Twenty Five Years of Warehouse-Scale Computing

Parthasarathy Ranganathan and Urs Hölzle, *Google, Mountain View, CA, 94043, USA*

When Google was founded in 1998, it was already clear that successful web search would require enormous amounts of computing power and storage, and that no single computer would be able to handle this task. Consequently, its infrastructure design marked a fundamental shift toward an approach now widely embraced as warehouse-scale computing (WSC). Starting as a niche approach optimized for web search, over the past two and a half decades, WSC has evolved dramatically. Today, it's the mainstream approach underpinning all hyperscale companies and cloud platforms and is poised to be the foundation for the next wave of artificial intelligence/machine learning computing in the cloud. In this article, we chronicle the evolution of WSC, highlighting pivotal milestones, lessons learned, and the vast opportunities that lie ahead.

Warehouse-scale computing (WSC),[5] as the name indicates, considers the delivery of computing infrastructure holistically, designing an entire *warehouse-scale data center as one computer* (see Figure 1). Warehouse-scale computers (WSCs) encompass multiple hardware infrastructure components (compute and storage servers, networking and distributed system design, and data center power and cooling) as well as multiple software components (e.g., data center scheduling, management, and hyperscale/cloud workloads). They optimize for multiple metrics: improved total cost of ownership (TCO), power efficiency, sustainability, reliability, and manageability at scale.

THE FIRST EPOCH (1999–2003): INTRODUCING SCALE-OUT COMPUTING AND SCRAPPY SPECIALIZATION

The first five years of infrastructure design at Google were about learning to build and deploy large scale-out distributed computing optimized for web indexing and web serving. The core problems to solve were reliability (especially for storage), scalability to larger search indexes and higher query volumes, and reducing cost per query.

Infrastructure design during this period consisted of cheap and scrappy optimizations. There was no time to "do it right," leading to a multitude of pragmatic shortcuts; for example, the initial BigFile storage system used mirrored disks to guard against failures but required manual work to define a replacement disk and replicate the surviving copy onto the new disk. Similarly, "babysitter" job scheduling scripts, written in Python, required manually placing each task on a particular machine and updating the script when a machine failed. Toward the end of this first period, these ad hoc systems were replaced by more robust designs, including the Google File System (GFS),[15] an early version of Borg,[34] and Google Front End, the reverse proxy in front of all Google services.

> ONE EARLY SERVER DESIGN EVEN USED CORK FOR INSULATION, WHICH LED TO THEM BEING CALLED CORKBOARD SERVERS.

All hardware consisted of minimalist low-cost servers built from commodity components normally seen in personal computers.[4] One early server design even used cork for insulation, which led to them being called *corkboard servers* (see Figure 2). Such choices led to

Digital Object Identifier 10.1109/MM.2024.3409469
Date of publication 6 June 2024; date of current version 14 October 2024.

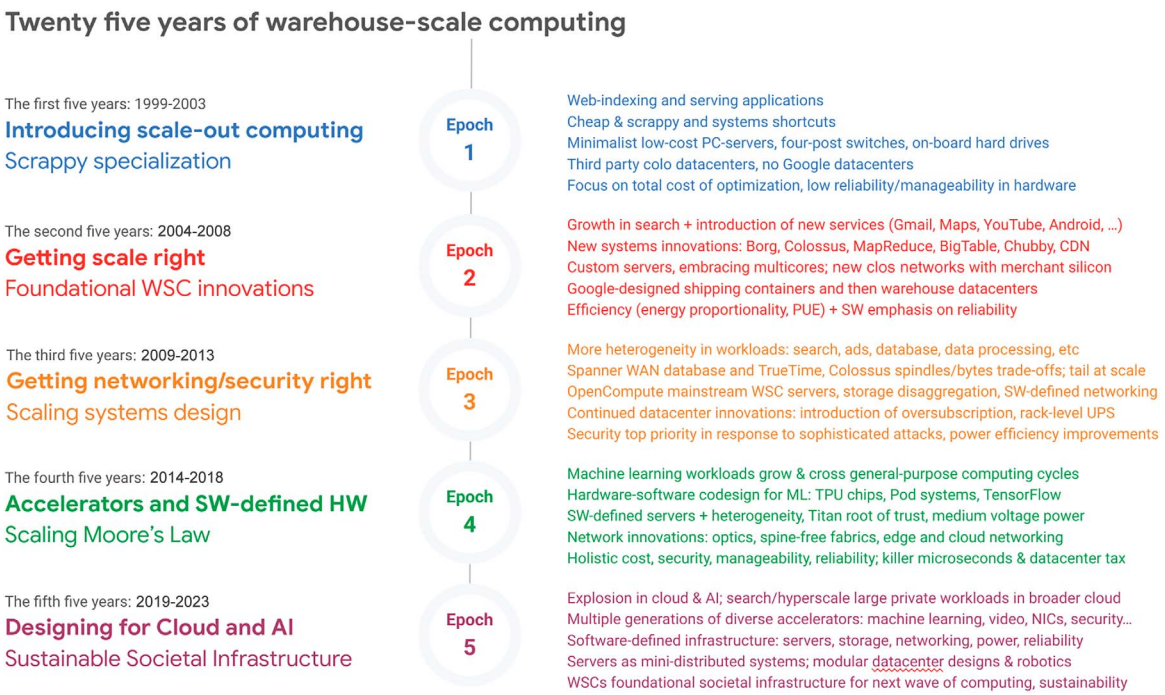


FIGURE 1. Twenty five years of WSC at Google. The infographic captures the key evolution of warehouse-scale computers in five-year epochs, summarizing key developments in applications, systems, data centers, platforms, and optimizations. CDN: content distribution network; HW: hardware; SW: software; colo: co-located; PUE: power usage efficiency; TPU: tensor processing unit; NICs: network interface cards.

unreliable hardware, so an important contribution during this time was the design of web search to account for and tolerate hardware failures. Although hardware becomes more reliable over time, daily failures are inevitable at the scale of tens of thousands of servers, and this emphasis on hardware co-designed with fault-tolerant software continues to be a core principle of WSC systems, even today.

