White Paper

Converged Edge Media Platform

Accelerating the Deployment and Scaling of Multiple Media and AI Microservices at the Network Edge

Authors:

Gang Shen, Intel Corporation
Brandon Gavino, Intel Corporation
Arindam Saha, Intel Corporation

Table of Contents

The vision of edge media 1
- Motivation 1
- The transformations of architecture ... 2

The Converged Edge Media Platform (CEMP) 2
- Architecture 2
- Media services at the edge 5

Highlighted features 5
- Orchestration of microservices 5
- Intel® Xeon® Scalable CPU optimizations 5
  - Core frequency scaling 5
  - NUMA alignment 6
- Intel® Data Center GPU Flex Series optimizations 7
- Platform awareness 7
- Intel device plugins for media accelerations 8

Summary 8

The vision of edge media

Motivation

Network Edge computing is undergoing tremendous growth, spurred by the proliferation of a variety of low-latency, high-bandwidth, secure media, and AI services, many of which are interactive and immersive in nature. Migration of computing to the network edge is happening from centralized public clouds for applications like streaming, CDNs, and cloud gaming. Additionally, applications like AI media enhancements and analytics are moving from an increasingly over-provisioned on-premises edge to the network edge. As a result, communications service providers (CoSPs), ISPs, CoLos, CDN, and gaming providers are building out their network edge infrastructure on a massive scale. The edge in this context is primarily the network edge managed by telco operators, as shown in Figure 1.

The inherently distributed nature of the edge, along with the space and power constraints of the edge, must be considered while deploying these services. Doing it in a siloed fashion, service by service, is inefficient and cost-prohibitive. Instead, a cloud-native architecture that leverages the proven cloud computing technologies but is optimized for the edge is needed to handle the continually changing combinations of workloads, with elasticity to address fluctuating traffic levels and support diverse service level agreements. Such a platform should be extensible and modular and can serve as a common foundation for the multitude of edge media and AI services. Intel's Converged Edge Media Platform (CEMP), a vertical instantiation of the Intel® Tiber™ Edge Platform optimized for media applications, is such a platform, designed to take advantage of the wide array of heterogeneous hardware offerings from Intel, including the latest CPUs, GPUs, accelerators, networking, and storage geared toward the edge.

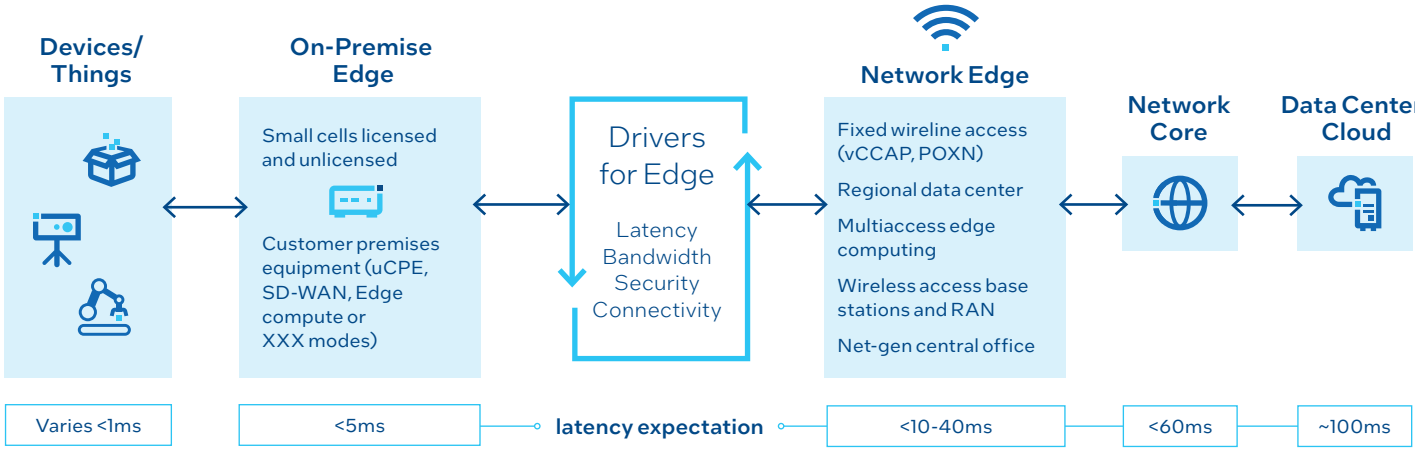

Figure 1. The devices, on-premises edge, network edge, network core, and cloud continuum[1]

As telco operators strive toward monetizing their 5G investments, they offer innovative services requiring them to augment their network edge with more computing and AI capabilities. One of the obvious advantages of edge networks is that the latency between client devices and servers is greatly reduced. Edge computing is essential for many latency-sensitive new media use cases, such as cloud gaming, XR (XReality), digital twin and collaborative conferencing (for example, metaverse), as well as AI-related workloads such as real-time video analytics, digitized and multimodal AI applications such as intelligent road surveillance (V2X), Cashierless Store and TeleHealth.

