Tmp/2020/48

Sri Lanka Institute of Information Technology

Project Topic Assessment – 2020 Regular

Topics

Network Science Based Approach for Optimal Microservice Governance

Abstract (200 Words Max):

Microservices have become one of the most popular development architectures for many software applications developed today, mainly due to its efficient an effective way to decompose a large and complex system into their functional components and implement a loosely coupled, self-contained system which supports scalability and performance. As a result of this, many organizations deploy their application embracing microservice architecture and make use of such platforms as Kubernetes to ensure successful deployment of their application in production.

Nevertheless, even with the use of microservice deployment tools and platforms such as Kubernetes, a common issue that many developers come across, is an effective approach to evaluate and monitor their effectiveness and viability of their microservice deployment and identify potential performance bottlenecks. This is especially an important factor to be considered in the case when there are a vast number of interconnected microservices in an application.

This research carried out aims to find an effective approach to the above identified problem through developing a model which will help and provide aid to evaluate, monitor, maintain and oversee microservice deployments through the use of several techniques which include improvement of efficiency through metric analysis gathered based on identified dependency measurements, effective autoscaling through load prediction, a microservice monitoring solution by providing a Business Intelligence dashboard as well as providing solutions to come up with an optimal deployment strategy for the microservice cluster.

Research Area/Group: Select the area by referring to the document uploaded to the Courseweb

Distributed and Parallel Computing

Supervisor :

Name: Dr. Dharshana Kasthurirathna

Added to the Project Registration System ☐

(Signature)

Research Problem:

Research Problem - In deploying Microservices through Kubernetes, there is no efficient and effective way for developers to evaluate and monitor the effectiveness and viability of a microservice deployment and identify possible performance bottlenecks. Furthermore, developers are not able to optimize their deployment such that they can make the optimal use of their deployed microservices in the cluster.

Throughout the years, with the introduction of many cloud-based platforms such as Azure, AWS and Google which primarily incur cost based on resource utilization, it has become an increasing need for developers and system administrators to monitor, evaluate and optimize their Kubernetes deployments such that they could be able to maximize the usage of their available resources without sacrificing performance. However, maintaining this balance between performance and resource utilization is quite challenging for developers and system administrators since this process requires constant monitoring of metrics and finetuning multiple variables and paraments in their deployment configurations such that the optimal performance criteria are met.

Although current Application Performance Management (APM) tools and platforms in Kubernetes allow gathering and observation of these metrics, there is no effective solution for developers and system administrators to evaluate the gathered metrics and get a clear idea on the effectiveness and viability of microservice deployments and identify the possible performance bottlenecks.

Key statements from which the above-mentioned research problem was identified, and ultimately led to the identification of the research objectives, can be found in the paper “Performance engineering for microservices: research challenges and directions”, published by Robert Heinrich and his team. The extensively detailed research paper, clearly highlights the importance of performance evaluation with regard to microservices and the importance of monitoring and monitoring solutions through the use of APM tools, especially in those that are deployed in containers. This paper further goes on to state the need for APM tools deployed in containers to include additional measures to monitor microservices such that they could be used as input for resilience mechanisms and creation of auto-scaling policies, and therefore, has since been one of the key inspirations for the initiation of this research project.

Some additional research papers which were used to identify the above-mentioned research problem as well as the objectives of this research have also been mentioned below.