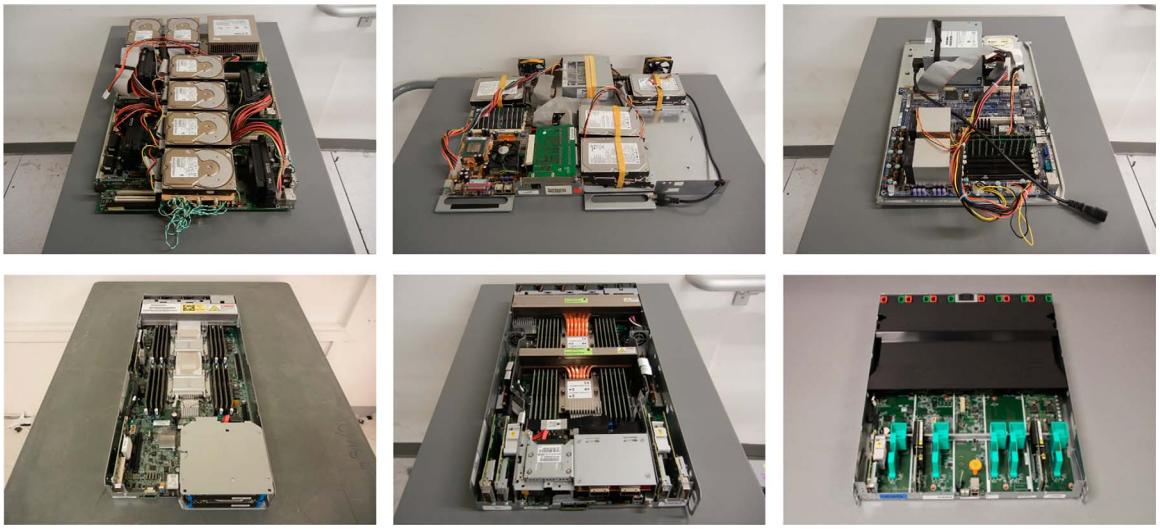


FIGURE 2. Multiple generations of servers. Six different server designs across 25 years of WSC design at Google. The top row shows earlier generations (note the corkboard and Velcro in some of the boards) and the bottom row shows more recent generations.

FIGURE 3. Servers housed in warehouse-scale computers: an evolution. (a) An early Google search server racks hosted at a colo facility. (b) A container-based data center in 2005. (c) Current data center buildings that host warehouse-scale infrastructure.

CPUs were mainly optimized for speed and cost. Storage consisted of third-party mainstream hard disk drives hosted on each server. "Scale-up" cluster networking was based on four-post 500-port switches with 1-Gbps links and high oversubscription, allowing for 100 Mbps of off-rack bandwidth per server. This design required rack-level data placement for large cluster-wide applications like web search, again emphasizing the co-design of infrastructure and applications. There were no Google data centers yet; all the infrastructure was hosted in third-party co-location (colo) data centers (see Figure 3).

Cost optimization was paramount, with a focus on "TCO." Reliability and manageability in hardware was underemphasized (or in some cases, ignored) but software was written to work around failures. Colo data centers (Figure 6) charged by space not power, so initially, space efficiency was more important than power efficiency. Toward the end of this epoch, the costs of inefficient power supplies and onboard dc–dc conversions became clear, leading to the first 12-V-only motherboard and highly efficient power supplies.

THE SECOND EPOCH (2004–2008): GETTING SCALE RIGHT AND FOUNDATIONAL WSC INNOVATIONS

The second five years of Google were all about growth in search (launching more categories, and becoming bigger, faster, and cheaper) and introduction of new services (Gmail, Maps, YouTube, Android, Chrome). Increased engineering resources allowed more ambitious solutions. The first production version of Borg[34] doubled cluster utilization with containers. Colossus overcame the scaling limitations of GFS. MapReduce[13] and Bigtable[8] simplified data processing, and many other fundamental services emerged, including the Chubby lock service,[7] a content distribution network for YouTube, and high-efficiency Google-built data centers. In many ways, this period was about getting foundational infrastructure right.

Infrastructure transitioned to more customized designs. Google started using shipping containers to build modular collections of servers, storage, networking, power, and cooling, and then transitioned to applying the same approach at the warehouse and data center scale. Data center efficiency greatly improved with hot-air capture, warm air cooling, and rack-level uninterruptible power systems, which are standard practices to this day. Two key concepts were developed during this period: 1) *energy proportionality*[2]—the importance of reducing power consumption when servers were not fully utilized, and 2) *power usage efficiency*—tracking and optimizing the efficiency of power delivery and cooling in the data center.

Servers transitioned to more custom-built designs, adapting to the transition of CPUs from the clock-speed era to the multicore era, and moving away from some of the more exotic ideas that didn't work well (no more corkboard or Velcro!). Disks continued to be directly attached to servers, but the introduction of perpendicular media recordings allowed for much higher density.

The very end of this epoch saw the introduction of CLOS data center networks based on merchant silicon, which greatly improved the off-rack bandwidth available per server. Large applications continued to share clusters, and global caches helped serve traffic closer to users.

Efficiency continued to be paramount, extending beyond traditional server costs to also include power/cooling efficiency. Workload tiers (different classes of service) and Linux containers allowed multiple workloads to share the same server. Hardware reliability improved, but at scale, failures were still assumed to be the default and not exceptional events. Dealing with increased scale continued to be an important concern.

THE THIRD EPOCH (2009–2013): GETTING NETWORKING AND SECURITY RIGHT AND SCALING SYSTEMS DESIGN

The next (third) five years of Google were characterized by even more scale and greater workload heterogeneity, with corresponding innovations in continuing to scale systems design. Three particularly notable highlights of this epoch were innovations in databases, networking, and security.

In terms of WSC workloads, search and ads still continued to be significant, but database and data processing workloads started coming into their own as well. Google Drive was introduced in 2012, and App Engine came out of preview in 2011. A small research group called *Google Brain* started looking at deep learning. To aid research on clusters and scheduling, Google released a full-month cluster job trace,[28] followed by an update in 2019.[33]

Spanner[9] departed from the NoSQL, eventual consistency tradition, and used distributed time stamping (TrueTime) to provide high performance with strong consistency (Figure 5). Colossus introduced erasure coding for better storage efficiency and allowed separate provisioning of bytes and spindles (seeks) for improved cost amortization.

