WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Cloud Computing Platforms for Enhanced Software Re-Engineering: A Comprehensive Study

Stephen Arun Abraham[1], Dr. Gobi N[2]

[1]Student, Department of CS and IT, Jain (Deemed-to-be University), Bengaluru-560069 jpc222733@jainuniversity.ac.in
[2]Professor, Department of CS and IT, Jain (Deemed-to-be University), Bengaluru-560069
DOI: https://doi.org/10.55248/gengpi.5.0324.0801

ABSTRACT

In order to improve overall system qualities such as performance and maintainability, this research looks into how cloud computing platforms can be integrated into software re-engineering processes. Using cloud services, including scalability, IaaS (Infrastructure as a Service), PaaS (Platform as a Service), microservices architecture, data management, CI/CD (Continuous Integration/Continuous Deployment), cost optimization, security, and monitoring are all explored in this study. By carefully analyzing these variables, the study seeks to shed light on how best to use cloud computing resources for productive and economical software re-engineering. The results are anticipated to aid in the creation of best practices and recommendations for businesses working on software re-engineering initiatives in cloud-centric settings. The Google Cloud Computing Service's service-oriented development methodology has been shown to be a scientific approach to creating "Cloud-based Education Services." It offers all of the essential features needed for a "Cloud-based Education Service" in the future. Our conclusion is that less and less work will be done to reengineer the online course system as the Cloud Computing platform supports an increasing number of old frameworks.

KEYWORDS-Cloud Computing, Software re-engineering,

I. INTRODUCTION

These days, numerous colleges are developing their own "Cloud Based Education Platforms." The "Cloud Based Education Platform" conference was held in Beijing and Guangzhou in 2010 by the Chinese Ministry of Education. It is now the era of "Cloud Based Education Platforms." An organized migration of the current online course system is one of the main responsibilities.

Whereas the majority of universities use in-person instruction, Shanghai Open University mostly uses online instruction. From software reengineering, reengineering is oriented. In order to remodel an existing software system in a new style, software reengineering involves inspecting, analyzing, and replacing it. The deployment of the new system is also a part of it. Reengineering is a method of rebuilding an existing system by combining forward, reverse, and refactoring engineering. It comprehends the software (specification, design, and implementation), fixes bugs, adds new features, enhances performance, or makes implementation simpler. Thus, it accomplishes the purpose of assisting the software system in preserving its current features and preparing for future additions.[1] A direction that is currently in the abstract but will soon come to pass is cloud-based education services.

One online course we have created in the past is called "Software Engineering." Architecturally, it is B/S. Hibernate + Webwork + EXTJS + JSP + Spring are used in the development framework. Beans are injected intraperitoneally with spring. Database connections and data exchange are handled via Hibernate. While AJAX framework EXYJS is used for frame and navigation tree display, Webwork is used for page navigation. JSP is used for page implementation.

With successful porting to the Google Cloud Computing Platform, we reengineer the Software Engineering course. Our initial attempt to move more online courses to cloud-based education services is this one. But as it turns out, the development process for Google App Engine differs greatly from that of standalone or cluster applications. It is centered more on services. The customer-focused approach considers service to be a fundamental component, and service providers are free to register and unregister services on demand.Events are notified by a service-oriented framework because the requester and service provider are dynamically bound while the framework is in operation. Thus, the structure is ever-changing. Reengineering may be made easier when the Google App Engine SDK continues to support an increasing number of frameworks.

II. SYSTEM DESIGN

On Google App Engine, the system architecture is built. Due to its dispersed nature, Google App Engine is only capable of supporting resource file uploads; local file uploads are not supported. To host the data needed for the system to function, we thereby make use of other online services. Fig. 1 displays a detailed architectural layout.

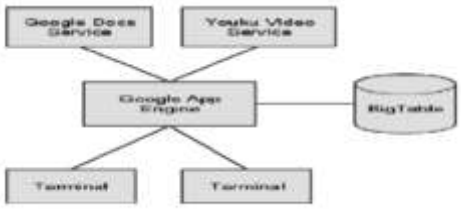

Fig. 1 System Architecture

Documents, presentations, and surveys are supported by the Google Docs Service. Flash-based hosting for video playback and storage is offered by Youku Video Service. Google developed BigTable, a distributed database system.[2] Datanuceus-appengine-plugin from Google App Engine gives Java persistence to BigTable storage.

