Modeling and Simulation of Dynamic Containerized Software Architectures using Palladio

Nathan Hagel[1]

Abstract: Nowadays, distributed applications are often not statically deployed on virtual machines. Instead, a desired state is defined declaratively. A control loop then tries to create the desired state in a cluster. To predict the impact on the performance of a system using these deployment techniques is difficult. This paper introduces a method to predict the performance impact of the usage of containers and container orchestration in the deployment of a system. Our proposed approach enables system simulation and experimentation with various mechanisms of container orchestration, including autoscaling and container scheduling using the Palladio-Component-Model (PCM). We validated this approach using a Kubernetes reference cluster which we modelled using a workflow defined in the authors bachelor thesis [Ha22] [Ha23]. Our findings suggest, that the most common concepts in container orchestration can be modelled and simulated using Palladio and the PCM-Extension of [Ha22].

Keywords: Container; Performance Prediction; Container Orchestration

1 Motivation

In recent years, container technologies have become increasingly important for software developement and deployment [Zh18]. Beyond the mere use of containers, container orchestration tools like Kubernetes play a significant role. They allow a declarative description of the desired system state, which the container orchestrator then automatically creates and monitors. The question arises, how these technologies impact a software's performance and which scaling and allocation strategies are the most appropriate.
Gaining these insights from real-world experiments can be costly or restrict the user experience. Using the software architecture simulator Palladio [Re16], component-based applications can already be analyzed. Extending it with container and container orchestration concepts can improve early stage performance analysis and simplify the decision-making process regarding a possible containerization of a system or architecture.

2 Goal and Methodology

To simulate dynamic, containerized software architectures using Palladio, we need to be able to model not only the concept of containers but also the main concepts of container orchestration. The selection of these cocepts was based on Kubernetes as the de facto

[1] Karlsruhe Institute of Technology, Chair of Modelling for Continuous Software Engineering (Prof. Dr. Anne Koziolek), Kaiserstraße 12, 76131, Karlsruhe, Germany, nathan@hagel.dev

doi:10.18420/sw2024-ws_15

standard in this area. The mapped concepts are: Clusters, Nodes, Pods, Ressource-Requests and Limits, Services, Ingress, Deployments, Pod-Scheduling and Autoscaling. Besides a mapping to model the concepts, a solution had to be found to analyze dynamic Deployments. After defining which concepts were needed to realistically simulate containerized systems in Palladio, a requirements-analysis was conducted for each concept to determine which properties and capabilities are required for the model and the simulation. Then, the existing PCM-Elements were analyzed to find out where existing concepts could be reused or extended. To implement the dynamic concepts like Pod-Allocation, Autoscaling etc., we looked at model transformations and developed a concept to implement capabilites like automatic Pod-Allocation and a Horizontal-Pod-Autoscaler (HPA). Futhermore, a Pod-Allocation-Scheduler for PCM-Models was implemented and tested. For evaluation purposes, we determined the share of implemented and for the analysis relevant concepts. Additionally, we defined a reference cluster based on an existing application and modeled this cluster to determine the limits of our extension. Finally, the decisions of the implemented Pod-Allocation-Scheduler were compared to the decisions of a standard Kubernetes scheduler.

3 Results

Based on the defined requirements for each Kubernetes' concept, we were able to find or define a mapping for almost all the concepts into the PCM. The only exception is the Replica Set which has a strong overlap with the Deployment definied in our extension and was therefore not mapped. For the implemented Pod-Allocation-Scheduler, tests showed that it behaves the same as the Kubernetes standard implementation. The defined reference cluster could be fully modeled using the workflow to use the PCM-Extension which was also defined in [Ha22]. A solution was proposed to implement the relevant control loops using model reconfigurations under consideration of possible transient effects. In a follow-up project the proposed dynamic simulation concept for this extension is currently implemented.

Literatur

[Ha22] Hagel, N.: Modellierung und Simulation von dynamischen Container-basierten Software-Architekturen in Palladio, Bachelor's Thesis, Karlsruher Institut für Technologie, 2022.

[Ha23] Hagel, N.: Modellierung und Simulation von dynamischen container-basierten Software-Architekturen in Palladio. In: Proc. 25. Workshop Software-Reengineering Evoluation (WSRE 2023). S. 22–23, 2023.

[Re16] Reussner et al.: Modeling and Simulating Software Architectures - The Palladio Approach. 2016, ISBN: 9780262034760.

[Zh18] Zhang et al.: A Comparative Study of Containers and Virtual Machines in Big Data Environment. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). S. 178–185, 2018.