Position Paper: Renovating Edge Servers with ARM SoCs

Mengwei Xu, Li Zhang, Shangguang Wang
Beijing University of Posts and Telecommunications, Beijing, China

***Abstract*—Edge servers are key to the success of edge computing. Compared to cloud servers, edge servers suffer from more constrained and costly electricity supply due to their dense, near-population deployment. Towards higher energy efficiency, we propose an extreme design of edge servers – SoC-Cluster that consists of massive, inter-connected ARM SoCs. Indeed, such SoC-Clusters have already been adopted to serve the cloud gaming application in the wild. In this paper, we present a concrete implementation of a COTS SoC-Cluster and its hardware specifications. We then discuss the potential killer applications that such SoC-Cluster can well serve and the major challenges to be solved. We also dive deep into two of such applications (live video transcoding and deep learning serving) and carry out a measurement study to demystify the application performance of SoC-Cluster. The results reveal that, compared to traditional servers, SoC-Cluster not only can reduce energy consumption but even deliver higher workload throughput in certain scenarios. Finally, we conclude the paper and discuss the primary research directions that can be explored by our community from applications, software, and hardware aspects.**

***Index Terms*—Edge computing, Mobile SoC, Energy efficiency**

I. Introduction

By sinking hardware resources in proximity to end users and devices, edge computing is regarded as a promising paradigm for many killer apps like AR/VR, auto-driving, and smart cities [1]–[3]. Such a concept has been landed in markets on large scale recently [4], [5], and is explosively growing to be a critical infrastructure of our digital world. It is estimated that over $700 billion in cumulative capital expenditure will be spent within the next decade on edge IT infrastructure [6]. Meanwhile, according to Gartner, around 75% of enterprise-generated data will be processed at the edge by 2025 [7].

Like cloud data centers, an edge site consists of multiple inter-connected edge servers. An edge server can be placed in micro-datacenters per city/town, server rooms per building, base station, or anywhere else that need local computation. What does a typical edge server look like? The answer is that they are alike, if not exactly the same as, the servers adopted in cloud data centers, i.e., many-core CPUs plus domain-specific accelerators (DSAs) as shown in Figure 1(a). Given that the major edge providers nowadays are those commercially successful in cloud computing as well such as AWS, Azure, and Alibaba [8]–[10], it is unblamable to extend their well-established cloud infrastructure to edges.

However, edge is a different context from cloud, making us suspicious of whether the traditional form of cloud server can still fit to edge scenarios. Specifically, we argue that

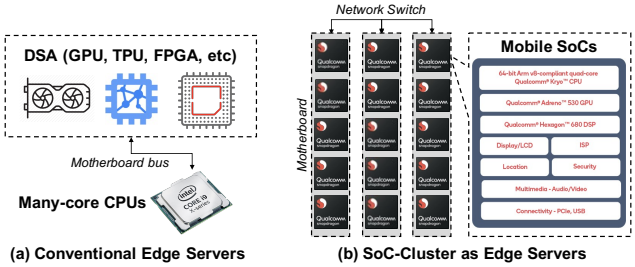


Fig. 1. A high-level comparison of conventional edge servers and the proposed SoC-Cluster in this work.

power consumption will be a crucial, limiting factor for large-scale, high-density edge server deployment in the future. First, the power supply to the edge is more constrained and costly, as the edge servers tend to be deployed closer to the population. Instead, the cloud data centers are known to be power-hungry [11]–[13], and their locations are often cherry-picked, e.g., closer to hydropower. Meanwhile, the global power footprint of edges is forecast to explode in the near future, e.g., 102 thousand megawatts in 2028 [6]. Second, the edge workloads are more variational than clouds, possibly because of the application characteristics they serve [1]. The traditional edge servers rely on increasingly powerful and monolithic processors whose energy consumption cannot proportionally scale with the workloads [14]. Furthermore, edge servers are also space-restricted and have one magnitude higher power density, putting enormous pressure on their cooling mechanism [15].

We therefore advocate for renovating edge servers to better serve the emerging edge workloads. In this position work, we explore a fundamentally new form of edge servers, namely SoC-Cluster, which consists of tens or hundreds of mobile SoCs like Qualcomm Snapdragon as shown in Figure 1(b). The underlying rationale is that mobile SoCs are designed to be energy-efficient by using a reduced instruction set and smaller transistors than traditional high-end processors. For instance, the Qualcomm Snapdragon 888 SoC, released in 2021, uses 5nm technology, whereas Intel Xeon CPU is still using $\geq$10nm. Moreover, each SoC can be independently turned on/off, or adjusted in frequency to adapt to the dynamic workloads, which is much more flexible than a monolithic powerful processor like NVIDIA GPU.

In addition to the energy efficiency advantage, we are driven by a few more rationales for exploring SoC-Cluster

at the edge. (i) Except for the general purpose computing unit, mobile SoCs incorporate heterogeneous co-processors to accelerate domain-specific workloads (e.g., GPU for image rendering, DSP for digital signal processing, and NPU for neural network inference). We will study the potential of some of these co-processors in §IV. (ii) A variety of leading chip firms are contributing their wisdom to the field of mobile SoCs, making them fast-evolving [16]. Building an edge server atop those mobile SoCs would be a free lunch. (iii) There have been a lot of efforts to optimize the software stack on mobile SoCs and OSes. A few examples include deep learning frameworks [17], [18], multimedia processing [19], and virtualization solutions [20], [21]. (iv) Mobile SoCs run mobile OSes and apps seamlessly, which allows mobile devices to offload computations and code execution to them directly.

