

Review **Reliablity and Security for Fog Computing Systems**

Egor Shiriaev ¹ [,](https://orcid.org/0000-0002-2359-1291) Tatiana Ermakova 2,[*](https://orcid.org/0000-0003-0864-3302) , Ekaterina Bezuglova ¹ [,](https://orcid.org/0000-0002-7608-0452) Maria A. Lapina [3](https://orcid.org/0000-0001-8117-9142) and Mikhail Babenko [1](https://orcid.org/0000-0001-7066-0061)

- ¹ Faculty of Mathematics and Computer Science Named after Prof. N. I. Chervyakov, Department of Computational Mathematics and Cybernetics, North-Caucasus Federal University, 355017 Stavropol, Russia; eshiriaev@ncfu.ru (E.S.); eksbezuglova@ncfu.ru (E.B.); mgbabenko@ncfu.ru (M.B.)
- ² School of Computing, Communication and Business, Hochschule für Technik und Wirtschaft,
- University of Applied Sciences for Engineering and Economics, 10318 Berlin, Germany 3 Institute of Digital Development, Department of Information Security of Automated Systems,
- North-Caucasus Federal University, 355017 Stavropol, Russia; mlapina@ncfu.ru

***** Correspondence: tatiana.ermakova@htw-berlin.de

Abstract: Fog computing (FC) is a distributed architecture in which computing resources and services are placed on edge devices closer to data sources. This enables more efficient data processing, shorter latency times, and better performance. Fog computing was shown to be a promising solution for addressing the new computing requirements. However, there are still many challenges to overcome to utilize this new computing paradigm, in particular, reliability and security. Following this need, a systematic literature review was conducted to create a list of requirements. As a result, the following four key requirements were formulated: (1) low latency and response times; (2) scalability and resource management; (3) fault tolerance and redundancy; and (4) privacy and security. Low delay and response can be achieved through edge caching, edge real-time analyses and decision making, and mobile edge computing. Scalability and resource management can be enabled by edge federation, virtualization and containerization, and edge resource discovery and orchestration. Fault tolerance and redundancy can be enabled by backup and recovery mechanisms, data replication strategies, and disaster recovery plans, with a residual number system (RNS) being a promising solution. Data security and data privacy are manifested in strong authentication and authorization mechanisms, access control and authorization management, with fully homomorphic encryption (FHE) and the secret sharing system (SSS) being of particular interest.

Keywords: fog computing; distributed computing systems; reliability; fault tolerance; data security

1. Introduction

Distributed computing systems (DCSs) have evolved over several decades, driven by the need for efficient and scalable computing models [\[1\]](#page-15-0). Starting in the 1960s, the issue of distributed computing began to be raised when researchers began to explore the concept of time sharing, allowing multiple users to access the same computer at the same time. This laid the foundation for the development of distributed systems, in which computing resources were shared across multiple nodes.

In the 1970s and 1980s, research efforts focused on the development of local area networks (LANs) and wide area networks (WANs) to connect geographically dispersed computers. This led to the emergence of a client–server architecture, in which a central server handled requests from multiple client devices. DCSs using a client–server architecture have become dominant, enabling resource sharing and centralized data management.

The advent of the Internet in the 1990s led to significant advances in distributed computing. The client–server model has expanded to include web applications, and the World Wide Web (WWW) has become a platform for distributed computing. This era saw the emergence of web services and application programming interfaces (APIs) that enabled interoperability between different systems, making it easier to exchange data and services over the Internet.

Citation: Shiriaev, E.; Ermakova, T.; Bezuglova, E.; Lapina, M.A.; Babenko, M. Reliablity and Security for Fog Computing Systems. *Information* **2024**, *15*, 317. [https://doi.org/10.3390/](https://doi.org/10.3390/info15060317) [info15060317](https://doi.org/10.3390/info15060317)

Academic Editors: Jose de Vasconcelos, Hugo Barbosa and Carla Cordeiro

Received: 26 April 2024 Revised: 18 May 2024 Accepted: 23 May 2024 Published: 29 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license [\(https://](https://creativecommons.org/licenses/by/4.0/) [creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/) $4.0/$).

Cloud computing (CC) as we know it today emerged in the early 2000s and revolutionized the way computing resources are provided, managed, and used [\[2\]](#page-15-1). Cloud technology is based on the idea of providing on-demand access to a pool of computing resources, such as computing power, storage, and software applications delivered over the Internet.

The concept of cloud computing builds on earlier developments in distributed systems, virtualization, and service computing. Virtualization technologies abstract away physical resources, allowing you to create virtual machines (VMs) that can be dynamically provisioned and deallocated as needed.

The key characteristics of cloud computing include on-demand self-service, broad network access, pooling of resources, fast elasticity, and controlled service. Users can access and allocate resources as needed, dynamically scale their usage up or down, and pay for consumed resources on a pay-as-you-go basis [\[3\]](#page-15-2).

While CC has revolutionized the IT landscape, some applications and use cases have required computing power closer to the network edge. Fog computing (FC) [\[4\]](#page-15-3) has emerged as an additional cloud computing paradigm to meet these requirements.

FC extends the cloud to the edge of the network, bringing computing, storage, and networking closer to the data source [\[5\]](#page-15-4). It uses a distributed architecture consisting of edge devices, gateways, and cloud resources, forming a continuum from the cloud to the edge device.

The FC concept recognizes the limitations of traditional cloud computing in scenarios that require low latency and faster response times, real-time data processing, efficient use of bandwidth, and increased data privacy and security. By processing data closer to the edge, FC reduces the need for extensive data transfer to remote cloud servers, resulting in lower latency and faster response times, as well as the efficient use of bandwidth. Fog devices are also power efficient.

These features make the use of FC attractive in a variety of applications, including IoT $[6]$, smart cities (SCs) $[7]$, automation $[8,9]$ $[8,9]$, healthcare $[10-12]$ $[10-12]$, and agriculture $[13,14]$ $[13,14]$. In applications such as IoT, SC, and healthcare, a continuous and uninterrupted service is of crucial importance. In the healthcare sector, for example, the failure of a medical monitoring system can have life-threatening consequences. Likewise, in smart cities, the malfunction of traffic management systems could lead to serious disruptions. We call this reliability, which is defined as the continuity of the correct service, e.g., regarding the absence of bugs and the masking of errors [\[15\]](#page-15-13). Indeed, shortcomings related to reliability are reported [\[16\]](#page-15-14). Furthermore, in healthcare and IoT applications, the protection of personal data is of paramount importance. Here, we refer to the term of security, which covers the ability of the system to keep data confidential, which, in turn, means protecting data from theft, copying, and disclosure of the owner's identity, as well as protection from unauthorized actions [\[17\]](#page-15-15). In summary, it can be said that security and reliability are of fundamental importance for the functionality and trustworthiness, and hence wider adoption, of FC systems used in various and critical applications.

The aim of this review was to identify and analyze the requirements for improving the security and reliability of FC, as well as the supporting methods and technologies. For this purpose, we applied the method of a systematic literature research. The literature search was carried out in databases, such as Core, Google Scholar, and Semantic Scholar. The search results were checked for their relevance to the topic, and security and reliability requirements for FC were derived.

