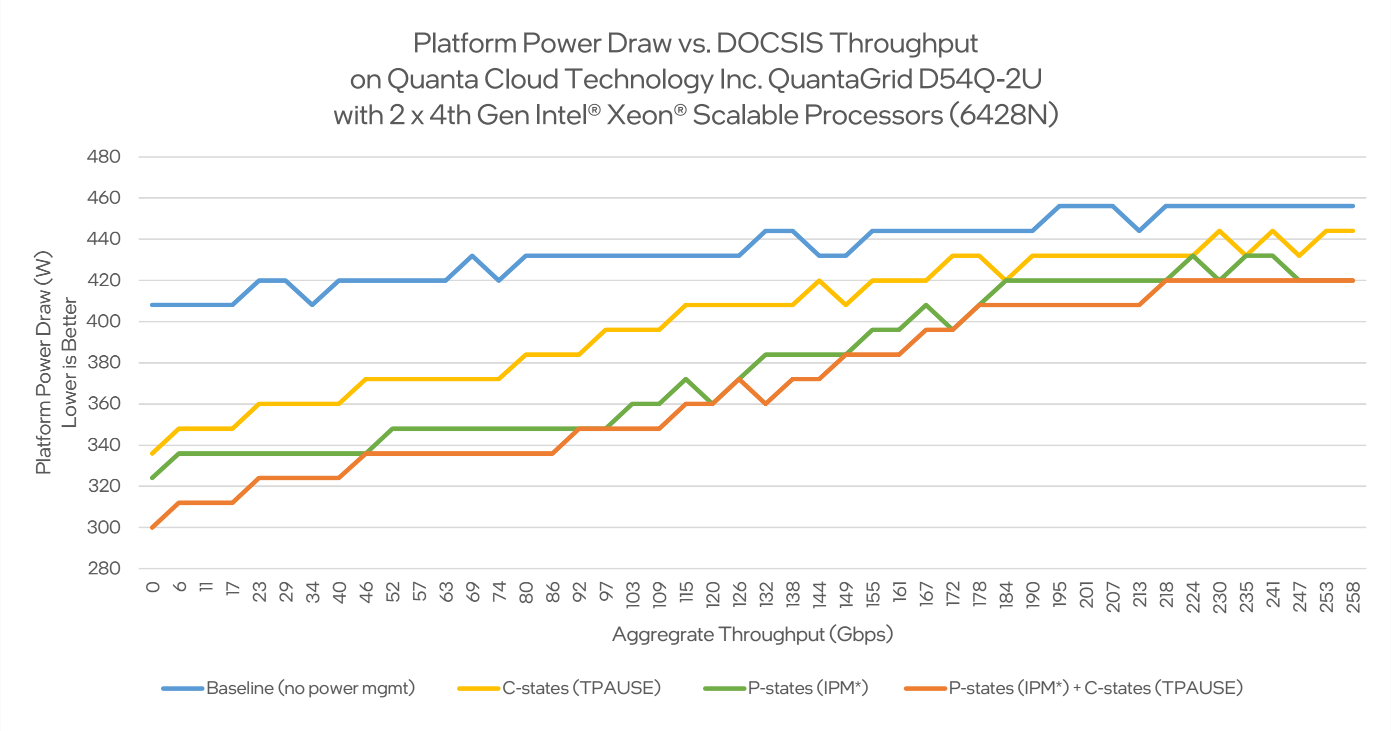


Figure 8. Platform power draw vs. DOCSIS throughput.

* IPM = Intel® Infrastructure Power Manager

The shape of the series in this graph is very similar to those of the CPU power draw graph, with greater savings as expected at the lower throughput rates and the combined usage of C-states and P-states, again giving the greatest savings overall. While the proportional power savings from the baseline at the platform level are less than those seen at the CPU level (due to the more constant power draw of platform peripherals, memory, etc.), the savings at the platform level are still significant, with up to:

- 72W (17.6%) savings when the new C-states are used.
- 84W (20.6%) when P-states are controlled by IPM.
- 108W (26.5%) when a combination of C-states and P-states are used.

