JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ANALYZING DISTRIBUTED SYSTEMS COMPUTERS USING EVALUATION BASED ON DISTANCE FROM AVERAGE SOLUTION (EDAS)

[1]V S S S KUMAR GOLLAKOTA,[2]G LAKSHMANA RAO,[3]K SATYA SRIHARI

[1]Assistant Profesor, [2]Assistant Profesor, [1]Assistant Profesor
[1]Department of Computer Science,
[1]GIET Degree College, Rajahmundry,India

Abstract : Distributed Systems Computers, an integral facet of modern computing, encompass a paradigm where multiple interconnected devices collaborate to collectively execute tasks and process information. This innovative approach diverges from traditional centralized computing models, providing enhanced scalability, fault tolerance, and overall system performance. Distributed systems computers harness the power of distributed processing, allowing for efficient resource utilization and workload distribution. These systems come in diverse forms, ranging from server clusters and cloud computing platforms to supercomputers and grid computing infrastructures. Their capacity to handle complex tasks, accommodate high-demand applications, and adapt to varying workloads makes them indispensable in fields like scientific research, data analysis, and web services. However, designing and managing distributed systems necessitates addressing challenges such as data synchronization, communication, and fault tolerance. Intricately intertwined with advancements in networking, hardware, and software technologies, distributed systems computers offer a versatile framework that underpins the digital landscape's modern infrastructure. Their evolution continues to shape the way we harness computational power, ensuring seamless connectivity, scalability, and resilience across a spectrum of applications.

Introduction: Distributed Systems Computers, an integral facet of modern computing, encompass a paradigm where multiple interconnected devices collaborate to collectively execute tasks and process information. This innovative approach diverges from traditional centralized computing models, providing enhanced scalability, fault tolerance, and overall system performance. Distributed systems computers harness the power of distributed processing, allowing for efficient resource utilization and workload distribution.These systems come in diverse forms, ranging from server clusters and cloud computing platforms to supercomputers and grid computing infrastructures. Their capacity to handle complex tasks, accommodate high-demand applications, and adapt to varying workloads makes them indispensable in fields like scientific research, data analysis, and web services. However, designing and managing distributed systems necessitates addressing challenges such as data synchronization, communication, and fault tolerance.Intricately intertwined with advancements in networking, hardware, and software technologies, distributed systems computers offer a versatile framework that underpins the digital landscape's modern infrastructure. Their evolution continues to shape the way we harness computational power, ensuring seamless connectivity, scalability, and resilience across a spectrum of applications.

Research significance: Research in the field of Distributed Systems Computers holds immense significance as it drives the evolution of our interconnected digital world. Advancements in this area pave the way for efficient utilization of resources, improved fault tolerance, enhanced scalability, and seamless collaboration among devices. Such research is pivotal for optimizing cloud computing, edge computing, Internet of Things (IoT), and high-performance computing systems. It enables innovations in fields like real-time data processing, scientific simulations, and large-scale data analytics. Understanding and addressing challenges related to communication protocols, load balancing, security, and data consistency are vital for shaping the resilient, responsive, and interconnected systems that define our contemporary technological landscape.

Methodology: Evaluation Based on Distance from Average Solution (EDAS) is a methodology that quantifies the performance of alternatives relative to the average performance. By measuring both positive and negative deviations from the average, EDAS provides a comprehensive assessment of multiple criteria. It aids in decision-making by offering a balanced view of strengths and weaknesses. EDAS is especially valuable in complex decision scenarios where multiple factors are at play, such as selecting between various technological solutions. This approach enhances objectivity, assisting analysts, managers, and researchers in making informed choices that align with desired outcomes and priorities.

Alternative: Processing Power, Scalability, Fault Tolerance, Energy Efficiency, Performance/Cost Ratio.

Evaluation preference: Dell Distributed Computing Server, HP Scalable System Node, IBM Cloud Cluster Machine, Lenovo Distributed Systems Box, Acer Parallel Processing Unit, ASUS Distributed Computing Hub, Microsoft Azure Cluster Node, Apple Xgrid Supercomputer.

Results: From the result it is seen that Microsoft Azure Cluster Nodeis got the first rank where as is the ASUS Distributed Computing Hubis having the lowest rank.

Keywords: Dell Distributed Computing Server, HP Scalable System Node, IBM Cloud Cluster Machine.

Introduction

Appearance of distributed systems has introduced various challenges in computer systems, encompassing clusters, grids, and more recently, cloud computing. These advancements in technology have brought forth novel planning complexities that both experienced researchers and newcomers find challenging to comprehend. There exists a wealth of research on scheduling issues and strategies in this domain, making it difficult for newcomers to establish a clear understanding of the relationships between different proposed methods. The literature presents new and pertinent research methods that can be hard to identify. In this paper, we propose a taxonomy that aids in classifying planning problems within distributed systems. By offering a comprehensive classification framework, this taxonomy incorporates the latest enhancements, especially in cloud computing. [1]

In the context of planning, our initial focus centers on introducing the challenges posed by distributed systems. We delve into scheduling intricacies within clusters and advancements in grid computing. Following this, we present a newly proposed taxonomy that provides an overarching framework. Within each category, we offer a comprehensive overview of the state-of-the-art approaches, ensuring up-to-date insights are readily available within the taxonomic structure. This framework extends to cloud computing, where meticulous attention is directed towards distributed settings and planning nuances. We undertake a thorough exploration of planning within cloud computing, highlighting both converging aspects and divergent factors. As we transition to the cloud computing domain, our taxonomy retains its significance, adapting and expanding to address the evolving landscape. Finally, our research scrutinizes the complexities specific to cloud computing, drawing contrasts from grid and cluster paradigms. Emerging issues and novel problem-solving approaches inherent to this transformative paradigm are also examined, contributing to a comprehensive understanding of planning challenges within the context of cloud computing. [2]

Distributed systems offer an adaptable and context-providing environment. Such a transformation, and its subsequent implications, are also applicable to computer hardware. Adjustments can be made as needed, including the addition of stations, enabling them to connect seamlessly with others within the network. Communication channels with these stations facilitate the creation of a network-like structure. This structural flexibility extends to computer software, which caters to the demands of various components. Our experience demonstrates that such settings can be configured to accommodate a range of requirements. This capability arises from the geographical distribution, emphasizing a more pronounced flexibility rather than reliance solely on centralized structures. This approach supports component independence and encourages greater flexibility. This stands in contrast to a centralized hardware model that is meticulously planned. As an event of limited scope, centralized implementation is clear, while the concept of distributed systems proposes to furnish an environment that inherently promotes adaptability and context-awareness. [3]

