ISSN:0975-9646

Some Issues, Challenges and Problems of Distributed Software System

Kamal Sheel Mishra[#1], Anil Kumar Tripathi[*2]

*[#1*2]Deapartment of Computer Science & Engineering,*
Indian Institute of Technology(Banaras Hinhu University)
Varanasi, India

***Abstract*— From the last two decades the trends in the Computing Industry is towards Distributed, low cost unit, and high unit volume product. Moreover software development activity is becoming more decentralized thereby expanding more development efforts for organizations. The performance of computing system has increased drastically with the inclusion of Multiprocessing and Multicomputing. This paper presents a systematic literature review on the issues, challenges, problems and solutions of the Distributed Software System (DSS). This paper is also aimed to give a report on the real problems and performance issues faced by software professionals and researchers working on Distributed Software System.**

***Keywords*— Distributed Software, Performance evaluation, task allocation, task scheduling, communication fault delay, computation fault delay, Distributed System.**

I. INTRODUCTION

With the advent of Internet and Network technologies the distributed Software System has become popular and important. Since the Industry is more concerned about the Distributed Software Development it becomes essential to discuss the Issues related to Distributed Software System. While discussing the Issues, challenges will come which is nothing but the constraints or restrictions on DSS. To minimize the challenges, problems on DSS must be discussed and the probable solution to the specific problem will definitely minimize the restrictions or constraints. It is accepted wisdom in the Software Engineering profession that developing Distributed Software System is a challenging activity. DSS are harder to design and program because of number of processes running in parallel or process may update its variable independently or problems specific to the development of distributed applications are not exactly implemented by known programming languages and software engineering environment. Distributed system [Fig -1] may be categorized as i. Only hardware and software are distributed. ii. Users are distributed or iii. Both Users and hardware, softwares are distributed. There are several definitions on what distributed systems are. Coulouris defines a distributed system as “a system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing” [17]; and Tanenbaum defines it as “A collection of independent computers that appear to the users of the system as a single computer” [18]. Leslie Lamport said that “A distributed system is one on which I cannot get any work done because some machine I have never heard of has crashed” [16]. Some of examples of Distributed Systems are world wide web – information, resource sharing, Clusters, Network of workstations, Distributed manufacturing system (e.g., automated assembly line), Network of branch office computers, Information system to handle automatic processing of orders and Network of embedded systems This paper is organized as follows: Section II discusses the related work done in this area. In Section III Different issues related to DSS are discussed. Section IV shows various challenges faced in the way of developing DSS Section V discusses the problems and solutions related to DSS so that the challenges should be minimized. Section VI summarizes the Conclusions. Lastly Section VII gives the different references used in writing this paper.

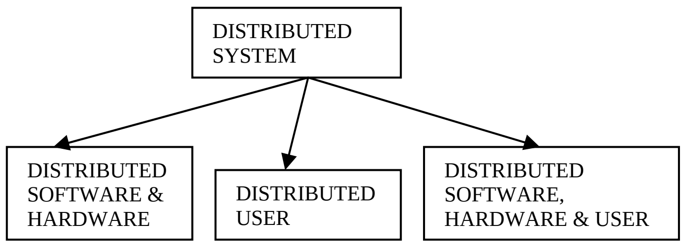
DISTRIBUTED SYSTEM
DISTRIBUTED SOFTWARE & HARDWARE
DISTRIBUTED USER
DISTRIBUTED SOFTWARE, HARDWARE & USER

Fig. 1 Types of Distributed System.

II. RELATED WORK

This section captures the significant work done towards the issues, challenges and problems of Distributed Software Systems. Many authors have identified different issues of distributed system. Sudipto Ghosh and Aditya P. Mathur[1] described the Issues in Testing component -based distributed systems related to concurrency , scalability, heterogeneous platform and communication protocol. Dan Nessett [2] focuses on Massively Distributed Systems: Design Issues and Challenges. Ahmed Khoumsi [3] worked on Temporal Approaches for Testing Distributed Systems. Hiroshi Tamura, Futoshi Tasaki, Masakazu Sengoku and Shoji Shinoda [4] focus on Scheduling Problems for a Class of Parallel Distributed Systems. Eric Koskinen and Maurice Herlihy [5] worked on Deadlocks: Efficient Deadlock Detection. Ajay Kshemkalyani and Mukesh Singhal [6] described the Distributed Computing: Principles, Algorithms, and Systems. S. Kartik and C. Siva Ram Murthy [7] worked on Improved Task-Allocation Algorithms to Maximize Reliability of Redundant Distributed Computing Systems. Veljko m. milutinovic, jakov j. crnkovic, and catherine e. houstis [8] worked on Simulation Study of Two Distributed Task Allocation

Procedures.Ian Sommerville[9] described the Distributed software Engineering. Issa Traoreand Isaac Woungang worked on [10] UML-Based Performance Modeling of Distributed Software Systems. Masoud Mansouri-Samani and Morris Sloman[11] presented the Monitoring of Distributed Systems.

