

# **COST OPTIMIZED SCHEDULING FOR MICROSERVICES IN KUBERNETES**

Sugunakuamr Arunan

219312R

Degree of Master of Science

Department of Computer Science and Engineering  
Faculty of Engineering

University of Moratuwa  
Sri Lanka

July 2023

# **COST OPTIMIZED SCHEDULING FOR MICROSERVICES IN KUBERNETES**

Sugunakumar Arunan

219312R

Thesis submitted in partial fulfillment of the requirements for the degree  
Master of Science in Computer Science

Department of Computer Science & Engineering  
Faculty of Engineering

University of Moratuwa  
Sri Lanka

July 2023

## DECLARATION

I declare that this is my own work, and this thesis 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 the thesis, 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).

Signature:

Date: 9<sup>th</sup> July 2023

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

Name of Supervisor: Prof. G.I.U.S. Perera

Signature of the Supervisor:

Date:

## **ACKNOWLEDGEMENT**

I would like thank Dr. Gayashan Amarasinghe who guided me from the beginning throughout the research and did not hesitate to help even after migrating. His expertise in related area helped me a lot in setting the right direction for this project and accomplishing the research goals. I would like to thank Dr Indika Perara for agreeing to be my supervisor.

I would not have achieved this without the immense support from my wife and family. A special thanks to my wife and family for coping with me throughout my research program and encouraging me to complete it. I would like to thank my parents for supporting me throughout my studies from the start the up until today.

I also want to thank the University of Moratuwa for giving me an opportunity to participate in the MSc program and for providing the necessary resources to complete this research. This would not have been easy without the support from my workplace as well. I would like to thank WSO2 for allowing me to do this part-time MSc and research while working with them.

## ABSTRACT

The usage of Container Orchestration Platform like Kubernetes for running Microservices applications is increasing nowadays. In a particular application, all Microservices do not have the same priority. Hence it is costly to allocate the same resources to both high and low-priority services. Spot instances are an attractive option for running low-priority services due to their significantly lower cost compared to On-Demand instances. Spot instances are available for use when cloud service providers have excess capacity and can be bid on at a much lower price than the On-Demand rate. But they can be revoked anytime by the Cloud provider which affects the availability of the services.

This research aims to utilize Spot instances to run low-priority services with the intention of reducing the cloud cost while providing overall high availability to the application. A thorough literature review has been conducted on existing research that utilizes Spot instances to save cost while maintaining high availability. This study builds upon previous work and proposes a new approach to run low priority Microservices to save cost. A service called KubeEconomy has been proposed to monitor and manage Kubernetes worker nodes to efficiently schedule the Microservices. Three functionalities of the KubeEconomy service have been explained which contributes to the cost optimization. The KubeEconomy service utilizes cloud APIs and Kubernetes APIs to promptly scale and reschedule pods within different nodes.

Two experiments were conducted to show the effectiveness of KubeEconomy service. In the first experiment, the KubeEconomy service was deployed on Azure cloud to manage a Kubernetes cluster. The experiment showed that the KubeEconomy service was able to dynamically provision and deprovision Spot instances based on the workload demand and Spot evictions, resulting in significant cost savings while maintaining high availability of the Microservices. In the second experiment, a simulation was conducted using the parameters gathered from the first experiment to calculate the cost savings of long running workloads. It is shown that it is possible to reduce the cloud cost up to 80% while maintaining 99% availability for the Microservices under optimal conditions.

**Keywords:** Cloud computing, Container Orchestration, Kubernetes, Microservices, Cost optimization, High availability, Spot Instances

# TABLE OF CONTENTS

Declaration.....	i
Acknowledgement .....	ii
Abstract.....	iii
Table of Contents.....	iv
List of Figures.....	vi
List of Tables .....	vii
List of Abbreviations .....	viii
Chapter 1: INTRODUCTION .....	1
Chapter 2 : LITERATURE REVIEW .....	5
2.1 Introduction.....	5
2.2 Scaling algorithms .....	9
2.3 Efficient scheduling .....	12
2.4 Fault tolerance.....	14
2.4.1 Migration .....	15
2.4.2 Checkpointing.....	17
2.4.3 Replication.....	18
Chapter 3: METHODOLOGY .....	20
3.1 Infrastructure.....	20
3.1.1 Managed Kubernetes solution .....	21
3.1.2 Scheduler .....	22
3.1.3 Scheduled Events service.....	23
3.1.4 Daemon Service.....	24
3.1.5 Test client.....	25
3.2 Scheduler functionalities.....	26
3.2.1 Peak-time Upscaling.....	26
3.3.2 Handling Eviction.....	30
3.3.3 Optimizing the scheduling frequently.....	34
Chapter 4 : EXPERIMENT .....	35
4.1 Experiment 1: Deployment on Azure .....	36

4.2 Experiment 2: Running simulations .....	39
Chapter 5 : EVALUATION.....	41
5.1 Pricing & Eviction History .....	42
5.1.1 Spot Pricing History .....	42
5.1.2 Spot Eviction Rate .....	43
5.2 Experiment 1 Results .....	44
5.2.1 Availability .....	44
5.2.2 Cost .....	45
5.3 Experiment 2 Results .....	46
5.3.1 Eviction Rate vs Additional VMs .....	46
5.3.2 Availability of a Microservice vs Eviction rate .....	47
5.3.3 Cost savings vs Spot price with different eviction rates.....	48
5.4 Summary .....	49
Chapter 6 : CONCLUSION & FUTURE WORKS.....	50
References.....	54

## LIST OF FIGURES

<b>Figure</b>	<b>Description</b>	<b>Page</b>
Figure 1	Relationship among different instances	7
Figure 2	Lifecycle of a Spot Instance	8
Figure 3	Relationship between Spot instances	9
Figure 4	Two stage Triple Modular Redundancy	15
Figure 5	Workflow scheduling on top of Spot instances	16
Figure 6	Infrastructure	20
Figure 7	KubeEconomy Scheduler	22
Figure 8	Rule & CronJobs	26
Figure 9	Upscaling & Downscaling	27
Figure 10	Handling Eviction	30
Figure 11	Cluster optimization	34
Figure 12	Teastore Microservices	37
Figure 13	Service calls for Teastore Microservices	38
Figure 14	Spot price comparison	42
Figure 15	Spot eviction rates	43
Figure 16	Eviction rates vs Additional VMs	46
Figure 17	Availability vs Eviction rate	47
Figure 18	Cost savings vs Eviction rate	48
Figure 19	Cost savings Heat Map	49
Figure 20	Elon Musk's tweet	50



## LIST OF TABLES

<b>Table</b>	<b>Description</b>	<b>Page</b>
Table 1	Teastore Microservices	36
Table 2	Different Node pools used in Experiment 1	37
Table 3	Experiment 2 configured parameters	39
Table 4	Availability of Teastore Microservices after Scenario 2	44
Table 5	Cost of running the Microservices	45

## LIST OF ABBREVIATIONS

<b>Abbreviation</b>	<b>Description</b>
SI	Spot Instance
VM	Virtual Machine
CRD	Custom Resource Definition
KE	KubeEconomy
HA	High Availability
API	Application Programming Interface
AKS	Azure Kubernetes Service
IMDS	Instance Meta Data Service
K8s	Kubernetes