2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Evolution of Issues in Distributed Systems: A Systematic Review

Maha M. AlShammari
Computational Unit, Department of Environmental Health
Institute for Research and Medical Consultations, Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
mmashammari@iau.edu.sa

Abstract*—The globalization has increased the demand for complex distributed systems that can be accessed at multiple locations by multiple users. However, the issues pertaining to the distributed systems need specific attention in order to minimize the performance failures and increase the Quality of Service. This study can be a useful source of information to understand the evolution of the issues associated with distributed systems, and the current state of issues, which can help the developers in managing the issues during the development and maintenance. This study reviewed various issues related to distributed systems under 12 themes. We found that Security and Privacy, quality of service, resource management, and synchronization issues are a few major subjects that are being focused on current research studies. This study has concluded that the issues relating to distributed systems and issues related to the changing requirements and technologies are evolving from time to time according to each stage of development of distributed systems such as the design phase or the maintenance phase, in addition to its changing according to the new developments in the technology.

Keywords- Scalability, Heterogeneity, Resource Management, Transparency, Openness, Quality of Service, Failure Management, Synchronization, Issues

I. INTRODUCTION

The rapid developments across computing technologies have led to the adoption of more effective communication systems that are cost-effective and highly efficient. Distributed Systems (DS) is one such approach that is being used to reduce the dependency on the physical location of the systems to transfer data, reduce operational costs, increase reliability and reduce the failures in the system [1]. However, there are various issues associated with the designing of DS in various aspects such as naming, communication, security, etc. [2], [3]. This paper considers these issues as important aspects in the area of distributed computing which need an effective review to understand the various issues in developing DS.

A DS is a system in which the components or resources are distributed over a network at different locations and interact with each other only by-passing messages, and that appears as a single coherent system to its users [4]. There are various issues that need to be considered while designing DS like increasing security and privacy in communication, ensuring reliability, scalability, heterogeneity, resource management, etc. [2]. The exhaustive testing procedures are inadequate in testing the systems as there is a high degree of non-determinism present in parallelism. Correctness of decision process and performance issues in intercross communications are two important issues that need to be addressed in the DS [5]. However, with the new advancements in DS new issues emerged over time. Data privacy and data protection are the two important security concerns identified with cloud computing, one of the fast deploying models of distributed computing [6], [7]. Authentication, integrity, and confidentiality of involved data and communications are few other problems identified in the various types of DS including cloud, grid, client/server computing [8]. Distributed Denial of Service (DDoS) attacks is another issue that is growing in the DS, especially in cloud computing [9]. Uncertainty of the architecture and shortage of practical experiences with security attacks are few reasons outlined for the increase in the issues associated with distributed computing [10]. The preliminary literature review thus identified various issues encompassing DS which are reducing efficiency and increasing security and privacy risks. However, there is a need for a detailed review in order to analyze the new challenges that have been emerging in DS.

The aim of this study is to investigate the various issues associated with DS. Accordingly, two questions need to be answered: What are the various issues associated with DS and how these issues are emerging with the development of DS over time?

However, there are not many studies conducted that have focused on various issues and challenges with a DS. Therefore, this comprehensive study using a systematic review focuses on gaining insight into the various issues, their evolution and how solutions have emerged for them.

II. RESULTS

Different studies have analyzed different issues and challenges associated with DS from various perspectives. There are various issues associated with object models and naming schemes, distributed coordination, inter-process communication, distributed resources, fault tolerance, and security, etc. [11]. Few issues are identified in naming the objects and servers, which are used in addition to addresses. They have to be identified in the system design as many operations later depend on them. Similarly, synchronization between concurrent processes is essential for the proper functioning of the system, and issues such as difficulty in obtaining complete information (no shared memory), information inaccuracy, deadlock processes are few issues associated with it. Such issues may lead to other problems in communication between the processes. In addition, compatibility, flexibility, and extensibility are few other aspects associated with system architecture and design [12]. The few other issues associated with DS are load sharing, load capacity, load distribution, failures, security violations, access violations[11], [13]. There were

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00018

12 themes identified from the systematic review for more simplify analysis of issues. These themes basically used for categorizing and analyzing issues and challenges associated with DS.

A. Scalability

The scalability is one of the major issues associated with DS. Issues such as communication capacity have to be carefully planned and designed keeping the need and requirements for the growth in the future [12]. Gandhi & Thakur [13] suggested to avoid centralized algorithms and entities to discuss scalability issues. Similarly, Wozniak et al.[14] proposed Swift/T, a programming language that supports code binding, parallel and concurrent tasks for addressing scalability challenges such as load balancing, concurrency, data distribution, etc.