More importantly, we observe that such SoC-Cluster servers have been already manufactured and deployed on edges. Alibaba ENS [10], which is one of the major edge service providers worldwide, has already deployed thousands of such servers in their edge sites. Currently, those SoC-Clusters are mainly serving cloud gaming services [22]–[26], which enables wimpy or low-battery smartphones to run resource-consuming games anytime and anywhere. Using SoC-Cluster, developers do not need to adapt their games to other platforms (e.g., x86) and directly deploy the server-side games. Further enhanced by the 5G technology, the users can obtain the same game experience as if the games run locally but with much less energy consumption.

But is SoC-Cluster capable to serve more general edge workloads? We give a positive answer by examining the potential killer applications on SoC-Cluster in §III. Those applications can enjoy the inherent vantages of SoC-Cluster as discussed above, yet also face unique challenges to be addressed. In §IV, we perform preliminary experiments using two applications (live video transcoding and deep learning serving) on an SoC-Cluster as case studies, with a head-to-head comparison with conventional servers. The experiment results quantitatively reveal that SoC-Cluster can significantly reduce the energy consumption in serving those edge-typical workloads and even deliver higher workloads throughput in certain scenarios. We also discuss the future work to be done to turn SoC-Cluster into general-purpose edge servers in §V.

II. Background and Related Work

Edge clouds are commonly regarded as critical infrastructure to achieve the vision of near-data processing. Major cloud providers like AWS, Microsoft, and Alibaba are expanding their clouds to edge sites [8]–[10]. Given their huge success in cloud computing, it is unblamable for them to reuse the cloud servers on edges, i.e., many-core CPUs plus domain-specific accelerators like GPU and TPU.

On cloud, energy efficiency has been recently recognized as a crucial criterion for building a data centers [11]–[13]; on edges, the energy issue will likely be aggregated as the edge servers tend to be deployed closer to the population, where the power supply is more constrained and costly than cherry-picked locations for data centers. Speaking of energy efficiency, people could think of smartphones and mobile SoCs, which are designed for low-power use cases. A straightforward idea is: can we turn tens or even hundreds of mobile SoCs into one single edge server, and use that server to handle typical edge workloads?

We are not the first trying to conceptualize a server consisting of tiny SoCs. There are attempts [27], [28] to investigate whether mobile SoCs can provide sufficient performance and reduce costs for HPC workloads. To reduce e-waste, Shahrad et al. [29] build computation nodes with used smartphones and gave an analysis of server design, but didn't evaluate the real workloads. Switzer et al. use only five smartphones to build a junkyard data center [30] with carbon concerns. Some work uses IoT/mobile SoCs to support specific applications, like video transcoding [31], key-value storage [14], and parallel computing [32].

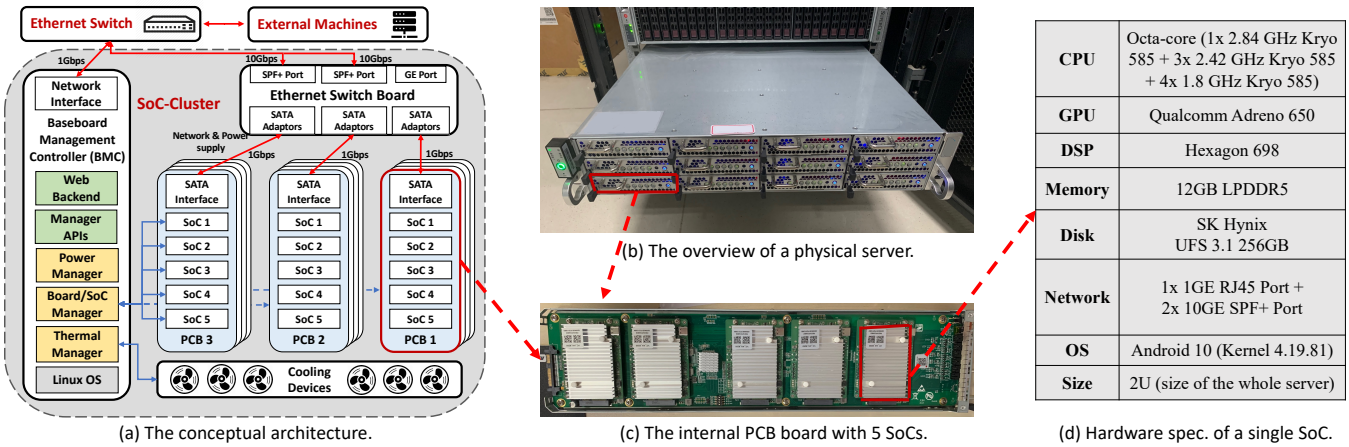
Those prior work have taken the very early but important attempts, mostly in a pure research manner, to harvest wimpy processors into a strong and general cloud server. Until recently, however, we are aware of their value in edges and the evolution has already began in the industry. The concept of SoC-Cluster has been turned into commercial products to serve an important application: cloud gaming. According to our investigation, major Edge Service Providers (ESPs) in China such as Alibaba ENS have deployed tens of thousands of SoC-Cluster servers on their edge sites to enable users to play mobile-native games anywhere with guaranteed QoE. However, according to the runtime traces collected in the wild, we conclude that those servers experience workloads with relatively low utilization ($\leq$20%) and high dynamic.

