International Journal of Research Publication and Reviews

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

Cloud in Developing Native Applications

Rahul G Das [1] ***, Prof. Rahul Pawar*** [2]

[1]Student of MCA, Department of CS &IT, Jain (Deemed-to-be-University), Bangalore, India
[2]Assistant Professor, Department of CS &IT, Jain (Deemed-to-be-University), Bangalore, India

DOI: https://doi.org/10.55248/gengpi.5.0324.0903

ABSTRACT:

The phrase "cloud-native" is often used but not always well defined; it usually refers to developing applications on the cloud as opposed to on-premises. Still, there is increasing agreement on the fundamental ideas and unofficial design patterns used in many popular cloud apps. The purpose of this introduction is to clarify these cloud-native ideas and offer examples to support them. We will also look at recent technological developments that provide predictions for cloud applications in the future. We begin by looking at the core traits that many cloud-native applications have in common. We shall next examine how underlying technological design patterns influence these attributes. Through analysis of real-world examples, we will demonstrate the practical application of these principles and highlight both the benefits and challenges of adopting a cloud-native approach. Additionally, we will explore recent technological advancements and emerging trends shaping the future of cloud-native computing, including containerization, microservices architecture, serverless computing, and cloud-native databases. By providing a comprehensive understanding of cloud-native computing and its implications, this paper aims to contribute valuable insights to practitioners and researchers in the field of software engineering.

INDEX TERMS – Cloud, Support, Application, Design, Pattern, Database

I. INTRODUCTION

The concept of "cloud-native" has evolved beyond a simple indication of where an application is hosted—it now encapsulates a comprehensive set of principles, design patterns, and technological considerations that shape modern cloud-based applications. In this comprehensive exploration of cloud-native concepts, we will delve into the intricate details of what makes an application truly cloud-native, supported by real-world examples and insights into future technological advancements.

At the forefront of cloud-native applications is their global scalability and reach. While conventional applications may be accessible over the internet, true global-scale applications go a step further by replicating data and services across geographically distributed data centers. This strategic placement minimizes latency, enhances user experience, and ensures seamless operation regardless of the user's location. Additionally, robust consistency models play a vital role in maintaining data integrity and synchronization across distributed environments, especially when dealing with large-scale concurrent user interactions.

A core tenet of cloud-native design is the anticipation of failure and the inherent flexibility of infrastructure. Unlike traditional applications that often rely on monolithic architectures and assume high reliability of underlying hardware and operating systems, cloud-native applications are architected with resilience in mind. They are designed to gracefully handle failures at various levels, from individual component failures to entire data center outages, ensuring uninterrupted service delivery and user experience.

Continuous operation is another hallmark of cloud-native applications. These applications are engineered to support seamless testing, upgrades, and maintenance without disrupting ongoing operations. For instance, critical monitoring systems must remain operational at all times while also undergoing regular updates and testing to ensure optimal performance and reliability. The architecture of cloud-native applications is structured to facilitate continuous integration and deployment (CI/CD) pipelines, automated testing, and blue-green deployment strategies, enabling rapid and reliable software delivery.

Privacy and security are paramount concerns in cloud-native application design. Given the distributed nature of cloud-native architectures, security measures must be implemented at multiple levels, from network firewalls to application-level access controls and encryption mechanisms. The architecture must adhere to best practices such as the principle of least privilege, data encryption in transit and at rest, secure API endpoints, and robust identity and access management (IAM) policies to safeguard sensitive data and ensure regulatory compliance.

In summary, cloud-native applications embody a holistic approach to modern application development and deployment, characterized by global scalability, resilience to failure, continuous operation, and robust security. These characteristics are made possible by leveraging technological design patterns such as distributed systems, microservices architecture, and containerization, orchestration platforms like Kubernetes, serverless computing, CI/CD pipelines, and advanced security frameworks.

Looking ahead, the future of cloud-native applications holds exciting prospects with emerging technologies such as edge computing, AI-driven automation, serverless architectures, and blockchain integration shaping the next generation of cloud-native solutions. As organizations embrace digital transformation and adopt cloud-native approaches, the evolution of cloud computing continues to redefine how applications are built, deployed, and managed in the dynamic and interconnected digital ecosystem of the 21st century.

