Beyond Von Neumann in the Computing Continuum: Architectures, Applications, and Future Directions

Dragi Kimovski, *University of Klagenfurt, 9020, Klagenfurt, Austria*
Nishant Saurabh, *Utrecht University, 3584 CC, Utrecht, The Netherlands*
Matthijs Jansen, *Vrije Universiteit Amsterdam, 1081HV, Amsterdam, The Netherlands*
Atakan Aral, *Umeå University, 90187, Umeå, Sweden*
Auday Al-Dulaimy, *Mälardalen University, 721 23, Västerås, Sweden*
André B. Bondi, *Software Performance and Scalability Consulting LLC, Red Bank, NJ, 07701, USA*
Antonino Galletta, *University of Messina, 98122, Messina, Italy*
Alessandro V. Papadopoulos, *Mälardalen University, 721 23, Västerås, Sweden*
Alexandru Iosup, *Vrije Universiteit Amsterdam, 1081HV, Amsterdam, The Netherlands*
Radu Prodan, *University of Klagenfurt, 9020, Klagenfurt, Austria*

The article discusses emerging non-von Neumann computer architectures and their integration in the computing continuum for supporting modern distributed applications, including artificial intelligence, big data, and scientific computing. It provides a detailed summary of available and emerging non-von Neumann architectures, which range from power-efficient single-board accelerators to quantum and neuromorphic computers. Furthermore, it explores their potential benefits for revolutionizing data processing and analysis in various societal, science, and industry fields. The article provides a detailed analysis of the most widely used class of distributed applications and discusses the difficulties in their execution over the computing continuum, including communication, interoperability, orchestration, and sustainability issues.

Computing technologies have profoundly impacted society, fundamentally transforming how people communicate and interact with their environment. With the rise of the Internet of Things and the continuous evolution of modern communication technologies, digital integration has become an essential aspect of our lives. Despite these advancements, the fundamental principles of computer architecture have not changed since the introduction of John von Neumann's stored-program concept for the Institute for Advanced Study machine in 1952.

Therefore, the traditional computing architectures based primarily on the stored-program concept are colloquially referred to as *von Neumann architectures*. In von Neumann architectures, programs and data are stored in a single operating memory and treated as the same unit. When introduced, this novel idea allowed for simple hardware abstraction, making computer programming flexible and straightforward. However, physical separation of the processing units from memory through limited shared communication buses leads to increased communication latency and reduced throughput. The single communication bus between the processing unit and the shared instructions and data memory

Digital Object Identifier 10.1109/MIC.2023.3301010
Date of publication 3 August 2023; date of current version 3 June 2024.

causes the so-called memory wall problem, which limits processing efficiency. The memory wall appears due to the limited data transfer capacity and low transfer scalability between the processing unit and memory. As the communication bus can only access either the shared data or instruction memory at each point in time, for most of the application classes the data transfer rate is inherently lower than the rate at which the processing units work. Therefore, the processing units constantly wait for the data to be read or stored in the memory.

To mitigate this problem, most modern computers differ to some degree from von Neumann architectures by separating the data and instruction memory and introducing parallel data processing concepts, such as instruction pipelining and separate cache for data and instructions. Therefore, today's most common computer architectures are not strictly based on the von Neumann model; they provide the illusion of using a von Neumann programmer's model implemented over a separate execution pipeline and a modified memory management system.

As modern applications increasingly rely on data-intensive artificial intelligence (AI) algorithms, the limitations of traditional von Neumann architectures have become a critical barrier to further improvements in computational time, memory performance, and energy efficiency. Specifically, these AI algorithms require a large amount of fast memory, aggravating the memory wall bottleneck further. To illustrate this issue, consider the rapid evolution of the Generative Pre-trained Transformer (GPT) model. In just four years, the state-of-the-art GPT-2 model with 1.5 billion parameters has evolved to more than 175 billion parameters for GPT-3, and in March 2023, to more than 100 trillion parameters for GPT-4.[1] This highlights the urgent need for new computing architectures that support modern AI applications' demands regarding memory, low-latency computation, and sustainable execution.

To address the issues of memory, high latency, and energy usage, many novel computing architectures beyond the traditional von Neumann model have been proposed in recent years. These new architectures, known as *non-von Neumann architectures*, range from power-efficient single-board AI accelerators to quantum and neuromorphic computers. These novel architectures have great potential for revolutionizing how we process and analyze data, leading to significant advancements in health care, transportation, and entertainment.

