iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.49805

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Survey on Graph DB for Impact Analysis in Payment Platforms

Isha Kanade[1], Manasi Hatekar[2], Janvi Santani[3], Prasad Gosavi[4], Deepa Sapkal[5]
[1, 2, 3, 4, 5]*Department of Information Technology, Pune Institute of Computer Technology*

Abstract: ***Impact analysis is the process of analyzing and predicting the consequences of any changes in the system. This analysis is important because it helps to determine which components get impacted when a component goes down or fails and how the system responds to this change. Payment platforms are a composition of several different components such as CBS, databases and SMS G/W, UPI switches. These components are dependent on each other and work together to perform successful transactions. A graph database (Graph DB) is a database that uses graph structures for storing data. Instead of using tables or documents to represent and store data, it makes use of nodes, edges, and relationships. The nodes of the graph contain the data, and the edges between the nodes show how the data are related to one another. Hence, a graph database can be easily used to represent various components of the payment platform and the relationships between them. Graph database makes it easy to understand the effect of change in one of these components on another.***
Keywords: graph database, Neo4j, impact analysis, payment platform, relationships.

I. INTRODUCTION

Impact analysis is a crucial process for businesses to evaluate the potential consequences of any changes or modifications made to their products or applications. It is a crucial risk management technique that can help teams reduce potential losses caused by significant changes to a company, process, or product. It also helps organizations prepare for potential issues that may arise and enables teams to work together to evaluate the impact of changes and make implementation plans, promoting team buy-in and preventing disruption of team cohesiveness. Refraining from making changes due to the possibility of issues is not always the best course of action, and impact analysis can help organizations become better prepared to address potential issues. In the case of payment systems, a failure in one component can lead to a chain reaction of failures, causing significant downtime and potential financial losses. Therefore, having an efficient impact analysis environment in place for payment systems is essential to mitigate these risks.

Graph databases are increasingly being used to represent the complex relationships between components in a payment system. Graph databases are specialized in the management and storage of data in the form of interconnected graph structures, with nodes representing entities and edges representing their relationships. This enables a more natural and visual portrayal of the connections between the different components, making it simpler to conduct impact analysis on any modifications made to the system.

By leveraging the power of graph databases, businesses can quickly and accurately identify the potential impact of any modifications to their payment systems. This not only helps to minimize downtime and prevent financial losses but also enables businesses to proactively address potential issues before they become critical. As such, incorporating graph databases into impact analysis environments is becoming an increasingly popular solution for payment system providers seeking to improve their system's reliability and performance.

II. LITERATURE SURVEY

Several research papers have proposed innovative techniques to facilitate impact analysis. The paper [1] presents a framework for managing traceability relationships between requirements and architectures. The framework uses a graph-based approach to model the relationships between requirements and architecture artifacts. The approach is based on a set of concepts, including requirements, architecture elements, and traceability relationships.

The framework provides a set of tools and techniques for managing traceability relationships, including traceability visualization, traceability querying, and traceability analysis. The framework is evaluated using a case study of a complex software system. The evaluation shows that the framework is effective in managing traceability relationships and can be used to improve the quality of software systems.

©IJRASET: All Rights are Reserved

Authors of paper [2] presented an innovative static technique supporting software modification impact analysis. To record the control connected to component calls and produce control flow channels in a programme, it uses a new model based on control call graphs. The potential set of components that might be impacted by a specific modification is found using the produced pathways. When compared to results produced using conventional methods based on call graphs, the suggested strategy may predict impact sets with greater accuracy. Regression testing and predicted impact analysis can both be done using it. The authors of [3] provided an overview of the methods put out to facilitate impact analysis that leverage traceability. Three were discussed, considering the different types of traceability links (vertical or horizontal), the source of information that was used to create them (knowledge-based or structural), and how they were represented in the traceability management system (implicit or explicit). The major challenges in traceability management were outlined which are namely traceability recovery and link evolution. Paper [4] explores the different programming algorithms like Call graph, Static Slice, Static Execute After, Co-changing files retrieved by SVN repositories which can be used for impact analysis. One issue is that hypothetical change requests might not accurately replicate actual maintenance scenarios, and another is that the algorithm's programmers might not be familiar with the specific project being used for the impact study.