The transformations of architecture

Since the edge is a new element in the global network infrastructure, it triggers new challenges. The introduction to edge computing does not work like moving the traditional workloads from cloud data centers to a server node in the edge location.

From the network perspective, the traditional traffic pattern between client devices and servers in cloud data centers will be intercepted by the edge — new traffic controls and orchestrations are needed to harmonize the *computation and communication* among client, edge, and cloud.

From the software architecture perspective, the definition of the 5G/6G core network will be a service-based architecture (SBA), where all functions are implemented as a set of software-defined services reachable over an HTTP-based service interface. To leverage the benefits of TCO-friendly orchestration of SBA, domain-specific services and workloads (such as media and AI) will also require a transformation toward a microservice-based cloud-native architecture.

There are foreseeable transformations at multiple levels:

- **Transformation of the workloads to microservice-oriented architecture**: Workloads will be dynamically deployed in different edge clusters based on demands — i.e., workloads may come and go in an edge node anytime. Conversely, supporting multiple forthcoming workloads requires a set of unified, domain-specific middleware/platform microservices. The traditional, pre-provisioned, and monolithic architecture cannot provide such scalability, portability, reusability, and manageability in this dynamic edge environment.
- **Transformation of the platform software**: Like an operating system, the cloud-native platform (or platform-as-a-service) becomes the key to this connected computing paradigm. To interface hardware capabilities, the platform is supposed to manage all compute and network resources — adaptation, virtualization, observability, extension, and utilization; to interface workloads, the platform is supposed to provide container runtime, resource management, orchestration (deployment, scaling, and scheduling) and foundational domain-specific functionality (such as AI engine and media engine). Kubernetes and its derivative platforms are widely adopted and actively evolving.
- **Transformation of hardware**: Dictated by the plethora of workloads and applications, the hardware devices at the edge and the cloud have become more diversified, for example, the recent adoption of GPU, IPU, and APU, augmenting the CPU and built-in acceleration within the CPU. It is also notable that disaggregated hardware configurations have become critical for enabling low-latency, network-attached everything at the edge. Disaggregated configurations, together with virtualization, containerization, and cloud-native architectures, make resource consumption on edge clusters more efficient in terms of operational costs.

Intel® Xeon® Scalable processors, along with technologies like Intel® Speed Select Technology (Intel® SST), Intel® Resource Director Technology (Intel® RDT), Intel® Data Streaming Accelerator (Intel® DSA), and Intel® Data Center GPU Flex Series, paved the foundation of computation and communication for these transformations in the network edge. Intel is a leader in cloud and edge software (for example, Intel is a major contributor to Kubernetes and Linux open-source technologies), defining software to deliver the full potential of both cloud-native platforms and heterogeneous hardware. Intel's Converged Edge Media Platform (CEMP) is a full-stack reference solution with Intel software and hardware (cited in this document). It provides dedicated recipes for media workload vendors and edge infrastructure providers.

The Converged Edge Media Platform (CEMP)

Architecture

The primary goal of CEMP is to showcase Intel's software and hardware so that the myriad of current and emerging media and edge AI services run best on Intel. The design of CEMP is modular, so each component and feature can be delivered independently and accommodated into customers' existing cloud-native edge platform.

CEMP is based on the Kubernetes framework and supports container deployment of containerized and VM-based workloads. The CEMP architecture, as shown in Figure 2, is a layered design. Starting with the various media and AI workloads on the top, then the Optimized Edge Media Software and the Edge Native Framework, the Operating System, and finally, the hardware at the bottom, the CEMP maps to the SaaS, PaaS, CaaS, and IaaS layers of conventional cloud computing architecture.

CEMP is designed to be independent of the specific Kubernetes distribution. It supports open-source Kubernetes and vendor offerings like the Red Hat OCP (OpenShift Container Platform).

Intel's hardware products and hero features in the bottom layer are exposed and managed by the middle software layers, which are then used by the workloads. The hardware Kubernetes device plugins are one of the key differentiators in the CEMP stack. The middle software layer has two categories of microservices: container management services and domain-specific shared services (i.e., media-specific). Those two categories of services are functionally independent but collaborate during operations such as auto-scaling and scheduling.