The paper is structured as follows: Section [2](#page-2-0) discusses the state of the problem. Section [3](#page-2-1) presents the background of FC and its application. Section [4](#page-3-0) presents the method used, while Section [5](#page-9-0) describes the derivation of the requirements. Section [6](#page-13-0) provides a discussion of the results and possibilities for future work. Finally, Section [7](#page-13-1) summarizes the main conclusions and opportunities for future research.

2. State of the Problem

In work [\[16\]](#page-15-14), the authors mentioned the disadvantages related to reliability. However, in order to indicate the relevance of the problem under consideration, we considered the following works. In work [\[18\]](#page-15-16), the authors considered the application of FC in SCs. The main focus of the work was on considering approaches to the integration of FC as the main SC network. The work focused on comparing FC with other types of DCSs, such as CC and edge computing. The work is organized as a review of other research on SC, IoT, and FC topics. The review is general in nature, with a slight refinement for the platform on which FC is deployed. The parameters that were considered in this study were mentioned less often, and the authors suggested paying attention to these parameters, and the use of platform tools is proposed as a solution. A similar trend was also observed in [\[7\]](#page-15-6). Here, we considered the integration of smart solutions, their interconnection, and the organization of the infrastructure. The main focus was on device configuration. The issue of fault tolerance was raised from the point of view of the dynamic configuration of device management. However, the authors used standardized data transmission technologies to measure their frequencies, battery life, and messaging speed. Touching on the topic of security, the authors operated with ciphers with a high level of security, but these ciphers were computationally complex. In conclusion, the authors reflected on the need for additional research.

However, the main focus of the authors of work [\[19\]](#page-15-17) was on the physical model and the construction of a monitoring system, and most of the research was on issues related to the physical embodiment of the system. The research conducted in this study was based on the levels of the OSI model, citing studies at a particular level. In this case, the authors considered a wider range of parameters, including addressing issues of reliability and safety. For example, from the point of view of data protection, the authors characterized FC as difficult to measure. From the point of view of reliability, the authors did not give clear formulations.

Thus, based on several reviews of FC topics, it can be argued that the interest in FC is significant, but at the moment, researchers are focused on building the system as a whole, how to connect devices, which data transfer protocol is more efficient, etc. Although these questions indirectly relate to the subject of this work, a clear answer to the question of "how to organize reliable and secure FC" is still not there. As already mentioned in the previous section, the main purpose of this work was to develop requirements for the reliability and safety of FC. The results will allow other researchers to refer to these requirements to conduct their own FC-related research. Requirements can be in the form of boundary values that will allow you to select the studied methods and algorithms for FC development.

3. Fog Computing and Applications

FC is essentially a distributed computing paradigm that extends cloud capabilities to the edge of the network (Figure [1\)](#page-3-1). While CCs have revolutionized the way computing resources are provided and used, some applications and use cases require low latency, real-time data processing, efficient use of bandwidth, and enhanced privacy and security. FC brings computing, storage, and networking closer to data sources for local processing and analysis [\[5\]](#page-15-4).

In work [\[20\]](#page-15-18), the researchers considered the prospects of using FC in real-time applications. Indeed, 8 years later, FC is widely used in smart city networks and the Internet of things. In [\[21\]](#page-15-19), the authors argued that the use of FC will increase the efficiency of smart city networks based on the Internet of things and proposed a multi-level FC architecture. In [\[22\]](#page-15-20), the authors also presented the design of a platform with several FC applications.

In [\[4\]](#page-15-3), the authors conducted research on the applicability of FC for medical purposes, namely, diabetes tracking. The authors compared FC with CC. In their study, they claimed that in terms of diabetes control, FC is more effective than CC in terms of speed, control, and lower network building costs.

FC finds use in a variety of applications, including the following:

- IoT: FC has a critical role in IoT deployment [\[6\]](#page-15-5). With the proliferation of connected devices generating massive amounts of data, FC brings local processing and analytics to the edge, reducing latency and enabling real-time decision making. This facilitates the efficient filtering and aggregation of data, optimizing the use of network bandwidth and reducing the load on cloud servers.
- Smart cities (SCs): FC plays an important role in the implementation of SC initiatives [\[7\]](#page-15-6). When fog nodes (FNs) are deployed throughout the city infrastructure, data from various sources, such as sensors, cameras, and connected devices, can be processed locally. This enables real-time monitoring, analysis, and decision making for applications such as traffic management, waste management, and public security.
- Automation: In industrial environments, FC enables real-time processing and analysis of data for mission-critical applications [\[8](#page-15-7)[,9\]](#page-15-8). Edge devices and gateways collect and process data from industrial sensors and equipment, providing local control and monitoring. This reduces latency, provides faster response times, and improves efficiency in industries such as manufacturing, energy, and logistics.
- Healthcare: FC plays a vital role in healthcare systems by facilitating real-time monitoring, analysis, and decision making. Edge devices and gateways can collect and process patient data, enabling timely medical intervention, remote patient monitoring, and personalized healthcare services [\[10](#page-15-9)[–12\]](#page-15-10). FC also addresses privacy issues by storing sensitive data locally, ensuring compliance with health regulations.
- Agriculture: In the agricultural sector, FC promotes precision farming and smart farming. Border devices and sensors collect data on soil, weather, and crop conditions to enable local decisions for irrigation, fertilization, and pest control. FC enables realtime analysis and monitoring to optimize resource usage and increase yields [\[13](#page-15-11)[,14\]](#page-15-12).

Thus, we can say that FC and FSs are a promising directions aimed at automating many areas of human activity. FC and FSs are closely related to the IoT and SC as their components.

4. Method

This study was based on a literature search through databases of scientific publications. Statistics on the number of publications from 2008 (the first mention of FC) to 2024 are listed here, analyzed according to keywords related to reliability and safety (see Figure [2](#page-4-0)

and Table [1](#page-4-1) in the attachment). These include fault tolerance, reliability of functioning, data privacy, data security, and robustness.

Database	Fault Tolerance	Reliability of Functioning	Data Privacy	Data Security	Robustness
Google Scholar	16,200	16,800	12,700	14,700	14,700
Core	1363	828	792	6014	1740
Semantic Scholar	3050	86	3920	4800	4490

Table 1. Scientific papers in keyword databases.

Google Scholar provided the highest number of scientific papers across all keywords, especially in the areas of fault tolerance and functional security. Core and Semantic Scholar also provided a large number of articles, albeit to a lesser extent than Google Scholar (see Figure [2\)](#page-4-0).

Figure 2. Keyword publication statistics.

As for Google Scholar, the reliability of functioning was the most frequently represented topic with 16,800 contributions, closely followed by fault tolerance with a total of 16,200 contributions. Data security and robustness shared third place with 14,700 publications each. Data privacy was represented by 12,700 publications. According to Core, data security was the most researched topic with 6014 publications. Robustness was in second place with 1740 articles, closely followed by fault tolerance with 1363 publications. The reliability of functioning and data privacy were at the bottom of the rankings with 828 and 792 publications, respectively. In Semantic Scholar, data security was the most researched topic with 4800 papers, closely followed by robustness with 4490 papers, data privacy with 3920 papers, and fault tolerance with 3050 papers. The reliability of functioning, with a count of only 86, completed the ranking (see Figure [3\)](#page-5-0).

