JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2024, VOL 7, NO S12

Integrating Distributed Systems Into Platform Engineering: Challenges And Solutions In High-Traffic Applications

Purushottam Raj

Sr Software Engineering Leader India.

Received: 25-Sep-2024
Accepted: 20-Nov-2024

ABSTRACT
Over the last several decades, computing paradigms have changed dramatically, going from massive room-sized resources (processors and memory) to minuscule computer nodes. Most modern application sectors have been lured to computers' potential. Cloud, fog/edge, IoT, and mobile device computing are currently being combined into a seamless continuum via distributed computing continuum systems (DCCSs). Its seamless design efficiently manages a range of processing needs while ensuring a consistent user experience. Additionally, it offers a comprehensive answer to the demands of contemporary computing. In light of this, this study offers a more thorough comprehension of DCCSs' possibilities in the current computing environment. We start by going over how computing paradigms have changed up to DCCS. The advantages and disadvantages of every computer paradigm are examined, along with generic designs, components, and different computing devices. Following that, we continue our conversation on the different computing devices that make up DCCS in order to accomplish computational objectives in both present-day and future applications. Finally, we discuss the unresolved challenges and potential future advancements required for DCCS to achieve its full potential across a wide range of applications.

Keywords: Internet of Things, edge computing, edge intelligence, artificial intelligence, machine learning, distributed computing continuum systems.

1. INTRODUCTION

Computing started to quickly answer complicated calculations a few decades ago, but it was prone to mistakes. Today, however, it epitomizes the transformational force of technology (DeDonno,et. al . 2019). Computers used to be big, roomy devices with little processing power, but now days they are portable, have massive memory capacity, and can process data quickly. Modern computing, which includes huge calculations in a distributed and parallel fashion, has benefited from advancements in hardware, software (via the growth of programming languages), and the Internet (Yuan, J.; Xiao, et. al 2023; Alsamhi, S.H. et. al. 2022). In this sense, the first distributed computer architecture built on the Internet in the early 1960s was mainframe-based computing. Moreover, cloud computing, fog/edge computing, grid computing, and cluster computing were all included in the computing paradigm (Singh, S. K., et. al 2022). Nearly every industry, including education, healthcare, entertainment, and space exploration, has changed as a result of computing accessibility and pervasiveness (Rekha, P. et. al 2017).

An age when the lines separating various computer environments blur and a smooth continuum appears is being ushered in by Distributed computer Continuum Systems (DCCSs), a breakthrough development in the computing industry [Suresh, K. et. al 2019). The seamless integration of mobile devices, cloud computing, edge computing, and other computing platforms into a unified ecosystem creates both opportunities and challenges (Bharathy, S. S. P. D., et. al 2017; Sujithra, M., et. al 2022). This way an ecosystem can be formed, which keeps on dynamically distributing resources for different tasks on different layers. There are many instances in which DCCSs are really effective against the traditional computing paradigm. Real- time processing can usually be done at the edge while adding data at fog and can also include huge computing on the cloud (Laxminarayana Korada et. al 2024). It allows for the applications depending on the requirements while granting them seamless access and required allocation of resources and use of smooth transitioning between them. This flexibility improves performance overall and allows resources to be used more effectively. Nowadays, most applications need a lot of processing power, such artificial intelligence (AI) and machine learning (ML) (Korada, L. 2024), which DCCSs can effectively complete with the resources at hand.

The characteristics of DCCSs, a rapidly developing technology, encourage its use in a variety of applications. We start by going over how several distributed computing concepts have evolved since 1960. Every computer paradigm is covered, along with its components, advantages, and drawbacks. The architecture of DCCSs and how it varies from conventional computing paradigms are then examined. We examine DCCSs' benefits and drawbacks from a number of angles. We list a number of applications that are in dire need of DCCSs and provide an example of how to use them instead of more conventional computing models.

First, we examine how the paradigm for computing changed from the 1960s to the present. Additionally, the overall advantages and disadvantages of DCCSs are examined. Additionally, we provide a range of real-time example situations and applications where computing paradigms are crucial. We emphasize the advantages of DCCSs for various use scenarios. Lastly, we go over a number of unresolved research issues and potential fixes for upcoming DCCS improvements that might increase their effectiveness.

This paper's remaining parts are arranged as follows. Following Section 6 is a list of the acronyms used in this work. Section 2 discusses the distributed computing continuum systems' year-by-year development. Section 3 talks about the possibilities and difficulties of DCCSs. A number of real-time applications that profit from DCCSs are covered in Section 4. Section 5 covers a broad variety of unresolved issues and potential areas for further investigation. In Section 6, we finally wrap up our paper.

2. THE DISTRIBUTED COMPUTING CONTINUUM'S DEVELOPMENT

This section explains the transition from mainframe-based computing to DCCSs, as illustrated in Figure 1. The advent of networking and the Internet in the late 1960s enabled computer systems to collaborate and operate concurrently, paving the way for the development of DCCSs.