Paper [5] examines the various issues that arise when software projects conduct impact analyses. First and foremost, it can be difficult to gather and organize all the potential dependencies of a decision. It's also crucial to make sure that these impacts are managed properly. The abundance of information sources and the fact that it has been used in various formats for years are other significant issues. Based on the recognized change impact patterns, this paper [6] studies the propagation of change amongst related services and processes. The intermediate outcomes of the analysis process, these change impact patterns are used in the creation and upkeep of applications and information systems. They developed a prototype known as the "service change analyzer" that put the suggested methodology for measuring change impact into practice.

Paper [7] explores the concept of graph DBs, Neo4j, and its applications. According to it, graph databases particularly excel when used in applications where knowledge of data architecture or interconnectivity is crucial. The focus of a graph database like Neo4J is on elements like data structures, data model characteristics, and query capabilities that are extensively utilized to describe various systems. For systems like biological, semantic, network, and recommender systems that need the kind of data model that only they can provide, the graph database makes perfect sense. In the paper [8], researchers introduced the “UML to Graph DB” framework, which creates database-level queries from business rules and invariants established using the Object Constraint Language and converts conceptual schemas described using the Unified Modelling Language or commonly known as UML into a graph representation. In order to make the integration of various types of graph databases easier, this framework uses a new Graph DB metamodel. They intend to add refactoring techniques on top of the Graph DB model in their further work to enable designers to adjust the data representation in accordance with certain requirements, such as query execution speed or memory use.

Authors of paper [9] defined a Model View Control (MVC) architecture. The Storage Model component uses graph databases. As graph databases are exceptionally scalable and their queries do not suffer from an increased amount of information, unlike relational databases. Impact analysis is all about analyzing relationships. Graph Databases prioritize relationships more than data itself, unlike relational databases. They defined System Lifecycle Objects mappings to the graph model but the formalization of how SLOs are mapped into graph models is required in their work.

Overall, these research papers offer a comprehensive understanding of impact analysis techniques and their applications in software engineering.

III.PROPOSED METHODOLOGY

Impact analysis is a critical process in software maintenance, which involves assessing the potential consequences of modifications to a system. The primary goal of impact analysis is to identify which parts of a system are most likely to be affected by a change so that developers can focus their efforts on those areas. This process is especially crucial for payment systems, which involve a large number of interconnected components and have zero tolerance for downtime. Through an extensive literature survey, we have discovered that Graph DB is an efficient method for representing complex systems and has been successfully used for impact analysis in various domains. Graph databases like Neo4j are designed to model complex relationships between entities, such as nodes and edges, in a more natural and intuitive way. Compared to traditional relational databases, graph databases are more suitable for handling linked data and can provide faster query results even for large datasets. In our proposed approach, we suggest using Neo4j to represent the payment system and assist in impact analysis. Neo4j has several features that make it ideal for modeling complex systems like payment networks. With Neo4j, we can easily create nodes that represent each component of the payment system, such as servers, databases, and switches. We can then establish relationships between these nodes according to the payment system requirements, which will make it easier to find out the dependencies between all the components of the network.

By using Neo4j for impact analysis, we can quickly identify which components of the payment system are most likely to be impacted by a change and focus our efforts on those areas. This will help reduce downtime and improve the overall reliability of the payment system. Moreover, Neo4j has functionalities to create nodes and establish relationships dynamically based on real-time data, which makes the approach more efficient and effective.

In conclusion, we propose using Neo4j as a graph database for impact analysis in payment systems. This approach will help us to understand the dependencies between different components of the payment system and assess the potential impact of any changes made to the system.

IV. SYSTEM ARCHITECTURE

Graph DB is a type of NoSQL database that has data storing and data processing methods different from other types of NoSQL databases and the RDBMS. A Node, Edge, and Relationship are the basic components of Graph DB. A Node is a data point that has a label (i.e., what category does that node belong to. E.g., User, Admin etc). By categorizing nodes into sets, where all nodes with a particular label are members of the same set, labels help to define the domain. There can be zero to many labels on a node. A node as well as an edge has a list of attributes that specifies the properties of them. An edge is a connecting link between a source node and a target node. A relationship describes how a connection between a source node and a target node is related. It is possible for a node to have a relationship with itself. Relationships organize nodes into structures so that a graph to resemble a list, a tree, a map, or a compound entity — any of them may be combined into more complex, richly interconnected structures.

