Review on Fault Tolerance Techniques in Cloud Computing

Gattu Prasad
Ph.D. Scholar, Department of Computer Science
Puducherry Technological University

Dr. E. Karunakaran
Professor, Department of Computer Science and Engineering
Puducherry Technological University

ABSTRACT

With the rapid growth of the internet and its users, cloud computing has emerged as a promising platform for both business and non-business users due to its flexibility, quality of service, and on-demand capabilities. It is an adaptable technology that integrates software and resources in a dynamically scalable manner. However, the dynamic nature of cloud environments leads to various unforeseen faults and failures. Fault-tolerance techniques are essential to ensure high availability and reliability in cloud computing systems. This review focuses on fault-tolerance within the context of cloud computing. Cloud-based environments have recently introduced new challenges in supporting fault-tolerance, paving the way for innovative strategies, architectures, and standards. We provide a comprehensive overview of cloud computing, covering both fundamental and advanced concepts. Key fault-tolerance components, system-level metrics, and the importance of fault-tolerance in cloud computing are highlighted. Additionally, we discuss state-of-the-art proactive and reactive fault-tolerance approaches. We also explore current research efforts on cloud computing fault-tolerance architectures and frameworks. Finally, we outline future research directions for advancing fault-tolerance in cloud computing.

Keywords:Cloud computing, fault-tolerance,, fault tolerance frameworks,emerging cloud technologies.

I. INTRODUCTION

Cloud computing significantly simplifies resource sharing and reduces computational costs. The National Institute of Standards and Technology (NIST), a non-regulatory agency within the United States Department of Commerce, defines cloud computing as: "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1].Cloud computing offers numerous benefits and features, allowing enterprises to develop cloud platforms tailored to their business models and customer needs. It supports multiple service models, including Infrastructure as a Service (IaaS) and Application as a Service (AaaS) [2]. Alongside these, Platform as a Service (PaaS) can be deployed in various forms—public, private, hybrid, and community clouds—based on strategies developed over the past decade to create new revenue streams. The key features and advantages of cloud computing [3]–[6] are summarized below.

(i) Multi-tenancy: Cloud computing is designed to support a multi-tenant model, allowing multiple end-users to share the same application on a common infrastructure while maintaining their privacy and security [7], [8].

(ii) Resource Pooling: Computing resources are pooled to serve multiple end-users within the multi-tenant model. Resources are dynamically assigned and reassigned based on the varying demands of end-users.

(iii) Dynamic Resource Provisioning: Resources are created and terminated on the fly to meet the current demands of end-users, rather than being pre-provisioned for peak-load scenarios. This dynamic allocation helps reduce operating costs.

(iv) On-demand Self-service: End-users can independently provision their cloud computing resources without human intervention, using a web-based self-service portal.

(v) Elasticity and Scalability: Resources are provisioned and released on-demand, often automatically, ensuring the application has the exact capacity needed at any given time. This allows for efficient scaling in and out based on end-user demand, providing a well-known set of advantages, such as cost-efficiency and performance optimization.

A. ADVANTAGES OF CLOUD COMPUTING

1) **Cost Savings**: Cloud computing significantly reduces costs for organizations by offering infrastructure resources through pay-as-you-go pricing models [9],[15]. This eliminates the need for upfront capital investment in infrastructure, allowing end-users and organizations to simply rent resources from cloud providers based on their needs.

2) **Disaster Recovery**: In cloud computing, service providers typically implement their ICT (Information and Communications Technology) infrastructure across multiple geographical locations. This strategy ensures effective recovery from natural or human-induced disasters, maintaining continuity for end-users and enabling quick data recovery.
3) **Sustainability**: Hosting applications in the cloud is more environmentally friendly compared to on-premises solutions. Recent studies demonstrate that adopting cloud services and virtual data options can significantly reduce carbon footprints and improve energy efficiency.
4) **Easy Backup and Data Restoration**: On-premises data storage often has limited capacity and backing up data can be time-consuming. Cloud computing, with its vast data storage capabilities, simplifies the process of backing up and recovering large amounts of data.
5) **Automatic Software Integration**: In the cloud, applications are automatically updated and software integrations are carried out seamlessly, eliminating the need for manual, time-consuming updates typically associated with on-premises systems.

As a result, cloud computing offers compelling features and provides significant opportunities [10], [11] for IT-based businesses and cloud infrastructure owners (cloud service/solution providers). However, fault-tolerance development within cloud computing is still in its early stages and requires thorough attention. Cloud architectures are complex, consisting of multiple interconnected servers housed in data centers [12],[14]. To maintain reliability, designing fault avoidance and prevention mechanisms [15] is essential to address failures caused by hardware or software issues.