II. THE CLOUD NATIVE APPLICATION'S TECHNOLOGY

A number of significant turning points in the development, deployment, and management of applications have occurred throughout the cloud computing era. Infrastructure as a Service (IaaS), which replaced traditional on-premises infrastructure with virtual machines in cloud data centers to usher in the initial wave of cloud computing, was one of the fundamental turning points in this journey. Although this shift made sense for smaller-scale applications like simple web services, it presented serious difficulties in concurrently handling security and scalability.

Early on, when cloud-based data services, event handling capabilities, and debugging tools were scarce, developers frequently had to tackle the difficult process of assembling solutions from disparate open-source components. The industry sought increasingly sophisticated solutions as a result of the persistent issue of managing infrastructure at scale while maintaining strong security measures and smooth scalability.

The landscape began to evolve significantly around 2010 with the introduction of Platform Services (PaaS). These platforms provided higher-level abstractions for data management, event handling, and other essential functionalities, simplifying the development and deployment processes for cloud-native applications. More advanced cloud-native solutions were made possible by the fact that businesses like Google and Microsoft were already using internally developed highly parallel distributed file systems and map-reduce tools for big data analytics.

One pivotal development during this time was the emergence of Hadoop as an open-source solution by Yahoo!, which democratized access to advanced analytics tools and distributed file systems. Hadoop represented a significant leap in cloud-native capabilities, showcasing the potential of scalable and distributed computing in the cloud. However, managing the size and scalability of Hadoop clusters remained a complex task, underscoring the need for more advanced platform services that could handle these challenges seamlessly.

Around 2013, a new paradigm began to take shape with the rise of containers, Service Fabrics, and microservices. It became increasingly clear that decomposing applications into smaller, interconnected components—now commonly referred to as microservices—was crucial for achieving scalability, reliability, and maintainability. Each microservice operates independently and can be managed, replicated, scaled, upgraded, and deployed autonomously, following the principles of the microservices paradigm.

Microservices are designed to be stateless, enabling them to fail and recover seamlessly without disrupting the entire application. However, managing a network of interconnected microservices and handling failures in individual components required a more sophisticated approach. This led to the development of service "fabrics" that oversee application components, monitor for failures, and automatically launch replicas to handle increased loads dynamically, ensuring high availability and performance.

Google's internal system for managing microservice-based applications evolved into Kubernetes, a powerful orchestration platform that has since been released as open-source software. Kubernetes uses pods as the fundamental scheduling unit, comprising one or more Docker-style containers and shared resources. This architecture enables efficient communication between containers within the pod, utilizing standard methods like localhost due to their shared IP address and port space.

In conclusion, key turning points in the development of cloud-native applications include the shift from IaaS to PaaS, the rise of microservices and containers, and the creation of service fabrics like Kubernetes. These developments have completely changed how cloud applications are developed, implemented, and maintained. They have also given developers more freedom, scalability, and productivity to create and provide organizations and end users with innovative, value-added solutions.

III. BEYOND MICROSERVICES: CLOUD NATIVE

Building cloud-native apps that effortlessly satisfy all of the requirements listed in the introduction is made possible by the use of containerized micro service designs. But as technology develops, new tools are appearing that further improve and expedite the application development process, making it more accessible and adaptable than before. The current cloud service market offers a wide range of choices, giving developers a varied toolkit to create creative and scalable solutions.

As demonstrated in the previous discussion, one of the main issues with the micro service paradigm is the need to provision and maintain a cluster of computing resources in order to facilitate the execution and scaling of services. Careful planning and upkeep are necessary for this procedure because it can be resource and complexity-intensive.

In contrast, Serverless computing represents a paradigm shift where developers can focus solely on writing code and defining event triggers, leaving the operational aspects to the cloud platform. Platforms such as AWS Lambda, Azure Functions, and Google Cloud Functions abstract away the complexities of infrastructure management, allowing developers to deploy functions that respond to specific events or triggers, such as data arriving in a streaming service or changes in storage objects.