To take a closer look, we bought one representative SoC-Cluster from a leading manufacturer. Figure 2 shows its overall architecture and specification. Physically, SoC-Cluster occupies 2U space in a standard rack. SoC-Cluster mainly consists of 12 PCB boards, each of them integrating 5 mobile SoCs and providing both power supply and network capabilities. There are 60 SoCs inside SoC-Cluster in total, and we list the specification of each SoC in Figure 2(d). Each PCB board is hot-pluggable, therefore providing more flexibility at the server design phase and operation phase. The Ethernet Switch Board is responsible for connecting all pluggable PCB boards (thus SoCs), and exposing them through its network interface (i.e., SPF+ ports or GE port). In addition, SoC-Cluster contains a Baseboard Management Controller (BMC) that provides a list of programming interfaces. Operators leverage these APIs to monitor the power consumption and temperature, manage the power supply, or adjust the policy of cooling devices.

Given its representativeness, in §IV, we will perform a preliminary measurement to reveal the application performance using this particular SoC-Cluster.

III. Applications and Challenges

In this section, we discuss (potential) killer applications on SoC-Cluster and the major challenges to realize those



CPU	Octa-core (1x 2.84 GHz Kryo 585 + 3x 2.42 GHz Kryo 585 + 4x 1.8 GHz Kryo 585)
GPU	Qualcomm Adreno 650
DSP	Hexagon 698
Memory	12GB LPDDR5
Disk	SK Hynix UFS 3.1 256GB
Network	1x 1GE RJ45 Port + 2x 10GE SPF+ Port
OS	Android 10 (Kernel 4.19.81)
Size	2U (size of the whole server)

(d) Hardware spec. of a single SoC.

Fig. 2. A look at a COTS SoC-Cluster.

potentials. Theoretically, mobile SoC can support any kind of application. Yet, as a new form of servers, we need strong incentives to move an application from conventional servers to SoC-Cluster. Such incentives come from the inherent vantages of SoC-Cluster: mobile-native support; heterogeneous and low-power processors (GPU, DSP, etc.); the large number of CPU cores and total I/O bandwidth, and so on.

A. Cloud Gaming

Cloud gaming [22]–[26] is the de-facto and perhaps the only existing application that SoC-Clusters are now serving in the wild, according to our communication with a few ESPs in China. Such commercial success comes from the recent flourish of mobile games like Genshin Impact which brings billions of dollars to the game providers. Through cloud gaming, a wimpy mobile device is capable of running high-end games anywhere and anytime. For the tight latency constraint, cloud gaming services are better to be placed on edges.

SoC-Cluster seamlessly supports mobile games, naturally. Without using SoC-Cluster, the game providers need to make tremendous efforts to adapt their game to different hardware platforms, even on an ARM server that has the same ISA as mobile SoC but with different hardware specifications. Android virtualization on x86/ARM servers is emerging [33]–[35], but they are far from being mature in the aspects of generality and performance. Even if the game providers have cross-platform support, the subtle differences in UI often lead game users to choose the mobile version – after all, the game experience means everything.

The major challenge of deploying cloud gaming on SoC-Cluster is the granularity. On one hand, we observe that the Adreno GPU on high-end Qualcomm SoC like Snapdragon 865 is powerful enough to simultaneously serve multiple streams of medium-end games, e.g., around 4 for Honor of Kings. It requires a container-like isolation technique on mobile SoCs. Unfortunately, the current Android OS is not designed for multi-app parallelism, especially when it comes to UI rendering. Second, emerging resource-intensive games might overwhelm the out-of-date mobile SoCs. Shall we retire those SoC-Clusters, or we can build a framework that allows multiple SoCs to serve one game stream collectively. To this end, automatically decomposing the game logic and distributing them across SoCs will be an interesting topic.

B. Mobile-native Offloading

Beyond cloud gaming, SoC-Cluster can run any software that runs on mobile devices. It leads us to an enchanting vision that, in an oblivious manner, the "hot spots" code regions on mobile devices can be offloaded to nearby edge servers so their battery life can be significantly lengthened. The offloading can be done by the OS, therefore requesting no assistance from users or app developers.

The mobile and edge research communities have invested tremendous efforts to realize such vision in last ten years [36]–[39]. Unfortunately, we do not see it totally come true. We deem the primary reason to be the huge gap of mobile SoC and conventional servers at both software and hardware levels. Therefore, we take the emergence of SoC-Clusters in edges as an once-in-a-lifetime opportunity to realize such vision.

The critical challenge in building such a ubiquitous and oblivious offloading system is the state synchronization. Mobile devices operate under physical context, which affects the data sources of many sensors such as GPS and accelerators; they are mobile and the context changes frequently. To ensure the offloaded code obtain the correct results, state synchronization is inevitable and can incur high latency overhead.

C. Live Video Transcoding

Video transcoding, i.e., the process of converting the video format such as resolution and FPS from one to another, is the key building block to many edge workloads like live video conferencing and live streaming. A recent empirical study shows that video transcoding is the dominating use case of public edge platforms [1]. SoC-Cluster is adept at video transcoding with its low-power CPUs and hardware codec as will be demonstrated in following preliminary experiments.