The rankings appeared to be inconsistent. The reasons for this, e.g., that the literature databases possibly only contained subsets of the existing publications or had different search mechanisms, can be further analyzed in future work and recommendations derived.

We built our research on the papers with the most citations and downloads (see Tables [2](#page-5-1) and [3](#page-7-0) below for reliability-related publications and security-related publications analyzed in the manuscript, respectively). They formed the basis for determining and analyzing the requirements for improving the security and reliability of FC.

Figure 3. Diagram of the popularity of research topics.

Table 2. Reliability-related publications.

Table 2. *Cont.*

Table 3. Security-related publications.

4.1. Reliability of FC

Reliability can be defined as the continuity of correct service, e.g., in terms of the absence of bugs and the masking of errors [\[15\]](#page-15-13). This property can thus refer to both physical characteristics of the network and the software.

In [\[23\]](#page-15-21), the authors considered the issue of FC reliability based on the reliability of the Internet of things (IoT). Both the IoT and FC assume a system consisting of computing devices with low power consumption.

In [\[24\]](#page-15-22), the authors carried out a more detailed reliability analysis and divided it into three parts: node reliability, network reliability, and software reliability. The authors proposed methods such as software-defined networking (SDN) and network function virtualization (NFV) as methods to ensure reliability. These methods should improve the reliability of the network if they go through the necessary preliminary stages (network modeling, etc.). To avoid data loss, the authors proposed the use of an adaptive joint protocol based on implicit recognition (AJIA), which is a common reliable and energyefficient mechanism for packet loss recovery and route quality assessment. In addition, the authors pointed out the need to increase fault tolerance by applying error correction codes.

In [\[25\]](#page-15-23), the authors present results related to analyzing and predicting the fault tolerance of FC based on continuous Markov chains. Next, the authors present an intelligent FC optimization method, which they call simulated annealing (ISA). The presented method makes it possible to predict which node is most likely to fail in order to take appropriate action.

All the work considered so far assumed the use of the speaker as part of the IoT system. In [\[26\]](#page-15-24), the authors studied FC in the context of a drone control system and considered the reliability of the system in this domain. In general, the guarantee of reliability is conditioned by the solution of the optimization problem. In this case, the loss of computing nodes (drones) has a higher probability, and thus, the optimization task becomes more complicated. The authors proposed two algorithms: LP-based and proximal Jacobi. For building a foggy drone swarm system, the authors recommended the proximal Jacobi algorithm due to its ratio of problem-solving quality to computational complexity.

The authors of [\[27\]](#page-16-0) also dealt with the development of FC reliability requirements. In their work, the authors analyzed the dependence of FC reliability depending on the number of computing nodes and workload. The result of the work was calculations that allow us to determine how appropriate it is to use FC on the system under study with specified parameters (the number of nodes and the load on them). The expediency here is the predicted reliability of FC. The authors' contribution to the development of requirements lies in the formulation of which FC can be considered reliable in the theoretical model, namely, such an FC that allows for processing the expected computational load on all nodes without failure.

Ref. [\[28\]](#page-16-1) took a practical approach and analyzed the reliability of FC depending on the type of system built. In particular, the authors analyzed information systems, such as food chains, healthcare and medical services, mobile-facility-based information systems, smart cities, and UFV monitoring and control. The study consisted of investigating the reliability of the nodes measured at specific points in time. It was carried out with different computational loads: low, medium, and maximum. The authors' research results are useful from the point of view of the practical implementation of FC, namely, for the design of an FC network architecture depending on the expected computational load and the degree and method of distribution of tasks among the computing nodes.

Based on the sources analyzed, the biggest "threat" to FC reliability is the high computational load on the FC nodes. Planning the distribution of the computing load can generally be divided into three requirements:

- Ensuring low latency and response times of the computing nodes;
- Scalability and resource management of a pool of computing nodes;
- Fault tolerance for each individual computing node, as well as general fault tolerance, which can be ensured, among other things, by redundancy of the processed data.

4.2. Security of FC

Security in this study included the ability of the system to keep data confidential, which, in turn, means that data are protected against theft, copying, disclosure of the owner's identity, and unauthorized actions [\[17\]](#page-15-15).

FC security is based on DCS and CC security since FC is essentially a development of these models. Confidentiality is also important for data in the system, not only in relation to the external environment of the system but also between neighboring nodes. Since the "owners" of computing nodes can be different, an attacker can also gain control over one or more nodes. In addition, FC is a network of low-power nodes, which is a limiting factor on the computational complexity of security methods. Thus, the issue of FC security is relevant and the subject of research in the scientific community.

In [\[29\]](#page-16-2), the authors provide an analysis of FC security, including CC and edge calculations for the integrity of the study. Given the specifics of the FC system, it can have a wide range of vulnerabilities. Thus, the authors identified 12 categories of security vulnerabilities for FC:

- 1. Advanced persistent threats (APTs) are cyberattacks that aim to compromise a company's infrastructure in order to steal data and intellectual property.
- 2. Access control issues (ACIs) can lead to poor management, and any unauthorized user will be able to obtain data and permissions to install software and change configurations.
- 3. Account hijacking (AH) is when an attack is aimed at hijacking user accounts for malicious purposes. Phishing is a potential method of account hijacking.
- 4. Denial of service (DoS) is when legitimate users are not allowed to use the system (data and applications) due to excessive use of limited system resources.
- 5. A data leak (DB) is when an attacker divulges or steals important, protected, or confidential data.
- 6. Data loss (DL) is the accidental (or malicious) deletion of data from the system. This does not necessarily have to be the result of a cyberattack and may result from a natural disaster.
- 7. Insecure APIs (IAs): Many cloud service providers provide application programming interfaces (APIs) for use by customers. The security of these APIs is crucial to the security of any implemented applications.
- 8. System and application vulnerabilities (SAVs) are vulnerabilities that can be exploited as a result of configuration errors in the ad software, which an attacker can use to infiltrate and compromise the system.
- 9. A malicious insider (MI) is a user who has gained authorized access to the network and system, but intentionally decides to act maliciously.
- 10. Insufficient due diligence (IDD) often occurs when an organization is in a hurry to adopt, design, and implement a system.
- 11. Abuse and unfair use (ANU) often occurs when resources are provided free of charge, and attackers use these resources to carry out malicious activities.
- 12. Problems with shared technologies (STIs) arise from sharing infrastructures, platforms, or applications. For example, the underlying hardware components may not have been designed to provide high-insulation properties.

The authors analyzed how these threats affect various FC-based applications. As a result of their work, the authors cite categories of threats, their types, and possible solutions. In principle, this work can be used as a basis for research in the field of FC security, as well as background information on preparing FC for threat confrontations. However, this work is an overview and does not provide research on the effectiveness of a particular method of protection or methods of attack.

Similar studies are given in the works of [\[30](#page-16-3)[–34\]](#page-16-4). In general, these studies can be characterized as follows. The researchers proceeded from the possible applications of FC and focused on their features. On the one hand, this is the right direction since if we compare FC for IoT and FC for healthcare since these are two completely different areas that require different approaches.

