JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 10, Issue.5, May 2022 E-ISSN: 2347-2693

Fault Tolerance Middleware Cloud Computing In Virtual Infrastructures

Sathishkumar D.[1*], R. Vadivel[2]

[1]PG Student, Department of Information Technology, Bharathiar University, Tamil Nadu India
[2]Assistant Professor, Department of Information Technology, Bharathiar University, Tamil Nadu, India

**Corresponding Author: sathishkumarddharmaraj@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v10i5.712 | Available online at: www.ijcseonline.org

Received: 24/Apr/2022, Accepted: 10/May/2022, Published: 31/May/2022

Abstract— As cloud computing becomes more popular as an attractive alternative to traditional information processing systems, accurate and continuous operation is becoming more important, even in the presence of faulty components. This white paper presents an innovative modular system-level perspective for building and managing cloud fault tolerance. It uses a dedicated service layer to provide application developers and users with a comprehensive, high-level approach to hiding the details of implementing fault-tolerance techniques. In particular, the service layer allows users to specify and apply the required level of fault tolerance without the need for knowledge of the fault-tolerance technologies available in the desired cloud and their implementation. Use run-time monitoring to determine the mechanism and characteristics of your fault-tolerance solution. Based on the proposed approach, design a framework that easily integrates with your existing cloud infrastructure and makes it easier for third parties to provide fault tolerance as a service. Our goal is to work directly at the VM instance level of Virtual Machine Manager and inject the framework as a dedicated service layer between the application and the hardware. Fault Tolerance Manager addresses the issue of computing resource inhomogeneity, achieves the goal of transparently providing fault tolerance support for node failures to user applications, and achieves scalability and interoperability goals. To overcome these challenges, we propose to build a fault tolerance manager according to the principles of service-oriented architecture.

Keywords— Cloud computing, Fault tolerance , Virtual machine, Service-oriented architecture.

I. INTRODUCTION

The growing demand for flexibility in cost-effectively procuring and releasing computing resources has led to widespread adoption of the cloud computing paradigm. The availability of an extensible pool of resources to users provides an effective alternative for deploying applications with high scalability and processing requirements. The traditional way to achieve reliable and highly available software is to use a fault-tolerant method at the time of procurement and development. This means that customers have to apprehend fault tolerance strategies and tailor their programs with the aid of using thinking about surroundings unique parameters for the duration of the layout phase. However, for the programs to be deployed withinside the Cloud computing environment, it's far hard to layout a holistic fault tolerance answer that effectively combines the failure conduct and device structure of the application. This problem arises from the cloud computing abstraction layer, which conveys the high complexity of the system and the limited information about the underlying infrastructure to the user. Unlike traditional approaches, it advocates a new dimension in which applications deployed in cloud computing infrastructure can get the required fault tolerance properties from third parties. To support a new dimension, we propose an approach that extends work and realizes a common fault tolerance mechanism as a separate module. This allows each module to function transparently in your application. Then enhance each module with a set of metadata that characterizes fault tolerance properties and use the metadata to select a mechanism that meets your needs. In addition, it presents a scheme that combines selected fault tolerance mechanisms to provide a comprehensive fault tolerance solution for user applications and uses run-time monitoring to determine the properties of the fault tolerance solution. Based on the proposed approach, design a framework that easily integrates with your existing cloud infrastructure to help third parties provide fault tolerance as a service.