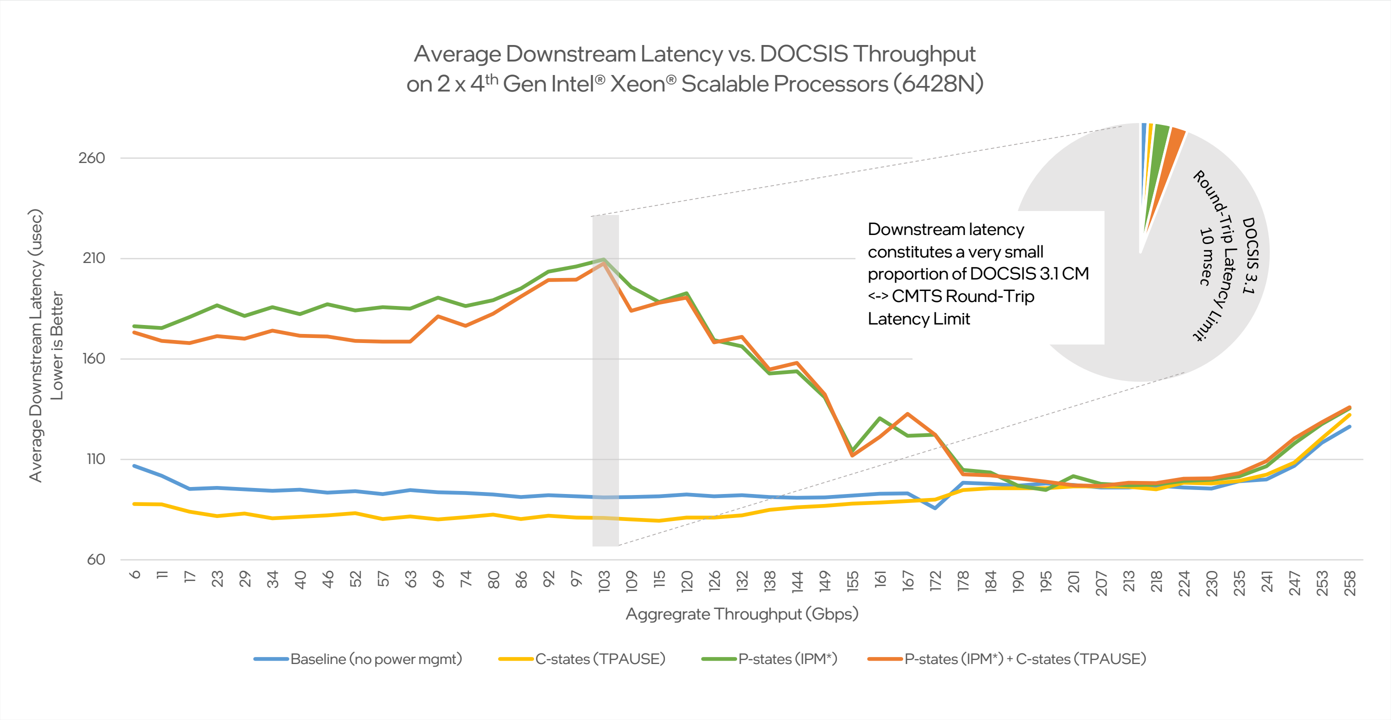
Latency Impact

The impact, if any, on application KPIs must be considered when discussing any potential power savings. Figure 9 shows the average latency of the vCMTS downstream data plane pipeline in microseconds for the baseline, C-states (enabled via User Wait TPAUSE), P-states (controlled via IPM), and combined C-states and P-states test runs. The X-axis shows the increasing DOCSIS throughput handled by the server, and the Y-axis shows the average application latency as measured from the Intel vCMTS Reference Dataplane. Also shown is the proportion of the overall 10 millisecond round-trip time between a cable modem and vCMTS that is accounted for by the vCMTS downstream pipeline latency for the four scenarios in a typical DOCSIS 3.1 network. In all scenarios, the downstream pipeline latency, which is in the order of sub-210 microseconds at all traffic rates, constitutes a very small proportion of the overall network latency and is well within the acceptable limits of a DOCSIS MAC data plane implementation.

The 'C-states (TPAUSE)' series shows little divergence from the baseline series in the graph, confirming that enabling this power management feature has no negative impact on the application average latency. This also proves that the algorithm monitoring the number of packets the application receives is extremely lightweight and does not impact application performance.

When IPM is used to manipulate P-states, the latency is unaffected at higher levels of traffic (as shown by the rightmost points of the graph). As IPM lowers the P-states on the CPU cores at lower traffic rates, the rate at which instructions are executed on the cores is reduced. This causes a proportionate increase in application latency. As already stated, however, this increased latency accounts for a near negligible increase in the total end-to-end latency of the DOCSIS access network. For the benchmarking described in this paper, IPM was configured to use a P-state range of 0.8 GHz to 1.8 GHz. The impact on average application latency can be reduced using a smaller P-state range (e.g., 1.2 GHz to 1.8 GHz). This allows for a trade-off between the power savings achieved and the impact on the application's average latency.

While the average latency is important, so is the application's maximum latency. The maximum latency shows the implications of transitioning in and out of power savings states (i.e., entering/exiting C-states and increasing/decreasing P-states). The maximum latency measured in the four scenarios is shown in Figure 10.


Average Downstream Latency vs. DOCSIS Throughput
on 2 x 4th Gen Intel® Xeon® Scalable Processors (6428N)

Downstream latency constitutes a very small proportion of DOCSIS 3.1 CM <-> CMTS Round-Trip Latency Limit

Figure 9. Downstream average latency vs. DOCSIS throughput.

* IPM = Intel® Infrastructure Power Manager

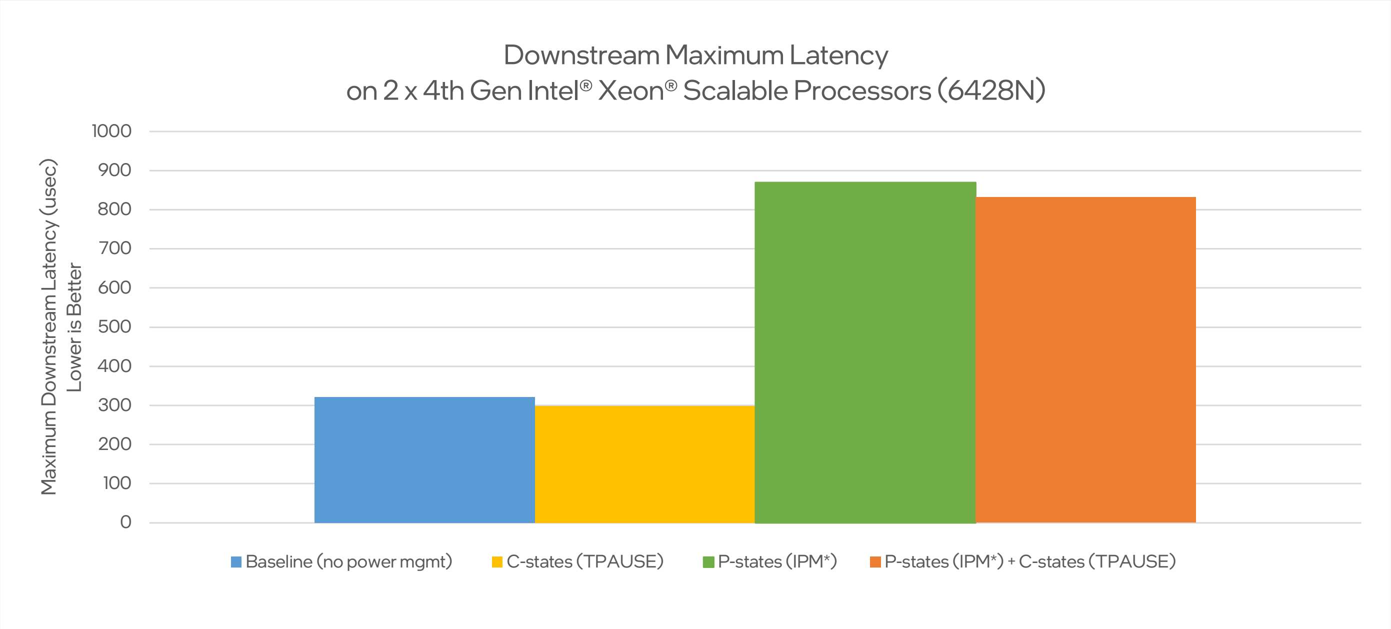


Figure 10. Downstream maximum latency.