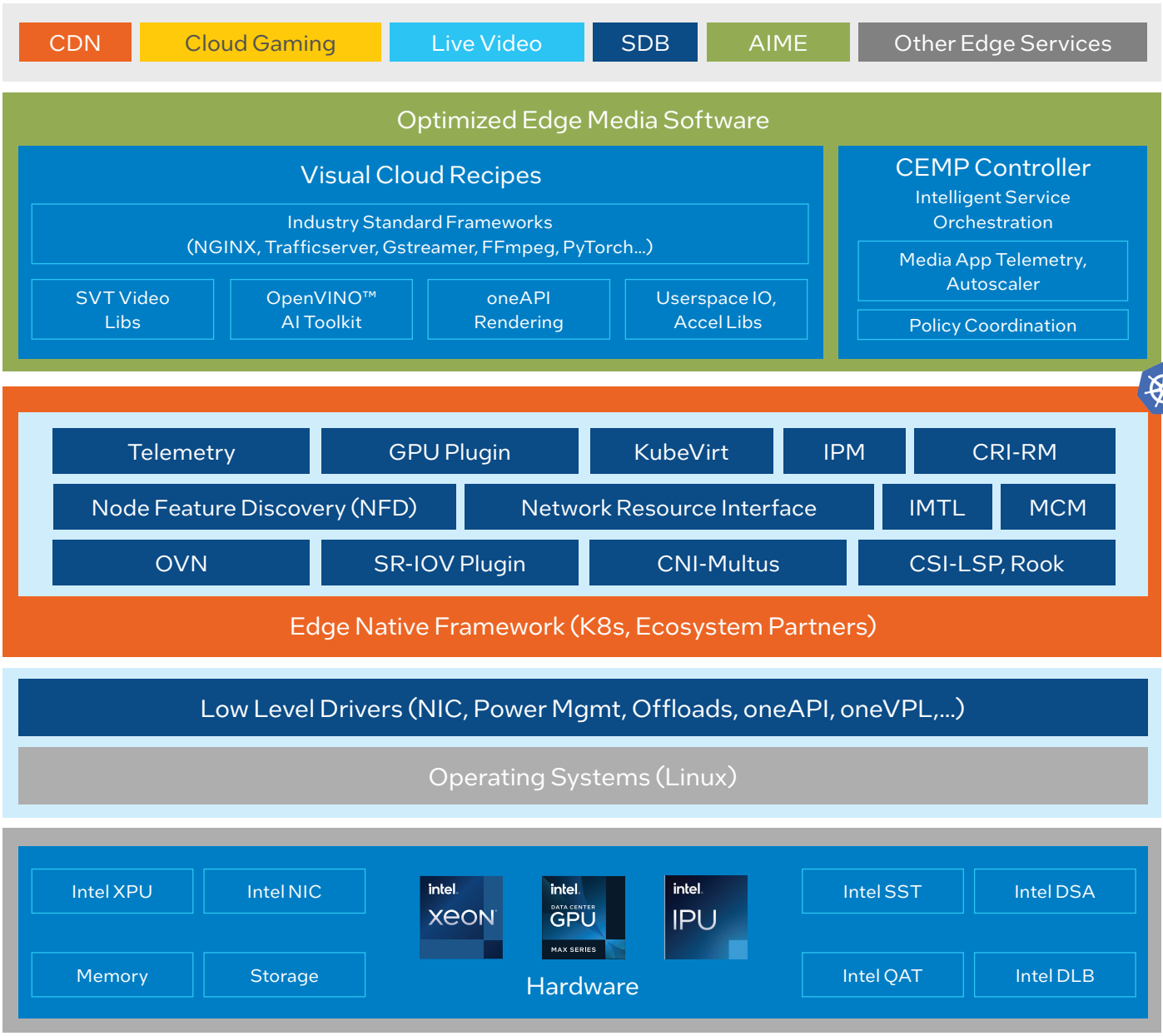

Figure 2. Converged Edge Media Platform Architecture.

CEMP can be deployed as one node, multi-node cluster, and even multi-cluster. The cluster cardinality, especially the number of worker nodes, depends on the applications and services that dictate the required compute performance, network bandwidth, and storage capacity. While clusters can have just one controller node, fault tolerance, and high availability require multiple controller nodes with failover capability. Figure 3 shows a 4-node CEMP cluster with multi-stream live transcode, live and VOD streaming, CDN (Content Delivery Network), and cloud gaming services orchestrated by open-source Kubernetes. CEMP has also infused AI with media, as shown in the 7-node cluster in Figure 4, supporting live streaming, AI-enhanced video Super Resolution, live video transcoding, and Intel OpenVINO™-based pose detection AI inferencing orchestrated by Red Hat's OCP (OpenShift Container Platform). Both clusters are built on heterogeneous hardware including both Intel CPUs and Intel Flex GPUs.

