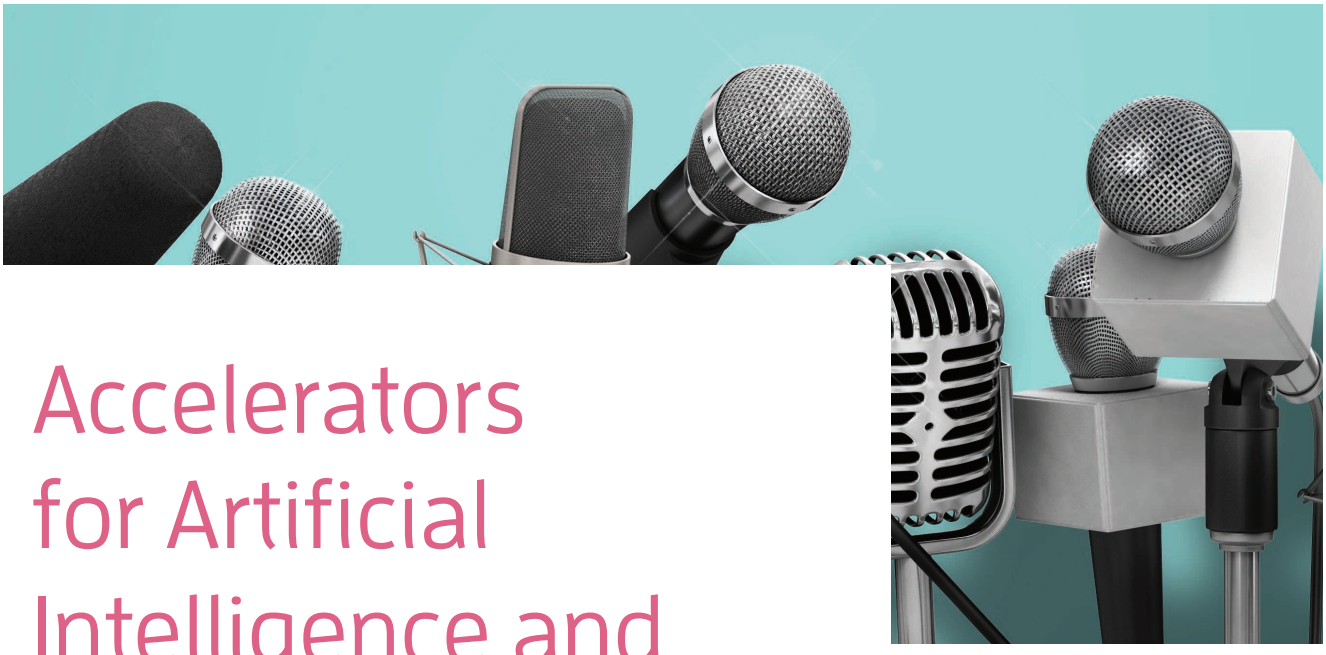

Accelerators for Artificial Intelligence and High-Performance Computing

Dejan Milojicic, Hewlett Packard Labs

Computer hosts a virtual roundtable to discuss artificial intelligence and high-performance computing.

In this virtual roundtable, we are looking at accelerators for artificial intelligence (AI) and high-performance computing (HPC). The roundtable was organized by Dejan Milojicic. Joining him are three experts in the field: Paolo Faraboschi, Satoshi Matsuoka, and Avi Mendelson.

Digital Object Identifier 10.1109/MC.2019.2954056
Date of current version: 12 February 2020

DEJAN MILOJICIC: We're here to talk about HPC and AI, but let's start with a little background. Tell us something about your experience in the field.

PAOLO FARABOSCHI: I received my Ph.D. from the University of Genoa, Italy, in electrical engineering and computer science, in 1993. I am currently a Hewlett Packard Enterprise fellow, vice president, and lead researcher in the Systems Architecture Lab at Hewlett Packard Labs. My technical interests lie at the intersection of hardware and software and include HPC, workload-optimized systems-on-chip (SoCs), and highly parallel systems.

I was the lead hardware architect on The Machine Project (2014–2017), and I am now the technical principal investigator of Hewlett Packard Enterprise's PathForward project, in collaboration with the U.S. Department of Energy (since 2017), accelerating technology toward exascale computing. In the past, I was a key contributor to Hewlett Packard's Project Moonshot on energy-efficient, software-defined servers (2009–2013); I led system-simulation activities (the COTSon simulator, 2004–2008); and I developed custom-instruction-level, parallel,

0018-9162/20©2020IEEE

very-long-instruction, word-embedded processors (the ST200/Lx family, 1997–2003) and compilers.