* IPM = Intel® Infrastructure Power Manager

Once again, the use of C-states shows little or no difference in maximum latency when compared to the baseline. This is thanks to both the negligible exit latencies of the C0.1 and C0.2 states being used and to the fast response from the User Wait TPAUSE algorithm. However, there are more significant increases in maximum latency when P-states are manipulated. The increases are due to the CPU cores running at reduced P-states and the small latency associated with P-state transitions. If these increases are deemed too high, the impact can be reduced by configuring a smaller P-state range in IPM as previously described.

Packet Loss

It is imperative that, while power management techniques are being used, packet loss is avoided, and there is no degradation in service. Any increase in load must be quickly detected, and the power-optimized state exited so that packets are not dropped. For this purpose, a test was performed where traffic rates were rapidly increased from idle and packet drops were monitored.

No packet loss was recorded as traffic increased from idle to 258 Gbps (i.e., 90% of max load) of DOCSIS traffic when the User Wait TPAUSE algorithm alone was enabled. This is due to the short TPAUSE duration used (1 microsecond) and the negligible exit latencies of the C0.1 and C0.2 power-optimized states and confirms that the algorithm can react quickly enough to avoid dropping packets or increasing latency.

When IPM was used to control P-states, however, packet loss was recorded when traffic rates were increased in the same manner as above. The reason is twofold. First, the transition latency required to change P-states, although small, is not as low as that of C0.1 and C0.2 states. Second, as IPM is an external application, there is a small delay between the actual increase in traffic and IPM detecting said increase. However, during benchmarking, it was shown that IPM could react in a just-in-time fashion to rapid increases in network traffic from idle to 152 Gbps (or 53% of max) of DOCSIS traffic without any packet loss occurring. This capability is beyond sufficient for a cable access network.

Power Savings Over a 24-Hour Period

To quantify the potential daily power savings, a 24-hour traffic profile was applied over a condensed period, with a peak traffic rate of 258 Gbps (or 90% of the maximum possible load). At each data point of the traffic profile, CPU and platform power consumption were measured. Once again, the test was repeated for the baseline case with no power management and the three scenarios with the power management features enabled. Figure 11 shows the total power draw of the two CPUs on the left-hand-side Y-axis plotted against each data point from the 24-hour period on the X-axis, with a series for each of the four power management scenarios. The grey line on the graph shows the aggregate DOCSIS throughput and is plotted against the right-hand-side Y-axis.

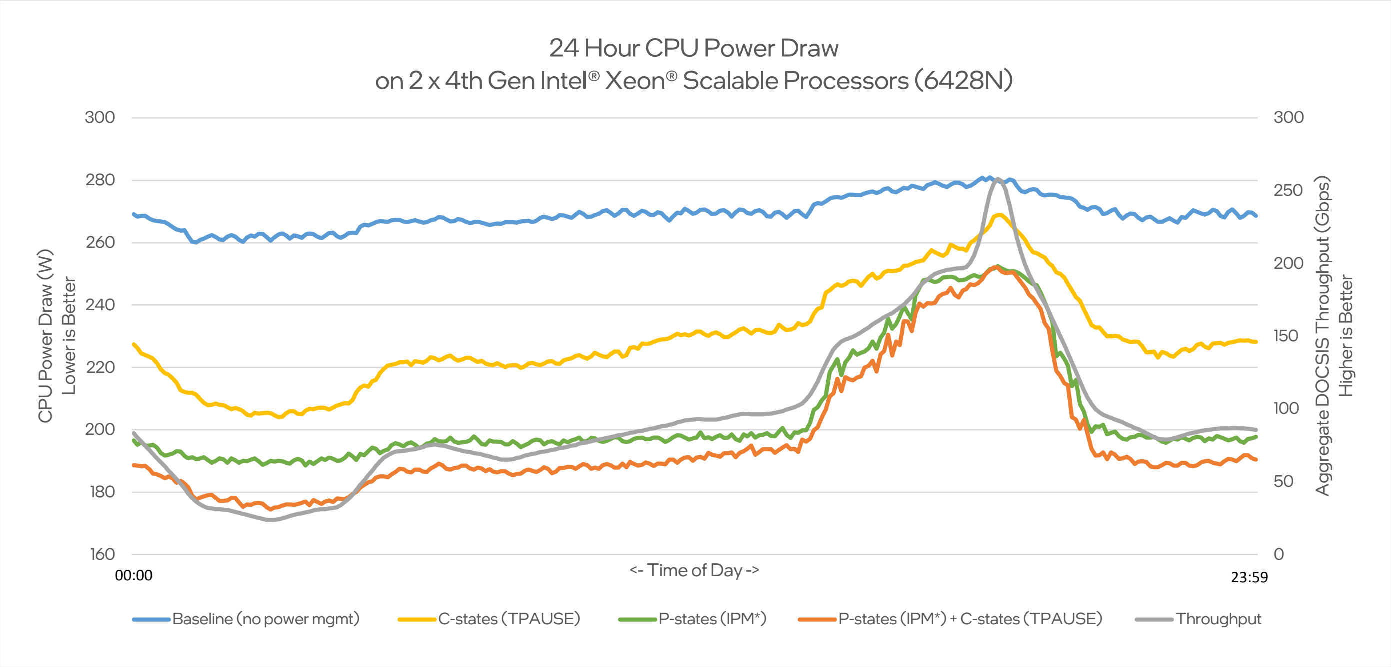


Figure 11. CPU power draw over 24 hours.

* IPM = Intel® Infrastructure Power Manager

The early morning hours, during which network load is hugely reduced, provide significant opportunities for power savings. Each of the techniques can achieve very good power savings during this period, with a reduction of up to 33.3% of CPU power achievable when combined usage of C-states and P-states (orange series) is employed. As network utilization fluctuates throughout the day, savings are still achieved by each technique. Across the 24-hour period, the possible savings average 26.8% for the combined usage of C-states and P-states and would considerably reduce OpEx. Given that the server can process the same throughput with or without the power management features enabled, this 26.8% average reduction in CPU power equates to a 37.6% improvement in performance per watt (Gbps/W) over a 24-hour period.

The percentage of CPU power saved throughout the day is further visualized in Figure 12, with the percentage of power saved shown on the Y-axis for each point in time.

