Warehouse-Scale Video Acceleration

Parthasarathy Ranganathan and Daniel Stodolsky, *Google Inc., Mountain View, CA, 94043, USA*
Jeff Calow and Jeremy Dorfman, *YouTube, San Mateo, CA, 94043, USA*
Marisabel Guevara, Clinton Wills Smullen IV, and Aki Kuusela, *Google Inc., Mountain View, CA, 94043, USA*

Video processing is foundational to a spectrum of important workloads: video sharing, video conferencing, photos/video archival, virtual/augmented reality, cloud gaming, and live streaming. The exponential growth in these workloads, coincident with a slowing Moore's law, poses significant challenges to deliver more computing at higher efficiencies. In this article, we present the design of a new accelerator—the* video coding unit*—targeted at warehouse-scale (cloud) video transcoding. Our deployment at scale in Google, the first of its kind, has demonstrated significant benefits on several products (YouTube, Meet, Stadia, Photos, etc.) serving billions of users.

Video processing plays a pivotal role in several key workloads in large cloud datacenters. Video sharing and video streaming workloads (e.g., YouTube, Netflix, Facebook, etc.) are well known as contributing to the dominant portion of Internet traffic.[1] Video processing also underpins important applications in education, business, collaboration, entertainment, etc. The recent pandemic has only accelerated and amplified the adoption of video processing even further: beyond the surge in adoption of video-conferencing workloads (e.g., Meet, Zoom, Teams) for social connection and education, medical practitioners in the front line of the pandemic relied on video platforms for life-saving procedures. Trends toward live-streaming and higher video quality/resolution (e.g., 4K/8K videos) and more immersive video (e.g., 360° views) further increase the computational demand for video processing. Several important emerging workloads—virtual/augmented reality, cloud gaming, cameras in IoT devices—are also very video centric, further increasing the future importance of video processing. While the computational demand for video processing is exploding, improvements from Moore's law are stalling with traditional approaches.[2] Future growth in this important area is not sustainable without adopting domain-specific hardware accelerators.

In response to these challenges, at Google, we have designed and deployed the first warehouse-scale video acceleration system, serving multiple production video-centric workloads with stringent quality, throughput, latency, and cost requirements. This article provides an overview of our three main contributions.

> *FUTURE GROWTH IN THIS IMPORTANT AREA IS NOT SUSTAINABLE WITHOUT ADOPTING DOMAIN-SPECIFIC HARDWARE ACCELERATORS.*

First, we present key insights on how warehouse-scale video processing is fundamentally different from consumer use cases. Specifically, we identify unique challenges and opportunities relating to workload and data diversity, and higher quality requirements as well as additional constraints around throughput, availability at scale, and co-design with warehouse-scale software stacks.

Second, leveraging these insights we present the design of a new video acceleration system co-designed across hardware and software. We introduce *the video coding unit (VCU)*, a new domain-specific accelerator that implements the computationally expensive parts of video processing, while at the same time carefully architecting the right abstractions and partitioning to work in large, distributed systems, and with warehouse-scale scheduling software. We also discuss the design of our scheduler and the use of high-level synthesis (HLS) for agile hardware development.

Finally, we present detailed data and insights from our experience deploying the system at Google over several years. Specifically, we present both benchmark and

Digital Object Identifier 10.1109/MM.2022.3163244
Date of publication 31 March 2022; date of current version 30 June 2022.

fleet-wide data that shows the effectiveness of our domain-specific acceleration: more than an order-of-magnitude performance per total cost of ownership, over prior well-tuned software implementations, all while maintaining stringent datacenter quality and availability requirements. But perhaps equally important, we also discuss how our accelerator system enables new capabilities that were otherwise not possible across a range of workloads, from video sharing, to photos and video archival, to live streaming, and cloud gaming.

A more detailed discussion of our design is available in our original article in ASPLOS 2021.[3]

WAREHOUSE-SCALE VIDEO PROCESSING CHALLENGES

Introducing video accelerators at warehouse scale[4] is a challenging endeavor.