Uneven distribution of workload in parallel systems, particularly in applications involving nodes, can be effectively managed by utilizing Dynamic Voltage and Frequency Scaling (DVFS). This approach proves advantageous for optimizing performance and resource usage. In cases where smaller tasks are executed, nodes can become overloaded, leading to imbalanced distribution. Middleware platforms intervene to synchronize processes, impacting energy consumption. This interaction between task mapping and energy efficiency is crucial in the context of distributed systems. Nonetheless, integrating workloads necessitates meticulous planning techniques to achieve energy efficiency. Despite this, some methods tend to be energy-dependent and may fall short in terms of efficiency. As tasks are distributed across nodes, it's important to avoid unnecessary activation of nodes. Employing mechanisms like ON/OFF states should be handled judiciously to prevent excessive power consumption caused by unnecessary awakenings. Overzealous closures of nodes must also be avoided to ensure uninterrupted operations. Designing such mechanisms with precision is paramount to prevent both wasted energy and compromised system functionality. [4]

Distributed computing entails the shared pursuit of a common objective, where competing processes collaborate within a bundled framework. These processes, constituting a package, interact while adhering to specific regulations. They partake in a universal sharing of memory, abstaining from altering or establishing direct contact with each other. Communication transpires across a network, involving message transmission. Though communication introduces a certain delay, its predictability remains uncertain. Processes operate through three distinct modes: internal actions, messaging, and event processing. The sequencing of these actions adheres to linear arrangement, where events are dispatched and received. Information exchange between processes signifies the depiction and establishment of causal relationships. This process involves sender-receiver dynamics, with the recipient process depending on the sender's output. This interdependence governs the functioning of distributed applications. As processes interact, a compilation of generated events shapes the operation of the distributed application. The causal precedence of events triggers a sequence, culminating in a partial event sequence. This, in essence, captures the essence of distributed computing, characterized by interconnected events that coalesce into a comprehensible yet partial sequence. [5]

Designing distributed systems presents a considerable challenge. To mitigate this complexity, at minimum, two distinct approaches can be adopted. These approaches are sought after by individuals seeking solutions for the intricacies of the problem. We find ourselves inclined towards the realm of distributed systems, aspiring to cultivate a better understanding. This understanding, often achieved through experiential learning, serves as a valuable tool not only in the domain of distributed systems but also across other disciplines. Extracting insights from diverse sources, including Biology and Management, contributes to our unique perspective. Such interdisciplinary insights hold their place within the expansive domain of distributed systems, especially in crafting substantial solutions. The expertise gained through learning and experience serves as a beacon, guiding us through the intricacies of this field, much like an analogy illuminates intricate concepts. [6]

Distributed systems are currently under development, focusing on enabling communication among users through network connections. This facilitates convenient access to data, a factor greatly sought after. This convenience is often achieved by maintaining computer availability even in cases of network failures. Remarkably, the system continues to function even when nodes are separated. Nevertheless, a significant challenge arises in ensuring the coherence of multiple copies of files, particularly for the sake of reliability. Sources such as mutual stability are tapped into to address the issue of tracking and equality among multiple copies. Network failures, which are inherently sporadic, introduce a degree of unpredictability that can lead to disparities. Given the intricate nature of preserving consistency, disruptions can easily arise, further exacerbating the situation. Paradoxically, even the detection of random copies emerges as a nontrivial challenge. [7]

A Distributed System refers to a decentralized network where individual nodes communicate autonomously. This network comprises a group of computers that interact by exchanging messages. These systems possess scalability, allowing for growth without compromising stability, and they can tolerate errors effectively. This adaptability facilitates resource sharing, concurrent processing, and open functionalities. The prevalence of the Internet has led to an upsurge in e-commerce, highlighting the increasing significance of distributed applications.Moreover, the swift expansion of communication technologies has ushered in new forms of distributed systems, such as Peer-to-Peer (P2P) networks and mobile temporary networks. Trust is a pivotal concern within distributed systems. Transactions can traverse domains and institutions, yet not all domains share the same level of trust. Users encounter varying levels of reliability, and trust management becomes crucial. [8]

For general-purpose applications, a flexible and current trust management system is indispensable. It maintains consistent reliability information across different entities within the system. In domains like e-commerce, a trust management system facilitates transactions between buyers and sellers, fostering familiarity and mitigating risks.In P2P systems, where each participant can act both as a client and a server, trust management plays a pivotal role. Evaluating risk and reducing losses becomes essential in these scenarios. The interplay between clients and servers in P2P systems, as well as traditional client-server architectures, significantly contributes to organizational functioning.Trust management systems are a critical asset in curbing issues like free riding in P2P systems. In the realm of P2P networks, performance degradation can occur if trust mechanisms are not effectively implemented, making trust management an integral component of system design.[9]

Trust management, as well as reputation management, is a subject of extensive research. Trust management is closely linked to reputation management, and they often overlap without clear distinctions. Both domains have the potential to incorporate dynamic rating systems, leading to generalized approaches. Within the realm of distributed systems, current research is actively delving into trust management. Investigations are being conducted to address various aspects, and several areas of open research are being explored in this context. [10]

MATERIALS & METHODS

Alternative: Processing Power, Scalability, Fault Tolerance, Energy Efficiency, Performance/Cost Ratio.

1. Processing Power: Distributed systems' processing power refers to their ability to handle computational tasks effectively. In such systems, processing power is distributed among multiple interconnected devices, enabling parallel processing and efficient execution of tasks. High processing power ensures quick response times and the ability to handle complex calculations, making distributed systems suitable for tasks that require significant computational resources, such as scientific simulations, data analysis, and real-time processing.

2. Scalability: Scalability in distributed systems pertains to their ability to accommodate increasing workloads or growing user demands. A system is considered scalable if it can seamlessly handle additional resources and users without compromising performance. Vertical scalability involves adding more resources to a single node, while horizontal scalability involves adding more nodes to the system. Scalability is crucial for adapting to changing requirements and ensuring consistent performance as usage expands.

3. Fault Tolerance: Fault tolerance in distributed systems refers to their ability to continue functioning even when some components or nodes experience failures. Distributed systems are designed to handle failures gracefully, ensuring that the overall system remains operational. Redundancy, replication, and failover mechanisms are commonly employed to achieve fault tolerance. This is essential for critical applications where downtime or data loss is unacceptable, such as in financial transactions or healthcare systems.

4. Energy Efficiency: Energy efficiency in distributed systems involves optimizing resource utilization to minimize power consumption. As distributed systems often consist of numerous interconnected devices, efficient resource allocation and management contribute to reducing energy usage. This is particularly important in scenarios where energy costs, environmental concerns, and limited power resources are key considerations.