Server designs mainly tracked the roadmap of the CPU suppliers. WSC server principles started receiving mainstream acceptance: many enterprise companies created "hyperscale" divisions, and OpenCompute (www.opencompute.org) was founded in 2011 to bring WSC computing beyond hyperscalers.

In-house field-programmable gate array-controlled NAND flash provided a new storage layer, initially for search and later for fleetwide workloads. The increased cross-section bandwidth of CLOS networks allowed storage disaggregation, moving disks from servers to separate disk appliances designed for better performance, maintenance, and more efficient storage provisioning.

Petabit-scale cross-section bandwidths for data center networks allowed almost all applications to ignore rack locality and allowed even solid-state drive (SSD) storage to be disaggregated from servers for better utilization and provisioning. Other continued networking innovations underpinned disaggregation and systems advances: notably, 1) cost-effective enterprise-level merchant silicon networking at the campus/wide area network (WAN) levels,[30] 2) software-defined networking (SDN)[17] in the WAN with centralized traffic engineering, and 3) end-to-end bandwidth enforcement across multi-priority traffic. Latency variability in building responsive large-scale web services forced a focus on "the tail at scale."[10]

Security became a big priority in this period brought to focus by events like sophisticated attacks from overseas actors and revelations from the Snowden leaks. System architectures responded with designs that assumed "zero trust" and the adoption of pervasive encryption of data at rest and in transit. WSC power oversubscription increased available data center space by right-sizing deployments to actual low server/cluster utilizations.

THE FOURTH EPOCH (2014–2018): ACCELERATORS AND SOFTWARE-DEFINED HARDWARE AND SCALING MOORE'S LAW

The next five-year (fourth) epoch responded to the dramatic slowing of performance-per-cost scaling associated with Moore's law with silicon accelerators, more heterogeneous hardware, and software-defined servers.

Making an inspired bet, Google pivoted WSC innovation to custom silicon for artificial intelligence (AI) and machine learning (ML) workloads. The first such accelerator, tensor processing units (TPUs) (see Figure 4),[18] reached production in 2015, demonstrating an order-of-magnitude better performance and power efficiency relative to CPUs and GPUs. Consistent with core WSC tenets, the solution included numerous innovations in hardware–software co-design (e.g., TensorFlow) and distributed systems design (e.g., TPU "pod" supercomputers). ML computing grew exponentially, soon catching up with the prior huge computing base for non-ML computing; in late 2017, ML cycles surpassed non-ML cycles in the fleet.

MAKING AN INSPIRED BET, GOOGLE PIVOTED WSC INNOVATION TO CUSTOM SILICON FOR ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING WORKLOADS.

To scale Moore's law, and marking a break from the homogenous server configurations in WSCs until then, systems were optimized for different workloads across heterogeneous CPUs, GPUs, and TPUs. Software-defined servers[12] abstracted hardware complexity, and the compiler/scheduler/management software matched workloads to the computing best suited for it. This also marked a shift in Google's approach to custom designs: in-house specialization focused more on strategic

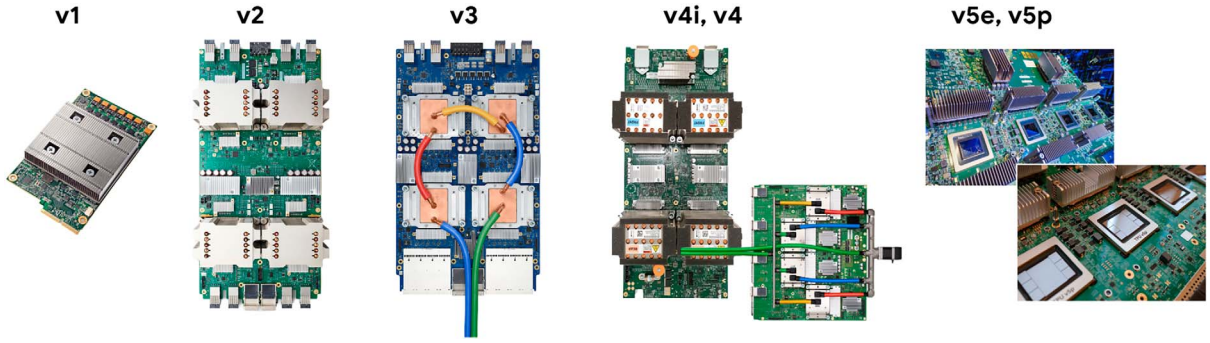


FIGURE 4. Multiple generations of tensor processing units (TPUs). Different generations of Google tensor processing units. (a) v1; (b) v2; (c) v3; (d) v4i, v4; and (e) v5e, v5p.

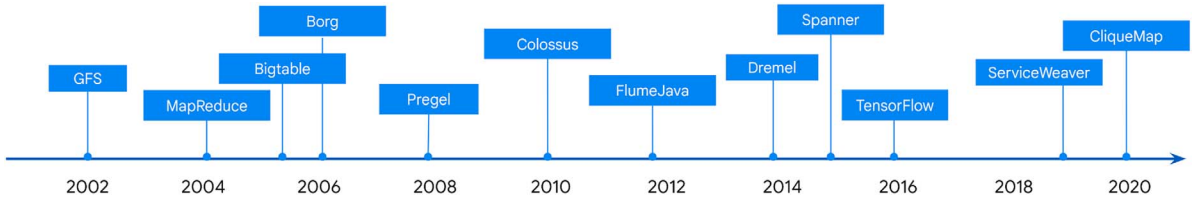


FIGURE 5. Distributed systems for compute, storage, and networking. The figure shows an approximate timeline of a representative (not complete) set of key distributed systems and frameworks introduced into Google WSCs.