Plethora of Output Files and Formats

Unlike the early days of classic TV broadcasting where all devices supported a single video format and a single resolution, today, we watch video on a wide variety of devices (desktops, phones, TVs, etc.). Given this large range of screen sizes/resolutions, most video platforms convert each uploaded video into a standard group of 16:9 resolutions. These different file sizes can also be used to adapt to a viewer's network bandwidth availability to deliver the best supportable resolution. Similarly, a number of video compression formats need to be supported (e.g., H.264, VP9, AV1, etc.). This production of multiple outputs per input is a key difference between a video sharing service and a consumer video application, such as video chat.

Algorithmic Complexity of Video Transcoding

The conversion of the original format to multiple resolutions and other formats is called *transcoding:* each video is decoded from its original format into raw frames, scaled to the new resolution and then potentially encoded into a different format with higher compression settings. Transcoding algorithms are computationally intense and include various knobs and tradeoffs (one-pass versus two-pass algorithms, low-latency versus offline modes, chunking and parallel multiple-output transcoding, multidimensional search, etc.). Video sharing platforms must optimize these tradeoffs to ensure that users receive playable and high-quality video bit streams while minimizing their own computational and network costs and operating at scale. As one illustration of scale, more than 500 h of video is uploaded to YouTube *every minute* with a similar volume being uploaded to Google Photos and Google Drive!

Workload Diversity and Usage Patterns

The diversity of workload and usage patterns adds additional challenges, as different video products have varying needs and latency/storage/playback tradeoffs. For example, we can spend spend more effort compressing YouTube uploads than YouTube Live streams due to more relaxed latency requirements. Photos and Drive videos are watched less than YouTube, so storage concerns have more weight. Similarly, upload volume can vary dramatically (for example, YouTube versus Google Drive). Finally, video popularity follows a power law: the head videos are very popular and worth spending a lot of resources to compress more; the middle bucket is somewhat popular and worth spending a moderate amount of resources; and the long tail needs to minimize resource usage while maintaining broadcast quality.

Warehouse-Scale "Datacenter as a Computer"

At the warehouse scale, where many thousands of devices will be deployed, there is an increased focus on cost efficiency that translates into a focus on throughput and scale-out computing.[4] The "time to market" also becomes critical, as launching optimized products faster can deliver significant cost savings at scale. In addition, unlike consumer environments where individual component reliability and a complete feature set are priorities, in a warehouse-scale context the constraints are different: fallback software layers can provide infrequently needed features, and reliability can be augmented by redundant deployments. Also, testing and deploying updates can be highly disruptive in large-scale data centers and, consequently, systems need to be optimized for change management.

VIDEO ACCELERATION DESIGNED GROUND-UP FOR WAREHOUSE-SCALE COMPUTING

Summarizing the discussion earlier, transcoding is the most important component of datacenter video platforms but poses unique challenges for hardware acceleration. These include being able to handle and scale to a number of different output resolutions and formats, as well as handling complex algorithmic tradeoffs and quality/compression/computing compromises. These challenges are compounded by attributes of warehouse-scale system design: inter and intratask parallelism, high performance at low costs, ease of deployment when operating at scale,

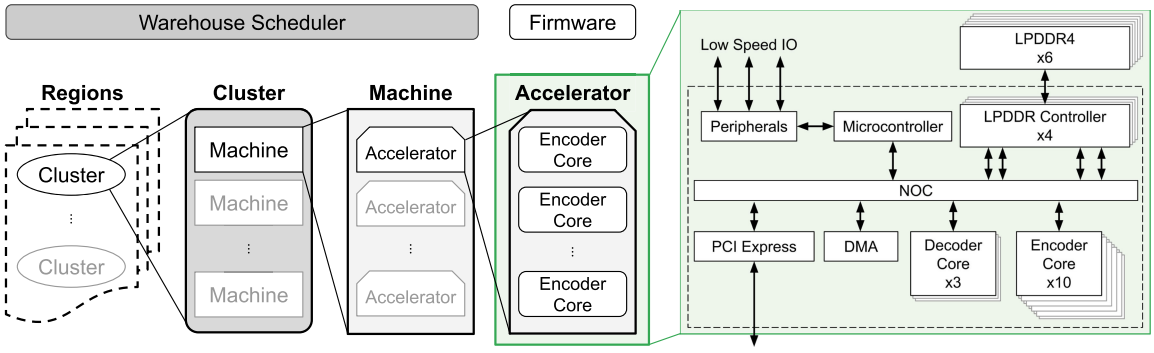

FIGURE 1. Overview of VCU system architecture at all scales.

coordinated scheduling and failure tolerance. Taken together, cloud video workloads on warehouse-scale computers are very different from their consumer counterparts, presenting new infrastructure challenges around throughput, quality, efficiency, workload diversity, reliability, and agility.