However, the very basis of the system is the same for them and the methods of providing the basic level are identical. This conclusion is justified by the conducted research. After analyzing the threats, you can see that, for example, the DB threat for most applications is the same as the ACI threat. Thus, we can say that the main requirement for data security is their confidentiality. Even if an attacker can get hold of one or another part of the data, the data should not have any significance for the attacker.

5. Requirements

Although FC offers many advantages, ensuring reliability and security is crucial for its successful implementation. Based on the review of the literature, we identified the following requirements for the reliability and security of FC.

5.1. Low Delay and Response

FC aims to reduce latency by processing data closer to the edge. It is very important to define acceptable latency thresholds for different applications and use cases. Requirements may vary depending on the need for real-time decision making, data transfer limitations, and application criticality. The following can be considered to optimize the latency and improve the responsiveness in FC environments:

- Edge caching: By caching frequently accessed data and content on edge devices or an FN closest to the end users, latency can be significantly reduced [\[35\]](#page-16-5). This allows for faster data and content retrieval as you do not have to traverse the entire network to get to the cloud or remote servers [\[36\]](#page-16-6). Edge caching improves the response times for applications such as video streaming, content delivery, and IoT data retrieval.
- Edge analytics: Real-time analytics and decision making at the edge delivers immediate responses without the need to transfer data to a remote server [\[37\]](#page-16-7). By deploying simplified analytics and machine learning models directly to the edge, you can instantly obtain insights and actions. Edge analytics is especially useful for time-critical applications, such as industrial automation, smart cities, and healthcare monitoring.
- Mobile edge computing (MEC): MEC brings the power of fog computing to a mobile network infrastructure, delivering low-latency services to mobile devices [\[38\]](#page-16-8). By deploying edge computing resources at cell base stations or access points, MEC shortens the distance between mobile devices and computing resources [\[39](#page-16-9)[,40\]](#page-16-10). This proximity facilitates real-time applications, such as augmented reality (AR) and virtual reality (VR), where responsiveness is critical. This is useful, for example, in various rehabilitation centers [\[11,](#page-15-25)[41\]](#page-16-11).

By using these techniques to achieve low latency and a quick response, fog computing environments can perform real-time processing and analysis, reducing the latency and improving the overall system responsiveness. These techniques play an important role in supporting time-critical applications, improving the user experience, and providing a wide range of latency-sensitive services in fog computing deployments.

5.2. Scalability and Resource Management

Fog systems (FSs) must scale easily to meet growing workloads and expanding device connectivity. Determining the requirements for mechanisms for dynamic resource allocation, load balancing and elastic scaling is necessary to ensure the efficient use of resources and system performance. FC systems require robust monitoring and control capabilities to ensure reliability and security. We defined the requirements for real-time monitoring, performance metrics, anomaly detection, and centralized management tools to enable proactive system maintenance, early problem detection, and efficient resource allocation.

- Load balancing: Load-balancing methods evenly distribute computing tasks and network traffic among multiple fog nodes to avoid overloading certain nodes and ensure the optimal use of resources [\[42\]](#page-16-12). Load-balancing algorithms consider factors such as the processing capability, network conditions, and node availability to intelligently distribute workloads. This approach avoids bottlenecks, reduces the response time, and improves the system scalability. There are many types of load balancing based on different technologies [\[43](#page-16-13)[–52\]](#page-16-14).
- Edge federation: Edge federation enables collaboration and resource sharing between multiple fog nodes or edge networks [\[53\]](#page-17-0). By forming federated networks, fog computing environments can leverage pooled resources and provide seamless scalability across multiple administrative domains. Federation platforms establish communication protocols, trust mechanisms, and resource sharing agreements to enable dynamic resource allocation and load balancing between federated fog nodes [\[54\]](#page-17-1).
- Virtualization and containerization: Virtualization technologies, such as hypervisors and VMs, abstract and isolate computing resources, improving the scalability and resource management [\[55\]](#page-17-2). Fog nodes can run multiple virtual machines, each with its own sandboxed environment, making efficient use of hardware resources. Containerization using technologies such as Docker [\[56\]](#page-17-3) provides lightweight and portable environments for applications, enabling efficient deployment, scalability, and resource isolation in fog computing.
- Edge resource discovery and orchestration: Fog computing systems can implement resource discovery mechanisms to determine the available resources in the fog infras-

tructure [\[57\]](#page-17-4). These mechanisms facilitate dynamic resource provisioning by allowing fog nodes to efficiently discover and use nearby resources. Orchestration environments coordinate resource provisioning and management, ensuring that workloads are distributed to the appropriate fog nodes based on availability, capability, and prox-imity [\[58](#page-17-5)[,59\]](#page-17-6). There are several effective solutions for orchestration. One of the most interesting is OASIS TOSCA [\[60\]](#page-17-7), which allows you to effectively manage different containers in distributed systems based on templates, which allows you to combine different approaches. For example, the use of various methods of load balancing.

By leveraging these scaling and resource management techniques, fog computing environments can efficiently handle changing workloads, optimize resource utilization, and ensure the efficient allocation of computing resources. These techniques improve the system scalability, adaptability, and performance, allowing fog computing systems to meet the requirements of various applications and effectively support the dynamic nature of edge computing environments.

5.3. Fault Tolerance and Redundancy

Given the distributed nature of FC, addressing the issues of fault tolerance and redundancy is critical. Defining requirements for error detection and recovery mechanisms and redundancy will help to ensure system reliability and resilience in the face of component failures or network outages. Defining system resiliency and disaster recovery requirements is critical to mitigating potential failures or disasters. Defining backup and recovery mechanisms, data replication strategies, and disaster recovery plans will help to ensure the availability and integrity of data and system functionality.

To meet this requirement, one of the promising areas is the residue number system (RNS), which is a non-positional number system based on modular arithmetic in the residue ring [\[61\]](#page-17-8). The RNS has various properties to improve reliability and fault tolerance; in addition, due to natural parallelism and the independence of residuals, the RNS is effectively implemented in distributed systems. The RNS also has self-correcting properties [\[62–](#page-17-9)[65\]](#page-17-10), which improves the system resiliency. The RNS has many applications, both for reliability and fault tolerance and for security, which is discussed later.

5.4. Data Privacy

FC involves processing sensitive data at the edge, which raises concerns about data privacy and security. Determining the requirements for data encryption, secure communication protocols, access control mechanisms, and compliance with data protection rules is essential to protecting sensitive information. Also, from a privacy perspective, an FS needs strong authentication and authorization mechanisms to ensure that only authorized users and devices can access and interact with the system. Defining requirements for secure user authentication, access control, and privilege management is vital to preventing unauthorized access and protecting system resources.

Maintaining security involves high computing effort and managing the security keys is a more difficult task for CC and FC in terms of data infrastructure compared with centralized systems. Often, centralized security is used for CC. To authenticate the user, the end device communicates with the key exchange server; however, this solution is also ineffective.

