

Efficient Hybrid Research for QoS-Aware Microservice Composition



Neha Singhal, Usha Sakthivel, Pethuru Raj

ABSTRACT--- Recent advances in the E-business enable the evolution of software development paradigms. MSA is emerging as an alternative approach to SOA and claiming to resolve various challenges in the software development mechanism. Microservices can be implemented as an independent module for software development. In the growing business era, the user requirement is complex and dynamic. User dynamic requirement is a challenge and cannot be completed by a single microservice hence there is a need of service composition in order to fulfill user dynamic business-related queries. For the appropriate service selection, QoS ontology semantic annotation is performed. There are two service composition methods are available i.e. orchestration and choreography. In this paper, we proposed an effective and efficient hybrid approach for the service composition.

Index Terms — *Choreography, Orchestration, MSA, QoS, Ontology, Microservice.*

I. INTRODUCTION

Now a day's Microservices are promising as a better way to build recent business applications. The microservice development approach is becoming more famous in cloud architectures are gaining much interest recently, MSA application is divided into different fine-grained lightweight chunk services to improve containerization, flexibility, agile development, efficiency [1]. Microservice the micro application is different from SOA and promises to change the way for the software development and service deployment style. Microservices are the basic building block of the microservice architecture that is one of the latest architectural upcoming trends in software development industry and promising to address several challenging issues in software development methodology [2]. Microservice architecture (MSA) is a famous architectural development style adopted by world-wide Internet services such as Netflix, Amazon and eBay. In comparison with monolithic architecture microservice architecture is simple decomposing of the monolith into a set of small chunked services and making them -communicate with each other through lightweight protocols [4] is the common way for software development and advocates rapping all functions in one single stand-alone application. The monolithic

applications are simple in many aspect like develop, test, deploy and scale. Nevertheless, the sheer size of the monolith can slow down the development and become an obstacle to continuous deployment because of the longer start-up time. At present, the advantages of microservices are commonly accepted and appreciated in both academia and software industry. Microservice includes maintainability, service- reusability, scalability, Availability may be incorporated in a microservice architecture.

The rest of the paper is organized as follows. Sect. II briefly defines the paradigms for service composition. Sect. III describes a hybrid approach. Sect. IV. Presents a discussion of the implementation and results. Finally, Sect. V concludes and gives the direction for future work.

II. RELATED WORK

Microservices Architecture (MSA) got significant publicity from the industry and research community in present years. The major impact of MSA is on building lightweight protocol services that should meet the complex and dynamic needs of business applications. This architectural style promotes isolation, autonomy, share-nothing, and independent service choreography. MSA supposed to increase the agility factor, developer productive speed, application scalability and monitoring, reliability, maintainability in the improved way. It is also a suitable choice for iterative development processes like agile software development and DevOps based business applicationd[5].

Service composition refers to the collaboration and integration of microservices in order to fulfill the complex requirements of the consumer. Service composition is merging various services together in order to complete the dynamic complex queries. There are two methods for service composition i.e. orchestration and choreography as per the literature.

A. Microservice Orchestration: -

Service orchestration represents a single centralized coordinated and single point control collaboration Process. The centralized coordinator is called as the orchestrator that controls and monitors the interaction among various independent microservices. The orchestrator is single authorized point for service invocation and combination. The relationship between all the involved services is managed by a single endpoint i.e. the composite service. The orchestrator monitors the management of transactions between individual services. Orchestration is a centralized approach for service composition [6].

Revised Manuscript Received on 30 July 2019.

* Correspondence Author

Neha Singhal*, Information Science and Engineering, Rajarajeswari College of Engineering, Bengaluru, India.

Usha Sakthivel, Computer Science and Engineering, Rajarajeswari College of Engineering, Bengaluru, India.

Pethuru Raj, Reliance Jio, Bangalore, India .

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

In orchestration, entire functioning is handled by the central controller and that is called as the orchestrator. Usually, microservice orchestration is good for small application. Security is also not a major concern in this centralized environment. Orchestration, scenario maintenance is also easy. But usually, it is good for localization.

The orchestration procedure is shown in figure [1].