WE OPTIMIZE SYSTEM BALANCE AND GLOBAL WORK SCHEDULING TO MINIMIZE STRANDING, SPECIFICALLY PAYING ATTENTION TO THE GRANULARITY AND FUNGIBILITY OF WORK.

In response to these challenges, we designed a new holistic system for video acceleration, built ground-up for datacenter-scale video workloads, with a new hardware accelerator building block, a VCU, co-designed to work in large distributed clusters with warehouse-scale schedulers. Core to our solution is *hardware–software co-design*, to architect the system to scalably partition and optimize *functionality at individual levels*—from individual hardware blocks to boards, nodes, and geographically distributed clusters, and across hardware, firmware, and distributed systems software—with appropriate *abstractions and interfaces between layers*.

We follow a few key high-level design principles in optimizing for the distinct characteristics and constraints of a datacenter deployment:

- *Globally maximize utilization:* Given power and die-area constraints are more relaxed, our datacenter application-specific integrated circuits (ASICs) are optimized for throughput and density, and multi-ASIC deployments amortize overheads. In addition, we optimize system balance and global work scheduling to minimize stranding (underutilized resources), specifically paying attention to the granularity and fungibility of work.
- *Optimize for deployment at scale:* Software deployments have varying degrees of disruption in datacenters: kernel and firmware updates require machine unavailability, in contrast to userspace deployments which only require, at most, worker unavailability. We, therefore, design our accelerators for userspace software control. Also, as discussed earlier, individual component reliability can be simplified at the warehouse level: hardware failures are addressed through redundancy and fallback at higher level software layers.
- *Design for agility and adaptability:* In addition to existing workload diversity, we have to plan for churn as applications and use cases evolve over time. We, therefore, design programmability and interoperability in hardware, ossifying only the computationally expensive, infrequently changing aspects of the system. Software support is leveraged for dynamic tuning ("launch-and-iterate") as well as to adapt to changing constraints. An emphasis on agility also motivates our use of HLS to take a software-like approach to hardware design.

Figure 1 shows our overall system design.

VCU ASIC

At the ASIC level, efficient multiple-output transcoding requires many encoder and decoder cores all in a single chip. We chose four channels of LPDDR4 with sideband

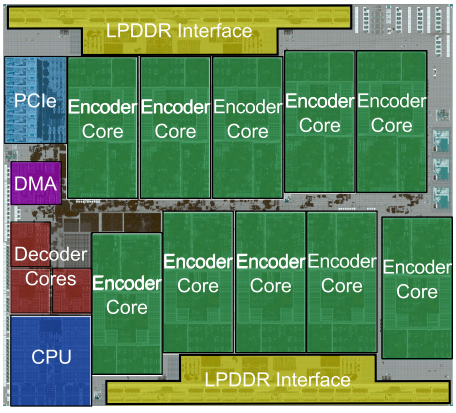

FIGURE 2. VCU chip floorplan.

error correction to provide the required bandwidth, and mostly use error correction throughout the rest of the chip to enable reliable operation. The floorplan (see Figure 2) shows that most of our area goes to the encoder cores, with the memory subsystem being the second largest. Not all transcodes can saturate a chip by themselves, so we use firmware-managed queues to allow concurrent transcodes.

VCU Board and Rack

Each system has 10 boards with two VCUs each, shown in Figure 3. We maximized the VCU density per system to improve performance per TCO dollar while avoiding bottlenecks in the host system and network. A caveat is on CPU decoding, which improves utilization in decode-limited scenarios, but can itself cause PCI Express bottlenecks. VCU hardware errors do not usually impact the system, as each ASIC is operated by a separate job.

Cluster and Warehouse-Scale Scheduling

At the cluster level, we offload nontranscode processing to the non-VCU machines, and VCUs are deployed globally to enable processing close to where videos are uploaded.

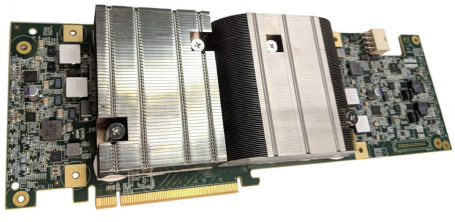

FIGURE 3. Two VCU chips on a board.