The major obstacle to in-the-field deployment of video transcoding on SoC-Clusters seems to be the immature software stacks. For example, current video transcoding services

heavily rely on FFmpeg, a library that provides comprehensive video operations and configurations. There lacks a FFmpeg-like software on ARM SoC, especially for its hardware-accelerated codec. Existing toolkits like LinkedIn *LiTr* [40] are designed for single-video transcoding but not many. Furthermore, a unified scheduling framework is needed as each SoC-Cluster server could handle hundreds of video streams so the network congestion or SoC performance variation need to be carefully handled. Nevertheless, we deem video transcoding to be very likely the next deployed domain for SoC-Clusters in the near future after cloud gaming.

D. Deep Learning Serving

Deep learning serving (or prediction) at the edge is an active research field in recent years. Numerous applications such as AR, autonomous driving, and object detectors have been built atop on-edge DL serving services [2], [41]. DL serving is known to be energy-intensive, for which reason SoC-Cluster could be good fit. Moreover, SoC-Cluster is equipped with heterogeneous processors like GPU, DSP, or even NPU that can highly accelerate DL workloads. The software stack of on-mobile DL is blossoming as well [18], [42]–[44]. As we will show, SoC-Cluster delivers impressive energy efficiency and even throughput in DL serving as compared to conventional servers.

The major challenge of deploying DL serving on SoC-Cluster is the inference latency. Large DNN models like YOLOv5x or ResNet-152 take hundreds of milliseconds to process per sample, which can not meet the end-to-end QoS requirement of the aforementioned applications. It urgently calls for collaborative inference across many SoCs – an unexplored topic as far as we know. Building such a system is not easy, as suggested by our ongoing efforts, because the network latency across SoCs (typically at the sub-millisecond level) could easily overturn the benefits of SoC parallelism. A sophisticated design is needed to overlap the communication and computation to achieve scaled inference speed with more SoCs.

E. Deep Learning Training

Unlike DL serving, DL training is not a typical workload at the edge. A single DL training task could take hours or even days to accomplish even with many datacenter-level GPUs. The incentives for moving DL training to SoC-Clusters are mainly twofold. First, the edge workloads experience high temporal variation so there is a huge amount of idle time in edge servers [1]. By placing time-insensitive DL training tasks on SoC-Clusters at the correct time, we can harvest such free cycles of ARM SoCs. Second, DL training is notoriously known to be energy-hungry. Shifting DL training tasks to low-power ARM SoCs could potentially save a significant amount of carbon emissions for our environment.

There are two major challenges in landing DL training to SoC-Clusters. First, like DL serving, multiple SoCs in one SoC-Cluster or even multiple SoC-Clusters need to collaborate to train a DNN model, as constrained by the memory size and computing capacity per SoC [45], [46]. While distributed training has been common sense in data centers, there have been very few practices in orchestrating such a large amount of processors. For instance, we estimate that many hundreds of SoCs are needed to collectively provide the computing capacity as 8 high-end NVIDIA GPUs. The training scalability is difficult to achieve as the network throughput can easily become the bottleneck. Moreover, each SoC is heterogeneous with GPU and DSP, each of which can train a DNN (portion) individually; to fully unleash their power, a novel, perhaps hierarchical network topology needs to be constructed. The second challenge is to avoid compromising delay-sensitive workloads such as cloud gaming on SoC-Cluster and minimize the switching overhead. Luckily, Microsoft has made an early effort in multiplexing low-power devices with cloud gaming and DL training [47]. The checkpoint technique for DL is also well explored on data center GPUs [48] but needs to be adapted to mobile SoCs.

F. Database Systems

Data-intensive applications, such as key-value storage systems, play key roles in cloud and edge as well. They are the key building blocks of major Internet services such as Amazon Dynamo and Facebook memcached. Those applications are I/O, not computation, intensive; they require massive parallelism with huge amount of independent concurrent operations. Those applications are known to be ill-served by conventional servers – they are either slow with repeated, continuous random-access to clumsy external disks; or expensive by using large DRAM arrays.

SoC-Cluster is fast and cost-effective in serving such data-intensive applications. In the SoC-Cluster we use, each SoC is equipped with a 256GB Sk-Hynix Flash storage (UFS 3.1). Each SoC provides around 1,733/328 MB/s for sequential read/write and 23.6/34.0 MB/s for random read/write as tested by *fio* [49], similar to an enterprise Samsung SSD and an order of magnitude faster than one Seagate HDD. Collectively, such an SoC-Cluster server provides 15.36TB disk storage with more than 1 GB/s I/O random-access throughput.

The major challenge to realize the I/O advantage of SoC-Cluster is to advisably distribute the data across SoCs so the read/write operations can be concurrently handled by different SoCs without encountering congestion. Back to 2009, a key-value system named FAWN-KV [14] has been designed to deal with this challenge. To expand the benefits to more data-intensive applications, the inherent characteristics of those applications must be incorporated.

G. Streaming Processing

Billions of IoT devices are deployed in field and generating massive data streams; processing those data often requires edge servers so as not to stress the wide-area network (WAN). The code logic of streaming processing is often as simple as a pipeline of a few operations like windowing, groupby, reducer, and aggregation. Yet, optimizing the processing speed could be complicated as the data records come in at high speed

Hardware	Live video transcoding	DL Serving
SoC-Cluster	FFmpeg (with libx264 support) [51]; LiTr (with MediaCodec support) [40].	TFLite [18]
Intel CPU	FFmpeg [52] (with libx264 support) [51]	TVM [53]
NVIDIA GPU	FFmpeg (with NVDEC/ENC support) [54]	TensorRT [55]

TABLE I
SOFTWARE USED IN OUR CASE STUDIES.

and are possibly out of order. To achieve high processing throughput in realtime, a server must have a large number of cores for massive in-parallel processing and enough memory bandwidth to avoid I/O bottleneck. SoC-Cluster satisfies these requirements: the server shown in Figure 2 has 480 CPU cores in total; its 60x LPDDR5 DRAM can be accessed at the same time and thus exhibit superior I/O bandwidth collectively.

