International Journal of
Algorithms Design and Analysis Review

Volume 1, Issue 2, 2023
July–December
DOI (Journal): 10.37591/IJADAR

https://journals.stmjournals.com/ijadar

Research IJADAR

Deadlock Controlling Algorithms for Distributed Database Systems

Z.L. Patricia[1], Daniel Matthias[2,*], E.O. Taylor[1]

Abstract

When the demand for a system resource exceeds the system's capacity, deadlock – an operating system problem – results. The problem of deadlock frequently causes a distributed database's performance to lag. This research critically examined two types of deadlock problems that have an impact on a distributed database's performance. Transaction control and transaction location deadlock difficulties were the specific challenges that the article specifically addressed. In this paper, deadlock prevention techniques for distributed database systems are suggested. The suggested system was developed utilizing MySQL, Hypertext Preprocessor, and Object-Oriented Analysis and Design Methodology (OOADM). Based on the deadlock management and control approaches known as hold and wait and mutual exclusion, the suggested system was able to accept two requests for system resources and then give the required system resource. The suggested system model outperforms the current system in terms of speed, accuracy, and deadlock avoidance, according to the performance evaluation of both systems. This paper used pre-defined parameters to illustrate the outcomes and performance assessment of the existing and new systems for deadlock management and control. Programs for both systems were run before compiling the findings for both models. The pre-defined parameters for both models include the number of executed iterations, the quantity of resources sought per iteration, the quantity of deadlock control strategies employed, and the quantity of concurrent requests granted depending on deadlock avoidance. For a distributed database system, this work also contributed deadlock prevention strategies.

Keywords: Deadlock, distributed databases, algorithm, performance, operating system

INTRODUCTION

The majority of users of distributed databases are unaware of the stalemate issue. In the realms of multi-processing, parallel computing, and distributed systems, deadlock is a common and challenging issue that often leaves individuals feeling perplexed and disappointed. Deadlock is described as a situation where a group of blocked processes are waiting for one another to release resources that they are all holding while being blocked themselves. A deadlock, as defined by Cleland-Huang et al. [1], occurs when each individual within a group is in a state of waiting, reliant on one or more of their fellow members to take action, often involving the sending of a message or the release of a lock. As an illustration, if a system has two tape drives, processes 1 and 2 both have one, and processes 1 and 2 both require the second tape drive, which is held by process 3. Physical occurrences like traffic jams and the management of crowds frequently experience the problem of deadlock [2].

In a distributed database environment, a deadlock arises when a process or thread enters a state of waiting, necessitating access to a system resource held by another process, which is concurrently waiting for a distinct resource held by yet another

***Author for Correspondence**
Daniel Matthias
E-mail: Matthias.daniel@ust.edu.ng

[1]Student, Department of Computer Science, Rivers State University, Port Harcourt, Nigeria
[2]Lecturer, Department of Computer Science, Rivers State University, Port Harcourt, Nigeria

Received Date: September 05, 2023
Accepted Date: September 20, 2023
Published Date: October 04, 2023

Citation: Z.L. Patricia, Daniel Matthias, E.O. Taylor. Deadlock Controlling Algorithms for Distributed Database Systems. International Journal of Algorithms Design and Analysis Review. 2023; 1(2): 10–17p.

waiting process. A system is deemed to be in a deadlock when a process cannot indefinitely change its status because the resources it needs are being utilized by another waiting process [2]. One of the most dreaded issues in distributed databases is deadlock, which prevents tasks from ever being completed and keeps them in a waiting state indefinitely [3]. Apt and Olderog [4] claim that in order to prevent deadlock, starvation, and data corruption, operating system kernels required tens of thousands of locks and synchronizations that demanded tight adherence to procedures and standards. Since the operating system's resources scheduler is aware of the resources that each process has locked and/or is presently requesting, it is simple to identify a stalemate that has already happened [5].

This study offers strategies for distributed database system deadlock control. A distributed database system fundamentally refers to a database that is spread out across multiple locations, such as different computers or a network of computers, rather than being confined to a single system. This distributed database system is distributed across various sites, each having its own distinct physical components.

When the demand for a system resource exceeds the system's capacity, deadlock, an operating system problem, results. The problem of deadlock frequently causes a distributed database's performance to lag. The performance of a distributed database is affected by two different types of deadlock problems, which were critically examined in this work. The work specifically addressed two types of deadlock problems: transaction control deadlock problems and transaction location deadlock problems. If two transactions (such as A and B) compete for exclusive usage of a resource (such as a specific record in a data set that the other already owns), such a situation is known as a transaction location deadlock. Transaction A, for instance, waits for the resource to become accessible. When database resources are negatively impacted because both processes are perpetually waiting for each other, a transaction control deadlock arises. The SQL server's assistance ends this contention problem.