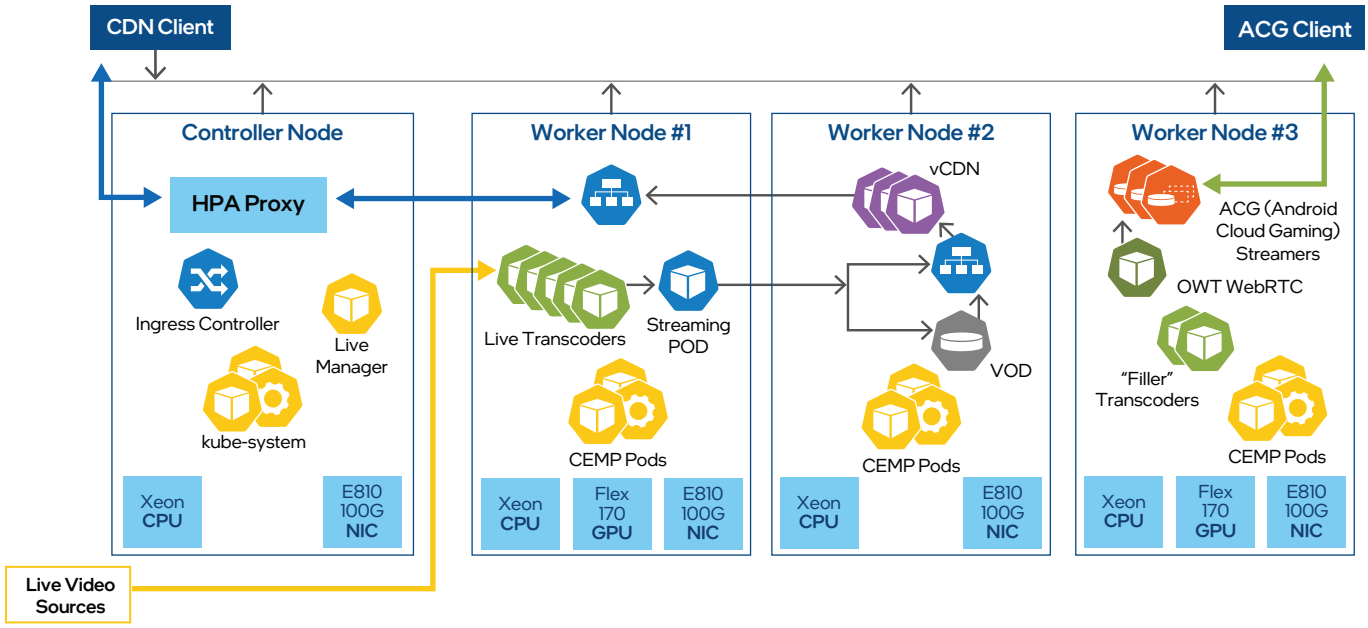

Figure 3. A 4-node CEMP cluster for media transcode, CDN, and cloud gaming.