1.1 Cloud Computing

Cloud computing is an internet technology that uses both remote central servers and the internet to manage data and applications. This technology enables many companies and users to use their data and applications without installing them. Users and businesses can access information and files from any computer system connected to the Internet. Cloud computing provides much more effective computing through centralized storage, processing, storage, and bandwidth. Cloud computing is a term used to describe a technology that distributes computer services from local clients. Cloud computing is an internet technology that uses both remote central servers and the internet to manage data and applications. This technology allows many businesses and users to use their data and applications without installing them. Users and businesses can access information and files from any computer system connected to the Internet. Cloud computing provides much more effective computing through centralized storage, processing, storage, and bandwidth. The process of cloud computing technology is divided into three segments. Platforms, applications and infrastructure are the three segments of this technology. Each component performs many functions and provides applications to individuals and businesses around the world

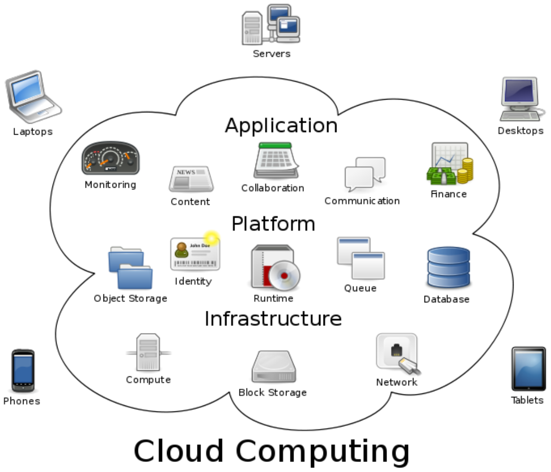


Figure 1 Cloud computing

1.2 Service models

Die Cloud-Computing-service model Software as a Service(SaaS), Platform as a Service(PaaS), Infrastructure as a Service(IaaS). Each component performs many functions and provides applications to individuals and businesses around the world. PaaS provides operating systems, hardware, and networks, and customers install or develop their own software and applications. The IaaS model provides only hardware and networks. Customers install or develop their own operating systems, software, and applications.

1.3 Deployment of cloud services

Cloud services are typically delivered via a private cloud, community cloud, public cloud, or hybrid cloud. Services provided by the public cloud are generally provided over the Internet and are owned and operated by cloud providers. Examples include public services such as online photo storage services, email services, and social networking sites. However, enterprise services can also be provided in the public cloud.

In a private cloud, the cloud infrastructure is dedicated to a particular organization and is managed by the organization or a third party. In the community cloud, services are shared by multiple organizations and are only available to those groups. Infrastructure may be owned and operated by your organization or cloud service provider. Hybrid clouds are a combination of different methods of resource pooling (for example, a combination of public and community clouds).

1.4 Cloud services are popular

Cloud services are popular because they can reduce the cost and complexity of owning and operating computers and networks. Cloud users don't have to invest in IT infrastructure or buy hardware or software licenses, reducing upfront costs, improving return on investment, rapid deployment, customization, flexible use, and new innovations.

In addition, cloud providers that specialize in specific areas (such as email) can offer advanced services that may not be available or developed by a single organization. Other benefits for users are scalability, reliability, and efficiency. Scalability means that cloud computing offers unlimited processing power and storage capacity. The cloud is reliable in that you can access your applications and documents from anywhere in the world over the Internet. Cloud computing is often seen as efficient because it allows enterprises to free up resources and focus on innovation and product development.

Another potential advantage is that personal data can be better protected in the cloud. In particular, cloud computing can embed privacy into technology from the beginning and improve efforts to use better security mechanisms.. Cloud computing enables more flexible IT procurement and expansion, and may be able to adjust steps based on data confidentiality. The proliferation of the cloud can also promote open standards for cloud computing that specify basic data security characteristics common to different services and providers. Cloud computing can also enable a better audit trail. Moreover, the information in the cloud is not easily lost.