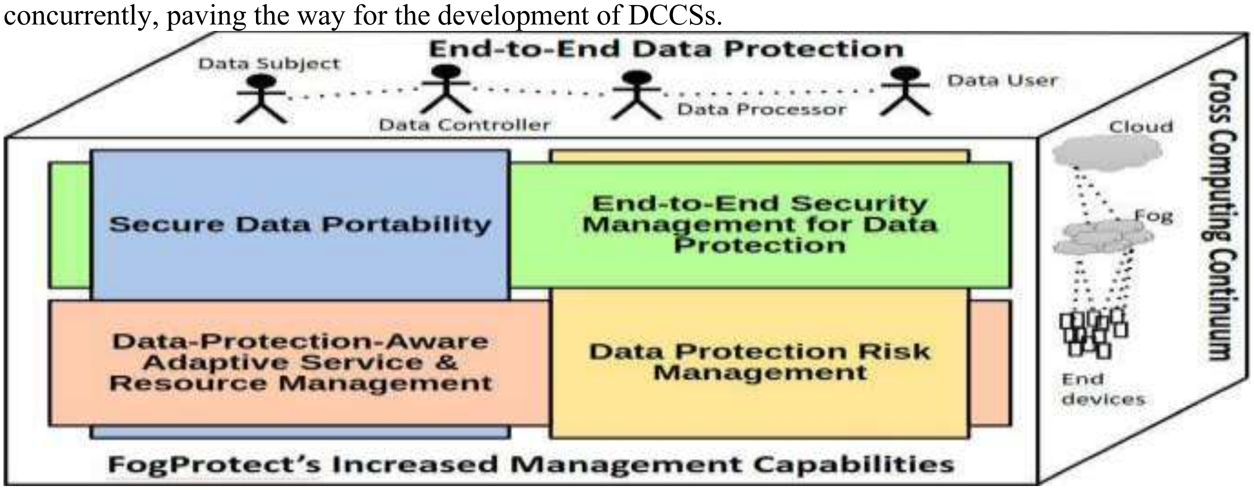


Figure 1: Development of continuum systems for computer processing

2.1. Computing Based on Mainframes

High-performance computation such as mainframe technology, employ large amounts of storage (private databases) and processing capacity (a transactional approach servers) to conduct computations in real time. Additionally, companies use a variety of applications to carry out a range of functions, including processing financial data and offering high security, agility, and resilience (Iansiti, M.; et. al 1994). Figure 2 replicates a standard mainframe-based computing paradigm.

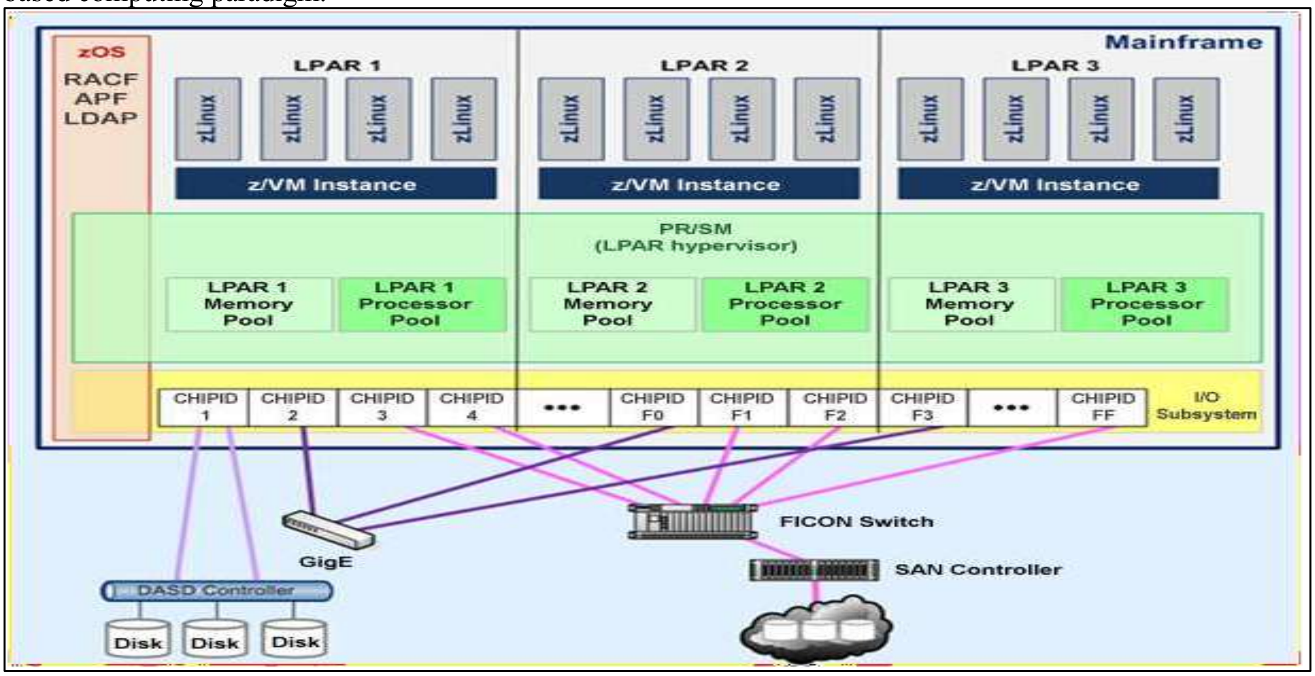


Figure 2: Mainframe-based computation architecture

Because mainframe computers are built with a large quantity of resources, they can do calculations quickly and centrally. However, there are a number of drawbacks to the mainframe computer paradigm. These could not be as adaptable as other computer models since they are designed for certain purposes. As a result, they cannot be customized or adjusted to meet changing company needs. Because of their intricacy, they need a specialized individual to run and maintain, and their hardware and software are costly.