References

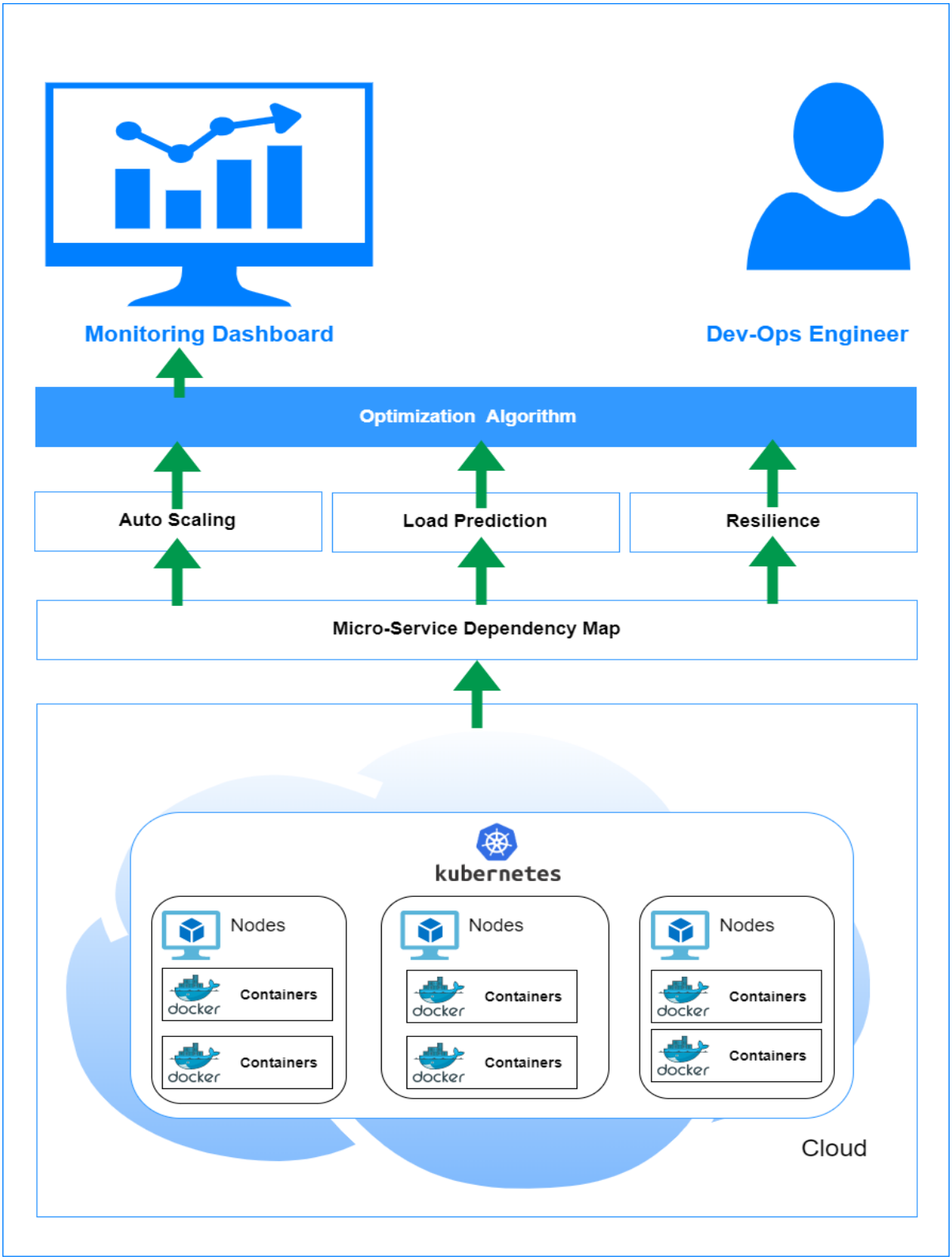
- Heinrich, Robert, et al. "Performance engineering for microservices: research challenges and directions." Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion. ACM, 2017.
- T. F. Düllmann and A. van Hoorn, Model-driven Generation of Microservice Architectures for Benchmarking Performance and Resilience Engineering Approaches. 2019.
- M. Fazio, A. Celesti, R. Ranjan, C. Liu, L. Chen and M. Villari, "Open Issues in Scheduling Microservices in the Cloud," in *IEEE Cloud Computing*, vol. 3, no. 5, pp. 81-88, Sept.-Oct. 2016.

Solution proposed:

- First, we will select a Kubernetes cluster that is currently deployed and identify the dependencies among microservices using tools such as Istio and Prometheus.
- Then, based on data gathered using those tools we hope to derive metrics and obtain quantitative measures to determine the dependency strength among the interconnected microservices.
- Based on the findings obtained from the previous step we hope to generate a real-time dependency chart to represent the strength among the interconnected microservices
- We also hope to use tools such as Chaos Monkey to evaluate the resilience of the microservice cluster and using the dependency chart developed previously, we will evaluate the resiliency of a particular microservice in the cluster.
- Meanwhile, we will also perform thorough analysis on the microservice cluster to evaluate the auto-scaling and load balancing policies used, and based on the dependency chart generated previously, we hope to develop an algorithm using AI and Machine Learning to optimize the current load balancing an auto-scaling policy such that the developed algorithm could be used to predict future loads and automatically scale the cluster in time.
- Furthermore, an optimization algorithm will also be developed using AI and Machine Learning, which takes into account all the data gathered from all other components to come up with an optimal deployment strategy.
- Lastly, all the data will be fed to a dashboard so that the user will be able to view all the findings.

System Overview Diagram for the solution proposed (Clearly indicate the main four components of the proposal)

Four main components - Auto Scaling, Load Prediction, Resilience, Microservice dependency map



Objectives (1 main objective and 4 sub-objectives):

Main objective

To model a network science-based approach to govern microservice deployments through evaluation and analysis of metrics gathered, and ultimately come up with a proposed model which aids to optimize microservice deployments.