II. BACKROUND STUDY

K. V. Vishwanath and N. Nagappan [1], “Characterizing cloud computing hardware reliability,” Modern data centers host hundreds of thousands of servers that coordinate tasks to provide highly available cloud computing services. These servers consist of multiple hard drives, memory modules, network cards, processors, etc., each of which can fail, even if carefully designed. The likelihood of such a failure can be somewhat small over the life of the server (typically 35 years in the industry), but these numbers increase on all devices hosted in the data center. At such large scales, hardware component failures are standard, not exceptions. Hardware failures can lead to poor end-user performance and business loss. A solid understanding of the numbers and root causes behind these failures can help improve the operational experience through engineering that not only increases resilience to failures, but also reduces hardware costs. This directly leads to business savings. As far as we know, this document is the first attempt to identify a server failure and hardware repair in a large data centre. Provides a detailed analysis of failure characteristics and a preliminary analysis of failure predictors. We hope that the results presented in this paper will motivate further research in this area.

R. Jhawar, V. Piuri, and M. D. Santambrogio [2], Fault tolerance, reliability, and restoring of cloud computing are paramount to ensuring continuous operation and correct results, even with a specific maximum number of faulty components. Most existing research and implementations focus on architecture-specific solutions for implementing fault tolerance. This means that users need to customize their application to account for environment-specific fault-tolerant characteristics. These needs lead to opaque and inflexible cloud environments that require a great deal of effort from developers and users. This white paper presents an innovative perspective on creating and managing fault tolerance and hides the details of implementing reliability technology from users through a dedicated service layer. This allows users to specify and apply the required level of fault tolerance without any implementation knowledge.

W. Zhao, P. M. Melliar-Smith, and L. E. Moser [3], The Low Latency Fault Tolerance (LLFT) Middleware uses a reader follower replication approach to provide fault tolerance for distributed applications deployed in cloud computing or data center environments. LLFT middleware consists of a low latency messaging protocol, a Leader Determined Membership Protocol, and a Virtual Determinizer Framework. The messaging protocol provides a reliable, fully ordered message delivery service using direct group-to-group multicast, ordered by the primary copy within the group. Membership logs provide fast reconfiguration and recovery services in the event of a replica failure, and if the replica joins or leaves the group. The Virtual Determinizer Framework captures the order information of the primary replica and applies the same order to the backup replicas of the nondeterministic primary sources. LLFT middleware maintains strong replica consistency, provides application transparency, and keeps end-to-end latency low.

Y. Mao, C. Liu, J. E. van der Merwe, and M. Fernandez [4], Cloud computing provides users with near-instant access to seemingly unlimited resources and provides service providers with the ability to provide their customers with a complex information technology infrastructure as a service. Vendors can benefit from economies of scale and multiplexing that can be gained by sharing resources through virtualization of the underlying physical infrastructure. However, the size and highly dynamic nature of cloud platforms presents significant new challenges for cloud service providers. In particular, delivering advanced cloud services requires a cloud governance framework that can coordinate the provisioning, configuration, consumption, and decommissioning of cloud resources across a distributed set of physical resources. This white paper advocates a data-centric approach to cloud orchestration. With this approach, cloud resources are modeled as structured data that can be queried in a declarative language and updated with well-defined transaction semantics. A data-centric management framework (DMF) as a solution that considers the feasibility, benefits, and challenges of this approach and uses a data model, query language, and semantics specifically designed for cloud resource orchestration.) Design and typical implementation.

C. A. Ardagna, E. Damiani, R. Jhawar, and V. Piuri[8], Presents a reliability authentication system in which services are modeled as discrete-time Markov chains. After verifying the authenticity characteristics, a machine-readable certificate is issued to the service and the validity of the certificate is checked by regular run-time monitoring. In addition, we will introduce a solution that allows users to search and select services using a specific set of reliability properties. Our solution is integrated into an existing service-oriented architecture (SOA) that enables user configuration validation both at discovery and at run time.