Fault-tolerance refers to a system's ability to continue providing services despite the occurrence of faults [16]. Cloud computing systems can be centralized, decentralized, or distributed (forming highly complex infrastructures), and due to their complexity, they are vulnerable to three main problems: Failures, Errors, and Faults. A **failure** occurs when a system cannot perform its specified function correctly. An **error** arises when one or more system states deviate from the expected sequence, potentially leading to service disruptions.

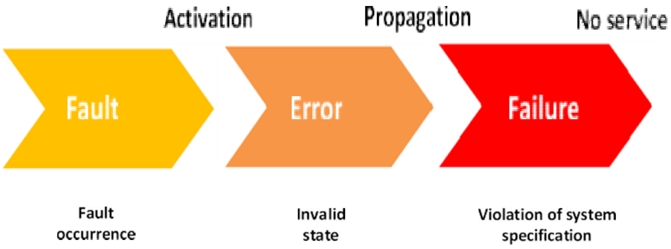

Fig.1. Relationship: Fault, error, and failure.

A ***fault*** is the assumed cause of the error, such as a software bug, human error, or hardware power failure [17]. Faults can lead to errors, which may result in single or multiple failures [18].The relationship between faults, errors, and failures [19] is illustrated in Fig. 1 [20]. Various system-level tools, such as HAProxy, SHelp, Assure, Hadoop, and Amazon EC2, are employed to implement fault-tolerance techniques in cloud environments [21], [18].

II. CLOUD COMPUTING: BACKGROUND AND RELATED CONCEPTS

In this section, we provide an overview of the key concepts related to cloud computing. Additionally, we explore how cloud architectures can be implemented within centralized, decentralized, and distributed systems. Generic research challenges in cloud computing are discussed in [26].

A. CLOUD COMPUTING OVERVIEW

Cloud computing is designed to deliver internet-based computing services using a pool of shared resources. It enables on-demand, seamless data processing and resource sharing across computers and other devices, offering significant benefits in terms of cost reduction, flexibility, and scalability. Typically, cloud computing consists of four main components: the client, data center, distributed servers, and virtualization/virtual machines.

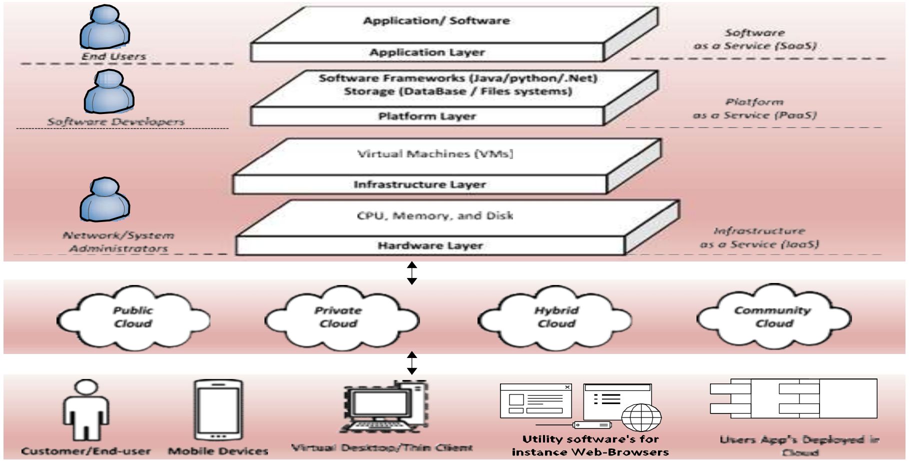
End Users
Software Developers
Network/System Administrators
Application/ Software
Application Layer
Software Frameworks (Java/python/.Net)
Storage (DataBase / Files systems)
Platform Layer
Virtual Machines (VMs)
Infrastructure Layer
CPU, Memory, and Disk
Hardware Layer
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Public Cloud
Private Cloud
Hybrid Cloud
Community Cloud
Customer/End-user
Mobile Devices
Virtual Desktop/Thin Client
Utility software's for instance Web-Browsers
Users App's Deployed in Cloud

Fig. 2 Cloud computing architecture

- ***Clients:*** End-user devices such as computers, laptops, mobile phones, and tablets that are used to exchange information with the cloud.
- ***Data Center:*** A collection of servers and ICT (Information and Communication Technology) infrastructure where cloud applications and services are hosted by the service provider.
- ***Distributed Servers:*** Servers located in different geographical regions, managed by the service provider, to ensure resilience, security, and high availability for end-users.