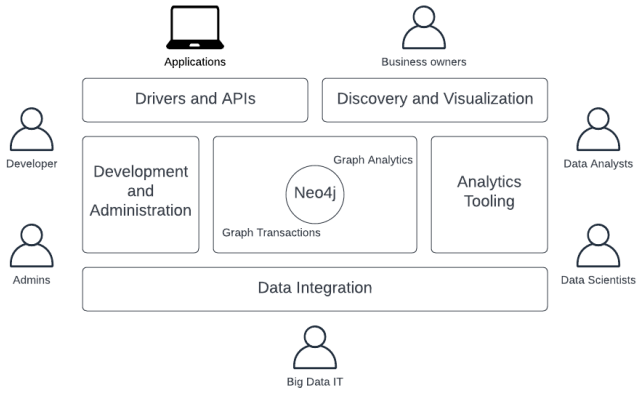


Figure 1: System architecture of Graph DB (Neo4j)

Neo4j is a popular Graph Database. The native property graph model is the data model that Neo4j uses. Neo4j supports full ACID (Atomicity, Consistency, Isolation, and Durability) rules. Cypher is a powerful declarative query language used by Neo4j. It displays graphs using ASCII graphics. Without the need for complicated queries like joins, Cypher may be used to establish and retrieve relationships between data. The expressive language, Cypher features sophisticated graph patterns and collection support. Under the hood, the cypher processing pipeline parses the query first, if it is not already in the cache, then goes through semantic verification and AST (Abstract Syntax Tree) rewriting, finds the cheapest execution plan (logical and physical) for every operation using the available planners, and finally executes the query.

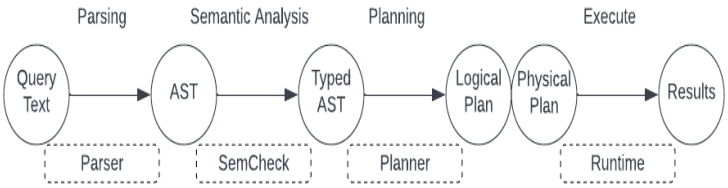


Figure 2: Query processing of cypher

V. CONCLUSIONS

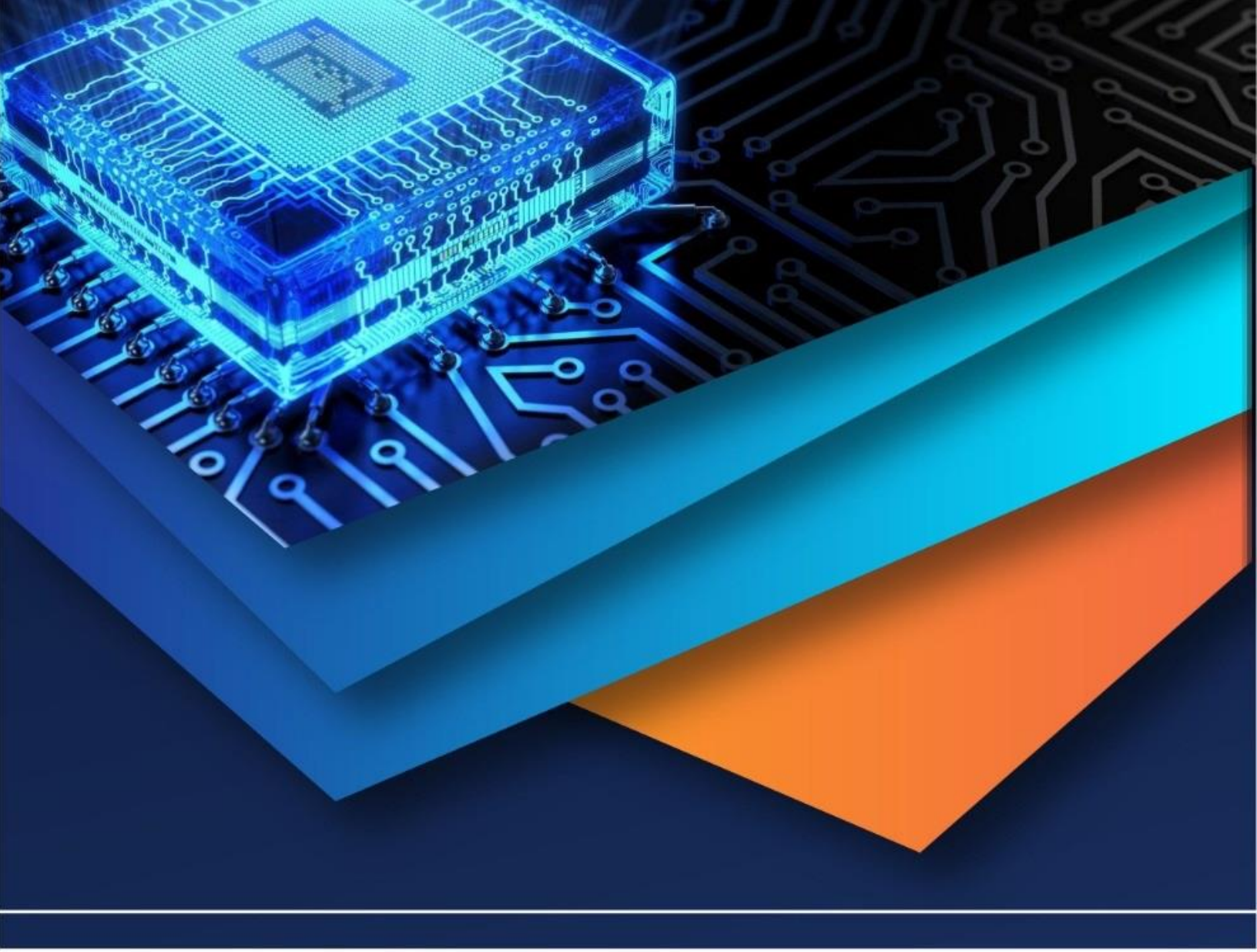
Impact analysis is necessary to find out the impact caused by a node or a data point of a system when it goes down or becomes non-functional. In a payment service environment, it becomes crucial to know the services that go down along with the failed service to reduce downtime. Due to the benefits of Graph DB such as scalability and better performance over other traditional databases implementation of impact analysis for payment platforms using graph DB proves to be beneficial causing convenience to the payment providers and the users.

