

Review

# State of the Art in Parallel and Distributed Systems: Emerging Trends and Challenges

Fei Dai <sup>1,\*</sup>, Md Akbar Hossain <sup>1</sup> and Yi Wang <sup>2</sup>

<sup>1</sup> School of Computing, Eastern Institute of Technology, Napier 4104, New Zealand; ahossain@eit.ac.nz

<sup>2</sup> School of Mathematical and Computational Sciences, Massey University, Palmerston North 4410, New Zealand; y.wang24@massey.ac.nz

\* Correspondence: tdai@eit.ac.nz

**Abstract:** Driven by rapid advancements in interconnection, packaging, integration, and computing technologies, parallel and distributed systems have significantly evolved in recent years. These systems have become essential for addressing modern computational demands, offering enhanced processing power, scalability, and resource efficiency. This paper provides a comprehensive overview of parallel and distributed systems, exploring their interrelationships, their key distinctions, and the emerging trends shaping their evolution. We analyse four parallel computing paradigms—heterogeneous computing, quantum computing, neuromorphic computing, and optical computing—and examine emerging distributed systems such as blockchain, serverless computing, and cloud-native architectures. The associated challenges are highlighted, and potential future directions are outlined. This work serves as a valuable resource for researchers and practitioners aiming to stay informed about trends in parallel and distributed computing while understanding the challenges and future developments in the field.

**Keywords:** parallel computing; distributed systems; emerging trends; system challenges; future directions



Academic Editors: Djamel Djenouri and Domenico Rosaci

Received: 5 December 2024

Revised: 25 January 2025

Accepted: 6 February 2025

Published: 10 February 2025

**Citation:** Dai, F.; Hossain, M.A.; Wang, Y. State of the Art in Parallel and Distributed Systems: Emerging Trends and Challenges. *Electronics* **2025**, *14*, 677. <https://doi.org/10.3390/electronics14040677>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

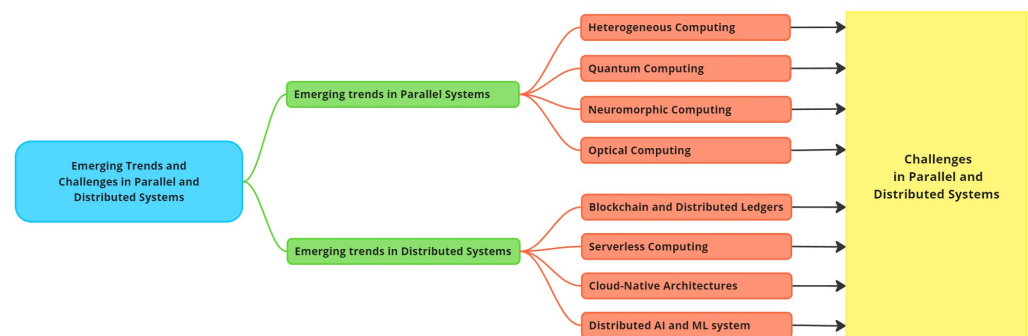
In the continually advancing field of computing, parallel and distributed systems have emerged as indispensable tools for addressing the escalating demands for computational power, scalability, and efficient resource utilisation. For instance, the rapid growth of artificial intelligence (AI) workloads has driven the need for computing systems capable of processing datasets exceeding petabyte scales, such as those required for training large language models like GPT-4, which involves hundreds of billions of parameters [1]. With advancements in interconnection networks, packaging technologies, system integration, and computational architectures, these systems have demonstrated remarkable improvements in performance, enabling the management of increasingly large-scale and complex workloads [2]. By facilitating the concurrent execution of tasks across multiple processors and nodes, parallel and distributed systems underpin modern solutions to critical computational challenges, including big data analytics, AI, real-time simulations, and cloud-based services.

The significance of parallel and distributed systems extends beyond their computational capabilities, as they play a pivotal role in driving innovation across various industries. For example, in high-performance computing (HPC), these systems enable climate modelling [3] and molecular dynamics simulations [4], while distributed architectures power applications like global-scale content delivery networks [5] and decentralised finance [6]. A

recent study indicates that distributed systems in finance have great potential to improve processing speeds for decentralised applications [7]. However, these benefits come with significant challenges, including scalability, security, interoperability, fault tolerance, legal compliance, and the integration of diverse and heterogeneous resources [7]. Addressing these challenges is essential for ensuring the sustained evolution and utility of parallel and distributed systems.

Despite their critical importance, many existing reviews of parallel and distributed systems either focus narrowly on specific aspects or lack comprehensive analyses of their historical development, emerging trends, and future challenges. This paper aims to bridge this gap by providing a holistic overview of these systems and exploring their evolution, interrelationships, and distinctions. Furthermore, it examines key challenges associated with parallel and distributed systems and proposes actionable future research directions to guide the field's continued advancement.

The organisation of this paper is illustrated in Figure 1, which provides a clear roadmap of the topics discussed. Section 2 defines parallel and distributed systems, introduces key categories, and explores their interrelationships and distinctions. Section 3 examines emerging trends in parallel systems, focusing on heterogeneous computing, quantum computing, neuromorphic computing, and optical computing. Section 4 explores emerging trends in distributed systems, highlighting blockchain and distributed ledgers, serverless computing, cloud-native architectures, and distributed AI and machine learning (ML) systems. Section 5 discusses the primary challenges facing these systems, providing specific metrics and real-world examples. Section 6 outlines actionable future research directions to address these challenges. Finally, Section 7 concludes this paper.



**Figure 1.** Logical overview of this paper's structure. This figure illustrates the organisation of sections, their interdependencies, and the logical progression of topics in this review.

## 2. Overview of Parallel and Distributed Systems

This section defines parallel and distributed systems, introduces various categories and common architectures, and explores their relationships and synergies. This foundational understanding sets the stage for a deeper examination of their historical context, key concepts, and terminologies.

### 2.1. Parallel Systems

Parallel systems are computational architectures designed to execute multiple tasks simultaneously by dividing computations into smaller sub-tasks processed concurrently across multiple processors or cores within a single machine or a closely connected cluster [8]. Their primary objective is to reduce computation time and improve performance efficiency, with applications in scientific simulations, image processing, and large-scale data analysis [9]. Key features of parallel systems include concurrency, coordination among processors, and efficient utilisation of shared resources.

Traditional parallel systems can be categorised into three main types: *Shared Memory Systems*, where multiple processors share a common memory space, allowing for direct communication through shared variables—examples include multi-core processors and symmetric multiprocessors (SMPs); *Distributed Memory Systems*, in which each processor has its own private memory and communicates with others by passing messages—examples include cluster computing and massively parallel processing (MPP) systems; and *Hybrid Systems*, which combine shared and distributed memory approaches, often seen in modern supercomputers and HPC clusters to leverage the advantages of both architectures. Common architectures include *Central Processing Units (CPUs)*, found in everyday devices like laptops and smartphones, enabling parallel task execution to improve performance and efficiency; *General-Purpose Graphics Processing Units (GPGPUs)*, used in gaming, video rendering, and AI applications to perform massive parallel computations; *Application-Specific Integrated Circuits (ASICs)*, custom-designed hardware optimised for specific applications such as cryptocurrency mining and specialised AI algorithms, providing high performance and energy efficiency; and *Field-Programmable Gate Arrays (FPGAs)*, which are reconfigurable silicon devices that can be electrically programmed to implement various digital circuits or systems [10], commonly used in scientific research, aerospace, and defence.

The origins of parallel computing can be traced back to the late 1950s with the advent of vector processors and early supercomputers like the IBM Stretch [11] and the CDC 6600 [12]. Significant advancements occurred in the 1980s with the introduction of MPP systems [13], including the Connection Machine [14] and the Cray series [15]. These systems utilised thousands of processors to perform simultaneous computations, paving the way for modern parallel architectures. In the 1990s and 2000s, the development of multi-core processors [16] and GPGPUs [17] revolutionised parallel computing by making it more accessible and efficient. The rise of ML, big data, and deep learning advancements led to a surge in demand for high-performance parallel processing hardware. However, traditional parallel hardware began to show limitations in providing the necessary processing capacity for AI training. Challenges such as insufficient interconnection bandwidth between cores and processors and the “memory wall” problem—where memory bandwidth cannot keep up with processing speed—became critical bottlenecks. To address these challenges, scientists and engineers have been developing innovative parallel computing systems tailored for AI and other demanding applications. Recent innovations, including heterogeneous computing, quantum computing, neuromorphic systems, and optical computing, aim to address these limitations, as discussed in Section 3.

## 2.2. Distributed Systems

Distributed systems are computational architectures where multiple autonomous computing nodes, often geographically separated, collaborate to achieve a common objective [18]. These nodes communicate and coordinate their actions by passing messages over a network [19]. Distributed systems emphasise fault tolerance, scalability, and resource sharing, making them essential for various applications, including cloud computing, distributed databases, and blockchain networks. Key features of distributed systems include the ability to handle node failures gracefully, scale out by adding more nodes, and efficiently manage distributed resources.

Distributed systems can be categorised into several types: *Client–Server Systems*, where clients request services and resources from centralised servers—examples include web applications and enterprise software; *Peer-to-Peer (P2P) Systems*, in which nodes act as both clients and servers, sharing resources directly without centralised control—examples include file-sharing networks and blockchain platforms; *Cloud Computing Systems*, which provide scalable and flexible resources over the Internet—examples include Amazon Web

Services (AWS) and Google Cloud Platform (GCP); and *Edge Computing Systems*, which process data near the source of generation to reduce latency and bandwidth usage—examples include Internet of Things (IoT) devices and real-time analytics systems. Common architectures in distributed systems include the *Client–Server Model*, used in web services where web browsers (clients) communicate with web servers to fetch and display content; *Cloud Infrastructure*, utilised for on-demand resource provisioning, hosting applications, and data storage, as seen in platforms like AWS and GCP; and *IoT Networks*, which connect various smart devices, enabling them to communicate and perform tasks collaboratively in real-time.

The concept of distributed systems emerged in the 1970s with the development of ARPANET, the precursor to the modern Internet [20]. Early distributed systems focused on resource sharing and remote access to computational power. The 1980s and 1990s witnessed the growth of distributed databases [21] and the Client–Server Model [22], which became fundamental in enterprise computing. The 2000s marked the rise of cloud computing and big data, epitomising the distributed system paradigm by providing scalable, on-demand computing resources over the Internet [23]. Technologies like Hadoop and MapReduce [24] enhanced the capability to process large datasets in a distributed manner. More recently, edge computing [25] and the IoT [26] have extended the reach of distributed systems to the periphery of networks, enabling real-time processing and decision-making at the edge. The development of digital cryptocurrencies and advancements in AI have further propelled the growth of distributed systems. In this paper, we focus on emerging trends such as blockchain and distributed ledgers, serverless computing, cloud-native architectures, and distributed AI and ML systems, which will be explored in Section 4.

### 2.3. Relationship and Synergy Between Parallel and Distributed Systems

Parallel and distributed systems are integral to modern computing, each contributing to efficiently executing large-scale and complex tasks. While they serve distinct purposes, their relationship is characterised by complementary roles and overlapping functionalities. Parallel systems are designed to maximise computational speed within a single machine or tightly coupled cluster [27]. By dividing a large task into smaller sub-tasks and processing them simultaneously across multiple processors, parallel systems achieve significant reductions in computation time. This makes them ideal for HPC applications like AI training and real-time data processing. Distributed systems, on the other hand, are engineered to leverage multiple autonomous nodes that collaborate over a network to achieve a common goal. This architecture prioritises scalability, fault tolerance, and resource sharing, making distributed systems suitable for applications that require robust, scalable, and reliable infrastructure, such as cloud computing and distributed databases.

In some scenarios, parallel and distributed systems can overlap, creating hybrid systems that combine the strengths of both architectures. For instance, a distributed system might employ parallel processing within individual nodes to further enhance performance. Conversely, a parallel system might distribute tasks across closely connected clusters, incorporating distributed computing elements. Both parallel and distributed systems aim to improve computational efficiency and handle large-scale problems, but they do so with different focuses and methods. The primary distinction between parallel and distributed systems lies in their architecture and operational focus:

- *Architecture:* Parallel systems use multiple processors or cores within a single machine or a closely connected cluster to perform concurrent computations [8]. Distributed systems, on the other hand, involve multiple independent machines that communicate over a network [19].

- *Coordination and communication:* In parallel systems, communication between processors is typically fast and direct due to their close proximity. Distributed systems require communication over potentially large distances, often leading to higher latency and the need for sophisticated communication protocols.
- *Scalability and fault tolerance:* Distributed systems are designed to scale out by adding more nodes and are built with fault tolerance in mind [28], allowing them to continue functioning even if some nodes fail. Parallel systems focus on scaling up by adding more processors to a single machine [29], with fault tolerance often a secondary consideration.
- *Resource sharing:* Distributed systems emphasise resource sharing and collaboration among independent nodes, each potentially equipped with its own local resources, such as distributed memory. Parallel systems concentrate resources within a single system, focusing on components like cache systems to enhance computational power.

Understanding the relationship and differences between parallel and distributed systems is crucial for engineers, researchers, and students as they explore the diverse applications and challenges within these fields. Both systems play vital roles in advancing computational capabilities and addressing the demands of modern technology.

### 3. Emerging Trends in Parallel Systems

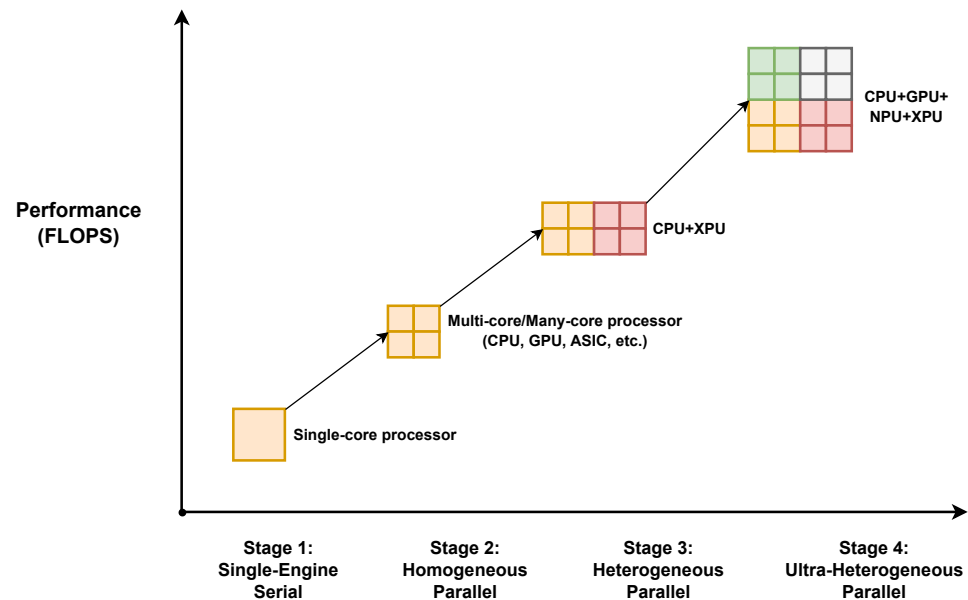
The development of parallel systems has primarily followed two main directions: enhancing existing computing architectures and creating new parallel architectures to adapt to new applications, such as ML. Industry leaders like Intel, AMD, and NVIDIA exemplify this trend by producing new products based on advanced architectures annually, targeting general tasks, servers, AI training, etc. The rapid development of deep learning has spurred the proposal of many innovative architectures, such as near-memory computing architecture, heterogeneous computing architecture, quantum computing architecture, neuromorphic computing architecture, and optical computing architecture, aimed at overcoming the memory wall of traditional Von Neumann architecture [30]. In response to the increasing volumes of processing data and advancements in AI, we explore the emerging trends in parallel systems across four key areas: heterogeneous computing, quantum computing, neuromorphic computing, and optical computing.

#### 3.1. Heterogeneous Computing

Heterogeneous computing integrates different types of processors and specialised computing units to work together, leveraging their unique strengths to enhance overall system performance and efficiency. As new architectures are proposed and technological advancements continue, heterogeneous computing continues to evolve. To explore the emerging trend of heterogeneous computing within parallel systems, we first examine the evolution of computing and then focus on advanced ultra-heterogeneous computing (UHC). Specifically, we discuss the software and hardware architectures that support UHC and provide an outlook on its future developments.