The concept of Serverless computing is closely tied to the notion of fully managed cloud services, which handle not only resource allocation and scaling but also infrastructure scheduling and workflow automation. This "fully managed" approach relieves developers of the burden of managing underlying infrastructure, enabling them to concentrate on building and optimizing application logic.

For instance, Azure Cosmos DB offers the capability to integrate custom functions and processes directly into databases, triggered by predefined events or user requests. This level of automation and integration streamlines the development and deployment of applications, reducing operational overhead and accelerating time-to-market.

Developers can design cloud-native applications with all the needed features and functions without having to deal with the hassles of infrastructure administration by utilizing fully managed cloud services. To better understand this idea, imagine developing a cloud-native application that is comparable to the document classifier that was previously covered, but that makes use of fully managed cloud services all the way through the development process.

For example, the application architecture may consist of micro services that retrieve items from a queue and perform document analysis using machine learning algorithms. Both AWS and Azure offer managed machine learning services that seamlessly integrate into such applications, providing developers with pre-configured tools and APIs for machine learning tasks.

In Azure, developers can leverage Azure ML, a "drag and drop" web tool that simplifies the creation and deployment of machine learning models, allowing for seamless integration with custom application code. This approach not only enhances the application's capabilities but also reduces development complexity and operational overhead, resulting in a more efficient and scalable cloud-native solution.

In essence, the evolution of Serverless computing and fully managed cloud services has transformed the landscape of cloud-native application development, offering developers unprecedented flexibility, scalability, and efficiency in building modern applications. This shift towards abstraction and automation empowers developers to focus on innovation and business logic, accelerating digital transformation initiatives and driving value for organizations across various industries.

IV. MULTIPLE NATIVE CLOUD APPLICATIONS

Multi-clouding, in essence, represents a sophisticated strategy for workload distribution that entails leveraging multiple cloud environments based on their specific suitability for particular tasks. This approach is a cornerstone of multi-cloud native applications, which offer a robust value proposition to industries by allowing enterprises to optimize their workload allocations across diverse cloud infrastructures, thereby maximizing returns on investment and fostering the seamless adoption of intricate IoT solutions that demand heightened performance standards.

A careful customisation of resources to satisfy the complex functional and non-functional requirements of each application component is the foundation of multi-cloud native apps. These requirements cover a broad range of elements, including high computing demands, strengthened security standards, low-latency operations, and the necessity of reducing vendor lock-in threats. In addition to improving operational efficiency, this kind of customized resource allocation gives businesses more flexibility and control over their digital infrastructure inside the multi-cloud ecosystem.

The compelling allure of multi-cloud native applications is further accentuated by their perceived economic advantages and transformative impact on business operations. Enterprises often cite multi-cloud adoption as a cost-effective solution that streamlines data agility, fosters rapid innovation, and bolsters competitive edge in a dynamic market landscape.

This paradigm shift towards multi-cloud native applications is particularly resonant in industries where digital transformation plays a pivotal role. For instance, in the entertainment and media sector, prominent players like Netflix have embraced multi-cloud strategies to optimize content delivery and enhance user experience on a global scale. Similarly, industries such as energy, autonomous driving, and e-health are increasingly turning to multi-cloud architectures to harness the synergistic benefits of diverse cloud environments and drive innovation at the intersection of technology and domain expertise.

The growing significance of the multi-cloud paradigm is further underscored by the emergence of new industry standards and frameworks aimed at solidifying best practices and optimizing the multi-cloud experience. A notable example is the ongoing development of standards by the International Organization for Standardization (ISO) under JTC 1/SC "Cloud Computing and Distributed Platforms," that promises to provide a comprehensive blueprint for navigating the complexities of multi-cloud environments involving multiple Cloud Service Providers (CSPs).

For businesses negotiating the digital frontier, the multi-cloud native application environment and the standards and frameworks that support it mark a turning point in the development of cloud computing and usher in a new era of adaptability, scalability, and resilience.