Our video processing system maintains a number of logical worker pools with VCU or CPU transcoding resources. As transcoding requests enter the system, work is scheduled onto a pool based on use case and priority. Each of these pools has its own scheduler, which assigns work items to workers based on an estimation of the resource requirements for a work item and based on the availability of resources in the pool. These resource dimensions are flexible: sometimes they map to physical resources, such as decode cores, encode cores, and memory, other times they map to some hard-to-attribute metric like PCI Express bandwidth. Because the fixed function hardware cores are nonfungible, we moved from a uniform CPU cost scheduler to an *online multidimensional bin packing scheduler* [shown in Sec. 3.3.3 of the work by Ranganathan *et al.*[3]] to ensure no single VCU becomes completely saturated and no encoder cores become starved.

The scheduler[5] is implemented as a sharded in-memory table of workers to resources; workers periodically update the scheduler with their available resources, and the scheduler sends work items to workers based on that availability. The scheduler seeks to maximize load on workers, because as workers become idle they can be released from a logical pool for use by other pools. In the event of a failure during transcoding, work is rescheduled within a logical pool, either onto another VCU or CPU.

Firmware Scheduling

Video coding on the VCU keeps persistent stream state in per-application VCU memory, and the codec cores are activated to perform some command and update the state on completion. Our codec cores typically have a per-frame granularity. For example, an operation could be "encode this frame to some VP9 bitstream." The firmware is responsible for multiplexing processes onto a VCU, and each application has a command queue and a response queue in memory shared with the VCU. Applications write commands into their command queues and the firmware then schedules between queues and activates codec cores with the commands. It is only at the application level where codec-specific programming happens; the firmware is oblivious to the contents of the commands. This programming changes often, and so it does not make sense to ossify it into silicon or even to run it in the firmware. This loose coupling allows for our typical datacenter cadence of multiple software rollouts per week, and has enabled us to improve our encoding efficiency, starting presilicon and continuing through the lifetime of the VCU.

High-Level Synthesis (HLS)

We used HLS for the encoder's design, as our experience is that HLS is much more efficient to design with. Our tooling consists of Siemens' *Catapult* and an in-house HLS framework called *Taffel*. Catapult is used for C-to-RTL synthesis for individual leaf blocks while Taffel enables modular hierarchical hardware design by gluing HLS leaf blocks together both presynthesis, as a C-model, and postsynthesis, as an RTL design. It also provides automatic tracing of data and control channels across the design and the generation of SystemVerilog unit test benches for leaf blocks and subtrees. Taffel automates low-level design specification, integration, and verification work typically done by hand, and provides productivity features, such as automatic design visualization, documentation and design search, buffer insertion, and the definition and distribution of memory mapped registers.

HW/SW CO-DESIGN HAS BEEN KEY TO ENABLING POSTDEPLOYMENT IMPROVEMENTS THAT HAVE REAL IMPACT TO THE END USER EXPERIENCE AND USABILITY OF THE VCU IN A WAREHOUSE ENVIRONMENT.

Our use of HLS allowed for novel use of the LLVM Sanitizers, especially AddressSanitizer and MemorySanitizer, to uncover issues where the hardware would have misbehaved or had some undefined behavior. Because our C-model was available throughout the project's development, we were able to use our video quality corpus of many thousands of videos along with many thousands of servers to identify and fix a number of perceptual quality issues presilicon. Perhaps most importantly, HLS's agility made evaluating and implementing encoding tools much faster compared to a conventional HDL design. Notably, we were able to add a few very late features that would have otherwise been prohibitive. HLS has been critical to reconciling the tension between YouTube's changing needs and the relatively long lead time of ASIC design.

WAREHOUSE-SCALE VIDEO ACCELERATION IN ACTION

Our video acceleration system has been deployed at Google for many years and has had impact on several high-profile products (YouTube, Meet, Stadia, Photos, etc.). In the following, we present some results and insights from both microbenchmarks as well as longitudinal studies across tens of thousands of production servers over time.