SATOSHI MATSUOKA: I have been a researcher on HPC systems since my Ph.D. days. I've worked on various system-software research on issues such as parallel programming languages and their runtime systems, resource-scheduling algorithms, large-scale system resilience, low-power computing in HPC, and scaling big data and machine learning on large HPC systems. Also, as a division leader of research infrastructures at the Global Scientific Information Computing Center, Tokyo Institute of Technology, since 2001, I have designed, deployed, and operated a series of modern-leadership supercomputers, especially the Tsubame series, which became the fastest machines in Japan, in 2006; the first supercomputers to deploy graphics-processing units (GPUs) at scale, in 2008; the first petascale system in Japan, in 2011; and the "greenest" supercomputer in the world multiple times, in 2013–2014 as well as 2017.

In April 2018, I became a director of the Riken Center for Computational Science (R-CCS), Japan's tier 1 supercomputing research and infrastructure center, hosting the leadership of the K computer and undertaking the development of the successor post-K machine with our corporate partners, which will be the first exascale machine in the world, to be deployed in 2020. Furthermore, I am leading the R-CCS research on the convergence of big data and modern AI with HPC as well as investigating the future of HPC hardware and software architectures, anticipating the arrival of the post-Moore era, in the late 2020s.

AVI MENDELSON: I am active in different aspects of HPC systems, including research in processor architectures for HPC; accelerators for HPC, such as general-purpose GPUs; field-programmable gate arrays (FPGAs); and heterogeneous systems. I started my research while I was at Intel and continue it now that I am in academia. Recently, I expanded my research to hardware support for machine learning and security aspects in HPC infrastructure. I am also involved with innovation-related activities as part of a European Union project, Eurolab4HPC (https://www.eurolab4hpc.eu/), that focuses on using open-source software and hardware in HPC and different forums that examine possible roadmaps in the field.

MILOJICIC: Let's start with the current state of HPC in AI. What is the current state of the field? What has been accomplished to date?

FARABOSCHI: If I had to define the state of AI and HPC today, I would call it "the era of mainstream heterogeneity." After decades of steady tick-tock gains driven by semiconductor-process improvements, we've seen a slowing of the traditional means of increasing processor speeds through the mechanisms of Moore's law. Today's architectures (and those in the foreseeable future) will need to rely a lot more on accelerators to deal with specific problems. I think the jury is still out on whether these accelerators will continue to look like GPUs or something else, such as processors that are considerably more dedicated and specialized. My vote goes for something a bit more dedicated, still optimized to a domain rather than to a single application.

Because HPC is increasingly relying on massively parallel accelerated hardware, it is starting to force software developers to create parallel software and think hard about data-domain decomposition that weakly scales almost without limits. For the exascale generation, which should arrive in roughly 2021 or 2022 by the latest projections, we will be seeing applications that run a billion-way parallelism. This is an astounding number that was unthinkable only a few years ago, but it is now almost taken for granted.

MATSUOKA: Until around 2004, achieving the highest single-threaded core/chip performance was technically feasible for HPC. As a result, a majority of systems still used hardware somewhat dedicated to HPC, such as Cray/NEC vector processors. The growth of the overall system parallelism was slow, and the system software and applications were still tailored for such systems.

There was a rise of commodity-hardware–based clusters that began in 1993, but these clusters were rather low-end machines and no match for the leadership-class machines, such as the U.S. Accelerated Strategic Computing Initiative machines and the Japanese Earth Simulator. However, when Dennard scaling ended around 2004, HPC quickly transitioned into increasing the parallelism as much as possible to achieve the highest performance with the lowest die area/power/cost. It attempted to utilize the overall IT-industry ecosystem, which went for parallelism for performance acceleration, including multicore CPUs and many-core GPUs, and algorithms, software, and applications adapting to the change.

The transition from the custom instruction-level/vector parallel machines to massively parallel machines has been remarkably successful. It has continued the so-called Moore's law, which projects an exponential increase in system performance, roughly a factor of 1,000 every 10 years. It has enabled the solutions to many difficult scientific problems that were deemed impractical at massive scale.

ROUNDTABLE PANELISTS

Paolo Faraboschi is a fellow at Hewlett Packard Labs. His research interests include the intersection of architecture and software. Faraboschi received a Ph.D. from the University of Genoa, Italy. He is a Fellow of the IEEE. Contact him at paolo.faraboschi@hpe.com.