B. Heterogeneity

A heterogeneous DS is a complicated system of designing, developing and managing the issues, consist of manageable hardware and software resources. It is considered a flexible DS because a variety of users can access it from different platforms [15]. The architectural design issues such as interconnectivity, interface or middleware compatibility, messaging and technological issues are few major issues that also affect the heterogeneous DS [16].

As the world rapidly moving towards Internet of Things (IoT) technology, heterogeneity can be one of the major issues in the DS; however, there are various technologies and applications are being developed to support heterogeneity in IoT [17].

C. Resource Management

In DS resources are located at different locations, which require an effective and efficient routing system for managing the communications. Managing resources over a heterogeneous network is more complicated due to the different types of hardware and software architecture and applications. The heterogeneous resources in a distributed networks, the relationship and dependencies between different resources, the variability and unpredictability of load in the network are a few major challenges associated with resource management in DS [18]. Similarly, Manvi & Krishna Shyam [19] identified major resource management issues in distributed cloud environment which include allocation, provisioning, discovery, and modeling; and stated effective resource management would result in improved scalability, Quality of Service (QoS), and efficiency of the DS. A consensus algorithm in this aspect was proposed by Xu & Li [20] for optimal resource management in a distributed environment, which proved to be effective in managing resources.

D. Security and Privacy

Security and Privacy are common issues facing various information and communication systems. DS, where there is a number of interconnected resources and systems, with a wide range of users having different accessibility features, with an architecture involving a number of regulations, standards, and policies are associated with a wide number of security and privacy issues [2]. Confidentiality, data integrity, authentication, access control, authorization, non-repudiation, accountability are a few security issues identified by Elkabbany & Rasslan [21]. Security attacks can include Distributed Denial of Service (DDoS) attacks, identity attacks, intruding, etc. [21]. Deploying security features in a DS is a complex process as there is no single point of control over the system that uses a wide range of both secure and unsecured networks in the process of communication. Messages authenticity secured communication without interruption, reliability, availability, etc. are a few major issues that need to be focused on the security aspects of DS. The traditional security measures are not effective in a dynamic environment of DS where heterogeneous hardware and software technologies are interconnected. This reflects the need for flexible security applications which may be effective in dealing with heterogeneity and dynamic environment [22]. It is necessary to focus on both storage security and computational security in DS. However, most of the studies solely focused on the storage security aspects, while neglecting the computational security [23].

E. Transparency

Transparency is an aspect that hides the complexity of the DS and reflects as a single system to the user rather than a complex system with distributed and interlinked resources [2]. Achieving complete transparency in DS can be a complex task as various aspects of transparency have to be managed effectively. The different forms of transparency include Access, Location, Replication, Failure, Migration, Concurrency, Performance, and Scaling [13]. It is essential that the movement of data and information in the DS must not affect the operations of the users.

F. Openness and Flexibility

Openness is an important feature that needs to be integrated into DS functionality. The ease of modification and enhancement determines the flexibility of the DS. The requirements of the users may change from time to time, which may result in adding additional users or additional functionalities or modules in the DS. Sometimes the existing systems may need to be replaced with few nodes to upgrade according to the user specifications. In all such situations, the DS must be flexible to adopt any such modifications without affecting the existing system functionality and operations of the users [13]. The differences in the data representation of different interfaces on different processes are a few important issues that need to be addressed. Peer to peer systems in which nodes can be replaced with other nodes is one of the effective solutions identified to support the flexibility in DS [24]. Similarly, Grid services which can provide an extensible approach for integrating the DS is one of the approaches identified by Foster et al. [25] to support flexibility.

G. Quality of Service (QoS)

Quality in DS can be related to the performance, reliability, availability, and security aspects [2], [13]. In fact, the issues in the delivery of quality can be related to most of the themes identified in this study. The presence of any major issue in all other themes can directly affect the functionality and delivery of users in the DS. Therefore, quality though an additional theme is an important factor to consider from the user perspective of the DS, as they

may identify issues in quality that may require modifications in the processers or modules of the DS. Task scheduling, resource allocation are a few aspects that are focused on developing QoS in DS. For instance, Wu et al. [26] developed a task scheduling algorithm based on the QoS driven approach which effectively processes task scheduling and management and resulted in improved load balancing and performance of the DS. Due to the heterogeneity factor in DS, it may not be possible to include all elements for evaluating the performance in a virtual environment. Considering this aspect, Bruneo [27] developed an analytical model based on stochastic rewards nets which can be used with different types of systems. The model defined several performance metrics to analyze the behavior of the data such as responsiveness, availability, utilization, etc. In addition, DS may utilize more energy in handling a large number of complex operations; which reflects the need to manage operations with minimal use of energy in order to deliver QoS at minimum costs. Traffic management, resource allocation, and task scheduling are the few important aspects that are identified and require effective management to minimize the utilization of energy resources [28].