- ***Virtualization/Virtual Machines:*** A technology that abstracts physical resources such as servers, storage, and networking, allowing for flexible and efficient resource management.

A simplified cloud computing architecture is shown in Fig. 2. Generally, cloud computing architecture can be divided into four layers: the hardware layer, the infrastructure layer, the platform layer, and the application layer. Additionally, cloud services are grouped into three main business models: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). These services can be deployed using different types of clouds: Public, Private, Hybrid, and Community clouds. Below is a brief description of each layer:

1) Hardware Layer:

This layer consists of the physical hardware components, such as servers, routers, switches, power supplies, and cooling systems. These physical resources, typically deployed in data centers, form the foundation of the cloud infrastructure.

2) Infrastructure Layer:

This layer provides an abstraction of the hardware layer, typically through the use of hypervisors, which create a virtual environment on top of the physical hardware. It is responsible for managing compute, storage, and network resources.

3) Platform Layer:

Built on top of the infrastructure layer, this layer provides software frameworks (e.g., Java, Python, .NET) that enable developers to create and deploy applications. It includes APIs and tools that support database management and storage solutions for web applications.

4) Application Layer:

This layer hosts and manages cloud applications, making them available to end-users over the internet on a subscription or on-demand basis. Cloud architectures are highly modular, with each layer being loosely coupled, allowing for better support and management of a wide range of applications compared to traditional dedicated server setups.

B. TYPES OF CLOUD

Cloud computing can be categorized into four types based on varying levels of security and management requirements: Public, Private, Hybrid, and Community clouds. Below is a brief description of each type:

1. ***Public Clouds:*** The entire computing infrastructure is hosted on the premises of a cloud service provider. Public clouds offer the lowest level of security and data control since the resources are shared with other users. However, there are no upfront costs for using public cloud services. Examples of public clouds include AWS/EC2 (Amazon), Microsoft Azure, and Google Cloud Platform.
2. ***Private Clouds:*** In a private cloud, the infrastructure is either located on-premises or exclusively used by a single organization. This type of cloud offers the highest level of security and data control, but it comes with an upfront cost for setting up and managing the infrastructure. Common examples include Eucalyptus Systems, OpenNebula, and OpenStack.
3. ***Hybrid Clouds:*** A hybrid cloud combines the advantages of both public and private clouds. Critical and sensitive applications are hosted on a private cloud, while less sensitive applications can be deployed on a public cloud. This approach provides flexibility and potential cost savings, as it allows organizations to use the strengths of both types of clouds.
4. ***Community Clouds:*** Community clouds are shared infrastructures used by a group of organizations that have similar objectives, concerns, or regulatory requirements. This collaborative effort allows organizations to share resources securely while maintaining privacy and data control.

As illustrated in Fig. 2, end-users and network devices can be deployed across various cloud deployment models, including public, private, hybrid, and community clouds, depending on the specific computing requirements. These cloud models enable businesses to achieve their goals while reducing costs. However, ensuring resiliency in such environments is a challenge, and fault-tolerance plays a crucial role in maintaining seamless operations and delivering a high-quality user experience.

III. FAULT-TOLERANCE CONCEPTS IN CLOUD COMPUTING

This section explains fault-tolerance in cloud computing and related concepts to build a foundational understanding. Faults in the cloud environment can be classified into two main categories: Crash Faults and Byzantine/Arbitrary Faults. Crash faults cause system failures, such as process crashes or power-related issues, while Byzantine/Arbitrary faults result in unpredictable system behavior, deviating from normal operations.In a cloud environment, faults often manifest as failures of resources like applications, storage, or hardware, which can impact end-users by causing performance degradation or system outages. Faults in cloud computing can be classified into three types:

- ***Transient Faults:*** These faults appear once and disappear without recurring.
- ***Intermittent Faults:*** These faults appear, disappear, and reappear unpredictably.
- ***Permanent Faults:*** These persist until the faulty component is repaired or replaced.

Based on the scope of this work, faults can be categorized into the following types:

- ***Physical Faults (Hardware Faults):*** These include faults related to physical components, such as CPU failures, memory malfunctions, storage issues, or power outages.
- ***Network Faults (Link Faults):*** In cloud computing, resources are accessed via networks, making them vulnerable to network-related issues, such as packet loss or link failures.
- ***Processor Faults (Node Faults):*** These faults occur due to software bugs, resource shortages, or inefficient processing of computing tasks.
- ***Service Expiry Faults:*** These faults occur when an application continues to run after its allocated service time has expired.
- ***Timing Faults:*** Timing faults occur when an application fails to complete its tasks within the specified time frame.