Satoshi Matsuoka is a full professor in the Global Scientific Information and Computing Center at Tokyo Institute of Technology. His research interests are large-scale parallel computing including clusters and grids as well as low-power and accelerated high-performance computing. Matsuoka received a Ph.D. in information science from the University of Tokyo. He is a Member of the IEEE, ACM, and the Information Processing Society of Japan. Contact him at matsu@is.titech.ac.jp.

Avi Mendelson is a professor of computer science and electrical engineering at Technion–Israel Institute of Technology, Haifa. His research interests include computer architecture, operating systems, reliability, cloud computing, and high-performance computing. Mendelson received a Ph.D. from the University of Massachusetts, Amherst. He is a Fellow of the IEEE. Contact him at avi.mendelson@tce.technion.ac.il.

We see examples in the ACM Gordon Bell awards, where scientific breakthroughs are achieved with multipetaflop performances. Now, we are on the verge of achieving exascale, where machines that are 50–100 times more powerful than those of the early 2010s are being planned. As Paolo noted, these will be deployed in roughly the 2020–2023 timeframe, and they will be enabled by large-scale projects in respective regions of the world, including the United States, Japan, China, and Europe.

MENDELSON: I believe that, from the system point of view, we are at a kind of inflection point. On one hand, we already reached the "endpoint" of some technologies. For example, we cannot significantly increase clock speeds anymore. We can use a huge amount of main memory, but we don't know how to manage it efficiently. Thus, it seems like a major change is about to happen, but it is not clear when it will happen, how the "new computational world" will look, and how long the transition will take. We need to examine this change from two different points of view: the system point of view and the usage-model point of view.

From the system point of view, people are using the term "end of Moore's law" to indicate that the technology doesn't scale anymore. It doesn't scale anymore, but that doesn't mean that we can't increase speeds. The main problem is that achieving a new stage of power/performance does not depend anymore on a single factor but on many factors.

In the past, you could trust that frequency scaling and architecture enhancements would double the performance of software. Today, we improve performance mainly by using accelerators and many cores. Thus, to take advantage of the potential of the next generation of hardware, you also need to change your software stack or the optimization that you are using and, in many cases, the algorithm. For example, you may need to replace floating-point operations with quantization. This has caused the pace of change to significantly slow down and the cost of development to significantly increase, and even more significantly, it makes the market move from general-purpose systems to domain-specific solutions. But domain-specific systems have a smaller market, and they reduce the interest of large companies to invest in new ideas, except for applications such as machine learning.

From the usage-model point of view, we are facing another major challenge; the market is moving from a computational-centric point of view (for example, what is the best way to compute) to a data-centric view or a combination of both. For example, it is known that the quality of many machine-learning algorithms depends on the amount (and quality) of data available to train them. It is also known that much of the energy we are spending for computation depends on the data movement and not only on the computations. Thus, building an efficient system requires a deep understanding of the data structure, memory subsystem, communication subsystem, and processor architecture.

MILOJICIC: So those are the problems. What are the challenges that we are facing? What problems do we need to solve to continue to expand HPC?

FARABOSCHI: It's like what Avi just said. One of the key challenges I see is preserving a system balance that helps programmers' productivity without requiring heroic, one-off, unportable coding efforts for each acceleration flavor du jour. In 2022, after the world will have recovered from the exascale "race to the Moon," we may end up with a handful of unbalanced machines. I expect that the scientific community, which is the real stakeholder for HPC, will take a step back and start asking for more balanced, possibly smaller, systems that deal with sparsity, irregular computation, and more complex workloads.

Two examples of clearly imbalanced areas that come to mind are memory and fabric. Since most AI and HPC algorithms are bandwidth hungry, we are seeing systems moving to an all-high-bandwidth-memory (HBM) configuration. That will happen very soon. However, it comes at the expense of memory capacity, so developers will find themselves having to search for locality in places where there may be none. When it's possible to trade compute for memory capacity, this state of affairs will lead to better algorithms, but, in other cases, it may slow down the science. So we will have to find better ways to keep bandwidth and capacity in balance. Similarly, on the interconnect, we are seeing trends that try to keep the interconnect cost and power under control by lowering the ratio between the fabric injection and computation. This again exacerbates the imbalance between local and remote computation.

MATSUOKA: As Paolo said, there are several problems facing the field. Most of them are related to the slowing down of semiconductor advances or approaching the so-called end of Moore's law, where there would no longer be a free lunch in increasing the number of transistors over time while sustaining the chip power. Thus, the days of an ever-increasing number of cores is starting to end. The expansion of many-core architectures is already starting to slow, as we can no longer facilitate cores without the corresponding rise in the die area, power, and cost.