Despite their potential benefits, it is difficult to integrate non-von Neumann architectures with the concurrent, well-established distributed computing paradigms, including cloud and edge computing and their amalgamation in the computing continuum.[2] Their integration in the computing continuum remains challenging due to significant architectural heterogeneity, data representation, communication, and instructions execution.

Therefore, in this work, we provide 1) a summary of the modern non-von Neumann architectures, 2) a classification of the most established application classes and their suitability for deployment on non-von Neumann architectures, and 3) a discussion of the main research directions toward overcoming the limiting factors for integrating non-von Neumann architectures in the computing continuum.

SUMMARY OF NON-VON NEUMANN ARCHITECTURES

This section describes a detailed summary of non-von Neumann architectures based on their architectural specifics. The differences among the presented architectures are summarized in Table 1, considering the computational models, processing paradigms, data representation and precision, and scalability.

We first define non-von Neumann architecture as *computer architecture that differs from traditional von Neumann, which relies on a single shared memory for instructions and data*. In non-von Neumann architectures, the memory model is usually distributed, divided into hierarchies, or even implements shared memory and processing components.

Classical Non-Von Neumann Architectures

Classical non-von Neumann architectures encompass various digital computers that rely on binary data representation. These architectures provide a simple programming and memory model that mimics the classical von Neumann shared memory abstraction. It is important to note that the following architectures can be implemented both as von Neumann and non-von Neumann, however, in the following, we refer to non-von Neumann implementations. We can generally categorize these architectures based on their field of application, namely, 1) general purpose and 2) application specific.

The most common architectures are general purpose, which are the core of various daily used computing devices. We can, therefore, highlight the following architectures, classified by the Instruction Set Architecture (ISA) and their memory organizations:

- RISC-V is a load-store ISA and core architecture provided under open source licenses. It can be implemented as either von Neumann or modified Harvard architectural[3] style, using separate

TABLE 1. Qualitative differences between classical von Neumann and nonclassical non-von Neumann architectures.

Architecture	Computational model	Processing paradigm	Representation and precision	Scalability
Von Neumann	Based on stored-program computer design	Sequential processing	Digital representation, high precision	Memory and processing bottlenecks
Classical non-von Neumann	Split memory	Parallel/concurrent processing	Digital representation, high precision	Memory and processing bottlenecks
Neuromorphic computing	Emulates biological neural networks	Highly parallel and event driven	Analog representation, lower precision	Networks with more than 100 million neurons
Quantum computing	Exploits principles of quantum mechanics	Quantum parallelism and entanglement	Quantum states, high precision (with errors and noise)	Dependent on the number of qubits (more than 1000 as of 2023)
Digital annealer	Exploits simulated annealing principles	Highly parallelized exploration of solution space	Binary representation, finite-adjustable precision	Scalable to large optimization problems (maximum problem scale of 100,000 bits)

level 1 (L1) caches for instructions and data. Therefore, multiple companies offer variations of RISC-V hardware together with open source operating systems, and the instruction set is supported in several popular software toolchains.

- ARM is one of the most popular intellectual property core architectures, based on the ARM load-store ISA. It is most commonly used for energy-efficient mobile computing devices, with more than 100 billion systems produced over the last five years.[4] Recently, version 8 of the ARM ISA was introduced, the first to be implemented as a non-von Neumann. It introduced the concept of separate instruction and data memory following the Harvard architectural style. The L1 cache is divided into instructions and data and supports larger level 2 (L2) and level 3 (L3) caches with a memory controller on the processor die.
- SPARC64 is a load-store microprocessor architecture introduced by Sun Microsystems.[5] Sparc follows the reduced Sparc V9 ISA. The latest SPARC64 XII processor architecture introduces a complex intercore communication network with superscalar implementation.
- x86 is the most widely used load-store ISA for personal computers. Although the initial ISA and corresponding implementations were entirely based on the von Neumann style (for example, the x86-16 variant), the recent implementations, starting even from the x86-32 variant, differ significantly and utilize a modified Harvard architecture. With its last extension, such as the x86-64, the architecture supports the level 0 (L0) operations cache.
- POWER is a RISC-based ISA developed by IBM. It has been widely used for designing superscalar multithreading processors for supercomputers and servers.[6] The latest POWER architecture iteration, POWER10, is an open source ISA and introduces asymmetrical instruction and data L1 caches.

In the last decade, many specialized non-von Neumann architectures have been introduced. Usually, these architectures are optimized for specific classes of applications, including machine learning (ML), video encoding, and real-time rendering. The architectures are combined as accelerators with classical von Neumann and non-von Neumann systems.