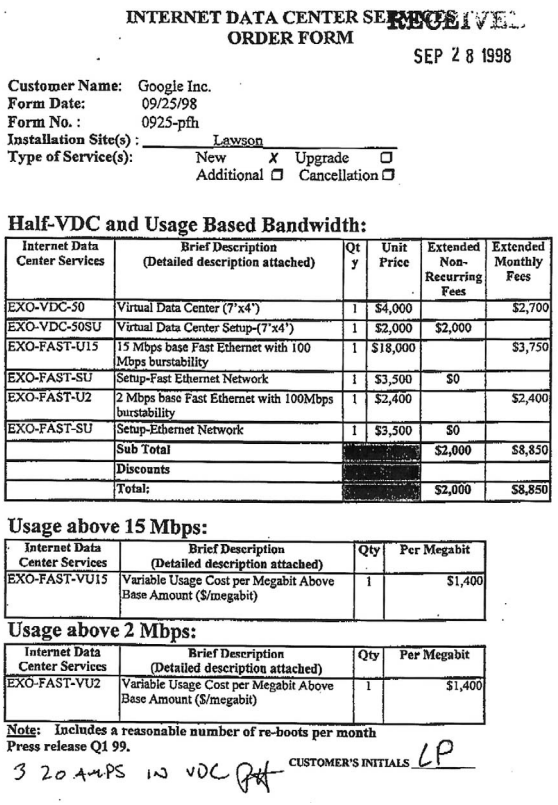
emerging workloads while embracing more industry standardization for mature WSC hardware. Borg evolved considerably during this period to accommodate the greater diversity of both workloads and servers while driving up average utilization significantly.[33] 2014 marked the first release of *Kubernetes*, an open source container manager inspired by Borg.

Server designs continued to follow industry trends: more core counts in CPUs, open 48-V designs, helium-filled disks for higher densities and lower power, industry microcontrollers for SSDs, and 3D-stacked NAND flash. Networking innovations continued: new network interface cards (NICs) at higher speeds, optical circuit switching,[22] "spine-free" cluster fabrics,[25] SDN management plane optimizations for improved reliability and feature velocity, smarter edge optimizations (Espresso),[35] and cloud virtual networking (Andromeda).[14] A big innovation in data center infrastructure was the successful deployment of medium voltage power delivery[32]; combined with aggressive tiered-workload management, this enabled much higher power oversubscription. Additionally, liquid cooling enabled the much higher power densities of TPU clusters.

The holistic focus on performance, cost, power, security, manageability, and reliability continued. Bare-metal communication, including operating-system-bypass techniques in user space ("Snap"),[24] allowed low-latency high-input-output-operations-per-second storage, and new latency-mitigating techniques addressed the "killer microsecond" era.[6] Key "data center tax" functions[19] responsible for WSC data management and distributed systems processing were identified, leading to a wave of optimizations. Power management introduced dynamic frequency and sleep state control, and environmental sustainability started becoming a more important consideration. Security improvements emphasized machine integrity and confidentiality and integrity of sensitive data. Google introduced its Titan root-of-trust security chip in 2017, dramatically unveiling it from the tiny stud earring of the keynote speaker to demonstrate its small footprint!

THE FIFTH EPOCH (2019–2023): DESIGNS FOR CLOUD AND AI, AND BUILDING SUSTAINABLE SOCIETAL INFRASTRUCTURE

The most recent five years saw a remarkable explosion of innovation around AI/ML and public cloud computing. Search and traditional hyperscale workloads continue to grow, but now represent only large private workloads within a larger public cloud computing infrastructure. AI/ML workloads continued to increase



INTERNET DATA CENTER SERVICES
ORDER FORM

RECEIVED
SEP 2 8 1998

Customer Name: Google Inc.
Form Date: 09/25/98
Form No. : 0925-pfh
Installation Site(s) : Lawson
Type of Service(s): New X Upgrade ☐
Additional ☐ Cancellation ☐

Half-VDC and Usage Based Bandwidth:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Unit Price	Extended Non-Recurring Fees	Extended Monthly Fees
EXO-VDC-50	Virtual Data Center (7'x4')	1	$4,000		$2,700
EXO-VDC-50SU	Virtual Data Center Setup-(7'x4')	1	$2,000	$2,000	
EXO-FAST-U15	15 Mbps base Fast Ethernet with 100 Mbps burstability	1	$18,000		$3,750
EXO-FAST-SU	Setup-Fast Ethernet Network	1	$3,500	$0	
EXO-FAST-U2	2 Mbps base Fast Ethernet with 100Mbps burstability	1	$2,400		$2,400
EXO-FAST-SU	Setup-Ethernet Network	1	$3,500	$0	
	Sub Total			$2,000	$8,850
	Discounts				
	Total:			$2,000	$8,850

Usage above 15 Mbps:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Per Megabit
EXO-FAST-VU15	Variable Usage Cost per Megabit Above Base Amount ($/megabit)	1	$1,400

Usage above 2 Mbps:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Per Megabit
EXO-FAST-VU2	Variable Usage Cost per Megabit Above Base Amount ($/megabit)	1	$1,400

Note: Includes a reasonable number of re-boots per month
Press release Q1 99.

3 20 AMPS in VDC CUSTOMER'S INITIALS LP

FIGURE 6. An early Google Internet Data Center Services Contract. The first Internet data center services contract was signed by Google for 2.5 m^2 of space and 2 Mbps of bandwidth.

in scale, with diverse workloads (training, tuning, and serving) and frameworks (e.g., Jax and PyTorch). WSCs are now foundational societal infrastructure for the next wave of computing.

This period marked yet another step-function increase in the scale of WSC infrastructure, amid continued headwinds in technology scaling. Multiple generations of new TPU systems introduced new innovations, both at the chip (e.g., support for sparse computation) and system levels (new interconnects including optics and cooling). Multiple generations of hardware acceleration helped video workloads, with dramatic cost advantages, and enabled new video-centric capabilities.[29] (The impact to the media industry was recognized with a 2024 Technical Emmy Award.)

Software-defined servers were additionally optimized for software-defined memory,[20] multi-instruction-set-architecture heterogeneity (ARM servers), and other hardware features (e.g., chiplet-aware scheduling). Software-defined power management expanded to include broader types of oversubscription, at multiple levels of the data center, and across power and cooling.[23] Networking saw the introduction of SmartNICs at WSC scale with accelerated network offloads, and delay-based congestion control achieved 10× tail latency improvements.[1] SDN continued to evolve (e.g., topology engineering for efficiency and host-based repathing) as did new distributed programming models (e.g., to embrace more declarative programming).

Foundational server design continued to evolve: industry standards and open reference implementations; modular trays to help with form-factor proliferation; supply chain, root of trust, confidential computing, and other security optimizations; environmental sustainability, and lifecycle carbon footprint management. Beyond traditional monolithic designs, servers evolved to mini-distributed systems managed across multiple "arenas" (daughterboards with their own CPUs for management, security, networking, storage, and so on).