Because mainframes are big and need specialized infrastructure and space, small businesses cannot use them. Switching from the present platform to another is very difficult because of the vendor lock-in mechanism (Greenstein, S.M . 1997).

2.2. Utilizing Grid Technology

In order to do typical computationally demanding activities that are impossible to complete on a single system, grid computing entails joining many computers over a network. To carry out these activities, a network of computers functions as a single virtual supercomputer. By building a single-system picture, users and apps may easily access IT capabilities in this manner (Schwiegelshohn, et. al 2010; Casanova, H. 2022). Users, grid nodes, and a central server are the main elements of the generic grid computing working paradigm, which is shown in Figure 3. The network pool's computational assignments and resources are managed and controlled by a central node. A grid node adds resources (such memory, CPUs, and storage) to a network pool. In this network, the nodes actively take part in the distributed grid's calculations (Yu, J et. al 2005).

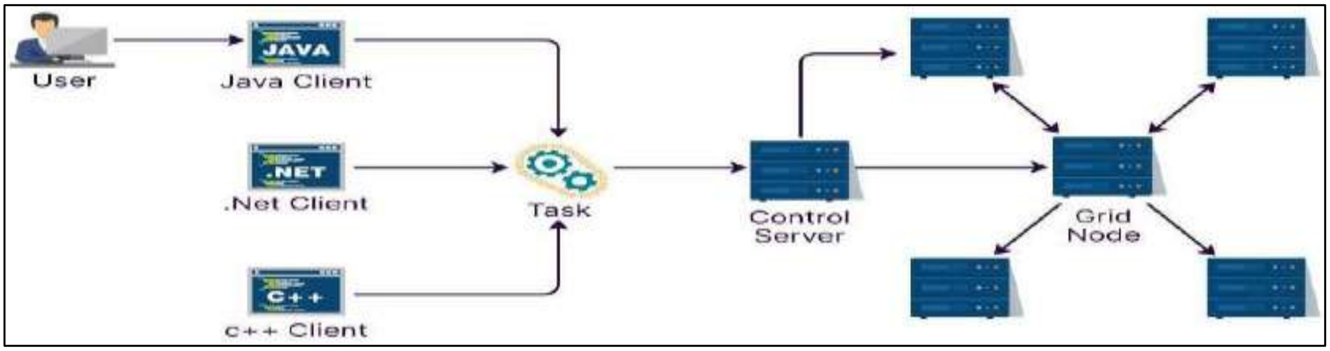


Figure 3: Grid computing general working paradigm

For huge calculations in a short period of time, the grid computing approach is quite effective. With this strategy, many organizations may work together to share computational resources. However, grid computing comes with a number of drawbacks. Additionally, this paradigm is not suitable for simple jobs. This architecture also requires a large bandwidth to facilitate data interchange between grid nodes. Interoperability and flexibility are not supported by this approach.

2.3 Systems for Distributed Computing Continuum

DCCSs consist of diverse, interconnected computing systems designed to manage data generated by devices such as smart phones, sensors, and IoT systems. DCCSs distribute computational workloads across various system components to perform tasks effectively. A piece of the calculation is completed by each device, and the final output is created by combining the results. Increased scalability and quicker processing speeds are made possible by this. Computations are carried out effectively using DCCSs, which also optimize resource use beyond conventional bounds and adjust to shifting needs (Yeo, C.S.et. al 2006; Baker, M. et. al 1999). Time-sensitive job prioritization, computing capacity, and resource closeness are some of the elements that determine how resources are allocated. This dynamic task allocation lowers latency while improving processing efficiency and system performance. Figure 4 provides an illustration of DCCS's overall architecture.