Sub objectives

- To increase the efficiency of microservices deployments by applying the metrics used in network analysis, such as centrality and resilience measures and link predictions on identified dependency measurements.
- To develop an improved auto-scaling policy for a deployment based on load prediction
- To development of a business intelligence dashboard to evaluate performance and monitor microservice deployments.
- To identify key factors which lead to performance reduction in microservice deployments and come up with an optimal deployment strategy.

Task List divided among the members

Member 1 - Saranga S.A.G

- Creation of the dependency map as well as the metrics to evaluate the dependency strength of a microservice cluster and working on configuring auto-scaling of the cluster based on the load-prediction algorithm.

Key tasks included in the development of the above process include:

- Setting up and configuring the initial Kubernetes microservice cluster for metric analysis.
- Setting up and configuring the relevant tools such as Prometheus and Istio etc. in order to retrieve the required metrics.
- Identify and retrieve the required metrics from the various metrics analyzing tools
- Analyze the retrieve metrics such that it could be used to determine the interdependencies between the deployed microservices.
- Development of custom metrics using the retrieved metrics which can be used to represent the interdependency between microservices.
- Development of a solution in order to display the interdependencies gathered using metrics in order to display as a dependency map.
- Performing configuration of the auto-scaling of the cluster based on the load-prediction algorithm

Member 2 - De Silva N.

- Developing the algorithm that will be used to predict future loads for automatic-scaling of the cluster

Key tasks included in the development of the above-mentioned task include:

- Retrieval of the necessary metric data from pods in the cluster in order to perform the load prediction based on the developed dependency map.
- Development of solution in order to store the gathered metric data for time series analysis.
- Preparing and manipulating the extracted data in order to establish a time series. (data cleaning)
- Development of a time series data set based on the historic metric data retrieved, such that future loads could be predicted.
- Selection of the most appropriate time series forecasting model for load prediction.
- Development of the algorithm with the help of Machine Learning, for the time series using the selected model in order to predict future loads.
- Finetune the developed model in order to get the most accurate prediction.

Member 3 - L.S. Jayasinghe

- Development of the optimization algorithm which will be used to come up with the optimal deployment strategy.

Key tasks included in the development of the above process include:

- Retrieval of the analyzed outputs from the load prediction, dependency analysis map as well as resiliency evaluation process.
- Formatting and data manipulation of the data received from the above-mentioned output processes, in order to be fed as inputs to the optimization algorithm.
- Development of final optimization algorithm making use of Machine Learning, which takes in inputs from the load prediction, dependency analysis map and resiliency evaluation process, and proposes the suggested optimal deployment strategy.
- Integrate with the UI dashboard in order to display the optimal deployment strategy to the user.
- Evaluate the effect of the developed optimization algorithm with respect to the performance of the microservice cluster.

Member 4- Lakshitha M.V.

- Evaluating the resilience of the microservice cluster through the use of a "Chaos Engineering" approach with the help of tools such as Chaos Monkey by performing targeted attacks on the microservices identified with the help of the developed dependency map and, also the task of the development of the business intelligence dashboard which displays the findings of the optimization process.

Key tasks included in the development of the above process include:

- Using the dependency map, perform the development of a system which analyses the resiliency of the cluster using "Chaos Engineering" tools such as "Chaos Monkey".
- Identification of the microservices which show low resilience from the above-mentioned process.
- Evaluating the deployed microservice cluster and coming up with proposed suitable measures in order to improve the resiliency of the deployed microservice cluster.
- Evaluating the effect of the proposed resiliency measures with respect to the performance of the microservice cluster.
- Development of the UI dashboard which displays the final findings of the optimization process to the user and the displaying of the optimal deployment strategy to the user.

Technologies to be used:

- Kubernetes
- Docker
- Istio
- Prometheus
- TensorFlow
- Chaos Monkey
- Azure Kubernetes Services (AKS)

Team Members:

Student Name	Student ID
Leader: Saranga S.A. G	IT17016230
Member 2: De Silva N.	IT17006880
Member 3: L.S. Jayasinghe	IT17012966
Member 4: M.V. Lakshitha	IT17410250

For official use only

Acceptable: YES/NO

Minor Corrections (if necessary)

_____

_____

_____

Any other Comments:

_____

_____

_____

Approved by the review panel:

Member's Name	Signature

Important:

1. According to the comments given by the panel, do the necessary modifications and get the approval by the **same panel**.
2. If the project topic is rejected, find out a new topic and inform the CDAP Group for a new topic pre-assessment.
3. A form approved by the panel must be attached to the **Project Charter Form**.