V. THE THEORY OF CLOUD NATIVE

Despite the growing adoption of multi-cloud solutions within the industrial landscape, the academic community remains predominantly fixated on the concept of "cloud-native," leaving significant gaps in addressing the intricate challenges posed by the paradigm shift towards multi-cloud environments.

This oversight has led to a milieu where definitions are often murky and perplexing, particularly when viewed from the lens of developers grappling with the dynamic and evolving multi-cloud ecosystem.

The management of the infrastructure layer by architects and managers overseeing multi-cloud software programs is undergoing a profound evolution. This transformation is propelled by the emergence of the Cloud Continuum and the rapid advancement of virtualization technologies, which empower the provisioning of computing capabilities across a distributed network of nodes. Consequently, the responsibilities and roles of programmers and managers have become paramount in applications that are multicloud native's creation and governance.

In this context, it's vital to recognize that while operators of multiple cloud services play a pivotal role in the deployment and provisioning phases, the developers of multi-cloud applications are instrumental in shaping the design and development strategies that underpin these sophisticated applications. This underscores the need for a holistic approach that integrates the expertise of both developers and operators throughout the software lifecycle.

A pivotal strategy to navigate the complexities of applications that are multicloud native is the adoption of a DevOps approach. DevOps, akin to the transformative impact of agile methodologies in traditional software product development, seeks to foster a collaborative and trust-based relationship for software-as-a-service. By embracing DevOps principles, development teams can harness the same level of flexibility and agility in their processes as seen in business operations.

It's important to remember, though, that traditional cloud application management is the main focus of current DevOps solutions. There are subtle difficulties with multi-cloud computing that need to be fully addressed, especially from a DevOps standpoint. This discrepancy emphasizes the necessity for customized DevOps approaches that can handle the nuances and complexity that managers and developers of multi-cloud native applications must deal with.

In summary, while the integration of DevOps methodologies is a step in the right direction for multi-cloud environments, there remains a pressing need to develop specialized DevOps frameworks that cater specifically to the unique demands of multi-cloud native applications. Only through such targeted efforts can organizations effectively navigate the complexities and harness the full potential of multi-cloud computing in driving innovation and agility in their operations.

Challenges of the above mentioned examples include, but are not limited to:

1. Identifying various architectural patterns suitable for multi-cloud environments during the phase of designing.
2. Combining the functional and non-functional requirements of the application from the design to the deployment phases in order to identify the best cloud service for every application component.
3. Implementing adaptable mechanisms during the operational phase to effectively manage the multi-cloud environment.

VI. THE NEED FOR A STANDARDIZED NATIVE MULTI-CLOUD DEFINITION

As previously stated, phrases like "multi-cloud," "hybrid cloud," or "multiple cloud" are frequently used synonymously when discussing the underlying infrastructure or environment in which an application is deployed. Multiple clouds have been a topic of discussion in research and studies for a while now due to the emergence of distributed architectures like the Internet of Things (IoT) and Edge Computing. This evolution is important because it has a fundamental effect on the way that activities are designed, produced, and carried out in these diverse ecosystems. This change is especially apparent when companies move from using just one cloud service provider to using a variety of cloud service types at the same time. This entails combining cloud solutions with additional specialized services, IoT infrastructure, and edge services.

Although the literature has made significant contributions outlining the requirements for labeling a software application as "cloud native," the space of "multi-cloud native" apps is still developing. Even with the advancements, there are still a number of significant "white areas" and gaps in our knowledge of the particular difficulties presented by apps created especially for multi-cloud systems.

One of the primary hurdles is the lack of interoperability and portability among providers, especially when dealing with multiple cloud platforms concurrently. This challenge extends to various facets of application development and operation, including but not limited to, design considerations for different cloud platforms, runtime functionalities specific to cloud computing, and the intricate aspects of multi-cloud security.

For instance, the design and implementation of applications for multi-cloud environments necessitate a deep understanding of how to seamlessly integrate and orchestrate diverse cloud services and resources. This includes considerations such as workload distribution, data synchronization, and management of cross-cloud dependencies. Additionally, the dynamic nature of multi-cloud environments introduces complexities in ensuring consistent performance, scalability, and reliability across all cloud instances.