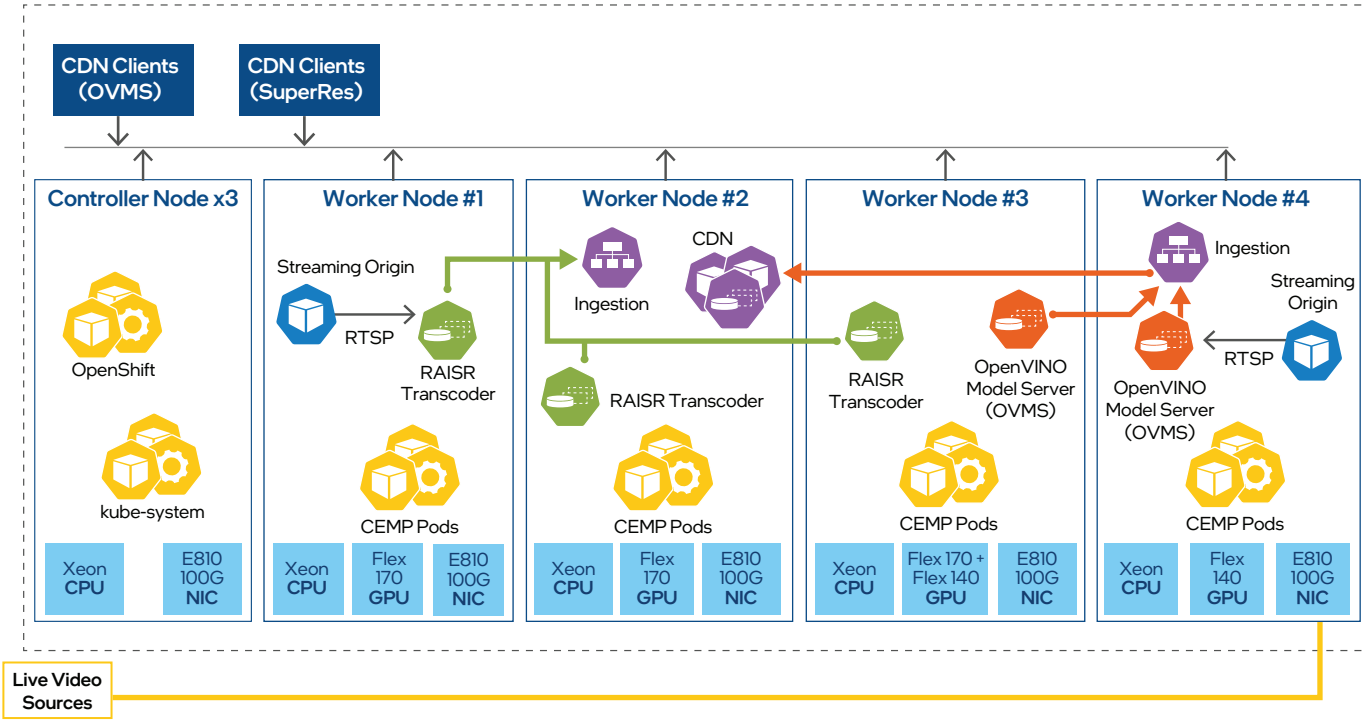

Figure 4. A 7-node CEMP cluster with AI-infused media services.

Media services at the edge

In monolithic designs, media services are built as single-purpose dedicated hardware devices in the cloud, for example, a TV set-top box or a transcoder server in an encoder farm. When making media services as microservices and putting them in daily operations on the edge, there are some unique challenges:

- Media workloads are usually comprised of multiple chained microservices and functions, such as filters, decoders, and encoders in FFmpeg. In the context of containers, the design of media workloads may take a tradeoff between cross-container (or even cross-node) communications and reusability/scalability.
- Media data has special structure and definitions — for example, audio/video compressions and file formats. To orchestrate multiple media microservices, a media-specific control plane may be needed — new APIs and protocols to connect different media types and contents, for example, MPEG-I NBMP (Network-Based Media Processing).
- In a cloud-native architecture, GRPC and REST APIs are the usual transports for data communication. However, media transports (the data plane) usually require UDP-based low-latency protocols (for example, RTP, WebRTC, etc.). Also, media streaming differs from common data streaming because timeliness is more important than completeness.
- Media workloads have vastly used hardware accelerations, i.e., new hardware devices (e.g., GPU, FPGA, etc.). Virtualization and orchestration are critical in a media platform for the TCO.
- Most media workloads are stateful, which requires workload-specific handling when scaling and scheduling (for example, preemption). This is a differentiating point for a media cloud-native platform.

Highlighted features

Orchestration of microservices

Cloud-native platforms stand out for dynamic resource allocation ("scheduling" in Kubernetes terminology), automated deployment, and life cycle management compared to bare-metal or VM infrastructures. Resource allocation for a given workload or microservice can be based on demand—for example, real-time bandwidth, workload characteristics, specific HW accelerators, etc.

It implies 1) a just-in-time model of resource usage and, therefore, more efficient TCO for both application providers and platform operators; 2) hardware resources can be allocated on-demand to avoid over-provisioning or under-provisioning as well as being energy-efficient; and 3) the resources themselves can be tailored for specific microservices.

CEMP brings the benefits of cloud-native technologies to the edge and highlights the unique capabilities of Intel hardware in terms of resource orchestration.

Intel Xeon Scalable CPU optimizations

Core frequency scaling

Intel Xeon Scalable processors have a rich feature set of advanced power management capabilities that were designed to allow users to adjust the compute performance-to-power-consumption ratio dynamically in response to the needs of the workload or the infrastructure owner.

Intel Speed Select Technology (Intel SST) is the collection of technologies giving users granular control over many aspects of the CPU power, such as base frequency (Intel SST-BF), core power (Intel SST-CP), and turbo frequency (Intel SST-TF). The controls provide the ability to further adjust all or some core frequencies beyond or below the base frequency, based on factors within the processor, including temperature, power consumption, and load.

CEMP integrated with the Intel SST to allow microservices and pods to benefit from frequency scaling on a per-core basis.

Figure 5 shows a pod with four cores executing the high-priority workloads configured to run at a constant turbo frequency (3.8 GHz) while the base frequency is 2.2 GHz.

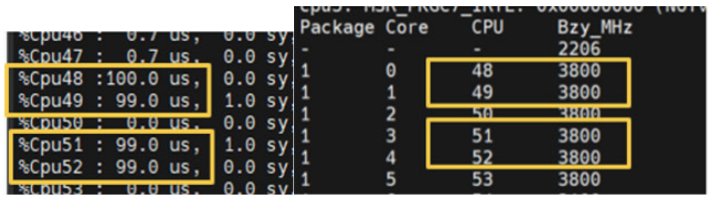

Figure 5. Core usage % and frequency with SST enabled.