One of the promising areas in this field is the so-called fully homomorphic encryption (FHE) [\[66\]](#page-17-11). The main feature of FHE is the ability to process data in encrypted form. Many researchers believe that FHE will improve the effectiveness of security and privacy in cloud applications. However, if we consider FC and consider that FNs are low-power devices, FHE is currently not appropriate for FC due to the high computational complexity of some operations [\[66\]](#page-17-11). FHE in FC can be examined in Figure [4.](#page-12-0) However, the current computational complexity is not an unbreakable wall. Many researchers are devoting their work to improving the efficiency of FHE in various applications, such as in works [\[67–](#page-17-12)[70\]](#page-17-13), where researchers developed an approximate FHE scheme for rational numbers. In addition, the authors of the papers proposed various modifications of FHE to speed up its operation, for example, by using an RNS [\[71–](#page-17-14)[73\]](#page-17-15).

Figure 4. Application of FHE in FC.

Currently, the secret-sharing system (SSS) is the most effective method to fulfill the requirements for security and data protection in FC (Figure [5\)](#page-12-1). The SSS is a security method based on the division of any information into several parts with the possibility of its recovery by a specific group of participants. The SSS is used in conjunction with other security methods to ensure the confidentiality of encryption keys. The SSS is often used in networks with an increased risk of security threats. There are several types of SSS. For example, there are methods that require the capture of all parts of the key (the full SSS) or only a certain proportion (the threshold SSS). Following their emergence, the threshold SSS has almost completely replaced the full SSS.

Figure 5. Application of SSS in FC.

Threshold circuits are also of interest for FC. This makes it possible to exchange data together with security keys without the entire system being involved, but only a specific part. This security also fulfills the requirement for FC reliability, as it reduces the load on the network. It is also worth noting that there are SSSs that use an RNS [\[74,](#page-18-0)[75\]](#page-18-1), which, in turn, allows for a combination of approaches to fulfill both the security and reliability requirements of FC [\[76–](#page-18-2)[78\]](#page-18-3).

6. Discussion and Future Directions

As a result, the following four key requirements were formulated: (1) low latency and response times; (2) scalability and resource management; (3) fault tolerance and redundancy; and (4) data protection and security. In addition, some methods and technologies are presented that enable FC systems to fulfill these requirements.

Low delay and response can be achieved through edge caching, i.e., caching of frequently accessed data and content on edge devices or the FN closest to the end users [\[35\]](#page-16-5), edge real-time analyses and decision making at the edge [\[37\]](#page-16-7), and mobile edge computing [\[38](#page-16-8)[–40\]](#page-16-10).

Scalability and resource management to avoid overloading certain nodes and ensure optimal resource utilization [\[42\]](#page-16-12) can be enabled by edge federation, i.e., collaboration and sharing of resources between multiple fog nodes or edge networks [\[53\]](#page-17-0), virtualization and containerization [\[55\]](#page-17-2), edge resource discovery and orchestration [\[57\]](#page-17-4) based on availability, and capabilities of and proximity to the appropriate fog nodes [\[58,](#page-17-5)[59\]](#page-17-6).

Fault tolerance and redundancy can be enabled by ensuring the availability and integrity of data and system functions with backup and recovery mechanisms, data replication strategies and disaster recovery plans, among other things, with a residual number system (RNS) [\[61\]](#page-17-8) being a promising solution.

Data security and data privacy are manifested in strong authentication and authorization mechanisms, access control and authorization management, etc., which are ensured by access control mechanisms, secure communication protocols, data encryption, etc., with fully homomorphic encryption (FHE) [\[66\]](#page-17-11) and a secret sharing system (SSS) being of particular interest.

These requirements can be taken into account when designing FC architectures. Furthermore, on this basis, investigations can be carried out into the extent to which the reliability and security of FC systems and other system properties can be impaired, e.g., to what extent overloads, underloads, and threats can occur. Certain templates can be created based on such data. If more specific templates are available, the developer only needs to specify certain parameters in perspective, e.g., the security level or the load on certain nodes. This will enable the more efficient development and application of FC-based networks and systems.

The focus of future research may include the development of FC architectures with the aim of developing best practices for the use of FC architectures and optimal FC model architectures with respect to the criteria discussed. Simulation tools and mathematical models can be used to evaluate the architectures [\[79\]](#page-18-4). For instance, the reliability property can be described by the probability that a system or system component will function successfully up to a certain period of time and can be modeled stochastically, e.g., mean time to failure and probability that a component will fail before reaching time T [\[15\]](#page-15-13). Simulators can be developed when mathematical modeling is difficult due to the size, complexity, and heterogeneity of the system [\[79\]](#page-18-4). After developing and evaluating these simulation and mathematical models, it is essential to test physical models in the next step in order to validate the results.

Thus, various aspects of FC system security can be studied with a focus on different types of attacks and unauthorized access, as well as the reliability and fault tolerance of FC systems with a focus on the traffic between nodes, queue utilization, and evaluation of device failures. Studies can also focus on the expected computing load with the degree and type of distribution of tasks among the computing nodes [\[28\]](#page-16-1) and the dependence of FC reliability on the number of computing nodes and the workload, while the expected computing load on all nodes without failure [\[27\]](#page-16-0) is calculated.

7. Conclusions

In this review, research was conducted on FC reliability and security. This research consisted of an analytical review of the FC literature in order to develop requirements for FC reliability and security. Due to many fundamental differences from CC, not all CC techniques can be directly transferred to FC. This analysis of the scientific literature on FC showed that the scientific interest in this paradigm is increasing due to the growing interest in smart solutions and local automation. This is due to the fact that FC is an attractive solution for various IoT- and SC-related applications.

Different research groups in the field of reliability have mainly focused on specific FC applications or key system parameters and used them to evaluate the reliability of the system. This study identified key FC parameters that can characterize FC reliability. These parameters have been reformulated to meet the requirements. The result of this part of the study can be considered a list of the formulated requirements themselves, as well as a proposed set of methods and solutions to build an FC that satisfies the requirements.

In the field of FC security, the situation was the opposite; the researchers tried to consider FC security from the point of view of individual threats and possible countermeasures for individual components. Some of the threats analyzed in this paper fall under the category of reliability and fault tolerance. On this basis, a single requirement, "data confidentiality", was defined in the field of security. Since in the case of FC, data confidentiality is guaranteed, it will be difficult for an attacker to successfully attack the system to steal important data. This paper also presents methods to ensure the required level of confidentiality.

In the future, practical research is planned, namely, the construction of FC architectures based on the results of the conducted research in order to work out FC system templates. Also, there is planned experimental study of the developed architectures on the subject of conformity to the developed requirements of reliability and security, interchangeability of architectures and templates, and methods of reliability and security.

Author Contributions: Conceptualization, M.B. and T.E.; methodology, E.S. and E.B.; validation, E.S. and M.A.L.; formal analysis, T.E. and E.B.; investigation, M.B.; resources, T.E.; writing—original draft preparation, E.S.; writing—review and editing, T.E.; visualization, E.S. and E.B.; supervision, T.E.; project administration, M.B.; funding acquisition, M.B. All authors read and agreed to the published version of this manuscript.