The problem spans all of IT but affects the HPC field the most, as it is the most performance-conscious domain. There are various solutions being worked on, but there is no single panacea to the problem, and discovering some sort of discipline that can sustain a continued speed-up over time, something that will replace lithographic techniques for shrinking elements and thereby increase in the cores, is been sought, not just one-time solutions that cannot be repeated.

In fact, we're starting to see that some strategies for increasing performance cause another set of problems to appear. For example, if you customize the architectures to tackle heterogeneous workloads, you can easily find that the limit on the total number of transistors can be a serious problem. For example, HPC recently helped to resurrect AI by accelerating deep learning over several orders of magnitude to make it practical. When it was invented in the 1980s, it was totally impractical. However, further recent research has shown that architectures more aggressively tuned for deep learning can achieve further dramatic acceleration, such as the use of very-low-precision arithmetic. However, this kind of change will not help the majority of HPC workloads that fundamentally require high precision.

The expansion of many-core architectures is already starting to slow, as we can no longer facilitate cores without the corresponding rise in the die area, power, and cost.

Another problem with heterogeneous computing is how to develop appropriate heterogeneous algorithms and enable effective programming across multiple architectures while maintaining some degree of portability. We have devoted a great deal of R&D to using GPUs in AI and seen a lot of success. However, it has taken a lot of research to learn how to use these processors across many disciplines. With architectures that require a customized program to be able to accelerate diverse applications, we will find that effective programming will be a challenge.

MENDELSON: There are many technical issues we need to address, including better handling power, massive parallel systems, treating security as a first-class citizen, and more. But on top of all of the technical problems, I think that the next main challenge is the need for multidisciplinary understanding, training, and experience. Our current educational system, the industry, and even the professional organizations, including the Association for Computing Machinery and IEEE, are organized around single disciplines, such as computer science, electrical engineering, movie makers, performance art, and so on. But building sophisticated new user interfaces may involve all of them. I believe that such a major restructuring of our professional communities is a key for future success.

MILOJICIC: Let's get into the details. What are the technical problems that we need to solve to make HPC advance?

FARABOSCHI: We've already mentioned keeping the system balance. On the technology side, this requires a different way to compose memory and fabric resources, which, in turn, requires lower cost and energy technologies. For example, at Hewlett Packard Labs, we have been investing in optical interconnects for more than a decade, and we have been advocating new interconnect protocols (like Gen-Z) that enable far more flexible and richer sets of system configurations. While we cannot overcome the laws of physics (the speed of light latency comes to mind), we can definitely make it cheaper and more efficient to increase the bandwidth per unit of computation. We can enable multiple tiers of memory and make storage more composable. We can utilize different accelerators to be able to share scarce resources (such as memory and interconnects) much more efficiently and through a shallower software stack than we do today.

MATSUOKA: In addition to system balance, we need to cope with increasing heterogeneity in the machine and massive parallelism to control complexity and enable us to use them effectively. Another problem is to find alternative methods of sustained speed-up. In the short to mid term, we will saturate the floating-point operations per second (FLOPS) in the system, and in the long term, we'll need to look to increase the speed of the system by increasing the bandwidth and developing algorithms, system software, and applications to exploit that bandwidth. For example, we can use high-bandwidth systems to utilize implicit solvers to get acceleration rather than using direct solvers that require more FLOPS. The former will achieve a far greater time to solution and higher performance over time as the system bandwidth increases.

In the longer term, we have to strive toward alternative device technologies to speed up beyond the CMOS, but this is currently deemed very difficult as those technologies are typically associated with undesirable properties, such as the further demand for increased parallelism possibly being curtailed by the Amdahl's law. Also, we need to look at alternative forms of computing, including neuromorphic and quantum, but they are still far away from being practical. At the moment, they're not practical for a couple of reasons. It is difficult to create appropriate hardware, especially quantum hardware, to surpass the current performance achieved by conventional CMOS-based computing. It is also hard to broaden the domain of specialized hardware to much larger problem domains; currently, their applicability is so narrow that, even with a machine that could exhibit supremacy in some problem domain, it would be quite infeasible to deploy it widely, let alone use it to solve practical problems.

MENDELSON: We need to focus on data centricity. To enable systems to be more data centric, we need to move more computations toward the data, that is, to make the computation where the data are rather than moving the data to the compute engine. To support it, we need to create new software/hardware interfaces and have a better software/hardware codesign. From the hardware point of view, we need to develop better integration technologies, better communication systems inter- and intra-system, different cooling techniques, new protection mechanisms, and more.