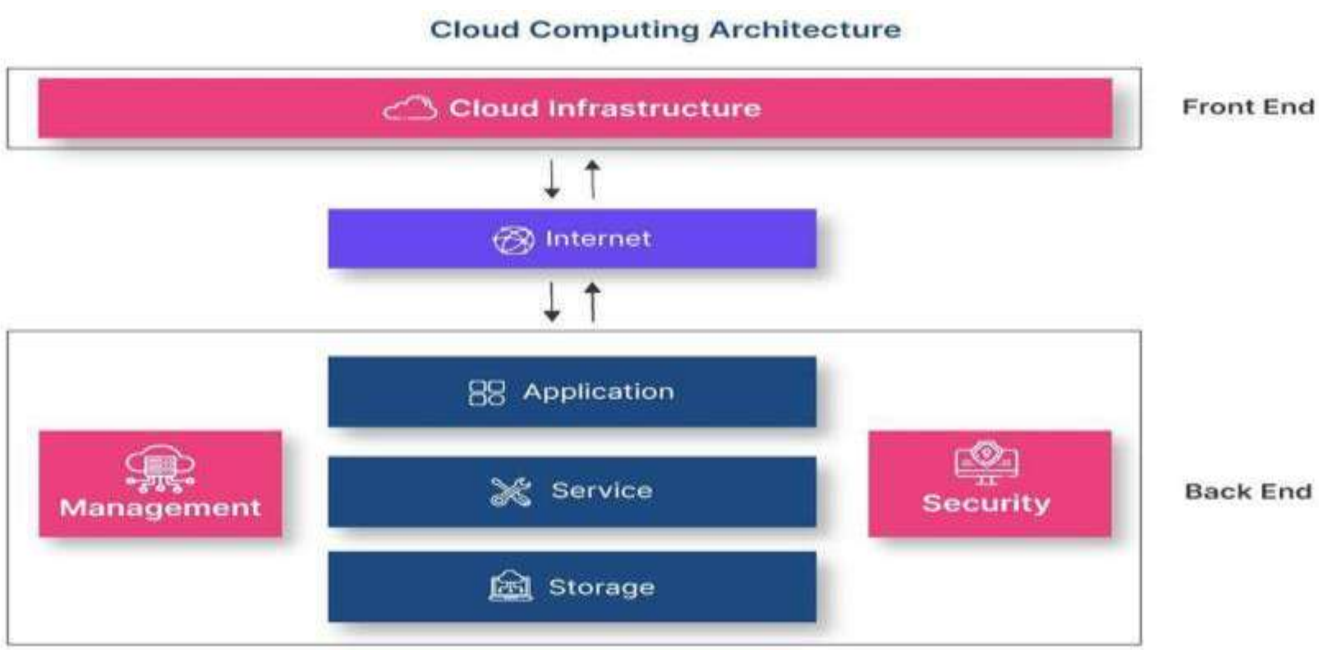


Figure 4: Conceptual architecture for continuous systems in cloud computing

3. THE DISTRIBUTED COMPUTING CONTINUUM'S POTENTIAL

Because DCCSs smoothly incorporate a variety of computing resources, they are fundamentally different from conventional computing paradigms. This technique includes integrating robust cloud servers with mobile endpoints that are everywhere and nimble edge devices. A comprehensive ecosystem is produced to satisfy the many demands of contemporary computing by using the advantages of each component. The kinds of computer devices used in DCCSs are first described in this context. DCCSs provide a way forward for computers to smoothly adjust to the demands of applications. This section examines the main advantages and difficulties of DCCSs in order to provide new levels of responsiveness and efficiency.

3.1. Computer Device Classes Used in DCCSs

In DCCSs, several kinds of computer devices collaborate to provide smooth data creation, processing, analysis, and system-wide communication. These devices are divided into five categories in our paper: servers, desktop computers, mobile devices, embedded computers, and the Internet of Things.

3.1.1. Computers that are embedded

Electronic or mechanical machines having computer parts (processors, such computer processors or microchips), power sources, actuators, sensing and ways of communicating, and memory, as well as software to carry out a particular job, are known as embedded systems for computers. Diagnostic and patient monitoring equipment, ATMs, engineering calculators, digital cameras, microwaves, automated washing machines, digital door locks, autonomous automobiles, and more now have computers. All of these gadgets are Internet-connected and controlled remotely due to the widespread use of smart and remote apps.

They are capable of making judgments on their own and acting appropriately with this technology and software. These gadgets were recently linked to the Internet, and data from them is sent and processed in the cloud to track and forecast the effectiveness and condition of the equipment. Networking gateways and routers are also embedded computer devices. Gateways use a chip that is configured to convert messages between different protocols. It serves as a mediator between the network and physical levels. Routers have the ability to send data packets across network interfaces and maintain routing tables. In accordance with their configurations, these two devices may also execute additional preprogrammed actions.

3.2. Benefits

The architectural agility offered by DCCSs includes latency reduction, scalability, bandwidth optimization, load balancing, resilience, adaptability, as well as dependability. On a shorter frill, DCCSs are capable of several advantages depending on the type and application for which they are meant. To enhance comprehension of the beneficial aspect of each, we have discussed examples.

Increase bandwidth efficiency: the sharing of computational tasks is carefully defined between centralized cloud resources and edge devices in DCCS. It avoids high-cost data transfers by sending very minimal data or intelligence to the cloud when local device resources exhaust. Further, it minimizes large data transfer through using localized caching and processing. When it comes to speedier decision-making and bandwidth preservation, that dynamic characteristic of DCCSs enhances their importance in counseled conditions marked by sparse or limited connectivity.

Scalability: DECLAR STT, when dynamic computation tasks to resources, is assigned to and thus exhibited to scalability. To visualize the scalability attribute of DCCSs, let us imagine a very simple scenario. For example, during normal traffic conditions, it would take real-time traffic data processing with local servers and edge devices; now, suddenly, when the traffic spike occurs or just the rise in traffic, for example, during the morning or evening, the system might require using DCCS. Then, to manage the increasing load without sacrificing speed, more resources (such the cloud) may be incorporated. Because DCCSs efficiently use available resources without overloading any one component, scalability is particularly beneficial when workloads vary or demand increases

unexpectedly. The increasing computing demands of contemporary applications and services may be readily accommodated by DCCS because of its architectural flexibility.
Low latency: Unlike cloud systems, which transport data over great distances, DCCS executes jobs locally to the data source, resulting in low latency. For Production and cloud-based methods send data to a distant cloud server, which slows down reaction times and introduces network latency. Smart city traffic control solutions need low latency for real-time responses.