RELATED WORKS

The study conducted by Duato et al. [6], which is titled "A theory for deadlock-free dynamic network reconfiguration" established a theoretical framework for assessing the potential for deadlocks in dynamic network reconfiguration methods. This framework serves as a foundation for creating design approaches that are valuable for implementing these techniques. Its relevance extends to system-on-chip networks, as well as the interconnection networks commonly employed in multiprocessor servers, network-based computing clusters, and distributed storage systems. This theory advances new ideas crucial to the case of dynamic reconfiguration while building on core ideas established by earlier theories.

A theoretical framework for dynamic deadlock avoidance in concurrent systems employing traditional mutual exclusion and synchronization primitives (such as multithreaded C/P threads programs) was described in the theory of deadlock avoidance via discrete control [7]. We build a formal model of the program starting with control flow graphs taken from the source code, and then use discrete control theory to automatically create deadlock avoidance control logic that is implemented by program instrumentation. By delaying lock acquisitions, the control mechanism prevents deadlocks during runtime. The software we instrumented with our synthesized control logic cannot deadlock, according to discrete control theory [8]. Zhan et al. [9] claim that the dining philosopher's dilemma may be summed up as five silent philosophers sitting at a circular table either eating or pondering in their paper titled a deadlock prevention utilizing adjacency matrix on dining philosophers' problem. They do not think while they are eating, and they do not eat while they are thinking. The difficulty is frequently illustrated using rice and chopsticks rather than spaghetti and forks because the huge bowl of spaghetti in the center takes two forks to serve and consume. Each adjacent pair of philosophers is given a fork, and each philosopher is only permitted to utilize the forks to his left and right. The philosophers do not converse with one another, though. The dining philosophers' conundrum is a well-known synchronization issue. By using an algorithm to restrict the damage caused by four essential circumstances, deadlock can be avoided. Java supports multithreading at the language level, allowing programmers to utilize it to simulate and verify the prevention of deadlock in the study of dining philosophers' problems. All resource utilization will decrease as a result of the pre-allocation strategy. We suggest a solution that is highly effective in terms of system performance and resource usage. To improve resource usability at the same time, Zhan et al. [9] revised the bound range between higher and lower. Even though the model is small and simplistic, it still addresses a crucial and intriguing issue in the field of multi-threading and multi-processing in computer programming.

Numerous strategies have been used to implement the aforementioned situation awareness models, as evidenced by the significance of the knowledge on deadlock avoidance as the foundation of the technology tools for situation awareness [6]. All degrees of situation awareness as outlined by his work have been model using the blackboard systems. The systems on the chalkboard are the product of language process model.

The second element of the blackboard system involves dissecting a given situation into one or more hierarchical panels containing symbolic information, often organized in layers of different levels of abstraction. In contrast to other sources of knowledge that analyze the posted information and make deductions about future situations or conditions, these perceptual knowledge sources encode sensory input and deposit it in the appropriate locations within the blackboard structure. Thirdly, since its inception in the early 1980s, the application of production rules and expert systems for situation awareness has been constant. In contrast, the model of situation awareness does not employ case-based reasoning. The model of the case-based reasoning paradigm in situation awareness has a lot of potential, nevertheless. Early analyses of expert systems focused primarily on uses that required drawing conclusions from a collection of observed facts in order to arrive at a more comprehensive assessment of the situation that succinctly explained the facts. The match-select-apply cycle, which can be applied in either a forward-chaining fashion from antecedents to consequents or in a backward-chaining fashion from consequents to antecedents, describes the typical operation of an expert system.

A significant aspect of understanding human reactions is situation awareness modeling and additionally, the use of specific rules for defining situations is still evolving. For instance, the following explicitly modeled rules for situation awareness are included:

IF [(event 1 is true) AND (event 2 is true) …] THEN [Situation 1 is true] (1)

Alternatively, production rules are more frequently defined implicitly in a fairly straightforward approach to implementing "situated action" agents, with nearly equivalent production rules for actions, such as:

IF [situation 1 is true] THEN [perform maneuver type 1] (2)

Consequently, this approach enables more extensive situational analyses, which prove to be highly valuable. The adoption of detecting technologies for operating system deadlock avoidance is urgently required. Additionally, based on the strategic environment, it is feasible to determine the concept's scope and specific areas that require further research. It is crucial to determine what is required to enable collaboration because the multilingual environment demands it. The conceptual model and guidelines are offered with organizational level support. The purpose of the conceptual model and rules is to provide a broad overview of the condition of the organization being tracked by his method.