From the software point of view, we need to increase parallelism, reduce the serialization parts, and massively reduce the communication cost by increasing locality, reducing the number of bits needed to represent data, using new compression techniques (in software and hardware), and moving to approximate computing whenever possible. On the top of all these "traditional" factors, security and privacy will play a major role. Unfortunately, there are many times when security and the optimal utilization of resources contradict each other, so new algorithms and hardware need to be developed to cope with the challenges.

> Research has shown that architectures more aggressively tuned for deep learning can achieve further dramatic acceleration, such as the use of very-low-precision arithmetic.

MILOJICIC: Let's talk about some concrete technologies, such as application-specific integrated circuits (ASICs) and GPUs. Why might one of these technologies be better than the other? Why might ASICs dominate over GPUs (or vice versa)?

FARABOSCHI: First, let me define what I think ASIC means in this context. I think of an ASIC as a highly optimized computing element with limited programmability. The tradeoff between ASICs and more programmable components (such as GPUs and CPUs) has been studied for a long time. ASICs are intrinsically more energy and cost-efficient because they save the fetch-decode-execute overhead. Depending on the algorithm, they can be up to 10 times more efficient. Because they are usually dedicated to a single application, they can be cheaper than a full processor. However, because the nonrecurring engineering cost of developing the ASIC is spread over only a single application, that cost burden can be high, especially given the astronomical costs of leading-edge silicon processes.

To summarize, ASICs have traditionally been successful when the time-to-market value is high (they can hit production faster than a general-purpose processor), when the application field is more or less standard (engineers can customize them without fear of the ground moving under their feet), and the volumes are high enough to amortize the nonrecurring engineering (NRE). For example, Google's tensor processing unit (TPU), as described in a recent article, is a great example of an ASIC where all of the criteria match. In other cases, I can see the argument becoming much weaker.

MATSUOKA: I agree. Obviously, any customized hardware can achieve the utmost efficiency, both in absolute performance as well as power, but this comes at the cost of a lack of flexibility and associated design and fabrication expenses, which can be quite high. That said, when one could identify some functions that were applicable to a significant number of chips and/or applications such that the NRE costs could be amortized, it would be effective, and

this has been the trend for modern architectures, especially in mass-market devices, including cell phone SoCs. For example, for every generation of Apple iPhone SoCs, the number and area of the fixed-function units are increasing. This is a favorable development, as the SoCs achieve high functionality, such as facial recognition with very low power, and their cost can be amortized over the multimillions of iPhones that are sold.

MENDELSON: The ASIC is an excellent example of special-purpose versus general-purpose computing. It was proven that a good software/hardware co-design and special hardware that support it can provide a factor of tens to hundreds of times better performance and/or power over a general-purpose GPU implementation. On the flip side, it may take you a long time to develop it and significant effort to maintain it.

One way to compromise between efficiency and the time to market is the use of FPGAs as an intermediate step since, although FPGAs are less efficient than ASICs, they are much more flexible for designing and debugging. More than that, modern developing environments for FPGA systems include support for high-level programming languages (for example, HTTP live streaming and Open Computing Language) that ease the migration from general-purpose to FPGA. After the algorithm is "stabilized" and seems to work correctly, it can be optimized and transferred to ASICs in a relatively fast and easy way.

MILOJICIC: Of all of the alternative technologies we've considered, which one stands the best chance to supplement existing accelerators and general-purpose computers?

FARABOSCHI: I already mentioned richer interconnects, and I will also highlight converged interconnects that can, for example, capture the convergence of storage, messaging, and memory traffic. This is exactly why I'm a big fan of technology like Gen-Z that is designed from the ground up to be versatile for all these use cases.

New storage elements may actually start to become real soon. The industry has been predicting software-configuration management (SCM) to mature for a decade, and I think we're finally starting to see some promising life signs. That will do to Flash what Flash did to disks (that is, push them to a lower tier). And due to the nature of SCM [be it 3D XPoint, a power-train control module, resistive random-access memory (ReRAM), or something entirely different], it will be usable as extended memory and, hence, alleviate the pressure of an all-HBM system. Many of the AI workloads are very data intensive, so they will be able to take advantage of a better data-provisioning system right away.