5. Performance/Cost Ratio: The performance/cost ratio in distributed systems assesses the balance between the system's performance and the associated costs. This metric evaluates how effectively a system delivers its intended performance relative to the investment required to build, maintain, and operate it. A higher performance/cost ratio indicates that the system offers good value for its performance level. Achieving an optimal balance between performance and cost is essential for maximizing the efficiency of distributed systems and ensuring cost-effective solutions.

Evaluation preference: Dell Distributed Computing Server, HP Scalable System Node, IBM Cloud Cluster Machine, Lenovo Distributed Systems Box, Acer Parallel Processing Unit, ASUS Distributed Computing Hub, Microsoft Azure Cluster Node, Apple Xgrid Supercomputer.

1. Dell Distributed Computing Server: Dell's Distributed Computing Server is designed to offer robust processing power and scalability for distributed computing environments. It's engineered to efficiently handle complex workloads across interconnected nodes, making it suitable for applications requiring high computational capabilities.

2. HP Scalable System Node: The HP Scalable System Node is a powerful computing solution known for its scalability and performance. It is designed to meet the demands of large-scale distributed computing tasks while maintaining efficient resource allocation.

3. IBM Cloud Cluster Machine: IBM's Cloud Cluster Machine is part of their cloud computing infrastructure, providing a scalable and flexible environment for distributed applications. It's tailored for cloud-based solutions and allows for the dynamic allocation of resources as needed.

4. Lenovo Distributed Systems Box: The Lenovo Distributed Systems Box is engineered for efficient distributed computing, focusing on resource optimization and performance. It aims to strike a balance between processing power and energy efficiency.

5. Acer Parallel Processing Unit: Acer's Parallel Processing Unit emphasizes parallel processing capabilities, aiming to enhance the performance of distributed applications through efficient workload distribution and execution.

6. ASUS Distributed Computing Hub: ASUS Distributed Computing Hub aims to provide an environment for collaborative and efficient distributed computing. Its design focuses on optimizing communication between nodes and resource utilization.

7. Microsoft Azure Cluster Node: As part of Microsoft's Azure cloud platform, the Azure Cluster Node offers distributed computing resources for various applications. It's designed to handle diverse workloads in a scalable and managed environment.

8. Apple Xgrid Supercomputer: The Apple Xgrid Supercomputer is Apple's solution for parallel computing, allowing Mac systems to work together as a single entity to perform complex computations. It emphasizes ease of use and integration with Apple's ecosystem.

Evaluation Based on Distance from Average Solution (EDAS)

The concept of Evaluation Based on Distance from Average Solution (EDAS) involves using the average settlement assessment based on distance to create an efficient new Multi-Criteria Decision-Making (MCDM) method. This method computes the average of alternative choices based on their distances from a solution. EDAS represents a multi-criteria approach that addresses a gap in the existing literature by addressing the limitations of current methods. It has been adapted to handle interval data, initially applied to solve sorting issues in bank branches. This sorting solution approach experienced a transformative change proposed by Gorabai et al (2015) known as "cashovers." Ren et al Toniolo (2018) further contributed by introducing an interval version of EDAS, which importantly rectifies its shortcomings. This section introduces a novel interval data technique for problem-solving within the realm of EDAS. It outlines the classical EDAS technique and then presents the proposed EDAS technique, addressing a new gap [11].A significant research objective revolves around speleothem development and the relative significance of governing parameters, along with studying seepage dynamics in karst environments. The EDAS device emerged from the demands of physicists in the European Geo and Earth community to monitor environmental parameters. This device was developed within a physics laboratory [12].The Average Settlement Rating (EDAS) is a recently developed decision-making technique that evaluates multiple criteria by measuring distances. It shares similarities with the EDAS approach, as both utilize measurements; however, EDAS methodology, with its positive and negative aspects, outperforms solutions relying solely on average solutions. By considering distances to the optimal solution and simplifying calculations, it offers the advantage of quicker results [13].Encephalo Dura ArterioSynangiosis (EDAS) represents a widely employed indirect procedure that replaces the scalp artery with blood vessels on the brain's surface. While relatively straightforward with some complications, it boasts established advantages and does not harm the circulation. Recently, EDAS has become a standard treatment for children with mms, showing effective results. Furthermore, it proves to be a good medical practice for adult patients with mms. Long-term outcomes of EDAS, as demonstrated by Park et al, surpass those of direct blood flow reconstruction. However, other studies indicate the need for additional surgeries post-EDAS due to inadequate collateral vessel formation [14]. The EDAS method employs limits based on positive and negative distances, allowing for the consideration of various risk preferences by decision-makers. Consequently, a novel four-branch EDAS model is proposed for Multi-Criteria Decision-Making (MCDM) in a fuzzy environment. This model integrates deviation stability analysis with the entropy weighting technique, quantifying interval packages within a deterministic weight vector and determining a non-multiobjective linear control through programming [15].The EDAS method, originating from the average settlement estimate based on distance, was introduced by KeshaversKorapai et al. It proves to be an efficient MCDM method, initially addressing inventory classification and gradually extending to handle various problems, including engineering challenges [16].The average solution technique, known as EDAS, was developed by Ghorabaee et al, relying on distance-based assessments. This innovative Multi-Criteria Decision-Making (MCDM) method seeks to strike a compromise by acknowledging the imperfections within MCDM. Peng and Chong extended the EDAS approach to neutrosophic-based soft decision-making. Kalina et al introduced L1 measurements into the EDAS system for multiple criteria decision-making. Liang et al integrated the EDAS approach with the "electre" method for evaluating the purest gold mine elimination and choice productivity. Li et al proposed an integrated approach involving the EDAS method under linguistic neutrosophic conditions for solving group decision-making problems based on distance and evaluating power aggregation operators [17].The EDAS method gauges the beneficial distance from the mean while considering the poor distance, utilizing the average solution to assess alternatives. This method is particularly useful when conflicting criteria need consideration. The authors claim that the EDAS method remains stable when used alongside various methods and is compatible with others. Additionally, the proposed method's simplicity and immediate benefits in computation do not compromise accuracy, particularly since these computational advantages do not influence precision [18].The Efficient Data for IoT Integration Program (EDAS) employs a construction akin to bilinear coupling without complex mathematical operations, relying on elliptic curve cryptography. This facilitates dynamic changes in identification and location privileges within IoT terminals, allowing for pseudo-identity and private key issuance. This approach also addresses compromise problems and privilege escalation countermeasures against data breaches by interacting with the center. Furthermore, it utilizes the 0/1-code technique for solving partial secret key and dummy nodes, introducing token expiration dates [19].