The model unifies all the levels and elements while capturing the core of Endsley's approach [10]. It is possible to view reality from a person's point of view and comprehend how particular elements affect particular degrees. An individual's unique qualities become a team asset when they are a team member. The person must be aware of the needs of the team and comprehend the tools, systems, and procedures relevant to their operations. With their assistance, one can get a personal understanding of each team member's needs and learn what is required for the group to behave and function as a unit in pursuit of the overarching objective. The question of how to create teams with the necessary rules, roles, and resources to meet the necessary needs as a team can also be regarded from an organizational standpoint.

And last, which people are best suited to work as a team in given building settings. Of course, from a technological perspective, it is also feasible to see what skills must be developed for a team to work well.

Additionally, sharing knowledge in an international collaboration in a maritime setting has been a fascinating atmosphere. Endsley will tie this incredible trip together with the situational awareness idea, which suggests the finished product he has been working so hard to create. Although it has been established that technology should be viewed as an enabler, not always the answer, the technological level is also taken into consideration. Technology is still being integrated more and more into the workplace. The monitoring tool is created to acquire the best performance possible from the technological equipment employed as information sharing moderators. The monitoring ideology is reviewed and aspects are shown that, from a technology standpoint, can either obstruct or promote information sharing. Endsley also made the point that any sort of collaboration, whether it is between national organizations or involves transnational actors, can become stuck due to time constraints and unforeseen circumstances. The idea can serve as a discussion starter for enhancing organizational inner readiness as well as for collaboration and understanding other organizations' limitations and potential for future collaboration because organizations and their methods of working toward the same end result vary. Although countries are making an attempt to coordinate collaboration among authorities, there are still obstacles to overcome and opportunities to seize. These obstacles and opportunities can be found within the organizational environment.

Although cloud computing has become one of the hottest buzzwords in recent years, it is surprising to learn that people have been using it for more than 10 years, according to a study on cloud storage and its problems in cloud computing [6]. Cloud-based services including Gmail, Facebook, Dropbox, Skype, PayPal, and Salesforce.com are examples of those that did not approach them in this way. The primary concept behind the cloud is that data may be accessible over the internet without needing to have a thorough understanding of the communications systems that make it possible. Cloud storage is one of the primary services provided by cloud computing. When using cloud storage, data might be kept on numerous servers owned by third parties without the user's knowledge or consent. The necessity to handle, organize, and most importantly store data is a fundamental issue that people or organizations must deal with as data volumes grow every day. Even if they performed a fantastic job, the authors may have improved their security strategy and paradigm by implementing their designed public cloud data storage with a body area network and steganography.

On cloud storage claims that organizations as well as the general public are using computers more frequently every day [11]. The answers to questions regarding the storage, sharing, global accessibility, management, constant availability, and affordability of crucial data lie within the realm of cloud computing. As per the report, cloud computing represents a concept aimed at delivering widespread, convenient, and on-demand network connectivity to a shared reservoir of adaptable computing resources that can be swiftly provisioned and deprovisioned with minimal administrative effort or service provider intervention. Easy access to the data is made possible via cloud computing, as is high-speed computing. Another significant issue that modern software organizations face is the ongoing availability of data storage at reasonable prices. Their study's main flaw, though, is that the constructed model lacked cost-benefit analysis and benchmarking.

The reviews and blog datasets that were downloaded from social networking sites lacked organization and required classification to yield useful data [9]. With the aid of supervised machine learning techniques, they might be divided into three categories: positive, negative, and neutral.

In that study, the authors presented four distinct machine learning algorithms—Naive Bayes (NB), Maximum Entropy (ME), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM)—for the classification of emotions using metrics such as precision, recall, F-measure, and accuracy. They utilized supervised machine learning algorithms and an n-gram approach to categorize movie reviews in their research, which was commendable. However, it is worth noting that their system lacks the ability to support multiple languages, which limits its capacity to leverage abstractions and accommodate the diversity found in extensive software repositories.

METHODS

Object-oriented analysis and design methodology (OOADM) was employed in this work. The goal of an object-oriented approach is to break down the behavior and structure of information systems into manageable modules that combine both data and process. Object-oriented design (OOD) seeks to enhance the quality and efficiency of system analysis and design by making these processes more user-friendly and accessible [10]. It permits the usage of MySQL and object-oriented programming languages like Hypertext Preprocessor, which is useful for server-side scripting.