The challenges to achieve high-speed streaming processing in SoC-Cluster are mainly twofold. First, the mobile CPUs are asymmetric (ARM big.LITTLE architecture [50]) and might cause significant straggler effects. Second, the data operation dependencies could lead to high synchronization overhead. It requires a heterogeneity-aware core-level workload scheduler that judiciously dispatches data to different SoC cores.

IV. CASE STUDIES

In this section, we quantitatively investigate two applications, live video transcoding and deep learning inference, that are most likely to gain commercial success in the near future on SoC-Cluster beyond cloud gaming. Using them as case studies, we want to answer the critical question about how efficiently SoC-Cluster can serve more general edge workloads as compared to conventional servers. Specifically, we use an edge-typical server with a 40-core (80-thread) Intel Xeon Gold 5218R processor and 8 NVIDIA A40 GPUs (released in the same year as Snapdragon 865) for comparison.

Software. We list the software we used in case studies in Table I. The software are chosen for their state-of-the-art performance and outstanding popularity. For live video transcoding, we randomly pick 3 videos from the cloud video transcoding benchmark–vbench [56]. For DL serving, we use ResNet-50 [57], a medium-sized DNN model for CV tasks.

Setup. The application throughput is measured by how many frames/streams each hardware can process per second, which is explicitly reported by the software or implicitly calculated using latency. Power consumption is measured using the software-level APIs, e.g., *turbostat* for Intel CPU, *nvidia-smi* for NVIDIA GPU and *pmbus* exposed by SoC-Cluster's BMC. During the major experiments of energy efficiency, we always fully load the hardware (e.g., many processes of live video streams), to eliminate the impacts of resource under-utilization on NVIDIA GPU. We report the energy consumption by subtracting the idle power consumption of server. To mitigate the power fluctuation during experiments: for DL serving, we use 1,000 frames in each test and then get the average power consumption for each frame; for live video transcoding, we simultaneously transcode the maximum number of live video streams supported on each hardware.

Video	Hardware	Throughput (# of streams)	Energy (frames/J)	PSNR (db)
V1-desktop Bitrate: 180 Kbps	Intel CPU (4-core container)	31	23	31.08
	NVIDIA A40	37	13	34.11
	SoC CPU	15	59	31.21
	SoC Codec	16	125	29.27
V2-game3 Bitrate: 5.6 Mbps	Intel CPU (4-core container)	8	11	39.69
	NVIDIA A40	18	12	40.73
	SoC CPU	4	32	40.37
	SoC Codec	12	167	34.72
V3-chicken Bitrate: 49 Mbps	Intel CPU (4-core container)	2	2	38.71
	NVIDIA A40	6	2	42.54
	SoC CPU	1	5	38.80
	SoC Codec	2	26	38.28

TABLE II
THE LIVE VIDEO TRANSCODING PERFORMANCE OF SOC-CLUSTER AND CONVENTIONAL SERVERS. VIDEOS ARE PICKED FROM A CLOUD VIDEO TRANSCODING BENCHMARK [56].

A. Live Video Transcoding

Table II summarizes the measurement results of live video transcoding on different videos and hardware.

Throughput. A single SoC CPU can transcode 1–15 video streams simultaneously. Utilizing the hardware codec improves the throughput by up to 3× (4 to 12 on V2). Collectively, an SoC-Cluster can provide a video transcoding service of 180–1,860 streams (using all 60 SoC CPUs and 60 SoC Codecs). Such a throughput is significantly higher than the 40-core CPU server that delivers only 20–310 streams capacity. Furthermore, the throughput of SoC-Cluster equals 30–53 NVIDIA A40 GPUs. Such a number of NVIDIA GPUs usually requires around 8–27 rack units to meet the deployment needs, considering the fact that the 2U rack size occupied by the SoC-Cluster can typically hold 4–8 GPUs. During the experiments, we observed that the throughput of NVIDIA GPU is always bounded by its hardware encoder, leaving most of its general-purpose computing units under-utilized. Indeed, improving video transcoding performance is not a primary design goal of mainstream GPUs.

Energy efficiency. SoC-Cluster's advantage in energy efficiency is even more significant compared to conventional servers. SoC CPU delivers higher energy efficiency than the conventional server using Intel CPU and NVIDIA GPU. Delegating video transcoding to hardware codec in SoC-Cluster achieves significant improvement in energy efficiency. SoC-Cluster's hardware codec can transcode 26–167 frames per Joule for diverse videos, which is up to 15.18× higher than the Intel CPU and up to 13.92× higher than NVIDIA A40 GPU. Our additional experiment shows that the fine-grained computation unit (per SoC) makes the power consumption of SoC-Cluster can proportionally scale with the workloads, which is hard for conventional edge servers comprised of monolithic CPUs or GPUs.

