JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DEMYSTIFYING RESEARCH CHALLENGES IN MICROSERVICES ARCHITECTURE

Chellammal Surianarayanan
Assistant Professor of Computer Science Centre for Distance and Online EducationBharathidasan Univesity, Tiruchirappalli, India

Abstract

Microservices Architecture is a service-based architecture for developing enterprise applications. This architecture becomes important as modern applications involve continuous delivery of the application in incremental fashion with frequent deployment. With MSA, a given enterprise application is broken down into independent microservices which are capable of individual deployment. Another key feature of MSA is that each microservice can be developed using appropriate tools and technologies irrespective of other services. This enhances the quality of services. The objective of this paper is to provide the fundamental concepts of MSA and highlighting different research challenges in microservices architecture

Keywords: Microservices, research challenges of MSA, discovery, QoS, testing, data consistency, communication overhead.

I INTRODUCTION

Microservices architecture (MSA) is an architectural style [1-2] used to design and develop enterprise application. With conventional development of enterprise applications, when the size of the application grows beyond a level, the maintenance of the application becomes a challenge. In addition, whenever a change needs to be applied to an existing application, the entire application needs to be redeployed which is complex and time consuming. The need for MSA can well be realized as follows. Very frequently business applications need to be dynamic in the sense that it has to satisfy the changinguser requirements. That is the requirements of business applications often tend to vary dynamically. In this situation, it is practically not feasible to adopt the conventional Software Development Life Cycle(SDLC) methods like waterfall methods where the requirements of applications are static. Nowadays business applications are adopting agile software development process where the applicationsare delivered in incremental fashion. As in Fig. 1, modern applications are extensively developed using agile model where the software product is developed in incremental fashion with user feedback [3]. Forapplications of this kind, developing the product with monolithic architectural style is complex and tedious. The complexity can be understood as follows. With monolithic architecture, the size of the project grows with respect to growing features. Many developers would be working on the project. Also, the entire project will be deployed as a single archive file. With this style, even if a small change needs to be implemented, the entire project as a whole will be deployed. Not only that, when many developers are working on different parts of the project, the developers have to wait for one another even if a developer complete his change as the monolithic style involves the redeployment of entire project. The above issues with conventional monolithic style of application development can be resolved with MSA.In addition, cloud native applications typically adopt the style of MSA [4].

II RESEARCH CHALLENGES

Service discovery

In MSA microservices are typically described and exposed through their interface descriptions.There are different standards such as REST API, Thrift API, Open API specification, etc. Service discovery is a challenging task due the availability of numerous services are available and matching involves picking up the right service according to the functional capabilities, input parameters, output parameters, preconditions, outcomes, etc. In addition, the service may be syntactically similar, but semantically different. Also, the services look syntactically different but semantically the same. Another aspect is that, incorporating semantics while describing the capability of microservices will make most of the business process invocation automatic.

Service Composition

Like microservices discovery, microservices composition is a very challenging task which can be realized in two ways with and without a central coordinator as discussed in [5] [6].

Service communication overhead

In MSA the services are of micro size and to implement real business applications, they have be loosely integrated through messages. Here, communication and interactions among themselves have led to major computing and processing overhead. This results in networks having high bandwidth and other resources.

Testing and debugging of microservices

As microservices are being deployed in different distributed nodes in cloud computing environment, testing and debugging of microservices is a challenge. There are no tools available. Since each service can be implemented using its own technologies debugging also become difficult.

Quality of Service

An important research aspect in regard to microservices is how to handle the Quality of Services(QoS) for microservices [7]. Basically, QoS refers to non-functional attributes of services including, availability, security, scalability, latency, response time, reliability, usability, etc. In generally, microservices are typically deployed in cloud environment where resources are provided to microservices in a dynamic provisioning method. Cloud service providers provide the resources to all microservices according to the demand arises. Here, more research can be done so that how to determinethe actual QoS requirements of services is an open issue.

Implementation of data consistency & two-way commit protocol

Secondly, in MSA, each service has its own database. If this is the case, coordinating the database operations among individual services becomes complex. Here implementing the basic properties, namely Atomicity, Consistency, Isolation and Durability(ACID) properties becomes complex. More specifically, in microservices, the data is not always consistent but it eventually becomes consistent

Operational Complexity

MSA is associated with increased operational complexity as each service should be ensured forits successful functioning, otherwise the entire application may get failed. As the microservices are distributed, operational complexity becomes still more.

Need for technical expertise

Ultimately expertise is also required to appropriately design and implement MSA. Other issues include network management, breakdown of communication among services, etc. Other research issues include security aspects, implementation of modern security techniques such as blockchain for microservices based application etc.

III CONCLUSION

In this paper, an introduction to microservices architecture is presented. The paper describes theneed for MSA, core concepts of MSA and various research challenges in MSA. Like any other architecture, one has to compare the pros and cons of MSA for a concerned application at hand before implementing MSA.

REFERENCES

[1] Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, and Mike Amundsen, “Microservice Architecture, Aligning Principles, Practices, and Culture”, Published by O’Reilly Media, Inc., 2016.

[2] C. Pahl and P. Jamshidi, "Microservices: A Systematic Mapping Study", Proc. 6th Int'l Conf. Cloud Computing and Services Science (CLOSER 16), pp. 137-146, 2016.

[3] C. Pahl, "Containerization and the PaaS Cloud", IEEE Cloud Computing, vol. 2, no. 3, pp. 24-31, 2015.

[4] O. Zimmermann, "Microservices Tenets: Agile Approach to Service Development and Deployment", Computer Science—Research and Development, vol. 32, no. 3–4, pp. 301-310, 2017.

[5] A. Balalaie, A. Heydarnoori and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture", IEEE Software, vol. 33, no. 3, pp. 42-52, 2016.

[6] Florian Daniel, Barbara Pernici, “Web Service Orchestration and Choreography: Enabling Business Processes on the Web” 2008, IGI Global

[7] Changlin Wan, Ullrich, Limin Chen, Zhongzhi Shi,”on solving QoS – AWARE service selection problem with service composition,” grid and cooperative computing,2008.pp 467-4

[8] Strunk,”QoS Aware service composition: a survey” ECOWS,IEEE 2010.