Future of HPC: Diversifying Heterogeneity

Dejan Milojicic
System Archtiecture Lab
Hewlett Packard Labs
Palo Alto, CA
dejan.milojicic@hpe.com

Paolo Faraboschi
AI Research Lab
Hewlett Packard Labs
Palo Alto, CA
paolo.faraboschi@hpe.com

Nicolas Dube
HPC CT Office
Hewlett Packard Enterprise
Quebec, Canada
nicolas.dube@hpe.com

Duncan Roweth
HPC CT Office
Hewlett Packard Enterprise
Bristol, UK
duncan.roweth@hpe.com

***Abstract*—*After the end of Dennard scaling and with the imminent end of Moore's Law, it has become challenging to continue scaling HPC systems within a given power envelope. This is exacerbated most in large systems, such as high end supercomputers. To alleviate this problem, general purpose is no longer sufficient, and HPC systems and components are being augmented with special-purpose hardware. By definition, because of the narrow applicability of specialization, broad supercomputing adoption requires using different heterogeneous components, each optimized for a specific application domain. In this paper, we discuss the impact of the introduced heterogeneity of specialization across the HPC stack: interconnects including memory models, accelerators including power and cooling, use cases and applications including AI, and delivery models, such as traditional, as-a-Service, and federated. We believe that a stack that supports diversification across hardware and software is required to continue scaling performance and maintaining energy efficiency.*

Keywords—High Performance Computing (HPC), Artificial intelligence (AI), interconnects, accelerators, delivery models, as-a-Service (aaS), heterogeneity, diversification).

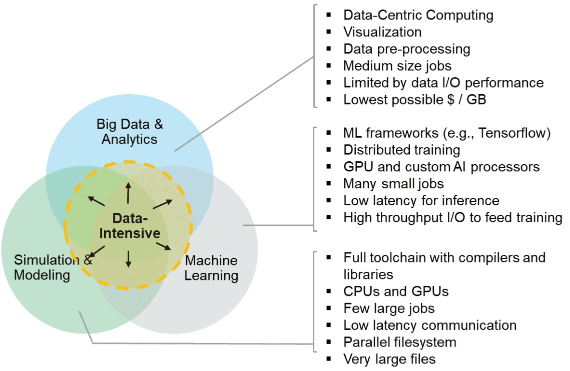
I. Introduction

During the last decade, high performance computing applications such as scientific discovery have moved into the "fourth paradigm" [1] of data-driven science (after the theoretical, experimental and computational paradigms). Instead of building models of ever increasing complexity, we can now develop new theories directly from the data coming out of experiments, simulations, and related sources. This convergence of Big Data, Artificial Intelligence (AI), and High-Performance Computing (HPC) provides a once in a generation opportunity to profoundly accelerate the way in which we advance science [2] [3].

After decades of steady gains driven by semiconductor process improvements, we have run out of the traditional means of increasing computational capacity. The HPC architecture of today, and the foreseeable future, will need to rely on specialization, such as custom accelerators. While the jury is still out on whether these accelerators will look like GPUs or something entirely different, we have entered the era of "mainstream heterogeneity".

The HPC computational platforms optimized for this new paradigm may be very different from traditional supercomputers [4], even more so because machine learning (ML) is expected to be an integral and essential component of the scientific discovery and poses very different requirements from the infrastructure. As Figure 1 shows, the convergence of data-intensive workloads and science demands a combination of HPC, analytics and ML technologies, deployed in distributed networks spanning the entire spectrum from the edge, to the supercomputer, and to the cloud.

Our main hypothesis and vision for the future of HPC is the need of diversified heterogeneity support across hardware, software, applications, CI/CD development tools, and delivery models. This kind of heterogeneous HPC infrastructure will be delivered in a federated manner with ad hoc bindings and "as a service (aaS)", both in terms of Edge-to-Supercomputer and multi-Cloud way.