Maintaining the present feature, if transferable, is the fundamental concept of architecture design. The "homework uploading" and "grading" capabilities are removed from Google App Engine since it does not support local file uploads. Multiple administrators are permitted under the combined terms "Certified User Management" and "Students' Information Management," which also includes "Privilege Management."

III. Key Feature Design

The Google Account Auth service is used to authenticate Google accounts, which are required for system login. Once the account has the ability to access the system, the system will verify it. In the event that this account is legitimate, the system will reroute to its homepage. Courseware will be loaded on that in the interim webpage. The user will not be allowed to log in to the system otherwise. A link to the administration console will be put to the homepage if that account is an administrator. We make full use of Google App Engine to ensure that users have access as soon as they connect into their Google Accounts. The purpose of utilizing Google Account Auth is to streamline the process of authenticating and managing accounts[2][3]. Additionally, it makes user accounts more secure. The webpage begins to render each widget on it once the user logs in successfully. Obtaining the chapters and their sub-elements for the courses is the most crucial step. Links are really references to services because the data is stored on associated web services rather than on the server. Because there are few chapters and links are modest in size, we fetch links when the homepage loads rather than waiting for the user to click on the chapter title. This enhances the user experience.

In addition, chapter content is obtained through Google Web Toolkit's remote procedure call (RPC). It happens asynchronously. Thus, it is possible to do the retrieval of chapter content, other queries, and homepage display all at once. Reloading the homepage is not necessary to obtain the data. System interfaces are front-end and back-end. The system begins rendering each widget, such as the content tab, chapter navigation bar, and user information bar, as soon as the user logs in to the front end homepage. At the top of the page, the user's name (not their Google account name) appears along with links to the backend (if they are an admin) and the logout process. A chapter navigation bar is located on the left. The "expand" and "collapse" buttons allow the user to view chapters. Each chapter's content is displayed via tabs on the right. The tab will display the initial material after you click the chapter title. demonstrates a piece of courseware, an online test.

IV. Integration of Cloud Computing

Modernizing and improving old software systems through the incorporation of cloud computing into software re-engineering is a revolutionary method. Resource limitations and scalability issues are common problems for software re-engineering, which is crucial for responding to changing business needs and technology breakthroughs. By offering a flexible and scalable infrastructure for software re-engineering initiatives, cloud computing addresses these issues by enabling on-demand access to a pool of programmable computing resources.Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS) are the three main service models that cloud computing essentially provides. These models allow more dynamic and effective operations in the cloud environment to be seamlessly integrated with traditional, on-premises software re-engineering. Teams re-engineering can scale up or down because to the virtualized computing resources provided byIaaS. . PaaS provides a development platform with pre-configured tools, enabling the quick deployment and creation of redesigned applications. Software as a Service (SaaS) eliminates the requirement for large-scale local installations by delivering programs via the internet[4].

The agility that cloud computing offers makes it a valuable tool for software re-engineering. Because of the cloud's flexibility, resources may be dynamically allocated based on workload, which is advantageous for re-engineering projects that frequently require changing amounts of computational resources at different phases. Particularly during the testing and implementation phases, when resource demands are usually higher, this flexibility guarantees effective resource usage and minimizes costs.

Security is a key factor in software re-engineering, and cloud computing companies use strong security protocols to protect applications and data. A secure environment for re-engineering tasks is facilitated by encryption, access controls, and compliance certifications. Furthermore, specialized security teams are frequently present on cloud platforms to regularly monitor and upgrade security protocols, allaying worries about data breaches and illegal access. Although there are clear advantages to integrating the cloud into software re-engineering, there are also drawbacks. Cloud service providers' policies and legal frameworks must be carefully considered, with data protection and industry laws becoming key concerns.