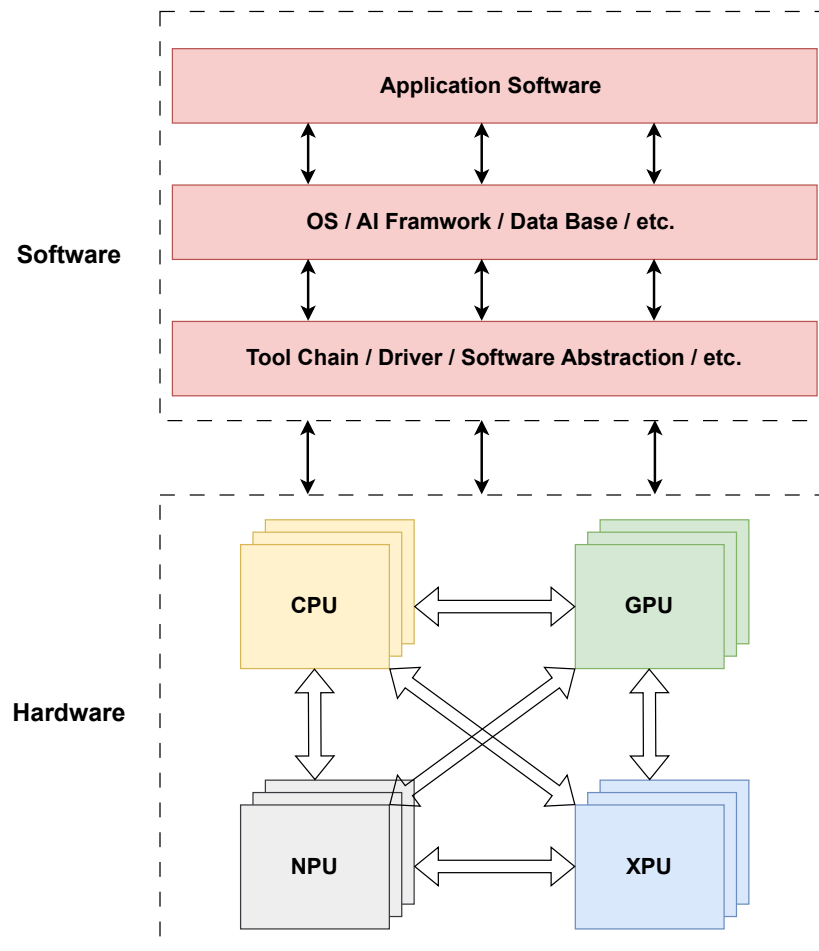
Figure 2 outlines the evolution of computing, beginning with single-engine serial processing followed by homogeneous computing then heterogeneous computing, and culminating in UHC. The evolution of heterogeneous computing can be described in four stages. In the first stage, a single processor handles all computational tasks sequentially, limiting performance to the capabilities of a single processing unit. As the demand for higher performance grew, this led to the second stage, which marked the introduction of homogeneous parallel processing. Here, multiple cores of the same type, such as multi-core CPUs or ASICs, work together to perform tasks in parallel. This approach improves performance by distributing workloads across several identical processors. However, the

need to optimise diverse tasks pushed the transition to the third stage: heterogeneous computing. In this stage, two types of processors, such as CPUs and GPUs, are combined to handle various computational tasks more effectively, with each processor type optimised for specific operations, thereby enhancing overall efficiency. Finally, as applications became more complex and diverse, the necessity to maximise computational efficiency and performance led to the final stage: UHC. This stage integrates multiple types of processors, such as CPUs, GPUs, neural processing units (NPUs), and data processing units (DPUs), combining their specialised strengths to address complex computational needs.



**Figure 2.** Evolution of various computing eras. This figure outlines the evolution of computing, from single-engine serial processing to ultra-heterogeneous parallel processing, highlighting key stages in this transformation. The different colours in the squares represent various processor types utilized in each stage.

With the development of technology, we are entering the early stages of UHC, which promises higher performance than in previous eras. For instance, systems integrating CPUs, GPUs, and DPUs have already demonstrated significant improvements in handling various AI tasks [31]. However, such systems rely on the support of both software and hardware. Figure 3 illustrates the software and hardware layers required for UHC systems. The software layer is responsible for effectively managing and optimising diverse processing units. Software frameworks support seamless communication and coordination between different types of processors, allowing tasks to be dynamically assigned to the most suitable processing unit. Advancements in frameworks like CUDA and OpenCL have significantly enhanced interoperability and workload allocation across processors, enabling efficient dynamic task management [32]. This involves developing sophisticated schedulers, resource managers, and communication protocols that can handle the complexities of UHC environments. Additionally, programming models and languages (e.g., CUDA, OpenCL, OpenMP, MPI, etc.) must evolve to provide abstractions that simplify the development of applications for UHC systems, enabling developers to leverage the full potential of diverse computing resources without needing to manage low-level hardware details [33].



**Figure 3.** Hardware and software layers of UHC. This figure depicts the essential software and hardware components required for UHC systems, emphasising interoperability and workload distribution.

The hardware architectures for UHC integrate multiple processing units into a cohesive system. This involves designing interconnects that provide high-bandwidth, low-latency communication between CPUs, GPUs, NPUs, DPUs, and other specialised processors. Memory architectures will also evolve to support efficient data sharing and movement between different processing units, minimising bottlenecks and maximising throughput. Innovations like 3D stacking and advanced co-packaging technologies play a pivotal role in enabling UHC systems by reducing communication delays and improving system performance [34].

The future of UHC is promising, with potential applications spanning various fields, including AI, scientific computing, and real-time data processing. As demand grows for more powerful and efficient systems, UHC architectures are poised to become increasingly prevalent. Advances in both technological infrastructure and development frameworks will be instrumental in driving this evolution, facilitating systems that seamlessly integrate diverse processing units to deliver unparalleled performance and efficiency.

### 3.2. Quantum Computing

Quantum computing represents a significant departure from classical computing paradigms, utilising the principles of quantum mechanics to perform computations. Unlike classical computers that process information as binary bits (0's and 1's), quantum computers leverage quantum bits (qubits), which can exist in multiple states simultaneously due to the phenomenon of superposition. This enables quantum computers to process vast

amounts of information in parallel, making them particularly powerful for certain types of computations. Quantum computing research began in the 1980s [35]. Although its initial development was slow due to technological barriers, it has accelerated rapidly in recent decades with the scaling up of qubit numbers in superconducting systems [36]. To explore the emerging trends in quantum computing, we start by discussing quantum computers and their applications, followed by an explanation of the different types of qubits and their development trends. Finally, we conclude with an overview of the current state and future prospects of quantum computing.

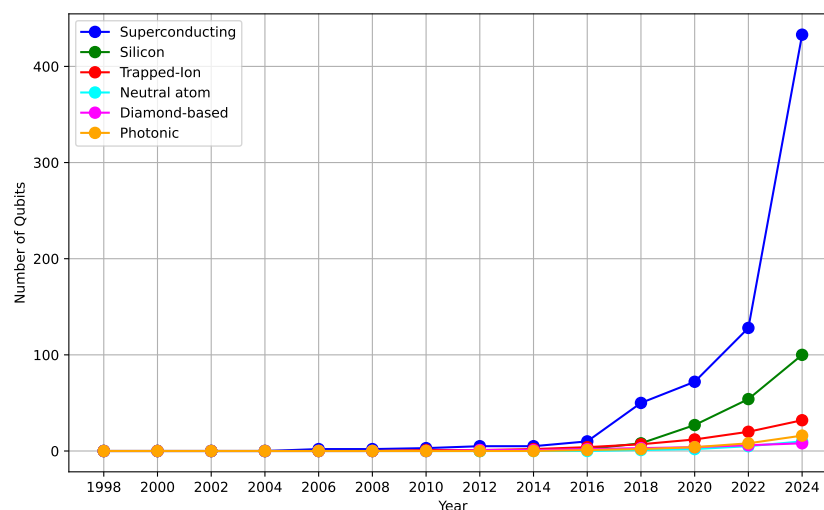
Quantum computers leverage qubits, which can exist in multiple states simultaneously (superposition) and be entangled with one another, enabling exponential increases in computational power for certain types of problems [37]. To illustrate superposition, consider a coin spinning in the air: unlike a classical bit that is either heads or tails, a qubit remains in a combination of both states until measured. Similarly, entanglement can be visualised as a pair of dice that always show the same number, regardless of their distance from each other. Despite these advantages, qubits are highly sensitive to environmental noise and interactions, leading to stability issues and significant error rates. These limitations present a major challenge to the development of practical quantum systems, as maintaining coherence and minimising errors often require complex error correction protocols and cryogenic environments.

Quantum gates are designed to manipulate the coefficients of basis states, performing general functions akin to logic gates in traditional computing systems [38]. Another essential concept, quantum interference, allows quantum algorithms to amplify correct solutions while cancelling out incorrect ones, significantly improving computational efficiency. Quantum algorithms specifically exploit the principles of superposition, entanglement, and quantum interference to execute computations more efficiently than classical computers [39]. Building on these unique properties, quantum computing holds promise for solving complex problems currently intractable for classical computers, such as large-scale optimisation, cryptography, and quantum physical system simulation [40]. Major technology companies and research institutions are heavily investing in quantum computing research, driving rapid advancements in practical quantum computers and efficient quantum algorithms.

There are various physical systems to realise qubits, each offering distinct advantages and contributing to the overall progress in quantum computing. Superconducting qubits utilise superconducting circuits and are among the most mature technologies in this domain [36]. However, they require extremely low temperatures, increasing operational complexity and cost. Silicon qubits, based on semiconductor technology similar to classical computer chips [41], offer compatibility with existing fabrication techniques but face scalability challenges, as quantum coherence deteriorates with size. Trapped-ion qubits use ions trapped in electromagnetic fields and manipulated with lasers [42], known for their high fidelity, but their operations are inherently slower, posing limitations for large-scale computations. Neutral atom qubits employ neutral atoms trapped in optical lattices [43], facilitating scalable quantum computing, yet achieving consistent trapping and manipulation across large arrays remains challenging. Diamond-based qubits utilise nitrogen-vacancy centres in diamonds [44], which can be manipulated at room temperature but often suffer from low qubit density and complex fabrication. Photonic qubits use photons to encode quantum information [45], providing advantages in communication due to their speed and low loss, but their integration into computational frameworks and achieving scalable photonic processors remain significant hurdles.

The current state of quantum computing demonstrates a promising trajectory, with continuous advancements in qubit technology and quantum algorithms. Despite earlier bottlenecks in qubit stability, fidelity, and scalability, ongoing research has successfully

addressed many of these issues, enabling steady progress in increasing qubit numbers. As depicted in Figure 4, the number of qubits in quantum processors has been steadily increasing across different technologies. IBM's roadmap outlines plans to scale its Flamingo systems to 1000 qubits by 2027 and deliver quantum-centric supercomputers with thousands of logical qubits by 2030 and beyond [46]. This trend highlights quantum computing's potential to revolutionise fields requiring immense computational power, such as materials science, AI, and high-energy physics.

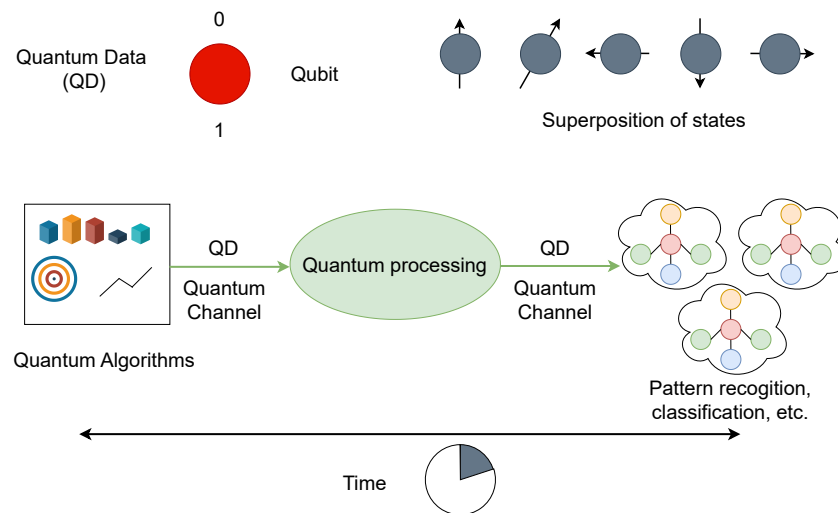


**Figure 4.** Qubit growth in quantum computers over recent years. This figure presents the increasing number of qubits in quantum processors, reflecting advancements in quantum computing technology.

Current applications of quantum computers span a wide range of domains, demonstrating tangible benefits in solving complex problems that challenge classical systems. In cryptography, quantum computers are revolutionising secure communication by exploiting superposition and entanglement to enhance encryption and decryption processes [47]. Similarly, in molecular simulation, quantum algorithms enable precise modelling of molecular structures and interactions, crucial for drug discovery [48], materials science [49], and other chemistry-related fields [50]. These advancements hold the potential to accelerate breakthroughs in healthcare, energy, and environmental sustainability. Moreover, financial modelling is another promising domain, where quantum computers optimise portfolios, predict market trends, and manage risk with unprecedented speed and accuracy [51].

The rise of quantum machine learning (QML) adds a new dimension to the application of quantum computers. QML leverages quantum algorithms to enhance ML tasks such as classification, pattern recognition, and autonomous decision-making [52]. By leveraging quantum speed-ups, QML can process complex datasets more efficiently than classical methods, offering advantages in fields such as finance, healthcare, and AI. Figure 5 illustrates the workflow of QML, highlighting the interaction between quantum data, quantum gates, and ML models in tasks such as image classification and dynamic decision-making in autonomous systems.

In conclusion, quantum computing represents one of the most transformative trends in the evolving landscape of parallel systems. By harnessing the fundamental principles of quantum mechanics, quantum computing is poised to complement classical HPC, unlocking unprecedented computational power for scientific discovery and industrial applications.



**Figure 5.** Overview of QML. This figure illustrates the integration of quantum computing principles in ML, showing how quantum algorithms leverage qubit-based computation. The green arrows indicate the data flow of quantum information between processing units.

### 3.3. Neuromorphic Computing

Neuromorphic computing is a class of brain-inspired computing architectures which, at a certain level of abstraction, simulate the biological computations of the brain. This approach enhances the efficiency of compatible computational tasks and achieves computational delays and energy consumption with biological computation. The term “neuromorphic” was introduced by Carver Mead in the late 1980s [53,54], referring to mixed analogue–digital implementations of brain-inspired computing. Over time, as technology evolved, it came to encompass a wider range of brain-inspired hardware implementations. Specifically, unlike the von Neumann architecture’s CPU–memory separation and synchronous clocking, neuromorphic computing utilises neurons and synapses, the fundamental components, to integrate computation and memory. It employs an event-driven approach based on asynchronous event-based spikes, which is more efficient for the brain-like sparse and massively parallel computing, significantly reducing energy consumption. At the algorithmic level, the brain-inspired Spike Neural Network (SNN) serves as an essential algorithm deployed on neuromorphic hardware, efficiently completing ML tasks [55,56] and other operations [57,58]. Recent advancements in VLSI technology and AI have propelled neuromorphic computing towards large-scale development [59]. This introduces developments in neuromorphic computing from both hardware and algorithmic perspectives and discusses future trends.

IBM TrueNorth is based on distributed digital neural models designed to address cognitive tasks in real time [60]. Its chip contains 4096 neurosynaptic cores, each core featuring 256 neurons, with each neuron having 256 synaptic connections. On the one hand, the intra-chip network integrates 1 million programmable neurons and 256 million trainable synapses; on the other hand, the inter-chip interface supports seamless multi-chip communication of arbitrary size, facilitating parallel computation. By using offline learning, various common algorithms such as convolutional networks, restricted Boltzmann machines, hidden Markov models, and multi-modal classification have been mapped to TrueNorth, achieving good results in real-time multi-object detection and classification tasks with milliwatt-level energy consumption.

Neurogrid, a tree-structured neuromorphic computing architecture, fully considers neural features such as the axonal arbor, synapse, dendritic tree, and ion channels to maximise synaptic connections [61]. Neurogrid uses analogue signals to save energy and

a tree structure to maximise throughput, allowing it to simulate 1 million neurons and billions of synaptic connections with only 16 neurocores and a power consumption of only 3 watts. Neurogrid's hardware is suitable for real-time simulation, while its software can be used for interactive visualisation.

As one of the neuromorphic computing platforms contributing to the European Union Flagship Human Brain Project (HBP), SpiNNaker is a parallel computation architecture with a million cores [62]. Each SpiNNaker node has 18 cores, connected by a system network-on-chip. Nodes select 1 neural core to act as the monitor processor, assigned an operating system support role, while the other 16 cores support application roles, with the 18th core reserved as a fault-tolerance spare. Nodes communicate through a router to complete parallel data exchange. SpiNNaker can be used as an interface with AER sensors and for integration with robotic platforms.

Intel's Loihi is a neuromorphic research processor supporting multi-scale SNNs, achieving performance comparable to mainstream computing architectures [63,64]. Loihi features a maximum of 128,000 neurons per chip with 128 million synapses. Its unique capabilities include a highly configurable synaptic memory with variable weight precision, support for a wide range of plasticity rules, and graded reward spikes that facilitate learning. Loihi has been evaluated in various applications, such as adaptive robot arm control, visual-tactile sensory perception, modelling diffusion processes for scientific computing applications, and solving hard optimisation problems like railway scheduling. Loihi2 [65], as a new generation of neuromorphic computing and an upgrade of Loihi, is equipped with generalised event-based messaging, greater neuron model programmability, enhanced learning capabilities, numerous capacity optimisations to improve resource density, and faster circuit speeds. Importantly, besides the features from Loihi1, Loihi2 has shared synapses for convolution, which is ideal for deep convolutional neural networks.

SNNs are an essential algorithmic component of neuromorphic computing. To accomplish a task, we should consider how to define a tailored SNN and deploy it on hardware [54]. From a training perspective, algorithms can be categorised into online learning and offline learning. The offline-learning approach first deploys the SNN on neuromorphic hardware and then uses the plasticity features to approximate backpropagation. This is a real-time method for optimising hardware simulation of plasticity. Offline learning involves training an Artificial Neural Network (ANN) on a CPU or GPU based on specific tasks and data, then converting the ANN to an equivalent SNN and deploying it on neuromorphic hardware. As a key to training algorithms, various studies have analysed backpropagation.

An Energy-Efficient Backpropagation approach successfully implemented backpropagation on TrueNorth hardware [56]. Importantly, this method treats spikes and discrete synapses as continuous probabilities, allowing the trained network to map to neuromorphic hardware through probability sampling. This training method achieved 99.42% accuracy on the MNIST dataset with only 0.268 mJ per image. Furthermore, backpropagation through time (BPTT) has been implemented on neuromorphic datasets, providing a training method for recurrent structures on neuromorphic platforms [66]. Benefiting from these training optimisations, SNNs in neuromorphic computing have been applied in various ML tasks such as Simultaneous Velocity and Texture Classification [67], Real-time Facial Expression Recognition [68], and EMG Gesture Classification [69]. Similarly, they have been used in neuroscience research [70,71]. SNN-based neuromorphic computing is also utilised in non-ML tasks. Benefiting from the neuromorphic vertex-edge structure, graph theory problems can be mapped onto the hardware [58,72,73]. Additionally, it has been applied to solving NP-complete problems [74].