Video quality indicates the quality of the output video perceived by consumers. Our experiments were performed with a fixed bitrate target. However, the output video quality could differ observably due to the nuance at both software

Model	Hardware	Latency (ms)	Throughput (frames/s)	Energy (samples/J)
ResNet-50 (FP32)	Intel CPU (4 cores)	12.47	80	2.6
	NVIDIA A40 (BS=1)	2.18	459	2.8
	NVIDIA A40 (BS=64)	23.45	2,729	10.2
	SoC CPU (4 big cores)	77.60	13	2.1
	SoC GPU	32.70	31	18.2
ResNet-50 (INT8)	NVIDIA A40 (BS=1)	0.45	2,202	18.6
	NVIDIA A40 (BS=64)	7.51	8,526	31.3
	SoC DSP	8.80	114	71.4

TABLE III
THE DL SERVING PERFORMANCE OF SOC-CLUSTER AND CONVENTIONAL EDGE SERVERS. DEFAULT BATCH SIZE (BS) IS 1.

encoders and hardware levels. To understand the difference, we saved the live video transcoding outputs to files and use Peak Signal-to-Noise Ratio (PSNR) to quantify the video quality. The results show that the software encoder using SoC CPUs can preserve almost the same video quality as Intel CPU and NVIDIA GPU, while videos generated by SoC-Cluster 's hardware codec have sightly poorer quality than others, i.e., 1.12%–14.76% lower PSNR. This is caused by the loose quality and bitrate requirements of mobile encoders inherently. Through our additional experiments, we find that simply loosing the target bitrate constraint using hardware codec still fails to meet the same video quality as the software codec on SoC CPUs. As such, for quality-sensitive applications, it's better to choose SoC CPUs for video transcoding.

B. Deep Learning Inference

Table III summarizes the measurement results of deep learning serving on different videos and hardware.

Throughput. With ResNet-50 (FP32) model, SoC CPU and SoC GPU can process 13 and 31 frames per second, respectively. Collectively, our SoC-Cluster delivers a maximal throughput at 2,640 FPS, which equals to 3.3× 40-core Intel CPU servers; or ∼1 NVIDIA A40 GPU when fully loaded with a relatively large batch size. Since a 2U rack can typically hold 4–8 GPUs, the throughput of SoC-Cluster is a few times smaller than a GPU server. Similarly, SoC Digital Signal Processor (DSP) is often used for integer-operation acceleration. With ResNet-50 (INT8 quantized), SoC-Cluster can provide a maximal throughput as high as 6,840 FPS, which is close to a NVIDIA A40 GPU.

Energy efficiency. Notably, SoC-Cluster's accelerators (GPU and DSP) provide observably higher energy efficiency than conventional servers. Concretely, SoC GPU can process 18.2 samples per Joule on ResNet-50 (FP32), which is 7× and 1.8× higher than Intel CPU and NVIDIA GPU, respectively. SoC DSP even shows higher energy efficiency compared with SoC GPU, i.e., 2.3× higher than NVIDIA A40 GPU when batch size is 64. NVIDIA GPUs operate more efficient with larger batch sizes. Thus, its energy efficiency drops significantly when workload is lighter, e.g., 10.2 to 2.8 frames per Joule when batch size drops from 64 to 1 on FP32 model. It shows that the monolithic design of datacenter-level GPU cannot proportionally scale its energy with workloads – a critical feature in edges as the workloads are highly variational. Instead, SoC-Cluster's each single SoC can efficiently process each sample with batch size 1 and some of them can be turned off to eliminate the energy waste without workloads.

Latency. We observe that SoC-Cluster's GPUs and DSPs deliver much lower latency than its CPUs. Especially, on a quantized ResNet-50 model, inference using the SoC DSP takes 8.8ms, which is almost eligible for most edge applications. Nevertheless, NVIDIA GPU delivers much lower latency than SoC-Cluster due to its high hardware-level parallelism and the well-optimized software (TensorRT). However, if a larger batch size is used, the latency on NVIDIA GPU increases significantly. In other words, NVIDIA GPU cannot achieve the best at both energy efficiency and inference speed; a trade-off needs to be made. To serve larger DNNs like YOLOv5x and BERT, a single SoC is not likely to achieve a satisfactory latency, and software that can orchestrate many SoCs is urgently demanded.

V. CONCLUSION AND FUTURE WORK

In this work, we explore the possibility of massively assembling mobile SoCs into an edge server. Such SoC-Cluster has advantages of high power efficiency, native support for mobile apps, and high memory/disk I/O bandwidth as compared to conventional edge servers. We discuss the potential killer applications for such a new form of hardware and the corresponding challenges to turn them into reality. We also present preliminary measurements that demonstrate the impressive performance of SoC-Cluster on two representative edge applications.

We believe the emergence of SoC-Cluster in industry will soon open a new research domain; we as the edge computing preachers are now at the perfect timing to embrace it. Here, we share the possible directions that can be explored to fully harness SoC-Cluster for the edge. (i) At application level, there is no doubt that tremendous efforts should be invested to migrate typical edge applications to SoC-Cluster. It sometimes requires a fundamental redesign of the application structure to fit the new hardware paradigm. An automatic or semi-automatic tool that facilitates such transformation could greatly catalyze this process. (ii) At system or middleware level, the infrastructure of traditional edge clouds needs to be renovated as well. It includes the mobile operating systems (e.g., Android) that are designed for interactive scenarios but not server workloads; the resource management and scheduler that often ignore the hardware heterogeneity; the virtualization techniques (e.g., Docker) that can barely run on resource-constrained SoCs; and so on. (iii) At hardware level, the SoC-Cluster we experimented with is designed for cloud gaming, but not more complicated workloads, especially those need cross-SoC collaboration. To scale the processing speed of applications like DL inference/training, SoC-Cluster must incorporate more enhanced network technique and topology.