Performance

To compare the efficiency of VCU to prior platforms, we use the metric Performance per Total Cost of Ownership, (perf/TCO), where TCO includes capital and operational costs associated with each platform. Table 1 compares throughput and perf/TCO for central processing unit (CPU) baseline, graphics processing unit (GPU), and two VCU system configurations, with perf/TCO normalized to the CPU system. For offline two-pass encoding, the VCU systems provide 8×–20× higher encoding throughput than software encoding, for H.264 and VP9, respectively. Our accelerator system has an order of magnitude performance-per-cost improvement over our prior well-tuned baseline system with state-of-the-art CPUs while still meeting strict quality requirements.

Our performance data highlights the value of hardware acceleration for new generations of video coding standards, where the compute load has greatly increased to provide higher compression efficiency. VCU has given us a break-through in compute capacity at scale, which has enabled several new capabilities across multiple products. One example is multiple-output transcoding (MOT), which was previously prohibitive, especially for VP9, due to the high latency of running many workers at scale. Figure 4 shows measured throughput per MOT and single-output transcoding (SOT) production worker. The lack of variability seen in the MOT trendline also illustrates that we are able to utilize the encoder cores close to maximum capacity.

Quality

The evaluation of a video encoder must also consider the encoding quality, especially to ensure we have broadcast-quality video for various Google products.

TABLE 1. Offline two-pass single output transcode (SOT) throughput in VCU versus CPU and GPU systems.

System	Throughput [Mpix/s]		Perf/TCO	
	H.264	VP9	H.264	VP9
Skylake	714	154	1.0×	1.0×
4×Nvidia T4	2,484	–	1.5×	–
8×VCU	5,973	6122	4.4×	20.8×
20×VCU	14,932	15,306	7.0×	33.3×

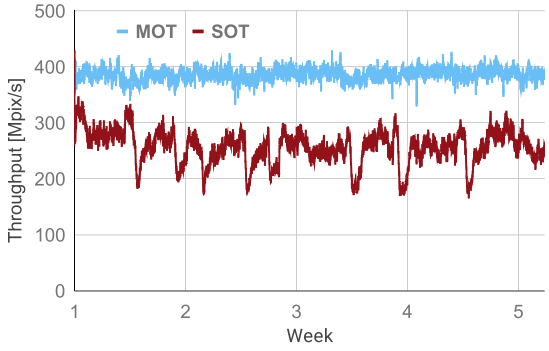

FIGURE 4. Production throughput per VCU.

Figure 5 shows how the encoding quality of VCU is comparable to software encoders.

We use RD curves (see pp. 26–28 in the work of Ortega and Ramchandran[6]) to visualize the perceptual quality, measured in peak signal to noise ratio, and compression efficiency of software and VCU. RD-curves are effective visualizations of the nature of lossy video encoding; encoders may represent a video using more or fewer bits (shown on the horizontal axis) to achieve higher or lower perceptual quality (measured by PSNR on the vertical axis). The points on the curves are formed by encoding the video at different bitrate targets.

We compare hardware-accelerated transcoding (right-hand side) with our previously well-tuned software baselines (left-hand side) for all the videos in the *vbench* suite. (Vbench is a public set of YouTube videos that demonstrates the diversity and encoding complexity of the content that is uploaded to video sharing services.[7]) Across all the videos, VCU achieves similar perceptual quality at similar bitrates as the the software baseline.

Adaptive and Resilient Through Co-Design

HW/SW co-design has been key to enabling postdeployment improvements that have real impact to the end user experience and usability of the VCU in a warehouse environment. Figure 6 shows how we improved the encoding quality of VCU, with bitrate on the vertical axis steadily dropping over time, by tuning the rate control algorithms that are guiding the encoder. The software baselines are also constantly improving, hence, a first order design goal for VCU has been this ability to parametrize the encoder so that even a year after it has been deployed, we have a better version of VCU that is just as good as the software.

HW/SW co-design also gave us the right knobs to tune the workload via scheduling. Figure 7(a) shows total VCU throughput increasing over time, as we migrated software transcoding to the VCU workers, and as we rolled out performance improvements. Another example can be seen in Figure 7(b), where the utilization of the HW decoder was at 98%, in fact stranding valuable HW encoder capacity. Leveraging our multidimensional bin packing mechanism for load