Neuromorphic computing often aims to replicate aspects of biological neural processing in hardware, but there is an ongoing debate over how strictly such systems must adhere to biophysical plausibility versus employing more abstract ML methods. On the one hand, SNN models, such as the Izhikevich formulation [75], focus on capturing the temporal dynamics of real neurons, which can yield insights into how biological brains encode and process information. Research has shown that such models can replicate a variety of neuronal firing patterns with computational efficiency, providing a bridge between computational neuroscience and neuromorphic engineering [76]. On the other hand, more traditional ML algorithms, such as Bayesian inference [77], support vector machines [78], or the large language models [79] dominating modern AI, tend to trade some fidelity to biological detail for mathematical tractability, scalability, and often better empirical performance on a range of industrial tasks.

Despite the proven feasibility of neuromorphic computing in many tasks, it remains largely experimental. In today's landscape of energy-consuming AI driven by GPU clusters, bringing neuromorphic computing out of the lab and achieving performance equal to or better than GPU-based AI with low energy consumption is a significant trend [80–82]. Standardised hardware protocols and community-maintained software will be crucial. From a neuroscience research perspective, neuromorphic computing simulates brain structures to varying degrees. Leveraging these simulations could provide new insights into neural mechanisms and brain function. Neuromorphic computing has a close-loop relationship with both AI and neuroscience, drawing inspiration from and serving both fields, tightly linking their development and advancing our understanding of intelligence.

### 3.4. Optical Computing

Optical computing utilises the properties of light to perform parallel computations, providing the potential to significantly exceed the speed and efficiency of electronic computing [83]. Unlike electronic computing, which relies on the movement of electrical charges, optical computing uses photons to carry and process information. Because light travels faster and experiences minimal resistance, optical computing has the potential to significantly improve processing speeds and energy efficiency. Research in optical computing can be traced back to the early 1960s [84]. Over the years, the primary focus of optical computing has been on integrating optical components for communication within computer systems or incorporating optical functions due to advances in electronic technology [84]. Although these elements remain under development and have yet to mature, the adaptation and exploration of optical computing, especially in AI, have grown rapidly in recent years due to the boom in AI and the limitations of traditional electrical architectures. To explore the emerging trends in optical computing, we first examine the different categories of optical computing systems and then discuss the potential and outlook of optical computing in AI.

Optical computing systems can be categorised into analogue, digital, and hybrid optical computing systems (OCS). Each category differs in how it processes information, balancing speed, precision, and scalability. Analogue optical computing systems (AOCS) utilise the continuous nature of light to perform computations, leveraging properties such as intensity, phase, and wavelength to represent and process data. This enables high precision and real-time processing capabilities, making AOCS suitable for signal processing and image recognition applications. On the other hand, digital optical computing systems (DOCS) operate on binary principles similar to traditional electronic computers, where light is used to represent binary data (0's and 1's) and perform logical operations through optical gates. DOCS can achieve exceptionally high-speed processing and parallelism, ideal for tasks requiring rapid data computation. However, scalability and integration difficulties

pose significant challenges for DOCS, particularly in large-scale systems. Hybrid optical computing systems (HOCS) combine the strengths of both analogue and digital approaches, integrating continuous and discrete data representations to optimise performance across a broader range of applications. By leveraging the unique advantages of light, such as its speed and bandwidth, these hybrid systems can enhance computational efficiency and open new frontiers in fields such as telecommunications, AI, and scientific simulations. Table 1 summarises the features of these three systems.

**Table 1.** Comparison of optical computing systems. This table compares analogue optical computing systems (AOCS), digital optical computing systems (DOCS), and hybrid optical computing systems (HOCS) based on key characteristics such as data type, speed, error susceptibility, complexity, integration challenges, and applications.

Feature	AOCS	DOCS	HOCS
Data type	Continuous	Discrete (binary)	Both continuous and discrete
Speed	Very high	High	High
Error susceptibility	Higher	Lower	Balanced
Complexity	Lower	Higher	Medium
Integration	Challenging	Easier	Moderate
Applications	Real-time processing, imaging	Logic operations, data storage	Neural networks, adaptive optics

Optical computing also leverages optical components such as microring resonators (MRRs) and Mach–Zehnder interferometers (MZIs) to design essential elements such as logic gates, switches, storage devices, routers, and photonic integrated circuits. Microring resonators act as miniature loops that guide and filter light, while Mach–Zehnder interferometers function as optical switches, enabling precise control over light-based computations. The development of these components has been pivotal in advancing the field, leading to the creation of more compact and powerful photonic circuits. Initially, research focused on the fundamental properties of light and how it could be manipulated for computation. Over time, advancements in materials science and nanofabrication have enabled greater miniaturisation and improved integration of optical components.

Optical computing is making significant strides across multiple domains, particularly in telecommunications, AI, and HPC. Key areas of impact include the following: 1. *Telecommunications*: Optical components enhance data transmission speeds and network capacity. Photonic technologies in fibre-optic networks reduce latency and increase bandwidth, making them integral to modern communication infrastructures [85]. 2. *AI*: Optical neural networks, particularly those utilising MZIs, enable AI computations at speeds beyond conventional electronic processors. One notable example is the use of optical matrix multiplication for accelerating deep learning models, significantly reducing energy consumption in AI training [86]. 3. *HPC*: The integration of photonic integrated circuits (PICs) and photonic–electronic co-design is advancing HPC infrastructures [87]. The adoption of optical interconnects in HPC provides high bandwidth and lower energy consumption, significantly improving data transfer efficiency for large-scale simulations and AI training [88]. Despite these advancements, optical computing still faces challenges, including manufacturing complexities, optical loss, and crosstalk, which hinder large-scale adoption [89]. However, ongoing research in photonic materials and integrated circuit design continues to address these limitations, paving the way for more scalable optical computing solutions.

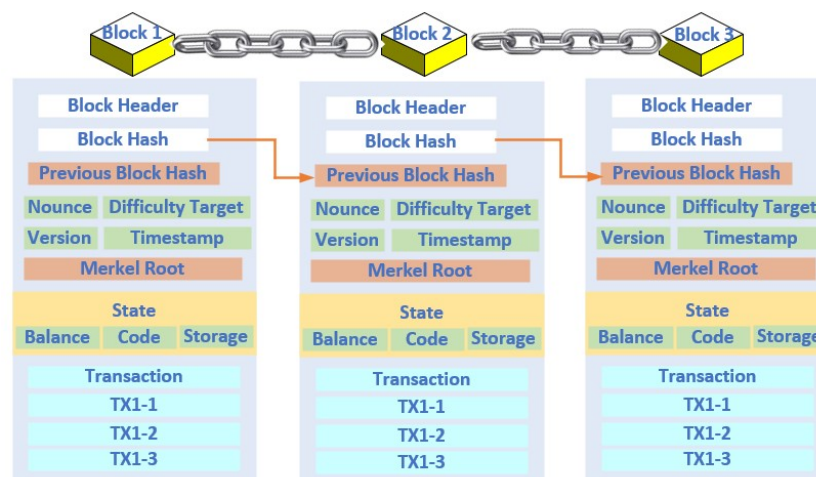
Optical computing has evolved significantly with advances in component technology, growing applications, and increasing research interest. While it is not yet poised to replace electronic computing entirely, it is expected to play a complementary role, particularly in areas demanding ultra-fast, energy-efficient computations. As research progresses, breakthroughs in nanophotonics, integrated optical chips, and AI-driven photonic computing will likely drive optical computing toward mainstream adoption. With further improvements in scalability and integration, optical computing may soon redefine HPC, revolutionising fields such as AI, communications, and beyond. For readers interested in a more in-depth exploration of optical computing, the review papers [84,86,90], as well as the book [91], provide comprehensive insights into its fundamentals and applications.

## 4. Emerging Trends in Distributed Systems

In the rapidly evolving landscape of computing, distributed systems have become integral to handling the scale, complexity, and diversity of modern applications. By leveraging multiple interconnected computing resources, distributed systems provide scalable, resilient, and efficient solutions that traditional centralised systems cannot offer. As data volumes grow exponentially and applications demand real-time processing and decision-making, innovative approaches in distributed computing are essential. This section explores the emerging trends in distributed systems, focusing on four key areas: blockchain and distributed ledgers, serverless computing, cloud-native architectures, and distributed AI and ML systems. These advancements are redefining how data are managed, processed, and secured across various industries, enabling new possibilities while addressing critical scalability, efficiency, security, and privacy challenges.

### 4.1. Blockchain and Distributed Ledgers

The concept of the blockchain was first coined by Satoshi Nakamoto in 2008 after the failure of the global financial system in 2007 [92]. Though he did not formally define the blockchain, he demonstrated the blockchain concept for electronic cash (called Bitcoin) transfers where no central authority is needed to prevent double-spending. The first successful Bitcoin transaction took place in 2009 when Satoshi Nakamoto transferred 10 BTC (Bitcoin) to Hal Finney. Satoshi uses a peer-to-peer network to timestamp transactions through a hash-based Proof-of-Work chain, which acts as an unchangeable record unless the Proof of Work is redone. However, the concept of blockchain is fundamentally based on three elements: (i) blind signature, a cryptographic concept proposed by David Chaum in 1989 for automation of payment [93]; (ii) timestamped documents which secure digital documents by stamping the documents with the date [94]; and (iii) Reusable Proof of Work (RPoW), a mechanism for preventing double-spending and securing decentralized networks, later extended into a reusable format by Hal Finney in 2004 [95]. Therefore, researchers formally defined the blockchain as a meta-technology which combines several computing techniques [96]. However, the most widely adopted definition of blockchain is a distributed digital ledger technology with a ledger of transactions, or blocks, that form a systematic, linear chain of all transactions ever made. Blockchain presents timestamped and immutable blocks of highly encrypted and anonymised data not owned or mediated by any specific person or groups [97,98]. A block in a blockchain is primarily identified by its block header hash or block hash, a cryptographic hash made by hashing the block header twice through the SHA256 algorithm. In addition, a block can also be identified by the block height, which is its position in the blockchain or the number of blocks preceding it in the blockchain. The Merkle tree offers a secure and efficient way to create a digital fingerprint for the complete set of transactions. A blockchain structure is shown in Figure 6, where blocks are connected through their respective hash code.



**Figure 6.** Basic structure of a blockchain block. This figure presents the fundamental components of a blockchain block, explaining how distributed ledger technology ensures security and integrity in decentralised networks.

Distributed ledger technology (DLT) is the underlying generalised concept that makes the blockchain work in a distributed platform. The concept of DLT incorporates principles from “The Byzantine General Problem”, described by Lamport et al. [99], which evaluates the strategies for achieving consensus in distributed systems despite conflicting information in an adversarial environment. Consensus protocols, like Proof of Stake, allow participants to achieve a shared view of the ledger without intermediaries. Emerging mechanisms, such as Proof of Space and Proof of Authority [100], have gained attention for their lower energy consumption and faster transaction verification times compared to Proof of Work. These mechanisms aim to address the inefficiencies and environmental impacts associated with traditional methods, offering tailored solutions for specific use cases. Additionally, cryptographic techniques, such as the Schnorr Signature Scheme and Merkle Tree, enhance data integrity and trust within blockchain frameworks, reinforcing secure data verification processes [101]. A distributed ledger is a digital record maintained across a network of machines, known as nodes, with any updates being reflected simultaneously for all participants and authenticated through cryptographic signatures [102].

Beyond cryptocurrencies, blockchain’s applications span a wide range of industries, including eHealth [103,104], intellectual property [105,106], education, digital identity, finance [107–109], supply chain [110–112], IoT [113–115], etc. In supply chain management, blockchain frameworks such as IBM Food Trust provide end-to-end traceability, ensuring transparency and accountability. Case studies, such as Walmart’s use of blockchain to track food provenance, have quantified significant reductions in tracing times, from days to seconds, illustrating blockchain’s potential to streamline operations and mitigate fraud. In healthcare, blockchain’s anonymity and immutability features make it unparalleled for secure information sharing among different providers, forming the foundation of modern healthcare, alternatively termed Healthcare 5.0. Numerous frameworks such as MeD-Share [116], Medblock [117], HealthBlock [118], and BLOSOM [119] have been developed to secure patient records. BCIF-EHR, an interoperable blockchain-based framework proposed in [103], facilitates seamless sharing and integration of electronic health records (EHRs) while preserving privacy and security. However, the framework requires a decentralised authentication and access control mechanism to restrict access to authorised entities only. Addressing this limitation, TrustHealth [104] integrates blockchain with a trusted execution environment, designing a secure database that ensures the confidentiality and integrity of EHRs. TrustHealth also incorporates a secure session key generation protocol,

enabling secure communication channels between healthcare providers and the trusted execution environment. Such advancements exemplify blockchain's ability to transform healthcare by improving interoperability, security, and trust.

Despite its broad applicability, blockchain faces challenges such as latency and high energy consumption, particularly in Proof-of-Work-based systems. These issues can hinder real-time applications and raise concerns about environmental sustainability. Additionally, blockchain's reliance on distributed consensus mechanisms can lead to cold-start issues in networks with low node participation, delaying transaction validation. Overall, blockchain's transformative potential lies in its ability to provide secure, transparent, and decentralised solutions across diverse sectors, fundamentally changing how data integrity and trust are managed.

#### 4.2. Serverless Computing

The concept of serverless computing emerged in the mid-2000s with cloud services like Amazon S3 and EC2, which simplified infrastructure management for developers [120]. However, a major breakthrough came in 2014 with the introduction of AWS Lambda [121], which established the Function-as-a-Service (FaaS) model. This allowed developers to execute code in response to events without managing servers, providing automatic scaling and reducing operational overhead [122]. IBM OpenWhisk (2016) later expanded on this concept by offering an open-source alternative that prioritised flexibility [123]. Further advancements included Microsoft Azure Functions [124] and Google Cloud Run [125], which integrated containerised workloads to extend serverless capabilities.

Serverless computing provides automatic scalability, eliminating the need for manual resource management. A key advantage of this model is its "pay-as-you-go" pricing structure, where users pay only for the compute time they consume rather than pre-allocated resources, significantly reducing costs for variable workloads [126]. These benefits make serverless computing ideal for applications such as web services [127], IoT [128], and large-scale data processing [129]. Industries including finance, healthcare, and e-commerce utilise serverless computing to enable rapid scaling and resource efficiency. Major companies like Netflix and Airbnb rely on serverless architectures to handle fluctuating traffic loads, ensuring a smooth user experience during peak demand [130]. Studies indicate that serverless platforms can handle up to 10,000 concurrent function executions while maintaining response times below 500 ms, making them suitable for real-time applications [131].

Despite its advantages, serverless computing presents several challenges. One major concern is cold-start latency, which occurs when an idle function is invoked and requires initialisation. To mitigate this, techniques such as function pre-warming, optimising container configurations, and adjusting function granularity have been developed, reducing cold-start delays by up to 50% in production environments [132,133]. Another issue is vendor lock-in, where applications become dependent on proprietary cloud provider implementations. To overcome this, multi-cloud serverless frameworks like Knative and OpenFaaS have emerged, allowing developers to deploy serverless workloads across multiple providers, increasing flexibility and reducing dependency risks [134]. Furthermore, serverless architectures are not well suited for long-running processes, as they impose execution time limits. Hybrid serverless–edge computing models are increasingly being explored to process latency-sensitive workloads closer to the data source, particularly for IoT applications [135].

Ongoing advancements aim to enhance serverless computing's flexibility and performance. AI-based function pre-warming, such as Alibaba Cloud's Function Compute prediction models, proactively warms up instances to reduce startup delays [136]. Federated serverless architectures provide cost-effectiveness and resource efficiency [137].

Additionally, confidential computing techniques like secure enclaves are being integrated to enhance function-level security, mitigating multi-tenant isolation concerns [138]. For high-frequency workloads, the unpredictable costs of serverless computing can sometimes make traditional cloud computing a more economical option. Research into more transparent and cost-efficient serverless pricing structures is ongoing [139].

Overall, while serverless computing offers scalability, cost efficiency, and operational flexibility, its adoption requires addressing challenges related to latency, vendor dependence, and security. Continued advancements in optimisation techniques, multi-cloud interoperability, and pricing models will further enhance its impact on the future of cloud computing.

#### 4.3. Cloud-Native Architectures

Cloud-native architectures began with distributed systems research in the 1990s [140] and the introduction of virtualisation by VMware in 1998 [141]. In 2006, AWS launched EC2 and S3, making on-demand cloud services widely available [120]. DevOps ideas took off around 2009 [142], combining development and operations to speed up software delivery. Docker emerged in 2013 as a platform for packaging applications into lightweight containers [143], followed by Google's open-source release of Kubernetes in 2014 to orchestrate and manage containerized workloads [144].

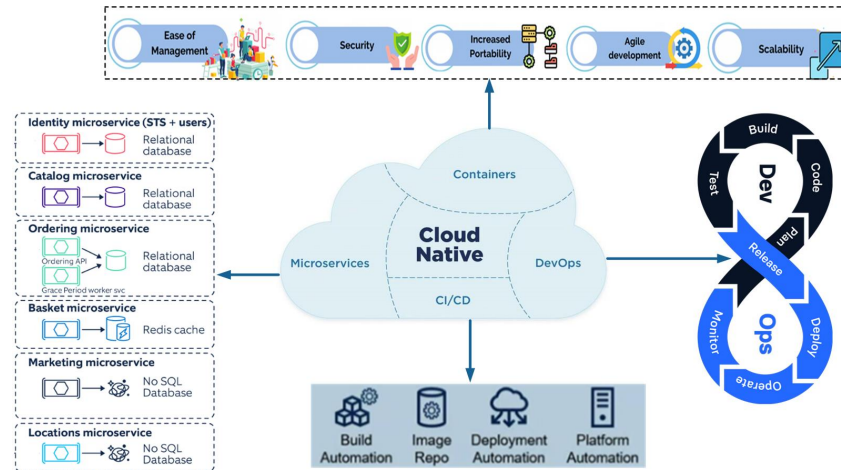
Docker emerged in 2013 for packaging applications into lightweight containers [143], followed by Google's open-source release of Kubernetes in 2014 to orchestrate and manage containerised workloads [144]. The Cloud Native Computing Foundation (CNCF) formed in 2015 and made Kubernetes its first project [144], while AWS Lambda (launched in 2014) introduced serverless computing [121], and service meshes emerged to handle microservice communication [128].