For comparison, FFmpeg transcoding tasks (1x and 4x 4k streams transcoded to 4k, 1080p, and 720p streams) are measured in a pod with turbo frequency and the other pod with base frequency. Table 1 shows the performance (transcoding FPS) improvement (53% to 64%).

Table 1. Performance boost with Intel Speed Select optimization.

Test case	Test result/FPS		Performance improvement
	w/o SST	w/ SST	
1x4K–4K	4.9	7.9	**1.61x**
4x4K–4K	4.85	7.6	**1.57x**
1x4K–1080p	12	19	**1.64x**
4x4K–1080p	11	18	**1.58x**
1x4K–720p	19	31	**1.63x**
4x4K–720p	18	27.5	**1.53x**

NUMA alignment

Non-Uniform Memory Access (NUMA) is a commonly used shared memory architecture used in distributed systems with multiple processors.[2] CEMP also integrates NUMA-aware capabilities that create deep optimization without a complex restructuring of underlying container implementation, as shown in Figures 6 and 7. Node Resource Interface (NRI) plugins create second-level scheduling through the container runtime, such as CRI-O or containerd, that enables the placement of containers and pods with contiguous memory allocation on NUMA zones — removing the cross-NUMA resource access penalty for memory operations. This is a hard-sought optimization in bare metal configurations. NRI plugins provide a second advantage for workloads needing access to platform components: containers and pods may be allocated to the NUMA zone where the device is attached. For example, a container that requests a GPU device may also be annotated to request scheduling for the same NUMA zone where the claimed resource (GPU device) is installed.

The NUMA alignment will directly impact performance. Table 2 compares two pods with the same transcoding test cases. The only difference between the pods' configurations is that one pod is configured with NUMA alignment with the GPU device, and the other is not. The one with NUMA alignment almost triples the transcoding performance (measured by output FPS).

Table 2. NUMA-unaligned/aligned/improvement in FPS.

Test case	Test result/FPS		Performance improvement
	GPU Non-NUMA	GPU NUMA	
1x4K–4K	38	87	**2.29x**
4x4K–4K	8.65	23.25	**2.69x**
1x4K–1080p	35	87	**2.49x**
4x4K–1080p	7.95	22.25	**2.80x**
1x4K–720p	38	91	**2.39x**
4x4K–720p	8.575	24	**2.80x**

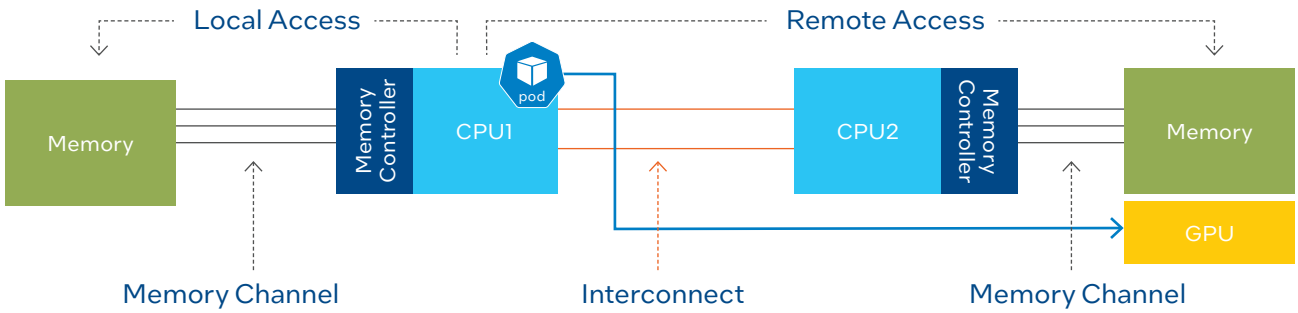

Figure 6. NUMA architecture without NUMA-aware scheduling.