Storage scaling slowed down both for density and cost per byte. Hybrid shingled magnetic recording disks improved data density but required new data placement algorithms. Flash appliances supported independent scaling of compute and SSD storage. Data center design reduced cost and construction times with modular prefabricated infrastructure deployed in smaller increments, and physical server management started using robotics to automate repetitive tasks.

This five-year period also saw an increased focus on sustainability as a first-class design criterion. Ambitious targets to achieve net-zero emissions and 24/7 carbon-free energy by 2030 required several new optimizations across hardware (e.g., reduced embodied carbon), software (e.g., carbon-aware scheduling), and operations (e.g., 24/7 matching of supply and demand). The new reliability changes stemming from ever-smaller transistor sizes manifested in increased faults (silent data corruption),[16] motivating additional focus on testing and reliability. As always, performance and cost-efficiency continued to be paramount, but with the additional challenge of achieving these goals for a heterogeneous multitenant cloud infrastructure environment.

KEY LESSONS FROM 25 YEARS OF WSCs

Although our discussions have focused on Google, they also serve as an overview of the broader industry trends around WSCs. In the following section, we summarize the things we learned (sometimes the hard way!) and sketch some important design principles for WSCs.

SCALE AS A FIRST-CLASS DESIGN CONSIDERATION

WSC infrastructure needs to scale globally to meet the growth of its services. Scale amplifies many traditional design considerations. Small components can add (hundreds of) millions of dollars of cost. Small power overheads can similarly compound to (hundreds of) megawatts of power. Failures traditionally considered "rare" (e.g., one in a million) or "unique" (e.g., cows causing intermittent failures on a fiber line) can (and do) show up at scale, motivating different approaches to reliability. Maintaining a large WSC fleet can involve handling hundreds of millions of parts per year, so the logistics and supply-chain aspects of WSCs can lead to different design choices. *Simplicity matters: complex systems are inherently harder to deploy and maintain, more prone to failure, and harder to scale.*

WSC design is distributed-systems centric, across compute, storage, and networking. But even with such an approach, every 10× growth in scale requires new thinking because it breaks existing approaches. *Architects are never done: important choices need to be revisited with every epoch of growth.*

RISK MANAGEMENT: SYSTEMS, SERVICES, OPERATIONS, AND TEAMS

You can't innovate if you can't manage risk. Being able to introduce possibly unreliable new hardware or software without threatening the availability of higher-level services like search was the key enabler of innovation in the early days of WSC computing. *Technical defenses work at multiple levels*: systems reschedule around failed servers or disks, block-level checksums detect data corruption, test-driven software deployment detects bugs early, and so forth.

But beyond technical means, *culture is the biggest enabler of innovation*, and that's especially true for "operations," the WSC aspect that's perhaps the most underappreciated. Failures will happen, and embracing them as learning opportunities (via blameless postmortems) allows establishing a rational balance between being safe (but going too slow) and being reckless (going too fast). The site reliability engineering discipline has emerged as a key ingredient of WSC design, enabling deep changes even in already-large systems. For example, when Google announced their SDN deployment at scale, a telco executive commented, "What is amazing is not that you designed the first SDN backbone, it is that you could actually roll it out," pointing to the unusual capability of an operations team to manage risk by co-designing the solution and its rollout.

MANAGING SECURITY DEEPLY

Security needs to go deep. Early generations of WSCs started out like the Internet, relatively open and trust based, and defended well against external attackers with commercial incentives. But to defend against nation-state actors, much deeper defenses are required. In particular, servers must have a secure, separate silicon root of trust to validate and protect firmware and operating systems; all data must be encrypted, ideally even when being processed; all employees must use phishing-resistant two-factor authentications; systems should assume zero trust; important actions or access to user or customer information must use multiparty authorization; and all production code must be reviewed, and all binaries must have verifiable provenance. Defenses (including physical security) must be regularly tested by highly skilled red teams.

BUT EVEN WITH SUCH AN APPROACH, EVERY 10× GROWTH IN SCALE REQUIRES NEW THINKING BECAUSE IT BREAKS EXISTING APPROACHES.

These past investments have yielded results; today, public cloud and hyperscaler security significantly exceeds typical enterprise security. But security is far from being a solved problem. Supply-chain attacks against open source and commercial software threaten every system and need deep defenses at every stage. Systems are too complicated to configure and need built-in, automated audits to prevent mistakes. AI-assisted systems need to detect needle signs of compromise in ever-growing hay stacks of logs. Even better abuse detection needs to prevent bad actors from using public clouds to attack others. Communication systems need to prevent phishing or impersonation attacks while preserving privacy. The ever-growing importance of digital infrastructure will create ever-growing expectations for its security. *Ignore security at your own peril; only the paranoid survive.*

VERTICALLY INTEGRATED SYSTEMS DESIGN

WSCs *vertically integrate multiple layers of system design*: data center infrastructure (power, cooling, and building), silicon and hardware (accelerators, servers,

switches, and storage), systems (SDN, scheduler, and data center management), and services/platforms (infrastructure as a service and platforms as a service). Similarly, designs *optimize across the entire lifecycle*: supply chain, engineering design, deployment, operations, and sustainability.

Be intentional about co-design, clearly specifying appropriate division of functionality and separation of concerns across layers with well-defined interfaces and abstractions. A good example is using software to augment hardware fault tolerance when it is too costly to eliminate all risk in hardware. At the same time, co-design without discipline can create systems that are too closely coupled or create "tech islands" (see the "Technology Islands and Industry Ecosystems" section).

It is likewise important to *pick the right metrics* for co-design. Although hardware-specific metrics can be helpful, higher-level application-level metrics (and service-level objectives) additionally help identify opportunities to optimize for a global objective function (e.g., service cost per query). Often, *instrumentation and monitoring*[27] are indispensable; while they may come with some overhead for data collection, we have consistently found that instrumentation-guided optimizations far outweigh these overheads. ML further reinforces the value of carefully curated operational data.

WITHOUT THIS FOCUS ON COMPOSABILITY AND STANDARDS, YOU MAY END UP ON A "TECH ISLAND" UNIQUE TO YOURSELF, WHERE ONE CUSTOM COMPONENT FORCES ALL OTHERS TO BE CUSTOM TOO.