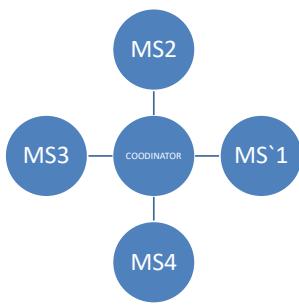


Figure:1 microservice orchestration

B. Microservice choreography: -

Service choreography is a description of the participating services, which is defined by the exchange of messages, rules of interaction and agreements between two or more endpoints. Choreography employs a decentralized approach for service composition. In service choreography, one service talk with another one and then the process continues in the decentralized manner [6].

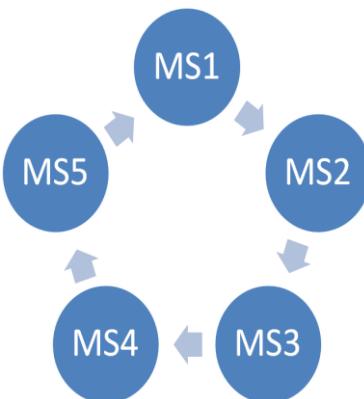


Figure 2: microservice choreography

Orchestration:

- The single coordinator controls the composite application, so security is not a major concern.
- The organization can monitor the performance and can increase the performance with additional investments in its own.
- The organization can monitor the level of reliability.
- If there is a failure in the system, it can be resolved because of a single point of control.
- It is flexible because the new technology can be easily implemented without reference to other organizations.

Choreography: -

- The composite application contains two or more

coordinating domains so that the dynamic linkage of Choreography is very helpful.

- If there is a highly specialized service component that is required to perform a particular function, then choreography is preferred.
- In intelligence gathering situation and in smaller businesses, linking to outside for various services may, likewise, be sensible.

Finally, if service components are treated as "modules for applications" then many of the benefits are the same as for end-user using "modules" on their wireless device.

The problem with choreography is that it is not within the single control.

Here we discussed both the techniques orchestration and the choreography, and we reached to the conclusion that for the real-life scenario services need to collaborate. For the local scope orchestration is a better approach but for the global dimension, the choreography is a better solution. For the dynamic real-time application, the services need to collaborate in both the manner. So there is a need to merge both.

A *choreographed event-based style* as well as orchestrated *command-based style* depending on the situation at hand. With a clever mix of both an optimal solution can be obtained.

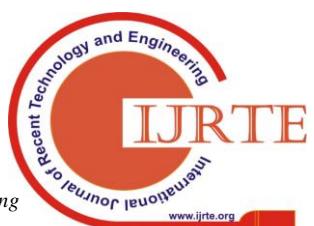
C. Appropriate Microservice Selection: -

Now a days for the researchers the hotspot topic is appropriate micro service selection as today's world can be referred to information world. E-commerce is becoming the backbone of this whole connected world. In recent time for the new software development, there is no need to develop any module from the scratch because each and every information and the required solution is available in the service repository. In service repository, each service can be referred to as microservice. For any user dynamic requirement or query related and appropriate services can be searched in the service repository but in these search mechanisms the appropriate service selection is becoming a challenge as many similar services are available in the repository. For appropriate service selection (quality of service) parameters plays a remarkable role.

QoS Parameters includes various nonfunctional aspects for the service selection mechanism. Many algorithms have been developed to solve these problems.

For example, suppose a user wants a hospital facility encompassing appointment booking, booking hospital room, selecting a doctor, ambulance booking, billing, insurance. A single service cannot fulfill all the requirements of the user so there is a need to compose various services together. Some point of time user demands that the cost of executing microservices to be minimized. Here the QoS requirement plays an important role. In other words, the user requirement should be handled with minimum cost. Like cost, there can be various other QoS attributes like availability, response time, and throughput, and success-ability, reliability etc.

Various QoS algorithms are available for composition.



D. Classification of QoS Composition Approaches: -

QoS Based service discovery approach is classified in to static approach and adaptive approaches.

Table 1: - Classification of QoS based Approaches

CLASSIFICATION OF QoS APPROACHES		
Algorithm type	CLASSIFICATION TYPE	Algorithms
Dynamic programming [7]	Static approach	Branch and bound [8]
	Local maximization approaches	Learning DFS [9]
		Simple additive weighing [9]
Linear programming techniques [7]	Static approach Linear optimization approaches	Linear integer programming [11]
Approximation approach [11]	Static Approximation method	Particle swarm optimization (PSO)[11]
		Genetic algorithm [12]
Pareto-optimization [13]	Static approach Pareto-optimization approaches	Simple additive weighting (SAW)[13]
Internal adaption approaches [14]	Adaptive approach	AI planning Reinforced learning [15]
External adaption techniques	Adaptive approaches	Social network analysis [[16]] Service rank [16]

The classification of QoS based approaches is represented in table 1.