III. PROPOSED METHODOLOGY

In the proposed system, the fault tolerance mechanism exists as a separate module, allowing each module to function transparently in the user application. Then enhance each module with a set of metadata that characterizes fault tolerance properties and use the metadata to select a mechanism that meets your needs. In addition, it presents a scheme that combines selected fault tolerance mechanisms to provide a comprehensive fault tolerance solution for user applications and uses run-time monitoring to determine the properties of the fault tolerance solution. Based on the proposed approach, design a framework that easily integrates with your existing cloud infrastructure to help third parties provide fault tolerance as a service. Our goal is to insert the framework as a dedicated service layer between your application and the hardware and work directly on top of Virtual Machine Manager at the VM instance level. Fault Tolerance Manager addresses the issue of computing resource inhomogeneity, achieves the goal of transparently providing fault tolerance support for node failures to user applications, and achieves scalability and interoperability goals. To overcome these challenges, we propose to build a fault tolerance manager according to the principles of service-oriented architecture.

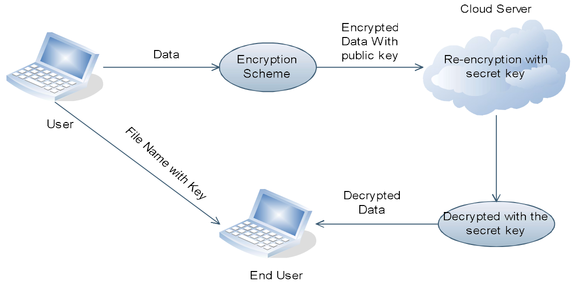
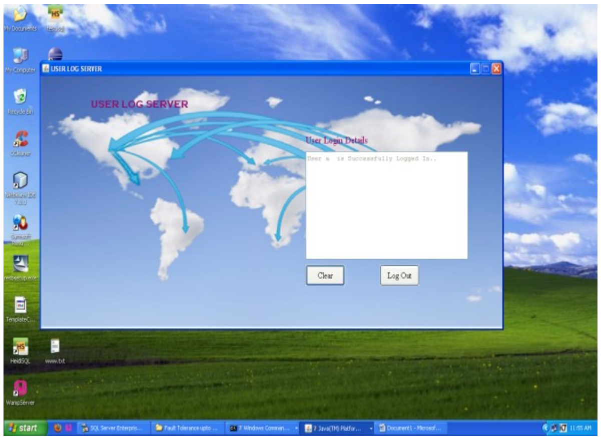

Cloud Server
Encrypted Data With public key
Data
Encryption Scheme
Re-encryption with secret key
User
File Name with Key
Decrypted Data
Decrypted with the secret key
End User

Figure 2 System flow diagram

IV. SYSTEM MODEL

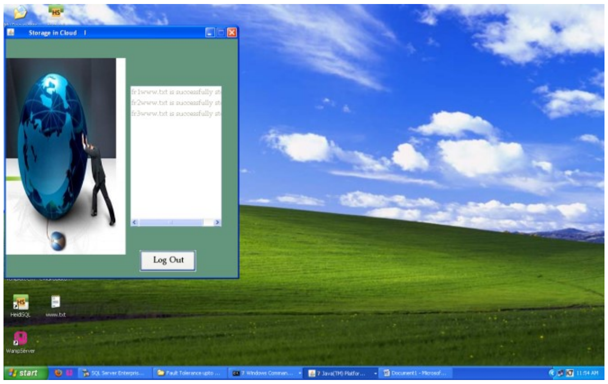
In this paper describes the process of identity verification. Use this process to prove that you are who you say you are. In order for a user to prove their identity, they must provide some form of identity that the system understands and trusts. The authentication process begins by creating an instance of the login context. Various constructors are available. This example uses the login context type. The first parameter is the name (which acts as an index to the login module stack configured in the configuration file) and the second parameter is the call back handler used to pass the login information to the log server. The call back handler has a handle method that forwards the required information to the log server. The call back handler has a handle method that forwards the required information to the log server. This example uses a very simple handler that stores the username and password in an instance variable so that the log server can pass them when it calls the Handle method. You can also create a call back that interacts with the user to get user credentials and sends that information to the log server for authentication.



User a is Successfully Logged In..