HARDWARE–SOFTWARE CO-DESIGN FOR WORKLOAD SPECIALIZATION

WSCs have always emphasized hardware–software co-design, and with decreasing general improvements from Moore's law, the case for workload specialization has become stronger. TPUs for AI training and inference represent the most prominent example of specialization across multiple layers: 1) systolic arrays for matrix multiply and optical switches at the *hardware* layer; 2) scale-up and scale-out systems with reconfigurable optical networks at the *systems* layer; 3) deep fusion, flexible parallelization, and communication scheduling in the *compiler*; 4) bfloat16 and fp8 numerics plus SparseCores at the *model/workload* layer; 5) optimizations like mixture of experts or neural architecture search at the *application and algorithmic layer*; and 6) energy optimizations, including liquid-cooled systems at the *data center* layer.

Our designs of video coding unit (VCU) video accelerators follow a similar approach to co-design across multiple layers.[29] Both examples demonstrate the benefits of thinking beyond classical co-design at the instruction-set architecture level to *embrace the full range of options across the systems stack*. Interestingly, the key techniques for successful co-design of new accelerators are not that different from those that underpin well-designed software systems: careful decomposition of the problem to minimize complexity, thoughtful definition of interfaces that abstract local information and minimize dependencies, and so on. At the same time, dealing with long lead times and costs for hardware development at scale *favor modular development/testing and reuse across projects*.

TECHNOLOGY ISLANDS AND INDUSTRY ECOSYSTEMS

The initial success of WSCs was driven by being different, out of necessity reinventing many of the then-conventional approaches to system design. However, as WSCs have scaled and their adoption has increased via public cloud providers, a broad industry ecosystem now supports WSC use cases, allowing "build versus buy" decisions.

Custom designs work best when they target some unique needs of WSC workloads or systems that are currently not satisfied cost-effectively by existing solutions in the market (for example, the design of TPU accelerators for WSC ML workloads). For more mature markets, however, volume economics often reduce costs and increase velocity, favoring products built on top of industry standards (for example, server and rack form factors).

Focus on building modular, composable, and interoperable architectures built on standardized interfaces; without this focus on composability and standards, you may end up on a "tech island" unique to yourself, where one custom component forces all others to be custom too. In many ways, this is the hardware equivalent of the monoliths versus microservices tradeoff.

A CULTURE OF LANDINGS VERSUS LAUNCHES

In more than 25 years of WSC design, we have learned a few important lessons about team culture. One of them is that it is far more important to focus

on *"what does it mean to land" a new product or technology*, instead of focusing on the launch. After all, it was the Apollo 11 landing, not the launch, that mattered. Product launches are well understood by teams, and it's easy to celebrate them. But a launch itself doesn't create success. Landings aren't always self-evident and require explicit definitions of success—happier users, delighted customers and partners, more efficient and robust systems—and may take longer to achieve.

Although picking such landing metrics may not be easy, forcing that decision to be made early is essential to success: *the landing is the "why" of the project*.

> *"WE CHOOSE TO GO TO THE ROOF NOT BECAUSE IT IS GLAMOROUS BUT BECAUSE IT IS RIGHT THERE. GO OUT THERE AND HAVE HUGE DREAMS, THEN SHOW UP TO WORK THE NEXT MORNING AND RELENTLESSLY INCREMENTALLY ACHIEVE THEM."*

A CULTURE OF REVISITING SUCCESSFUL DECISIONS

Across epochs, we have had to revisit past technical decisions, even when the original decision led to exemplary success. For example, for many years, we resisted building custom silicon because custom designs can easily be overtaken by the pace of improvements in general CPUs, and they require large volumes to amortize the engineering effort required to build them.

But then deep learning became practical, and the speedups of highly specialized processors over CPUs were immense, so we created our first TPU. Almost 10 years later, TPUs remain a key differentiator for ML-based products. Similarly, YouTube's large bandwidth costs, combined with the lack of a market for streaming video compression chips, led us to create VCUs, now in their third generation. A similar story has played out in several other design choices over the years. *Do not be afraid to revisit design decisions, especially at every step function of scaling.*

CREATING MOONSHOTS THROUGH ROOFSHOTS

Over the past 25 years, we have achieved orders-of-magnitude improvements (the much publicized 10× improvements, or "moonshots"). Many of them have been the result of the methodical, relentless, sustained pursuit of a series of smaller (1.3–2×) opportunities, or what we have internally dubbed "roofshots."

For example, early WSC infrastructure started with a pile of machines in third-party facilities and an intention to do better. We kept improving electrical efficiencies, changing how airflow was provisioned, learned how to prefabricate pieces of the facility for faster construction and higher cost-efficiency, and eventually figured out how to nearly eliminate scheduled downtimes. Roofshot after roofshot, and suddenly we had some of the most efficient and reliable data centers in the world, fundamentally different from a traditional design.

Seen from afar, these kinds of achievements could be mistaken as moonshots. They were, in fact, a sequence of roofshots. A sequence of roofshots can produce both quick returns and sustained transformative results. In the words of our late colleague, Luiz Andre Barroso: "We choose to go to the roof not because it is glamorous but because it is right there. Go out there and have huge dreams, then show up to work the next morning and relentlessly incrementally achieve them."

LOOKING AHEAD

After 25 years of WSCs, we are at an interesting inflection point. On one hand, computing demand is poised to explode, driven by growth in cloud computing and AI. On the other hand, technology scaling slowdown poses continued challenges to scale costs and energy efficiency. Combined, these trends mean that some of the most exciting years for WSC design are still ahead of us.[26] In the next sections, we summarize some grand challenges and opportunities for the community, specifically highlighting key themes around scaling, agility, trust, and sustainability, and the disruptive potential of using AI for WSC design.

THE NEXT WAVE OF SCALE: ACCELERATORS, SOFTWARE-DEFINED HARDWARE, AND THE ROOM AT THE TOP

Responding to the challenges outlined earlier will require continued innovation across both *efficient design* (custom silicon), *efficient utilization* (software-defined hardware), and *efficient software* (better algorithms).