- AI accelerators, such as tensor processing units (TPUs), are domain-specific computer architectures designed to train deep neural networks with lower precision.[7] AI accelerators are usually modular and utilize a systolic array to interconnect multiple devices in complex 3-D topologies. In the case of TPUs, they include large 28–144-MB register files, specialized scratchpad memory, and a hierarchical set of cache memory.
- Stream multiprocessor architectures are implemented for general-purpose graphics processors. Although originally designed for processing

graphics, as their name implies, today, they are implemented almost as general purpose.[8] These architectures rely on the single instruction, multiple threads approach, which allows for limiting instruction-fetching overhead. The most common architectures using this paradigm are Nvidia Ampere and AMD RDNA3. These architectures use specific caches, usually three levels (from L1 to L3), per streaming multiprocessor (Ampere) or compute unit (RDNA2) to overcome the memory wall and limited processing performance.

Barriers to Integration in the Computing Continuum

Currently, von Neumann and classical non-von Neumann architectures represent the backbone of the computing continuum. This is due to their significant proliferation in the market, high availability, simple memory and programming models, and low purchasing price. However, due to their different architectures, network technologies and protocols, and various operating systems, there are still multiple barriers to their integration, including interoperability issues, application orchestration difficulties, and performance prediction instability, which we discuss further in the "Future Research Directions for Adopting Non-Von Neumann Architectures in the Computing Continuum" section.

Nonclassical Non-Von Neumann Architectures

Several factors limit classical computing architectures. These factors include power limitations, memory bandwidth, high heat dissipation, and nonsustainable manufacturing processes. Due to these limitations, academia and industry recently introduced numerous novel architectures based on technologies other than traditional semiconductors. These novel architectures explore significantly different approaches, such as using analog data representation or even concepts from quantum mechanics, to encode and process information.

- *Neuromorphic computing*[9] refers to architectures that imitate human neurobiological processes through massively parallelized electronic circuits. Neuromorphic hardware is not based on the von Neumann memory model. Instead, it comprises neurons and synapses responsible for both processing and memory. Input neurons are charged with incoming analog inputs (spikes) and eventually fire further spikes through the outgoing synapses, which in turn charge other neurons. The timing and strength of the spikes can be modulated via synaptic weights. Neuromorphic computing facilitates massively parallel event-driven processing as each neuron and synapse is independent, and spikes are asynchronous. Moreover, due to its event-driven design, a part of the hardware is inactive when the corresponding input signal is inactive. Considering sparse analog signals, this results in immense energy savings.
- Quantum computer architectures represent a novel approach to computing that utilizes phenomena from quantum mechanics, such as superposition and entanglement, to encode, store, and process information.[10] Therefore, the memory model in quantum computers is also represented using different quantum states. Furthermore, unlike classical binary states, quantum computers use qubits that can exist simultaneously in 0 and 1 states, enabling quantum computers to perform certain computations exponentially faster than classical computers.

One of the most widely used ISAs is the Quil architecture, which first introduced a shared quantum and von Neumann memory model. Quil utilizes an abstract quantum machine, which is similar to the classical Turing machine, but allows for practically solving real-world tasks. The Quil architecture provides a high-level programming language for quantum computing that allows for the description of quantum circuits and the integration of classical computing instructions.

- A digital annealer is a computing architecture inspired by quantum concepts such as superposition and entanglement.[11] Although the digital annealer is implemented using classical computing technologies, it uses a logical-level representation of information similar to that of quantum computers. The architecture can perform parallel, real-time combinatorial optimization calculations with much higher precision and scale than the classical non-von Neumann architectures. Although the digital annealer is not a quantum computer, it shares some of the advantages of quantum computing, such as the ability to evaluate many potential options simultaneously. Related to the memory model, the specificity of digital annealers is that their memory is nonvolatile. It is organized as a matrix for the input data storage, and an array representation is used to store the output, thus efficiently supporting combinatorial problems. Fujitsu is currently the only commercial provider

of digital annealer technology integrated within the DAU series computers.

Barriers to Integration in the Computing Continuum

Most of the nonclassical non-von Neumann architectures are still in the experimental or early industry-adaptation stage, which can lead to interoperability issues and the impossibility of porting or even adapting source codes between them. From a programmer's point of view, they all rely on different programming concepts, processing, and memory models, which require additional overhead. Moreover, they still lack mature-enough networking and resource-sharing possibilities, which are discussed in the "Future Research Directions for Adopting Non-Von Neumann Architectures in the Computing Continuum" section.