4.1 Construction of Cloud Data Storage

The data tier uses IP-provided storage services to store and retrieve customer data, and the application tier uses IP's computing services to handle operations and respond to customer requests. This system architecture allows banking services to meet diverse business needs in terms of computing resource scalability and resilience. However, using traditional methods, the fault-tolerance characteristics of banking services remain constant throughout the life cycle. Therefore, from the customer's point of view, operate the SP, specify reliability and availability requirements based on business needs, and transparently obtain the fault-tolerance characteristics required for the application, as well as the operational status of the network connection. Is easier. Also, the VM instance must be maintained by the resource manager. Database and resource graphs point out that service providers are essential to ensuring the correct operation of the fault tolerance mechanism.



fr1www.txt is successfully st
fr2www.txt is successfully st
fr3www.txt is successfully st

Cloud data storage

4.2 Replica Allocation & Data Encryption

This component supports the replication mechanism by calling replicas and managing their execution based on the needs of the client. A set of VM instances controlled by a single implementation of the replication mechanism is called a replica set. Each replica in the set is uniquely identifiable and specifies the set of rules R that the replica set must meet. The job of the replication manager is to ensure that the client recognizes the replica set as a single service and that a good replica works correctly at run time. To support the replication mechanism, the replication caller must have the required replication parameters, such as replication style (active, passive, cold, passive, hot, passive), number of replicas, and constraints on the relative placement of individual replicas. First consider. That is, the replica caller receives the client application reference as input from the FTM kernel, analyzes the expected fault tolerance properties, and interacts with the resource manager to get the location of each replica. Data uploaded by customers is encrypted for secure data storage in the cloud. The encrypted data is stored on various virtual servers along with the fragments.

4.3 Fault Tolerance

The task of providing fault tolerance as a service requires the service provider to implement a generic fault tolerance mechanism to allow client applications deployed on virtual machine instances to transparently retrieve fault tolerance properties. To this end, we define the ft-unit as a basic module that applies a coherent fault-tolerance mechanism to recurring system failures at the VM instance granularity. The term ft-unit is based on the observation that the impact of hardware failures on client applications can be managed by applying a fault tolerance mechanism directly to the virtualization layer rather than the application itself. For example, banking service fault tolerance can be improved by replicating the entire VM instance where the application layer is deployed on multiple physical nodes, and server crashes use well-known fault detection algorithms such as the Heartbeat Protocol. Can be detected using. The primary and backup components run on VM instances that are independent of the banking services application layer. The design phase begins when a client requests a service provider to provide fault-tolerant support for its application. In this phase, the service provider must first analyze the customer's requirements and match them against the available ft units to form a complete fault-tolerant solution with the appropriate ft units. It turns out that each ft unit provides its own set of fault-tolerance properties that can be characterized by functional, operational, and structural attributes.

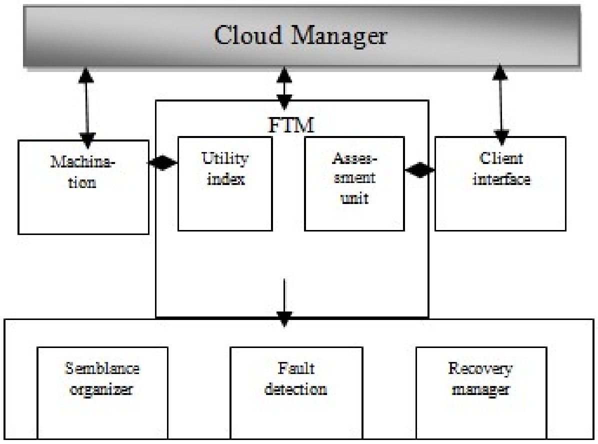


Figure 3.4 Fault tolerance management in cloud computing

4.4 Data Retrieval

The Data Retrieval Module describes how to retrieve data from various virtual servers that authenticate only users. Data is encrypted with different virtual servers using fragments. The data is fetched from various virtual servers, combined and converted into a decrypted format. The decrypted data is converted to the original data for user extraction. The goal of this component is to achieve system-level restoring force by minimizing system downtime in the event of a failure. To this end, this component supports an ft unit that implements a recovery mechanism so that a fault-prone node can be returned to normal operating mode.