III. Issues Of Distributed Softwre System

Scalability: Scaling is one of the major issues of Distributed System. The scaling issue consists of dimensions like communication capacity. The system should be designed such that the capacity may be increased with the increasing demand on the system.

Heterogeneity: It is an important design issue for the distributed systems. The communications infrastructure consists of channels of different capacities. End-Systems will possess a wide variety of presentation techniques.

Objects representation and translation: Selecting the best programming models for distributed objects like CORBA, Java etc. is an important issue.

Resource management: In distributed systems, objects consisting of resources are located on different places. Routing is an issue at the network layer of the distributed system and at the application layer. Resource management in a distributed system will interact with its heterogeneous nature.

Security and privacy: How to apply the security policies to the interdependent system is a great issue in distributed system. Since distributed systems deal with sensitive data and information so the system must have a strong security and privacy measurement. Protection of distributed system assets, including base resources, storage, communications and user-interface I/O as well as higher-level composites of these resources, like processes, files, messages, display windows and more complex objects, are important issues in distributed system

Transparency: Transparency means up to what extent the distributed system should appear to the user as a single system? Distributed system must be designed to hide the complexity of the system to a greater extent.

Openness: Openness means up to what extent a system be designed using standard protocols to support Interoperability. It is desired for developers to add new features or replace subsystem in future. To accomplish this, distributed system must have well defined interfaces.

Quality of Service: How to specify the quality of service given to system users and acceptable level of quality of service delivered to the users. The quality of service is heavily dependent on the processes to be allocated to the processors in the system, resource distribution, hardware, adaptability of the system, network etc. A good performance, availability and reliability are required to reflect good quality of service.

Failure management: How can failure of the system be detected and repaired.

Synchronization: One of the most important issues that engineers of distributed systems are facing is synchronizing computations consisting of thousands of components. Current methods of synchronization like semaphores, monitors, barriers, remote procedure call, object method invocation, and message passing, do not scale well.

Resource identification: The resources are distributed across various computers and a proper naming scheme is to be designed for exact reference of the resources.

Communications: Distributed Systems have become more effective with the advent of Internet but there are certain requirements for performance, reliability etc. Effective approaches to communication should be used.

Software Architectures: It reflects the application functionality distributed over the logical components and across the processors. Selecting the right architecture for an application is required for better quality of service.

Performance analysis: The Performance analysis of Distributed software system is a great issue. It is expected that DSS should be high speed, fault tolerant and cost effective. It is also essential towards evaluating alternative design to meet the Quality of service. [10].Moreover the ability to estimate the future performance of a large and complex distributed software system at design time can reduce the software cost and risk.

Generating a Test Data: Generating a test data to cover the respective test criteria for testing the component is a difficult task. It becomes more difficult in case of distributed system because the number of possible paths increases significantly. Test cases must cover the low level elements.

Component selection for testing: Testing distributed components require the services of other components. When a component is used with other there could be a possibility of deadlocks and race condition. There may be no error detected when only one client is used because only one thread is executed but in the case of multithreading the number of clients used for testing the components may detect the errors.

Test Sequence: The component is to be tested along with other components. What orders should be followed in testing components? If the components do not follow the layered architectural model, there could be chances of cycles among the components.

Testing for system scalability and performance: Scalability of conventional test criteria of data is a major issue in the context of testing. The concept of threading may be used in the components for improved performance while testing. But using multiple threads is a challenging task in testing.

Redundant testing during integration of component: components are first tested separately. When the entire system is tested, lots of retesting of component occurs.

Availability of source code: Software components may be developed in house or off- the- shelf. Depending upon the availability of source code various testing techniques are used for the system testing.

Heterogeneous languages, platform and Architecture: Different languages may be used for writing the components of the system. The components may be used on different hardware and software platform.