The mutual exclusion, holds, and wait approaches are all integrated in the suggested system for controlling deadlock. The purpose of the mutual exclusion technique is to demonstrate the necessity of holding at least one resource in a non-shareable mode, which allows only one process to utilize it at a time. If not, the processes would not be stopped from utilizing the resource as required. The resource can only be used by one process at a time. The purpose behind the hold-and-wait technique is to indicate that a process currently holds at least one resource while simultaneously requesting additional resources that may be held by other processes. It is essential to note that there is no pre-emption allowed; only the process that currently holds a resource has the authority to release it. The suggested method for distributed database system deadlock control aims to identify stalemate by routine system state checks, then recovers to a safe state utilizing recovery techniques. The proposed system for deadlock control in a distributed database is depicted in Figure 1 by its architecture.

Algorithm 1: Mutual Exclusion Model

Step 1: Let Work and Finish be vectors of length ‘m’ and ‘n’ respectively.
Step 2: Initialize: Work = Available
Step 3: Finish[i] = false; for i=1, 2, 3, 4….n
Step 4: Find an i such that both
Step 5: Finish[i] = false
Step 6: Needi <= Work
Step 7: if no such i exists goto step (4)
Step 8: Work = Work + Allocation[i]
Step 9: Finish[i] = true
Step 10: goto step (2)
Step 11: if Finish [i] = true for all i
Step 12: then the system is in a safe state

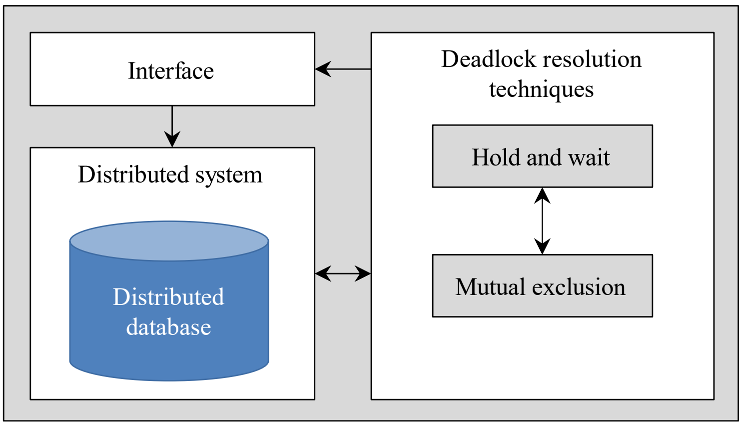

Figure 1. Architecture of the proposed system for deadlock control in distributed databases.

Algorithm 2: Hold and Wait

Step 1: If Requesti <= Needi
Step 2: If Requesti <= Available
Step 3: Have the system pretend to have allocated the requested resources to process Pi by modifying the state as follows:
Step 4: Available = Available – Requesti
Step 5: Allocationi = Allocationi + Requesti
Step 6: Needi = Needi – Requesti

RESULTS

This paper used pre-defined parameters to illustrate the outcomes and performance assessment of the existing and new systems for deadlock management and control. Programs for both systems were run before compiling the findings for both models. The pre-defined parameters for both models include the number of executed iterations, the quantity of resources sought per iteration, the quantity of deadlock control strategies employed, and the quantity of concurrent requests granted depending on deadlock avoidance as shown in Table 1.

DISCUSSION

In this study, stalemate from earlier efforts is discussed. The study provided a theoretical foundation that may be used to assess the deadlock characteristics of dynamic network reconfiguration techniques and to develop design methodologies for obtaining such techniques. This concept can be applied in system-on-chip networks and interconnection networks commonly found in multiprocessor servers, network-oriented computing clusters, and distributed storage systems. This theory advances new ideas crucial to the case of dynamic reconfiguration while building on core ideas established by earlier theories.

PHP and the MySQL Relational Database were used to build the designed system in this study. A server-side programming language called PHP is used to connect a web application's front end and back end. The relational database management system MySQL uses pre-defined tuples and attributes to store data. An open-source platform called Hypertext Preprocessor offers affordable fixes for problems with online applications. After running the applications for both systems, the results for both were compiled. The number of executed iterations, the quantity of resources demanded per iteration, the number of deadlock management strategies employed, and the number of concurrent requests granted based on deadlock avoidance are all pre-defined parameters for both systems.