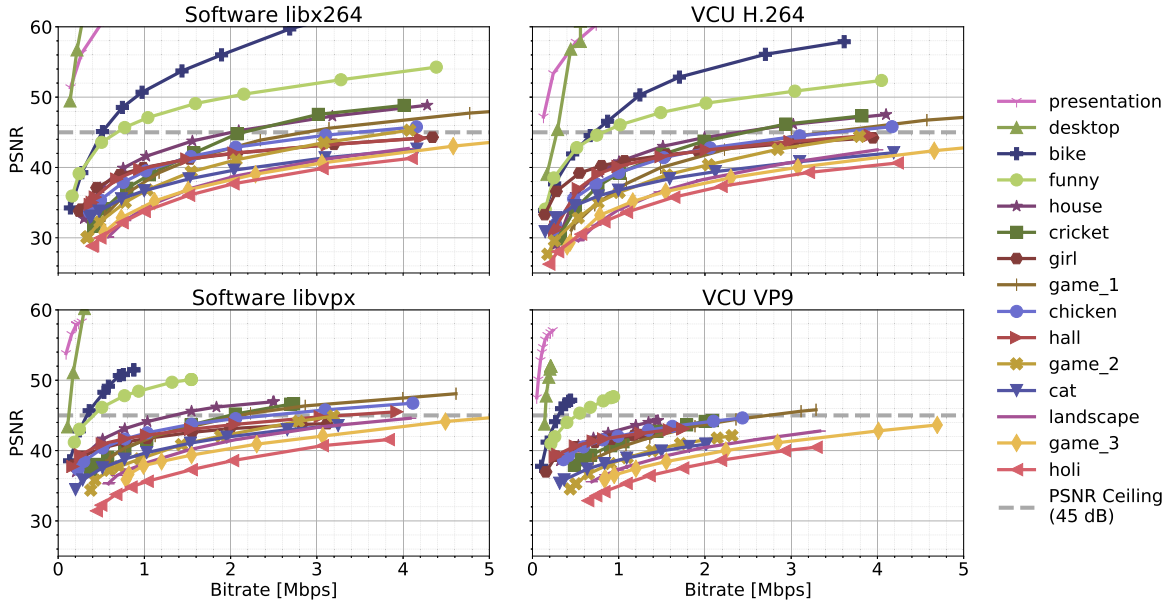

FIGURE 5. Rate-distortion (RD) curves comparing VCU encodings of the vbench video suite, to software libx264 and libvpx.

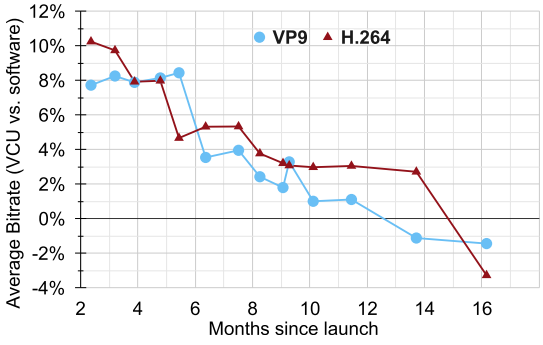

FIGURE 6. Hardware bitrate improvement over time relative to software (data points weighted by per-format egress).

balancing, we shifted some of the decode load to the host CPU, bringing down the HW decoder utilization to 91% and further boosting encode throughput.

Finally, HW/SW co-design led to a more resilient VCU system, where we have the ability to disable individual devices rather than a full system in the presence of failures. This is further discussed in detail in our ASPLOS'21 paper.[3]

New Capabilities Enabled by Acceleration

Successful accelerators are not just about cost reductions but *fundamentally enable new capabilities that were not previously possible*. Our accelerator, beyond improving efficiency, enabled new live/video-on-demand workloads, and increased bandwidth and storage compression, unlocking new workloads and capabilities that were otherwise not possible.

Two specific examples that were enabled by our VCU systems that were infeasible at scale (too expensive or too complex) with our legacy software infrastructure are 1) enabling higher compression video coding standards like VP9 and increasing their coverage for more videos, and 2) new use cases for much smaller end-to-end latencies (and reduced eyeball-to-camera delays) for low-latency use cases, such as internet broadcasting and cloud gaming.