Monitoring and control mechanism in testing distributed software: Distributed software system involves multiple computers on the network. Testing monitoring and control mechanism in distributed environment is complex compared with centralized software system. Monitoring Distributed System services are also important for debugging during system development and required as part of the application itself like process control and automation [11].

Reproducibility of Events: Reproducibility of events is a challenging task because of concurrent processing and asynchronous communication occurring in the distributed environment. Moreover the lack of full control over the environment is another hurdle in this regards.

Deadlocks and Race Conditions: Deadlocks and race conditions are other great issues in distributed system especially in the context of testing. It becomes more important issue especially in shared memory multiprocessor environment [5].

Detection of existing deadlocks and Resolution of detected deadlocks [6]: Detection of deadlocks involves two issues: Maintenance of the Wait For Graph and searching of the Wait For Graph for the presence of cycles.

Testing for fault tolerance: The ability to tolerate faults in software system is required in applications like nuclear plant, Space missions, medical equipments etc. Testing for fault tolerance is challenging because the fault recovery code hardly gets executed while testing. Different fault injection techniques are used for fault tolerance by injecting faults in the system under test.

Scheduling issue for distributed system: [4] Focuses on Scheduling problems in homogeneous and heterogeneous parallel distributed systems. The performance of distributed systems are affected by Broadcast/multicast processing and required to develop a delivering procedure that completes the processing in minimum time.

Controllability and Observability issues [3]: Controllability and observability are two important issues in testing because they have an effect on the capability of the test system to check the conformance of an implementation under test. Controllability is the capability of the Test System to force the Implementation under Test to receive inputs in a given order.

Distributed Task Allocation: Finding an optimal Task allocation in distributed computing system is a challenging job keeping in mind the concept of reliability and performance[7],[8].

IV. Challenges Of Distributed Software System

Performance improvement: With the rapid growth of parallel and distributed processing in modern computers a high demand for performance improvement and low cost productivity in real life is desired. Moreover challenges like communication fault delay or computation fault delay may occur because of network failure or machine failure respectively.

Making fault tolerant DSS is a tough job. The ability to tolerate the fault and functioning normally is required by DSS. Synchronization is another important aspect in DSS because Distributed System do not have a global clock. It is required that synchronization be done as per the actual real time.

Scalability: The system must remain effective when there is a significant increase in either number of resources or number of users. The architecture and algorithms must be efficiently used under these circumstances.

Security: Security in terms of confidentiality, integrity and availability must be provided in DSS. The probable threats are information leakage, integrity violation, denial of services and illegitimate usage.

Design challenges: design of DSS must take care of responsiveness, throughput, load sharing and load balancing of the tasks.

Concurrency: Shared access to resources must be made available to the required processes.

Openness and Extensibility: Interfaces should be separated and publicly available to enable easy extensions to existing components and add new components [16].

Migration and load balancing: Task must be allowed to move within the system without affecting the operation of users or applications and load must be distributed among the available resources for better performance.

Security: Access to resources should be secured to ensure only known users are able to perform allowed operations.

There are other important challenges that need to be discussed by researchers to succeed. A deep understanding of the information flows and models of collaboration in different distributed software development activities have to be developed.

V. Problems And Solutions For Distributed Software System

Performance is an important issue and challenge of Distributed Software System. To minimize the constraints, and thus challenges, problems are to be discussed and solutions are to be provided.

In distributed software system different task scheduling algorithms are developed. These algorithms should be evaluated on different available task evaluation parameters for a specific task graph which ultimately should represent the DSS and the best algorithm performance result should ultimately be adopted. This approach will minimize the challenges of DSS.

This can further be elaborated as to be more practical a fault is to be injected in the DSS and the performance of the algorithms on certain parameter should be observed. The fault injected may be Communication fault delay or Computation fault delay or both as the case may be. Communication fault delay may be because of network or link failure and Computation fault delay may be because of hardware problem or processor not responding. Now the performance of algorithms should be observed under these circumstances thereby reducing the challenges and improving the performance of the DSS. Authors [15] describes the MetaPL approach to the performance analysis of distributed software systems which is an XML based language that can describe programs written in different distributed programming language and environments.

VI. Conclusions

Distributed systems are an important field both in research oriented environments and in industrial organization. They are complex to build and maintain and can be error prone. The nature of this paper is exploratory. Different issues, challenges and problems of Distributed Systems are discussed. It has succeeded if it motivates readers to develop an alternative taxonomy of the issues, challenges and problems associated with distributed software systems.