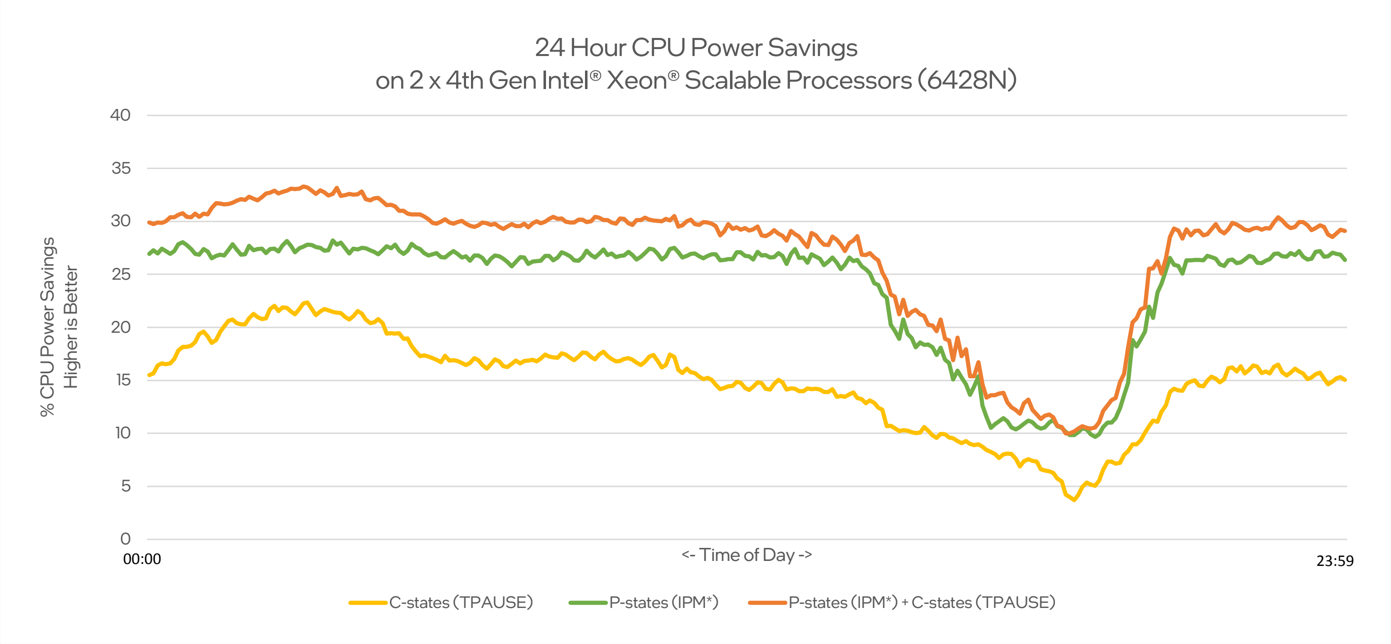


Figure 12. CPU power savings over 24 hours.

* IPM = Intel® Infrastructure Power Manager

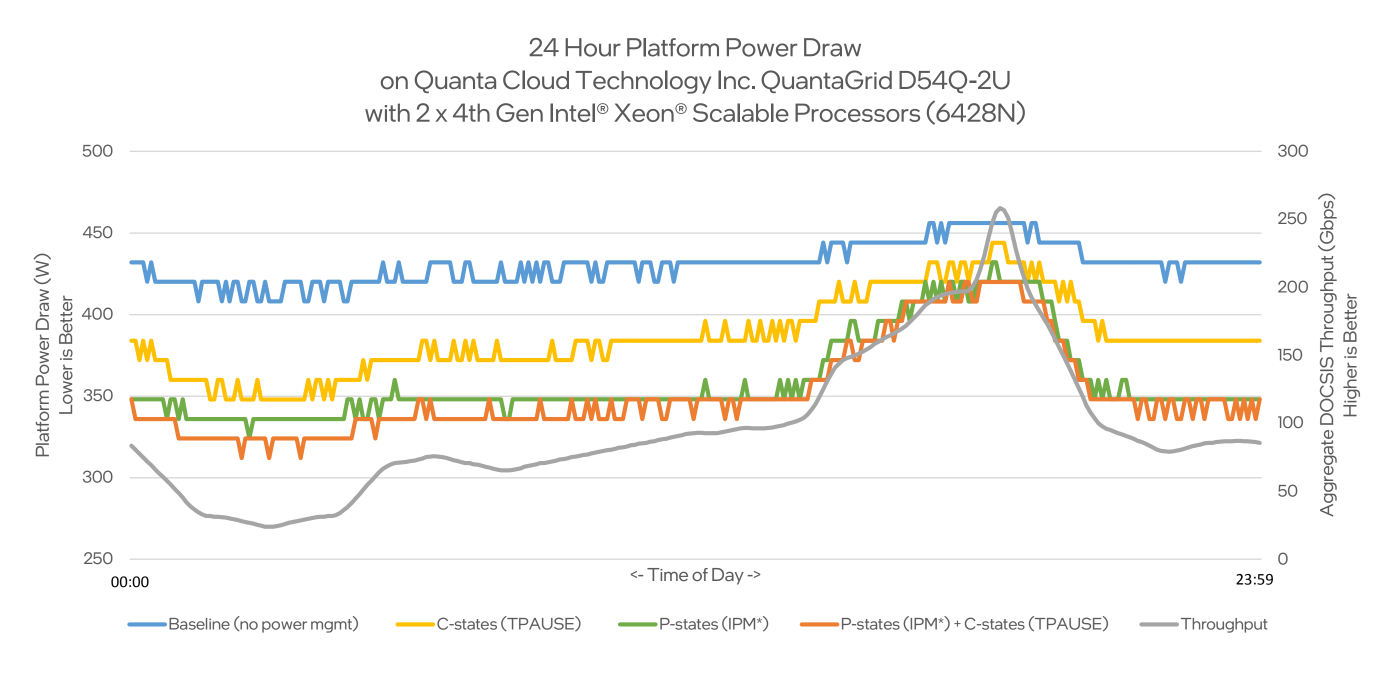


Figure 13. Platform power draw over 24 hours.