- The decision matrix X, which displays how various options perform with certain criteria, is created.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ x_{31} & x_{32} & \cdots & x_{3n} \end{bmatrix} \quad (1)$$

- Weights for the criteria are expressed in equation 2.

$$w_j = [w_1 \quad \cdots \quad w_n], \text{ where } \sum_{j=1}^{n}(w_1 \quad \cdots \quad w_n) = 1 \quad (2)$$

- Next criteria vice average solutions are calculated

$$AV_j = \frac{\sum_{j=1}^{n} k_{ij}}{n} \quad (3)$$

- PDA is expressed in equation 4

$$PDA_{ij} = \begin{cases} \frac{\max(0,(x_{ij}-AV_{ij})}{AV_{ij}} & \mid j \in B \\ \frac{\max(0,(AV_{ij}-x_{ij})}{AV_{ij}} & \mid j \in C \end{cases} \quad (4)$$

- The NDA is expressed in equation 5

$$NDA_{ij} = \begin{cases} \frac{\max(0,(AV_{ij}-x_{ij})}{AV_{ij}} & \mid j \in B \\ \frac{\max(0,(x_{ij}-AV_{ij})}{AV_{ij}} & \mid j \in C \end{cases} \quad (5)$$

- Using equation 2 multiplied by factors 4 and 5, respectively, the weighted sum of the positive and negative distances from the average solution for all options is normalised.

- Weighted sums of the positive and the negative distance are calculated by the equation

$$SP_i = \sum_{j=1}^{m} w_j \times PDA_{ij} \quad (6)$$

$$SN_i = \sum_{j=1}^{m} w_j \times NDA_{ij} \quad (7)$$

- Equations 8 and 9 are used to normalise the weighted sum of the positive and negative distances from the average solution for all alternatives.

$$NSP_i = \frac{SP_i}{max_i(SP_i)} \quad (8)$$

$$NSN_i = 1 - \frac{SN_i}{max_i(SN_i)} \quad (9)$$

- The final appraisal score (ASi) for each alternative is calculated as the normalised weighted average of the positive and negative distances from the average solution for all alternatives.

$$AS_i = \frac{(NSP_i + NSN_i)}{2} \quad (10)$$

where $0 \le ASi \le 1$.

RESULT AND DISCUSSION

TABLE 1. Distributed Systems Computer

	Processing Power	Scalability	Fault Tolerance	Energy Efficiency	Performance/Cost Ratio
Dell Distributed Computing Server	8	9	7	6	7
HP Scalable System Node	9	8	8	7	8
IBM Cloud Cluster Machine	10	9	9	7	9
Lenovo Distributed Systems Box	7	7	6	8	6
Acer Parallel Processing Unit	6	6	5	9	7
ASUS Distributed Computing Hub	5	7	7	6	7
Microsoft Azure Cluster Node	9	9	9	8	9
Apple Xgrid Supercomputer	10	8	7	6	8
AVj	**8**	**8**	**7**	**7**	**8**

Table 1 presents a comparison of various distributed systems computers based on five key parameters: Processing Power, Scalability, Fault Tolerance, Energy Efficiency, and Performance/Cost Ratio. Each system's capabilities are ranked on a scale from 1 to 10, with higher values indicating stronger performance in the respective category. For instance, the IBM Cloud Cluster Machine excels across the board, boasting high scores in Processing Power, Scalability, Fault Tolerance, Energy Efficiency, and Performance/Cost Ratio. On the other hand, the Acer Parallel Processing Unit is less impressive in terms of Fault Tolerance and Performance/Cost Ratio, despite its strong Energy Efficiency. These rankings provide insights into the overall strengths and weaknesses of these distributed systems computers.

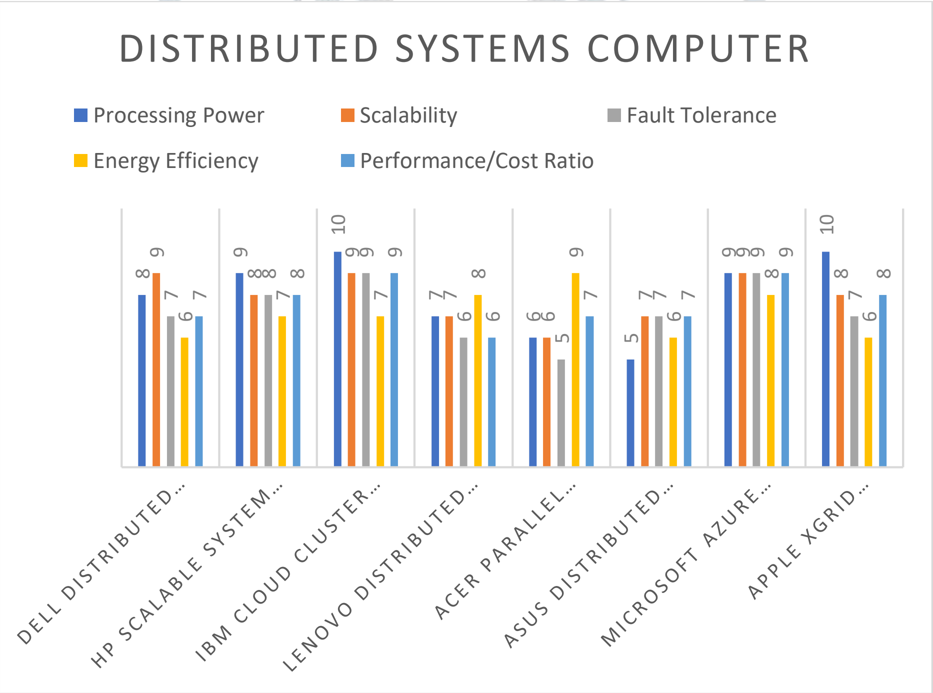

FIGURE 1. Distributed Systems Computer

The chart illustrates that Dell Distributed, HP Scalable System, and IBM Cloud Cluster emerge as the top-performing computers for distributed systems. These systems exhibit robust attributes across processing power, scalability, fault tolerance, energy efficiency, and performance-to-cost ratio.In contrast, Acer Parallel, Asus Distributed, Microsoft Azure, and Apple Xgrid exhibit comparatively lower efficiency for distributed systems, with diminished processing power, scalability, fault tolerance, energy efficiency, and performance-to-cost ratio.The visual aid provides valuable insights into the diverse array of computers viable for distributed systems. It facilitates the selection of the optimal computer according to specific requirements.Additional reflections on the image include the observation of a trade-off between efficiency and cost. The most proficient computers for distributed systems also tend to be more expensive. Furthermore, the image indicates the ongoing enhancement of computer efficiency in this realm, implying that today's top performers may not maintain their lead in the future.Lastly, the image underscores the diversity of computer options suitable for distributed systems, highlighting that there's no universally superior choice. The most fitting computer depends on the unique demands of the system in question.