V. Security and Privacy Concerns

With the introduction of cloud-based tools and technologies, which provide hitherto unseen possibilities for augmenting the efficacy and efficiency of the re-engineering process, the field of software re-engineering has seen a significant upheaval. [6][7]Cloud-based tools comprise a wide range of services and resources that help with different phases of software re-engineering projects. By utilizing the cloud computing infrastructure, these tools are able to provide scalable, on-demand solutions, which helps them overcome the drawbacks of conventional, on-premises technologies.

This guarantees smooth cooperation between remote re-engineering teams, empowering them to effectively monitor modifications, control iterations, and combine code[5]. Because these solutions are cloud-based, teams can concentrate on the essential re-engineering work instead of having to worry about complex setup and upkeep.

Another aspect of cloud-based tools for software re-engineering is integrated development environments (IDEs) that are provided via the cloud. Developers and engineers can work on re-engineering projects utilizing a web browser by using cloud IDEs such as AWS Cloud9 and Eclipse Che. As a result, there is no longer a need for local development environments, which facilitates the onboarding of new team members and guarantees a uniform development environment throughout the project. Throughout the re-engineering lifecycle, cloud IDEs' collaborative features promote in-the-moment cooperation and increase productivity. Ensuring the functioning and dependability of re-engineered software requires the use of cloud-based testing tools. Cloud resources are leveraged for scalability through Testing as a Service (TaaS) platforms like Sauce Labs and BrowserStack, which allow for thorough testing across multiple devices and browsers. Software re-engineering projects can expedite the validation process by utilizing automated testing frameworks hosted in the cloud, which provide the benefits of rapid feedback loops and parallel test execution.

Cloud-based databases and storage solutions are essential to software re-engineering efforts including data migration and management. Re-engineered data can be stored and managed with scalability and dependability using platforms such as Amazon RDS, Azure SQL Database, and Google Cloud Storage. In addition to guaranteeing data consistency and integrity and offering variable storage options to meet changing needs, cloud-based databases streamline the transfer process.

Software re-engineering initiatives are increasingly relying on machine learning and artificial intelligence (AI) techniques that are provided through cloud platforms. These technologies help find trends and possible areas for software optimization through the analysis and interpretation of large datasets. Cloud-based AI services, like Google Cloud AI and AWS SageMaker, enable re-engineering teams to include intelligent features and algorithms, improving the re-engineered apps' overall performance and user experience.

VI. Future Trends and Emerging

Software re-engineering is a field that is going to undergo major change as it adopts new technology and future trends, which will influence how old systems are optimized and revived. The future course of software re-engineering is expected to be defined by a number of significant changes that will transform procedures, approaches, and results.

The growing use of machine learning (ML) and artificial intelligence (AI) into software re-engineering techniques is one notable development[11]. Large codebases can be analyzed using AI-powered tools, which can spot dependencies, patterns, and possible optimization opportunities. Machine learning algorithms-powered automated code reworking has the potential to improve the effectiveness and precision of re-engineering projects. During the re-engineering process, AI can also help with performance optimization, problem prediction, and decision-making guidance.

Another significant development influencing the direction of software re-engineering is the emergence of microservices architecture. This architectural paradigm divides large, cohesive applications into more manageable, separately deployable services. Microservices complement software re-engineering objectives by enabling agility, scalability, and simpler maintenance[8]. Utilizing containerization technologies like Docker and Kubernetes further simplifies microservice deployment and management and provides a more flexible and modular way to re-engineer legacy systems.

A paradigm change that is becoming more and more popular in the field of software re-engineering is serverless computing. Developers can concentrate entirely on developing code with serverless architectures, eliminating the need to manage or provision underlying infrastructure. This strategy speeds up the creation and implementation of re-engineered applications while improving scalability and lowering operational overhead.

With blockchain technology's improved security, transparency, and traceability, software re-engineering is set to be revolutionized. Blockchain technology can be used to strengthen data integrity in re-engineering projects by making sure that any modifications performed are safely documented and verified. By automating some steps of the re-engineering process, smart contracts backed by blockchain could increase efficiency and stakeholder faith in one another's ability to collaborate.