Today, cloud-native architecture optimises cloud application performance by integrating microservices, containerisation, and continuous integration/continuous delivery (as shown in Figure 7) [145,146]. These techniques enable modularity, scalability, and reliability. Microservices divide applications into independent, manageable services, containerisation ensures consistent deployment, and CI/CD accelerates the development life cycle, creating a robust framework for efficiently handling dynamic workloads. Tools like Docker and Kubernetes simplify container orchestration, streamline scaling, and accelerate deployment pipelines [147]. Unlike traditional monolithic structures, cloud-native applications are modular, allowing components to be managed, scaled, and updated independently. This makes cloud-native architectures highly effective in dynamic environments demanding rapid iteration and resilient deployment [148–150].

Alongside these core components, advanced communication paradigms such as Partitioned Global Address Space (PGAS) models and Remote Direct Memory Access (RDMA) further enhance cloud-native platforms [151]. PGAS models provide a shared memory abstraction across distributed systems, emphasising data locality and reducing communication overhead, making them particularly suitable for high-performance applications in cloud environments. RDMA further enhances infrastructure efficiency by enabling direct memory-to-memory transfers between nodes, bypassing CPU involvement to minimise latency and maximise throughput. These technologies are critical for optimising the performance of modern distributed systems and are increasingly adopted in cloud-native platforms.

Cloud-native architectures also play a pivotal role in Industry 4.0, where real-time data processing across IoT and edge devices matters most. These architectures realise smooth integration in distributed systems, hence managing large-scale, latency-sensitive data efficiently [152]. By incorporating PGAS and RDMA, these architectures can handle

complex data flows and resource-intensive tasks with greater efficiency, supporting the scalability demands of Industry 4.0. One of the prominent design elements of cloud-native designs is the multi-cloud and distributed cloud models that, for instance, enable the deployment of applications across multiple cloud providers [153]. This increases availability and avoids vendor lock-in, giving flexibility and resilience to enterprises by utilising unique services across cloud platforms [154].



**Figure 7.** Key building blocks of a cloud-native architecture. This figure illustrates the four fundamental components of cloud-native systems: containers, microservices, DevOps, and CI/CD. These elements enable scalability, automation, and continuous deployment in modern cloud computing environments.

On top of this, cloud-native architectures leverage Platform-as-a-Service (PaaS) environments to simplify infrastructure management and scaling [155]. Cloud federation strategies improve interoperability across providers, enabling seamless service migration and management in heterogeneous systems [156]. Infrastructure as Code (IaC) automates resource provisioning, ensuring efficient and secure application deployment [157]. By combining these methods with advanced communication paradigms, cloud-native architectures offer robust fault tolerance and high resource utilisation, supporting a range of workloads from e-commerce to scientific computing.

Despite these benefits, cloud-native approaches come with their own set of challenges. Integrating PGAS and RDMA can be complex, requiring specialised hardware and in-depth expertise, which may raise costs and limit portability across diverse platforms. Deploying microservices at scale also necessitates comprehensive observability solutions to handle complex debugging and performance monitoring tasks. In multi-cloud scenarios, while the flexibility is appealing, organisations may still encounter partial vendor lock-in due to unique service integrations. Security remains a prominent concern, as misconfigurations in container orchestration or vulnerabilities within microservices can open pathways for data breaches. Additionally, the rapid pace of innovation in the cloud-native ecosystem demands continual learning and adaptation, placing pressure on both developers and operators to stay abreast of emerging tools and best practices [144]. Balancing these challenges with the clear advantages of agility, scalability, and resilience is essential for successful adoption across various industries.

#### 4.4. Distributed AI and ML Systems

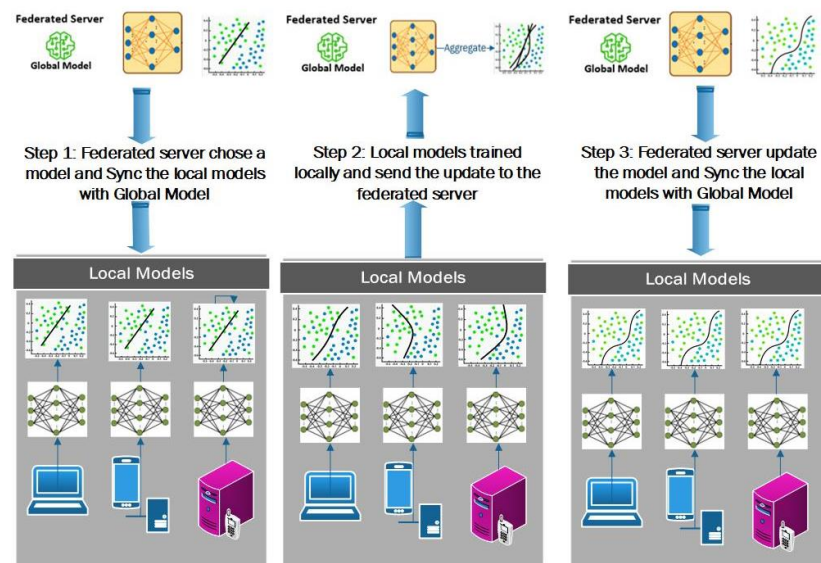
Distributed AI and ML systems are the backbone for scalable training and deployment of complex models across decentralised networks [158]. Unlike the centralised approach, this architecture allows the computation to be distributed among different nodes, reducing

the latency in training and efficiently processing large datasets [159]. This ML approach can optimise learning and AI inference, particularly for resource-constrained devices such as IoT or edge computing devices used in real-time applications [160]. It aligns with the principles of federated learning, which allow for collaborative model training without the need to share raw data, thus preserving data privacy and reducing bandwidth demands [161]. By leveraging intelligent agents in a distributed environment, these systems can significantly reduce model training time while maintaining robust fault tolerance [162]. Moreover, distributed learning algorithms applied in different application areas, such as 6G [163] and smart grid systems [164], illustrate how these methods can optimise resource usage and enable real-time decision-making with minimal latency. Advanced variants, such as AutoDAL, enable automatic hyperparameter tuning within distributed learning frameworks, addressing scalability and efficiency challenges in large-scale data analysis [165].

Federated learning is an emerging area in distributed AI, allowing model training across decentralised devices or servers without centralising raw data. This approach improves privacy and reduces data transfer costs, with models trained locally on edge devices and only shared parameters sent back to central servers, as shown in Figure 8 [166]. Federated learning is especially valuable in applications with strict privacy requirements, such as healthcare and finance, where regulatory constraints limit centralised data storage [167]. However, federated systems face significant challenges in balancing privacy preservation and model accuracy. Privacy-preserving techniques, such as differential privacy and secure multi-party computation, introduce noise or encryption that can reduce model performance [168]. To address this, privacy-aware optimisation algorithms, such as those incorporating adaptive noise levels or secure aggregation protocols, have been proposed to maintain accuracy while ensuring data security [169,170]. Another critical challenge in federated systems is communication overhead, especially in scenarios involving frequent synchronisation of model updates across devices. This overhead can significantly increase latency and reduce efficiency in large-scale systems. Potential solutions include strategies like periodic aggregation, where updates are transmitted at predefined intervals rather than continuously [171], and selective model updates, which prioritise transmitting critical updates based on gradient sparsity or importance [172]. Additionally, techniques such as gradient compression and quantised updates can minimise communication costs without sacrificing accuracy, making federated learning more scalable and efficient in distributed environments [173,174]. These advancements demonstrate that federated learning can address privacy and efficiency challenges effectively, paving the way for its widespread adoption in privacy-sensitive domains.

Distributed training systems enable simultaneous model training across multiple nodes, significantly accelerating the development of complex AI models. Techniques like data parallelism, model parallelism, and pipeline parallelism optimise resource usage, making them essential for large-scale training tasks in fields like natural language processing and computer vision, where computational demands are exceptionally high [175]. By distributing workloads across multiple nodes, these systems reduce the dependence on centralised infrastructures, promoting scalable, efficient, and resource-adaptive ML [176]. Despite these advantages, distributed training systems face several challenges that limit their efficiency and effectiveness. Communication overhead, caused by frequent synchronisation of parameters across nodes, can result in increased latency and inefficient bandwidth utilisation, particularly in large-scale systems [177]. Techniques like gradient sparsification [178], optimised collective communication protocols [31], and Asynchronous Stochastic Gradient Descent (ASGD) [179,180] aim to mitigate these issues by reducing the volume of data transmitted during updates and allowing nodes to operate more independently.

However, these methods often struggle to maintain model accuracy due to inconsistent parameter updates [181], requiring advanced consistency management algorithms, such as dynamic weighting of updates, to address this trade-off. Another significant challenge is managing data heterogeneity, as data distributed across nodes are often non-IID (non-independent and identically distributed), leading to skewed model updates that hinder training effectiveness [182]. Solutions like adaptive loss functions, dynamic weighting of local models, and frameworks such as AdaFed [183] dynamically adjust the contributions of local models based on data quality, improving convergence. Privacy-preserving methods, such as differential privacy and secure multi-party computation, add further complexity by introducing noise or encryption to protect sensitive data, which can degrade model accuracy [184]. Privacy-aware optimisation strategies such as PSDF [185] are being developed to balance security with performance. Resource optimisation is another critical issue [186], particularly in decentralised environments with heterogeneous hardware capabilities and network reliability. Adaptive resource allocation frameworks that dynamically adjust computation and communication parameters based on workload demands and node capacities [187] are essential for efficient resource utilisation, but implementing these frameworks requires robust scheduling algorithms and real-time monitoring. Addressing these challenges through innovative algorithms, resource management strategies, and privacy-aware techniques is essential for unlocking the full potential of distributed training systems.



**Figure 8.** Step-by-step illustration of federated ML. This figure explains the federated learning process, highlighting key stages such as local model training, aggregation, and privacy-preserving updates.

In summary, distributed AI and ML systems offer transformative potential by enabling scalable, efficient, and secure training across decentralised networks, while challenges such as communication overhead, data heterogeneity, and synchronisation remain, ongoing advancements in adaptive algorithms and privacy-preserving methods continue to address these issues, paving the way for widespread adoption in sectors like healthcare, finance, and IoT.

## 5. Challenges in Parallel and Distributed Systems

Parallel and distributed systems have revolutionised the way computational tasks are performed, enabling the handling of complex and large-scale applications. However, these

systems face several challenges that can hinder their efficiency and effectiveness. This section delves into the key challenges in parallel and distributed systems, including scalability and performance, security and privacy, fault tolerance and reliability, interoperability and standardisation, energy efficiency, and ethical concerns.

### 5.1. Scalability and Performance

Achieving scalability while maintaining high performance is one of the foremost challenges in parallel and distributed systems. As the number of processors or nodes increases, bottlenecks can arise due to limitations in network bandwidth, synchronisation overhead, and resource contention [188]. These issues can significantly degrade system performance, negating the benefits of adding more computational resources. A notable example of scalability challenges is in distributed AI systems, where training large-scale models like GPT3 involves 175 billion parameters spread across thousands of GPUs [189,190]. Synchronisation overhead during gradient updates can significantly impact training efficiency, especially as the number of GPUs increases [191]. Studies have shown that communication overhead during parameter updates and gradient synchronisation can dominate the training time in large-scale distributed systems, reducing the benefits of scaling out [192]. Additionally, memory bandwidth and latency constraints exacerbate the problem, reducing the overall efficiency of these systems [175].

Addressing these challenges requires integrated solutions that consider the distinct demands of heterogeneous, quantum, neuromorphic, and optical computing paradigms. In heterogeneous computing, task scheduling algorithms ensure efficient workload distribution among diverse processing units (e.g., CPUs, GPUs, DPUs) to prevent resource underutilisation [193]. Advanced scheduling algorithms dynamically assign tasks to appropriate processors, optimising execution and guaranteeing energy consumption [194]. For quantum computing, modular quantum architectures and hybrid quantum–classical systems help manage the scalability of qubit systems while reducing error propagation [195]. In neuromorphic computing, innovations such as photonics integration, online learning, and 3D stacking enhance the scalability of ANNs by increasing density and reducing power consumption [196]. For optical computing, material advancements such as silicon photonics and integrated photonic circuits enable the scaling of optical interconnects while minimising crosstalk and optical loss. Hardware/software co-design innovations further enhance the performance of optical computing systems [87]. Optimising communication protocols [197], dynamic resource allocation [198], and adaptive scheduling algorithms [199] improve data transfer and task management. Gradient compression techniques in distributed AI systems reduce communication delays, while optical interconnects [85] and optical wireless communication [200] provide high-bandwidth, low-latency data transfer, enhancing overall system efficiency. Together, these advancements improve workload distribution, enabling parallel and distributed systems to scale efficiently and meet the demands of complex modern applications.

### 5.2. Security and Privacy

Security and privacy are paramount concerns in distributed environments where data and resources are shared across multiple nodes [201]. Threats such as unauthorised access, data breaches, and malicious attacks can compromise the integrity and confidentiality of the system. Distributed systems are particularly vulnerable due to their open and interconnected nature, which can be exploited by attackers. A notable case involved a major cloud service provider experiencing downtime across its network due to a coordinated ransomware attack, resulting in financial losses exceeding USD 1.85 million and extensive recovery efforts [202]. Similarly, in parallel systems used for HPC, side-channel attacks that

exploit shared memory vulnerabilities have exposed sensitive data, highlighting the need for enhanced security measures [203].

To address these challenges, robust security solutions should span hardware, software, and cryptographic advancements [204]. Encryption methods such as AES-256 and secure communication protocols like TLS ensure data protection during storage and transmission, while authentication mechanisms, including multi-factor authentication, enhance access control [205]. Zero-trust architectures and Trusted Execution Environments limit attack surfaces by isolating sensitive computations and continuously validating user and device credentials [206]. In distributed systems like blockchain, mechanisms such as Merkle trees and Proof-of-Stake consensus algorithms maintain data integrity and ensure secure transaction validation [207]. With the advent of quantum computing, post-quantum cryptography, including lattice-based cryptography, and quantum key distribution are critical for securing communications and future-proofing systems against quantum-enabled threats [208]. These solutions, when integrated into parallel and distributed systems, provide resilience against evolving cyber threats, safeguarding user privacy and ensuring system reliability.

### 5.3. Fault Tolerance and Reliability

Fault tolerance and reliability are critical in ensuring that parallel and distributed systems continue to operate correctly even in the presence of component failures [209]. Hardware malfunctions, network issues, or software errors can lead to system downtime or data loss, which is unacceptable in mission-critical applications. For instance, distributed systems supporting global financial transactions need to maintain uninterrupted operation despite hardware failures or network disruptions, as downtime can result in significant financial and reputational losses [210].

Many methods have been proposed to address these challenges in various distributed computing scenarios. Redundancy and replication ensure high availability and data integrity by maintaining multiple copies of critical data across nodes [206], while checkpointing periodically saves system states, enabling recovery without restarting entire processes [211]. Self-healing algorithms and dynamic task migration mitigate the impact of hardware and software failures by redistributing workloads to healthy nodes or components [212]. Similarly, modular architectures and error-correcting codes enhance the reliability of quantum systems by addressing decoherence and qubit failures [209]. Neuromorphic systems benefit from fault-tolerant designs and techniques that accommodate various types of resistive random-access memory faults [213]. Optical interconnect systems rely on the ONOS SDN controller for dynamic provisioning of data connectivity services and advanced automatic failure recovery [214]. Middleware solutions, such as those supporting distributed frameworks (e.g., Apache Spark) or blockchain consensus algorithms (e.g., Proof of Stake), enhance robustness against node failures and maintain consistency across distributed systems [210]. By integrating these strategies, parallel and distributed systems can enhance reliability, minimise disruptions, and meet the demands of modern mission-critical applications.

### 5.4. Interoperability and Standardisation

In heterogeneous environments where diverse systems and technologies coexist, interoperability becomes a significant challenge [215]. Orchestrating operations across different platforms, protocols, and interfaces requires careful coordination. Without standardisation, integrating new components or scaling the system can lead to incompatibilities and increased complexity. Managing heterogeneity in UHC systems, where CPUs, GPUs, NPU, and DPU have to collaborate seamlessly, exacerbates these challenges.

To address these challenges, adopting standardised communication protocols and resource allocation frameworks is essential [216]. Protocols like MPI and NCCL enable efficient data exchange in parallel systems [197], while resource allocation frameworks such as Kubernetes facilitate task distribution in distributed systems [144]. Middleware solutions abstract hardware and platform differences, simplifying the integration of components in heterogeneous and distributed environments [206]. For quantum systems, modular architectures and standardised quantum gates ensure compatibility between quantum and classical components, enabling hybrid quantum–classical workflows [217]. Neuromorphic systems require different neural coding schemes for achieving the best performance of neuromorphic systems under different design constraints [218]. In optical systems, the optical gates, photonic integrated circuits, and optical architectures are still evolving, and the development of standards and related protocols is ongoing [89]. Industry standards and open architectures promote interoperability, allowing diverse systems to work together while fostering collaborative innovation. For example, distributed frameworks like Apache Hadoop and TensorFlow support heterogeneous hardware, ensuring compatibility across CPUs, GPUs, and accelerators [219]. Such standardisation efforts reduce development costs, streamline integration, and enable parallel and distributed systems to scale efficiently, incorporating emerging technologies with minimal complexity.

### 5.5. Energy Efficiency

As parallel and distributed systems scale up, power consumption becomes a growing concern [220]. High energy usage not only increases operational costs but also has environmental implications due to the carbon footprint associated with large data centres and computing clusters. A notable example is the training of large-scale AI models like GPT-3, which reportedly consumed approximately 1287 megawatt-hours (MWh) of electricity during its training phase, emitting over 550 metric tons of carbon dioxide if powered by non-renewable sources [221]. This substantial energy use underscores the importance of implementing energy-efficient solutions across all parallel and distributed systems.