REFERENCES

[1] M. Xu, Z. Fu, X. Ma, L. Zhang, Y. Li, F. Qian, S. Wang, K. Li, J. Yang, and X. Liu, "From cloud to edge: a first look at public edge platforms," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021, pp. 37–53.

[2] X. Zhang, A. Zhang, J. Sun, X. Zhu, Y. E. Guo, F. Qian, and Z. M. Mao, "Emp: edge-assisted multi-vehicle perception," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 545–558.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] "Aws iot for the connected home," https://aws.amazon.com/iot/solutions/connected-home/, 2020.

[5] "Remote rendering," https://azure.microsoft.com/en-us/services/remote-rendering/, 2022.

[6] "A market and ecosystem report for edge computing," https://www.lfedge.org/wp-content/uploads/2020/04/SOTE2020.pdf, 2020.

[7] "What edge computing means for infrastructure and operations leaders," https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders, 2021.

[8] "Azure edge zone," https://docs.microsoft.com/en-us/azure/networking/edge-zones-overview, 2020.

[9] "Aws local zones," https://aws.amazon.com/about-aws/global-infrastructure/localzones/, 2020.

[10] "Extending the boundaries of the cloud with edge computing," https://www.alibabacloud.com/blog/extending-the-boundaries-of-the-cloud-with-edge-computing_594214, 2020.

[11] S. Li, X. Wang, X. Zhang, V. Kontorinis, S. Kodakara, D. Lo, and P. Ranganathan, "Thunderbolt: Throughput-Optimized, Quality-of-Service-Aware power capping at scale," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. USENIX Association, Nov. 2020, pp. 1241–1255. [Online]. Available: https://www.usenix.org/conference/osdi20/presentation/li-shaohong

[12] S. Govindan, D. Wang, L. Chen, A. Sivasubramaniam, and B. Urgaonkar, "Towards realizing a low cost and highly available datacenter power infrastructure," in *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*, ser. HotPower '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/2039252.2039259

[13] L. Liu, C. Li, H. Sun, Y. Hu, J. Gu, T. Li, J. Xin, and N. Zheng, "Heb: Deploying and managing hybrid energy buffers for improving datacenter efficiency and economy," in *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, ser. ISCA '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 463–475. [Online]. Available: https://doi.org/10.1145/2749469.2750384

[14] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "Fawn: A fast array of wimpy nodes," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, 2009, pp. 1–14.

[15] Q. Pei, S. Chen, Q. Zhang, X. Zhu, F. Liu, Z. Jia, Y. Wang, and Y. Yuan, "Cooledge: Hotspot-relievable warm water cooling for energy-efficient edge datacenters," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 814–829. [Online]. Available: https://doi.org/10.1145/3503222.3507713

[16] "Qualcomm showcases future technology roadmap to drive the connected intelligent edge and lead the world to 5g advanced and beyond," https://www.qualcomm.com/news/releases/2022/02/28/qualcomm-showcases-future-technology-roadmap-drive-connected-intelligent, 2022.

[17] X. Jiang, H. Wang, Y. Chen, Z. Wu, L. Wang, B. Zou, Y. Yang, Z. Cui, Y. Cai, T. Yu *et al.*, "Mnn: A universal and efficient inference engine," *arXiv preprint arXiv:2002.12418*, 2020.

[18] "Tensorflow Lite," https://www.tensorflow.org/lite, 2022.

[19] "Mediacodec," https://developer.android.com/reference/android/media/MediaCodec, 2022.

[20] W. Song, J. Ming, L. Jiang, Y. Xiang, X. Pan, J. Fu, and G. Peng, "Towards transparent and stealthy android os sandboxing via customizable container-based virtualization," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2858–2874. [Online]. Available: https://doi.org/10.1145/3460120.3484544

[21] W. Chen, L. Xu, G. Li, and Y. Xiang, "A lightweight virtualization solution for android devices," *IEEE Transactions on Computers*, vol. 64, no. 10, pp. 2741–2751, 2015.

[22] "Amazon Luna," https://www.amazon.com/luna/landing-page, 2022.

[23] "Geforce Now," https://www.nvidia.com/en-us/geforce-now/, 2022.

[24] "Google Stadia," https://stadia.google.com/, 2022.

[25] "X-cloud Game Pass," https://www.xbox.com/en-US/xbox-game-pass/cloud-gaming?xr=shellnav, 2022.

[26] "Cloud Gaming, Meet Facebook Gaming," https://www.facebook.com/fbgaminghome/blog/cloud-gaming-meetfacebook-gaming, 2022.

[27] N. Rajovic, P. M. Carpenter, I. Gelado, N. Puzovic, A. Ramirez, and M. Valero, "Supercomputing with commodity cpus: Are mobile socs ready for hpc?" in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2013, pp. 1–12.

[28] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramirez, "Tibidabo: Making the case for an arm-based hpc system," *Future Generation Computer Systems*, vol. 36, pp. 322–334, 2014.