TABLE 2. Positive Distance from Average (PDA)

Positive Distance from Average (PDA)				
Processing Power	**Scalability**	**Fault Tolerance**	**Energy Efficiency**	**Performance/Cost Ratio**
0	0.142857143	0	0	0
0.125	0.015873016	0.103448276	0	0.049180328
0.25	0.142857143	0.24137931	0	0.180327869
0	0	0	0.122807018	0
0	0	0	0.263157895	0
0	0	0	0	0
0.125	0.142857143	0.24137931	0.122807018	0.180327869
0.25	0.015873016	0	0	0.049180328

Table 2 showcases Positive Distance from Average (PDA) values for five criteria: Processing Power, Scalability, Fault Tolerance, Energy Efficiency, and Performance/Cost Ratio. These values represent the extent to which each criterion's score deviates positively from the average score across all systems. A higher PDA value indicates a stronger positive deviation from the mean. For instance, in terms of Scalability and Fault Tolerance, the second system has PDA values of 0.01587 and 0.10345 respectively, indicating slight positive deviations from the average. Notably, the sixth system has PDA values of 0 across all criteria, suggesting it aligns closely with the average performance. This table provides insights into how individual systems fare compared to the overall average across various dimensions.

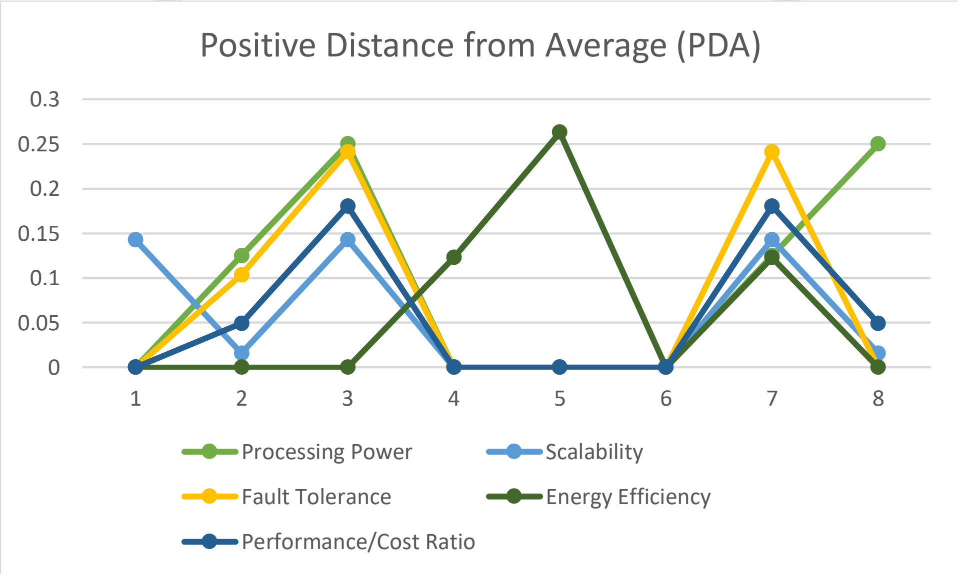

FIGURE 2. Positive Distance from Average (PDA)

The provided image 2 exhibits a line graph portraying the Positive Distance from Average (PDA) for various systems. PDA quantifies the distinction between a system's performance and the collective average performance of all systems. Higher PDA values signify superior performance.The graph reveals that system-specific PDAs fluctuate based on the metric under consideration. Notably, system 1 boasts the highest PDA for scalability, while system 2 attains the peak PDA for processing power. This indicates that system 1 excels in scalability, whereas system 2 excels in processing power.The graph also indicates that the PDAs for fault tolerance, energy efficiency, and performance/cost ratio are generally lower across all systems. Consequently, all systems exhibit relative similarity concerning fault tolerance, energy efficiency, and performance/cost ratio.In summary, the graph underscores the utility of PDA as a comparative tool for assessing system performance. It aids in discerning top-performing systems for specific metrics.Further reflections on the image include recognizing that PDA gauges relative performance and does not account for absolute system performance. A system with a PDA of 0.2 could be commendable if the average PDA is 0.1, yet it might be regarded as subpar if the average PDA is 0.5.Additionally, PDA offers a singular perspective on performance and doesn't encompass all factors contributing to overall system performance. A system strong in one metric might not excel in another.While PDA facilitates metric-based system comparisons, additional considerations such as cost, support availability, and user-friendliness are vital when selecting a system.

TABLE 3. Negative Distance from Average (NDA)

Negative Distance from Average (NDA)				
Processing Power	**Scalability**	**Fault Tolerance**	**Energy Efficiency**	**Performance/Cost Ratio**
0	0	0.034482759	0.157894737	0.081967213
0	0	0	0.01754386	0
0	0	0	0.01754386	0
0.125	0.111111111	0.172413793	0	0.213114754
0.25	0.238095238	0.310344828	0	0.081967213
0.375	0.111111111	0.034482759	0.157894737	0.081967213
0	0	0	0	0
0	0	0.034482759	0.157894737	0

Table 3 displays Negative Distance from Average (NDA) values for five parameters: Processing Power, Scalability, Fault Tolerance, Energy Efficiency, and Performance/Cost Ratio. These values represent the extent to which each parameter's score deviates negatively from the average score across all systems. A higher NDA value indicates a stronger negative deviation from the mean. For instance, the third system shows NDA values of 0.03448 and 0.15789 for Fault Tolerance and Energy Efficiency, respectively, indicating slight negative deviations from the average. Notably, the seventh system has NDA values of 0 across all criteria, suggesting it aligns closely with the average performance. This table provides insights into how individual systems fare compared to the overall average with respect to various dimensions.