SUMMARY OF MODERN APPLICATION CLASSES

The non-von Neumann architecture in the computing continuum provides a suitable foundation, particularly for handling the computational requirements of data-intensive tasks, enabling efficient and effective execution of data-driven applications. The following sections explore the three most widespread classes of modern data-driven applications. They discuss their relationship to non-von Neumann architectures in terms of computational performance and fields of application. We also summarize the relationship between the application classes and their characteristic requirements, with classical non-von Neumann architectures listed in Table 2 and nonclassical non-von Neumann architectures listed in Table 3.

ML

In ML, models are trained to recognize features of input data, e.g., image classification and object recognition. They have been a key application for specialized architectures, such as stream multiprocessors, for years due to their increased processing power over general-purpose processing units.[12] To increase the processing speed of ML applications on stream processors, general 32- and 64-bit floating-point processing units are swapped for lower-precision 16- and 8-bit units, such as TPUs. These compute optimizations have shifted the performance bottleneck further to the data path, a traditional problem for von Neumann architectures. More radical non-von Neumann architectures, such as neuromorphic computing and quantum computing,[9] can revolutionize the way we process and store information for ML. The former class of architectures can process artificial neural networks highly efficiently. Notably, deep learning in the form of spiking neural networks[13] is particularly suitable for neuromorphic hardware. The latter class, i.e., quantum computing, can also offer significant speedup thanks to quantum parallelism.[14]

Scientific Computing

Scientific computing is applied in various domains, including linear algebra, molecular dynamics, material sciences, and drug design. Emerging non-von Neumann architectures can accelerate such scientific applications with high-end computational and performance requirements. Sparc and RISC-V processors can enable optimizing data movement and computation, minimize latency, and maximize throughput for complex numerical simulations. Similarly, TPUs can be used in scientific applications to accelerate matrix operations in deep learning and train large neural networks.

Nonclassical quantum-based non-von Neumann architectures are also useful in scientific applications, thanks to the proven theoretical speedup for different scientific problems and the native modeling of many scientific phenomena in quantum programs.[15] Quantum techniques such as quantum matrix inversion[16] can be used to solve linear systems of equations and perform linear algebra operations much faster than classical computers. Quantum computing has also shown its potential utility in the form of an accelerator and for modeling complex processes in molecular dynamics and material science. Examples include accelerating eigenvalue and Euclidean distance matrices calculations in target molecular dynamic workflows, modeling of quantum properties of microscopic particles, molecular simulation, and speeding up the discovery process in drug design.[17]

Big Data Analytics

Big data analytics involves collecting, storing, processing, and analyzing large volumes of data to extract valuable insights, identify patterns, and make data-driven decisions in various industries.

Big data applications can benefit from using non-von Neumann architectures for accelerated data analytics. Recently, novel concepts have been proposed for supporting big data analytics for robotic systems that utilize neuromorphic processors to support the decision-making process of robots while performing data-heavy analysis in the cloud. In general, neuromorphic computing is being explored to enhance big data analysis for time-sensitive operations, particularly in medicine and industry. Specifically, analog-based

TABLE 2. Summary of the identified application classes and their suitability for efficiently exploiting the available classical non-von Neumann architectures in the computing continuum.

Architecture → Application class↓	**x86-32/64**	**RISC-V**	**Sparc**	**ARM v9**	**Power**	**TPU**	**Ampere**
ML	X	X	X	X	X	X	X
Vectorization	AVX instruction set	RVV instruction set	VIS	SVE instruction set	VSX instruction set	MXU	Cuda/tensor cores
High parallelism	Hyperthreading	Multicore	Multicore	Multicore	Simultaneous multithreading	TPU cores	Multistreaming processors
Quantization	Low-precision SIMD	Low-precision SIMD	Low-precision SIMD	Low-precision SIMD	Low-precision SIMD	Low-precision SIMD	Low-precision SIMD
Scientific computing	X	X	X	X	X	X	X
Floating-point performance	High-performance floating-point units (FPUs)	Customizable FPU	Flexible	ARM SVE2 support	Wider vector units	Multiples of eight [memory operations (ops)], 128 (matrix ops) floating point	Strong
Mixed-precision computing	Single (32 bit), double (64 bit)	Flexible precision support	Single, double, extended (128 bit)	Single, half (16 bit), bfloat16 (16 bit)	Single, double	Single, bfloat16	Single, double, half, brain (BF16)
Big data analytics	X	X	X	X	X	X	X
Data-parallel operations	SIMD support	RVV vector extensions	Advanced vector processing	SIMD support	SIMD support tensor based	SIMD support (AVX-512)	
In-memory computing	Possible	Possible	Possible	Possible	Possible	High-bandwidth memory leverage	Possible
Data locality	Caching	Load/store instructions	Advanced caching	Advanced caching	Advanced caching	Advanced caching	Advanced caching

AVX: Advanced Vector Extensions; RVV: RISC-V Vector Extension; VIS: Visual Instruction Set; VSX: Vector Scalar Extension; MXU: matrix-multiply unit; SIMD: single instruction, multiple data.