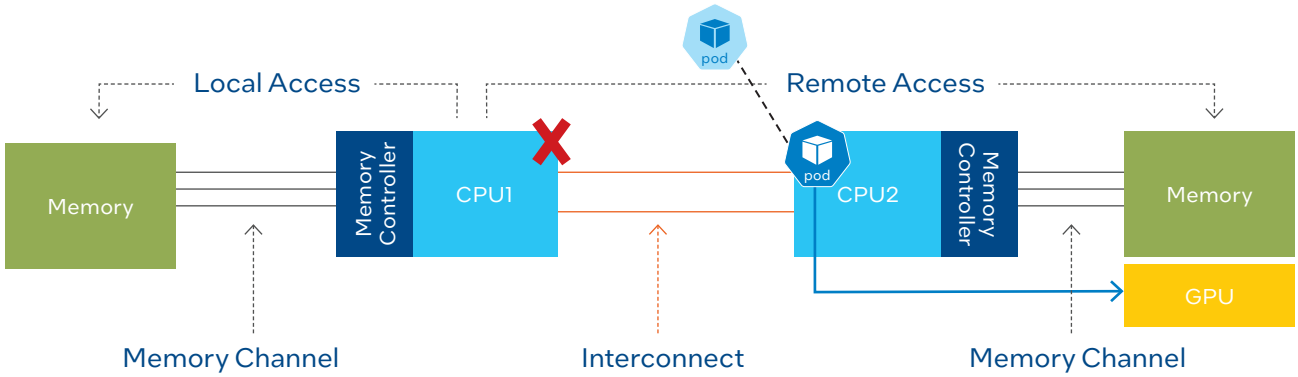

Figure 7. NUMA-aware scheduling.

Intel Data Center GPU Flex Series optimizations

Intel GPUs are supported through Kubernetes device plugins (or operators) in CEMP. Device plugins for Kubernetes provide an integrated approach to abstracting basic compute resources. The device plugins of the Intel Data Center GPU Flex Series are available for open-source Kubernetes and distributions such as Red Hat OpenShift. They work hand in hand with Node Feature Discovery (NFD) to expose optimized resources across the cluster. These components abstract resource management for Kubernetes policy application and allow for fine-granularity usage of node platform resources.

The Intel GPU devices exposed by the device plugins and managed by GAS (GPU-Aware Scheduling) support hybrid types of multiple GPU usage in a cluster and fractional usage of one GPU device. This is critical for GPU resource allocation for multi-services during Kubernetes scheduling (not supported by default K8s). One example is the ability to request portions of the total GPU cores as millicores — multiple workloads can share a limited number of GPUs. For example, media upscaling from 1080p to 4K may consume 20–25% of GPU compute cores; by requesting a fixed number of GPU millicores, the GPU resources can be used much more efficiently across clusters — a huge benefit for TCO. This concept is summarized in Figure 8.

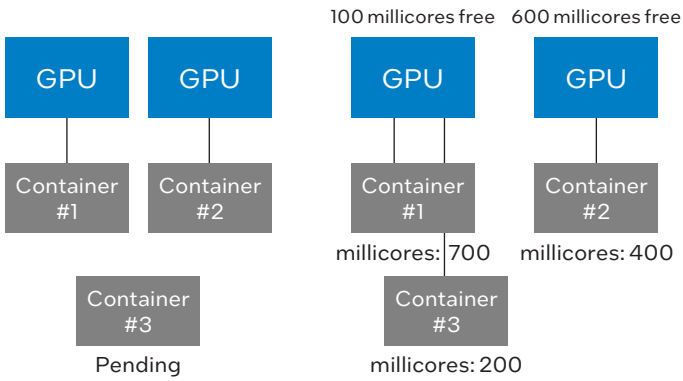

Figure 8. Scheduling GPU resources with millicores.

Platform awareness

Intel's Platform Aware Scheduling (PAS), a key value-add of CEMP, is used to leverage the platform telemetry metrics to inform the CEMP scheduler and scaler. It is comprised of a group of related projects that expose platform-specific attributes to the K8s scheduler and make this data available for scheduling and de-scheduling decisions in Kubernetes. Current extenders include Telemetry Aware Scheduling (TAS) and GPU Aware Scheduling (GAS). As shown in Figure 9, TAS helps Kubernetes and the community to improve resource optimization based on the telemetry data generated from the workload(s) resources. It is based on written policies to provide pod placement and management in a closed loop.

GAS allows the usage of GPU resources, such as memory amount, for scheduling decisions in Kubernetes. This allows users to handle fractional GPU allocation and differentiate usages of multi-type (hybrid) GPUs.

PAS works with Node Feature Discovery (NFD) to optimize the scheduling of workload pods and containers in the cluster-level scheduler. NFD can detect and advertise the hardware features of each node in the Kubernetes cluster and facilitate workload scheduling. This includes GPUs, instruction sets such as AVX and device accelerators such as Intel® QuickAssist (Intel® QAT).

CEMP incorporates Prometheus as a metrics collector to add value to media use cases to ensure that workloads have full access to cluster and node platform metrics. This allows cluster administrators to ensure resources are allocated and provisioned per real-time demands. CEMP uses open-source Grafana to visualize all metrics through a single pane of observability glass.

