USING DEPENDENCY GRAPH AND GRAPH THEORY CONCEPTS TO IDENTIFY ANTI-PATTERNS IN A MICROSERVICES SYSTEM: A TOOL-BASED APPROACH

Isuru Udara Piyadigama Gamage

189316U

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

April 2021

USING DEPENDENCY GRAPH AND GRAPH THEORY CONCEPTS TO IDENTIFY ANTI-PATTERNS IN A MICROSERVICES SYSTEM: A TOOL-BASED APPROACH

Isuru Udara Piyadigama Gamage

189316U

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

April 2021

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

………………………………………….. ………………………………………

I.U. Piyadigama Gamage Date

The above candidate has carried out research for the Masters thesis/ Dissertation under my supervision.

………………………………………….. ………………………………………

Dr. Indika Perera (Research Supervisor) Date

ABSTRACT

Microservice based application developments are becoming more and more popular and becoming the common trend in implementing large scale applications. Unlike the traditional monolith applications, microservice applications are composed of many services hence there is an immense possibility of anti-patterns introduced in to the system. To identify these design problems, detailed analysis of the architecture needs to be performed. Tracing data plays a significant role here but it is less useful on its own because of the complexity and the increased quantity of information. Diving in to this data and analysing them to identify anti-patterns exist is a burdensome task because of the lack of tools available thus it requires an immense human intervention and effort due to its complexity. The most common tools available for this kind of scenarios are focused on presenting the information visually so that they are more human readable. The dependency graph of a microservices system, generated by these tools is a good example. However, these tools are not up to performing analysis on the traced data and presenting developers with more statistical information such as metrics along with visualization techniques so that developers can take actions by evaluating metrics and visualizations both, not just by the visualizations. In this thesis we present a solution, a tool for this problem which is capable of utilizing traced data of a microservice system to generate dependency graph and thereby extract metrics using graph theory concepts and algorithms from the dependency graph. The graph theory concepts such as Degree, Clustering coefficient, Paths and distances are used to generate relevant metrices. These metrices are presented to the user in a statistical and a visualized way by the proposed solution so that developers can evaluate them and take decisions on refactoring. To build the proposed tool, we used some already available, industry well recognized tools to develop relevant applications, trace the data and store the data. We use a system which has been developed with microservice style for the analysis. The system has been introduced with anti-patterns intentionally. In this research we focus on anti-patterns specifically The Knot, Nano service, Chained service, Bottleneck service and Cyclic dependency. We analyse the microservice system we have developed with the solution tool presented to identify these anti-patterns.

Keywords

Microservices, Distributed systems, Anti patterns, Graph theory, Dependency graph, Tracing

ACKNOWLEDGEMENTS

I would like to specially thank my supervisor Dr. Indika Perera for the immense support and guidance given throughout the research. He conducted the lectures for core modules related to Software Architecture that were immensely useful in this research. Then I would like to thank Dr. Sanath Jayasena who conducted the lectures for Advanced Algorithm module which the knowledge gained was also extremely helpful in this research.

My gratitude towards all the lecturers for all the modules I took during the course that equipped me with basic to advanced knowledge in Computer Science. I am grateful to Dr. Charith Chitraranjan, Mrs. Vishaka Nanayakkara and Dr. Malaka Walpola who accepted me for this MSc course in the selection interview.

I would like to thank both technical and non-academic staff of the department for their kind and prompt support which made the process smooth. I would like to thank my dear colleagues in the MSc course that supported me throughout the course.

Finally, I wish to thank my parents for all the support given throughout the course to complete it successfully.