TABLE 3. Summary of the identified application classes and their suitability for efficiently exploiting the available nonclassical non-von Neumann architectures in the computing continuum.

Architecture→ Application class↓	**Quantum/hybrid**	**Digital annealer**	**Neuromorphic**
ML	X	X	X
Vectorization	Qubits	DAU	SNN
High parallelism	Superposition	Array of qubits	Spike based
Quantization	—	—	—
Scientific computing	X	—	—
Floating-point performance	Quantum parallelism	—	—
Mixed-precision computing	—	—	—
Big data analytics	—	—	X
Data-parallel operations	—	—	SNN based
In-memory computing	—	—	Memory-compute coupling
Data locality	—	—	Localized memory access

analytics significantly reduces the computation complexity of big data applications by reducing algorithms' space and time dimensions. Furthermore, streaming multiprocessors and general-purpose graphical units have been widely used since the beginning of the 21st century, especially in high-performance computing centers, to enable fast analysis of big data streams.[18] In addition, classical and specialized architectures, such as Sparc, TPUs, and other application-specific matrix-based programmable logic, have been used to support big data analytics by optimizing hardware concerning the specifics of the input data streams.[19]

FUTURE RESEARCH DIRECTIONS FOR ADOPTING NON-VON NEUMANN ARCHITECTURES IN THE COMPUTING CONTINUUM

The following section discusses the barriers to adopting non-von Neumann architectures and possible future research directions.

Device Heterogeneity

The heterogeneity of non-von Neumann architectures hinders their transparent integration in the computing continuum. As described in the "Summary of Non-Von Neumann Architectures" section, due to the specific implementation of the most common non-von Neumann architectures, computational performance is highly affected by the exact technical implementation and the nature of the utilized algorithms. For example, neuromorphic is well suited for a class of ML applications, however, classical non-von Neumann architectures, such as general-purpose ARM and x86, can support a much larger set of applications, albeit with lower computational performance. Therefore, managing complex distributed applications over computing continuum infrastructures containing various non-von Neumann computing nodes requires rewriting the source code and a compilation for the set of suitable architectures. In practice, this could be difficult to achieve for a vast set of applications. Therefore, it is relevant to explore novel approaches to research for analyzing the similarities among the architectures in the computing continuum based on the programming and data models.

Communication and Data Movement

The unprecedented and sudden improvement in computational power at the network's edge due to the integration of non-von Neumann architectures might require a significant data transfer among the different systems. This can result in high latency and data transfer costs, particularly for real-time applications. Consequently, application providers might prefer task partitioning and utilization of both cloud and edge deployments to mitigate these issues. Furthermore, non-von Neumann architectures require adaptive communication protocols that can accommodate drastically different systems. This is particularly true for architectures capable of processing analog or mixed signals internally, including neuromorphic and quantum computers. The currently available protocols are static and use limited rules to enable communication. They are inefficient or even incapable of data transmissions in the form of spikes or qubits. The fragility of

such data, especially quantum information that environmental factors can easily disrupt, would also limit the communication range, at least in early implementations. Therefore, exploring novel adaptive protocols that utilize AI approaches to accommodate highly heterogeneous systems is an essential research track.

Interoperability

With von Neumann and non-von Neumann architectures offering highly variable performance for different application classes, different applications can be better optimized for a specific architecture, reducing the need for strict interoperability between architectures. However, with the variety in modern architectures increasing, application developers might still want to support multiple architectures in case users cannot access specific sparsely available resources. For example, although quantum computing supports ML very efficiently, the current scarcity of quantum computing machinery might push the application developers to support an x86 source to ensure that the application can be executed when no quantum computer is available, or the waiting times are long. Moreover, the optimal target architecture might differ depending on conditions; for example, with ML, deployment on a general-purpose ISA (ARM, x86, Sparc, or RISC-V) might be preferable to stream processors (RDNA and Ampere) because of their significant memory capacity, even though the computing capacity is generally superior for the given type of applications.

Furthermore, when fitting the computing continuum with various architectures, these devices need to be managed by shared software components such as resource managers and operating services, adding a need for interoperability. This shared management requires applications, their data, and possible isolation mechanisms (containers and virtual machines) to be uniform to allow management services to operate each application's lifecycle similarly, independent of the target architecture. The alternative would be to deploy management software for each specific architecture. However, this would remove any opportunity for interoperability, which is especially important for workloads consisting of multiple services deployed across multiple architectures.