- Data-Centric Computing
- Visualization
- Data pre-processing
- Medium size jobs
- Limited by data I/O performance
- Lowest possible $ / GB

- ML frameworks (e.g., Tensorflow)
- Distributed training
- GPU and custom AI processors
- Many small jobs
- Low latency for inference
- High throughput I/O to feed training

- Full toolchain with compilers and libraries
- CPUs and GPUs
- Few large jobs
- Low latency communication
- Parallel filesystem
- Very large files

Figure 1. The convergence of big data, HPC and AI

II. Background

A. Accelerators for the masses

Special purpose machines, including vector instruction sets and specialized hardware, were actually the mainstream of leadership-class supercomputers until the '90s. The rise of commodity hardware supercomputers, fuelled by Moore's law and the corresponding "Killer Micro" era, lasted from the early '90s until recently, and has been the dominant form of HPC. At the same time, the corresponding advances in software scaling to parallel machines has been remarkably successful. It's only been after the end of Dennard scaling (roughly 2005) that GPUs started their ascent into the HPC world. The trend is now accelerating, with more dedicated specialized hardware appearing at the horizon.

The computing industry beyond HPC is not new to the concept of dedicated hardware. At the edge, including mobile devices and embedded systems, accelerators have been the norm for at least the past two decades. While the recent generation of edge and consumer devices are adding significant more *hardware acceleration capabilities to address new functions,* such as AI, we see this as a continuation of an established trend. The "onion skin" structure of vertically integrated embedded software stacks lends itself well to accelerators, since only few developers are exposed to unconventional hardware structures, with most of the code written to higher-level APIs that abstract the hardware away.

In HPC, the software ecosystem is very different, more horizontal and based on large amounts of open source and home-grown code (in scientific HPC), or commercial ISV software (in industrial HPC). Because of the limited return-on-programming-investment (especially when non-portable) accelerator adoption in the past has only been limited to "high value" computation, such as financial services, oil & gas, or

978-3-9819263-5-4/DATE21/©2021 EDAA

drug discovery. That's where computational advantages translate into large financial returns and justify customization.

Driven by similar economics considerations, the first wave of AI acceleration was driven by the large cloud service providers, where any efficiency improvement gets multiplied by the traffic of billion users and the warehouse scale infrastructure needed to support them. However, with the improvement of software tools, the emergence of frameworks (such as TensorFlow or PyTorch) that abstract away many of the implementation details, and the increased deceleration of general-purpose hardware improvement, we are definitely entering the era of acceleration for the masses, and that is the direction we explore in this paper.

B. Mainstream Interconnect

Evolution of the HPC workload drove a new generation of system-wide interconnects to embrace Ethernet, bringing the features of proprietary HPC networks to an open, standard environment. The combination of performance and scalability with datacenter interoperability proved popular, with the adoption of Slingshot [5] by the US Exascale systems. These systems will run a diverse mix of traditional HPC applications and emerging hybrid HPC/AI applications. With this came a focus on sustained performance under load – with global bandwidth and tail latency the key metrics. Slingshot tackles congestion management at scale for the first time. It uses a novel flow-based approach in which congesting flows are identified and network hardware applies selective back pressure. Flow-based congestion management is the subject of active research in the wider Ethernet community from which standards are expected to emerge.

When it comes to local connectivity (within an individual server, or rack-scale collection of servers), PCIe has become ubiquitous for device connectivity including network devices, but bandwidth achieved on large transfers remains the priority. PCIe latencies are far too high for memory access and each of the CPU vendors is developing its own point-to-point interconnect, with efforts such as CCIX [6], OpenCAPI [7], Gen-Z [8] and CXL [9]. None of the existing standard so far has emerged as the clear winner. For example, Gen-Z provides a path to a standard low-latency interconnect [10], but lacks a direct connect to CPUs or GPUs. The recent accord between the CXL and Gen-Z consortia may offer a way forward, extending the scale of memory driven computing beyond the individual server. If the same interface is used to connect a high-speed network adapter, the latency savings can be extended to the system scale and open up new composable architecture that can more flexibly combine different computing elements with network interfaces.

Considering the physical layer, recent years have seen a convergence of standards, with network vendors and their suppliers adopting 4×56 Gbps PAM-4 (i.e., the current-generation 200 Gbps links) signaling and actively developing 4×112 Gbps (I.e., the next-generation 400 Gbps links). Increases in link speed have brought reductions in electrical reach and increased platform costs. Pressure to move to optical interconnect is increasing, but costs remain high.