Funding: This research was supported by the Russian Science Foundation Grant No. 22-71-10046, 414 [\(https://rscf.ru/en/project/22-71-10046/](https://rscf.ru/en/project/22-71-10046/) (accessed on 24 April 2024)).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

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

Abbreviations

The following abbreviations are used in this manuscript:

- AG Augmented reality
- DCS Distributed computing system
- CC Cloud computing
- FC Fog computing
- FHE Fully homomorphic encryption
- FN Fog node
- FS Fog system
- IaaS Infrastructure as a service
- IoT Internet of things
- MEC Mobile edge computing
- PaaS Platform as a service
- RNS Residue number system SaaS Software as a service
- SC Smart city
- SSS Secret sharing system

VR Virtual reality

References

- 1. Singh, K.; Alam, M.; Kumar, S. A Survey of Static Scheduling Algorithm for Distributed Computing System. *Int. J. Comput. Appl.* **2015**, *129*, 25–30. [\[CrossRef\]](http://doi.org/10.5120/ijca2015906828)
- 2. Kratzke, N. A Brief History of Cloud Application Architectures. *Appl. Sci.* **2018**, *8*, 1368. [\[CrossRef\]](http://dx.doi.org/10.3390/app8081368)
- 3. Mell, P.; Grance, T. The NIST Definition of Cloud Computing. National Institute of Standards and Technology. 2011. Available online: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf> (accessed on 22 April 2024).
- 4. Klonoff, D.C. Fog Computing and Edge Computing Architectures for Processing Data From Diabetes Devices Connected to the Medical Internet of Things. *J. Diabetes Sci. Technol.* **2017**, *11*, 647–652. [\[CrossRef\]](http://dx.doi.org/10.1177/1932296817717007)
- 5. Abouaomar, A.; Cherkaoui, S.; Kobbane, A.; Dambri, O.A. A Resources Representation for Resource Allocation in Fog Computing Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
- 6. Sarkar, S.; Chatterjee, S.; Misra, S. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. *IEEE Trans. Cloud Comput.* **2015**, *6*, 46–59. [\[CrossRef\]](http://dx.doi.org/10.1109/TCC.2015.2485206)
- 7. Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog Computing for Sustainable Smart Cities: A Survey. *ACM Comput. Surv.* **2017**, *50*, 32:1–32:43. [\[CrossRef\]](http://dx.doi.org/10.1145/3057266)
- 8. Pop, P.; Raagaard, M.A.L.; Gutierrez, M.; Steiner, W. Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN). *IEEE Commun. Stand. Mag.* **2018**, *2*, 55–61. [\[CrossRef\]](http://dx.doi.org/10.1109/MCOMSTD.2018.1700057)
- 9. Rani, S.; Kataria, A.; Chauhan, M. Fog Computing in Industry 4.0: Applications and Challenges—A Research Roadmap. In *Energy Conservation Solutions for Fog-Edge Computing Paradigms*; Tiwari, R., Mittal, M., Goyal, L.M., Eds.; Springer: Singapore, 2022; pp. 173–190, ISBN 9789811634482.
- 10. Kraemer, F.A.; Braten, A.E.; Tamkittikhun, N.; Palma, D. Fog Computing in Healthcare—A Review and Discussion. *IEEE Access* **2017**, *5*, 9206–9222. [\[CrossRef\]](http://dx.doi.org/10.1109/ACCESS.2017.2704100)
- 11. Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N. Fog Computing for Healthcare 4.0 Environment: Opportunities and Challenges. *Comput. Electr. Eng.* **2018**, *72*, 1–13. [\[CrossRef\]](http://dx.doi.org/10.1016/j.compeleceng.2018.08.015)
- 12. Shi, Y.; Ding, G.; Wang, H.; Roman, H.E.; Lu, S. The Fog Computing Service for Healthcare. In Proceedings of the 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), Beijing, China, 28–30 May 2015; pp. 1–5.
- 13. Guardo, E.; Stefano, A.D.; Corte, A.L.; Sapienza, M.; Scatà, M. A Fog Computing-Based IoT Framework for Precision Agriculture. *J. Internet Technol.* **2018**, *19*, 1401–1411.
- 14. Hsu, T.-C.; Yang, H.; Chung, Y.-C.; Hsu, C.-H. A Creative IoT Agriculture Platform for Cloud Fog Computing. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100285. [\[CrossRef\]](http://dx.doi.org/10.1016/j.suscom.2018.10.006)
- 15. Berger, C.; Eichhammer, P.; Reiser, H.P.; Domaschka, J.; Hauck, F.J.; Habiger, G. A Survey on Resilience in the IoT: Taxonomy, Classification, and Discussion of Resilience Mechanisms. *ACM Comput. Surv.* **2022**, *54*, 1–39. [\[CrossRef\]](http://dx.doi.org/10.1145/3462513)
- 16. Yi, S.; Li, C.; Li, Q. A Survey of Fog Computing: Concepts, Applications and Issues. In Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, 21 June 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 37–42.
- 17. Soomro, Z.A.; Shah, M.H.; Ahmed, J. Information Security Management Needs More Holistic Approach: A Literature Review. *Int. J. Inf. Manag.* **2016**, *36*, 215–225. [\[CrossRef\]](http://dx.doi.org/10.1016/j.ijinfomgt.2015.11.009)
- 18. Songhorabadi, M.; Rahimi, M.; MoghadamFarid, A.; Kashani, M.H. Fog Computing Approaches in IoT-Enabled Smart Cities. *J. Netw. Comput. Appl.* **2023**, *211*, 103557. [\[CrossRef\]](http://dx.doi.org/10.1016/j.jnca.2022.103557)
- 19. Sabireen, H.; Neelanarayanan, V. A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express* **2021**, *7*, 162–176.
- 20. Peter, N. Fog computing and its real time applications. *Int. J. Emerg. Technol. Adv. Eng.* **2015**, *5* , 266–269.
- 21. Zhang, C. Design and Application of Fog Computing and Internet of Things Service Platform for Smart City. *Future Gener. Comput. Syst.* **2020**, *112*, 630–640. [\[CrossRef\]](http://dx.doi.org/10.1016/j.future.2020.06.016)
- 22. Yi, S.; Hao, Z.; Qin, Z.; Li, Q. Fog Computing: Platform and Applications. In Proceedings of the 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, USA, 12–13 November 2015; pp. 73–78.
- 23. Madsen, H.; Burtschy, B.; Albeanu, G.; Popentiu-Vladicescu, F.L. Reliability in the Utility Computing Era: Towards Reliable Fog Computing. In Proceedings of the 2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP), Bucharest, Romania, 7–9 July 2013; pp. 43–46.
- 24. Popentiu-Vladicescu, F.; Albeanu, G. Software Reliability in the Fog Computing. In Proceedings of the 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), Karachi, Pakistan, 5–7 April 2017; pp. 1–4.
- 25. Zhang, P.; Chen, Y.; Zhou, M.; Xu, G.; Huang, W.; Al-Turki, Y.; Abusorrah, A. A Fault-Tolerant Model for Performance Optimization of a Fog Computing System. *IEEE Internet Things J.* **2022**, *9*, 1725–1736. [\[CrossRef\]](http://dx.doi.org/10.1109/JIOT.2021.3088417)