We will continue to see a plethora of custom silicon accelerators, both coarse- and fine-grained, and optimized for heterogeneous accelerator-level parallelism. Beyond "doing existing things faster and cheaper,"

accelerators create opportunities for new functionality at the application level that were not possible before.

Software-defined hardware will need more sophisticated control and automation across multiple levels of software. Automation for data center management, including managing mechatronics and robotics in WSC environments, will become widespread. An important area of research will be around "room-at-the-top"[21] optimizations to address inefficiencies in higher-level languages and algorithms, software development practices, and hardware-specific tuning. WSC-scale co-design across hardware and software will continue to be critical to enable these.

OPTIMIZING THE TIME VARIABLE OF MOORE'S LAW: AGILITY, MODULARITY, AND INTEROPERABILITY

The traditional formulation of Moore's law (performance doubles every two years for the same cost) focuses on three variables: performance, cost, and time. As performance and cost improvements slow down, focusing on time—the *velocity* of hardware development—can be a good way to optimize the "area under the curve" for continued improvements. Incremental smaller benefits, when compounded, can still achieve exponential benefits.

To achieve such agile, faster improvements, we need to build more modular hardware platforms with appropriate investment in interfaces, standards, and so on. Chiplets in particular allow us to co-design in a multidie system context,[26] allowing cost advantages from die geometries, but also mix-and-match integration across heterogeneous intellectual property (IP) blocks and different process technologies.

ALTHOUGH WSC DESIGNS ARE FOCUSED ON AI/ML WORKLOADS NOW, WE HAVEN'T YET SEEN AN EQUIVALENT FOCUS ON USING AI TO DESIGN WSCs.

A particularly exciting development, open source hardware, enables a collaborative and higher-velocity ecosystem that hardware designers can build on: open source IP blocks (e.g., Caliptra root of trust), verification and testing suites (e.g., CHIPS alliance and OpenCompute), and even open source tools/process design kits (e.g., OpenRoad). Given how profound open source software has been to WSCs, the opportunity for open source hardware is significant.

TRUSTED COMPUTING: RELIABILITY, PRIVACY, SECURITY, AND SOVEREIGNTY

New reliability challenges, particularly recent increases in silent data corruptions, require new research: from discoverability, localization, and root cause analysis, to serviceability and resilience, to new architectural improvements. WSCs need to be designed to support large-scale in situ monitoring and testing, with improved software-defined resilience and adaptive service-level optimizations.

Similarly, emerging privacy and security challenges—both new internal and external threats—need new solutions. (We distinguish between security, an infrastructure property, and privacy, an application property.) Confidential computing and homomorphic encryption as well as even more sophisticated fraud/abuse/threat detection at higher levels of the stack all require significant investment. Sovereignty will become a first-class WSC design consideration to provide data residency and jurisdictional clarity and enable more compartmentalized security properties.

SUSTAINABILITY FOR A SOCIETAL INFRASTRUCTURE

Environmental sustainability is another challenge and opportunity, touching different areas such as greenhouse gas emissions, water usage, and hazardous materials. Emissions fall into three categories: *scope 1* emissions (e.g., from process gasses with high global warming potential), *scope 2* emissions (e.g., from purchased electricity), and *scope 3* emissions (e.g., indirect emissions in the value chain). Opportunities abound in reducing all three of these categories: reduced transportation footprint, improved water utilization and efficiency, minimizing hazardous materials, reduced energy consumption and cleaner sources, adopting circular economies, moving computing to the cloud, and so forth. An important first step in this area is to adopt consistent metrics to measure and report environmental impact as well as broader industry ecosystems to standardize and share best practices.

AI FOR WSC DESIGN

Although WSC designs are focused on AI/ML workloads now, we haven't yet seen an equivalent focus on *using AI to design WSCs*. Future WSCs will feature hardware and software *created* by AI. AI can enable faster iterations with increased coverage and diversity but will also pose new challenges for testing and validating. AI-assisted tools can potentially enable rewrites

of large software systems, opening up new opportunities to quickly evolve out of old ecosystems and evolve interfaces across complex interdependent systems. AI/ML techniques can also optimize control and automation, particularly in the context of software-defined hardware.

AI may be particularly transformative for chip design.[11] The last few years have seen significant advances in using ML across the lifecycle, including Register Transfer Language (RTL) synthesis, placement and routing, and verification and validation. More sophisticated models across larger state spaces can optimize across multiple complex objectives spanning power, performance, area, timing rules, congestion, and so forth. Emerging large language models may dramatically simplify the developer experience. Unlike today, where designing a chip can take hundreds of people working for months, ML may enable the dream of designing chips in weeks or even days. We are only getting started in this area.

CLOSING REMARKS

This article built on the work of thousands of talented engineers and researchers who have worked on WSCs over the past quarter decade. Their work has helped develop WSC from a set of hand-built computers in a lab running rudimentary web search algorithms to the foundational societal infrastructure it has become today.

We want to particularly dedicate this article to our late colleague, Luiz Barroso. Luiz joined Google in the early 2000s and played a key part in enabling many of the ideas discussed in this article. His work laid the foundation of an entire industry. He didn't just work on how to scale WSCs, he also thought deeply about how to scale WSC innovation. We are grateful for the opportunity this article gave us in looking back at the history of WSC innovation and hope that it inspires others in the community to continue his great legacy.

As we reflect on 25 years of WSCs and look ahead, the future looks bright. The next era of WSCs will continue to see disruptive innovations: from "mud to cloud," that is, from data center infrastructure to broader cloud computing services, and from "chip to ship," that is, from the design of hardware to its deployment and use in production. Co-design and collaboration—across the hardware–software stack, across disciplines, and across communities—will be key to this exciting new future.