Addressing energy-efficiency challenges in parallel and distributed systems requires a holistic approach that integrates energy-efficient hardware, intelligent algorithms, dynamic power management, and sustainable infrastructure. Hardware innovations, such as neuromorphic chips like Intel's Loihi [63] and optical network processors [222,223], significantly reduce energy consumption through specialised designs and advanced technologies. Energy-aware algorithms, such as SkipTrain in decentralised learning [224], enhance efficiency by strategically skipping certain training rounds and replacing them with synchronisation rounds. Quantum algorithms like QAOA further minimise computational overhead, improving overall energy efficiency [216]. At the infrastructure level, renewable-powered data centres [220] and dynamic workload migration support sustainable operations. Additionally, techniques such as dynamic voltage and frequency scaling (DVFS) and adaptive power gating optimise energy usage by adjusting power levels based on workload demands [225]. For blockchain systems, energy-efficient consensus protocols such as Proof of Authority reduce power consumption while maintaining security and operational viability [100], helping to mitigate their environmental impact.

### 5.6. Emerging Ethical Concerns

As AI becomes increasingly integrated into parallel and distributed systems, ethical concerns have emerged as a critical challenge [226]. Issues such as algorithmic bias, misuse of sensitive user data, and lack of transparency in decision-making processes can undermine trust, fairness, and accountability in AI-driven systems [227]. For example, biased AI models deployed in distributed healthcare platforms can lead to unequal treatment

outcomes, disproportionately disadvantaging marginalised groups [228]. Similarly, inadequate data governance in cloud-based AI systems can result in privacy violations, exposing sensitive user information to misuse or unauthorised access [229].

Addressing these challenges requires a multi-faceted approach across governance, technology, and collaboration. Robust governance frameworks and adherence to ethical guidelines throughout the AI life cycle are essential for ensuring accountability. Explainable AI (XAI) techniques can improve transparency by providing interpretable insights into decision-making processes, reducing the risk of biased or opaque outcomes [230]. Privacy-preserving technologies, such as federated learning, allow data to remain decentralised, mitigating risks associated with data misuse or breaches [167]. Federated learning has shown promise in fields like healthcare, enabling collaborative model training without compromising data privacy [231]. Additionally, interdisciplinary collaborations among technologists, ethicists, and policymakers are vital for establishing standards and policies that promote equitable and responsible AI deployment. Standards such as fairness metrics, model validation protocols, and data auditing mechanisms ensure AI systems align with ethical principles. For instance, blockchain-based audit trails can improve accountability in distributed systems by recording data usage and decision-making processes securely and transparently [232]. By integrating these strategies, parallel and distributed systems can address emerging ethical concerns, fostering trust and ensuring sustainable, equitable development.

## 6. Future Directions

Building on the challenges outlined in this paper, it is evident that significant advancements are still needed to overcome scalability, energy efficiency, and security limitations in parallel and distributed systems. As these systems evolve, several emerging technologies and research areas show promise for addressing current obstacles and driving innovation. This section discusses the future directions of each class of parallel and distributed systems.

- *Heterogeneous computing:* As computing moves towards UHC architectures integrating diverse processors such as CPUs, GPUs, TPUs, FPGAs, and specialised accelerators, significant advancements are required to address challenges in scalability, energy efficiency, and complexity [233]. These architectures have the potential to revolutionise computing by leveraging the unique strengths of each processor type; however, their successful implementation depends on overcoming several critical obstacles. One key research direction is the development of hybrid scheduling algorithms [234]. These algorithms should dynamically adapt to varying computational demands, both online and offline, while optimising energy efficiency and performance [235]. Additionally, designing energy-aware resource management frameworks that minimise power consumption without compromising computational throughput is crucial for meeting sustainability goals [236]. Another vital area of focus is high-bandwidth, low-latency interconnect technologies, which are essential for seamless data exchange among heterogeneous components [237]. Innovations such as photonic interconnects and 3D packaging can alleviate bandwidth bottlenecks and reduce latency, enabling efficient communication between processors [238]. To enhance developer adoption and simplify programming for heterogeneous systems, further refinement of frameworks such as CUDA, OpenCL, SYCL, and oneAPI, as well as emerging unified programming models like CodeFlow [239], is essential. These frameworks should provide robust abstractions, allowing developers to harness the full potential of diverse architectures without dealing with low-level hardware complexities. Finally, synergies among quantum computing, neuromorphic systems, optical computing, and optical interconnects present exciting opportunities for future exploration. Advancing

these interdisciplinary technologies will be critical in shaping the next generation of high-performance, energy-efficient computing architectures.

- *Quantum computing*: The future trajectory of quantum computing is shaped by several critical technological and practical imperatives. At the hardware level, the ongoing development of diverse qubit technologies—including superconducting, silicon-based, trapped-ion, and photonic implementations—remains essential for advancing quantum computing capabilities [36,42,43]. While these platforms have demonstrated significant progress, challenges such as noise reduction, high error rates, and decoherence should be effectively addressed to realise practical quantum advantage [37]. Current quantum error correction protocols require substantial qubit overhead, necessitating innovative approaches that can scale efficiently with system size [240]. Industry roadmaps, such as IBM’s plan to develop processors with thousands of qubits [46], highlight the importance of achieving fault tolerance while maintaining quantum coherence across larger qubit arrays. The integration of quantum computing with classical computing represents a promising direction for near-term applications. Hybrid quantum–classical systems, particularly in ML and optimisation tasks, can leverage the complementary strengths of both paradigms [217]. To facilitate broader adoption, the field will address interconnected challenges, including quantum infrastructure development. Establishing robust quantum networking protocols and leveraging optical interconnects will be crucial for scaling quantum systems beyond single-processor implementations [195]. Additionally, the development of standardised quantum software frameworks and advanced error mitigation techniques will be instrumental in enhancing accessibility and usability [240]. Beyond technical advancements, the socioeconomic implications of quantum computing warrant careful consideration. The transformative potential of quantum technologies spans multiple industries, with significant applications in cryptography [47] and molecular simulation [48]. Ensuring equitable access to quantum resources and fostering a skilled quantum workforce will be critical in maximising the societal benefits of quantum computing across diverse sectors and regions.
- *Neuromorphic computing*: Inspired by the brain’s architecture, neuromorphic computing is rapidly emerging as a promising solution for achieving energy-efficient, event-driven processing, particularly in AI and ML tasks [53]. Despite its potential, scalability remains a significant hurdle, as building larger neuromorphic systems demands advancements in technological infrastructure, development tools, and integration strategies [59]. Future progress should focus on enhancing the programmability of neuromorphic hardware to enable larger, more complex systems capable of addressing diverse AI and ML workloads [241]. This includes improving the flexibility and accessibility of programming environments to facilitate adoption by a broader range of developers and researchers. In parallel, the development of SNNs as foundational algorithms requires further exploration, particularly in areas such as backpropagation [56] and online learning [242], to enhance their adaptability, scalability, and real-time performance. The practical adoption of neuromorphic hardware faces challenges such as the lack of standardised protocols and the high costs of chip fabrication. Initiatives like Intel’s Loihi 2 platform have demonstrated progress in commercialising neuromorphic computing [65], but broader collaboration among academia, industry, and policymakers will be necessary to standardise frameworks, reduce costs, and accelerate adoption. Integrating neuromorphic computing with photonics presents a promising avenue for addressing key challenges, including scalability, energy efficiency, precision, and standardised performance benchmarks [196]. As the technology evolves, addressing ethical concerns and promoting the responsible

use of brain-inspired systems will be critical [243]. Ensuring equitable access, avoiding misuse, and fostering transparency in neuromorphic applications will help ensure that the technology benefits society responsibly.

- *Optical computing:* The future of optical computing holds transformative potential for meeting the escalating demands of modern computing systems, particularly in AI, telecommunications, and HPC [84]. Advancing this technology requires addressing several critical research challenges through innovative solutions and interdisciplinary collaboration. A key research direction is the development of next-generation photonic integrated circuits, with a particular focus on advancing core components such as MRRs and MZIs [238]. These components will evolve to meet stringent requirements for scalability, efficiency, and reliability. The advancement of all-optical processing presents promising opportunities, including the development of optical gates and logical units, high bit-rate signal processing, and optical quantum computing [89]. High-performance optical interconnects offer significant advantages over traditional electrical interconnects, enabling efficient data transmission in large-scale systems such as data centres, supercomputers, and quantum networks [85]. Industry adoption is already underway, as demonstrated by Google's integration of photonic components in data centres and the emergence of optical neural network research prototypes [88]. In the quantum computing domain, optical components play a crucial role in facilitating high-bandwidth communication between quantum processors, addressing key challenges related to quantum network scalability and efficiency [195]. To accelerate the practical deployment of optical computing systems, research efforts should focus on three key areas: miniaturisation techniques, advanced materials development, and scalable manufacturing processes. These technological advancements are essential for achieving cost-effective, energy-efficient solutions that can expand access to HPC capabilities. This expansion is particularly crucial for small and medium-sized enterprises and academic institutions, which stand to benefit significantly from more accessible advanced computing resources. As optical computing technologies mature, they are poised to revolutionise industries by delivering unprecedented computational power, sustainability, and accessibility. This evolution represents a major step toward meeting the growing computational demands of modern society while aligning with global sustainability goals.
- *Blockchain and distributed ledgers:* Blockchain and DLTs present a decentralised, tamper-resistant way to ensure security and transparency in distributed systems [102]. These technologies eliminate intermediaries and offer immutable transaction records, enabling trustless environments in applications like cloud computing, IoT, and supply chain management. However, challenges such as latency, high energy consumption in Proof-of-Work-based systems, and cold-start delays hinder their scalability and responsiveness. Future research should prioritise the development of scalable blockchain architectures with energy-efficient consensus mechanisms [244]. Innovations such as Proof of Stake and sharding can significantly reduce energy consumption while maintaining robust security and enabling high transaction throughput [245]. These advancements are essential to ensuring blockchain's feasibility in real-time applications and resource-constrained environments. Another promising direction is the creation of tailored blockchain frameworks for specific distributed computing applications. Decentralised file systems, for example, can leverage blockchain to ensure data availability, integrity, and secure sharing [246], while decentralised cloud services can benefit from blockchain's capabilities in managing resource allocation and security [113]. Interoperability among blockchain networks is another key area, requiring standardised protocols and cross-chain communication to enable multi-

platform applications. Practical use cases, such as supply chain management and IoT, already demonstrate blockchain's potential to enhance traceability, secure resource sharing, and improve trust [110]. Efforts to minimise blockchain's environmental impact through energy-efficient mechanisms and green blockchain initiatives further align with global sustainability goals. By addressing these challenges, blockchain and DLTs can revolutionise distributed systems, transforming how data integrity, transparency, and trust are managed across industries.

- *Serverless computing*: Serverless computing, which abstracts infrastructure management and allows developers to focus solely on code execution, is emerging as a transformative paradigm in parallel and distributed systems. By automatically scaling based on demand, serverless architectures are particularly well suited for distributed applications with highly variable workloads, providing cost efficiency, flexibility, and ease of deployment [134]. However, serverless computing faces challenges such as cold-start latency, latency associated with initialising functions, and difficulties in managing stateful, resource-intensive applications [132,135]. Future advancements should address these limitations. Improving the latency and scalability of serverless frameworks is essential, particularly for HPC and real-time distributed systems [132]. Fine-grained resource management techniques and enhanced serverless orchestration mechanisms are needed to efficiently handle parallel tasks across distributed nodes, ensuring optimised workload distribution and responsiveness [236]. Serverless systems show significant potential in AI/ML workflows, enabling seamless deployment of ML models and distributed training pipelines [247]. Their adoption in multi-cloud environments can ensure interoperability across cloud platforms, reducing vendor lock-in and improving resource utilisation [248]. Additionally, techniques like container pre-warming, lightweight virtualisation, and predictive scaling can mitigate cold-start issues, making serverless computing viable for latency-sensitive and resource-constrained environments [132]. By overcoming these challenges, serverless computing can significantly contribute to the evolution of parallel and distributed systems, enabling more scalable, efficient, and adaptable architectures across a wide range of industries.
- *Cloud-native architectures*: Cloud-native architectures are transforming distributed computing by leveraging microservices, containerisation, and orchestration tools like Kubernetes to enable auto-scaling, fault tolerance, and resilience. By decomposing applications into smaller, independent components, these architectures provide flexibility and adaptability, ensuring consistent performance even under varying workload demands [146]. Future advancements should enhance the coordination and orchestration of microservices to ensure data consistency across geographically dispersed cloud resources. For instance, an optimised communication solution has been proposed to enhance inter-service communication in microservices [249]. Synergies with large generative AI models are essential to enable dynamic load balancing between cloud and edge nodes, optimising costs of goods sold and improving resource accessibility [250]. Multi-cloud orchestration initiatives, such as the expansion of the Kubernetes ecosystem [144] and platforms like Google's Anthos [251], demonstrate the feasibility of cross-cloud collaboration for managing complex workloads. Energy efficiency is a critical challenge as cloud-native systems scale. Green computing strategies, such as intelligent container scheduling and life-cycle management, can reduce energy consumption and environmental impact [225]. Additionally, improved container orchestration algorithms that dynamically allocate resources are vital for aligning these architectures with sustainability goals [252]. Security and privacy are paramount due to the decentralised nature of microservices [253], which increases vulnerabilities in

inter-service communication. Robust encryption, authentication, and real-time monitoring are needed to mitigate risks, particularly in sensitive domains like healthcare and finance. By addressing these challenges and fostering synergies with emerging technologies, cloud-native architectures can drive innovation and sustainability across industries such as smart cities, real-time analytics, and scientific research. These systems will remain a cornerstone of distributed computing, delivering efficiency, resilience, and adaptability.

- *Distributed AI and ML:* The future of distributed AI and ML presents transformative opportunities alongside significant technical challenges that require innovative solutions. As distributed workloads grow in scale and complexity, addressing fundamental issues in model synchronisation, communication efficiency, and computational overhead becomes increasingly critical [158,177]. A key research direction is the development of advanced distributed learning frameworks, with a particular emphasis on federated learning architectures, which enable privacy-preserving training across decentralised nodes [171]. These frameworks will evolve to handle heterogeneous data distributions and varying computational capabilities across nodes while maintaining model consistency and performance. Establishing standardised benchmarks for federated learning, particularly in sensitive domains such as healthcare and financial services, will be crucial for validating system robustness and reliability [167]. Such benchmarks should assess not only model accuracy but also critical metrics such as communication efficiency, privacy preservation, and resource utilisation. Another crucial research direction is the advancement of edge AI technologies, which enable sophisticated AI processing at the network edge [160]. This paradigm shift toward edge-centric AI architectures promises significant improvements in latency reduction and bandwidth optimisation, particularly for real-time applications in autonomous systems and IoT networks. Future research should focus on developing lightweight, efficient models capable of operating within the resource constraints of edge devices while maintaining high-performance standards [160]. The integration of distributed AI with emerging computing paradigms opens new avenues for innovation. Hybrid architectures combining classical systems with quantum processors hold promise for solving complex optimisation problems [217], while neuromorphic computing offers potential for energy-efficient, event-driven processing [56]. These integrations require interdisciplinary research efforts to address challenges in cross-platform optimisation, data flow management, and system interoperability. Additionally, the development of standardised interfaces and programming abstractions will be essential to enabling seamless integration across these diverse computing platforms. To fully realise the potential of these advancements, the field should also address broader socio-technical challenges. This includes developing robust frameworks for ethical AI deployment [254], ensuring equitable access to distributed AI resources, and establishing clear guidelines for responsible innovation. The long-term success of distributed AI systems will ultimately depend on balancing technical advancements with practical considerations of cost, scalability, and societal impact.

## 7. Conclusions

This paper has provided a comprehensive overview of parallel and distributed systems, emphasising their pivotal role in meeting the escalating computational demands of modern applications. By exploring their interrelationships and key distinctions, we established a foundation for understanding the emerging trends shaping their evolution. In the domain of parallel systems, we analysed four emerging paradigms: heterogeneous computing,

quantum computing, neuromorphic computing, and optical computing. In the sphere of distributed systems, we examined several emerging trends: blockchain and distributed ledgers, serverless computing, cloud-native architectures, and distributed AI and ML systems. Additionally, we discussed the challenges that persist in these systems, including scalability limitations, security and privacy concerns, fault tolerance, interoperability issues, and energy efficiency demands. Addressing these challenges is crucial for the continued evolution and broader adoption of parallel and distributed systems.

Future research should focus on advancing software frameworks, developing innovative hardware architectures, optimising communication protocols, and designing efficient algorithms. By embracing these emerging trends and proactively tackling associated challenges, we can develop more powerful, efficient, and adaptable computing systems. Such advancements will drive innovation across various sectors, contribute to scientific and technological progress, and meet the complex demands of the future computational landscape.

**Author Contributions:** Conceptualisation, F.D.; methodology, F.D.; validation, F.D., M.A.H., and Y.W.; investigation, F.D., M.A.H., and Y.W.; resources, F.D.; data curation, F.D.; writing—original draft preparation, F.D.; writing—review and editing, F.D., M.A.H., and Y.W.; visualisation, F.D. and M.A.H.; funding acquisition, F.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data sharing is not applicable to this article as no new data were created or analysed in this study.