Last, network topologies also have evolved to take into account the evolution of link technology and the economic tradeoffs of optical interconnects. For example, low-diameter networks such as dragonfly [11] and Hyper-X [12] provide a path to low system latency and high global bandwidth. State of the art switches (12.8 Tbps) combine high radix and high per-port bandwidth. Current designs have one more natural step (to 25.6 Tbps with 64 ports at 400 Gbps). These designs have a very high wire density, much of their area is taken up by SerDes, and they make only limited gains from improvements in process technology. Radical change is required beyond this point.

C. Delivery models

Traditional HPC is historically delivered on-premise, through a collection of commodity servers and specialized interconnects. Leadership-class supercomputers today run in dedicated data centers due to their extreme power and cooling requirements. For example, the exascale supercomputing generation is expected to require a 30-40 MW datacenter with aggressive liquid cooling and very high-density racks, up to 400 kW per rack. In addition, typical cloud interconnects create impediments to strong scaling (increasing parallelism for individual application), and only embarrassingly parallel applications with infrequent synchronization that are little impacted by noise were possible to execute in Cloud. These requirements are usually very difficult to meet in a general-purpose cloud datacenter, although some of the service providers are starting to make progress in this direction by offering HPC-optimized partitions.

To overcome the limitations of rigid static on premise supercomputers, the desire of a more flexible and dynamically growing HPC infrastructure has been an aspiration for several decades. There were several attempts to introduce computing federations, such as OpenCirrus [13] and Future Grid [14]. The whole concept of Grid Computing was also going in that direction [15]. The complications of managing service-level agreements (SLAs) and quality-of-service (QoS) were two of the major impediments to the success of Grid computing.

As cloud computing became more widely available more HPC applications were executed in the Cloud [16], except for synchronization-sensitive applications. The biggest issue for cloud computing to widen the HPC adoption is the built-in sharing of infrastructure and the interference of other applications running over the same interconnect, storage network and compute that creates noise and makes barrier-based synchronizations ineffective (the slowest component dictates performance). Only applications with complimentary resource utilization, that don't interfere with each other, can run without compromising each other's performance.

III. Key disruptive Technologies

A. Use cases and applications

As we described in the introduction, AI can greatly enhance HPC applications, so we expect it will be one of the key use case drivers for new technology development. However, commercial AI solutions and algorithms have been primarily developed by large cloud service providers to address the need of their consumers, for example in areas such as imaging, text, and natural language processing. These are typically not sufficient to address the needs of HPC applications, which will require new AI algorithms and tools. For example, HPC data sets tend to be sparse, bit-rich (large samples) and information-poor (largely unlabeled), and in many cases tightly constrained by the laws of the physical world. The use of AI in mission-critical applications must have a much stronger explainability basis, and correctness proofs, given the consequence of making wrong decisions.

This new generation of AI-enhanced HPC use cases is going to require a lot more than just compute power, and we expect that the first generation of post-exascale machines (roughly speaking, in the 2025 timeframe) is going to look very different from the exascale machines of 2021-2023.

For example, we see the world of HPC, and specifically scientific or industrial discovery, to start gradually expanding into the instrumentation "heavy edge". Complex instruments such as particle accelerators or light sources involve several electrical, mechanical, magnetic and fluidic subsystems, with several control points whose optimization quickly becomes an intractable problem. Today, all the instrumentation data goes back to the HPC core, but that has become a critical bottleneck, which is expected to get even worse with new generations of faster and more detailed experimental facilities. So, the next HPC frontier requires moving some elements of data analysis, and the related AI inference, close to the data source at the facility edge, together with sufficient computing, networking and data support, and well-coordinated with the supercomputing core.

Novel edge-to-supercomputer scenarios require technology breakthroughs in several areas. For example, real-time predictive analytics, control, and optimization is needed to minimize the need of a human-in-the-loop for operating the instrumentation edge. In areas such as smart energy grids and smart transportation, predictive models trained with infrastructure data will be key to be able to efficiently navigate the large design space. AI-enhanced cybersecurity algorithms will become necessary for detecting and diagnosing attacks in real-time.