Understanding these fault types is critical in developing robust fault-tolerance mechanisms in cloud computing systems to ensure system reliability and minimize service disruptions.

A. CLOUD IMPLEMENTATION: CENTRALIZED VERSUS DECENTRALIZED VERSUS DISTRIBUTED SYSTEMS

Computer networks are designed to facilitate resource sharing through communication. In networking, computers can be connected in various configurations, leading to systems categorized as Centralized, Decentralized, and Distributed, as illustrated in Fig. 4. Clouds are implemented using these different architectural approaches. Understanding these systems is essential, as the concepts of fault-tolerance and system resilience can vary based on the scale of implementation.

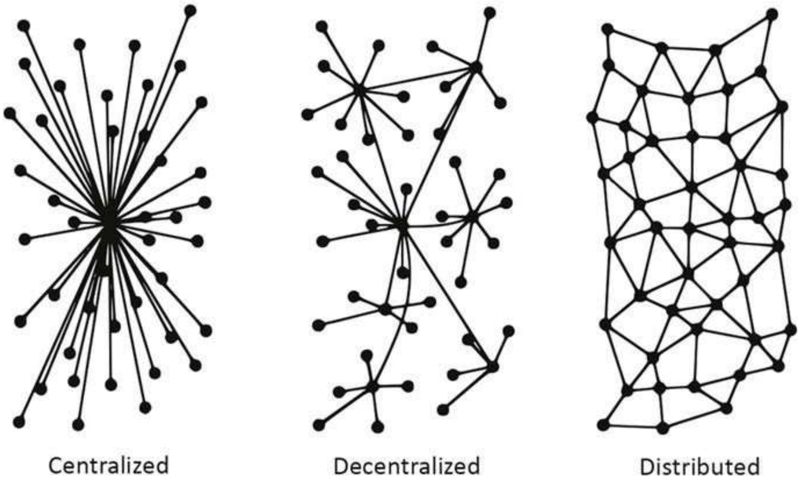

Fig. 3 Centralized vs decentralized vs distributed systems.

Distributed systems can be either centralized or decentralized. To provide easy access and ensure the availability of computing resources, cloud providers offer data storage across multiple geographical locations, although access control may be centralized. These concepts are crucial in the cloud computing environment, and the principles of centralized, decentralized, and distributed systems apply to emerging cloud technologies, including Block chain. Public block chains, such as Bitcoin and Ethereum, are both distributed (with independent nodes located in various places) and decentralized (where stored data cannot be altered). Below is a brief discussion of centralized, decentralized, and distributed systems.

1. **Centralized System**

Centralized systems, often referred to as command-and-control systems, rely on a single central command node where all decisions are made. Other connected nodes must strictly adhere to the commands issued by this central node. Centralized systems are relatively easy to maintain, but they pose a risk of instability due to a single point of failure. While the development of such systems is straightforward because of centralized control, they typically lack extreme scalability.

2. **Decentralized System**

Decentralized systems, sometimes called partially distributed systems, do not rely on a single central command node. Instead, decisions are made independently by multiple parties and sub-nodes, allowing for collaboration to achieve system-wide objectives. Maintenance of decentralized systems is moderate.

3. **Distributed System**

A distributed system is defined as "a collection of autonomous computing elements that appears to its users as a single coherent system." In this architecture, all connected nodes participate in decision-making to achieve system-wide goals. Although distributed systems are complex and challenging to maintain, they offer high scalability and can manage large-scale complexities effectively.

B. NEEDS AND APPLICATIONS FOR FAULT-TOLERANCE COMPUTING

Fault tolerance is a critical aspect of designing any communication system. However, the design requirements vary across different environments and depend on the specific needs of the operational context. For instance, mission-critical computation systems, which are essential for high-stakes applications, require sophisticated and expensive fault-tolerance measures to ensure accurate operation. This is particularly important in settings such as aircraft systems, medical equipment, e-commerce financial applications, and space shuttle communications. In these critical environments, even minor malfunctions can lead to catastrophic outcomes. To mitigate such risks, these systems are designed to maintain a very low or negligible probability of failure. As a result, they are highly reliable, equipped with advanced fault-tolerance features, and come at a significant cost .