**Acknowledgments:** The authors sincerely thank the editors of *Electronics* Journal for their valuable support. We are also deeply grateful to the reviewers for their insightful comments and constructive feedback, which have greatly enhanced the quality of this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
2. Das, A.; Palesi, M.; Kim, J.; Pande, P.P. Chip and Package-Scale Interconnects for General-Purpose, Domain-Specific and Quantum Computing Systems—Overview, Challenges and Opportunities. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2024**, *14*, 354–370. [[CrossRef](#)]
3. Michalakes, J. Hpc for weather forecasting. *Parallel Algorithms Comput. Sci. Eng.* **2020**, *2*, 297–323. [[CrossRef](#)]
4. Pronk, S.; Pouya, I.; Lundborg, M.; Rotskoff, G.; Wesen, B.; Kasson, P.M.; Lindahl, E. Molecular simulation workflows as parallel algorithms: The execution engine of Copernicus, a distributed high-performance computing platform. *J. Chem. Theory Comput.* **2015**, *11*, 2600–2608. [[CrossRef](#)] [[PubMed](#)]
5. Scellato, S.; Mascolo, C.; Musolesi, M.; Crowcroft, J. Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades. In Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India, 28 March–1 April 2011; pp. 457–466.
6. Sharma, R.; Singh, A. Blockchain Technologies and Call for an Open Financial System: Decentralised Finance. In *Decentralized Finance and Tokenization in FinTech*; IGI Global: Hershey, PA, USA, 2024; pp. 21–32.
7. Raj, K.B.; Mehta, K.; Siddi, S.; Sharma, M.; Sharma, D.K.; Adhav, S.; González, J.L. Optimizing Financial Transactions and Processes Through the Power of Distributed Systems. In *Meta Heuristic Algorithms for Advanced Distributed Systems*; Wiley Online Library: Hoboken, NJ, USA, 2024; pp. 289–303.
8. Hockney, R.W.; Jesshope, C.R. *Parallel Computers 2: Architecture, Programming and Algorithms*; CRC Press: Boca Raton, FL, USA, 2019.
9. Navarro, C.A.; Hitschfeld-Kahler, N.; Mateu, L. A survey on parallel computing and its applications in data-parallel problems using GPU architectures. *Commun. Comput. Phys.* **2014**, *15*, 285–329. [[CrossRef](#)]
10. Farooq, U.; Marrakchi, Z.; Mehrez, H.; Farooq, U.; Marrakchi, Z.; Mehrez, H. FPGA architectures: An overview. In *Tree-Based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization*; Springer: Midtown Manhattan, NY, USA, 2012; pp. 7–48.

11. Halsted, D. The origins of the architectural metaphor in computing: Design and technology at IBM, 1957–1964. *IEEE Ann. Hist. Comput.* **2018**, *40*, 61–70. [[CrossRef](#)]
12. Chawan, M.P.; Patle, B.; Cholake, V.; Pardeshi, S. Parallel Computer Architectural Schemes. *Int. J. Eng. Res. Technol.* **2012**, *1*, 9.
13. Batcher. Design of a massively parallel processor. *IEEE Trans. Comput.* **1980**, *100*, 836–840.
14. Leiserson, C.E.; Abuhamdeh, Z.S.; Douglas, D.C.; Feynman, C.R.; Ganmukhi, M.N.; Hill, J.V.; Hillis, D.; Kuszmaul, B.C.; St. Pierre, M.A.; Wells, D.S.; et al. The network architecture of the Connection Machine CM-5. In Proceedings of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures, San Diego, CA, USA, 29 June–1 July 1992; pp. 272–285.
15. Alverson, B.; Froese, E.; Kaplan, L.; Roweth, D. *Cray XC Series Network*; White Paper WP-Aries01-1112; Cray Inc.: Seattle, WA, USA, 2012.
16. Keckler, S.W.; Hofstee, H.P.; Olukotun, K. *Multicore Processors and Systems*; Springer: Berlin/Heidelberg, Germany, 2009.
17. McClanahan, C. History and evolution of gpu architecture. *Surv. Pap.* **2010**, *9*, 1–7.
18. Kshemkalyani, A.D.; Singhal, M. *Distributed Computing: Principles, Algorithms, and Systems*; Cambridge University Press: Cambridge, UK, 2011.
19. Ali, M.F.; Khan, R.Z. Distributed computing: An overview. *Int. J. Adv. Netw. Appl.* **2015**, *7*, 2630.
20. Paloque-Bergès, C.; Schafer, V. Arpanet (1969–2019). *Internet Hist.* **2019**, *3*, 1–14. [[CrossRef](#)]
21. Bonifati, A.; Chrysanthis, P.K.; Ouksel, A.M.; Sattler, K.U. Distributed databases and peer-to-peer databases: Past and present. *Acm Sigmod Rec.* **2008**, *37*, 5–11. [[CrossRef](#)]
22. Oluwatosin, H.S. Client-server model. *Iosr J. Comput. Eng.* **2014**, *16*, 67–71. [[CrossRef](#)]
23. Qian, L.; Luo, Z.; Du, Y.; Guo, L. Cloud computing: An overview. In Proceedings of the Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, 1–4 December 2009; Proceedings 1; Springer: Berlin/Heidelberg, Germany, 2009; pp. 626–631.
24. Dittrich, J.; Quiané-Ruiz, J.A. Efficient big data processing in Hadoop MapReduce. *Proc. Vldb Endow.* **2012**, *5*, 2014–2015. [[CrossRef](#)]
25. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [[CrossRef](#)]
26. Madakam, S.; Ramaswamy, R.; Tripathi, S. Internet of Things (IoT): A literature review. *J. Comput. Commun.* **2015**, *3*, 164–173. [[CrossRef](#)]
27. Roosta, S.H. *Parallel Processing and Parallel Algorithms: Theory and Computation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
28. Burns, B. *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2018.
29. Parhami, B. *Introduction to Parallel Processing: Algorithms and Architectures*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999; Volume 1.
30. Santoro, G.; Turvani, G.; Graziano, M. New logic-in-memory paradigms: An architectural and technological perspective. *Micromachines* **2019**, *10*, 368. [[CrossRef](#)]
31. Ben-Nun, T.; Hoefler, T. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *Acm Comput. Surv. Csur* **2019**, *52*, 1–43. [[CrossRef](#)]
32. Asaduzzaman, A.; Trent, A.; Osborne, S.; Aldershof, C.; Sibai, F.N. Impact of CUDA and OpenCL on parallel and distributed computing. In Proceedings of the 2021 8th International Conference on Electrical and Electronics Engineering (ICEEE), Antalya, Turkey, 9–11 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 238–242.
33. Fang, J.; Huang, C.; Tang, T.; Wang, Z. Parallel programming models for heterogeneous many-cores: A comprehensive survey. *CCF Trans. High Perform. Comput.* **2020**, *2*, 382–400. [[CrossRef](#)]
34. Prasad, A.; Muzio, C.; Ton, P.; Razdaan, S. Advanced 3D Packaging of 3.2 Tbs Optical Engine for Co-packaged Optics (CPO) in Hyperscale Data Center Networks. In Proceedings of the 2024 IEEE 74th Electronic Components and Technology Conference (ECTC), Denver, CO, USA, 28–31 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 101–106.
35. Horowitz, M.; Grumbling, E. *Quantum Computing: Progress and Prospects*; The National Academies Press: Washington, DC, USA, 2019.
36. Huang, H.L.; Wu, D.; Fan, D.; Zhu, X. Superconducting quantum computing: A review. *Sci. China Inf. Sci.* **2020**, *63*, 180501. [[CrossRef](#)]
37. Yang, Z.; Zolanvari, M.; Jain, R. A survey of important issues in quantum computing and communications. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1059–1094. [[CrossRef](#)]
38. Radtke, T.; Fritzsche, S. Simulation of n-qubit quantum systems. I. Quantum registers and quantum gates. *Comput. Phys. Commun.* **2005**, *173*, 91–113. [[CrossRef](#)]
39. Mosca, M. Quantum Computer Algorithms. Ph.D. Thesis, University of Oxford, Oxford, UK, 1999.
40. Nofer, M.; Bauer, K.; Hinz, O.; van der Aalst, W.; Weinhardt, C. Quantum Computing. *Bus. Inf. Syst. Eng.* **2023**, *65*, 361–367. [[CrossRef](#)]

41. Gonzalez-Zalba, M.; De Franceschi, S.; Charbon, E.; Meunier, T.; Vinet, M.; Dzurak, A. Scaling silicon-based quantum computing using CMOS technology. *Nat. Electron.* **2021**, *4*, 872–884. [[CrossRef](#)]
42. Schäfer, V.; Ballance, C.; Thirumalai, K.; Stephenson, L.; Ballance, T.; Steane, A.; Lucas, D. Fast quantum logic gates with trapped-ion qubits. *Nature* **2018**, *555*, 75–78. [[CrossRef](#)] [[PubMed](#)]
43. Graham, T.; Song, Y.; Scott, J.; Poole, C.; Phuttitarn, L.; Jooya, K.; Eichler, P.; Jiang, X.; Marra, A.; Grinkemeyer, B.; et al. Multi-qubit entanglement and algorithms on a neutral-atom quantum computer. *Nature* **2022**, *604*, 457–462. [[CrossRef](#)] [[PubMed](#)]
44. Chu, Y.; Lukin, M.D. Quantum optics with nitrogen-vacancy centers in diamond. In *Quantum Optics and Nanophotonics*; Harvard of University: Cambridge, MA, USA, 2015; pp. 229–270.
45. Lukens, J.M.; Lougovski, P. Frequency-encoded photonic qubits for scalable quantum information processing. *Optica* **2017**, *4*, 8–16. [[CrossRef](#)]
46. AbuGhanem, M. IBM quantum computers: Evolution, performance, and future directions. *arXiv* **2024**, arXiv:2410.00916.
47. Easttom, C. Quantum computing and cryptography. In *Modern Cryptography: Applied Mathematics for Encryption and Information Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 397–407.
48. Blunt, N.S.; Camps, J.; Crawford, O.; Izsák, R.; Leontica, S.; Mirani, A.; Moylett, A.E.; Scivier, S.A.; Sunderhauf, C.; Schopf, P.; et al. Perspective on the current state-of-the-art of quantum computing for drug discovery applications. *J. Chem. Theory Comput.* **2022**, *18*, 7001–7023. [[CrossRef](#)]
49. Bauer, B.; Bravyi, S.; Motta, M.; Chan, G.K.L. Quantum algorithms for quantum chemistry and quantum materials science. *Chem. Rev.* **2020**, *120*, 12685–12717. [[CrossRef](#)] [[PubMed](#)]
50. Ollitrault, P.J.; Miessen, A.; Tavernelli, I. Molecular quantum dynamics: A quantum computing perspective. *Accounts Chem. Res.* **2021**, *54*, 4229–4238. [[CrossRef](#)]
51. Orús, R.; Mugel, S.; Lizaso, E. Quantum computing for finance: Overview and prospects. *Rev. Phys.* **2019**, *4*, 100028. [[CrossRef](#)]
52. Cerezo, M.; Verdon, G.; Huang, H.Y.; Cincio, L.; Coles, P.J. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.* **2022**, *2*, 567–576. [[CrossRef](#)]
53. Mead, C. Neuromorphic electronic systems. *Proc. IEEE* **1990**, *78*, 1629–1636. [[CrossRef](#)]
54. Schuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* **2022**, *2*, 10–19. [[CrossRef](#)]
55. Shrestha, A.; Ahmed, K.; Wang, Y.; Qiu, Q. Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1999–2006.
56. Esser, S.K.; Appuswamy, R.; Merolla, P.; Arthur, J.V.; Modha, D.S. Backpropagation for energy-efficient neuromorphic computing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. Available online: [https://proceedings.neurips.cc/paper\\_files/paper/2015/hash/10a5ab2db37feedfdeaab192ead4ac0e-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2015/hash/10a5ab2db37feedfdeaab192ead4ac0e-Abstract.html) (accessed on 7 January 2025).
57. Alom, M.Z.; Van Essen, B.; Moody, A.T.; Widemann, D.P.; Taha, T.M. Quadratic unconstrained binary optimization (QUBO) on neuromorphic computing system. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3922–3929.
58. Corder, K.; Monaco, J.V.; Vindiola, M.M. Solving vertex cover via ising model on a neuromorphic processor. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5.
59. Furber, S. Large-scale neuromorphic computing systems. *J. Neural Eng.* **2016**, *13*, 051001. [[CrossRef](#)] [[PubMed](#)]
60. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [[CrossRef](#)]
61. Benjamin, B.V.; Gao, P.; McQuinn, E.; Choudhary, S.; Chandrasekaran, A.R.; Bussat, J.M.; Alvarez-Icaza, R.; Arthur, J.V.; Merolla, P.A.; Boahen, K. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* **2014**, *102*, 699–716. [[CrossRef](#)]
62. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The spinnaker project. *Proc. IEEE* **2014**, *102*, 652–665. [[CrossRef](#)]
63. Davies, M.; Wild, A.; Orchard, G.; Sandamirskaya, Y.; Guerra, G.A.F.; Joshi, P.; Plank, P.; Risbud, S.R. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proc. IEEE* **2021**, *109*, 911–934. [[CrossRef](#)]
64. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [[CrossRef](#)]
65. Davies, M. Taking neuromorphic computing to the next level with Loihi2. *Intel Labs' Loihi* **2021**, *2*, 1–7.
66. Zenke, F.; Neftci, E.O. Brain-inspired learning on neuromorphic substrates. *Proc. IEEE* **2021**, *109*, 935–950. [[CrossRef](#)]
67. Brayshaw, G.; Ward-Cherrier, B.; Pearson, M.J. Simultaneous Velocity and Texture Classification from a Neuromorphic Tactile Sensor Using Spiking Neural Networks. *Electronics* **2024**, *13*, 2159. [[CrossRef](#)]

68. Smith, H.; Seekings, J.; Mohammadi, M.; Zand, R. Realtime Facial Expression Recognition: Neuromorphic Hardware vs. Edge AI Accelerators. In Proceedings of the 2023 International Conference on Machine Learning and Applications (ICMLA), Jacksonville, FL, USA, 15–17 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1547–1552.
69. Bezugam, S.S.; Shaban, A.; Suri, M. Neuromorphic recurrent spiking neural networks for emg gesture classification and low power implementation on loihi. In Proceedings of the 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 21–25 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
70. Frady, E.P.; Sommer, F.T. Robust computation with rhythmic spike patterns. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 18050–18059. [[CrossRef](#)]
71. Yakopcic, C.; Rahman, N.; Atahary, T.; Taha, T.M.; Beigh, A.; Douglass, S. High speed cognitive domain ontologies for asset allocation using loihi spiking neurons. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
72. Kay, B.; Date, P.; Schuman, C. Neuromorphic graph algorithms: Extracting longest shortest paths and minimum spanning trees. In *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–6.
73. Cong, G.; Lim, S.H.; Kulkarni, S.; Date, P.; Potok, T.; Snyder, S.; Parsa, M.; Schuman, C. Semi-supervised graph structure learning on neuromorphic computers. In Proceedings of the International Conference on Neuromorphic Systems, Knoxville, TN, USA, 27–29 July 2022; pp. 1–4.
74. Aimone, J.B.; Date, P.; Fonseca-Guerra, G.A.; Hamilton, K.E.; Henke, K.; Kay, B.; Kenyon, G.T.; Kulkarni, S.R.; Mniszewski, S.M.; Parsa, M.; et al. A review of non-cognitive applications for neuromorphic computing. *Neuromorphic Comput. Eng.* **2022**, *2*, 032003. [[CrossRef](#)]
75. Izhikevich, E.M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **2003**, *14*, 1569–1572. [[CrossRef](#)] [[PubMed](#)]
76. Gerstner, W.; Kistler, W.M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*; Cambridge University Press: Cambridge, UK, 2002.
77. Box, G.E.; Tiao, G.C. *Bayesian Inference in Statistical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
78. Cortes, C. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
79. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223.
80. Destrycker, N.; Benoot, W.; Mattias, J.; Rodriguez, I.; Steenari, D. EDGX-1: A New Frontier in Onboard AI Computing with a Heterogeneous and Neuromorphic Design. In Proceedings of the 2023 European Data Handling & Data Processing Conference (EDHPC), Juan-Les-Pins, France, 2–6 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
81. Ng, L.K.; Chow, S.S. {GForce}::{GPU-Friendly} oblivious and rapid neural network inference. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Online, 11–13 August 2021; pp. 2147–2164.
82. Luo, T.; Wong, W.F.; Goh, R.S.M.; Do, A.T.; Chen, Z.; Li, H.; Jiang, W.; Yau, W. Achieving green ai with energy-efficient deep learning using neuromorphic computing. *Commun. AcM* **2023**, *66*, 52–57. [[CrossRef](#)]
83. Guenther, B.D.; Steel, D. *Encyclopedia of Modern Optics*; Academic Press: Cambridge, MA, USA, 2018.
84. Kazanskiy, N.L.; Butt, M.A.; Khonina, S.N. Optical computing: Status and perspectives. *Nanomaterials* **2022**, *12*, 2171. [[CrossRef](#)]
85. Mekawey, H.; Elsayed, M.; Ismail, Y.; Swillam, M.A. Optical interconnects finally seeing the light in silicon photonics: Past the hype. *Nanomaterials* **2022**, *12*, 485. [[CrossRef](#)] [[PubMed](#)]
86. Xia, C.; Chen, Y.; Zhang, H.; Zhang, H.; Dai, F.; Wu, J. Efficient neural network accelerators with optical computing and communication. *Comput. Sci. Inf. Syst.* **2023**, *20*, 513–535. [[CrossRef](#)]
87. Ning, S.; Zhu, H.; Feng, C.; Gu, J.; Jiang, Z.; Ying, Z.; Midkiff, J.; Jain, S.; Hlaing, M.H.; Pan, D.Z.; et al. Photonic-electronic integrated circuits for high-performance computing and ai accelerators. *J. Light. Technol.* **2024**, *42*, 7834–7859. [[CrossRef](#)]
88. Cheng, Q.; Bahadori, M.; Glick, M.; Rumley, S.; Bergman, K. Recent advances in optical technologies for data centers: A review. *Optica* **2018**, *5*, 1354–1370. [[CrossRef](#)]
89. Minzioni, P.; Lacava, C.; Tanabe, T.; Dong, J.; Hu, X.; Csaba, G.; Porod, W.; Singh, G.; Willner, A.E.; Almaini, A.; et al. Roadmap on all-optical processing. *J. Opt.* **2019**, *21*, 063001. [[CrossRef](#)]
90. Wu, J.; Lin, X.; Guo, Y.; Liu, J.; Fang, L.; Jiao, S.; Dai, Q. Analog optical computing for artificial intelligence. *Engineering* **2022**, *10*, 133–145. [[CrossRef](#)]
91. Li, X.; Shao, Z.; Zhu, M.; Yang, J. *Fundamentals of Optical Computing Technology*; Springer: Singapore 2018.
92. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Satoshi Nakamoto: Online, 2008. Available online: <https://www.bitcoin.org> (accessed on 7 December 2024).
93. Chaum, D. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*; Springer: Boston, MA, USA, 1983.
94. Haber, S.; Stornetta, W.S. *How to Time-Stamp a Digital Document*; Springer: Berlin/Heidelberg, Germany, 1991.