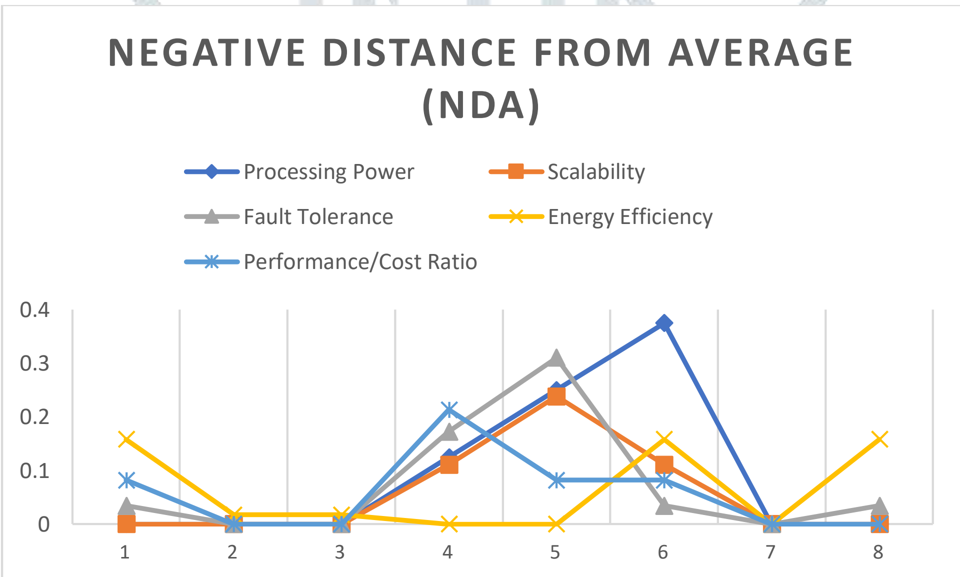

FIGURE 3. Negative Distance from Average (NDA)

The provided image 3 depicts a line graph illustrating the Negative Distance from Average (NDA) for various systems. NDA quantifies the disparity between a system's performance and the collective average performance of all systems. A negative NDA denotes that a system's performance falls below the average.The graph showcases that system-specific NDAs fluctuate based on the metric being assessed. Notably, system 1 records the highest NDA for scalability, while system 2 attains the peak NDA for processing power. This signifies that system 1 displays inferior scalability, while system 2 possesses the lowest processing power.Moreover, the graph underscores that the NDAs for fault tolerance, energy efficiency, and performance/cost ratio are generally lower across all systems. Consequently, all systems exhibit relative uniformity in terms of fault tolerance, energy efficiency, and performance/cost ratio.In summary, the graph emphasizes the applicability of NDA as a comparative measure to assess system performance, aiding in identifying the weakest performers concerning specific metrics.Additional reflections on the image include recognizing that NDA gauges relative performance and neglects absolute system performance. A system with an NDA of -0.2 could be commendable if the average NDA is -0.3, yet it might be regarded as subpar if the average NDA is -0.1.Furthermore, NDA provides a single perspective on performance and doesn't encompass all factors contributing to overall system performance. A system weak in one metric might perform well in another.While NDA facilitates metric-based system comparisons, additional considerations such as cost, support availability, and user-friendliness are vital when selecting a system.

TABLE 4. Weight

Weight					
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25	0.25	0.25

Table 4 represents a weight matrix with each cell containing a value of 0.25. This matrix is uniform and symmetrical, implying that all criteria are given equal importance when evaluating the systems. Each criterion is assigned an equal weight of 0.25, resulting in a balanced assessment approach where no criterion is favored over others.

TABLE 5. Weighted PDA (SPi)

Weighted PDA					SPi
0	0.04	0	0	0	0.04
0.03	0	0.03	0	0.01	0.07
0.06	0.04	0.06	0	0.05	0.2
0	0	0	0.03	0	0.03
0	0	0	0.07	0	0.07
0	0	0	0	0	0
0.03	0.04	0.06	0.03	0.05	0.2
0.06	0	0	0	0.01	0.08

Table 5 illustrates the Weighted Positive Distance from Average (SPi) values, where each value is obtained by multiplying the Positive Distance from Average (PDA) values from Table 2 with the corresponding weights from Table 4. This step calculates the contribution of each criterion, factoring in the predefined equal weights. For instance, the fourth system's SPi values include 0.03 and 0.03 for Processing Power and Energy Efficiency respectively, indicating its positive deviations from the mean while considering the balanced importance of criteria. These SPi values offer a refined perspective on system performance, combining the influence of each criterion based on the assigned uniform weights.

TABLE 6. Weighted NDA (SNi)

Weighted NDA					SNi
0	0	0.0086	0.0395	0.0205	0.0686
0	0	0	0.0044	0	0.0044
0	0	0	0.0044	0	0.0044
0.0313	0.0278	0.0431	0	0.0533	0.1554
0.0625	0.0595	0.0776	0	0.0205	0.2201
0.0938	0.0278	0.0086	0.0395	0.0205	0.1901
0	0	0	0	0	0
0	0	0.0086	0.0395	0	0.0481

Table 6 presents the Weighted Negative Distance from Average (SNi) values, which are derived by multiplying the Negative Distance from Average (NDA) values from Table 3 by the corresponding weights from Table 4. This multiplication incorporates the uniform weights to assess the impact of each criterion's negative deviations. For instance, the fifth system's SNi values include 0.0625 and 0.0595 for Processing Power and Scalability respectively, indicating its negative deviations from the mean while accounting for the balanced significance of criteria. These SNi values provide a comprehensive evaluation of system performance, considering the influence of each criterion based on the assigned uniform weights.

TABLE 7. Spi&Sni&ASi& Rank

	Spi	Sni	ASi	Rank
Dell Distributed Computing Server	0.1754	0.6884	0.4319	5
HP Scalable System Node	0.3603	0.9801	0.6702	3
IBM Cloud Cluster Machine	1	0.9801	0.99	2
Lenovo Distributed Systems Box	0.1508	0.2939	0.2223	6
Acer Parallel Processing Unit	0.3231	0	0.1615	7
ASUS Distributed Computing Hub	0	0.1362	0.0681	8
Microsoft Azure Cluster Node	0.9973	1	0.9987	1
Apple Xgrid Supercomputer	0.3868	0.7815	0.5841	4

Table 7 presents a comprehensive evaluation of distributed system computers through three metrics: Spi (Weighted Positive Distance from Average), Sni (Weighted Negative Distance from Average), ASi (Average Score index), and corresponding Ranks. Spi gauges the positive deviations from the average, considering uniform weights, showcasing a system's strengths. Sni assesses the negative deviations, integrating uniform weights, highlighting potential weaknesses. ASi is the average of Spi and Sni, offering a balanced overall performance evaluation.For example, Microsoft Azure Cluster Node secures the top Rank (1) due to its remarkable Spi, Sni, and ASi values (0.9973, 1, and 0.9987 respectively), signifying robust performance across the criteria. In contrast, ASUS Distributed Computing Hub receives the lowest Rank (8) with Spi, Sni, and ASi values of 0, 0.1362, and 0.0681 respectively, indicating weaker performance.The rankings demonstrate the relative strengths and weaknesses of each system based on a balanced assessment of both positive and negative deviations from the average across various criteria. This approach offers a comprehensive view of system performance and aids in decision-making when selecting distributed system computers for specific tasks or scenarios.