LOOKING AHEAD

New Paradigm of Warehouse-Scale Video Acceleration

Video processing is an important and fast-growing foundational workload in warehouse-scale datacenters and clouds. However, server-side video processing brings significant challenges around dealing with the workload complexity (transcoding algorithm tradeoffs, quality/throughput requirements) and datacenter-scale (co-design with distributed processing at scale and with high churn). Our paper is *the first work* to address these challenges at scale, presenting the design and deployment of cloud video transcoding in a large production fleet supporting multiple video-centric workloads (video sharing, photos/video archival, live streaming, cloud gaming) with stringent quality, throughput, latency, and cost requirements. Our work has been deployed at Google for multiple years and has had impact on several high profile products (YouTube, Meet, Stadia, Photos, etc.).

But, beyond Google, we also believe that our work will have *bigger impact on the broader industry*. The order-of-magnitude improvement in video transcoding from the VCU accelerator can save the industry billions of dollars, but also enable new capabilities and innovations in video applications (low-latency cloud gaming, immersive high-quality video, etc.). Similar to how the Google TPU work led to a large number of startups and other products from key industry players, we hope the VCU work also leads to an explosion of

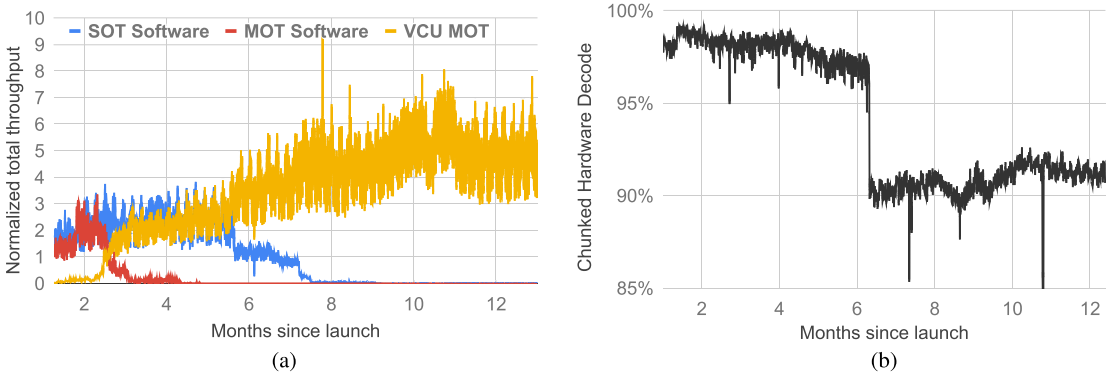

FIGURE 7. Postlaunch accelerator workload scaling. (a) Multiple-output transcoding on the VCU. (b) Opportunistic software decoding to reduce hardware decode contention.

interest and innovation in video processing accelerators for the cloud.[8]

Opportunities for Follow-on Research

Several aspects of our design are novel and first-of-a-kind. We introduce the *VCU* with several novel design features for datacenter video transcoding. Similarly, our *hardware/software co-design methodology* has several unique design features for fungibility, work scheduling, and rapid design and deployment (including algorithmic optimizations, such as in the online multidimensional bin packing scheduler). In addition, our design featured a greater *emphasis on HLS* than traditional product designs: this approach reduced code size by 5× compared to raw RTL and helped to dramatically reduce bugs before computationally expensive RTL simulation and to add new features late in the design cycle.

HW/SW CO-DESIGN LED TO A MORE RESILIENT VCU SYSTEM, WHERE WE HAVE THE ABILITY TO DISABLE INDIVIDUAL DEVICES RATHER THAN A FULL SYSTEM IN THE PRESENCE OF FAILURES.

But we believe that we have only touched the *tip of the iceberg in this new area*. There are several system design tradeoffs and opportunities that merit increased analysis: e.g., host computing design, accelerator disaggregation and sharing, new specifications, such as AV1, compiler-assisted software sanitizers applied to HLS C-simulation, etc. Similarly, rich opportunities for future innovation lie in combining transcoding with machine learning on video (for example, to automatically generate captions or enable video search) or, more broadly, offloading more of the video processing currently applied between decoding and encoding. Our hope is that our work (and our supporting data and insights) provide the starting point for more rich explorations in this area.

Insight Into Co-Design and Distributed Systems.