III. PROPOSED WORK

At present only two techniques are available for microservice collaboration i.e. **microservice choreography** approach where one service talk with another n process continues. **Microservice orchestration** approach is where a central controller will manage the overall collaboration of services.

But still, these two ideas cannot be far from each other up to an extent. Both the approaches having dependency based on various dimensions like dependency based on structure and dependency based on their functionality and the resource dependency etc.

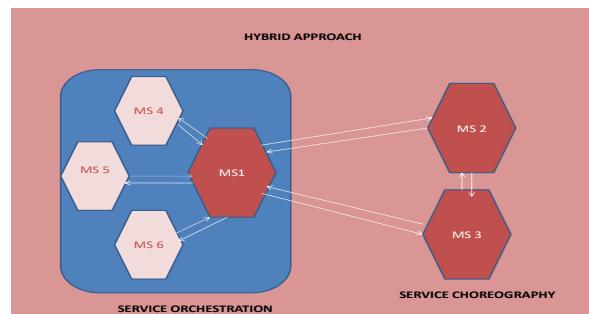
In this paper, we propose a **hybrid approach** that is the combination of both choreography and orchestration. In the hybrid architecture we do the clever mix of both the approaches in order to achieve the better performances and an optimal solution.

In various research works, it is observed that one single method cannot be fit into all the software development designs. Especially in the software industry, the requirement is changing day to day. In this flexible industry one approach can not fit for all and even it is applicable for the existing collaboration approaches also i.e. micro service orchestration

And choreography too as both are having its advantages and disadvantages .In one case one approach is better and viceversa.in this situation we propose a hybrid approach that can add value to the current software development industry.

Hybrid approach: -

In a hybrid pattern the service choreography uses between services and the orchestration is called within a particular service. In figure 3: microservice MS1, MS2, MS3 are collaborating in choreography approach. Microservice MS1 consumes additional services MS4, MS5, MS6 will be called in orchestration pattern. Service MS1 works as a central coordinator for service MS4, MS5, MS6. Service MS1 is an orchestrator for services MS4, MS5, MS6. Later all the services execute in a choreographic manner. In figure 3: the architecture diagram is given for the proposed hybrid approach.

**Figure 3 Architecture of a hybrid approach**

In figure 3 the proposed architecture of hybrid approach is shown which is the combination of choreography and microservice orchestration. MS1 microservice consumes microservice 4,5,6. Ms1 is the orchestrator. MS1 collaborate with microservice 2,3 in a choreographic manner.

IV. IMPLEMENTATION AND RESULT ANALYSIS

For the proposed research work the health care domain is taken into consideration. The health care application is implemented using JAVA eclipse. Various Restful services are implemented in JAVA for the health care domain like appointment booking, insurance, bill payment, ambulance etc. For the implementation of semantic web and web ontology protégé 3.4 is installed with eclipse IDE. For the selection of appropriate microservices parameters are taken into consideration response time, availability, throughput, reliability, etc. for the similar microservices k-means clustering techniques is applied the K-means algorithm is implemented to frame the various clusters. Each cluster contains a similar services.K-means algorithm is used to select the most suitable service. Later apriori algorithm is implemented to make the association among the other relevant services from the other clusters.

For this association analysis, we used support count and confidence count for the implementation of the apriori algorithm.



Efficient Hybrid Research for QoS-Aware Microservice Composition

Later we performed the service collaboration by using service orchestration, service choreography, and the proposed hybrid approach. Later we monitored the performance of execution based on time, memory consumption and the power consumption for all the three approaches i.e. orchestration, choreography, and the hybrid approach. The below-given figures show the execution performances in the graphical representation.

Configuration of Machine: -

Windows 8.x
Windows 64-bit server
RAM 128 MB
Minimum Pentium 2 266MHz processor
Apache Tomcat 7.0
Postgres 9.0
MySQL 5.17.

Time performance: -

In figure 4 execution time is evaluated for the various microservice execution with service orchestration, service choreography, and the proposed hybrid approach. In the below-given figure, it is observed that the proposed hybrid approach is time effective compare to orchestration and choreography.