Another example involves systems with high availability and reliability requirements, often specified as "Five 9's" reliability. This means that a system should not be unavailable for more than five minutes and 15 seconds per year. Carrier-grade networks typically adhere to this stringent reliability standard. Research on fault-tolerant cloud computing systems encompasses a broad spectrum of applications, ranging from general-purpose computing to highly available systems in sectors like space, transportation, and military operations. Below, we outline several applications and briefly discuss their unique fault-tolerant design considerations and challenges:

- Durable and Heavy-Duty Systems
- Highly Complex Computer Systems
- Overlay Computer Network Systems

1) DURABLE AND HEAVY-DUTY SYSTEMS

Durable and heavy-duty systems are engineered to function effectively in harsh environments, withstanding electromagnetic interference and external noise. These systems often comprise both electrical and mechanical components. Given their design for long-term operation, repairs can be challenging, necessitating a fully redundant system that includes all components for continuous operation. Eventually, these systems are replaced rather than repaired.

2) HIGHLY COMPLEX COMPUTER SYSTEMS Another example includes highly

Highly complex systems consist of billions of interconnected devices, each with a probability of failure. The sheer number of devices can lead to a significant overall system failure probability. To counteract this, various hardware configurations and software redundancy and replication techniques are employed to minimize the likelihood of failures. Distributed systems in computer networks often rely on these complex configurations.

3) OVERLAY COMPUTER NETWORKS SYSTEMS

These systems utilize existing hardware infrastructure to create a virtualized environment through technologies such as Software-Defined Networking (SDN), Network Functions Virtualization (NFV), and cloud computing, often associated with Fifth-Generation (5G) networks. Unlike traditional static distributed systems, overlay networks are highly dynamic. To support these evolving technologies, a comprehensive set of fault-tolerance mechanisms is essential, ensuring high availability and reliability.

C. TYPES OF FAULT-TOLERANCE

There are two primary types of techniques employed in designing a fault-tolerant system: hardware fault-tolerance and software fault-tolerance [21].

1) Hardware Fault-Tolerance

Hardware fault-tolerance involves the design and implementation of hardware components to ensure they can perform tasks and interact correctly with one another within a system. In a computing system, these components include the CPU, memory, hard drives, and other hardware-based devices. Hardware fault-tolerance techniques are essential for developing a structured computing environment that not only tolerates faults but can also initiate self-recovery processes. Typically, the system recovery process involves dividing the computing system into modules, each equipped with protective redundancy. This design provides a degree of automatic recovery in the event of a failure [23].

2) Software Fault-Tolerance

The advent of virtual networks (software-defined networks) and the softwarization of telecommunications systems have significantly advanced software fault-tolerance. This area has become increasingly important for researchers. Like hardware fault-tolerance, software fault-tolerance is a continuous development process aimed at creating software capable of withstanding faults, specifically programming errors. Both active and passive redundancy techniques are utilized in designing fault-tolerant software. In cloud computing, the need for fault-tolerance may vary depending on real-time scenarios. However, it is crucial that the data plane, control plane, and specialized hardware providing abstraction are fault-tolerant, equipped with built-in self-recovery and self-healing mechanisms to ensure quality of service and scalability. This highlights the necessity of both software and hardware-based fault-tolerance. While hardware fault-tolerance is vital for establishing a reliable infrastructure, it is typically straightforward but can also be costly. Conversely, software fault-tolerance is often regarded as more critical in real-time environments; without it, a service may deviate from its intended function and fail to achieve cost reductions.

Broadly, the two main approaches to implementing fault tolerance are Recovery and Redundancy [44]:

- **Recovery**: This approach restores the system state to a predefined checkpoint, effectively rolling back to a stable condition.
- **Redundancy**: This involves replicating hardware, software, and computing components by adding extra components to the system for backup purposes.

Both Recovery and Redundancy approaches can influence reactive and proactive policies in fault-tolerance strategies.

IV. EXISTING CLOUD COMPUTING FAULT-TOLERANCE ARCHITECTURAL FRAMEWORKS

In this section, we present the existing fault-tolerance architectures developed for cloud computing. The classification of these architectures, based on fault-tolerance policies, is illustrated in Fig. 6. In cloud computing, fault-tolerance architectures are generally built around proactive and reactive policies, often combining multiple fault-tolerance approaches to ensure error detection and recovery, which are essential for end-to-end service delivery. For example, architectures such as MapReduce and FT Cloud follow proactive policies, while HAProxy and BFT Cloud are based on reactive policies. Previous surveys primarily focused on reactive fault-tolerance architectures and only briefly covered proactive approaches. In this section, we provide a detailed discussion of both proactive and reactive architectures.

The simplified taxonomy of fault-tolerance architectures/frameworks is shown in Fig. 4. Additionally, we compare these architectures in relation to fault-tolerance policies and the various aspects of fault tolerance they cover[23]. A brief comparison is provided, based on the state-of-the-art research efforts, along with the type of fault-tolerance support offered by these architectures.