95. Finney, H. *RPOW—Reusable Proofs of Work*; Nakamoto Institute: Online, 2004. Available online: <https://nakamotoinstitute.org/finney/rpow/index.html> (accessed on 7 December 2024).
96. Mougayar, W. *The Business Blockchain: Promise, Practice, and Application of the next INTERNET Technology*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
97. Zhao, J.L.; Fan, S.; Yan, J. Overview of business innovations and research opportunities in blockchain and introduction to the special issue. *Financ. Innov.* **2016**, *2*, 28. [[CrossRef](#)]
98. Tripathi, G.; Ahad, M.A.; Casalino, G. A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. *Decis. Anal. J.* **2023**, *9*, 100344. [[CrossRef](#)]
99. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. In *Concurrency: The Works of Leslie Lamport*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 203–226.
100. Alrubei, S.; Ball, E.; Rigelsford, J. Securing IoT-blockchain applications through honesty-based distributed proof of authority consensus algorithm. In Proceedings of the 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Virtual, 14–18 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
101. Garg, N.; Nehra, A.; Baza, M.; Kumar, N. Secure and efficient data integrity verification scheme for cloud data storage. In Proceedings of the 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
102. Benčić, F.M.; Žarko, I.P. Distributed ledger technology: Blockchain compared to directed acyclic graph. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–5 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1569–1570.
103. Reegu, F.A.; Abas, H.; Gulzar, Y.; Xin, Q.; Alwan, A.A.; Jabbari, A.; Sonkamble, R.G.; Dziauddin, R.A. Blockchain-based framework for interoperable electronic health records for an improved healthcare system. *Sustainability* **2023**, *15*, 6337. [[CrossRef](#)]
104. Li, J.; Luo, X.; Lei, H. TrustHealth: Enhancing eHealth Security with Blockchain and Trusted Execution Environments. *Electronics* **2024**, *13*, 2425. [[CrossRef](#)]
105. Maesa, D.D.F.; Tietze, F. Automating Intellectual Property License Agreements with DLT. In Proceedings of the 2023 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Athens, Greece, 17–20 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 48–56.
106. Ito, K.; O’Dair, M. A critical examination of the application of blockchain technology to intellectual property management. In *Business Transformation Through Blockchain: Volume II*; Palgrave Macmillan: Cham, Switzerland, 2019; pp. 317–335.
107. Xu, R.; Zhu, J.; Yang, L.; Lu, Y.; Xu, L.D. Decentralized finance (DeFi): A paradigm shift in the Fintech. *Enterp. Inf. Syst.* **2024**, *18*, 2397630. [[CrossRef](#)]
108. Daugaard, T.V.; Jensen, J.B.; Kauffman, R.J.; Kim, K. Blockchain solutions with consensus algorithms and immediate finality: Toward Panopticon-style monitoring to enhance anti-money laundering. *Electron. Commer. Res. Appl.* **2024**, *65*, 101386. [[CrossRef](#)]
109. Rijanto, A. *Blockchain Technology Roles to Overcome Accounting, Accountability and Assurance Barriers in Supply Chain Finance*; Asian Review of Accounting; Emerald Group Publishing Ltd.: Bingley, UK, 2024.
110. Trollman, H.; Garcia-Garcia, G.; Jagtap, S.; Trollman, F. Blockchain for ecologically embedded coffee supply chains. *Logistics* **2022**, *6*, 43. [[CrossRef](#)]
111. Herbe, A.; Estermann, Z.; Holzwarth, V.; vom Brocke, J. How to effectively use distributed ledger technology in supply chain management? *Int. J. Prod. Res.* **2024**, *62*, 2522–2547. [[CrossRef](#)]
112. Odimarha, A.C.; Ayodeji, S.A.; Abaku, E.A. The role of technology in supply chain risk management: Innovations and challenges in logistics. *Magna Sci. Adv. Res. Rev.* **2024**, *10*, 138–145. [[CrossRef](#)]
113. Selvarajan, S.; Shankar, A.; Uddin, M.; Alqahtani, A.S.; Al-Shehari, T.; Viriyasitavat, W. A smart decentralized identifiable distributed ledger technology-based blockchain (DIDLT-BC) model for cloud-IoT security. *Expert Syst.* **2024**, *42*, e13544. [[CrossRef](#)]
114. Fitzpatrick, P.; Thorpe, C. An Investigation into the Feasibility of using Distributed Digital Ledger technology for Digital Forensics for Industrial IoT. In Proceedings of the European Conference on Cyber Warfare and Security, Jyväskylä, Finland, 27–28 June 2024; Volume 23, pp. 827–835.
115. Ling, X.; Le, Y.; Zhang, B.; Wang, J.; Gao, X. Blockchain-Enhanced Open Internet of Things Access Architecture. U.S. Patent 11,954,681, 9 April 2024.
116. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MedShare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [[CrossRef](#)]
117. Fan, K.; Wang, S.; Ren, Y.; Li, H.; Yang, Y. Medblock: Efficient and secure medical data sharing via blockchain. *J. Med. Syst.* **2018**, *42*, 136. [[CrossRef](#)]
118. Abdelgalil, L.; Mejri, M. HealthBlock: A framework for a collaborative sharing of electronic health records based on blockchain. *Future Internet* **2023**, *15*, 87. [[CrossRef](#)]
119. Johari, R.; Kumar, V.; Gupta, K.; Vidyarthi, D.P. BLOSOM: BLockchain technology for Security of Medical records. *Ict Express* **2022**, *8*, 56–60. [[CrossRef](#)]

120. Li, Y.; Lin, Y.; Wang, Y.; Ye, K.; Xu, C. Serverless computing: State-of-the-art, challenges and opportunities. *IEEE Trans. Serv. Comput.* **2022**, *16*, 1522–1539. [[CrossRef](#)]
121. Serverless Function, FaaS Serverless—AWS Lambda. Available online: <https://aws.amazon.com/lambda/> (accessed on 31 October 2024).
122. Patil, R.; Chaudhery, T.S.; Qureshi, M.A.; Sawant, V.; Dalvi, H. Serverless computing and the emergence of function-as-a-service. In Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 27–28 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 764–769.
123. Sadowski, M.; Frantzell, L.; Sadowski, M.; Frantzell, L. Apache OpenWhisk—Open Source Project. In *Serverless Swift: Apache OpenWhisk for iOS Developers*; Apress: Berkeley, CA, USA, 2020; pp. 37–57.
124. Kumar, V.; Agnihotri, K. *Serverless Computing Using Azure Functions: Build, Deploy, Automate, and Secure Serverless Application Development with Azure Functions (English Edition)*; BPB Publications: Noida, India, 2021.
125. Bisong, E.; Bisong, E. An overview of google cloud platform services. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*; Apress: Berkeley, CA, USA, 2019; pp. 7–10.
126. Rajan, R.A.P. Serverless architecture—a revolution in cloud computing. In Proceedings of the 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 13–15 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 88–93.
127. Sok, C.; d’Orazio, L.; Tekin, R.; Tombroff, D. WebAssembly serverless join: A Study of its Application. In Proceedings of the 36th International Conference on Scientific and Statistical Database Management, Rennes France, 10–12 July 2024; pp. 1–4.
128. Ouyang, R.; Wang, J.; Xu, H.; Chen, S.; Xiong, X.; Tolba, A.; Zhang, X. A Microservice and Serverless Architecture for Secure IoT System. *Sensors* **2023**, *23*, 4868. [[CrossRef](#)]
129. Werner, S.; Tai, S. A reference architecture for serverless big data processing. *Future Gener. Comput. Syst.* **2024**, *155*, 179–192. [[CrossRef](#)]
130. Guhan, T.; Sekhar, G.C.; Revathy, N.; Baranidharan, K.; Aancy, H.M. Financial and Economic Analysis on Serverless Computing System Services. In *Essential Information Systems Service Management*; IGI Global: Hershey, PA, USA, 2025; pp. 83–112.
131. Lee, C.; Zhu, Z.; Yang, T.; Huo, Y.; Su, Y.; He, P.; Lyu, M.R. SPES: Towards Optimizing Performance-Resource Trade-Off for Serverless Functions. *arXiv* **2024**, arXiv:2403.17574.
132. Golec, M.; Walia, G.K.; Kumar, M.; Cuadrado, F.; Gill, S.S.; Uhlig, S. Cold start latency in serverless computing: A systematic review, taxonomy, and future directions. *Acm Comput. Surv.* **2024**, *57*, 1–36. [[CrossRef](#)]
133. Karamzadeh, A.; Shameli-Sendi, A. Reducing cold start delay in serverless computing using lightweight virtual machines. *J. Netw. Comput. Appl.* **2024**, *232*, 104030. [[CrossRef](#)]
134. Lyu, S.; Rzeznik, A. Going Serverless with the Amazon AWS Rust SDK. In *Practical Rust Projects: Build Serverless, AI, Machine Learning, Embedded, Game, and Web Applications*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 201–246.
135. Aslanpour, M.S.; Toosi, A.N.; Cicconetti, C.; Javadi, B.; Sbarski, P.; Taibi, D.; Assuncao, M.; Gill, S.S.; Gaire, R.; Dustdar, S. Serverless edge computing: Vision and challenges. In Proceedings of the 2021 Australasian Computer Science Week Multiconference, Dunedin, New Zealand, 1–5 February 2021; pp. 1–10.
136. Lekkala, C. AI-Driven Dynamic Resource Allocation in Cloud Computing: Predictive Models and Real-Time Optimization. *J. Artif. Intell. Mach. Learn. Data Sci.* **2024**, *2*, 450–456. [[CrossRef](#)]
137. Grafberger, A.; Chadha, M.; Jindal, A.; Gu, J.; Gerndt, M. Fedless: Secure and scalable federated learning using serverless computing. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 164–173.
138. Li, X.; Leng, X.; Chen, Y. Securing serverless computing: Challenges, solutions, and opportunities. *IEEE Netw.* **2022**, *37*, 166–173. [[CrossRef](#)]
139. Liu, F.; Niu, Y. Demystifying the cost of serverless computing: Towards a win-win deal. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *35*, 59–72. [[CrossRef](#)]
140. Kosińska, J.; Baliś, B.; Konieczny, M.; Malawski, M.; Zieliński, S. Toward the observability of cloud-native applications: The overview of the state-of-the-art. *IEEE Access* **2023**, *11*, 73036–73052. [[CrossRef](#)]
141. Fazuludeen, N.; Banu, S.S.; Gupta, A.; Swathi, V. Challenges And Issues of Managing the Virtualization Environment Through Vmware Vsphere. *Nanotechnol. Perceptions* **2024**, *20*, 281–292.
142. Krishna Kaiser, A. Introduction to devops. In *Reinventing ITIL® and DevOps with Digital Transformation: Essential Guidance to Accelerate the Process*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 3–38.
143. Pandey, B.; Mishra, A.K.; Yadav, A.; Tiwari, D.; Pandey, M.S. Virtualization using Docker container. In *Emerging Real-World Applications of Internet of Things*; CRC Press: Boca Raton, FL, USA, 2022; pp. 157–181.
144. Hightower, K.; Burns, B.; Beda, J. *Kubernetes: Up and Running Dive into the Future of Infrastructure*; Oreilly Media Inc.: Sebastopol, CA, USA, 2017.
145. Balalaie, A.; Heydarnoori, A.; Jamshidi, P. Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Softw.* **2016**, *33*, 42–52. [[CrossRef](#)]

146. Pahl, C.; Brogi, A.; Soldani, J.; Jamshidi, P. Cloud container technologies: A state-of-the-art review. *IEEE Trans. Cloud Comput.* **2017**, *7*, 677–692. [[CrossRef](#)]
147. Bernstein, D. Containers and cloud: From lxc to docker to kubernetes. *IEEE Cloud Comput.* **2014**, *1*, 81–84. [[CrossRef](#)]
148. Oyeniran, C.; Adewusi, A.O.; Adeleke, A.G.; Akwawa, L.A.; Azubuko, C.F. Microservices architecture in cloud-native applications: Design patterns and scalability. *Comput. Sci. IT Res. J.* **2024**, *5*, 2107–2124. [[CrossRef](#)]
149. Bharadwaj, D.; Premananda, B. Transition of cloud computing from traditional applications to the cloud native approach. In Proceedings of the 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijayapura, India, 20–21 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
150. Goniwada, S.R.; Goniwada, S.R. Enterprise Cloud Native Software Engineering. In *Cloud Native Architecture and Design: A Handbook for Modern Day Architecture and Design with Enterprise-Grade Examples*; Apress: Berkeley, CA, USA, 2022; pp. 495–522.
151. Farreras, M.; Almási, G.; Cascaval, C.; Cortes, T. Scalable RDMA performance in PGAS languages. In Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, Rome, Italy, 23–29 May 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–12.
152. Gil, G.; Corujo, D.; Pedreiras, P. Cloud native computing for industry 4.0: Challenges and opportunities. In Proceedings of the 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 7–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 01–04.
153. Alonso, J.; Orue-Echevarria, L.; Casola, V.; Torre, A.I.; Huarte, M.; Osaba, E.; Lobo, J.L. Understanding the challenges and novel architectural models of multi-cloud native applications—A systematic literature review. *J. Cloud Comput.* **2023**, *12*, 6. [[CrossRef](#)]
154. Pham, X.Q.; Nguyen, T.D.; Huynh-The, T.; Huh, E.N.; Kim, D.S. Distributed cloud computing: Architecture, enabling technologies, and open challenges. *IEEE Consum. Electron. Mag.* **2022**, *12*, 98–106. [[CrossRef](#)]
155. Deng, S.; Zhao, H.; Huang, B.; Zhang, C.; Chen, F.; Deng, Y.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Cloud-native computing: A survey from the perspective of services. *Proc. IEEE* **2024**, *112*, 12–46. [[CrossRef](#)]
156. Sana, K.; Hassina, N.; Kadda, B.B. Towards a reference architecture for interoperable clouds. In Proceedings of the 2021 8th International Conference on Electrical and Electronics Engineering (ICEEE), Antalya, Turkey, 9–11 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 229–233.
157. Tankov, V.; Valchuk, D.; Golubev, Y.; Bryksin, T. Infrastructure in code: Towards developer-friendly cloud applications. In Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 15–19 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1166–1170.
158. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A survey on distributed machine learning. *Acm Comput. Surv. Csur* **2020**, *53*, 1–33. [[CrossRef](#)]
159. Dai, F.; Chen, Y.; Huang, Z.; Zhang, H. Wrht: Efficient all-reduce for distributed DNN training in optical interconnect systems. In Proceedings of the 52nd International Conference on Parallel Processing, Salt Lake City, UT, USA, 7–10 August 2023; pp. 556–565.
160. Filho, C.P.; Marques, E., Jr.; Chang, V.; Dos Santos, L.; Bernardini, F.; Pires, P.F.; Ochi, L.; Delicato, F.C. A systematic literature review on distributed machine learning in edge computing. *Sensors* **2022**, *22*, 2665. [[CrossRef](#)] [[PubMed](#)]
161. Hudson, N.; Hossain, M.J.; Hosseinzadeh, M.; Khamfroush, H.; Rahnamay-Naeini, M.; Ghani, N. A framework for edge intelligent smart distribution grids via federated learning. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–9.
162. Reijonen, J.; Opsenica, M.; Morabito, R.; Komu, M.; Elmusrati, M. Regression training using model parallelism in a distributed cloud. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 741–747.
163. Majumdar, S.; Trivisonno, R.; Poe, W.Y.; Carle, G. Distributing intelligence for 6G network automation: Performance and architectural impact. In Proceedings of the ICC 2023-IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 6224–6229.
164. Gholizadeh, N.; Musilek, P. Distributed learning applications in power systems: A review of methods, gaps, and challenges. *Energies* **2021**, *14*, 3654. [[CrossRef](#)]
165. Chen, X.; Wujek, B. Autodal: Distributed active learning with automatic hyperparameter selection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3537–3544.
166. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; PMLR:W&CP; pp. 1273–1282.
167. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [[CrossRef](#)]

168. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 1175–1191.
169. Geyer, R.C.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.
170. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; pp. 1–11.
171. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
172. Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; ACM: New York, NY, USA, 2015; pp. 1310–1321.
173. Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, W.J. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv* **2017**, arXiv:1712.01887.
174. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1709–1720.
175. Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Ranzato, M.; Senior, A.; Tucker, P.; Yang, K.; et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*; 2012 ; Volume 25. Available online: [https://proceedings.neurips.cc/paper\\_files/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html) (accessed on 7 November 2024).
176. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2224–2287. [[CrossRef](#)]
177. Cao, X.; Başar, T.; Diggavi, S.; Eldar, Y.C.; Letaief, K.B.; Poor, H.V.; Zhang, J. Communication-efficient distributed learning: An overview. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 851–873. [[CrossRef](#)]
178. Chouvardas, S.; Slavakis, K.; Kopsinis, Y.; Theodoridis, S. A sparsity promoting adaptive algorithm for distributed learning. *IEEE Trans. Signal Process.* **2012**, *60*, 5412–5425. [[CrossRef](#)]
179. Zhang, S.; Zhang, C.; You, Z.; Zheng, R.; Xu, B. Asynchronous stochastic gradient descent for DNN training. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 6660–6663.
180. Du, Y.; You, K. Asynchronous stochastic gradient descent over decentralized datasets. *IEEE Trans. Control. Netw. Syst.* **2021**, *8*, 1212–1224. [[CrossRef](#)]
181. Ko, Y.; Choi, K.; Seo, J.; Kim, S.W. An in-depth analysis of distributed training of deep neural networks. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Portland, OR, USA, 17–21 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 994–1003.
182. Mahmoudi, A.; Ghadikolaei, H.S.; Fischione, C. Cost-efficient distributed optimization in machine learning over wireless networks. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
183. Giuseppi, A.; Della Torre, L.; Menegatti, D.; Pietrabissa, A. AdaFed: Performance-based adaptive federated learning. In Proceedings of the the 5th International Conference on Advances in Artificial Intelligence, Virtual, 20–22 November 2021; pp. 38–43.
184. Byrd, D.; Polychroniadou, A. Differentially private secure multi-party computation for federated learning in financial applications. In Proceedings of the First ACM International Conference on AI in Finance, New York, NY, USA, 15–16 October 2020; pp. 1–9.
185. Xu, X.; Liu, W.; Zhang, Y.; Zhang, X.; Dou, W.; Qi, L.; Bhuiyan, M.Z.A. Psdf: Privacy-aware iov service deployment with federated learning in cloud-edge computing. *Acm Trans. Intell. Syst. Technol. Tist* **2022**, *13*, 1–22. [[CrossRef](#)]
186. Qiao, D.; Li, M.; Guo, S.; Zhao, J.; Xiao, B. Resources-efficient Adaptive Federated Learning for Digital Twin-Enabled IIoT. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 3639–3652. [[CrossRef](#)]
187. Ao, Y.; Wu, Z.; Yu, D.; Gong, W.; Kui, Z.; Zhang, M.; Ye, Z.; Shen, L.; Ma, Y.; Wu, T.; et al. End-to-end adaptive distributed training on paddlepaddle. *arXiv* **2021**, arXiv:2112.02752.
188. Jin, Y.; Wang, H.; Tang, X.; Hoefler, T.; Liu, X.; Zhai, J. Identifying scalability bottlenecks for large-scale parallel programs with graph analysis. In Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, CA, USA, 22–26 February 2020; pp. 409–410.
189. Jiang, Z.; Lin, H.; Zhong, Y.; Huang, Q.; Chen, Y.; Zhang, Z.; Peng, Y.; Li, X.; Xie, C.; Nong, S.; et al. {MegaScale}: Scaling large language model training to more than 10,000 {GPUs}. In Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24), Santa Clara, CA, USA, 16–18 April 2024; pp. 745–760.