H. Failure Management

DS comprise of various subsystems. Failures may occur in systems due to the faults in software or hardware. As the failures may occur in some parts of the system, while others continue to function normally, detecting faults in DS is usually a complex process and requires good methods to identify and recover faults. Moreover, there is a need to manage multiple systems effectively to increase reliability by detecting and addressing failures. The reliability of a system relates to the fault identification, fault tolerance, fault tracing, and fault mitigation or recovery [4]. Fault tolerance and fault avoidance are such practices that can be implemented to failure management. For instance, Chandra & Toueg [29] proposed unreliable failure detectors for achieving consensus with crash failures in asynchronous systems. The unreliable failure detectors, which can make a large number of mistakes, can be used for achieving a consensus to determine the mistakes that can be solved or mitigated. Similarly, Lupu & Slowman [30] presented two policies; authorization policy and obligation policy to address the faults occurring in the management of DS during updating or modifying the components. These policies have an aim to prevent faults by limiting the users and process restrictions to avoid any faults occur during the system upgradation. It is essential to have clear standards or outline for adopting various strategies in developing the DS along with the different types of faults that can occur so that the faults can be avoided well in advance.

I. Synchronization

Synchronizing computations in the system that has a large number of components is one of the major issues facing the DS developers. There are various synchronization methods such as semaphores, message passing, remote procedure calls, etc., which are not sufficient enough to handle the issues associated with the synchronization in DS [2]. For example, if Process A asks time and kernel responds to it with a specific time; as little later, Process B asks for time, the time that kernel responds with will be higher than that was provided for process A. The complexity of DS affected when there is no synchronized method In addition, with the heterogeneity and the dynamic nature of DS in the events of modifications, synchronization is one of the important issues that need to be addressed in order to maintain the QoS.

Various methods are studied and reviewed by different authors. For instance, grid synchronization issues of DS that may be developed in accordance with the grid disturbances were evaluated with the use of various methods in a study by Timbus et al. [31], which found the inefficiency of the methods to address all issues. Distributed shared memory can be related to the synchronization processes, from which the data can be accessed by different processes. Nitzberg & Lo [32] identified various algorithms to supports synchronization and memory management in order to identify various issues in synchronization and shared memory access in the DS.

J. Communication

Communication is an essential part of any DS which enables the service requests and responses between client and servers; between nodes; between processes and various other entities within the system. Remote Procedure Call, Message-oriented, stream-oriented, and multicast are few types of communications in DS [4]. Synchronous or blocking; and asynchronous or non-blocking are two other modes of communication that are used in the DS. Various approaches are used to address the communication issues. For addressing fault tolerance issues, diagnosers are used, which are linked with the centralized diagnosers. Inter messaging aspects and communication between diagnosers evaluated by Sengupta [34] and identified various issues in the communication process. The communication issues have to be carefully assessed in order to reach an agreement on the correctly performing processors. To address this issue, Cristian [35] proposed three protocols that provide delays, communication failures, and processor group membership information to the correct processors for reaching an agreement in synchronous DS.

K. Software Architecture

It defines the functionality of the application distributed over the various components and processors in the DS. It can be defined in layers, shown in Figure 1, which include applications, middleware, operating system, and computer and network hardware. These layers define the various functionalities and features of the DS. With the development of new technologies, innovative architectures are being proposed that can support various features such as flexibility, reliability, adaptability, security, etc. Bastin et al. [36] proposed InstaGENI, for largescale systems, which integrates lightweight cluster designs with software-defined networking that is being used in various sectors. Similarly, Balalaie et al. [37] proposed microservices architecture that supports development operations and addresses various issues during the migration.

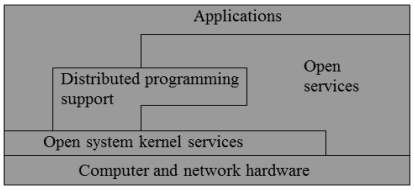

Figure 1. Layers and dependencies in DS

L. *Performance*

The performance analysis of the DS can be linked to various aspects such as fault tolerance, adaptability, flexibility, speed, cost-effectiveness, etc which would result in analyzing QoS. In addition, the ability to analyze the future growth or modifications in the DS at the design stage could save time and reduce the risk of failure and faults occurrence. Various methods and techniques, such as parallel computing, processes that use fewer resources without affecting the functionality have to be used for achieving better performance in DS [2].