Finally, I want to mention mixed-precision arithmetic. Gone are the days when all that mattered was the 64-bit floating point. New science is looking (about time!) into using lower-precision arithmetic, possibly even analog computation, to solve some of the problems that engineers are dealing with, starting with some of the deep-learning inference techniques

MATSUOKA: As indicated, the key to performance increases in the middle term is to increase the bandwidth in the system, memory, interconnects, input/output (I/O), and so on using new device and packaging technologies. This is where there is still significant headroom for growth. As an example, the new A64FX advanced reduced-instruction-set computing machine (ARM) chip that the R-CCS developed with Fujitsu, despite being general purpose with many cores and a standard ARMv8 set enabling it to run general workloads, is several times faster compared to any mainstream server CPU of the same generation. This is because its memory bandwidth is almost an order of magnitude faster, at 840 GB/s using 2.5-D HBM technologies, compared to the latter using conventional double-data-rate-4 memory, at approximately 100 GB/s, and most HPC applications, such as computational fluid dynamics, are memory-bandwidth bound. By achieving a further performance increase via low-power memory devices that enable multilayer 3D stacking, packaging technologies that facilitate multiple vias in 3D for an even greater bandwidth increase, and photonic interconnects that facilitate sustained high-bandwidth-across chips, we should be able to achieve continued increase even after the FLOPs saturate due to the end of the Moore's law.

MENDELSON: Process-in-memory and process-near-memory. Both of them facilitate off-loading computations from general processors and even from accelerators. Another aspect that can significantly help to accelerate computations are new techniques for fast and wide (bandwidth) communication between chips and systems. Applying that may also enable us to convert many current algorithms from being optimized to the locality of the references to distributed computations where locality is mainly preserved at the different nodes.

MILOJICIC: What specific industry verticals offer the most promising applications for the new accelerators? Where do you think the early applications will be found?

FARABOSCHI: First, at the edge (including mobile devices and embedded systems), accelerators have been the norm for at least the past two decades, so I don't think anything has significantly changed recently.

When I think about the data center, historically, accelerators have mostly been successful in what I call "high-value" computation. For example, financial services, oil and gas, and drug discovery are three verticals where any computational advantage directly translates into (very large) financial

benefits right away. Consider, for example, the cost of drilling for oil in the wrong place, starting a phase three trial of the wrong drug, or the cost of slowing the reaction time for low-latency trading. It should be clear that these areas (and a few others with similar properties) can tolerate very expensive components and large software-optimization efforts, which is what many accelerators require.

More recently, large service providers are finding themselves increasingly looking at accelerators for another reason: the large amount of processing that they have to sustain (in terms of "free products") to get to the single revenue-generating click on an advertisement link. That is probably the reason why the hyperscale service providers are among the first to truly embrace acceleration at scale, be it GPUs, dedicated ASICs (like Google's TPU), or FPGAs (like Microsoft's Catapult).

MATSUOKA: Accelerators, by definition, are narrow in their applicability and have been avoided for data center workloads. If we have killer application areas where acceleration is economically feasible or new technologies that enable sets of heterogeneous accelerators to be easily reconfigured dynamically, such as with FPGAs with appropriate workloads, or we can achieve a much faster design cycle with a modern tool chain, then we may see their proliferation. This is already happening with areas including AI and security. We may see other areas of acceleration, such as I/O, which will enable many data center workloads to be accelerated, as many applications are I/O bound.

MENDELSON: I believe that, in the future, we will see more SoC-like designs that integrate ASICs, GPUs, and general-purpose processors. All of the applications that need high performance and low power will enjoy it, including automotive systems, autonomous cars, cellular phones, robots, and so on.

MILOJICIC: Can we compare three different groups of high-performance technology? The first would be reconfigurable coarse-grain architectures, including FPGAs. A second would be more static architectures, such as ASICs. Finally, we have to look at the more general-purpose accelerators, including GPUs. How should we compare these three classes of technology?

FARABOSCHI: As someone once told me, "FPGA will always be a very promising technology." Reconfigurability, whether it is coarse or fine grain, comes at a cost, and unless that cost is justified by providing sufficient value over time, it will always be greater than the cost of solving the same problem using nonreconfigurable logic. Nonconfigurable logic will always have a fundamental cost, energy, and performance advantage.

I see the role of FPGAs increasing as prototyping vehicles and some applications that require field upgrades where it is too expensive to replace the part. For example, for AI at the edge, FPGAs may be a great technology (at the right cost structure). This said, the increasing cost of leading-edge silicon processes may give FPGAs an edge that changes the fundamental equation. Let's postulate that the development of a 5-nm ASIC design will become so expensive that it requires hundreds of millions of parts to amortize the NRE. In that case, an FPGA, which can spread the NRE of a single part over several market segments, may have a fundamental advantage that is not there today. So, I'm keeping my eyes open and tracking the reconfigurable field, but it will take a couple of process generations to cross that point, I think.