Application Orchestration and Systems Adaptation

Applications scheduling and orchestration across the computing continuum is a process that requires identifying proper resources and considering the requirements of each application in terms of computational performance, available memory, and network bandwidth, among others. Many of the non-von Neumann computers have fundamentally different computational models, memory structures, and communication patterns. This results in challenges in developing efficient and effective scheduling algorithms to handle these systems' constraints and limitations. For instance, in a neuromorphic computer, data processing occurs massively parallel, with many neurons simultaneously computing on a large dataset. This makes it difficult to schedule tasks and allocate resources as there is no clear separation between computation and communication. In quantum computers, the scheduling problem is exacerbated by the probabilistic nature of quantum states and quantum gates, which requires careful consideration of the order and timing of operations to ensure that the computation is correct.

Furthermore, reliance of the orchestration approaches on real-time monitoring data aggravates the problem further. There is a multitude of monitoring platforms available, however, none supports the monitoring of quantum or neuromorphic computers. Additionally, defining unified monitoring metrics can be difficult.[10] For example, it is challenging to cross quantify the resource utilization rate for TPU devices, ARM devices, or digital annealers.

Related to monitoring, it is also essential to discuss how the computing continuum infrastructure can be adapted if the application's performance is insufficient. Multiple approaches for adaptation, such as the free energy principle and Markov blanket approach,[20] can be used to adapt the system based on monitoring data.

In conclusion, scheduling applications on non-von Neumann computers is a challenging task that requires specialized algorithms and monitoring approaches that can handle the unique characteristics of these systems. As these emerging computing paradigms become more prevalent, developing new scheduling techniques to utilize available resources effectively is essential.

Performance Predictability

The performance prediction of a system implemented on top of a non-von Neumann architecture requires understanding which phases can be executed in parallel and which can be executed serially This is a necessary step in performance prediction because of the heterogeneous nature of deployments.

Whether a phase that must be executed serially on a given architecture has components that might be parallelizable when deployed on different architectures depends on the nature of the algorithm within the

phase. When such possibilities have been identified, we wish to predict the performance of the sequential baseline and the various options for parallelization and deployment.

Consider a computation that can be decomposed into a set of serial phases. On a Gantt chart, each phase corresponds to a horizontal bar whose length represents the anticipated phase duration. If the phases must be executed one after the other, the Gantt chart will look like a staircase descending from left to right with steps of unequal lengths. We can visualize the performance benefits of different non-von Neumann architectures by looking at bars with reduced lengths. The benefits of decomposition into parallel phases are visualized by looking at how overlaps reduce the length of the span time from the leftmost endpoint to the rightmost endpoint.

The execution times could be calculated via mathematical models or obtained by running benchmarks of calculations on platforms with different architectures. This is especially desirable when the characteristics of the program or the platform make it difficult to build an analytic model of how the phase will behave. Predictions might be aided by using a library of benchmark programs compiled differently for different target platforms.

Sustainability

High power usage in computing caused by the von Neumann bottleneck is a significant sustainability issue, especially when accessing memory through low-bandwidth buses. Therefore, it is crucial to enhance energy efficiency in computing to ensure sustainability. Using specialized memory in non-von Neumann architectures eliminates the need for data transfer between memory and processing units. Their design to perform tasks in parallel reduces the time needed to complete tasks, thereby reducing energy consumption.

However, implementing non-von Neumann architectures in the computing continuum faces challenges, particularly at the edge layer, where devices have restricted compute electrical power and limited battery capacity compared to devices in the cloud. Even with non-von Neumann architectures, there are still technical challenges due to the heterogeneity of nodes in the continuum layers. For example, neuromorphic computers are very power efficient, however, due to their size and production cost, they cannot be distributed in large quantities at the network's edge. Therefore, other architectures with lower power requirements could be more suitable as a compromise, such as ARM or TPUs.

Despite these challenges, non-von Neumann architectures offer promising solutions for sustainable and energy-efficient nodes, especially at the edge layer. However, to make them feasible for devices in the continuum, there are various aspects to consider, such as integrating and implementing these architectures into complex continuum systems, optimizing their energy consumption, and addressing the technical issues associated with the heterogeneity of nodes. By overcoming these challenges, non-von Neumann architectures could significantly contribute to future computing sustainability.