4.5 RSA Algorithm

The RSA algorithm is an asymmetric encryption algorithm. Asymmetry actually means that it works with two different keys. H. Public and private keys. As the name implies, the public key is shared with everyone and the private key remains secret.

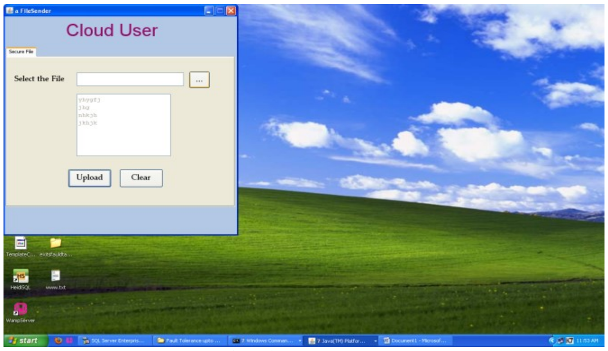
Generate a secure, secret key using a Key Generator.

The server encrypts the data using client's public key and sends the encrypted data.

Write and Read encrypted or decrypted data using Cipher Output Stream and Cipher Input Stream.

V. RESULT AND DISCUSSION

We have presented an approach to transparently provide fault tolerance to applications deployed on virtual machine instances. In particular, it presents an approach that implements a general fault tolerance mechanism as an independent module, validates the fault tolerance properties of each mechanism, matches the user's requirements with the available fault tolerance modules, and provides the desired properties. A comprehensive solution has been realized.

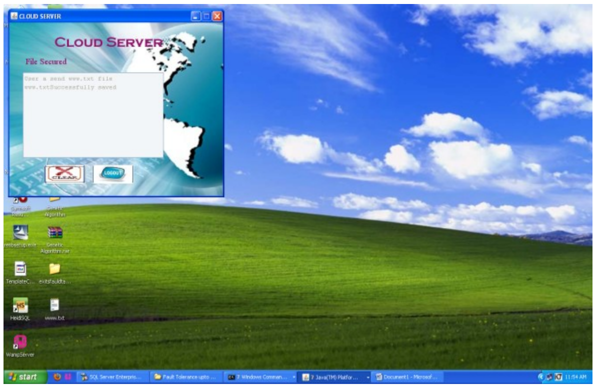


Select the File

```
yhygfj
jhg
nhkjh
jkhjk
```

Cloud user screen

A Cloud User may also comprise multiple individuals or business units. Cloud User means any individual person making use of a CSP's Cloud Services provided to a Cloud Customer, based on a relationship between that Cloud Customer and the Cloud User.



```
User a send www.txt file
www.txtSuccessfully saved
```

Cloud server

A cloud server is a pooled, centralized server resource that is hosted and delivered over a network—typically the Internet—and accessed on demand by multiple users.

By combining the proposed approach with our deployment scheme, service providers can provide long-term fault tolerance support for their customers' applications. In addition, we have developed a framework that allows service providers to integrate their systems with their existing cloud infrastructure. This provides the basis for a general implementation of the approach for providing fault tolerance as a service. You can extend the components of the framework to improve the overall restoring force of your cloud infrastructure. Future work will focus primarily on implementing a framework for measuring the strength of fault tolerance services and transferring cloud data to end users with secure file transfers.