* IPM = Intel® Infrastructure Power Manager

The power draw across the Quanta Cloud Technology Inc. QuantaGrid D54Q-2U platform over the 24-hour period can be seen in Figure 13. Once again, the power draw profiles at the platform level closely match those at the CPU level. For combined usage of C-states and P-states, there is a maximum platform power saving of 25.7%, with an average across the day of 18.4%.

Table 1 summarizes the maximum, minimum, and average power savings, and the performance per watt increases at both the CPU and platform level for the three power management techniques when compared to the baseline over a typical 24-hour period. The greatest power savings are achieved, as expected, when both the C-states and P-states are utilized. But very significant savings can also be made if only one of the techniques is used. It must be reiterated that the power savings presented here are on a platform provisioned to handle 90% of its maximum possible load. These savings would be further increased if the platform was provisioned to handle even less capacity.

CPU Level	Power Savings over 24 hours			Average Perf. per Watt Increase over 24 hours
	Maximum	Minimum	Average	
C-states (TPAUSE)	22.4%	3.7%	14.9%	17.8%
P-states (IPM*)	28.2%	9.7%	23.9%	32.1%
P-states (IPM*) + C-states (TPAUSE)	33.3%	10%	26.8%	37.6%

Platform Level	Power Savings over 24 hours			Average Perf. per Watt Increase over 24 hours
	Maximum	Minimum	Average	
C-states (TPAUSE)	17.1%	2.6%	10.5%	11.2%
P-states (IPM*)	22.9%	5.3%	16.7%	20.3%
P-states (IPM*) + C-states (TPAUSE)	25.7%	7.9%	18.4%	22.9%

Table 1. Summary of power savings and performance per watt increases over 24 hours.

* IPM = Intel® Infrastructure Power Manager

Conclusion

As the deployment of vCMTS on Intel Xeon Scalable processors continues to accelerate, significant power-related OpEx savings are unlocked. Initially, power and space savings are achieved purely through the adoption of a virtualized deployment on commercial-of-the-shelf servers in place of single-purpose "big-iron" chassis. The urgency for a greener, more sustainable industry emphasizes the importance of further optimizing power consumption. As shown in this paper, several power management features and controls are available on Intel Xeon Scalable processors, enabling MSOs to realize further power savings on the path towards their climate goals.

4th Gen Intel Xeon Scalable processors have introduced new power-optimized C-states, namely C0.1 and C0.2, which can provide significant power savings for vCMTS solutions. Moreover, they provide power savings in an autonomous, management-free fashion without negatively impacting performance KPIs such as latency, throughput, or packet loss. Once enabled during vCMTS application initialization, no other code changes are needed to the application. Power management is entirely contained within the relevant layers of DPDK.

P-states, or core frequencies, have been available on all generations of Intel Xeon Scalable processors. Their suitability towards providing power savings for data plane applications, such as vCMTS, has improved on 3rd and 4th generation processors due to the reduction in P-state transition latency [9].

Furthermore, Intel has recently introduced the Intel Infrastructure Power Manager (IPM), which can monitor the real-time and true busyness of DPDK PMD-based workloads and dynamically adjust the frequency of the CPU cores to match the current load. Like the C-states, IPM can provide savings to a vCMTS deployment autonomously, with no significant impact on the performance KPIs. As a self-contained, workload-agnostic, and platform-feature-agnostic solution, IPM is an attractive power saving option for ISVs and MSOs who do not wish to change their vCMTS software in any way.

The above two power management techniques can be used on their own to provide substantial power savings or combined to provide even greater savings. A detailed configuration guide describing how to enable these techniques in a vCMTS solution is provided in [8].

The paper has shown that across a typical 24-hour period, average CPU power consumption can be reduced by up to 26.8% and average platform power by up to 18.4% by using a combination of C-states and P-states, with maximum CPU and platform power savings of 33.3% and 25.7% respectively at the lowest traffic periods. Table 1 summarizes the power savings that can be achieved using one or both techniques described, all of which give considerable increases in performance per watt. These power savings, coupled with the hands-free nature in which they can be achieved, make a vCMTS deployment on Intel Xeon Scalable processor-based servers the perfect solution for ISVs and MSOs seeking to maximize the energy efficiencies of their cable access network.

Appendix A: Intel vCMTS Reference Dataplane Packet-processing Pipeline Stages

The following describes the packet-processing stages of the Intel vCMTS Reference Dataplane upstream and downstream pipelines as shown in Figure 3.

Downstream Data Plane Pipeline Stages

1. Receive IP Frames

Using the DPDK Ethdev API, IP packet bursts are received via the DPDK Poll Mode Driver (PMD) from the Rx queue of a NIC virtual function (VF) port. These packets are read by a data plane software thread that begins vCMTS downstream packet processing. Packets are steered to a service group specific VF-based on the destination MAC address.

2. Cable Modem Lookup & Subscriber Management

The DPDK Hash API is used to do a bulk lookup (i.e., with multiple packets) based on the destination IP address of the received frames to retrieve cable modem records containing MAC address, DOCSIS filter, DOCSIS classifier, service flow queue, and security info. The Destination MAC address of the Ethernet frame is also updated to a cable modem-specific address in this stage. The number of active subscriber IP addresses is checked against the DOCSIS limit (tracked by a destination IP address list per cable modem).

3. DOCSIS Filtering

The DPDK Access Control List (ACL) API is used to apply an ordered list of DOCSIS filter rules to Ethernet frames. DOCSIS filter rule configuration is described in Appendix C.

4. DOCSIS Classification

The DPDK ACL library is used to apply an ordered list of rules to classify Ethernet frames for enqueuing to cable modem service-flow scheduler queues. DOCSIS service-flow scheduler rule configuration is described in Appendix C.

5. DOCSIS QoS – Service Flow & Channel Access Scheduling

The DPDK hierarchical QoS (HQoS) scheduler API is used to apply rate-shaping, congestion control, and weighted-round-robin (WRR) scheduling to cable modem service flow queues. The DPDK Scheduler API has also been adapted to perform channel access scheduling on data packets after service-flow scheduling. Channel access scheduling is optimized by performing it in an earlier pipeline stage than is typically done in other implementations. This scheduling stage takes into account the DEPI and DOCSIS encapsulation overhead added later in the pipeline.

6. Lower MAC Interface

A DPDK ring transfers packets between upper MAC and lower MAC processing. This allows upper and lower MAC processing to be executed on separate threads.