TABLE OF CONTENTS

Declaration ii
Abstract iii
Acknowledgements iv
Table of Contents v
List of Figures vii
List of Tables ix
List of Equations ix
List of Abbreviations x
1. Introduction 1
1.1 Background 1
1.2 Problem Statement 3
1.3 Motivation 4
1.4 Objectives 5
1.5 Thesis Structure 6
2 Literature Review 7
2.1 Concepts 7
2.1.1 Microservices 7
2.1.2 Anti Patterns 9
2.1.3 Distributed Tracing 14
2.1.4 Service Dependency Graph 17
2.1.5 Graph Theory Concepts 19
2.1.6 Graph Databases 27
2.2 Related Work 30
2.2.1 Service Based Metrics 30
2.2.2 Service Dependency Graph 32
2.2.3 Anti-patterns in SOA 35
3 Methodology 36
3.1 Introduction 36

3.2 Proposed Solution..37
3.2.1 Data extraction..38
3.2.2 Evaluate dependency graph with metrices 38
4 Implementation of MAIG...45
4.1 Collect and persist data...46
4.2 Graph database..50
4.3 Process data 51
4.4 Frontend application 56
4.5 Data visualization and metrics..57
4.6 Zipkin configuration 59
5 Evaluation and Discussion...61
5.1 Capturing trace data of the system under test (SUT)..61
5.2 Evaluation...62
5.2.1 Dependency graph evaluation...62
5.2.2 Metrics evaluation 64
6 Conclusions...75
6.1 Contributions 75
6.2 Limitations..77
6.3 Future work...77
7 References..79

LIST OF FIGURES

Figure 1: Monolithic and Microservice architectural styles [7] 8
Figure 2: The Knot in a microservice system 10
Figure 3: A microservice system with three nano services 11
Figure 4: A microservice system with a service chain 12
Figure 5: A microservice system with a bottleneck service 12
Figure 6: A possible cyclic dependency in a microservice system 13
Figure 7: Communication sequence of a microservice application [14] 15
Figure 8: Complete call sequence of a user request [14] 16
Figure 9: Types of Graphs 17
Figure 10: Microservice dependency graph 18
Figure 11: Associated Degrees of nodes in a graph 20
Figure 12: A microservice dependency graph with degrees 21
Figure 13: Path in a graph highlighted 22
Figure 14: A microservice dependency graph with trace paths highlighted 22
Figure 15: A Non-cyclic and a Cyclic path highlighted in a graph 23
Figure 16: Weighted graph 24
Figure 17: Cluster Coefficient of few graphs 25
Figure 18: A microservice dependency graph with Clustering Coefficients of services .26
Figure 19: An example graph database 28
Figure 20: Microservice architecture evaluation process 37
Figure 21: Microservice Anti Patterns Insights Generator (MAIG) architecture 45
Figure 22: A sample trace data provided by Zipkin interface 47
Figure 23: Dashboard of the MAIG 56
Figure 24: An example service dependency graph generated by MAIG 57
Figure 25: MAIG section to identify Knots in the service architecture 58
Figure 26: Dependency graph of the SUT 62
Figure 27: Clustering coefficient bar chart for the SUT 64

Figure 28: Degree_Out metric bar chart for the SUT 66
Figure 29: Visualization for outgoing connections of SUT 67
Figure 30: Degree_In metric bar chart for the SUT 69
Figure 31: Visualization for incoming connections to Inventory service of SUT 70
Figure 32: Path Length metric bar chart for the SUT...72
Figure 33: Visualization for unique traces of SUT 72
Figure 34: Visualization of cyclic dependencies of SUT...74

LIST OF TABLES

Table 1: Anti-patterns and corresponding metrics 39
Table 2: Clustering coefficients of a service dependency graph....................................40
Table 3: Degree_Out values of a service dependency graph 41
Table 4: Path Lengths of traces of a service dependency graph 42
Table 5: Degree In values of a service dependency graph 43
Table 6: Trace collector component interfaces 47
Table 7: Properties of nodes and edges of the dependency graph...................................51
Table 8: RESTful interfaces of Graph processor 51

LIST OF EQUATIONS

Equation 1: Total Degree of a Graph 20
Equation 2: Degree of a Graph node..20
Equation 3: Clustering Coefficient...25

LIST OF ABBREVIATIONS

API	Application Programming Interface
IOT	Internet of Things
SIC	Service Invocation Chain
REST	Representational State Transfer
IIS	Internet Information Services
HTTP	Hypertext Transfer Protocol
SUT	System Under Test
QOS	Quality of Service
SOA	Service Oriented Architecture
OOP	Object Oriented Programming