VI. CONCLUSION AND FUTURE WORK

We have presented an approach to transparently provide fault tolerance to applications deployed on virtual machine instances. In particular, it implements the general fault tolerance mechanism as a separate module, validates the fault tolerance properties of each mechanism, matches the user's requirements with the available fault tolerance modules, and is comprehensive with the desired properties. I introduced an approach to getting a solution. By combining the proposed approach and deployment scheme, service providers can provide long-term fault tolerance support for their customers' applications. In addition, we have developed a framework that allows service providers to integrate their systems with their existing cloud infrastructure. This provides the basis for a general implementation of the approach for providing fault tolerance as a service. You can extend the components of the framework to improve the overall restoring force of your cloud infrastructure. Our destiny paintings will especially be pushed closer to the implementation of the framework to degree the electricity of fault tolerance provider and to make an in-intensity evaluation of the fee advantages amongst all of the stakeholders.

REFERENCE

[1] K. V. Vishwanath and N. Nagappan, “Characterizing cloud computing hardware reliability,” in *Proc. 1st ACM Symp. Cloud Comput.*, pp.**193-204**, **2010**.

[2] R. Jhawar, V. Piuri, and M. D. Santambrogio, “A comprehensive conceptual system-level approach to fault tolerance in cloud computing,” in *Proc. IEEE Int. Syst. Conf.*, Mar. **2012**.

[3] W. Zhao, P. M. Melliar-Smith, and L. E. Moser, “Fault tolerance middleware for cloud computing,” in *Proc. 3rd Int. Conf. Cloud Comput.*, pp.**67-74**, Jul. **2010**.

[4] Y. Mao, C. Liu, J. E. van der Merwe, and M. Fernandez, “Cloud resource orchestration: A data-centric approach,” in *Proc. 5th Biennial Conf. Innovative Data Syst. Res.*, **2011**.

[5] G. Koslovski, W.-L. Yeow, C. Westphal, T. T. Huu, J. Montagnat, and P. Vicat-Blanc, “Reliability support in virtual infrastructures,” in *Proc.IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. **2010**.

[6] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G.Pelosi, and P. Samarati, “Encryption-based policy enforcement for cloud storage,” in *Proc. 30th Int. Conf. Distributed Comput. Syst. Workshop*, **2010**.

[7] P. Samarati and S. De Capitani di Vimercati, “Data protection in outsourcing scenarios: Issues and directions,” in *Proc. 5th ACM Symp.Inform. Comput. Commun. Security*, **2010**.

[8] C. A. Ardagna, E. Damiani, R. Jhawar, and V. Piuri, “A model-based approach to reliability certification of services,” in *Proc. 6th IEEE Int. Conf. Digit. Ecosyst. Technol.*, Jun. **2012**.

[9] Yookesh, T. L., et al. "Efficiency of iterative filtering method for solving Volterra fuzzy integral equations with a delay and material investigation." Materials today: Proceedings 47: **6101-6104, 2021.**

[10] Kumar, E. Boopathi, and V. Thiagarasu. "Segmentation using Fuzzy Membership Functions: An Approach." IJCSE, Vol.**5.** Issue**.3,** pp**.101-105, 2017.**

AUTHORS PROFILE

Mr.D.Sathishkumar received Bachelors Degree in Computer Science in the year 2020 from kongunadu Arts and Science Collage, Coimbatore, Tamil Nadu, affiliated to Bharathiar University. He is currently pursuing a Masters Degree in Information Technology from 2020 to 2022, at Bharathiar University, Coimbatore, Tamil Nadu.

Dr. R.Vadivel is an Assistant Professor in the Department of Information Technology, Bharathiar University, Tamil Nadu, India. He received his Ph.D. degree in Computer Science from Monomaniam Sundaranar University in the year 2013. M.E., Degree in Computer Science and Engineering from Annamalai University in the year 2007. B.E., Degree in Computer Science and Engineering from Periyar University in the year 2002. He obtained his Diploma in Electronics and Communication Engineering from State Board of Technical Education in the year 1999. He had published over 88 journals papers and over 45 conferences papers both at National and International level. His areas of interest include Computer Networks, Network Security, Information Security, etc.