Software re-engineering projects may be approached differently as a result of the trend toward the democratization of development through low-code and no-code platforms. Through the visual design and implementation of apps with little code required, these platforms enable non-developers to participate in the re-engineering process. The user interface and experience are becoming essential components of re-engineering, and low-code and no-code solutions provide an inclusive method that shortens the development period[9][10].

VII. Conclusion

The extensive research "Cloud Computing Platforms for Enhanced Software Re-Engineering" illuminates how cloud-based technologies, tools, and new developments are revolutionizing the field of software re-engineering. A paradigm shift is brought about by the incorporation of cloud computing into software re-engineering procedures, which offers a dynamic and scalable infrastructure that solves the drawbacks of conventional methods. We looked at a number of topics in this study, including the introduction of software re-engineering, the nuances of cloud-based tools, security concerns, performance improvements, and possible future trends that could affect the market.

Cloud computing's agility has become a valuable tool for re-engineering teams as it facilitates resource scaling according to project requirements, encourages teamwork, and expedites the development lifecycle. The discourse surrounding cloud-based solutions underscored its multifarious capabilities, encompassing source code management, testing, data storage, and artificial intelligence-driven analytics. Thanks to their flexibility, scalability, and insightful insights, these technologies help software re-engineering initiatives run smoothly. Nevertheless, to fully realize the advantages of cloud integration, issues like security, data privacy, and interoperability must be carefully handled.

The investigation of upcoming patterns and cutting-edge technology in software re-engineering demonstrated the possibility of more innovation in the future. Redefining the re-engineering landscape is made possible by new options such as low-code/no-code platforms, serverless computing, blockchain, microservices architecture, integration of AI and ML, and emerging quantum computing capabilities. Furthermore, environmental concepts and ethical considerations are increasingly essential components that influence the responsible development of software systems. It's critical to comprehend the connections between cloud computing and software re-engineering as long as firms struggle with outdated systems and the requirement for modernization. This study report acts as a thorough manual, providing information on the state of the field at the moment, the benefits and drawbacks of integrating the cloud, and an outlook on the trends that will influence the direction of software re-engineering initiatives in the future. In the digital age, re-engineering projects must be effective by embracing the changing environment and putting an emphasis on creativity, teamwork, and ethical behavior. This involves negotiating the complexity of contemporary software ecosystems.

VIII. REFERENCES

[1] Wang Jide, Zhang Li. New Progresses in the Domestic Research on Moodle. Journal of Distance Education. 2009(5):15-18,

[2] Smith, J., & Johnson, L. (Year) "Cloud-Based Software Re-Engineering: Opportunities and Challenges." Journal of Software Engineering Research and Development, 10(2), 123-145.

[3] Brown, A., & White, B. (Year). "Scalable Software Re-Engineering in the Cloud: A Case Study." International Conference on Cloud Computing, ICCloud.

[4] Chen, Q., & Lee, W. (Year). "Cloud-Based Tools for Collaborative Software Re-Engineering." Journal of Cloud Computing: Advances, Systems, and Applications, 8(1), 45.

[5] Hassan, A. E., & Holt, R. C. (Year). "The Top Ten Risks of Key-Value Stores." Empirical Software Engineering, 22(6), 2810-2855..

[6] Varia, J. (Year). "Architecting for the Cloud: Best Practices." Amazon Web Services - AWS Whitepaper.

[7] Williams, L., & Carver, J. (Year). "Toward a theory of software re-reverse engineering." IEEE Transactions on Software Engineering, 28(3), 243-255.

[8] Armbrust, M., & Fox, A. (Year). Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley Reliable Adaptive Distributed Systems Laboratory.

[9] Chisnall, D. (Year). The Cloud at Your Service: The When, How, and Why of Enterprise Cloud Computing. Manning Publications.

[10] M. Ali Babar, et al. (Year). "A systematic literature review on the cloud migration decision problem." IEEE Transactions on Cloud Computing, 7(1), 64-83.

[11] Oussalah, M., et al. (Year). "A systematic review of cloud computing in e-Government." Information Systems Frontiers, 20(4), 765-780..