Comprehensive Study of Container Orchestration Frameworks

Ashwini Ravikumar
School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland
X00180728@myTUDublin.ie

Introduction

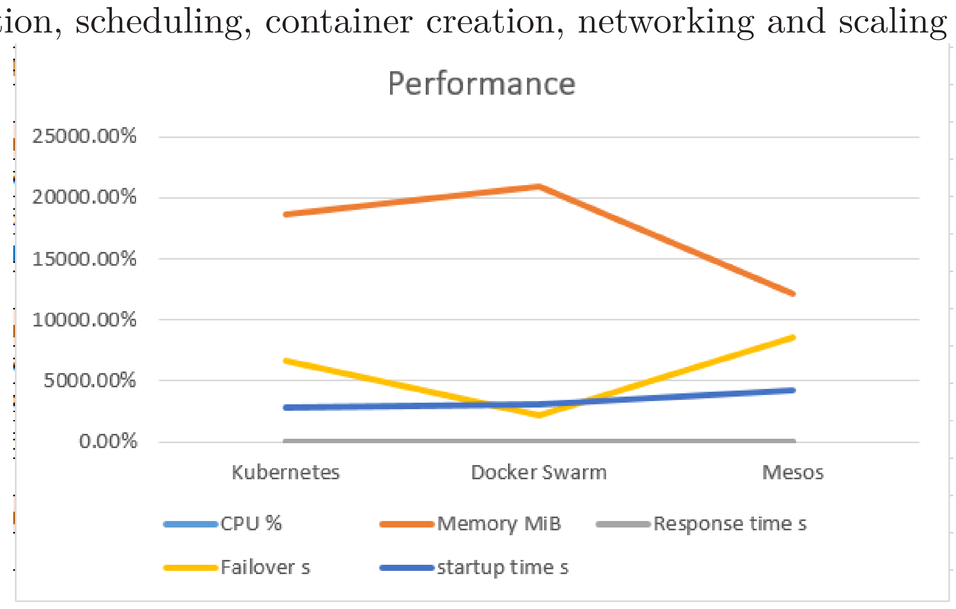
Container orchestration has emerged as a crucial technology in modern software development and deployment, providing a systematic and automated approach to application definition, deploying, managing resource allocation, scheduling, container creation, networking and scaling containerized applications. This paper offers a comprehensive overview of container orchestration, fundamental functionalities, performance and future prospects. Conducting a qualitative analysis of container orchestration technologies involves evaluating their characteristics, skills ease-of-use, advantages, disadvantages, and appropriateness for various scenarios/use cases based on the organisational needs. While they all offer container orchestration capabilities, they differ in their approach, features, and strengths, catering to different use cases and preferences. Tools leveraged for comparative study: Kubernetes,Docker Swarm,Apache Mesos. These tools are presently a shortfall of thorough and objective assessment of their functionality and performance. This absence hinders IT managers in selecting the most appropriate orchestration solution. The empirical data demonstrates that Kubernetes surpasses its competitors in terms of performance when it comes to very complex application deployments.

Functional and Performance metrics evaluation

RQ1: Comparing the key features of chosen orchestration frameworks
RQ2: Comparing performance metrics for bench-marking
1.Qualitative Analysis: Comparing Common features vs Unique features: Gave us a clear understanding of features from which we can choose a CO framework which suits our needs and case scenario.
2.Quantitative Analysis: We used Grafana, Prometheus, and built-in cAdvisor to monitor the performance of CO frameworks for the below scenarios and conducted test experiments to measure performance of the chosen CO frameworks.
Scenario 1: Deploying 2 Container (Nginx-Go)
Scenario 2: Deploying 3 Container (InfluxDB)
Scenario 3: Deploying 4 Container (WordPress)

Topic Overview

Will a functional and performance comparative evaluation be good enough to identify the best container orchestration tool? Evaluate the experiments such as cluster provisioning time the orchestration tools take to deploy, failover times, startup time and so on. This thesis contributions are mainly of two research parts. The 1st is the study aiming to offer the functionality analysis of Apache Mesos, Docker Swarm and Kubernetes. The 2nd is the study aiming to offer the performance analysis conducting experiments to measure or analyse the application definition, deploying, managing resource allocation, scheduling, container creation, networking and scaling containerized applications.



Conclusion and Future Work

Selecting an orchestration tool is a strategic choice that affects how well and efficiently applications operate in a clustered environment, not just a question of personal taste. This study presents a comprehensive analysis of the three container orchestration tools, comparing the various features and services provided by different container orchestrators.
Answering RQ1 comparing common and unique features of chosen container orchestration tools. We examined based on application provisioning, varied complexity test, container failover, CPU and Memory usage, response time to answer the RQ2 comparing performance metrics. Functional and performance comparative evaluation facilitates in identify the best container orchestration tool based on the use case and complexity of the project.

Kubernetes is currently one of the most comprehensive orchestrators when it comes to functional comparison available in the market. This is why practitioners are choosing to favour it above others. Simultaneously, the intricate structure of the system can, in certain instances, impose a substantial burden that could impede its performance. We also faced challenges like cost constraints of cloud resources, Security challenges, Scalability challenges during our research work.

Future work: Organizations can efficiently adopt and extend the utilization of Istio in Kubernetes. Security implementation and default feature provided the container itself can be studied in depth on CO frameworks.

QR Code for Recording