MATSUOKA: In the modern volume market with ASIC/CPU hybrids in an SoC, such as cell phones, turning specific application-programming interface (API) functions into ASIC hardware evolves over time; certain functions are initially implemented in software. When the API is stable and there is sufficient motivation in terms of performance and power so that such a function will be made into customized hardware, it will be done in the future generation of chips. One can envision that there could be an intermediary step where the API functions in software would be implemented as hardware in an embedded FPGA and eventually turned into an ASIC portion. As such, FPGAs will be used as an intermediary prototyping vehicle for testing fixed hardware with very specialized functions until they become sufficiently stable. This will actually be more motivating for data centers and HPC, where the volume is lower and the workload more diverse, such that the conversion of APIs into hardware will occur during the much longer range and be a challenge.

MENDELSON: The use of ASICs and GPUs heavily depends on the hardware/software interfaces and characteristics of the software that it executes; for example, GPUs assume that the software has a huge amount of parallelism that can be exposed. The main advantage of general-purpose processors is their flexibility to meet different software characteristics, but their main disadvantage is the amount of power they consume.

The use of reconfigurable general-purpose architectures enables us to achieve the flexibility of general-purpose processors with the power consumption near to ASICs. Now, since the reduction of power and energy can be "translated" to a higher frequency and better performance, it can also serve to close the performance gap between ASICs and general-purpose computing engines.

MILOJICIC: So far, we have been talking about existing accelerator technology. When do you think a new accelerator technology will appear? What do you think it might look like?

FARABOSCHI: By definition, accelerators appear when new important

applications do. Right now, we're at the peak of the hype curve for deep learning, machine learning, and AI, so it's a natural consequence that we're seeing an abundance of AI accelerators. Natural selection will run its course, and in a few years, we'll probably be left with just a handful.

As for new acceleration technologies, I consider ideas that attempt to take advantage of the approximate nature of AI the most promising. With traditional technology, we are spending a lot of transistors to attempt to solve, very accurately, (for example, with 64-b precision) problems that are intrinsically very inaccurate. The world of deep learning has found that out, and it's pushing for new digital formats (8 and 16 bit) that greatly enhance the throughput without compromising accuracy. However, this is just the beginning, and there is a batch of new analog-inspired technology that attempts to use other forms of nondigital computation to reach similar accuracy. At Hewlett Packard Labs, we've been developing one of these (we call it the "dot product engine") using a ReRAM (memristor) crossbar for matrix multiply, and I'm aware of a few other efforts following a similar approach.

MATSUOKA: Acceleration is a means to an end; thus, the right question to ask would be, "What application areas will blossom toward the future the most, and do these areas need to be accelerated beyond the IDC and switched-capacitator technologies and infrastructures we have today?" The most important direction that I believe the IT infrastructure must achieve is ubiquitous intelligence, an intelligence that is far more than we see with today's AI technologies. It is debatable whether the extensions of the current AI technologies based on (engineered) deep learning will achieve the goal of ubiquitous intelligence or those based on computing paradigms that resemble brains to a greater degree (for example, neuromorphic computing) will become more dominant. There is ongoing research in both directions. In both cases, accelerating the processing of neural networks, plus the ability to move, digest, and generate massive data, will be subject to acceleration.

> We're at the peak of the hype curve for deep learning, machine learning, and AI, so it's a natural consequence that we're seeing an abundance of AI accelerators.

There are other areas that require HPC to be further accelerated beyond today's parallel processing and architectural customization. These areas may be subject to new computing paradigms, especially quantum computing. Unfortunately, due to the fundamental characteristics of quantum computing, especially its inability to digest large amounts of data, quantum will likely have a restricted utility, even if the so-called quantum supremacy were to be achieved. Quantum may become a niche accelerator for very narrow sets of problems, including quantum chemistry and cryptography.

MENDELSON: I define "accelerator" as any technology that helps to speed the performance of an application. Using this definition, I believe that the first "next-generation" technology that will impact massive computations, such as machine learning, will focus on better communication and memory subsystems.

Accelerating computations will come next since, in most of the cases, the acceleration of computing requires the development of new applications, languages, optimizations, and hardware features and materials. Changing so many components requires a long time, maybe even 10–15 years.

MILOJICIC: As you're thinking about new technologies, might it be possible to combine existing silicon-based technologies and new ones (neuromorphic) to produce something new?

FARABOSCHI: Definitely. One set of interesting technologies that can enable this combination falls into the category of copackaging, which some people call silicon-scale integration. For example, through silicon interposer layers, it's possible to integrate digital, analog, and optical mixed-signal components in the same package. Interfaces within the package can be very parallel (thousands of wires per channel), energy efficient (single-ended), and low latency (no need to serialize and deserialize or worry too much about signal integrity). For example, at Hewlett Packard Labs, we've been working on copackaged optics for almost a decade. Once that technology and the supply chain complexity are mastered, nothing prevents putting together other combinations, including neuro-inspired circuitry, when they become available.