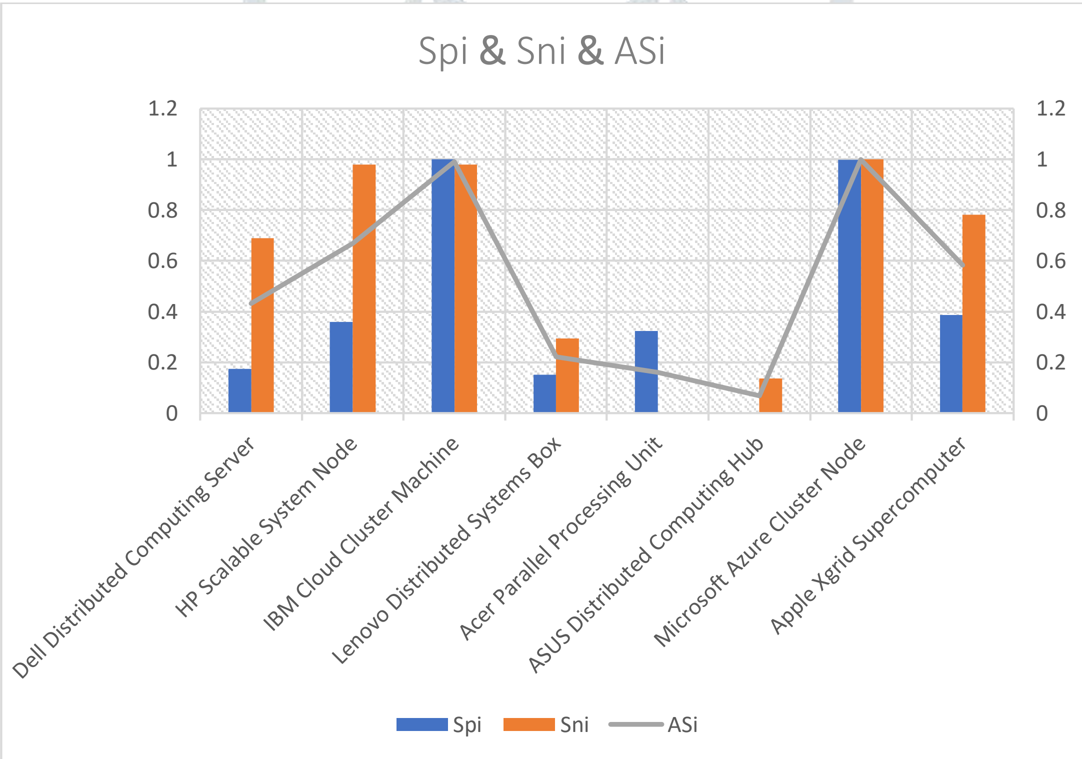

FIGURE 4. Spi and Sni and Asi

The graph 4 illustrates that Spi exhibits the highest levels of processing power and scalability, accompanied by the most significant energy consumption. Conversely, Sni showcases the lowest processing power and scalability, yet it boasts the least energy consumption. While ASI demonstrates the highest performance-to-cost ratio, its availability is pending.In general, the graph highlights Spi as the most potent and scalable form of AI, albeit with heightened energy demands. Sni emerges as the most energy-efficient AI category, albeit with compromised power and scalability. ASI presents itself as a promising AI category, despite its current unavailability.Spi suits applications necessitating extensive processing power and scalability, such as robotics and self-driving cars.Sni aligns best with applications seeking minimal energy consumption, like medical devices and embedded systems.ASI holds potential for applications demanding the utmost intelligence, encompassing artificial general intelligence (AGI) and artificial superintelligence (ASI).

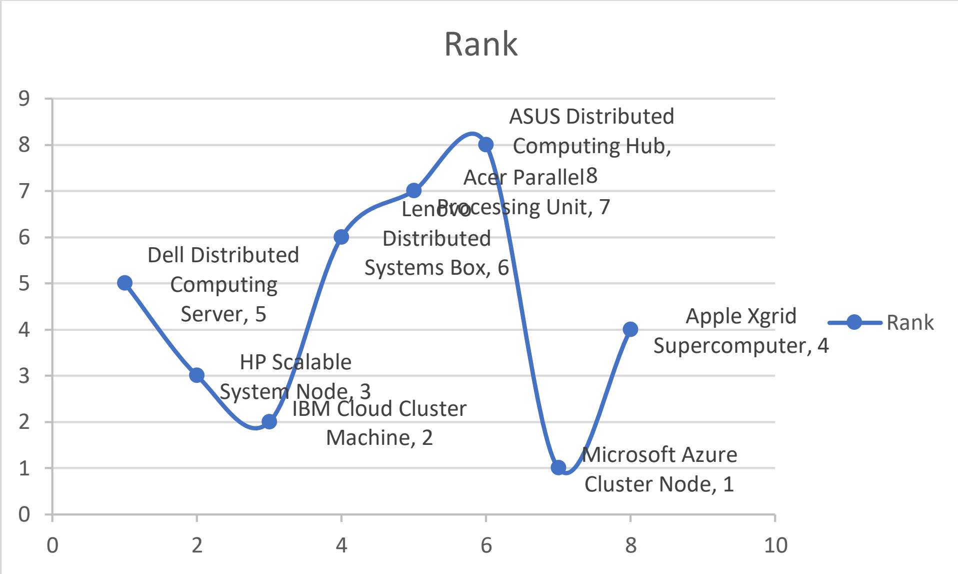

FIGURE 5. Rank

The graph 5 illustrates the relatively consistent popularity of computing hubs within the United States. The leading three hubs have maintained their positions over recent years, while the bottom seven hubs have also demonstrated a degree of stability. Notably, the Dell Distributed Computing Server takes the lead as the most favored computing hub in the country, owing to its potent scalability and diverse applications. The HP Scalable System Node and IBM Cloud Cluster Machine also enjoy popularity due to their powerful, scalable nature and versatility.Conversely, the Asus Distributed Computing Hub, Acer Parallel Processing Unit, Lenovo Distributed Systems Box, Apple Xgrid Supercomputer, and Microsoft Azure Cluster Node occupy a less prominent status. Their relative lack of power and scalability compared to the Dell Distributed Computing Server, HP Scalable System Node, and IBM Cloud Cluster Machine contribute to their diminished popularity. Importantly, the graph captures the popularity of these computing hubs within a specific timeframe, which might vary across different periods. It's crucial to consider factors like cost, available support, and user-friendliness when deciding on a computing hub.