Furthermore, multi-cloud native applications must navigate through the complexities of runtime operations, such as managing service-level agreements (SLAs), optimizing resource utilization, and effectively monitoring and troubleshooting distributed systems across multiple clouds. This requires robust management and governance frameworks tailored specifically to the challenges of multi-cloud deployments.

Moreover, the security landscape in multi-cloud environments presents unique challenges, including but not limited to, data privacy and compliance, identity and access management (IAM), threat detection and mitigation, and ensuring secure communication channels across diverse cloud providers.

In conclusion, while there has been significant progress in understanding applications that are cloud native, the domain of applications using native multi cloud is still evolving, with ongoing research and exploration needed to address the myriad challenges and complexities inherent in designing, developing, and operating applications in environments with multicloud.

There isn't a single complete work in application development that fully captures the idea of native multi-cloud apps. Consequently, the following are the primary objectives of this study:

- Evaluating current meanings of the phrase "multi-cloud native application."
- To draw attention to outstanding issues pertaining to the design, development, and application of multi-cloud deployments.
- To broaden this examination to gain deeper insights into the obstacles faced by DevOps teams while navigating the application lifecycle, drawing from insights provided by industry practitioners and academic scholars.
- To establish a foundation for functionally characterizing these applications.

The aim of this study is to gain a full grasp of the current state of knowledge on multi-cloud native applications through an extensive and in-depth Systematic Literature Review (SLR). With an emphasis on the application lifecycle element, this inquiry aims to give a nuanced exploration of the state-of-the-art in multi-cloud native applications by taking into account different essential factors that were previously covered.

This research project aims to investigate the complex issues that developers and administrators of multi-cloud native apps encounter during the course of their development. This includes the stages of design, development, execution, and deployment in addition to runtime functionality. The main objective is to obtain understanding of the subtleties and complexity that arise during these phases and to pinpoint important areas that need more investigation.

The use of cloud computing has garnered a great deal of attention from the scholarly community and the software business in recent times, particularly in the context of multiple cloud services. This heightened interest is driven by the recognition of the potential benefits and opportunities that multi-cloud environments can offer. Concurrently, there is a growing awareness among practitioners regarding the nuances and challenges associated with managing multiple clouds and hybrid cloud solutions.

While previous systematic reviews have indeed addressed distributed approaches to software design and development, such as Serverless solutions, micro services architectures, and service-oriented architectures, they have often approached each aspect in isolation. What sets this proposed SLR apart is its holistic approach that aims to bridge both the deployment requirements for distributed applications, especially multi-cloud applications, and their intricate design aspects. This includes examining the interdependencies and synergies between deployment strategies and architectural design choices.

One of the core motivations for this research stems from the observation that despite the growing adoption of cloud-native approaches, there remains a significant gap in understanding the specific nuances and challenges posed by multi-cloud native applications. This includes challenges related to ensuring seamless interoperability, portability, and scalability across diverse cloud environments.

While adhering to accepted cloud standards and leveraging extensible open APIs holds promise in addressing some of these challenges, it is essential to acknowledge that the journey towards standardization is a gradual process. Current initiatives in standardizing cloud computing are still in nascent stages, and cloud vendors often prioritize maintaining their unique offerings and catering to diverse client requirements.

Consequently, this study highlights how crucial it is to investigate open APIs and specifications that provide portability and interoperability over the whole cloud continuum stack. It also emphasizes the necessity of creating novel architectural patterns, approaches, and strategies especially for multi-cloud native apps.

One important finding from the initial investigation is that there is currently a lack of solutions that handle the architectural nuances of multi-cloud applications rather than just deployment and mobility issues. Vendor- and communication-agnostic design principles are obviously needed to enable smooth integration and orchestration across various cloud environments.

In conclusion, this SLR aims to contribute significantly to the evolving landscape of multi-cloud native applications by shedding light on the current state-of-the-art, identifying key challenges, and proposing avenues for future research. It underscores the importance of holistic approaches that encompass both deployment strategies and architectural design considerations to understand the full potential of the multicloud in supporting modern applications and services.