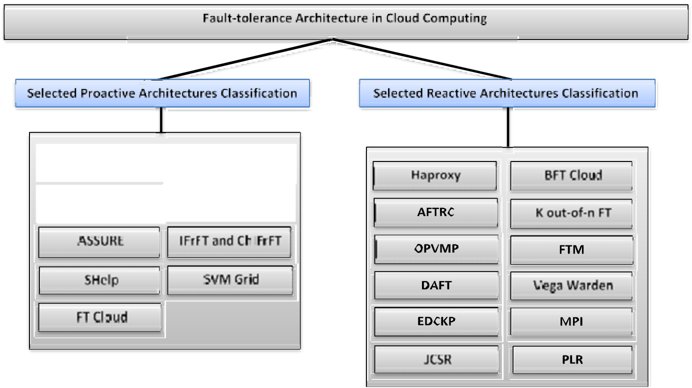
Fault-tolerance Architecture in Cloud Computing
Selected Proactive Architectures Classification
Selected Reactive Architectures Classification
ASSURE
IFrFT and ChFrFT
SHelp
SVM Grid
FT Cloud
Haproxy
BFT Cloud
AFTRC
K out-of-n FT
OPVMP
FTM
DAFT
Vega Warden
EDCKP
MPI
JCSR
PLR

Fig. 4 Fault-tolerance architectures in cloud computing.

V. FUTURE RESEARCH DIRECTION FOR FAULT-TOLERANT CLOUD COMPUTING

Cloud computing models and frameworks have attracted significant attention from both the academic research community and industry [10], [25]. In this section, we outline the future directions for fault-tolerance cloud computing development from the perspective of its role in advancing machine learning tools, energy-aware infrastructure, and finally enabling fault-tolerance in emerging architectures based on cloud related technologies.

A. ADVANCING CLOUD FAULT-TOLERANCE USING MACHINE LEARNING TOOLS

Cloud technology has been proposed to meet the diverse demands of future networks. To provide shared resources and infrastructure, cloud computing models and frameworks are implemented within data centers. In these data centers, virtualization techniques are used to transform physical machines into virtual machines (VMs)[76], improving resource allocation and utilization. However, cloud service providers face significant challenges, such as task scheduling, resource allocation, and managing virtual machine workloads. VMs in data centers are deployed using cloud infrastructures that consist of multiple high-capacity servers.

Task scheduling involves the allocation of VMs to form a service chain that minimizes both the overall runtime of service execution and energy consumption [27]. This process is complex and must be conducted with precision to avoid performance degradation or disruption to the cloud's operations.

Resource allocation in the cloud requires efficient algorithms to identify where VMs should be placed within high-capacity servers in the data center. This enables the live migration of VMs between different locations to optimize resource utilization. This flexible placement of VMs supports load balancing and improves traffic flow management. However, existing solutions to these problems still consume excessive power and time and often lack the necessary quality and accuracy.

Advanced machine learning and artificial intelligence tools[80]must be developed to improve resource allocation, reduce power consumption, and address VM workload management challenges in the cloud. The integration of machine learning and AI algorithms with fault-tolerant methods is expected to significantly enhance cloud performance [21].

B. ENERGY-EFFICIENT FAULT-TOLERANT SCHEDULING ALGORITHMS

In recent years, cloud usage has seen substantial growth, with a projected increase in cloud infrastructure usage that will lead to higher energy consumption [22]. Currently, the ICT sector faces global challenges regarding energy efficiency and carbon (CO2) emissions [23]–[25]. Research has focused on creating energy-efficient, cloud-based infrastructure to promote a greener, more eco-friendly networking environment [26]. Despite these efforts, energy efficiency remains a critical issue, and further development of an energy-efficient, energy-aware ICT infrastructure (cloud-based ecosystem) is necessary. This infrastructure should utilize energy-efficient fault-tolerant scheduling algorithms[27].

The rise of IoT technology has extended the use of ICT infrastructure [89], with new IoT-enabled applications in areas such as smart home control [90], industrial automation [91], and healthcare [92] growing rapidly. To support IoT applications, energy management and fault-tolerance within the ICT infrastructure are essential. Energy-efficient and optimal fault-tolerant scheduling algorithms [93], [94] are key requirements for cloud infrastructure to meet the goals of a green, energy-efficient networking environment [28].

C. HYBRID CLOUD FAULT-TOLERANCE

As hybrid cloud architectures become more prevalent, fault-tolerance must effectively bridge public and private cloud environments. Future research should focus on ensuring seamless fault-tolerance across these diverse infrastructures, including cross-cloud failover mechanisms, multi-cloud redundancy, and the interoperability of fault-tolerant protocols between different cloud providers.