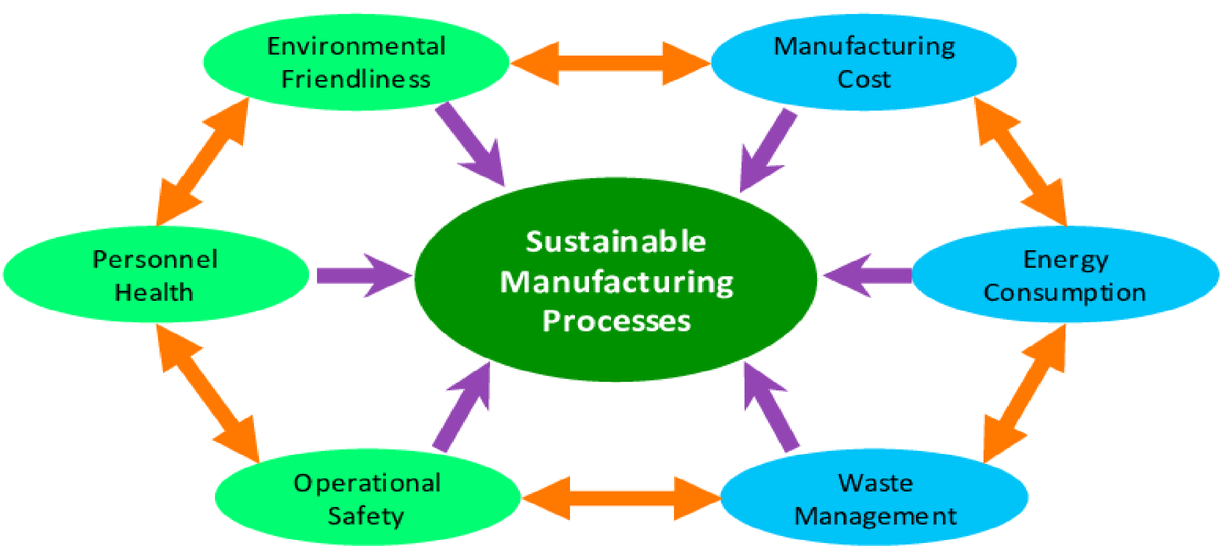
4. USES

Because of its ability to smoothly integrate a large range of computing resources, DCCSs can execute applications in a number of areas. In order to illustrate the differences between contemporary technologies with a computing continuum, we address a few applications (industrial automation, transport systems, smart cities and healthcare using a use case example. Depending on the needs of the application, these applications may gain from improved resource use, quicker decision-making, and a number of other advantages by using DCCS functionalities.

4.1. Automation in Industry

Technology is used in a wide range of industries and enterprises to enhance processes and operations. Controlling processes (e.g., oil refining or pharmaceuticals), tracking the environment, food and beverage processing along with assurance checks, electronic grids, or manufacturing automation are examples of industrial applications. The majority of these functions have been processed by machines, increasing efficiency, safety, and production. Failures of machines lower maintenance, safety, and production costs. Equipment failure may be reduced and productivity increased by routine, real-time, and predictive maintenance. IoT is used in industrial applications to gather data from equipment and transmit it to the edge or cloud for analysis. DCCSs were created for industrial applications and provide real-time monitoring with a minimal central server interface (fig. 5).

Figure 5: Illustration of how to distinguish between good and bad components in an automated production system.



4.2. Systems of Transportation

Transportation systems are structured networks and infrastructure that make it easier for people or commodities to travel between different places using different means of transportation, such as roads, railroads, aero planes, or waterways. By fusing real-time data processing with the computing continuum, DCCSs revolutionize contemporary transportation systems. In order to improve traffic flow, lessen congestion, and guarantee safe navigation, it analyses data obtained from sensors installed in infrastructure and automobiles. They also assist with infrastructure upkeep, emergency response, and effective mobility services.

4.3. Medical Care

A range of medical services, tools, and systems intended to prevent, identify, treat, and manage illnesses and ailments are collectively referred to as healthcare. From wearable sensors to expensive equipment (located in hospitals and healthcare facilities) that gather patient data and analyze it via smart phones (edge devices) or the cloud, a number of medical gadgets have advanced in recent years. Computing equipment must provide precise and fast analytical findings for the healthcare sector. Intense work like genome sequencing or medical pictures (X-rays or CT scans) may sometimes need to be analyzed, but the results should be provided quickly. In some cases, predicting the patient's state requires the application of AI or ML, which requires greater processing power.

5. OPPORTUNITIES FOR FURTHER RESEARCH

Investigating DCCSs creates new opportunities and motivates further research into its many facets. This part seeks to shed light on DCCSs and provide potential avenues for more study into the issues covered in earlier sections.