MATSUOKA: Indeed, we will likely need materials to enable effective neuromorphic computing, since it is fundamentally analog in nature and, as such, not a good fit for CMOS, which is more appropriate for digital circuits. In that case, however, the major problem will be how to interface with the digital CMOS portions of the ASIC. Converting from analog to digital is expensive with respect to power. That problem may nullify any advantage that neuromorphic computing might have in power efficiency.

MENDELSON: New technologies are integrated as part of the SoC design. At that point, we consider them

as accelerators and not as a paradigm shift.

MILOJICIC: Let's look to the future. Where do you think the field be in 2025?

FARABOSCHI: The year 2025 is closer than people think. First, the advantage of new silicon processes will be much smaller than in the past, so the incentive to move to the leading-edge node will greatly diminish. Only massive-volume, high-value products will be able to afford the latest and greatest semiconductor node. By 2025, I expect the mainstream node of silicon technology to be 5 nm, with very few high-end designs pushing toward 3 nm. Using terminology that I believe Bob Colwell first introduced, I would characterize 2025 as the era of clean-up and specialization at the tail end of silicon scaling. So, essentially, there will be nothing truly novel, but there will be a more judicious use of different silicon processes and definitely a lot more specialization for all of the designs that can't afford jumping on the new node. I expect it will take another decade before we can see a truly novel paradigm that is not based on CMOS transistors.

MATSUOKA: I expect 2025 to be a transitional period when everybody admits that Moore's law is ending, and, moreover, some of the low-hanging fruits in achieving architectural efficiency will have been exploited, such as low-precision computing. Thus, the field really needs to start thinking about novel concepts for acceleration into the 2030s, when Moore's law is completely expired, and, hopefully, develop novel research and systems, in particular, to increase the system bandwidth to achieve continued acceleration, as well as novel architectural concepts for extreme heterogeneity.

MENDELSON: Five years is a long enough time to improve existing hardware components and optimize the new software stack that will use it, which may take another 10–15 years. Based on that, I assume that in 2025 we will have significantly more integrated devices that will use memory-over-logic and logic-over-logic (over memory) and make wider use of SoC technologies. Great emphasis will be given to the security-related aspects, and I hope that some of the new technologies will enable increasing the number of transistors on the die, better communication (for example, graphene), and new types of memories (such as memristors) that facilitate performing computations within the memory. I also assume a huge improvement in compiler technologies for heterogeneous systems and libraries that enables the better use of massive parallel systems, including heterogeneous architectures.

MILOJICIC: It's time to bring this virtual roundtable to a close. Each of you must have some final thoughts on the subject.

FARABOSCHI: I think we live in interesting times for computer architects, due to the increase in heterogeneity (some compare it to the "Cambrian explosion" of new species during the Paleozoic era). This, in turn, creates new needs in interconnects, memory systems, storage systems, and, of course, software. We can't expect developers to explicitly program for specialization, so either compilers, runtimes, or new frameworks will have to become a lot smarter to deal with variety. After decades of "boring" architectures, I find this quite exciting and a phenomenal opportunity to innovate at the hardware-software boundary.

MATSUOKA: Again, the biggest challenge is not only in hardware but how to maintain the ecosystem for software development. For bandwidth increases, we may use conventional programming, but the algorithm itself must be changed (say, from explicit methods that are FLOPS bound to implicit methods that are more bandwidth bound) as well as the software system to exploit the bandwidth because the bandwidth in the system will also be heterogeneously diverse. For heterogeneous architectures, this will be a big challenge, and methodologies whereby software and hardware generations are transparently handled will be necessary.

MENDELSON: We are facing a new revolution in the way we build systems. It will be based on multidiscipline software/hardware/application co-design and require out-of-the-box thinking. Thus, I expect that it will require 10–15 years before we start seeing systems that are based on new concepts. The main question is if the industry and the interest of academia will last that long before we can declare victory.

MILOJICIC: Thank you all.

DEJAN MILOJICIC is a distinguished technologist at Hewlett Packard Labs. His research interests include operating systems, distributed systems, and systems management. Milojicic received a Ph.D. from the University of Kaiserslautern, Germany. He received multiple best paper awards. He is a member of the editorial boards of *IEEE Internet Computing* and *IEEE Transactions on Cloud Computing*. He is general cochair of the IEEE infrastructure and Association for Computing Machinery (ACM) Middleware (industry track) conferences. He is a Fellow of the IEEE, an ACM distinguished technologist, and a member of Eta Kappa Nu and USENIX. Contact him at dejan.milojicic@hpe.com.