7. DOCSIS Framing

DOCSIS MAC headers are generated, including DOCSIS header check sequence (HCS), for prepending to packets. The DPDK CRC API is used to generate the DOCSIS HCS. Intel® AVX-512 instructions are used for optimum performance on 4th Gen Intel Xeon Scalable processor platforms.

8. IP Frame CRC Generation and DOCSIS BPI+ Encryption

The packet's 32-bit Ethernet cyclic redundancy code (CRC) is generated, and DOCSIS baseline privacy interface (BPI+) encryption is applied. These two stages are performed using DPDK combined crypto-CRC processing. Intel® AES-NI, Intel AVX-512 and other vectorized instructions, and Intel® QuickAssist Technology (Intel® QAT) are used for optimum performance on 4th Gen Intel Xeon Scalable processor platforms.

9. DEPI Encapsulation

DEPI encapsulation is performed based on the DOCSIS 3.1 specification. Frames are converted to Packet Streaming Protocol (PSP) segments, concatenated using DPDK Mbuf chaining, and encapsulated into L2TP frames of maximum transmission unit size. PSP segments are fragmented across DEPI frames, so all transmitted frames are of maximum transmission unit (MTU) size to ensure maximum utilization of the Remote-PHY (R-PHY) link.

10. Transmit DEPI/L2TP Frames

Using the DPDK Ethdev API, bursts of DEPI/L2TP frames are transmitted via the DPDK PMD to the NIC VF Tx queue of the associated service group.

Upstream Data Plane Pipeline Stages

1. Receive UEPI/L2TP Frames

Using the DPDK Ethdev API, bursts of L2TP/IP frames containing UEPI-encapsulated DOCSIS streams are received via the DPDK PMD from the Rx queue of a NIC VF port. These frames are read by a data plane software thread that begins vCMTS upstream packet processing. Frames are steered to the service group-specific VF based on the destination MAC address.

2. Validate Frame and Strip IP Headers

The L2TP/IP frame is validated, and IP headers are stripped.

3. UEPI Decapsulation

UEPI decapsulation is performed based on the DOCSIS 3.1 specification. UEPI/PSP sequence numbers are verified to be in order.

4. Service ID Lookup and DOCSIS Segment Reassembly

UEPI PSP header, data, and trailer segments are traversed, and the data segments are reassembled into DOCSIS stream segments. The DPDK Hash API performs lookups based on service ID values to retrieve cable modem info.

5. DOCSIS Frame Extraction and Upstream Scheduling

DOCSIS frames are extracted from DOCSIS stream segments, including the reassembly of fragmented frames using the DPDK Mbuf API. Any bandwidth requests extracted from the UEPIs are forwarded to an upstream scheduler at this point.

6. DOCSIS Frame HCS Verification

Header check sequence (HCS) verification is performed for the extracted DOCSIS frames using the DPDK CRC API.

7. DOCSIS BPI+ Decryption and IP Frame CRC Verification

DOCSIS BPI+ decryption is applied to DOCSIS frames for AES or DES-encrypted frames, and the 32-bit Ethernet CRC of the resulting Ethernet packet is verified. These two stages are performed using the DPDK combined crypto-CRC processing. Intel AES-NI, Intel AVX-512, and other vectorized instructions are used for optimum performance on 4th Gen Intel Xeon Scalable processor platforms. Note that CRC verification is generally not required for upstream encapsulated packets, so it is disabled by default.

8. Transmit IP Frames

Using the DPDK Ethdev API, bursts of IP frames are transmitted via the DPDK PMD to the NIC VF Tx queue of the associated service group.

Appendix B: Performance/Power Test Environment

The performance and power test environment for the Intel vCMTS Reference Dataplane consists of a vCMTS node and a software-based traffic-generator node, as shown in Figure 14.

The vCMTS node is based on a server blade with dual 4th Gen Intel Xeon Scalable processors, with 100GbE Intel® Ethernet 800 Series NICs and on-CPU Intel QAT devices.

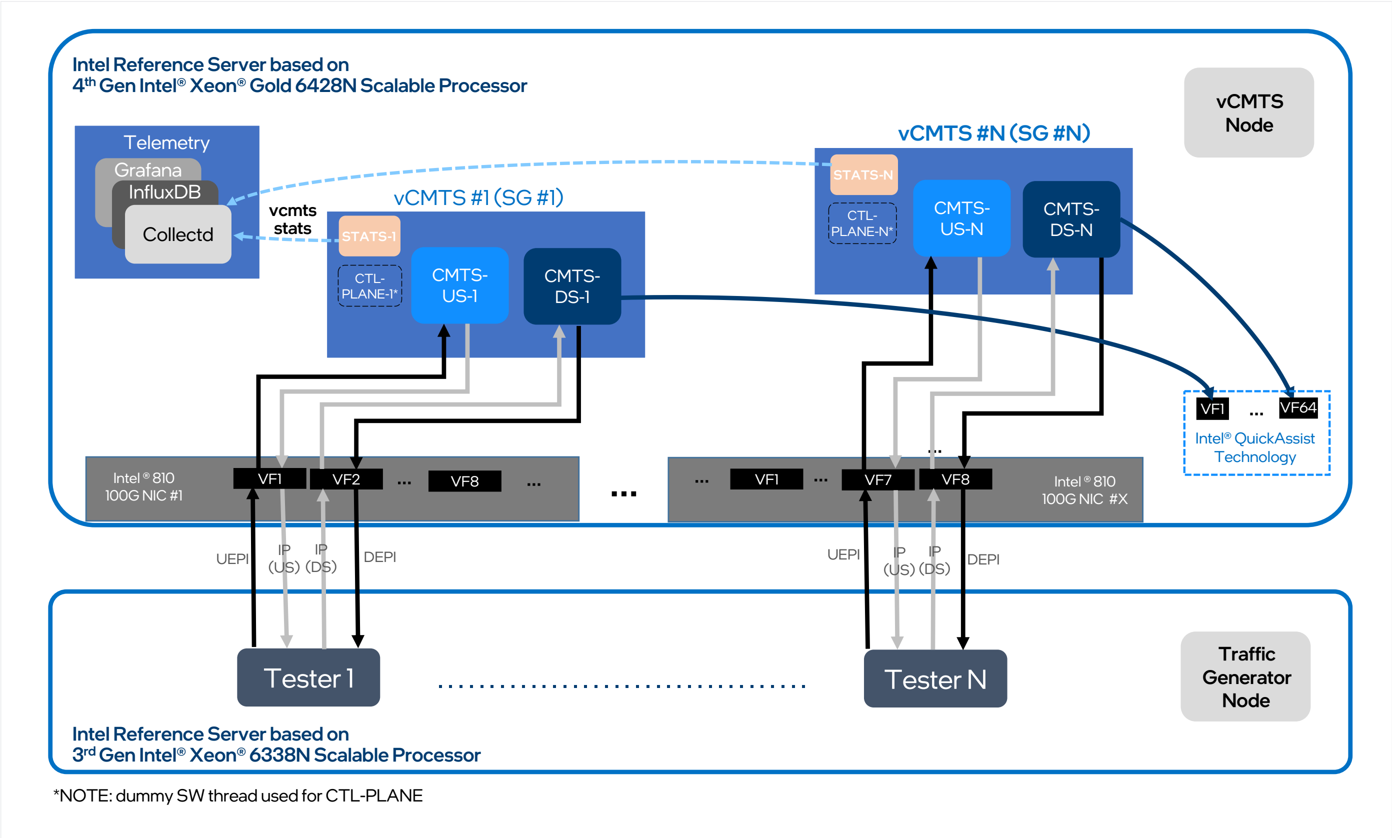
The traffic-generator node is based on a server blade with dual 3rd or 4th Gen Intel Xeon Scalable processors[1], again with 100GbE Intel Ethernet 800 Series NICs.