D. SECURITY-ENHANCED FAULT-TOLERANCE

With increasingly sophisticated cybersecurity threats, integrating fault-tolerance with security is a crucial area of research. Future systems must handle not only system failures but also malicious attacks, such as Byzantine faults, ransomware, and denial-of-service (DoS) attacks. Research should focus on fault-tolerant architectures that improve data integrity, confidentiality, and resilience against cyber threats.

E. REAL-TIME AND MISSION-CRITICAL APPLICATIONS

As cloud computing expands to include real-time and mission-critical applications (e.g., autonomous vehicles, healthcare, and financial transactions), low-latency fault-tolerance solutions are vital. Future research should aim to develop ultra-reliable low-latency mechanisms that minimize service interruptions, ensure high fault tolerance, and provide fast recovery times for these sensitive systems.

F. QUANTUM COMPUTING AND FAULT-TOLERANCE

As quantum computing emerges, fault-tolerance must adapt to this new domain, which presents challenges such as quantum decoherence and high error rates. Research should focus on designing quantum fault-tolerant architectures that address these unique error models, ensuring that cloud services utilizing quantum hardware are reliable and resilient.

G. PROACTIVE FAULT-TOLERANCE TECHNIQUES

Proactive fault-tolerance remains under-researched compared to reactive methods. Future research should focus on predictive maintenance, where faults are anticipated and addressed preemptively through machine learning, automated diagnostics, and self-optimizing systems. These approaches can minimize downtime and enhance system reliability without relying solely on post-failure recovery.

H. SCALABILITY AND AUTOMATION

As cloud infrastructures grow, fault-tolerance mechanisms must be highly scalable and automated. Research should explore how to design systems capable of autonomously scaling fault-tolerance techniques in response to increased workloads, minimizing human intervention while maintaining system reliability and performance.

I. COST-EFFICIENT FAULT-TOLERANCE

At scale, fault tolerance can be resource-intensive and costly. Future research should focus on developing cost-efficient fault-tolerance mechanisms that reduce the overhead associated with redundancy and recovery processes. This includes optimizing resource usage, minimizing downtime, and implementing just-in-time fault-tolerant services that are activated only when necessary to improve efficiency and reduce costs.

VII. CONCLUSION

In this survey, we provide an overview of the key features, advantages, components, business/service deployment models, and types of cloud computing. We also explore the implementation of cloud computing using distributed, centralized, and decentralized architectures to offer a comprehensive understanding of cloud-related concepts. Additionally, we examine fault-tolerance in cloud computing by analyzing fault categories, methods, tools, and applied fault-tolerance frameworks.The fault-tolerance frameworks studied address specific types of faults, and no single framework can handle all fault types. Currently, service providers select fault-tolerance mechanisms based on end-user requirements. However, the development of a unified fault-tolerance framework, capable of addressing most fault types, could be possible. While such a framework would be easier to manage, it presents significant challenges due to its design complexity.

In conclusion, this paper highlights the key features and advantages of cloud computing and provides an in-depth background on the subject. We also present an overview of fault-tolerance techniques in cloud computing and discuss the support offered by existing fault-tolerance architectures and frameworks. Furthermore, we outline important future research directions for fault-tolerance in cloud computing from various perspectives.

REFERENCES:

[1] The NIST Definition of Cloud Computing, Standard SP 800-145, National Institute of Science and Technology, 2011.

[2] A. Keshavarzi, A. T. Haghighat, and M. Bohlouli, ‘‘Research challenges and prospective business impacts of cloud computing: A survey,’’ in Proc. IEEE 7th Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS), vol. 2, Sep. 2013, pp. 731–736.

[3] M. G. Avram, ‘‘Advantages and challenges of adopting cloud computing from an enterprise perspective,’’ Proc. Technol., vol. 12, pp. 529–534, Jan. 2014.

[4] A. Apostu, F. Puican, G. Ularu, G. Suciu, and G. Todoran, ‘‘The advantages of telemetry applications in the cloud,’’ in Recent Advances in Applied Computer Science and Digital Services, vol. 2103, H. Fujita and M. Tuba, Eds. 2013, pp. 118–123. [Online]. Available: https://www.wseas.org/main/books/2013/Morioka/DSAC.pdf

[5] A. Gajbhiye and K. M. P. Shrivastva, ‘‘Cloud computing: Need, enabling technology, architecture, advantages and challenges,’’ in Proc. 5th Int. Conf. Confluence Next Gener. Inf. Technol. Summit (Confluence), Sep. 2014, pp. 1–7.

[6] V. Rajaraman, ‘‘Cloud computing,’’ Resonance, vol. 19, no. 3, pp. 242–258, 2014.