[29] M. Shahrad and D. Wentzlaff, "Towards deploying decommissioned mobile devices as cheap Energy-Efficient compute nodes," in *9th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 17)*, 2017.

[30] J. Switzer, R. Kastner, and P. Pannuto, "Architecture of a junkyard datacenter," *arXiv preprint arXiv:2110.06870*, 2021.

[31] P. Liu, J. Yoon, L. Johnson, and S. Banerjee, "Greening the video transcoding service with Low-Cost hardware transcoders," in *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016, pp. 407–419.

[32] F. Büsching, S. Schildt, and L. Wolf, "Droidcluster: Towards smartphone cluster computing–the streets are paved with potential computer clusters," in *2012 32nd International Conference on Distributed Computing Systems Workshops*. IEEE, 2012, pp. 114–117.

[33] K. Barr, P. Bungale, S. Deasy, V. Gyuris, P. Hung, C. Newell, H. Tuch, and B. Zoppis, "The vmware mobile virtualization platform: is that a hypervisor in your pocket?" *ACM SIGOPS Operating Systems Review*, vol. 44, no. 4, pp. 124–135, 2010.

[34] C. Dall and J. Nieh, "Kvm/arm: the design and implementation of the linux arm hypervisor," *Acm Sigplan Notices*, vol. 49, no. 4, pp. 333–348, 2014.

[35] J. Shuja, A. Gani, K. Bilal, A. U. R. Khan, S. A. Madani, S. U. Khan, and A. Y. Zomaya, "A survey of mobile device virtualization: Taxonomy and state of the art," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, pp. 1–36, 2016.

[36] I. Zhang, A. Szekeres, D. Van Aken, I. Ackerman, S. D. Gribble, A. Krishnamurthy, and H. M. Levy, "Customizable and extensible deployment for Mobile/Cloud applications," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, 2014, pp. 97–112.

[37] C. Xie, X. Li, Y. Hu, H. Peng, M. Taylor, and S. L. Song, "Q-vr: system-level design for future mobile collaborative virtual reality," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021, pp. 587–599.

[38] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.

[39] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen, "COMET: Code offload by migrating execution transparently," in *10th USENIX symposium on operating systems design and implementation (OSDI 12)*, 2012, pp. 93–106.

[40] "linkedin/litr: Lightweight hardware accelerated video/audio transcoder for Android." https://github.com/linkedin/LiTr, 2022.

[41] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th annual international conference on mobile computing and networking*, 2019, pp. 1–16.

[42] Q. Zhang, X. Li, X. Che, X. Ma, A. Zhou, M. Xu, S. Wang, Y. Ma, and X. Liu, "A comprehensive benchmark of deep learning libraries on mobile devices," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3298–3307.

[43] M. Xu, J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu, "A first look at deep learning apps on smartphones," in *The World Wide Web Conference*, 2019, pp. 2125–2136.
[44] M. Xu, M. Zhu, Y. Liu, F. X. Lin, and X. Liu, "Deepcache: Principled cache for mobile deep vision," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 129–144.
[45] D. Xu, M. Xu, Q. Wang, S. Wang, Y. Ma, K. Huang, G. Huang, X. Jin, and X. Liu, "Mandheling: Mixed-precision on-device dnn training with dsp offloading," *arXiv preprint arXiv:2206.07509*, 2022.
[46] Q. Wang, M. Xu, C. Jin, X. Dong, J. Yuan, X. Jin, G. Huang, Y. Liu, and X. Liu, "Melon: Breaking the memory wall for resource-efficient on-device machine learning," 2022.
[47] "PilotFish: Harvesting free cycles of cloud gaming with deep learning training," in *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. Carlsbad, CA: USENIX Association, Jul. 2022. [Online]. Available: https://www.usenix.org/conference/atc22/presentation/zhang-wei
[48] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *CoRR*, vol. abs/1604.06174, 2016. [Online]. Available: http://arxiv.org/abs/1604.06174
[49] "axboe/fio: Flexible i/o tester," https://github.com/axboe/fio, 2022.
[50] "big.little - Arm," https://www.arm.com/technologies/big-little, 2022.
[51] "H.264 video encoding guide," https://trac.ffmpeg.org/wiki/Encode/H.264, 2022.
[52] "Ffmpeg," https://ffmpeg.org/, 2022.
[53] T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, H. Shen, M. Cowan, L. Wang, Y. Hu, L. Ceze, C. Guestrin, and A. Krishnamurthy, "TVM: An automated End-to-End optimizing compiler for deep learning," in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. Carlsbad, CA: USENIX Association, Oct. 2018, pp. 578–594. [Online]. Available: https://www.usenix.org/conference/osdi18/presentation/chen
[54] "Using ffmpeg with nvidia gpu hardware acceleration," https://docs.nvidia.com/video-technologies/video-codec-sdk/ffmpeg-with-nvidia-gpu/, 2022.
[55] "Nvidia TensorRT," https://developer.nvidia.com/tensorrt, 2022.
[56] A. Lottarini, A. Ramírez, J. Coburn, M. A. Kim, P. Ranganathan, D. Stodolsky, and M. Wachsler, "vbench: Benchmarking video transcoding in the cloud," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2018, Williamsburg, VA, USA, March 24-28, 2018*, X. Shen, J. Tuck, R. Bianchini, and V. Sarkar, Eds. ACM, 2018, pp. 797–809. [Online]. Available: https://doi.org/10.1145/3173162.3173207
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.