Similarly, in order to improve the performance and scalability of the DS, a high performance distributed file system (Ceph) was proposed by Sage et al. [38]. This system intelligently addresses the failure detection and recovery operations by distributing them to the object storage devices.

III. DISCUSSION

The systematic review of issues and challenges associated with DS has identified 68 studies that focused on the issues and solutions in 12 aspects of features and functionalities. The number of studies identified by the year of publication is presented in Figure 2. There are relatively fewer studies published during the period 1990 to 2000 (20 studies) compared to the period from 2001 to 2018 (48 studies). The increase in the studies after 2000 can be compared to the increase in the deployment and issues identified in DS [39]. In addition, the development of new technologies such as Cloud computing, IoT, etc might have led to the various migration and management issues in DS. There are various studies focusing on issues relating to failure management, scalability, heterogeneity, resource management, QoS, security and privacy since 2000. The number of studies focused on issues in security and privacy has increased in recent years as presented in Figure 3.

The increase in the complexity of the systems with rapid changes in the requirements of technologies might have led to an increase in security and privacy risks. Studies focusing on the issues in transparency, openness, synchronization, software architecture, performance, and communication are identified from 1990 to 2000. These issues can be related to the initial phases of deploying DS such as the design phase or during the maintenance phase. Similarly, issues such as failure management, scalability, and heterogeneity are mostly discussed in the studies during the period 2000-2011, which are mainly associated with maintenance operations. Similarly, resource management, security and privacy, and QoS are the major issues identified in various studies from 2011 to 2018. These results indicated that the DS were largely deployed in the early 1990s to 2000s, which since then are growingly adopted. The next decade focused on the maintenance issues, which usually occur after deploying these systems. Since 2011 most of the issues are associated with external factors such as expectations of the users in security and privacy, rising security threats, and the quality of service expected by the users.

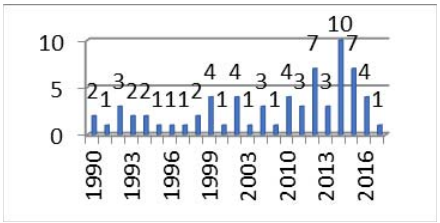

Figure 2. Yearly classification of identified studies

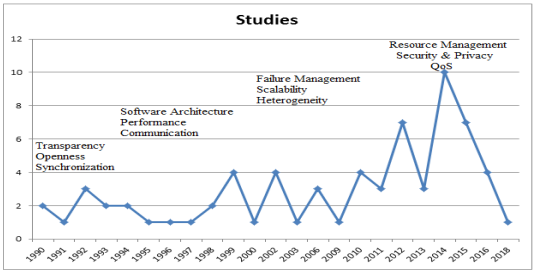

Figure 3. Evolution of the issues in different aspects of DS

IV. CONCLUSION

This study has been conducted with an objective of assessing the risks and issues associated with DS and examine how these issues are affecting the development of DS and how they are changing over time. This study has identified some major issues associated with DS associated with each phase of development. Finally, this study acts as a guide to the issues and risks that are associated with DS, which can be a useful source of information for the companies, developers, and users of DS; and mainly for the researchers studying the issues of DS.

The study can be extended further based on the time period for analyzing the different types of issues. Moreover, it can be extended further to develop a framework for DS development and maintenance from the perspective of risk management.

REFERENCES

[1] G. C, J. D, and T. K, *Distributed Systems: Concepts and Design*, 2012.

[2] A. K. T. Kamal Sheel Mishra, "Some Issues , Challenges and Problems of Distributed Software System," *Int. J. Comput. Sci. Inf. Technol.*, 2014.

[3] H. B and J. D, *Formal methods for distributed processing : a survey of object-oriented approaches*. Cambridge University Press, 2011.

[4] A. S. T and M. V S, *Distributed Systems: Principles and Paradigms*, 2007.

[5] A. P. S, "THEORETICAL ISSUES IN THE DESIGN AND VERIFICATION OF DISTRIBUTED SYSTEMS," 1983.

[6] S. S and V. K, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, 2011.

[7] R. H. W, “Internet of things: Privacy issues revisited,” *Comput. Law Secur. Rev.*, 2015.

[8] D. Z and D. L, “Addressing cloud computing security issues,” *Futur. Gener. Comput. Syst.*, 2012.

[9] Q. Y, F. R. Y, Q. G, and J. L, “Software-defined networking (SDN) and distributed denial of service (DDOS) attacks in cloud computing environments: A survey, some research issues, and challenges,” *IEEE Commun. Surv. Tutorials*, 2016.