190. Narayanan, D.; Shoeybi, M.; Casper, J.; LeGresley, P.; Patwary, M.; Korthikanti, V.; Vainbrand, D.; Kashinkunti, P.; Bernauer, J.; Catanzaro, B.; et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, St. Louis, MO, USA, 14–21 November 2021; pp. 1–15.
191. Dai, F.; Chen, Y.; Huang, Z.; Zhang, H.; Zhang, F. Efficient all-reduce for distributed DNN training in optical interconnect systems. In Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, Montreal, QC, Canada, 25 February 2023–1 March 2023; pp. 422–424.
192. Hoefler, T.; Belli, R. Scientific benchmarking of parallel computing systems: Twelve ways to tell the masses when reporting performance results. In Proceedings of the International Conference for High Performance Computing, networking, Storage and Analysis, Austin, TX, USA, 15–20 November 2015; pp. 1–12.
193. Xie, G.; Xiao, X.; Peng, H.; Li, R.; Li, K. A survey of low-energy parallel scheduling algorithms. *IEEE Trans. Sustain. Comput.* **2021**, *7*, 27–46. [[CrossRef](#)]
194. Li, C.; Chen, L. Optimization for energy-aware design of task scheduling in heterogeneous distributed systems: A meta-heuristic based approach. *Computing* **2024**, *106*, 2007–2031. [[CrossRef](#)]
195. Sakuma, D.; Taherkhani, A.; Tsuno, T.; Sasaki, T.; Shimizu, H.; Teramoto, K.; Todd, A.; Ueno, Y.; Hajdušek, M.; Ikuta, R.; et al. An Optical Interconnect for Modular Quantum Computers. *arXiv* **2024**, arXiv:2412.09299.
196. Brunner, D.; Shastri, B.J.; Qadasi, M.A.A.; Ballani, H.; Barbay, S.; Biasi, S.; Bienstman, P.; Bilodeau, S.; Bogaerts, W.; Böhm, F.; et al. Roadmap on Neuromorphic Photonics. *arXiv* **2025**, arXiv:2501.07917.
197. Thakur, R.; Rabenseifner, R.; Gropp, W. Optimization of collective communication operations in MPICH. *Int. J. High Perform. Comput. Appl.* **2005**, *19*, 49–66. [[CrossRef](#)]
198. Wang, X.; Martínez, J.F. XChange: A market-based approach to scalable dynamic multi-resource allocation in multicore architectures. In Proceedings of the 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), Burlingame, CA, USA, 7–11 February 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 113–125.
199. Reuther, A.; Byun, C.; Arcand, W.; Bestor, D.; Bergeron, B.; Hubbell, M.; Jones, M.; Michaleas, P.; Prout, A.; Rosa, A.; et al. Scalable system scheduling for HPC and big data. *J. Parallel Distrib. Comput.* **2018**, *111*, 76–92. [[CrossRef](#)]
200. Shamim, M.S.; Muralidharan, J.; Ganguly, A. An interconnection architecture for seamless inter and intra-chip communication using wireless links. In Proceedings of the 9th International Symposium on Networks-on-Chip, Vancouver, BC, Canada, 28–30 September 2015; pp. 1–8.
201. Belapurkar, A.; Chakrabarti, A.; Ponnappalli, H.; Varadarajan, N.; Padmanabhuni, S.; Sundarajan, S. *Distributed Systems Security: Issues, Processes and Solutions*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
202. Sophos. The State of Ransomware 2021. In *Technical Report*; Sophos Group: Abingdon, UK, 2021. Available online: <https://news.sophos.com/en-us/2021/04/27/the-state-of-ransomware-2021/> (accessed on 7 January 2025).
203. Baviskar, S.; Sanoj, R.; Nath, H.V. A Survey on Harnessing High-Performance Computing and Machine Learning for Detecting Stealthy Cache-Based Side-Channel Attacks. In Proceedings of the 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), Pune, India, 17–19 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
204. Kakoulli, E.; Zacharioudakis, E. Survey on Cryptoprocessors Advances and Technological Trends. In Proceedings of the The International Conference on Innovations in Computing Research, Madrid, Spain, 4–6 September 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 411–430.
205. Swanzy, P.N.; Abukari, A.M.; Ansong, E.D. Data Security Framework for Protecting Data in Transit and Data at Rest in the Cloud. *Curr. J. Appl. Sci. Technol.* **2024**, *43*, 61–77. [[CrossRef](#)]
206. Egwutuoha, I.P.; Levy, D.; Selic, B.; Chen, S. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *J. Supercomput.* **2013**, *65*, 1302–1326. [[CrossRef](#)]
207. Lashkari, B.; Musilek, P. A comprehensive review of blockchain consensus mechanisms. *IEEE Access* **2021**, *9*, 43620–43652. [[CrossRef](#)]
208. Chahar, S. Exploring the future trends of cryptography. In *Next Generation Mechanisms for Data Encryption*; CRC Press: Boca Raton, FL, USA, 2025; pp. 234–257.
209. Gärtner, F.C. Fundamentals of fault-tolerant distributed computing in asynchronous environments. *Acm Comput. Surv. CSUR* **1999**, *31*, 1–26. [[CrossRef](#)]
210. Bazgir, E.; Haque, E.; Uddin, M.S. Analysis of Distributed Systems. *Int. J. Comput. Appl.* **2024**, *975*, 8887. [[CrossRef](#)]
211. Jayasekara, S.; Karunasekera, S.; Harwood, A. Optimizing checkpoint-based fault-tolerance in distributed stream processing systems: Theory to practice. *Softw. Pract. Exp.* **2022**, *52*, 296–315. [[CrossRef](#)]
212. Kirti, M.; Maurya, A.K.; Yadav, R.S. Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8081. [[CrossRef](#)]
213. Liu, M.; Xia, L.; Wang, Y.; Chakrabarty, K. Fault tolerance in neuromorphic computing systems. In Proceedings of the 24th Asia and South Pacific Design Automation Conference, Tokyo, Japan, 21–24 January 2019; pp. 216–223.

214. Campanella, A.; Yan, B.; Casellas, R.; Giorgetti, A.; Lopez, V.; Zhao, Y.; Mayoral, A. Reliable optical networks with ODTN: Resiliency and fail-over in data and control planes. *J. Light. Technol.* **2020**, *38*, 2755–2764. [[CrossRef](#)]
215. March, S.; Hevner, A.; Ram, S. Research commentary: An agenda for information technology research in heterogeneous and distributed environments. *Inf. Syst. Res.* **2000**, *11*, 327–341. [[CrossRef](#)]
216. Razon, A.; Thomas, T.; Banunarayanan, V. Advanced distribution management systems: Connectivity through standardized interoperability protocols. *IEEE Power Energy Mag.* **2019**, *18*, 26–33. [[CrossRef](#)]
217. Pulicharla, M.R. Hybrid Quantum-Classical Machine Learning Models: Powering the Future of AI. *J. Journal Sci. Technol.* **2023**, *4*, 40–65. [[CrossRef](#)]
218. Guo, W.; Fouda, M.E.; Eltawil, A.M.; Salama, K.N. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Front. Neurosci.* **2021**, *15*, 638474. [[CrossRef](#)] [[PubMed](#)]
219. Wang, Z.; Liu, K.; Li, J.; Zhu, Y.; Zhang, Y. Various frameworks and libraries of machine learning and deep learning: A survey. *Arch. Comput. Methods Eng.* **2019**, *31*, 1–24. [[CrossRef](#)]
220. Sheikh, H.F.; Tan, H.; Ahmad, I.; Ranka, S.; Bv, P. Energy-and performance-aware scheduling of tasks on parallel and distributed systems. *Acm J. Emerg. Technol. Comput. Syst. JETC* **2012**, *8*, 1–37. [[CrossRef](#)]
221. Alzoubi, Y.I.; Mishra, A. Green artificial intelligence initiatives: Potentials and challenges. *J. Clean. Prod.* **2024**, *468*, 143090. [[CrossRef](#)]
222. Krishnamoorthy, A.; Schwetman, H.; Zheng, X.; Ho, R. Energy-efficient photonics in future high-connectivity computing systems. *J. Light. Technol.* **2015**, *33*, 889–900. [[CrossRef](#)]
223. Gu, H.; Chen, K.; Yang, Y.; Chen, Z.; Zhang, B. MRONoC: A low latency and energy efficient on chip optical interconnect architecture. *IEEE Photonics J.* **2017**, *9*, 1–12. [[CrossRef](#)]
224. Dhasade, A.; Dini, P.; Guerra, E.; Kermarrec, A.M.; Miozzo, M.; Pires, R.; Sharma, R.; de Vos, M. Energy-Aware Decentralized Learning with Intermittent Model Training. *arXiv* **2024**, arXiv:2407.01283.
225. Bambagini, M.; Marinoni, M.; Aydin, H.; Buttazzo, G. Energy-aware scheduling for real-time systems: A survey. *Acm Trans. Embed. Comput. Syst. TECS* **2016**, *15*, 1–34. [[CrossRef](#)]
226. Yellu, R.R.; Maruthi, S.; Dodda, S.B.; Thuniki, P.; Reddy, S.R.B. AI Ethics—Challenges and Considerations: Examining ethical challenges and considerations in the development and deployment of artificial intelligence systems. *Afr. J. Artif. Intell. Sustain. Dev.* **2021**, *1*, 9–16.
227. Mensah, G.B. Artificial intelligence and ethics: A comprehensive review of bias mitigation, transparency, and accountability in AI Systems. *Africa Journal For Regulatory Affairs (AJFRA)* **2024**, *1*, 32–45.
228. Celi, L.A.; Cellini, J.; Charpignon, M.L.; Dee, E.C.; Dernoncourt, F.; Eber, R.; Mitchell, W.G.; Moukheiber, L.; Schirmer, J.; Situ, J.; et al. Sources of bias in artificial intelligence that perpetuate healthcare disparities—A global review. *PLoS Digit. Health* **2022**, *1*, e0000022. [[CrossRef](#)]
229. Salako, A.O.; Fabuyi, J.A.; Aideyan, N.T.; Selesi-Aina, O.; Dapo-Oyewole, D.L.; Olaniyi, O. Advancing Information Governance in AI-Driven Cloud Ecosystem: Strategies for Enhancing Data Security and Meeting Regulatory Compliance. *Asian J. Res. Comput. Sci.* **2024**, *17*, 66–88. [[CrossRef](#)]
230. Patidar, N.; Mishra, S.; Jain, R.; Prajapati, D.; Solanki, A.; Suthar, R.; Patel, K.; Patel, H. Transparency in AI Decision Making: A Survey of Explainable AI Methods and Applications. *Adv. Robot. Technol.* **2024**, *2*, 1 .
231. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, *5*, 1–19. [[CrossRef](#)] [[PubMed](#)]
232. Turker, I.; Bicer, A.A. How to use blockchain effectively in auditing and assurance services. In *Digital Business Strategies in Blockchain Ecosystems: Transformational Design and Future of Global Business*; Springer International Publishing: Cham, Switzerland, 2020; pp. 457–471.
233. Mzukwa, A. Exploring Next-Generation Architectures for Advanced Computing Systems: Challenges and Opportunities. *J. Adv. Comput. Syst.* **2024**, *4*, 9–18.
234. Singh, H.; Tyagi, S.; Kumar, P.; Gill, S.S.; Buyya, R. Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions. *Simul. Model. Pract. Theory* **2021**, *111*, 102353. [[CrossRef](#)]
235. Kocot, B.; Czarnul, P.; Proficz, J. Energy-aware scheduling for high-performance computing systems: A survey. *Energies* **2023**, *16*, 890. [[CrossRef](#)]
236. Shabestari, F.; Navimipour, N.J. An Energy-Aware Resource Management Strategy Based On Spark and YARN in Heterogeneous Environments. *IEEE Trans. Green Commun. Netw.* **2023**, *8*, 635–644. [[CrossRef](#)]
237. Lu, P.J.; Lai, M.C.; Chang, J.S. A survey of high-performance interconnection networks in high-performance computer systems. *Electronics* **2022**, *11*, 1369. [[CrossRef](#)]
238. Mahajan, R.; Li, X.; Fryman, J.; Zhang, Z.; Nekkanty, S.; Tadayon, P.; Jaussi, J.; Shumarayev, S.; Agrawal, A.; Jadhav, S.; et al. Co-packaged photonics for high performance computing: Status, challenges and opportunities. *J. Light. Technol.* **2021**, *40*, 379–392. [[CrossRef](#)]

239. Wang, Z.; Zhao, J. Fork is All You Needed in Heterogeneous Systems. *arXiv* **2024**, arXiv:2404.05085.
240. Cai, Z.; Babbush, R.; Benjamin, S.C.; Endo, S.; Huggins, W.J.; Li, Y.; McClean, J.R.; O'Brien, T.E. Quantum error mitigation. *Rev. Mod. Phys.* **2023**, *95*, 045005. [[CrossRef](#)]
241. Abreu, S.; Pedersen, J.E. Neuromorphic Programming: Emerging Directions for Brain-Inspired Hardware. In Proceedings of the 2024 International Conference on Neuromorphic Systems (ICONS), Arlington, VA, USA, 30 July–2 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 358–365.
242. Imam, N.; Cleland, T.A. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nat. Mach. Intell.* **2020**, *2*, 181–191. [[CrossRef](#)] [[PubMed](#)]
243. Rabadán, A.T.; Ammar, A. Neurochips: An Ethical Consideration. In *Learning and Career Development in Neurosurgery: Values-Based Medical Education*; Springer: Cham, Switzerland, 2022; pp. 101–109.
244. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Comprehensive Survey on Green Blockchain: Developing the Next Generation of Energy Efficient and Sustainable Blockchain Systems. *arXiv* **2024**, arXiv:2410.20581.
245. Asif, R.; Hassan, S.R. Shaping the future of Ethereum: Exploring energy consumption in Proof-of-Work and Proof-of-Stake consensus. *Front. Blockchain* **2023**, *6*, 1151724. [[CrossRef](#)]
246. Ismail, A.; Toohey, M.; Lee, Y.C.; Dong, Z.; Zomaya, A.Y. Cost and performance analysis on decentralized file systems for blockchain-based applications: State-of-the-art report. In Proceedings of the 2022 IEEE International Conference on Blockchain (Blockchain), Espoo, Finland, 22–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 230–237.
247. Dulam, N.; Shaik, B.; Allam, K. Serverless AI: Building Scalable AI Applications Without Infrastructure Overhead. *J.-Assist. Sci. Discov.* **2022**, *2*, 519–542.
248. Soltani, B.; Ghenai, A.; Zeghib, N. Towards distributed containerized serverless architecture in multi cloud environment. *Procedia Comput. Sci.* **2018**, *134*, 121–128. [[CrossRef](#)]
249. Weerasinghe, S.; Perera, I. Optimized Strategy in Cloud-Native Environment for Inter-Service Communication in Microservices. *Int. J. Online Biomed. Eng.* **2024**, *20*, 1. [[CrossRef](#)]
250. Lu, Y.; Bian, S.; Chen, L.; He, Y.; Hui, Y.; Lentz, M.; Li, B.; Liu, F.; Li, J.; Liu, Q.; et al. Computing in the Era of Large Generative Models: From Cloud-Native to AI-Native. *arXiv* **2024**, arXiv:2401.12230.
251. Gulli, A. *Google Anthos in Action: Manage Hybrid and Multi-Cloud Kubernetes Clusters*; Simon and Schuster: New York, NY, USA, 2023.
252. Kaul, D. AI-Driven Self-Healing Container Orchestration Framework for Energy-Efficient Kubernetes Clusters. *Emerg. Sci. Res.* **2024**, *2024*, 1–13.
253. Theodoropoulos, T.; Rosa, L.; Benzaid, C.; Gray, P.; Marin, E.; Makris, A.; Cordeiro, L.; Diego, F.; Sorokin, P.; Girolamo, M.D.; et al. Security in Cloud-Native Services: A Survey. *J. Cybersecur. Priv.* **2023**, *3*, 758–793. [[CrossRef](#)]
254. Peters, D.; Vold, K.; Robinson, D.; Calvo, R.A. Responsible AI—Two frameworks for ethical design practice. *IEEE Trans. Technol. Soc.* **2020**, *1*, 34–47. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.