Servers based on other Intel Xeon Scalable processors or Intel Xeon D processors with different core counts are also supported. Different types of Intel Ethernet NICs may also be used.

On the vCMTS node, multiple vCMTS data plane instances run DPDK-based DOCSIS MAC upstream and downstream data plane processing pipelines for individual cable service groups. On the traffic-generator node, DPDK Pktgen-based traffic tester instances simulate traffic into corresponding vCMTS data plane instances.

During the benchmarking tests, all the vCMTS data plane instances, each representing an individual service group, handle the same traffic rates at the same time. The tests DO NOT consider a scenario where different vCMTS data plane instances handle different traffic rates.

For this paper, the Intel vCMTS Reference Dataplane was deployed on a bare-metal Linux stack. However, it can also be deployed in a Kubernetes-orchestrated environment based on either the Bare Metal Reference Architecture (BMRA) for containers from Intel or on the Red Hat OpenShift Container Platform.



*NOTE: dummy SW thread used for CTL-PLANE

Figure 14. Performance test environment used to measure vCMTS performance and power.

Appendix C: Test Environment Configuration Information and Relevant Variables

Service Groups	28 (14 per CPU)
CM Lookup and Subscriber Management	300 subscribers per service group, 4 IP addresses per subscriber
DOCSIS Filtering	22 filter groups, 22 filter rules per group (18 IPv4 rules, 4 IPv6 rules) 1 filter group per cable modem (i.e., same filter group for all CPE types) 10% matched, 90% unmatched (default action – permit)
DOCSIS Classification	16 rules per subscriber 10% matched – enqueue to one of 3 service-flow queues 90% unmatched – enqueue to default service-flow queue
Downstream Service-Flow Scheduling	8 service-flow queues per subscriber (m4 active)
Downstream Channel Scheduling	6 x OFDM (1.89 Gbps) channels, 2 x channel-bonding groups NOTE: channel-bonding groups are distributed evenly across cable modems
Upstream Bandwidth Scheduling	Upstream scheduler not used Upstream bandwidth pre-allocated in grants of 2KB per service ID. Bandwidth grants balanced evenly across 300 cable modems
Ethernet CRC	Downstream: 100% CRC re-generation Upstream: 0% CRC verification NOTE: CRC relates to inner frames
Encryption	100% AES, 0% DES NOTE: all crypto processing performed using the Intel® Multi-Buffer Crypto for IPSec library
Packet IMIX Distribution	Upstream 65% : 70B, 18% : 256, 17% : 1280B Downstream 15% : 84B, 10% : 256B, 75% : 1280B
Power Management	User Wait TPAUSE and/or IPM Enabled User Wait Pause Duration: 1 microsecond User Wait Empty Poll Threshold: 32 IPM P-state range: 0.8 GHz – 1.8 GHz
Latency Statistics	Enabled
Core Configuration	1us1t_1ds2t: 1 single-threaded upstream pipeline per core, 1 dual-threaded downstream pipeline per core, stats/management thread on hyper-thread sibling of upstream pipeline
Downstream NIC RXQ Size	512

Appendix D: System Configuration

vCMTS Server – based on 4th Gen Intel® Xeon® Scalable Processor	
Hardware	
Platform	Quanta Cloud Technology Inc. QuantaGrid D54Q-2U
CPUs	2 x 4th Intel® Xeon® Gold 6428N Processors, 1.8 Ghz, 32 cores Uncore Frequency: 1.6 GHz Microcode: 0x2B000181
Memory	16 x 32GB DDR5 4800 MT/s [4000 MT/s]
Hard Drive	1 x 894.3G Intel® SSDSC2KG96
Network Interface Cards	2 x Intel® Ethernet Network Adapter E810-2CQDA2 (1 per CPU)
Software	
Host OS	Ubuntu 22.04, Linux Kernel v5.15.x

Kernel Options	default_hugepagesz=1G hugepagesz=1G hugepages=72
	intel_iommu=on iommu=pt
	intel_pstate=disable
	isolcpus=2-31,34-63,66-95,98-127
	rcu_nocbs= 2-31,34-63,66-95,98-127
	nohz_full= 2-31,34-63,66-95,98-127
	nr_cpus=128
	vfio-pci.disable_denylist=1
	nmi_watchdog=0 audit=0 nosoftlockup hpet=disable mce=off tsc=reliable numa_balancing=disable memory_corruption_check=0