References

[1] Issues in Testing distributed component -based systems, Sudipto Ghosh,Aditya P. Mathur, Software Engineering Research Centre,West Lafayette, March 1999.

[2] Massively Distributed Systems: Design Issues and Challenges, Dan Nessett, Technology Development Center, 3Com Corporation. Proceedings of the Embedded Systems Workshop Cambridge, Massachusetts, USA, March 29–31, 1999

[3] A Temporal Approach for Testing Distributed SystemsAhmed Khoumsi, Member, IEEE, IEEE Transactions on Software Engineering, vol. 28, no. 11, november 2002

[4] Scheduling Problems for a Class of Parallel Distributed Systems,Hiroshi Tamura, Futoshi Tasaki, Masakazu Sengoku and Shoji ShinodaNiigata Institute of Technology, Japan , Faculty of Engineering, Niigata University, Japan, IEEE 2005.

[5] Deadlocks: Efficient Deadlock Detection, Eric Koskinen, Maurice Herlihy Computer Science Department Brown University Providence. Copyright 2008 ACM 978-1-59593-973-9/08/06.

[6] Distributed Computing: Principles, Algorithms, and Systems ,Chapter 10, Ajay Kshemkalyani and Mukesh Singhal.

[7] Improved Task- Allocation Algorithms to Maximize Reliability of Redundant Distributed Computing Systems, S. Kartik, C. Siva Ram Murthy, Wipro Infotech Limited, Bangalore Indian Institute of Technology, Madras, IEEE transactions on reliability, vol. 44. no. 4, 1995 december.

[8] A Simulation Study of Two Distributed Task Allocation Procedures, veljko m. milutinovic, senior member, ieee, jakov j. crnkovic, and catherine e. houstis, member, IEEE, IEEE transactions on software engineering. vol.14, no i . january 1988

[9] Distributed software Engineering, Ch-10,Ian Sommerville, Page:479-501.

[10] UML-Based Performance Modeling of Distributed Software Systems, Issa Traore, Department of Electrical and Computer Engineering University of Victoria ,Canada, Isaac Woungang , Canada, Ahmed Awad El Sayed Ahmed ,Canada, Mohammad S. Obaidat , USA.

[11] Monitoring Distributed Systems, Masoud Mansouri-Samani and Morris Sloman , Imperial college.

[12] A review of current research and critical issues in distributed System software.", J. A. Stankovic, K. Ramamrithan and H. K. Walter, lEEE Distributed Processing Tech. Committecj Newslett., pp. 14- 47, Mar. I, 1985.

[13] Scheduling in Distributed Computing System., D P Vidyarthi, B K Sarkar, A K Tripathi and L T Yang, Springer..

[14] Problems and Solutions in Distributed Software Development: A Systematic Review, Miguel Jiménez and Mario Piattini, Spain.

[15] The MetaPL approach to performance analysis of distributed software systems, N. mazzacca, M.Rak and U. Villano, Italy, ACM 2002.

[16] Distributed Systems and Recent Innovations: Challenges and Benefits, Krishna Nadiminti, Marcos Dias de Assunção, and Rajkumar Buyya, The University of Melbourne, Australia

[17] Distributed Systems – Concepts and Design, G. Couloris, J. Dollimore, and T. Kinberg, 4th Edition, Addison-Wesley, Pearson Education, UK, 2001.

[18] Distributed Systems: Principles and Paradigms, A. Tanenbaum and M. Van Steen, Prentice Hall, Pearson Education, USA, 2002.

Author's Profile:

Anil Kumar Tripathi is Professor of Computer Engineering at Indian Institute of Technology (Banaras Hindu University), Varanasi, India. He received his Ph.D. degree in Computer Engineering from Banaras Hindu University; and M.Sc. Engg. (Computer) degree from Odessa National Polytechnic University, Ukraine. His research interests include parallel and distributing computing, and software engineering. He has to his credit more than 60 research papers published in International journals. He has co-authored two research monographs: one from Springer USA and other from John Wiley USA. Fifteen students have completed their Ph.D under his supervision.

Kamal Sheel Mishra is M.Tech (Computer Engg.) and working as Associate Professor and Dean ,Computer Science department in the School of Management Sciences , Varanasi, India. He is having more than 18 years of teaching and Industry experience. His research interests include Software Engineering, parallel and Distributed Computing. Currently he is pursuing Ph.D. from Department of Computer Science and Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi, India.