REFERENCES

1. S. Arslan et al. "Bolt: Sub-RTT congestion control for ultra low-latency," in *Proc. 20th USENIX Symp. Netw. Syst. Design Implementation (NSDI 23)*, Boston, MA, USA, Apr. 2023, pp. 219–236.
2. L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007, doi: 10.1109/MC.2007.443.
3. L. A. Barroso, "A brief history of warehouse-scale computing," *IEEE Micro*, vol. 41, no. 2, pp. 78–83, Mar./Apr. 2021, doi: 10.1109/MM.2021.3055379.
4. L. A. Barroso, J. Dean, and U. Hölzle, "Web search for a planet: The google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, Mar. 2003, doi: 10.1109/MM.2003.1196112.
5. L. A. Barroso, U. Hölzle, and P. Ranganathan, *The Datacenter as a Computer: Designing Warehouse-Scale Machines*. Berlin, Germany: Springer Nature, 2019.
6. L. A. Barroso et al., "Attack of the killer microseconds," *Commun. ACM*, vol. 60, no. 4, pp. 48–54, Apr. 2017, doi: 10.1145/3015146.
7. M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Proc. 7th Symp. Operating Syst. Design Implementation (OSDI)*, 2006, pp. 335–350.
8. F. Chang et al., "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, 2008.
9. J. C. Corbett et al., "Spanner: Google's globally distributed database," *ACM Trans. Comput. Syst.*, vol. 31, no. 3, pp. 1–22, 2013.
10. J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, 2013, doi: 10.1145/2408776.2408794.
11. J. Dean, "The potential of machine learning for hardware design," in *Proc. ACM/IEEE 58th Design Autom. Conf. (DAC)*, 2021.
12. S. Dev et al., "Autonomous warehouse-scale computers," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2020, pp. 1–6, doi: 10.1109/DAC18072.2020.9218509.
13. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008, doi: 10.1145/1327452.1327492.
14. M. Dalton et al., "Andromeda: Performance, isolation, and velocity at scale in cloud network virtualization," in *Proc. 15th USENIX Symp. Netw. Syst. Design Implementation (NSDI 18)*, Renton, WA, USA, 2018, pp. 373–387.
15. S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in *Proc. 19th ACM Symp. Operating Syst. Princ.*, 2003, pp. 29–43, doi: 10.1145/945449.945450.

16. P. H. Hochschild et al., "Cores that don't count," in *Proc. Workshop Hot Topics Operating Syst. (HotOS)*, 2021, pp. 9–16, doi: 10.1145/3458336.3465297.
17. S. Jain et al., "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.
18. N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2017, pp. 1–12.
19. S. Kanev et al., "Profiling a warehouse-scale computer," in *Proc. 42nd Annu. Int. Symp. Comput. Archit. (ISCA)*, 2015, pp. 158–169.
20. A. Lagar-Cavilla et al., "Software-defined far memory in warehouse-scale computers," in *Proc. 24th Int. Conf. Archit. Support Program. Lang. Operating Syst. (ASPLOS)*, 2019, pp. 317–330, doi: 10.1145/3297858.3304053.
21. C. E. Leiserson et al., "There's plenty of room at the Top: What will drive computer performance after Moore's law?" *Science*, vol. 368, no. 6495, 2020, doi: 10.1126/science.aam9744.
22. H. Liu, R. Urata, and A. Vahdat, "Optical interconnects for scale-out data centers," in *Optical Interconnects for Future Data Center Networks. Optical Networks*. New York, NY, USA: Springer-Verlag, 2012, pp. 17–29.
23. S. Li et al., "Thunderbolt: Throughput-optimized, quality-of-service-aware power capping at scale," in *Proc. USENIX Symp. Operating Syst. Design Implementation*, 2020, pp. 1241–1255.
24. M. Marty et al., "Snap: A microkernel approach to host networking," in *Proc. 27th ACM Symp. Operating Syst. Princ. (SOSP)*, 2019, pp. 399–413.
25. L. Poutievski et al., "Jupiter evolving: Transforming Google's datacenter network via optical circuit switches and software-defined networking," in *Proc. ACM SIGCOMM Conf.*, 2022, pp. 66–85.
26. P. Ranganathan, "A six-word story on the future of VLSI: AI-driven, software-defined, and uncomfortably exciting," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Kyoto, Japan, 2023, pp. 1–4, doi: 10.23919/VLSITechnologyandCir57934.2023.10185339.
27. G. Ren et al., "Google-wide profiling: A continuous profiling infrastructure for data centers," *IEEE Micro*, vol. 30, no. 4, pp. 65–79, Jul./Aug. 2010, doi: 10.1109/MM.2010.68.
28. C. Reiss et al., "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. 3rd ACM Symp. Cloud Comput. (SoCC)*, 2012, pp. 1–13.
29. P. Ranganathan et al., "Warehouse-scale video acceleration," *IEEE Micro*, vol. 42, no. 4, pp. 18–26, Jul./Aug. 2022, doi: 10.1109/MM.2022.3163244.
30. A. Singh et al., "Jupiter rising: A decade of clos topologies and centralized control in Google's datacenter network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 183–197, 2015.
31. M. Schwarzkopf et al., "Omega: Flexible, scalable schedulers for large compute clusters," in *Proc. 8th ACM Eur. Conf. Comput. Syst. (EuroSys)*, 2013, pp. 351–364.
32. V. Sakalkar et al., "Data center power oversubscription with a medium voltage power plane and priority-aware capping," in *Proc. 25th Int. Conf. Archit. Support Program. Lang. Operating Syst. (ASPLOS)*, 2020, pp. 497–511, doi: 10.1145/3373376.3378533.
33. M. Tirmaz et al., "Borg: The next generation," in *Proc. 15th Eur. Conf. Comput. Syst. (EuroSys)*, 2020, pp. 1–14.
34. A. Verma et al., "Large-scale cluster management at Google with Borg," in *Proc. 10th Eur. Conf. Comput. Syst. (EuroSys)*, 2015, pp. 1–17, doi: 10.1145/2741948.2741964.
35. K. Yap et al., "Taking the edge off with espresso: Scale, reliability and programmability for global internet peering," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, 2017, pp. 432–445.

PARTHASARATHY RANGANATHAN is an engineering fellow and area technical lead for systems hardware and data centers at Google, Mountain View, CA, 94043, USA. Ranganathan received his Ph.D. degree in computer engineering from Rice University. He is a Fellow of IEEE and the Association for Computing Machinery. Contact him at partha.ranganathan@google.com.

URS HÖLZLE is a fellow at Google, Mountain View, CA, 94043, USA, who leads the development of technical infrastructure. Hölzle received his Ph.D. degree in computer science from Stanford University. He is a Fellow of the Association for Computing Machinery and the American Association for the Advancement of Science and a member of the Swiss Academy of Technical Sciences and the National Academy of Engineering. Contact him at urs@google.com.