Data Plane Development Kit (DPDK)	DPDK v22.07, with following patches applied: ▪ All patches in Intel® vCMTS Reference Dataplane v22.10.0 package ▪ Pause patch for DPDK rings [7] ▪ Telemetry patches for IPM [11]
Intel® Multi-Buffer Crypto for IPSec	intel-ipsec-mb v1.3
Intel® Infrastructure Power Manager (IPM)	IPM v23.06
vCMTS	Intel® vCMTS Reference Dataplane v22.10.0, with updates to: ▪ Enable User Wait TPAUSE for NIC and Ring interfaces
Tested by Intel on July 14, 2023	

Traffic-Generator Server	
Hardware	
Platform	Intel® Server Board M50CYP2SBSTD
CPUs	2 x 3rd Gen Intel® Xeon® Gold 6348 Processors, 2.2 GHz, 26 Cores
Memory	16 x 16GB DDR4 3200 MT/s [3200 MT/s]
Hard Drive	1 x 447.1G Kingston SA400M8480G
Network Interface Cards	2 x Intel® Ethernet Network Adapter E810-2CQDA2 (1 per CPU)
Software	
Host OS	Ubuntu 22.04, Linux Kernel v5.15.x

Kernel Options	default_hugepagesz=1G hugepagesz=1G hugepages=72
	intel_iommu=on iommu=pt
	isolcpus=2-27,30-55,58-83,86-111
	nr_cpus=112
	vfio-pci.disable_denylist=1

Data Plane Development Kit (DPDK)	DPDK v20.08
Traffic-Generator	DPDK Pktgen v19.10.0

Appendix E: Acronyms and Definitions

Acronym or Term	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Programmable Interface
AVX-512	Advanced Vector Extensions 512
BMRA	Bare Metal Reference Architecture
BPI	Baseline Privacy Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DEPI	DOCSIS External Downstream Interface
DES	Data Encryption Standard
DOCSIS	Data Over Cable Service Interface Specification
DPDK	Data Plane Development Kit
GHz	Gigahertz
Gbps	Gigabits Per Second
HCS	Header Check Sequence
HQoS	Hierarchical Quality of Service
Intel® QAT	Intel® QuickAssist Technology
IP	Internet Protocol
IPM	Intel® Infrastructure Power Manager
ISV	Independent Software Vendor
KPI	Key Performance Indicator
L2TP	Layer 2 Tunneling Protocol
MAC	Media Access Control
MSO	Multiple system Operator
MTU	Maximum Transmission Unit
NIC	Network Interface Card
OpEx	Operational Expenditure
OS	Operating System

PHY	Physical Radio Frequency Layer
PMD	Poll Mode Driver
PSP	Packet Streaming Protocol
R-PHY	Remote-PHY
Rx	Receive
TCO	Total Cost of Ownership
TDP	Thermal Design Power
Tx	Transmit
UEPI	Upstream External Physical Interface
vCMTS	Virtualized Cable Modem Termination System
VF	Virtual Function
W	Watts
WRR	Weighted Round Robin

References

[1] D. Coyle, K. O'Sullivan, M. A. Siddiqui and S. Ravisundar, "Maximizing vCMTS Data Plane Performance with 4th Gen Intel® Xeon® Scalable Processor Architecture," [Online]. Available: https://www.intel.com/content/www/us/en/content-details/784410/maximizing-vcmts-data-plane-performance-with-4th-gen-intel-xeon-scalable-processor-architecture.html?DocID=784410.

[2] T. Muders, R. Elftmann, T. Nguyen and E. Heaton, "Vodafone Network Evolution Paves the Way for Energy Savings," [Online]. Available: https://www.vodafone.com/news/technology/vodafone-intel-paper-lower-network-energy-bills-better-environmental-footprint.

[3] "DPDK (Data Plane Development Kit)," Linux Foundation Projects, [Online]. Available: https://www.dpdk.org/.

[4] K. Devey, D. Hunt and C. MacNamara, "Power Management - Technology Overview," [Online]. Available: https://builders.intel.com/docs/networkbuilders/power-management-technology-overview-technology-guide.pdf.

[5] "Intel® 64 and IA-32 Architectures Software Developer Manuals," [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html.

[6] "DPDK Power Management Library," [Online]. Available: https://doc.dpdk.org/guides/prog_guide/power_man.html.

[7] "Pause Patch for DPDK Rings," Intel Corporation, [Online]. Available: https://patches.dpdk.org/series/27919/mbox.

[8] R. Sexton, D. Coyle and D. Przychodni, "Guidelines for Optimizing Power Consumption of vCMTS Deployments on Intel® Xeon® Scalable Processors," [Online]. Available: https://www.intel.com/content/www/us/en/content-details/778903/guidelines-for-optimizing-power-consumption-of-vcmts-deployments-on-intel-xeon-scalable-processors.html.

[9] D. Hunt, R. Pattan, C. MacNamara, K. Shemer, N. Palit and P. Shah, "Enhanced Power Management for Low-Latency Workloads," [Online]. Available: https://networkbuilders.intel.com/solutionslibrary/power-management-enhanced-power-management-for-low-latency-workloads-technology-guide.

[10] "Dynamically Tune Intel® CPUs to Maximize Network Energy Efficiency," [Online]. Available: https://www.intel.com/content/www/us/en/wireless-network/core-network/infrastructure-power-manager-solution-brief.html.

[11] "Telemetry Patches for IPM," [Online]. Available: https://github.com/intel/CommsPowerManagement/tree/master/ipm/patches.

[12] "Intel® vCMTS Reference Dataplane," [Online]. Available: https://www.intel.com/content/www/us/en/developer/topic-technology/open/vcmts-reference-dataplane/overview.html.

Footnotes

1. Performance and power measured using the test environment, scenario and system configuration described in Appendix B, C and D. Results based on a different test environment, scenario or system configuration may differ. Note that there will be a margin of error due to the action of taking performance and power measurements. Results shown are for a reference implementation of a vCMTS data plane and not a production system. These numbers should be treated strictly as a reference only.

2. Hyper-threaded siblings are hardware threads of execution contained within the same physical CPU core and which share the same set of core resources. For data plane cores on the Intel® vCMTS Reference Dataplane system, each hyper-thread runs its own data plane software thread.

Notices & Disclaimers

Performance varies by use, configuration, and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software, or service activation.

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.

The products described 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.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a nonexclusive, royalty-free license to any patent claim thereafter drafted, which includes subject matter disclosed herein.

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

1123/BB/DJA/HO9/PDF ♻ Please Recycle 356668-001US