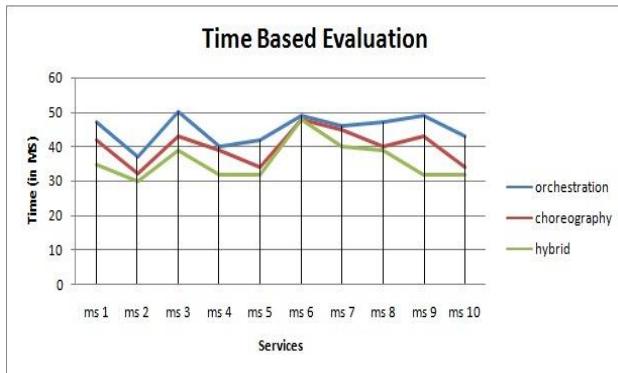


Figure 4: Time based Evaluation

Memory Utilization: -

Memory utilization refers to space utilization for the various service executions. Figure 5 shows that memory utilization is better in a hybrid approach to compare to orchestration and choreography.

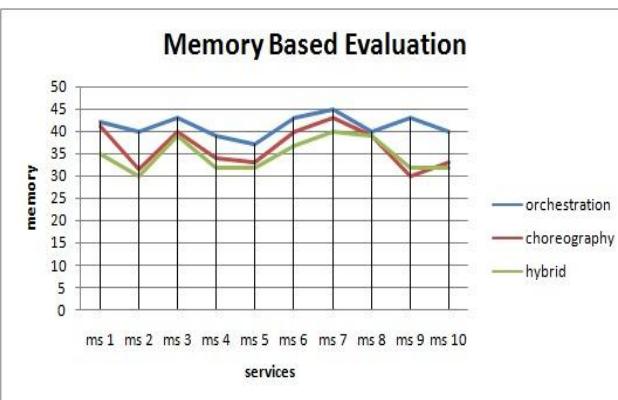


Figure 5: Memory-based Evaluation

Power utilization: -

Power utilization is less for a hybrid approach compares to the existing one. the hybrid approach is efficient for power-based utilization. Figure 6 shows power utilization for all three approaches.

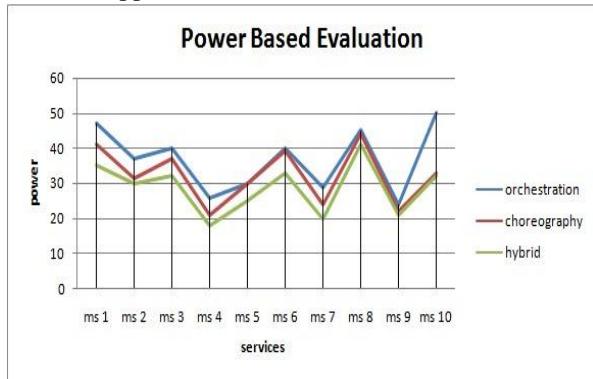


Figure 6:-Power utilization graph

V. CONCLUSION & FUTURE WORK

In this paper, we present an evaluation of our proposed hybrid approach for service composition method. In order to evaluate the performance of the proposed hybrid approach i.e. the combination of orchestration and choreography, we measure the execution time, memory utilization and the power consumption. At the time of execution QoS Parameters is taken into consideration and for the semantic annotation protégé is used with JAVA. For the appropriate service selection, QoS parameters and k-means clustering algorithm and the association analysis methods are implemented to automate the procedure. Fig 4, Fig 5, Fig 6 shows the time evaluation, memory utilization, power consumption with the improved performances compare to the existing two approaches i.e orchestration and choreography. In future work, the hybrid approach can be improved further. The implemented hybrid approach is time, memory, power efficient.