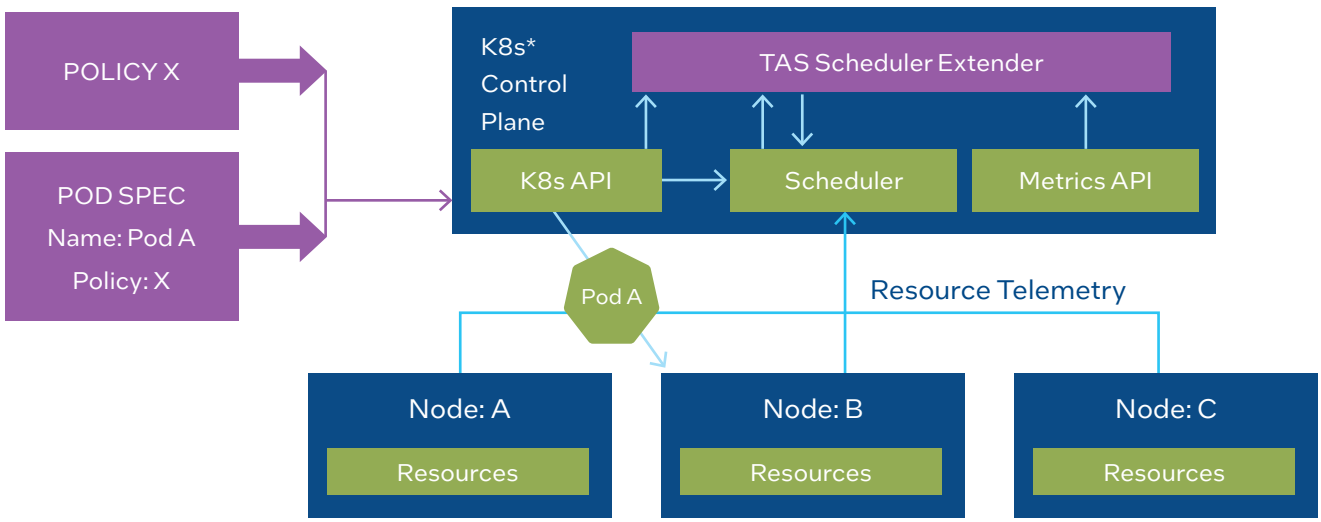

Figure 9. Telemetry Aware Scheduling for Kubernetes[3]

Intel device plugins for media accelerations

Various Intel device plugins beneficial for media/AI workloads will be selectively added to CEMP (based on customer interests). With those plugins, media microservices will exploit Intel hardware accelerations in Kubernetes platforms in the same way as bare-metal environments. Those plugins include:

- Intel® QuickAssist (Intel® QAT): Acceleration toolkits for cryptography and data compression. **More about Intel QAT**
- Intel® Dynamic Load Balancer (Intel® DLB): Achieves hardware-assisted core-to-core queue communication and subsequentially enables coordination and collaboration of multiple CPU cores on elephant flow handling and linear scalability. **More about Intel DLB**
- Intel® Data Streaming Accelerator (Intel® DSA): Accelerates workloads by offloading data movement from the processor, allowing the CPU to perform other advanced analytics-related tasks. **More about Intel DSA**

Summary

Communications Service Providers are increasingly investing in their network edge rollout to offer compelling media and AI services with improved performance and user experience. The proven benefits of cloud computing are being leveraged at the edge. The constraints and the diversity of the distributed edge require the architecture to be modular, extensible, and cost-effective such that it can support a plethora of services with a common platform. Intel's Converged Edge Media Platform (CEMP), a vertical instantiation of the Intel Tiber Edge Platform, is a cloud-native platform built for deploying and scaling media and AI workloads at the network edge. With a combination of open-source technologies and workload-driven optimizations leveraging the underlying heterogeneous Intel hardware innovations, the CEMP makes the edge deployment viable for service providers and developers. The architecture is future-proof such that it can not only support existing edge applications but also work well for emerging and yet-to-be envisioned services. The hardware-based validation in CEMP with real-world workloads provides a high degree of confidence and time-to-market acceleration. Finally, the CEMP enables a rich ecosystem of ISVs, OSVs, SIs, solution providers, and OxMs that can innovate, iterate, and adapt as needed.

[1] Intel Corporation, "What is the Network Edge?," https://www.intel.com/content/www/us/en/edge-computing/what-is-the-network-edge.html.

[2] Frank Denneman, "NUMA Deep Dive Part 1: From UMA to NUMA," https://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa/.

[3] Adrian Hoban, "Kubernetes - Resource Orchestration for 4th Gen Intel® Xeon® Scalable Processors," https://networkbuilders.intel.com/docs/networkbuilders/kubernetes-resource-orchestration-for-4th-gen-intel-xeon-scalable-processors-technology-guide-1675291057.pdf.

0524/AS/MESH/PDF 360310-001US