CONCLUSION

This article provided a detailed summary of available and emerging non-von Neumann architectures and their specific characteristics regarding memory and computational management. We formulated the most commonly used application classes, exploring their particular characteristics and current efforts to execute them on non-von Neumann architectures. In addition, we discussed the current barriers to adopting the computing non-von Neumann architectures and explored new research tracks in terms of architecture heterogeneity, communication, interoperability, orchestration, performance prediction, and sustainability.

REFERENCES

1. K. Sanderson, "Gpt-4 is here: What scientists think," *Nature*, vol. 615, no. 7954, p. 773, Mar. 2023, doi: 10.1038/d41586-023-00816-5.
2. D. Kimovski, R. Mathá, J. Hammer, N. Mehran, H. Hellwagner, and R. Prodan, "Cloud, fog, or edge: Where to compute?" *IEEE Internet Comput.*, vol. 25, no. 4, pp. 30–36, Jul./Aug. 2021, doi: 10.1109/MIC.2021.3050613.
3. V. A. Konyavsky and G. V. Ross, "Secure computers of the new Harvard architecture," *Asia Life Sci.*, vol. 28, no. 1, pp. 33–53, 2019.
4. D. Seal, *ARM Architecture Reference Manual*. London, U.K.: Pearson Education, 2001.
5. T. Maruyama et al., "Sparc64 VIIIfx: A new-generation octocore processor for petascale computing," *IEEE Micro*, vol. 30, no. 2, pp. 30–40, Mar./Apr. 2010, doi: 10.1109/MM.2010.40.
6. W. J. Starke, B. W. Thompto, J. A. Stuecheli, and J. E. Moreira, "IBM'S power10 processor," *IEEE Micro*, vol. 41, no. 2, pp. 7–14, Mar./Apr. 2021, doi: 10.1109/MM.2021.3058632.
7. Y. E. Wang, G.-Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU platforms for deep learning," 2019, arXiv:1907.10701.
8. J. Pool, "Accelerating sparsity in the NVIDIA ampere architecture," *GTC*, 2020. [Online].

Available: https://developer.download.nvidia.com/video/gputechconf/gtc/2020/presentations/s22085-accelerating-sparsity-in-the-nvidia-ampere-architecture%E2%80%8B.pdf
9. C. D. Schuman et al., "Opportunities for neuromorphic computing algorithms and applications," *Nature Comput. Sci.*, vol. 2, no. 1, pp. 10–19, Jan. 2022, doi: 10.1038/s43588-021-00184-y.
10. A. Furutanpey et al., "Architectural vision for quantum computing in the edge-cloud continuum," 2023, arXiv:2305.05238.
11. S. Matsubara et al., "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in *Proc. 25th IEEE Asia South Pacific Des. Automat. Conf. (ASP-DAC)*, 2020, pp. 667–672, doi: 10.1109/ASP-DAC47756.2020.9045100.
12. P. Mo et al., "Accurate and efficient molecular dynamics based on machine learning and non von Neumann architecture," *NPJ Comput. Mater.*, vol. 8, no. 1, pp. 1–15, May 2022, doi: 10.1038/s41524-022-00773-z.
13. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019, doi: 10.1016/j.neunet.2018.12.002.
14. Y. Mezquita, R. S. Alonso, R. Casado-Vara, J. Prieto, and J. M. Corchado, "A review of k-NN algorithm based on classical and quantum machine learning," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.*, Cham, Switzerland: Springer, 2020, pp. 189–198.
15. W.-L. Chang, J.-C. Chen, W.-Y. Chung, C.-Y. Hsiao, R. Wong, and A. V. Vasilakos, "Quantum speedup and mathematical solutions of implementing bio-molecular solutions for the independent set problem on IBM quantum computers," *IEEE Trans. Nanobiosci.*, vol. 20, no. 3, pp. 354–376, Jul. 2021, doi: 10.1109/TNB.2021.3075733.
16. B. Duan, J. Yuan, Y. Liu, and D. Li, "Quantum algorithm for support matrix machines," *Phys. Rev. A*, vol. 96, no. 3, Sep. 2017, Art. no. 032301, doi: 10.1103/PhysRevA.96.032301.
17. S. S. Cranganore, V. D. Maio, I. Brandic, T. M. A. Do, and E. Deelman, "Molecular dynamics workflow decomposition for hybrid classic/quantum systems," in *Proc. 18th IEEE Int. Conf. e-Sci.*, Salt Lake City, UT, USA, Oct. 2022, pp. 346–356, doi: 10.1109/eScience55777.2022.00048.
18. A. N. Sisyukov, V. K. Bondarev, and O. S. Yulmetova, "ERP data analysis and visualization in high-performance computing environment," in *Proc. IEEE Conf. Russian Young Res. Elect. Electron. Eng. (ElConRus)*, 2020, pp. 509–512, doi: 10.1109/EIConRus49466.2020.9038949.
19. M. Kopczynski and T. Grzes, "FPGA supported rough set reduct calculation for big datasets," *J. Intell. Inf. Syst.*, vol. 59, no. 3, pp. 779–799, Jul. 2022, doi: 10.1007/s10844-022-00725-5.
20. S. Dustdar, V. C. Pujol, and P. K. Donta, "On distributed computing continuum systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4092–4105, Apr. 2023, doi: 10.1109/TKDE.2022.3142856.