5.1. DCCS Learning Models

The quick development of learning models has facilitated creative applications in many domains and resulted in breakthroughs in a variety of industries. Furthermore, DCCSs produce a lot of data from their devices, which for a number of reasons need further processing. In this sense, real-time data analysis and decision-making are more suited for learning models. The literature has a number of algorithms and new methods are developing quickly. Deep learning models, for instance, are being developed as generative adversarial networks (GANs), which are driving innovations across several domains. In order to improve an application, two or more learning algorithms are combined to create an ensemble learning model. Additionally, like human learning, a number of experimental learning models pick up information by interaction with their surroundings.

To increase efficiency, these algorithms need a vast quantity of organized data and resources. Models may use transfer and few-shot learning techniques to learn from short datasets or to transfer information across tasks. When computation is restricted to edge devices, this approach is more effective.

DCCS presents distinct challenges and characteristics in terms of the complexity of data generation, device heterogeneity, and communication. Thus, novel teaching approaches will help DCCSs fulfill their potential and reduce their challenges. Energy efficiency, resource and data limits, data volumes, instantaneous choice making with local and global maximum values, cost effectiveness, flexibility, and scalability must be considered while designing these algorithms. The accuracy of these recently created learning algorithms in real time, or run time, must also be estimated. Both learnt models using historical data and learnt models with restricted data availability may perform better when incremental learning features are included. Domain generalization will improve DCCS efficiency and effectiveness by allowing models to make accurate predictions or decisions across devices, locations, and scenarios. These models will help DCCSs manage changing environments and enhance user experience while maximizing their capabilities.

5.2. Causality Use

The link between a cause and an effect, when a cause leads to an effect, is sometimes referred to as causation. It is essential to comprehending how choices, events, and actions affect results in DCCSs. This enables us to comprehend a system's behavior, forecast its behavior, filter data, and guarantee consistency. This knowledge and a priori analysis make it simple to reduce the impacts of uncertainty. There are a number of advantages to using causality in DCCSs, including fault or failure predictions, preserving system integrity, determining the intuitive ordering of execution during parallel computation across multiple computing devices, and forecasting resource availability for future uses to prevent latency. Chen et al., for instance, used causality in to model fault propagation channels and determine the actual culprits behind performance problems in distributed systems. Graph knowledge representation

(GKR) via representation learning is one of the techniques for determining the best suitable causal connection with little analysis that have been published in the literature.

5.3 Regulating Data Friction and Gravity

Information gravity is the tendency of data to attract additional data. This is based on the idea that the more applications and data that are stored in one location, the more appealing it is to store further data there. Data friction is resistance that stops data from flowing. The dynamic allocation of DCCS is likely to result in data gravity and friction on certain network devices. Data gravity and friction in the computer continuum impair performance and resource utilization. These problems could be resolved by intelligent data placement, replication, caching, and data transmission, but further study is required.

Table 1: Key summarization of the concept

Key Points	Details	Implications	Applications
Transition from mainframe computing to DCCSs enabled by networking & the Internet.	The evolution from mainframe to distributed systems, driven by networking and internet advancements.	DCCS systems provide a more flexible, scalable approach to computing than traditional mainframes.	Facilitates modern computing paradigms with scalability, flexibility, and networked efficiency.
Mainframes handle high-performance computing but are costly, rigid, and require specialized maintenance.	Mainframes are large systems with high storage and processing capacity, but their rigidity and cost limit their use.	Small businesses are often unable to afford or maintain mainframe systems due to their complexity and cost.	Mainframes are ideal for large enterprises but are impractical for small businesses due to high operational costs.
Grid computing Connects multiple computers over a network to perform complex tasks.	Grid computing requires high bandwidth and lacks flexibility, often unable to support simple tasks efficiently.	Grid systems enable large-scale computations but may not be suitable for all use cases due to infrastructure demands.	Useful for scientific research, simulations, and tasks that require distributed processing.
DCCS integrate cloud, edge, and mobile devices to optimize resource usage and performance.	DCCS splits tasks among different system components, providing scalability and low latency while optimizing resources.	DCCS enhances performance, efficiency, and adaptability across diversecomputing resources.	Supports applications that require real-time data processing, such as IoT, AI, and cloud computing.
DCCSs offer scalability, low latency, and efficiency by integrating different computing resources.	DCCS connects cloud servers, mobile devices, and edge devices, improving flexibility and responsiveness.	Offers a solution to modern computing demands, allowing dynamic adjustments based on application needs.	Can be applied across industries like smart cities, manufacturing, and autonomous systems.
DCCS use servers, desktops, mobile devices, embedded systems, and IoT devices.	Devices collaborate to gather, process, and communicate data for various applications.	Devices in DCCS provide a seamless interaction for real-time data analysis and decision-making.	Key to industries requiring real-time decision-making, such as healthcare, transportation, and industrial automation.