Finally, AI results are only as good as the quality of the data used to train it. So, the creation of a common data foundation for AI will be the glue that ties together the intelligent HPC infrastructure of tomorrow. Well-defined foundational data protocols can accelerate innovation by providing actionable metadata and preserving important aspects such as lineage and provenance. A common edge-to-supercomputer data infrastructure will allow for rapid deployment of AI capabilities, while preserving security, interoperability and data governance.

B. Accelerators

A primary thesis of this paper is that future HPC workflows can be fundamentally enhanced by specialized hardware, specifically AI-optimized accelerators. The "first wave" of accelerators, which is just now entering the market, was designed to operate in tight coupling with general purpose computing. In other words, the accelerators all looked like PCIe cards attached to a standard server. We expect the "second wave" of accelerators to be substantially different in two fundamental ways. At the supercomputing core, new accelerators (mostly for training) are starting to look at lot more like large, standalone systems with their own networking and dedicated memory for the deep learning models. At the facility edge, new accelerators (for inference) will need to be lighter, power optimized, in some cases tightly integrated with sensors and instruments themselves, and design to operate in "hostile" environments across very aggressive temperature ranges, and even radiation in some cases.

The combination of these two types of accelerators will significantly improve HPC by enabling closed-loop combinations of classical simulation and deep-learning inference (to accelerate some simulation steps). It will also enable the simultaneous co-design and autonomous operation of the entire edge-to-supercomputer complex systems.

One of the key challenges we see in this "second wave" of accelerators is preserving the system balance to help programmers' productivity without requiring heroic (and non-portable) coding efforts for every new flavor of compute element. The extension to the edge also introduces a "wide-area networking" context that is foreign to the traditional HPC world and requires rethinking all the established rules-of-thumb, such as injection and bisection per FLOPS or arithmetic intensity rooflines. If our prediction is true, the HPC of the future will look at lot like an archipelago of tightly connected supercomputing islands, some containing combinations of very large accelerators and massive compute capabilities, some distributed at the edge with far less powerful computing, and all of them connected through a data foundation layer that keeps track of the workflow and the various data transformation steps.

If we zoom within the accelerator architecture, a few trends that are likely to remain dominant for the foreseeable future. Digital accelerators are squeezing the inefficiencies away from deep learning algorithms. They do that by reducing bit precision, exploiting in-situ on-chip memory, model sparsity by spatially mapping the neural networks into computing tiles, removing fetch-decode-execute overheads through dataflow and/or systolic computation (like Google's TPU [17]). Some ambitious designs (like Cerebras [18]) even take advantage of wafer-scale integration to further reduce the communication overhead, by widening the chiplet-to-chiplet paths that a notebook-sized piece of silicon enables. However, there is only so much you can do with digital logic, and we expect some amount of convergence in 1-2 generation of designs. We are already witnessing some features, like specialized reduced precision floating point formats and "tensor cores" capable of high-throughput matrix multiplication operations, becoming mainstream.

Alternatively, other accelerators take the "neuromorphic" approach, by exploiting physical properties that execute fast matrix multiplications in linear power and time complexity. These are interesting because they change an O(N2) problem to an O(N) problem. Analog "dot-product engines" exploit combination of Ohm and Kirchhoff laws in memory arrays to implement a dot product [19]. Similarly, optical engines exploit properties of coherent photonics to implement a matrix multiplication [20]. While these approaches are still early today, we expect them to mature in a few generations, and become formidable candidates for AI inference at the edge.

C. Interconnects

Accelerators are driving demand for bandwidth at all scales: on the device, across a package, within a server or throughout a system. While there will always be pressure to focus as much computation in as small a volume as possible, there will always be bigger problems and with them a cliff edge in performance when a problem *doesn't fit* and a scale out solution is required. Tighter integration of the accelerators and the network is required to mitigate this issue. Modern NICs are highly specialized, offloading the whole protocol stack for HPC programming environments or standards-based communication. While some vendors may integrate this function into their accelerators, this approach is at the expense of computational logic.