VI. ACKNOWLEDGMENT

We would like to express our sincere gratitude to all the individuals who generously gave their time and shared their expertise to help us in the preparation of this survey paper. We are particularly grateful to Mrs. Deepa Sapkal for their valuable guidance and insightful comments, which greatly improved the quality of our work. We would also like to acknowledge Mr. Ganesh Pise who provided us with helpful resources, feedback, and suggestions. Finally, we extend our appreciation to our department, college, colleagues, friends, and family for their encouragement and support throughout this endeavor.

REFERENCES

[1] Sherba, Susanne & Anderson, Kenneth. (2003). A Framework for Managing Traceability Relationships between Requirements and Architectures.. 150-156.
[2] L. Badri, M. Badri and D. St-Yves, "Supporting predictive change impact analysis: a control call graph based technique," 12th Asia-Pacific Software Engineering Conference (APSEC'05), Taipei, Taiwan, 2005, pp. 9 pp.-, doi: 10.1109/APSEC.2005.100.
[3] A. De Lucia, F. Fasano and R. Oliveto, "Traceability management for impact analysis," 2008 Frontiers of Software Maintenance, 2008, pp. 21-30, doi: 10.1109/FOSM.2008.4659245.
[4] Gabriella Tóth, Péter Hegedűs, Árpád Beszédes, Tibor Gyimóthy, and Judit Jász. 2010. Comparison of different impact analysis methods and programmer's opinion: an empirical study. In Proceedings of the 8th International Conference on the Principles and Practice of Programming in Java (PPPJ '10). Association for Computing Machinery, New York, NY, USA, 109–118. https://doi.org/10.1145/1852761.1852777
[5] H. O. Ali, M. Z. A. Rozan and A. M. Sharif, "Identifying challenges of change impact analysis for software projects," 2012 International Conference on Innovation Management and Technology Research, 2012, pp. 407-411, doi: 10.1109/ICIMTR.2012.6236428.
[6] Wang, Y., Yang, J., Zhao, W. et al. Change impact analysis in service-based business processes. SOCA 6, 131–149 (2012). https://doi.org/10.1007/s11761-011-0093-8
[7] Miller, Justin J., "Graph Database Applications and Concepts with Neo4j" (2013). SAIS 2013Proceedings. 24. https://aisel.aisnet.org/sais2013/24
[8] G. Daniel, G. Sunye and J. Cabot, “UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases,” Springer International Publishing, 2016
[9] M. M. El Nemr and D. S. Elzanfaly, "A Framework for Advancing Change Impact Analysis in Software Development Using Graph Database," 2018 International Conference on Computer and Applications (ICCA), 2018, pp. 1-438, doi: 10.1109/COMAPP.2018.8460394



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)