Embedded devices, such as medical equipment and IoT sensors, play a critical role in DCCS.	Embedded systems are used for specialized tasks like monitoring, decision-making, and Communication in IoT environments.	Widely used in sectors like healthcare, manufacturing, and smart homes for real-time monitoring and decision-making.	Examples include smart medical devices, automotive systems, and IoT-enabled manufacturing.
Advantages include latency reduction, scalability, bandwidth optimization, and load balancing.	DCCS offers agility, resilience, and efficiency by distributing tasks intelligently across multiple devices.	Maximizes resource utilization, minimizes delays, and adapts to changing conditions, benefiting real-time applications.	Used in applications requiring high responsiveness, such as smart cities, emergency systems, and large-scale computing.
DCCS can be applied in industrial automation, transportation systems, and healthcare.	DCCS improves efficiency by enhancing decision-making, resource allocation, and task distribution.	Enables industries to optimize operations and services, from manufacturing to smart cities and healthcare.	Major sectors benefiting from DCCS include manufacturing, logistics, healthcare, and urban infrastructure.
DCCS enables real-time monitoring and predictive maintenance in industries.	IoT-enabled systems monitor machinery, improve safety, and predict failures to prevent downtime.	Helps industries optimize productivity, reduce downtime, and maintain safety standards through predictive analysis.	Applied in industries like oil & gas, pharmaceuticals, and food processing for real-time equipment management.
DCCS processes real-time data to enhance traffic flow and mobility services.	DCCS enables smarter transportation systems by analyzing sensor data in real-time to optimize traffic and reduce congestion.	Improves traffic management and infrastructure efficiency, contributing to smarter urban mobility solutions.	Applicable in smart city traffic systems, autonomous vehicles, and public transportation management.
DCCS aids in real-time analysis of medical data, improving Healthcare decision-making.	DCCS supports fast data processing from medical equipment, aiding in timely decision-making and diagnosis.	Enhances healthcare systems by providing faster, more accurate analyses for better patient care and outcomes.	Critical in genomics, diagnostics, and patient monitoring, especially for time-sensitive conditions.
DCCS opens new avenues for research in learning models, causality, and data management.	Areas for further Exploration include algorithms, real-time learning, and managing data friction/gravity.	Provides opportunities for further enhancing DCCS capabilities and overcoming challenges related to data processing.	Research in AI, deep learning, and real-time analytics can benefit from DCCS advances in data management and learning.

The table 1 gives a comprehensive overview of the Distributed Computing Continuum (DCCS), outlining its key components, benefits, and applications across different domains. DCCS is a modern computing framework that integrates diverse computational resources, including cloud servers, edge devices, mobile devices, and embedded systems, to optimize performance, scalability, and efficiency. Evolution from the traditional mainframe computing towards DCCS- based flexible and scalable systems reflects the growth of needs for distributed processing power that has been driven by developments in networking and the Internet.

The traditional mainframe computing, large centralized systems handling high- performance tasks, though very expensive, rigid, and require specialist maintenance, making it infeasible to apply them in smaller business

concerns. Distributed computing systems fill in this gap by making flexible and cost-effective solutions. Grid computing is a scalable option that connects multiple computers over a network to perform complex tasks, but it has the drawbacks of requiring high bandwidth and inflexibility in allocating tasks. All these are taken care of by DCCS, which brings together the strength of cloud, edge, and mobile computing for optimal results through intelligent task distribution and resource utilization. DCCS systems are based on a variety of computing devices, such as servers, desktops, mobile devices, embedded systems, and IoT devices. These components work together in harmony to process data and complete tasks in real-time. In the healthcare and manufacturing sectors, for example, and IoT applications, there is a great need for embedded systems to monitor, make decisions, and communicate. The integration of these devices in DCCS enables an efficient management of resources along with a reduction of latency and ensures that the tasks get completed swiftly and with accuracy.
One of the significant advantages that DCCS can offer includes low latency, scalability, bandwidth optimization, and load balancing. By appropriately distributing tasks across a variety of devices, DCCS ensures that every device runs at its optimal capacity, and hence, improved efficiency along with performance can be witnessed. The versatility of DCCS, which can run from very small real-time processing to heavyweight resource-intensive applications, also makes it amenable for application in virtually any kind of industry such as health, transportation, or industrial automation sectors, all because of the enhancement of the decision-making processes, resource utilization, and the workloads.
For example, in the healthcare sector, DCCS can support real-time analysis of medical data to make decisions on time and improve the outcomes of patients. In the transportation sector, DCCS can help improve traffic management and optimize mobility services through real- time processing of sensor data. In the industrial environment, DCCS can support predictive maintenance through the continuous monitoring of machinery and the provision of alerts before failures happen, reducing downtime and enhancing operational efficiency. The table also identifies other opportunities for further research in DCCS, including better algorithms for real- time learning and data friction management. These research avenues address the current limitations and push the boundaries of what DCCS can achieve, especially in handling large- scale data and complex computations. As DCCS continues to grow, the integration of AI, machine learning, and advanced data management techniques will only add to its potential across many fields, leading to even more intelligent, responsive, and adaptive computing environments. So, the DCCS framework represents a significant advancement in the way computing resources are managed and utilized. DCCS offers immense potential for transforming industries, improving systems, and creating new possibilities for research and innovation by enabling a dynamic, scalable, and efficient approach to distributed computing.