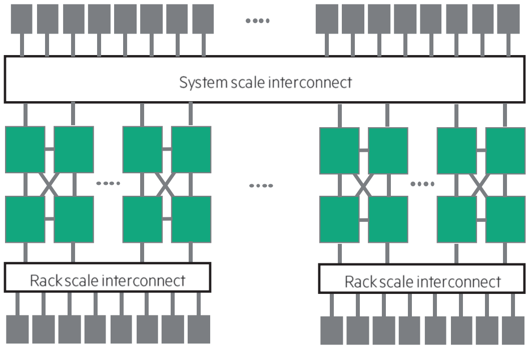


Figure 2. Interconnect at the device, rack and system scale

A more flexible approach is to provide bandwidth in a way that it can be divided between local, rack-scale and system-wide connectivity (Figure 2). The same physical interfaces, such as a future, low-latency implementation of CXL perhaps, can be used for both local connectivity amongst CPUs or accelerators, access to persistent memory, and connectivity to high bandwidth networks at the rack or system scale. The design separates persistent memory, the first storage tier, from processing. It ensures global accessibility for resilience and capacity, while maintaining low latency for local access. With this framework in place remote memory access and message passing can be offloaded efficiently to specialized network hardware as can complex communication patterns, the bulk-data all reduction operations used in training for example.

In a unified infrastructure, it must be possible to bring together any selection of processing and memory/storage resources – based on demand from the application or efficient use of the infrastructure. Key metrics for the network fabric are cost effective provision of global bandwidth and its ability to ensure performance under load.

High-radix switches provide the means for constructing large diameter 2 or diameter 3 networks but packaging them is increasingly costly as link speeds increase. Silicon photonics (SiPh) provides the means to bring bandwidth off the switch devices and directly into a low-cost optical network. An example of such key IP, being developed by Hewlett Packard Labs, is in integration of SiPh into the ASIC design workflow and CMOS manufacturing path. With this approach, it will be possible to take hundreds of fibres from each switch ASIC to optical-to-optical connectors at the card edge. A system fabric of essentially unlimited scale can be constructed from low-cost switches and passive optical cables.

The system will instantiate a virtual network for each application or workflow, a secure environment with strong service level guarantees that allows a heterogeneous mix of processing capabilities to be used together on solving a single problem. The network will protect itself from the tenants "zero trust" and isolate them from each other. Integration of strong encryption in the network with that in the CPUs will ensure that data can only be accessed by its owners and that they are confident of its authenticity.

D. Software and Programming Environments

Traditionally, there were only two programming models for HPC: Message passing, exemplified with MPI and multi-threaded, represented by a variety of shared memory models, (SHMEM and Partitioned Global Address Space (PGAS) as well as UPC and Chapel as languages to use shared memory).

Especially well-suited for distributed heterogenous architectures, data-centric runtime environments like Legion [21] are also rapidly emerging. They enable the programmer to embed the data structure to facilitate the extraction of task and data parallelism, and to map more easily to complex, multi-level, memory hierarchies.

With the imminent convergence of HPC and AI, Python-based frameworks might become this third option. The AI community has done a good job of hiding the diversity of heterogeneous hardware using programming environments, such as PyTorch, Caffe, or TensorFlow. Intermediate layers, such as ONNX, play an important interoperability role in hiding heterogeneity of both programming environments and the underlying hardware, for example by decoupling model training from model inference. When it comes to emerging accelerators, the impact to the programming model varies. For example, approaches, such as analog matrix-vector multiplications based on in-memory computation [22][23] map easily into existing programming environments and can be hidden within runtime implementations and model compilation to reduced precision arithmetic. For other less conventional approaches, such as Quantum Computing, new radically different programming models will be required. In this paper, we focus on the former class of accelerators.

In either of these programming models (message passing and shared memory multithreading), moving data across hierarchies of computation and memory/storage has a dominant cost in HPC computation. In addition, the size, and sparsity of models plays an important role in AI/deep learning computation. Due to the high cost of data movement, computing in memory has been revisited [24] and approaches to memory driven computing have been explored [25][26].

Some of the grand challenges for software, in particular AI models is in terms of explainability and balancing the degree of human in the loop—just enough to maintain control over some of the high-level decisions—not too much to maintain the sufficient automation. Explainability is crucial for any behavior analysis and auditing. As the AI-HPC integration progresses, explainability will increase in relevance. Other challenges include tighter integration of AI into HPC computing in terms of simulation and workflows on one hand and in terms of real-time runtimes. Neither is trivial as AIs, in particular areas were developed independently from HPC community. Tighter integration of the communities will need to take place to overcome this challenge.