Our work is at the intersection of hardware (ASIC development and system architecture), distributed systems (datacenter scheduling and video processing platform), and video processing research (transcoding algorithms). We believe that our multidisciplinary approach provides *a template and reusable insights for other accelerator designs*. This is particularly valuable to the community in the era of custom silicon research, post the slowing of Moore's law. In addition, as discussed previously, the key thesis of our work is around co-design across the warehouse-scale stack, and many of our results are excellent showcases of the synergy between architecture and systems (e.g., system balance, failure management, responsiveness to workload diversity, and churn). We believe that such a focus on the *system* is again an important area, currently often underemphasized in traditional chip-centric architecture innovation. We hope the insights in this article on co-designing accelerators with large-scale distributed systems at datacenter-scale spurs subsequent accelerator projects to also take a pragmatic full-system *datacenter-as-an-accelerator* view of design.

REFERENCES

1. C. Cullen, "Sandvine internet phenomena report Q3 2019," Waterloo, ON, Canada, Sandvine, Tech. Rep., 2019. [Online]. Available: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/Internet%20Phenomena/Internet%20Phenomena%20Report%20Q32019%2020190910.pdf
2. J. Hennessy and D. Patterson, "A new golden age for computer architecture: Domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development," in *Proc. ACM/IEEE 45th Annu. Int. Symp. Comput. Archit.*, 2018, pp. 27–29, doi: 10.1109/ISCA.2018.00011.
3. P. Ranganathan *et al.*, "Warehouse-scale video acceleration: Co-design and deployment in the wild," in *Proc. 26th ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2021, pp. 600–615, doi: 10.1145/3445814.3446723.
4. L. A. Barroso, U. Hölzle, and P. Ranganathan, *The Datacenter as a Computer*, 3rd ed. San Rafael, California, USA: Morgan & Claypool, Oct. 2018.
5. A. Verma, L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proc. Eur. Conf. Comput. Syst.*, 2015, pp. 1–17, doi: 10.1145/2741948.2741964.
6. A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998, doi: 10.1109/79.733495.
7. A. Lottarini *et al.*, "vbench: Benchmarking video transcoding in the cloud," in *Proc. 23rd Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2018, pp. 797–809, doi: 10.1145/3173162.3173207.
8. A. Jani, "Google builds video-transcoder chip," *Microprocessor Rep.*, Oct. 2021. "[Online]. Available: https://www.linleygroup.com/newsletters/newsletterdetail.php?num=6373

PARTHASARATHY RANGANATHAN is currently a VP, technical fellow with Google, Mountain View, CA, 94043, USA, where he is the area technical lead for hardware and datacenters. He is the coauthor of the popular *Datacenter as a Computer* textbook. His research includes widely used innovations in energy-aware user interfaces, heterogeneous multicores, power-efficient servers, custom silicon accelerators, and disaggregated and data-centric datacenters. Ranganathan received a Ph.D. degree from Rice University, Houston, TX, USA. He is a Fellow of IEEE and ACM. Contact him at partha.ranganathan@google.com.

DANIEL STODOLSKY is an engineering VP with Google, Compute Infrastructure, Mountain View, CA, 94043, USA. Stodolsky received an M.S. degree in mathematics from Carnegie Mellon University, Pittsburgh, PA, USA. Contact him at dstodolsky@google.com.

JEFF CALOW is a principal engineer with YouTube. Calow received an B.Sc. degree in computer science and physics from Carleton University, Ottawa, ON, Canada. Contact him at jcalow@google.com.

JEREMY DORFMAN is a staff software engineer with YouTube, San Mateo, CA, 94043, USA. Dorfman received a B.S. degree in computer engineering from Northeastern University, Boston, MA, USA. Contact him at jdorfman@google.com.

MARISABEL GUEVARA is a staff software engineer with the SysInfra Performance team Google, Mountain View, CA, 94043, USA. Guevara received a Ph.D. degree in computer science from Duke University, Durham, NC, USA. Contact her at marisabel@google.com.

CLINTON WILLS SMULLEN IV is a senior staff software engineer with Google, Mountain View, CA, 94043, USA, which he joined in 2011 to work on novel datacenter systems and has been the system architect for a number of accelerator systems. Smullen received a Ph.D. degree in computer science from the University of Virginia, Charlottesville, VA, USA. Contact him at clintsmullen@google.com.

AKI KUUSELA is an engineering manager with the Devices and Services unit, Google, Mountain View, CA, 94043, USA. He has worked on video compression and various hardware accelerators for more than 20 years. Kuusela received an M.S. degree in electrical engineering from the University of Oulu, Oulu, Finland. Contact him at akikuusela@google.com.