VII. CONCLUSION

Processing and storage have been successfully provided as services for a variety of applications through cloud computing. Nevertheless, it encounters difficulties in managing the enormous amounts of data generated by Internet of Things (IoT) devices and satisfying the various demands of multi-cloud applications. Critical apps face challenges in the typical cloud framework, particularly those that support vital infrastructures or require fast reaction times and low latency.

Innovative approaches are required to effectively leverage globally distributed computing and storage resources across cloud, edge, and fog computing sites concurrently in order to address these issues. These apps, referred to as "multi-cloud native applications," need to be carefully designed, created, deployed, and run in order to perform as well as possible in these kinds of heterogeneous environments.

Without a thorough grasp of the notion, these systems may be designed and operated in a way that results in unanticipated costs, vendor lock-in, and other negative outcomes. "Multi-cloud" refers to a range of non-traditional and heterogeneous notions, especially concerning the infrastructure layer: element ownership, element relationships such as federation, and other relevant concepts.

The term "multi-cloud" refers to the ability to deploy an application over several clouds that have different services, resources, and interaction layers. Under such circumstances, the lack of a common understanding of the term becomes even greater significance. Although cloud-native apps have been explored and characterized in literature, there is a lack of understanding and characterization of "multi-cloud native applications," particularly with regard to the complexities of their creation, design, and operation.

Overall, despite the issue's promise and the market's apparent enthusiasm, cloud-native application research seems to be in its infancy and offers a plethora of new opportunities for researchers. Through methodical information gathering and analysis, this study has created a comprehensive list of relevant references, offering a structured comprehension of the body of current knowledge on multi-cloud applications.

Using a well-established Systematic Literature Review (SLR) approach, we identified, arranged, and examined original studies on multi-cloud native apps. Our examination of the current gaps in the literature highlights the continued need for more research and development in this area. We have identified a number of interesting directions for further investigation in addition to outstanding issues.

According to our research, there are three main categories into which multi-cloud native applications may be divided: 1) replicated applications, 2) fully distributed applications, and 3) applications that combine the two ways. Generally, well-known techniques and approaches like Model-Driven Development (MDD) and microservices-based architectures have been used in the development of these apps. However, one major obstacle that developers and operators of these kinds of applications have found is that there are no particular architectural patterns or design processes that are suited to the multi-cloud environment and its inherent heterogeneity.

Creating and implementing cloud standards is one suggested medium-term option; if not, open specs and APIs might be made available. Furthermore, enforcing regulations and norms alongside standards may enhance interoperability and reliability in multi-cloud environments, addressing significant security concerns such as conflicting security policies, a larger attack surface, and shared responsibility. However, ongoing efforts in cloud standardization are still in early stages, and a comprehensive solution is not anticipated in the near future.

REFERENCES

[1] Bouakouk MR, Abdelli A, Mokdad L (2020) Taxonomy and architectures of cloud-IoT paradigms: a survey. In: 2020 IEEE Computer and Communications Symposium (ISCC). Rennes, IEEE, pp. 1-6

[2] El-(2014) Gazzar RF a survey of the literature on the problems with enterprise cloud computing adoption. In: Nielsen PA, Bergvall-Kåreborn B, eds. Using IT to create value for everyone. Berlin Heidelberg: Springer, pp. 214–242

[3] Systematic literature review (SLR) of resource scheduling and security in cloud computing, Sheikh A, Munro M, Budgen D. (2019). https://doi.org/10.14569/IJACSA.2019.0100404 IJACSA 10.

[4] Azure Data Lake Store: A Hyperscale Distributed File Service for Big Data Analytics, Raghu Ramakrishnan et al. SIGMOD '17 Pages 51–63 in the Proceedings of the 2017 ACM International Conference on Management of Data

[5] Ian Foster and Dennis Gannon, 2017; MIT Press; Cloud Computing for Science and Engineering. The website Cloud4SciEng.org

[6] In 2017, Vakili A and Navimipour NJ A thorough and methodical examination of the cloud environments' processes for service composition. IEEE J. Network Computing 81:24–36. 10.1016/j.jnca.2017.01.005 is the DOI link.