In terms of operating system, while Linux has been widely adopted across data center and edge, the new standalone accelerators may open up new opportunities for the (real-time) kernels, AI engines, and CD/CI development tools better suited for the converged AI/HPC application domains [27].

E. Business Execution Challenges of Heterogeneity

Not so long ago, most system integrators were building HPC platforms for just one or two CPU architectures. As the programmability of GPUs improved beyond graphics, "GPGPUs" then emerged as a capable silicon alternative for a mix of HPC workloads and then accelerated the rebirth of machine learning. As of today, many CPU architectures are available, and multiple variants are available for every one of them (with different socket, TDP, co-packaged memory, and other characteristics.) GPUs can also be sourced from at least three major industry players. If we think of this in terms of supercomputer system design, already, we're looking at a combinatorial equation that demands the design, manufacturing and support of more than a dozen configurations. The system options further multiply when considering FPGAs, custom ASICs, and the "Cambrian explosion" of machine learning silicon.

In parallel, following the increased computational capability of the new silicon, the I/O interfaces are being driven at signaling rates pushing the boundaries of electronics. Low loss, high heat resistance, high-speed signaling multi-layer materials (like Megtron 6 [28]), are therefore becoming a de facto standard for system boards. Because of routing complexity, signal integrity challenges and timing requirements, any given platform enablement effort, can now easily reach a few million dollars in development cost.

These two pre-conditions are putting the industry in front of a difficult conundrum, where the silicon ecosystem is blooming but the ever more expensive system development process can really sustain fewer and fewer options. The system vendors, and especially the HPC community, need to align and enable a tectonic shift away from the way supercomputers have been built for the last few decades.

To offset the development costs of system boards, especially for low volume or early products, the industry should drive towards a standard for motherboards and other electronic sub-components. In this model, with similarities to the Open Compute Project [29], any system vendor could integrate a new silicon device and its off-the-shelf system board into their platform. The standard should allow for high-power devices, liquid-cooling options, custom management ASICs but ultimately all fit within the same mechanical and electrical specifications.

The default interface to the larger clustered system or supercomputer then becomes the network interface and the software stack. Any new silicon player could enter the market by having a system board built, that could then get integrated by all traditional system vendors and hyper-scalers. This new approach would lower the hurdle to new technology enablement and truly enable a diverse silicon ecosystem.

For system vendors, differentiation will come with higher level features that deliver the ease of use, data-security, a compelling runtime environment and the most competitive cost per workload. Ultimately, this model could lay the foundation to a hardware Dev/Ops model, where new silicon could get rolled in with minimum lift on the system side, and integration testing could get automated for as long as the silicon drivers meet the interfaces to the runtime.

F. Delivery models

In the future, we believe HPC systems will be inherently heterogeneous and distributed from edge to core. The interconnect fabric and the software stack, from low level drivers to the runtime environment, will have to support this radically new system architecture. Users will have their workloads run across a breadth of silicon options, ideally with a meta-scheduler that selects the best available for the job, but in a completely transparent manner to the applications.

HPC centers won't likely be able to procure and maintain the full breadth of computational options, therefore we argue that the community will enter a new era of "Federated HPC". Going back 20 years, "grid computing" proposed an approach to federate resources across sites and administrative domains. "The Grid" meant to integrate, virtualize, manage and share resources and services on the scale of distributed, heterogeneous and multi-institutional virtual organizations [15]. The philosophical foundation of the Grid initiative was to increase resources utilization and access to a broader set of systems through facilitated sharing between sites.

Following the data deluge fueled by more capable instruments at the edge, coupled to the breadth of compute options for processing and thanks to significantly more capable WAN interconnects, we believe the conditions are being set for a rebirth of the Grid. This is not without similarities to the new age of machine learning that followed the new silicon capable of computing large neural networks radically faster.

The conditions are set to take workload fluidity to an entire new level. The new framework will enable the analysis of data "gravitational" aspects, where workloads may not only be scheduled following compute resources availability but targeting the optimization of job completion time end to end, including the data transfer.

Next generation HPC will get delivered in a multi-cloud environment where a broad ensemble of users with diverse requirements will be able to compute and transfer data amongst an extensive set of resources. It will also put in place the monitoring and accounting framework to capture the resource exchange between the sites. Such resource consumption data collection could lay the foundation to an "Open Compute Exchange".