6. CONCLUSION

This trajectory of progress keeps speeding up, offering previously unheard-of capabilities that will transform science, industry, and society as a whole. In an attempt to demonstrate the shift, this document gives a summary of how computer paradigms have changed from the 1960s to the present. We covered everything from modern distributed computing continuum systems (DCCSs) to massive mainframe computers. We examined each computer model's structures, constituent parts, advantages, and drawbacks. We spoke about the main benefits of DCCSs that are appropriate for the computing needs and application situations of today. We looked at suitable use cases in relation to conventional technologies and the benefits of DCCSs. For each application, we provide an example use case to help explain how to utilize DCCS and its benefits. There are many open research difficulties and potential for more study since DCCSs are a developing topic of study. We spoke about potential unresolved research issues and offered appropriate study avenues.

REFERENCES

1. Alsamhi, S.H.; Shvetsov, A.V et. al (2022). Computing in the sky: A survey on intelligent ubiquitous computing for uav-assisted 6g networks and industry 4.0/5.0. Drones 2022, 6,
2. 177. DOI: https://doi.org/10.3390/drones6070177
3. Baker, M.; Buyya, R.(1999). Cluster computing: The commodity supercomputer. Soft. Pract. Exp. 1999, 29, 551–576. DOI: https://doi.org/10.1002/(SICI)1097- 024X(199905)29:6%3C551::AID-SPE248%3E3.0.CO;2-C

4. Bharathy, S. S. P. D., Preethi, P., et. al (2017). Hand Gesture Recognition for Physical Impairment Peoples. SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), 6-10. Retrieved From: file:///C:/Users/arseam/Downloads/Article64.pdf
5. Casanova, H. (2022). Distributed computing research issues in grid computing. ACM SIGAct News 2002, 33, 50–70. DOI: https://doi.org/10.1145/582475.582486
6. De Donno, M.; Tange, K.; Dragoni, N. (2019). Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. IEEE Access 2019, 7, 150936–150948. DOI: https://doi.org/10.1109/ACCESS.2019.2947652
7. Greenstein, S.M . (1997). Lock-in and the costs of switching mainframe computer vendors: What do buyers see? Ind. Corp. Chang. 1997, 6, 247–273. Retrieved From: https://ideas.repec.org/a/oup/indcch/v6y1997i2p247-73.html
8. Iansiti, M.; Clark, K.B. (1994). Integration and dynamic capability: Evidence from product development in automobiles and mainframe computers. Ind. Corp. Chang. 1994, 3, 557– 605. DOI: https://doi.org/10.1093/icc/3.3.557
9. Korada, L. (2024). Data Poisoning-What Is It and How It Is Being Addressed by the Leading Gen AI Providers. European Journal of Advances in Engineering and Technology, 11(5), 105-109. DOI: http://dx.doi.org/10.5281/zenodo.13318796
10. Laxminarayana Korada, D. M. K., et. al (2024). Artificial Intelligence on The Administration Of Financial Markets. DOI: https://doi.org/10.62441/nano-ntp.vi.2493
11. Rekha, P., Saranya, T., Preethi, P., et. al (2017). Smart Agro Using Arduino and GSM. International Journal of Emerging Technologies in Engineering Research (IJETER) Volume, 5. Retrieved From: https://www.ijeter.everscience.org/Manuscripts/Volume- 5/Issue-3/Vol-5-issue-3-M-09.pdf
12. Schwiegelshohn, U.; Badia, R et al. (2010). Perspectives on grid computing. Future Gener.
13. Comput. Syst. 2010, 26, 1104–1115. DOI: https://doi.org/10.1016/j.future.2010.05.010 Singh, S. K., Choudhary, S. K., et. al (2022). Comparative Analysis of Machine Learning
14. Models and Data Analytics Techniques for Fraud Detection in Banking System. International Journal of Core Engineering & Management, 7(1), 64. ISSN 2348-9510.
15. Retrieved From:
16. https://www.researchgate.net/publication/384678305_COMPARATIVE_ANALYSIS_OF_MACHINE_LEARNING_MODELS_AND_DATA_ANALYTICS_TECHNIQUES_FOR_FRAUD_DETECTION_IN_BANKING_SYSTEM
17. Sujithra, M., Velvadivu, P., Rathika, J., (2022). A Study on Psychological Stress Of Working Women In Educational Institution Using Machine Learning. In 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE. DOI: https://doi.org/10.1109/ICCCNT54827.2022.9984460
18. Suresh, K., Reddy, P. P., et. al (2019). A novel key exchange algorithm for security in internet of things. Indones. J. Electr. Eng. Comput. Sci, 16(3), 1515-1520. DOI: http://dx.doi.org/10.11591/ijeecs.v16.i3.pp1515-1520
19. Yeo, C.S.; Buyya, R. (2006). A taxonomy of market-based resource management systems for utility-driven cluster computing. Softw. Pract. Exp. 2006, 36, 1381–1419. DOI: https://doi.org/10.1002/spe.725
20. Yu, J.; Buyya, R. (2005). A taxonomy of workflow management systems for grid computing. J. Grid Comput. 2005, 3, 171–200. Retrieved From: https://link.springer.com/article/10.1007/s10723-005-9010-8
21. Yuan, J.; Xiao, H.; Shen, Z.; et. al (2023). ELECT: Energy-efficient intelligent edge–cloud collaboration for remote IoT services. Future Gener. Comput. Syst. 2023, 147, 179–194. DOI: https://doi.org/10.1016/j.future.2023.04.030