DRAGI KIMOVSKI is a habilitated researcher on a tenure track at the University of Klagenfurt, 9020, Klagenfurt, Austria. His research interests include cloud and edge computing, distributed systems, and multi-objective optimization. Kimovski received his Ph.D. degree in computer science from the Technical University of Sofia, Bulgaria, and the Habilitation degree in distributed systems from the University of Klagenfurt, Austria. Contact him at dragi.kimovski@aau.at.

NISHANT SAURABH is an assistant professor at Utrecht University, 3584 CC, Utrecht, The Netherlands. His research interests include edge-cloud computing distributed infrastructures and performance modeling. Saurabh received his Ph.D. degree in computer science from the University of Innsbruck, Austria. He is a Member of IEEE. Contact him at n.saurabh@uu.nl.

MATTHIJS JANSEN is a Ph.D. student at the Vrije Universiteit Amsterdam, 1081HV, Amsterdam, The Netherlands. His research interests include resource management and scheduling for edge and cloud computing. Jansen received his master's degree in distributed deep learning from the Dutch supercomputing center SURFsara, The Netherlands. Contact him at m.s.jansen@vu.nl.

ATAKAN ARAL is an assistant professor at Umeå University, 90187, Umeå, Sweden, and a principal investigator at the University of Vienna, Vienna, Austria. His research interests include distributed systems, edge computing, resource allocation, fault tolerance, and edge artificial intellingence. Aral received his Ph.D. degree in computer engineering from Istanbul Technical University, Türkiye. He is a Senior Member of IEEE. Contact him at atakan.aral@univie.ac.at.>

AUDAY AL-DULAIMY is an assistant professor at Mälardalen University, 721 23, Västerås, and Dalarna University. His research interests include distributed systems, cloud computing, edge computing, and computing continuum. Al-Dulaimy received his Ph.D. degree in computer science from

Beirut Arab University, Lebanon. Contact him at auday.aldulaimy@gmail.com.

ANDRÉ B. BONDI is the president of Software Performance and Scalability Consulting LLC, Red Bank, NJ, 07701, USA, and an adjunct professor of software engineering at Stevens Institute of Technology, USA. His research interests include software performance engineering, governance, and processes; scalability, performance and functional requirements engineering, and performance testing. Bondi received his Ph.D. degree in computer science from Purdue University, USA. Contact him at andrebbondi@gmail.com.

ANTONINO GALLETTA is an assistant professor at the University of Messina, 98122, Messina, Italy. His research interests include security of cloud/edge/Internet of Things technologies for smart cities and eHealth solutions, including big data management and blockchain. Galletta received his Ph.D. degree in computer engineering from the University of Reggio Calabria, Italy. Contact him at antonino.galletta@unime.it.

ALESSANDRO V. PAPADOPOULOS is a professor at Mälardalen University, 721 23, Västerås, Sweden. His research interests include robotics, control theory, real-time systems, and autonomic computing. Papadopoulos received his Ph.D. degree in information technology from Politecnico di Milano, Milan, Italy. He is a Senior Member of IEEE. Contact him at alessandro.papadopoulos@mdu.se.

ALEXANDRU IOSUP is a professor at Vrije Universiteit Amsterdam, 1081HV, Amsterdam, The Netherlands. His research interests include massivizing computer systems, distributed systems, performance evaluation, cloud computing, big data, and computer ecosystems. Iosup received his Ph.D. degree in computer science from Delft University of Technology, the Netherlands. He is a Member of IEEE. Contact him at a.iosup@vu.nl.

RADU PRODAN is a professor at the University of Klagenfurt, 9020, Klagenfurt, Austria. His research interests include performance, optimization and resource management tools for parallel and distributed systems. Prodan received his Ph.D. degree in computer science from the Vienna University of Technology, Austria, and the Habilitation degree in distributed systems from the University of Innsbruck, Austria. Contact him at radu.prodan@aau.at.