In many ways similar to existing commodity exchange (e.g., the Chicago Mercantile), an Open Compute Exchange would enable trading of resources between sites and users, providers and consumers, and would pave the way to a true commoditization of workflows.

For this model to be successful several conditions need to be met. Large established players need to provide the experience, reputation and financial backing to the exchange. In addition, the computing exchange needs a robust and scalable IT infrastructure, similar to other exchanges and electronic trading floors around the world.

Formally, the underlying economic model is nothing but a non-cooperative, zero-summed game, that eventually reaches equilibrium. Users and providers are encouraged to trade on the market, at their will, and at the price they feel inclined to buy for, or sell for, whatever that price may be, as the "market is always right". This approach thus paves the way to a more effective compute resources sharing system, that is otherwise a lot more liquid than if only supplied by a few service providers. In the end, this new sharing paradigm for compute resources instantiates an entire new economy; consumer and provider market orders strategies, third-party brokers, technology speculators and future HPC architectures risk hedging are only some of the possibilities that could now be envisioned.

G. Leading the Way to Democratized Compute and Data

Several intermediate steps are necessary to pave the way towards the envisioned compute exchange. For example, several HPC customers only need a bursting capability in its simplest form to deal with demand peaks or mismatch of workloads vs local resources. A second step could then build on the bursting functionality to enhance the software and hardware technology layers to make HPC workloads truly fluid between different sites under different administrations. Then, a bootstrapping stage for a “new compute grid” is what comes next, where the cross-institutional and geographical hurdles (such as security and data governance) are to be addressed. This vision culminates into an Open Compute Exchange, where anyone can contribute to the supply and demand equilibrium, setting the foundation to a new era of data and compute fluidity for HPC, and potentially to the benefit of the broader IT community (Figure 3).

IV. Comparison to Related Work

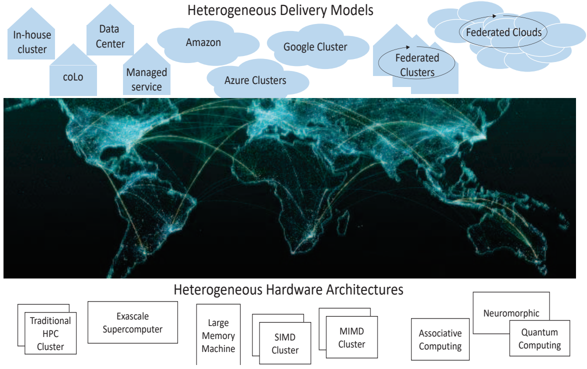


Figure 3. Types of HPC Computing and Delivery Models. Both exhibit substantial heterogeneity.

One of the early approaches similar to what we propose was envisioned by Grid Computing [15]. Despite large effort by the scientific community, it was never embraced by the IT industry outside of a few narrow domains. Instead, Cloud computing grew on a set of completely different premises, and achieved much broader adoption. Cloud computing brought elements of federation, especially in terms of connecting to traditional data centers, but was driven by a few large service providers who all had ramped up significant computing infrastructure for an adjacent business (such as search or e-commerce) and could leverage the economy of scale of that infrastructure. In the HPC domain, on premise data center deployments are still dominant in terms of investments but they are rapidly losing ground to Clouds, for applications that only have moderate performance requirements. Recently, to avoid the lock-in into any single Cloud service provider, federated models are again starting to appear.

This paper advocates horizontal and vertical federation. Horizontal federation is the distribution of applications across different service providers and on premise data centers, regulated by an open compute exchange that acts like other commodity markets. Vertical federation is the distribution of applications across the cloud, the supercomputing core, and the edge (e.g., the instrumentation edge). Horizontal federation is driven by economics, to optimize the infrastructure vs workload fluctuation. Vertical federation is driven by the data architecture and the need to optimize computing to avoid excessive data movement.