The existing system and the suggested system were both modified and revised a total of five times. After five repetitions, the current system architecture was unable to approve requests based on deadlock avoidance. The suggested system, which the present system could not, as stated in Table 2, was able to accept two requests for system resources and then grant the requests based on the deadlock management and control techniques known as hold and wait and mutual exclusion. The suggested system model outperforms the current system in terms of speed, accuracy, and deadlock avoidance, according to the performance evaluation of both systems. In multi-processing, parallel computing, and distributed systems, deadlock is a typical issue. Deadlock is essentially a situation where a group of halted processes are each holding a resource and waiting for another process in the group to release a resource that they are all waiting for.

Table 1. Result of the new system.

S.N.	Parameters	Values
1.	Number of iterations carried out	5
2.	Number of requested resource per iteration	2
3.	Number of applied techniques for deadlock mgt. and control	2
4.	Number of concurrent granted request based on deadlock avoidance	2

Table 2. Comparative analysis of the existing model and newly developed model.

S.N.	Parameters	Old system	New system	Expected output of perfect system
1.	Number of iterations carried out	5	5	Excellent
2.	Number of requested resource per iteration	1	2	Very good
3.	Number of applied techniques for deadlock management and control	1	2	Very good
4.	Number of concurrent granted request based on deadlock avoidance	0	2	Excellent

CONCLUSION

This study created and applied strategies for deadlock prevention in networked databases. The hold and wait and mutual exclusion approaches make up the system that has been designed. The purpose of the mutual exclusion strategy is to guarantee that the distributed database's non-shareable mode must hold at least one resource.

When employing the hold and wait strategy, a process must hold at least one resource continuously while attempting to acquire additional resources currently held by other processes. On the basis of a survey of recognized deadlock problems in several information systems, the designed system was effectively implemented. Using MySQL and Hypertext Preprocessor, the created system was also implemented in this study. The developed system's performance was also assessed. A distributed database management system's deadlock was especially addressed by the developed technology. A serious problem that impacts information systems is deadlock. The majority of users of distributed database management systems are not aware of deadlock problems. Hold and wait and mutual exclusion strategies are used in the developed system to manage deadlock.

There was a total of five iterations completed for both the current and proposed system systems models. After five repetitions, the current system architecture was unable to approve requests based on deadlock avoidance. Based on the deadlock management and control approaches known as hold and wait and mutual exclusion, the suggested system was able to accept two requests for system resources and then give the required system resource. The suggested system model outperforms the current system in terms of speed, accuracy, and deadlock avoidance, according to the performance evaluation of both systems.

REFERENCES

1. Cleland-Huang J, Chambers T, Zudaire S, Chowdhury MT, Agrawal A, Vierhauser M. Human-machine teaming with small unmanned aerial systems in a MAPE-K environment. ACM Trans Autonomous Adaptive Syst. In press. 2023. doi: 10.1145/3618001.
2. Viloria A, Lezama OB, Mercado-Caruzo N. Unbalanced data processing using oversampling: machine learning. Procedia Computer Sci. 2020; 175: 108–113.
3. Borri E, Tafone A, Romagnoli A, Comodi G. A preliminary study on the optimal configuration and operating range of a “microgrid scale” air liquefaction plant for liquid air energy storage. Energy Conversion Manage. 2017; 143: 275–285.
4. Apt KR, Olderog ER. Fifty years of Hoare's logic. Formal Aspects Comput. 2019; 31: 751–807.
5. Dai X, Long S, Zhang Z, Gong D. Mobile on ant colony robot path algorithm planning with based A* heuristic method. New Advances at the Intersection of Brain-inspired Learning and Deep Learning in Autonomous Vehicles and Robotics. Sep 2020; 13: 47615.
6. Duato J, Lysne O, Pang R, Pinkston TM. A theory for deadlock-free dynamic network reconfiguration. IEEE Trans Parallel Distributed Syst. 2005; 16 (5): 412–427.
7. Rezende D, Danihelka I, Gregor K, Wierstra D. One-shot generalization in deep generative models. In: ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, June 19–24, 2016. pp. 1521–1529.
8. Sethi S, Rout A, Mishra D. An effective and scalable AODV for wireless ad hoc sensor networks. International Journal of Computer Applications. Aug 2010; 5(4): 33-38.

9. Zhan JS, Guo YN, Liu CL. A deadlock prevention using adjacency matrix on dining philosophers problem. Appl Mech Mater. 2011; 121: 1191–1195.
10. Endsley EW, Almeida EE, Tilbury DM. Modular finite state machines: Development and application to reconfigurable manufacturing cell controller generation. Control Engineering Practice. Oct 2006; 14(10): 1127–1142.
11. Veerabhadrappa, Rangarajan L. Bi-level dimensionality reduction methods using feature selection and feature extraction. Int J Computer Appl. 2010; 4 (2): 33–38.