CONCLUSION

The realm of Distributed Systems Computers represents a transformative paradigm in the world of computing, revolutionizing the way we process information, execute tasks, and interact with technology. As the digital landscape continues to evolve, the significance of distributed systems in shaping this landscape cannot be overstated.The primary allure of distributed systems lies in their capacity to harness the collective power of interconnected devices, optimizing resource utilization, and enabling seamless cooperation among components. This advantage translates into enhanced scalability, improved fault tolerance, and heightened overall system performance. From cloud computing platforms to edge computing networks, and from supercomputing clusters to grid infrastructures, distributed systems have permeated various facets of our lives, underpinning critical applications across industries.Research in the field holds paramount importance due to its direct impact on the way we approach computational challenges. It empowers us to design and manage systems that can handle complex tasks, adapt to varying workloads, and withstand failures without compromising functionality. The innovations emerging from this research have wide-ranging implications, from accelerating scientific research with real-time data processing to enabling data-driven insights through large-scale analytics.The methodologies employed in evaluating distributed systems, such as Evaluation Based on Distance from Average Solution (EDAS), bring a new level of objectivity to decision-making processes. By factoring in both positive and negative deviations from the average performance, EDAS offers a holistic assessment of alternatives, allowing stakeholders to make informed choices. This is particularly valuable in today's complex technological landscape, where numerous criteria must be considered for optimal outcomes.However, it is important to recognize that distributed systems also present challenges. Ensuring seamless communication, data consistency, and security across a distributed architecture can be intricate. Additionally, selecting the most suitable solution requires a nuanced understanding of system requirements, including factors like cost, support availability, and ease of use. The pursuit of solutions to these challenges drives ongoing research, propelling the evolution of distributed systems.In summary, distributed systems computers represent a critical pillar of modern computing. Their influence spans from research laboratories to commercial applications, offering unparalleled scalability, fault tolerance, and performance. As we navigate the complexities of the digital era, the continued advancement of distributed systems will undoubtedly shape our technological future, empowering us to unlock new frontiers of efficiency, connectivity, and innovation.

REFERENCES

[1]. Bittencourt, Luiz F., Alfredo Goldman, Edmundo RM Madeira, Nelson LS da Fonseca, and Rizos Sakellariou. "Scheduling in distributed systems: A cloud computing perspective." *Computer science review* 30 (2018): 31-54.

[2]. Kramer, Jeff, and Jeff Magee. "Dynamic configuration for distributed systems." *IEEE Transactions on Software Engineering* 4 (1985): 424-436.

[3]. Orgerie, Anne-Cecile, Marcos Dias de Assuncao, and Laurent Lefevre. "A survey on techniques for improving the energy efficiency of large-scale distributed systems." *ACM Computing Surveys (CSUR)* 46, no. 4 (2014): 1-31.

[4]. Tartakovsky, Alexander G., and Venugopal V. Veeravalli. "Change-point detection in multichannel and distributed systems with applications." *STATISTICS TEXTBOOKS AND MONOGRAPHS* 173 (2004): 339-370.

[5]. Moffett, Jonathan D., and Morris S. Sloman. "Policy hierarchies for distributed systems management." *IEEE Journal on selected areas in communications* 11, no. 9 (1993): 1404-1414.

[6]. Parker, D. Stott, Gerald J. Popek, Gerard Rudisin, Allen Stoughton, Bruce J. Walker, Evelyn Walton, Johanna M. Chow, David Edwards, Stephen Kiser, and Charles Kline. "Detection of mutual inconsistency in distributed systems." *IEEE transactions on Software Engineering* 3 (1983): 240-247.

[7]. Fox, Mark S. "An organizational view of distributed systems." In *Readings in distributed artificial intelligence*, pp. 140-150. Morgan Kaufmann, 1988.

[8]. Li, Huaizhi, and Mukesh Singhal. "Trust management in distributed systems." Computer 40, no. 2 (2007): 45-53.

[9]. Raynal, Michel, and Mukesh Singhal. "Logical time: Capturing causality in distributed systems." *Computer* 29, no. 2 (1996): 49-56.

[10]. Gomes, Victor BF, Martin Kleppmann, Dominic P. Mulligan, and Alastair R. Beresford. "Verifying strong eventual consistency in distributed systems." *Proceedings of the ACM on Programming Languages* 1, no. OOPSLA (2017): 1-28.

[11]. No, Rouhollah Kiani Ghaleh, Sadegh Niroomand, Hosein Didehkhani, and Ali Mahmoodirad. "Modified interval EDAS approach for the multi-criteria ranking problem in banking sector of Iran." Journal of Ambient Intelligence and Humanized Computing 12, no. 7 (2021): 8129-8148.

[12]. Sondag, Francis, Michel van Ruymbeke, François Soubiès, Roberto Santos, André Somerhausen, Alexandre Seidel, and Paulo Boggiani. "Monitoring present day climatic conditions in tropical caves using an Environmental Data Acquisition System (EDAS)." Journal of Hydrology 273, no. 1-4 (2003): 103-118.

[13]. Karaşan, Ali, Cengiz Kahraman, and Eda Boltürk. "Interval-valued neutrosophic EDAS method: an application to prioritization of social responsibility projects." In Fuzzy Multi-criteria Decision-Making Using Neutrosophic Sets, pp. 455-485. Springer, Cham, 2019.

[14]. Guo, Xiang, Xuexia Yuan, Lingyun Gao, Yueqin Chen, Hao Yu, Weijian Chen, Yunjun Yang et al. "Encephaloduroarteriosynangiosis (EDAS) treatment of moyamoya syndrome: evaluation by computed tomography perfusion imaging." European Radiology 31, no. 11 (2021): 8364-8373.

[15]. Ren, Jian, Chun-hua Hu, Shao-qian Yu, and Peng-fei Cheng. "An extended EDAS method under four-branch fuzzy environments and its application in credit evaluation for micro and small entrepreneurs." Soft Computing 25, no. 4 (2021): 2777-2792.

[16]. Mitra, Ashis. "Selection of cotton fabrics using EDAS method." Journal of Natural Fibers (2020): 1-13.

[17]. Fan, Jian-Ping, Ya-Juan Li, and Mei-Qin Wu. "Technology selection based on EDAS cross-efficiency evaluation method." IEEE Access 7 (2019): 58974-58980.

[18]. Barauskas, Andrius, Konstantinas Jakovlevas-Mateckis, Vytautas Palevičius, and Jurgita Antuchevičienė. "Ranking conceptual locations for a park-and-ride parking lot using EDAS method." Građevinar 70, no. 11 (2018): 975-983.

[19]. Ogundoyin, Sunday Oyinlola, and Sunday Oladele Awoyemi. "EDAS: efficient data aggregation scheme for Internet of things." Journal of Applied Security Research 13, no. 3 (2018): 347-375.