V. Summary

We presented a vision for HPC with diversified heterogeneity. This vision embraces heterogeneity across hardware, software, applications, CI/CD development tools, and delivery models. In addition, we believe that heterogeneity is going to push towards a federated HPC model that is different from today’s cloud service providers. The federation we envision comes with ad hoc bindings and as-a-service delivery, in a vertical (Edge-to-Supercomputer-to-cloud) and horizontal (multi-cloud and federated) manner and regulated through an open-exchange compute marketplace This approach will make it possible to continue scaling HPC across architectures, interconnects, and accelerators that no single site or user can afford on its own. Even more importantly, it will prevent any single vendor or provider lock in, but rather pave the way to democratizing HPC capabilities and delivery. HPC will benefit from leveraging standardized Cloud security interfaces while AI will accelerate simulations in HPC, enable use of GANs for synthetic data, improve imaging and many other applications. Selective federation will be a workaround for political road-blocks.

References

[1] T. Hey, S. Tansley, K. Tolle. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

[2] Y. Gil and B. Selman. A, “A 20-Year Community Roadmap for AI Research in the US,” CCC & AAAI, Aug 6, 2019.

[3] R. Stevens, J. Nichols, K. Yellick, “AI for Science,” DoE Report.

[4] T. Trader, J. Russell. AI is the next Exascale. Rick Stevens on What that Means and Why It’s Important. HPCwire 2019.

[5] https://www.hpe.com/us/en/compute/hpc/slingshot-interconnect.html

[6] https://www.ccixconsortium.com/

[7] https://opencapi.org/

[8] https://genzconsortium.org/

[9] https://www.computeexpresslink.org/

[10] Patrick Knebel , et al., “Gen-Z Chipsetfor Exascale Fabrics,” 2019 IEEE Hot Chips 31.

[11] J. Kim, W. J. Dally, S. Scott and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," 2008 ISCA, Beijing, 2008, pp. 77-88.

[12] J. H. Ahn, et al, "HyperX: topology, routing, and packaging of efficient large-scale networks," Proceedings Conference on High Performance Computing Networking, Storage and Analysis, Portland, OR, 2009

[13] Avetisyan, A., et al., “Open Cirrus A Global Cloud Computing Testbed,” IEEE Computer, vol 43, no 4, pp 42-50, April 2010.

[14] G. von Laszewski et al., "Design of the FutureGrid experiment management framework," 2010 GCE, New Orleans, 2010, pp. 1-10.

[15] Kesselman, Carl, Foster, I., “The Grid: Blueprint for a New Computing Infrastructure,” Morgan Kaufmann, Elsevier, 1999.

[16] Gupta, A.; et al. “Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud,” IEEE Transactions on Cloud Computing, vol 4, issue 3, 2016.

[17] N. Jouppi, et al., "In-datacenter performance analysis of a tensor processing unit," 2017 ACM/IEEE 44th ISCA, 2017 pp. 1-12.

[18] Sean Lie, Wafer Scale Deep Learning, Hot Chips 31, 2019).

[19] Miao Hu, et al.. 2016. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to acceleratematrix-vector multiplication. In DAC, 2016 53nd ACM/EDAC/IEEE.

[20] Carl Ramey, Silicon Photonics for Artificial Intelligence Acceleration. Hot Chips 32, 2020.

[21] Michael Bauer, et al., “:Legion: Expressing Locality and Independence with Logical Regions,” Proc. Supercomputing (SC 2012).

[22] Ankit, A., et al., “PANTHER: A Programmable Architecture for Neural Network Training Harnessing Energy-efficient ReRAM,” IEEE Transactions on Computers, , vol 69, issue 8, August 2020.

[23] Ankit, A., et al., “PUMA: A Programmable Ultra-efficient Memristor-based Acceleratorfor Machine Learning Inference” ACM ASPLOS’19.

[24] D. Milojicic, et al, "Computing In-Memory, Revisited," 2018 IEEE 38th (ICDCS), Vienna, 2018, pp. 1300-1309.

[25] Paolo Faraboschi, et al., “Beyond Processor-centric Operating Systems,” USENIX HotOS XV, May 2015.

[26] https://www.labs.hpe.com/memory-driven-computing

[27] Milojicic, D., Roscoe, T., “Operating Systems Outlook”, IEEE Computer (Cover Story), January 2016, vol 48, no 1.

[28] https://www.matrixelectronics.com/products/panasonic/megtron-6/

[29] https://www.opencompute.org/