REFERENCES

1. Haitao Zhang, Ning Yang, Zhengjun Xu, Bingchang Tang, Huadong Ma" Microservice based Video Cloud Platform with Performance-aware Service Path Selection" 2018 IEEE International Conference on Web Services, DOI 10.1109/ICWS.2018.00048
2. Kleanthis Thramboulidis, Danai C. Vachtsevanou, Alexandros Solanios" Cyber-Physical Microservices An IoT-based Framework for Manufacturing Systems"IEEE 2018
3. M.Fowler and J. Lewis, "Microservices," 10 June 2014. [Online]. Available: <https://martinfowler.com/articles/microservices.html>
4. C. Richardson, "Pattern: monolithic architecture," 22 June 2017.
5. Muhammad Waseem, Peng Liang" Microservices Architecture in DevOps" 2017 24th Asia-Pacific Software Engineering Conference Workshops
6. Ján Terpák, Pavel Horovák, Matej Luká" Mathematical models creation using orchestration and choreography of web services" 2016 IEEE DOI:- 978-1-4673-8606-7
7. Zhenqiu huang, wei jiang, zhiyong liu,"Effective Pruning algorithm for QoS –AWARE service composition"commerce and enterprise computing,2009 CEC09, IEEE ,VOLNO PP 519-522.
8. Wonhong nam,h yunyoung kil,jungjae lee,"QoS DRIVEN service composition using learning based depth first search,"commerce and enterprise computing,2009 ,IEEE VOL NO,PP 507-510.
9. Bansal S,Blake M" trust based dynamic web service composition using social network analysis," arizona state university pp1-8 dec 2010.



10. Changlin wan,ullrich,limin chen,zhongshi shi,"on solving QoS – AWARE service selection problem with service composition," grid and cooperative computing,2008,pp 467-474.
11. Lou yuan-sheng,Hu pa, Tao fu-ling, "An improved particle swarm optimization and its application on web service composition" ICCASM,2010 international conference vol.11 .
12. A wang, H.Ma, and M. zhang , "Genetic programming with greedy search for web service composition," in proc,database expert syst.appl 2013,vol 8056,pp. 9-17.
13. Yoo,kumara,dongwon lee,Seog-chan" A web service composition framework using integer programming with non -functional objectives and constraints,"IEEE 2008,PP 347-350
14. Zibin zheng, lyu," distributed QoS evaluation for real world web service," ICWS IEEE2010.
15. Rami mounla "QoS aware web service composition" 2008.
16. Strunk,"QoS Aware service composition: a survey" ECOWS,IEEE 2010.
17. Florian Daniel, Barbara Pernici" Web Service Orchestration and Choreography: Enabling Business Processes on the Web" 2008, IGI Global.

AUTHORS PROFILE



Neha Singhal is having 9.5 years of teaching experience, presently working as an Assistant Professor, Dept. of Information Science and Engineering since 2010. She obtained her M.Tech from banasthali university, Rajasthan. She is pursuing her Ph.D. under VTU in the area of web services. Her teaching and research interests are in the field of web services.



S. Usha, is working as a professor and head, CSE, RRCE. Graduated from Manonmanium Sundaranar University, in CSE during the year 1998. She obtained her master's degree in CSE and PhD degree from Satyabhama university in the area of Mobile Ad Hoc Networks in the year 2013. She has 54 publications in International and National conferences, 22 publication in national journal and international journals in the area of Mobile Ad hoc Networks and wireless security. Most of the publications are having impact factor cited in google scholar (h index and i10index), Microsoft etc.. Received fund from AICTE under NCP scheme and SERB(DST). Received best teacher award from Lions club in the year 2010&2012. Developed Centre of Excellence lab in IoT with industry collaboration. Organized many conferences ,FDPs and Technical Talks. Associated with ISTE, CSI, IEEE, IAENG, IDES and IACSIT. Reviewed papers in IJCs and CiiT journals. Acted as a TPC member in MIRA'14 IoTBDS '17 and IoTBDS'18 Portugal. Chaired sessions in FCS'14, ICISC'13 & ICCCT'15., ICCCT'17and IoTBDS'18.



Dr. Pethuru Raj Chelliah Finished the CSIR-sponsored PhD degree at Anna University, Chennai and continued with the UGC-sponsored postdoctoral research in the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. Thereafter, I was granted a couple of international research fellowships (JSPS and JST) to work as a research scientist for 3.5 years in two leading Japanese universities. Published more than 30 research papers in peer-reviewed journals such as IEEE, ACM, Springer-Verlag, Inderscience, etc. Have authored 8 books thus far and focus on some of the emerging technologies such as IoT, Cognitive Analytics, Blockchain, Digital Twin, Docker Containerization, Data Science, Microservices Architecture, fog / edge computing, etc. Have contributed 30 book chapters thus far for various technology books edited by highly acclaimed and accomplished professors and professionals