[10] P. J, N. A, and V. C. M. L, “A survey on security issues in smart grids,” *Int. J. Appl. Eng. Res.*, 2014.

[11] C. Ghezzi, M. Jazayeri, and D. M, *Fundamentals of software engineering*. Prentice Hall, 2003.

[12] I. A. T. H, I. Y, N. B. A, S. M, A. G, and S. U K, “The rise of ‘big data’ on cloud computing: Review and open research issues,” *Inf. Syst.*, 2015.

[13] S. K. Gandhi and P. K. Thakur, “Designing Issues For Distributed Computing System: An Empirical View,” *Int. J. Innov. Res. Dev.*, 2012.

[14] J. M. W, T. G. A, M. W, D. S. K, E. L, and I. T. F, “Swift/T: Large-Scale Application Composition via Distributed-Memory Dataflow Processing,” in *2013 13th IEEE/ACM Int. Symposium on Cluster, Cloud, and Grid Computing*, 2013.

[15] Z. S, S. A, A. G, and R. B, “Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges,” *IEEE Commun. Surv. Tutorials*, 2014.

[16] W. H and L. D X, “Integration of Distributed Enterprise Applications: A Survey,” *IEEE Trans. Ind. Informatics*, 2014.

[17] D. M, S. S, F. D P, and I. C, “Internet of things: Vision, applications and research challenges,” *Ad Hoc Networks*, 2012.

[18] B. J and R. S, “Resource Management in Clouds: Survey and Research Challenges,” *J. Netw. Syst. Manag.*, 2015.

[19] S. S. M and G. K S, “Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey,” *J. Netw. Comput. Appl.*, 2014.

[20] Y. X and Z. L, “Distributed Optimal Resource Management Based on the Consensus Algorithm in a Microgrid,” *IEEE Trans. Ind. Electron.*, 2015.

[21] G. F. E and M. R, *Security Issues in Distributed Computing System Models*, no. October. 2016.

[22] S. S, A. R, L. A. G, and A. C-P, “Security, privacy and trust in Internet of Things: The road ahead,” *Comput. Networks*, 2015.

[23] L. W *et al.*, “Security and privacy for storage and computation in cloud computing,” *Inf. Sci. (Ny).*, 2014.

[24] E. M K, S. L. M, M. S, and B. M B, “Modeling and analysis of access transparency and scalability in P2P distributed systems,” *Int. J. Commun. Syst.*, 2014.

[25] I. F, C. K, J. M. N, and S. T, “Grid Services for Distributed System Integration The Open Grid Services Architecture enables the integration of services and resources across distributed, heterogeneous, dynamic virtual organizations-whether within a single enterprise or extending to external resource-sharing and service-provider relationships,” 2002.

[26] X. W, M. D, R. Z, B. Z, and S. Z, “A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing,” *Procedia Comput. Sci.*, 2013.

[27] D. B, “A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems,” *IEEE Trans. Parallel Distrib. Syst.*, 2014.

[28] A.-C. O, M. D. d A, and L. L, “A survey on techniques for improving the energy efficiency of large-scale distributed systems,” *ACM Comput. Surv.*, 2014.

[29] T. D. CHANDRA, “Unreliable Failure Detectors for Reliable Distributed Systems,”, 1996.

[30] E. C. L and M. S, “Conflicts in policy-based distributed systems management,” *IEEE Trans. Softw. Eng.*, 1999.

[31] A. T, M. L, R. T, and F. B, “Synchronization Methods for Three Phase Distributed Power Generation Systems. An Overview and Evaluation,” in *IEEE 36th Conference on Power Electronics Specialists, 2005.*, 2006.

[32] B. N and V. L, “Distributed shared memory: a survey of issues and algorithms,” *Computer (Long. Beach. Calif).*, 1991.

[33] A. D and F. L. L, “Distributed adaptive control for synchronization of unknown nonlinear networked systems,” *Automatica*, 2010.

[34] R. Sengupta, “Diagnosis and Communication in Distributed Systems,” 1999.

[35] F. C, “Reaching agreement on processor-group membrship in synchronous distributed systems,” *Distrib. Comput.*, 1991.

[36] N. Bastin *et al.*, “The InstaGENI initiative: An architecture for distributed systems and advanced programmable networks,” *Comput. Networks*, 2014.

[37] A. B, A. H, and P. J, “Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture,” *IEEE Softw.*, 2016.

[38] S. A. W, S. A. B, E. L. M, D. D. E. L, and C. M, “Ceph: A Scalable, High-Performance Distributed File System,” 2006.

[39] A. W and W. B, “The Continuing Evolution of Distributed Systems Management,” 2003.