[7] P. R. Kumar, P. H. Raj, and P. Jelciana, ‘‘Exploring data security issues and solutions in cloud computing,’’ Proc. Comput. Sci., vol. 125, pp. 691–697, Jan. 2018.

[8] N. Subramanian and A. Jeyaraj, ‘‘Recent security challenges in cloud computing,’’ Comput. Electr. Eng., vol. 71, pp. 28–42, Oct. 2018.

[9] R. Buyya et al., ‘‘A manifesto for future generation cloud computing: Research directions for the next decade,’’ ACM Comput. Surv., vol. 51, no. 5, pp. 1–38, Nov. 2018.

[10] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, ‘‘Cloud computing: Opportunities and challenges,’’ IEEE Potentials, vol. 33, no. 1, pp. 34–36, Jan./Feb. 2014.

[11] F. Durao, J. F. S. Carvalho, A. Fonseka, and V. C. Garcia, ‘‘A systematic review on cloud computing,’’ J. Supercomput., vol. 68, no. 3, pp. 1321–1346, Jun. 2014.

[12] R. Lin, Y. Cheng, M. D. Andrade, L. Wosinska, and J. Chen, ‘‘Disaggregated data centers: Challenges and trade-offs,’’ IEEE Commun. Mag., vol. 58, no. 2, pp. 20–26, Feb. 2020.

[13] C. Kachris and I. Tomkos, ‘‘A survey on optical interconnects for data centers,’’ IEEE Commun. Surveys Tuts., vol. 14, no. 4, pp. 1021–1036, 4th Quart., 2012.

[14] H. Qi, M. Shiraz, J.-Y. Liu, A. Gani, Z. A. Rahman, and T. A. Altameem, ‘‘Data center network architecture in cloud computing: Review, taxonomy, and open research issues,’’ J. Zhejiang Univ. Sci. C, vol. 15, no. 9, pp. 776–793, Sep. 2014.

[15] M. Rajendra Prasad, R Lakshman Naik and V. Bapuji, "Cloud Computing: Research Issues and Implications", *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol. 2, no. 2, pp. 134-140, 2013.

[16] H. P. Zima and A. Nikora, Fault Tolerance. Boston, MA, USA: Springer, 2011, pp. 645–658.

[17] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, ‘‘Basic concepts and taxonomy of dependable and secure computing,’’ IEEE Trans. Depend. Sec. Comput., vol. 1, no. 1, pp. 11–33, Jan. 2004.

[18] M. V. Steen and A. S. Tanenbaum, Distributed Systems, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2017.

[19] S. Hukerikar and C. Engelmann, ‘‘Resilience design patterns—A structured approach to resilience at extreme scale (version 1.0),’’ 2016, arXiv:1611.02717.

[20] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, ‘‘Fault-tolerance in the scope of software-defined networking (SDN),’’ IEEE Access, vol. 7, pp. 124474–124490, 2019.

[21] A. Ganesh, M. Sandhya, and S. Shankar, ‘‘A study on fault tolerance methods in cloud computing,’’ in Proc. IEEE Int. Adv. Comput. Conf. (IACC), Feb. 2014, pp. 844–849.

[22] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes, and J. N. de Souza, ‘‘Elasticity in cloud computing: A survey,’’ Ann. Telecommun.-Annales des Télécommun., vol. 70, no. 7, pp. 289–309, 2015.

[23] M. N. Cheraghlou, A. Khadem-Zadeh, and M. Haghparast, ‘‘A survey of fault tolerance architecture in cloud computing,’’ J. Netw. Comput. Appl., vol. 61, pp. 81–92, Feb. 2016.

[24] M. Hasan and M. S. Goraya, ‘‘Fault tolerance in cloud computing environment: A systematic survey,’’ Comput. Ind., vol. 99, pp. 156–172, Aug. 2018.

[25] P. Kumari and P. Kaur, ‘‘A survey of fault tolerance in cloud computing,’’ J. King Saud Univ., Comput. Inf. Sci., vol. 33, no. 10, pp. 1159–1176, 2018.

[26] Q. Zhang, L. Cheng, and R. Boutaba, ‘‘Cloud computing: State-of-theart and research challenges,’’ J. Internet Services Appl., vol. 1, no. 1, pp. 7–18, May 2010.

[27] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ‘‘A view of cloud computing,’’ Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[28] T. Noor, S. Zeadally, A. Alfazic, and Q. Z. Sheng, ‘‘Mobile Cloud computing: Challenges and future research directions,’’ J. Netw. Comput. Appl., vol. 115, pp. 70–85, Aug. 2018.