- 26. Hou, X.; Ren, Z.; Wang, J.; Zheng, S.; Cheng, W.; Zhang, H. Distributed Fog Computing for Latency and Reliability Guaranteed Swarm of Drones. *IEEE Access* **2020**, *8*, 7117–7130. [\[CrossRef\]](http://dx.doi.org/10.1109/ACCESS.2020.2964073)
- 27. Melnik, E.V.; Klimenko, A.B. A Condition of Reliability Improvement of the System Based on the Fog-Computing Concept. *J. Phys. Conf. Ser.* **2020**, *1661*, 012007. [\[CrossRef\]](http://dx.doi.org/10.1088/1742-6596/1661/1/012007)
- 28. Klimenko, A.B.; Melnik, E.V. An Experimental Study of the Fog-Computing-Based Systems Reliability. In *Artificial Intelligence and Bioinspired Computational Methods*; Silhavy, R., Ed.; Springer International Publishing: Cham, Switzerland, 2020; pp. 438–449.
- 29. Khan, S.; Parkinson, S.; Qin, Y. Fog Computing Security: A Review of Current Applications and Security Solutions. *J. Cloud Comp.* **2017**, *6*, 19. [\[CrossRef\]](http://dx.doi.org/10.1186/s13677-017-0090-3)
- 30. Stojmenovic, I.; Wen, S.; Huang, X.; Luan, H. An Overview of Fog Computing and Its Security Issues. *Concurr. Comput. Pract. Exper.* **2016**, *28*, 2991–3005. [\[CrossRef\]](http://dx.doi.org/10.1002/cpe.3485)
- 31. Stojmenovic, I.; Wen, S. The Fog Computing Paradigm: Scenarios and Security Issues. In Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 September 2014; pp. 1–8.
- 32. Yi, S.; Qin, Z.; Li, Q. Security and Privacy Issues of Fog Computing: A Survey. In *Wireless Algorithms, Systems, and Applications*; Xu, K., Zhu, H., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 685–695.
- 33. Zhang, P.; Zhou, M.; Fortino, G. Security and Trust Issues in Fog Computing: A Survey. *Future Gener. Comput. Syst.* **2018**, *88*, 16–27. [\[CrossRef\]](http://dx.doi.org/10.1016/j.future.2018.05.008)
- 34. Thota, C.; Sundarasekar, R.; Manogaran, G.; Varatharajan, R.; Priyan, M.K. Centralized Fog Computing Security Platform for IoT and Cloud in Healthcare System. In *Fog Computing: Breakthroughs in Research and Practice*; IGI Global: Pennsylvania, PA, USA, 2018; pp. 365–378, ISBN 978-1-5225-5649-7.
- 35. Zahmatkesh, H.; Al-Turjman, F. Fog Computing for Sustainable Smart Cities in the IoT Era: Caching Techniques and Enabling Technologies—An Overview. *Sustain. Cities Soc.* **2020**, *59*, 102139. [\[CrossRef\]](http://dx.doi.org/10.1016/j.scs.2020.102139)
- 36. Li, Q.; Zhang, Y.; Li, Y.; Xiao, Y.; Ge, X. Capacity-Aware Edge Caching in Fog Computing Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9244–9248. [\[CrossRef\]](http://dx.doi.org/10.1109/TVT.2020.3001301)
- 37. Anawar, M.R.; Wang, S.; Azam Zia, M.; Jadoon, A.K.; Akram, U.; Raza, S. Fog Computing: An Overview of Big IoT Data Analytics. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 7157192. [\[CrossRef\]](http://dx.doi.org/10.1155/2018/7157192)
- 38. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 2322–2358. [\[CrossRef\]](http://dx.doi.org/10.1109/COMST.2017.2745201)
- 39. Ren, J.; Zhang, D.; He, S.; Zhang, Y.; Li, T. A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet. *ACM Comput. Surv.* **2019**, *52*, 125:1–125:36. [\[CrossRef\]](http://dx.doi.org/10.1145/3362031)
- 40. Dolui, K.; Datta, S.K. Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing. In Proceedings of the 2017 Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017; pp. 1–6.
- 41. Paul, A.; Pinjari, H.; Hong, W.-H.; Seo, H.C.; Rho, S. Fog Computing-Based IoT for Health Monitoring System. *J. Sens.* **2018**, *2018*, e1386470. [\[CrossRef\]](http://dx.doi.org/10.1155/2018/1386470)
- 42. Jafarnejad Ghomi, E.; Masoud Rahmani, A.; Nasih Qader, N. Load-Balancing Algorithms in Cloud Computing: A Survey. *J. Netw. Comput. Appl.* **2017**, *88*, 50–71. [\[CrossRef\]](http://dx.doi.org/10.1016/j.jnca.2017.04.007)
- 43. Aote, S.S.; Kharat, M.U. A Game-Theoretic Model for Dynamic Load Balancing in Distributed Systems. In Proceedings of the International Conference on Advances in Computing, Communication and Control, Bangalore, India, 28–29 December 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 235–238.
- 44. Baihaqi, M.R.; Negara, R.M.; Tulloh, R. Analysis of Load Balancing Performance Using Round Robin and IP Hash Algorithm on P4. In Proceedings of the 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 8 December 2022; pp. 93–98.
- 45. Mishra, S.K.; Sahoo, B.; Parida, P.P. Load Balancing in Cloud Computing: A Big Picture. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 149–158. [\[CrossRef\]](http://dx.doi.org/10.1016/j.jksuci.2018.01.003)
- 46. Kumar, P.; Kumar, R. Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. *ACM Comput. Surv.* **2019**, *51*, 120:1–120:35. [\[CrossRef\]](http://dx.doi.org/10.1145/3281010)
- 47. Jader, O.; Zeebaree, S.; Zebari, R. A State Of Art Survey For Web Server Performance Measurement And Load Balancing Mechanisms. *Int. J. Sci. Technol. Res.* **2019**, *8*, 535–543.
- 48. Chandak, A.; Ray, N.K. A Review of Load Balancing in Fog Computing. In Proceedings of the 2019 International Conference on Information Technology (ICIT), Shanghai, China, 20–23 December 2019; pp. 460–465.
- 49. Devaraj, A.F.S.; Elhoseny, M.; Dhanasekaran, S.; Lydia, E.L.; Shankar, K. Hybridization of Firefly and Improved Multi-Objective Particle Swarm Optimization Algorithm for Energy Efficient Load Balancing in Cloud Computing Environments. *J. Parallel Distrib. Comput.* **2020**, *142*, 36–45. [\[CrossRef\]](http://dx.doi.org/10.1016/j.jpdc.2020.03.022)
- 50. Devi, D.C.; Uthariaraj, V.R. Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks. *Sci. World J.* **2016**, *2016*, e3896065. [\[CrossRef\]](http://dx.doi.org/10.1155/2016/3896065)
- 51. Wan, J.; Chen, B.; Wang, S.; Xia, M.; Li, D.; Liu, C. Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4548–4556. [\[CrossRef\]](http://dx.doi.org/10.1109/TII.2018.2818932)
- 52. Ningning, S.; Chao, G.; Xingshuo, A.; Qiang, Z. Fog Computing Dynamic Load Balancing Mechanism Based on Graph Repartitioning. *China Commun.* **2016**, *13*, 156–164. [\[CrossRef\]](http://dx.doi.org/10.1109/CC.2016.7445510)
- 53. Savi, M.; Santoro, D.; Di Meo, K.; Pizzolli, D.; Pincheira, M.; Giaffreda, R.; Cretti, S.; Kum, S.; Siracusa, D. A Blockchain-Based Brokerage Platform for Fog Computing Resource Federation. In Proceedings of the 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 24–27 February 2020; pp. 147–149.
- 54. Veillon, V.; Denninnart, C.; Salehi, M.A. F-FDN: Federation of Fog Computing Systems for Low Latency Video Streaming. In Proceedings of the 2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC), Larnaca, Cyprus, 14–17 May 2019; pp. 1–9.
- 55. Sri Raghavendra, M.; Chawla, P. A Review on Container-Based Lightweight Virtualization for Fog Computing. In Proceedings of the 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 29–31 August 2018; pp. 378–384.
- 56. Bellavista, P.; Zanni, A. Feasibility of Fog Computing Deployment Based on Docker Containerization over RaspberryPi. In Proceedings of the 18th International Conference on Distributed Computing and Networking, Hyderabad, India, 5–7 January 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 1–10.
- 57. Hoque, S.; De Brito, M.S.; Willner, A.; Keil, O.; Magedanz, T. Towards Container Orchestration in Fog Computing Infrastructures. In Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 4–8 July 2017; Volume 2, pp. 294–299.
- 58. Santoro, D.; Zozin, D.; Pizzolli, D.; De Pellegrini, F.; Cretti, S. Foggy: A Platform for Workload Orchestration in a Fog Computing Environment. In Proceedings of the 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, 11–14 December 2017; pp. 231–234.
- 59. Jiang, Y.; Huang, Z.; Tsang, D.H.K. Challenges and Solutions in Fog Computing Orchestration. *IEEE Netw.* **2018**, *32*, 122–129. [\[CrossRef\]](http://dx.doi.org/10.1109/MNET.2017.1700271)
- 60. OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC. Topology and Orchestration Specification for Cloud Applications Version 1.0. 2013. Available online: <http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html> (accessed on 22 April 2024).
- 61. Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. Application of the Residue Number System to Reduce Hardware Costs of the Convolutional Neural Network Implementation. *Math. Comput. Simul.* **2020**, *177*, 232–243. [\[CrossRef\]](http://dx.doi.org/10.1016/j.matcom.2020.04.031)
- 62. Babenko, M.; Tchernykh, A.; Pulido-Gaytan, B.; Cortés-Mendoza, J.M.; Shiryaev, E.; Golimblevskaia, E.; Avetisyan, A.; Nesmachnow, S. RRNS Base Extension Error-Correcting Code for Performance Optimization of Scalable Reliable Distributed Cloud Data Storage. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 17–21 June 2021; pp. 548–553.
- 63. Tay, T.F.; Chang, C.-H. A New Algorithm for Single Residue Digit Error Correction in Redundant Residue Number System. In Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 1–5 June 2014; pp. 1748–1751. [\[CrossRef\]](http://dx.doi.org/10.1109/ISCAS.2014.6865493)
- 64. Ananda Mohan, P.V. Error Detection, Correction and Fault Tolerance in RNS-Based Designs. In *Residue Number Systems: Theory and Applications*; Mohan, P.V.A., Ed.; Springer International Publishing: Cham, Switzerland, 2016; pp. 163–175, ISBN 978-3-319-41385-3.
- 65. Chang, C.-H.; Molahosseini, A.S.; Zarandi, A.A.E.; Tay, T.F. Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications. *IEEE Circ. Syst. Mag.* **2015**, *15*, 26–44. [\[CrossRef\]](http://dx.doi.org/10.1109/MCAS.2015.2484118) 66. Gentry, C. A Fully Homomorphic Encryption Scheme. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2009.
- 67. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology—ASIACRYPT 2017*; Takagi, T., Peyrin, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 409–437.
- 68. Cheon, J.H.; Kim, D.; Kim, Y.; Song, Y. Ensemble Method for Privacy-Preserving Logistic Regression Based on Homomorphic Encryption. *IEEE Access* **2018**, *6*, 46938–46948. [\[CrossRef\]](http://dx.doi.org/10.1109/ACCESS.2018.2866697)
- 69. Cheon, J.H.; Kim, D.; Kim, D. Efficient Homomorphic Comparison Methods with Optimal Complexity. In *Advances in Cryptology— ASIACRYPT 2020*; Moriai, S., Wang, H., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 221–256.
- 70. Kim, S.; Lee, K.; Cho, W.; Cheon, J.H.; Rutenbar, R.A. FPGA-Based Accelerators of Fully Pipelined Modular Multipliers for Homomorphic Encryption. In Proceedings of the 2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 9–11 December 2019; pp. 1–8.
- 71. Al Badawi, A.; Polyakov, Y.; Aung, K.M.M.; Veeravalli, B.; Rohloff, K. Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 941–956. [\[CrossRef\]](http://dx.doi.org/10.1109/TETC.2019.2902799)
- 72. Gomathisankaran, M.; Tyagi, A.; Namuduri, K. HORNS: A Homomorphic Encryption Scheme for Cloud Computing Using Residue Number System. In Proceedings of the 2011 45th Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, 23–25 March 2011; pp. 1–5.
- 73. Lee, J.-W.; Lee, E.; Lee, Y.; Kim, Y.-S.; No, J.-S. High-Precision Bootstrapping of RNS-CKKS Homomorphic Encryption Using Optimal Minimax Polynomial Approximation and Inverse Sine Function. In *Advances in Cryptology—EUROCRYPT 2021*; Canteaut, A., Standaert, F.-X., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 618–647.
- 74. Goyal, V.; Kumar, A. Non-Malleable Secret Sharing. In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, Los Angeles, CA, USA, 25–29 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 685–698.
- 75. Applebaum, B.; Beimel, A.; Farràs, O.; Nir, O.; Peter, N. Secret-Sharing Schemes for General and Uniform Access Structures. In *Advances in Cryptology—EUROCRYPT 2019*; Ishai, Y., Rijmen, V., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 441–471.
- 76. Gladkov, A.; Gladkova, N.; Kucherov, N. Analytical Review of Methods for Detection, Localization and Error Correction in the Residue Number System. In *Mathematics and its Applications in New Computer Systems*; Tchernykh, A., Alikhanov, A., Babenko, M., Samoylenko, I., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 507–514.
- 77. Gladkov, A.; Shiriaev, E.; Tchernykh, A.; Deryabin, M.; Bezuglova, E.; Valuev, G.; Babenko, M. SNS-Based Secret Sharing Scheme for Security of Smart City Communication Systems. In *Smart Cities*; Nesmachnow, S., Hernández Callejo, L., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2023; pp. 248–263.
- 78. Boyle, E.; Gilboa, N.; Ishai, Y. Function Secret Sharing: Improvements and Extensions. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1292–1303.
- 79. Bachiega, J.; Costa, B.; Carvalho, L.R.; Rosa, M.J.F.; Araujo, A. Computational Resource Allocation in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv.* **2023**, *55*, 1–31. [\[CrossRef\]](http://dx.doi.org/10.1145/3586181)

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