



Dynamic Orchestration

BigDataStack Software Component developed by NEC

Description

The Dynamic Orchestrator (DO) triggers the redeployment of BigDataStack applications during runtime to ensure they comply with their Service Level Objectives (SLOs.) The DO utilizes an in-house Reinforcement Learning-based logic, which combines domain knowledge with machine learning for bootstrapping performance and obtaining a more robust and stable behaviour.

Features

The DO, alongside the Triple Monitoring Engine, monitors and triggers redeployment changes in the application during runtime, ensuring the compliance with SLOs. This improves the performance of applications effortlessly and can help to reduce costs related to overprovisioning of resources in applications with variable load. The main features of the DO are:

Flexible: it is possible to define different SLOs that the DO will monitor, and it is also possible to define different redeployment actions that the DO will trigger when a change is necessary. Also, thanks to the use of RL, the DO will be able to manage newly defined actions and learn from its experience when these actions should be applied.

Stable and robust: the DO implements a logic that combines machine learning and domain knowledge, obtaining the best of both worlds: on one side, we ensure the agents' performance is reasonable at startup using heuristics, but the agent can improve upon these heuristics as it gathers more experience, thanks to its RL-based logic.



Dynamic Orchestration

Rapid-reaction: redeployment changes are triggered in real-time when an SLO violation is detected, reducing the time in which any SLO is violated.

Areas of Application

The DO can be used by all applications' developers in BigDataStack, and other platforms in which the DO is integrated such as FogFlow. In particular, the applications that will benefit the most from the DO are applications with varying load: for applications that need to serve a varying quantity of requests and/or users, the DO provides an important mechanism to enable applications to comply with their SLOs at all times, without incurring into unnecessary costs such as overprovisioning.

Market trends & opportunities

Cloud applications that serve mobile and web applications with a highly variable load are the norm nowadays, and its use will only increase. Additionally, this trend is also seen nowadays in machine learning applications run on the Cloud to serve users' requests, and in particular, these applications are highly affected by a changing load of requests and users. What is more, ML applications many times cannot be replicated as easily as other applications, so the automatic mechanisms incorporated in commercial platforms such as Amazon Web Services and Azure, many times fall short of expectations and require high effort from developers. The DO is an essential tool for managing these applications and thanks to its flexibility, can implement dynamic adaptation with little effort from developers. DO is classified into the "Data Processing Architectures" within the Big Data Reference categories.

Customer benefits

Developers benefit from optimizing the performance of their applications



Dynamic Orchestration

effortlessly, reducing costs and avoiding SLOs violation, which can cause economical penalties and losing users.

Technological novelty

The DO implements a novel approach called Tutor4RL24 that combines domain knowledge with Reinforcement Learning. This approach has been designed and developed in BigDataStack, implementing into the DO. With Tutor4RL, we address two common problems in the use of RL for real-world use cases:

1. Initial poor performance of the agent: in Tutor4RL, we implement domain knowledge in the form of programmable functions that will guide the behaviour of the agent in its initial steps, providing a decent performance while the agent gathers experience to learn from the application and its context, in BigDataStack, these programmable functions are already implemented following heuristics that represent “rules of thumb” for managing the deployment of applications on the platform.
2. Robustness of the agent’s behaviour: because we are dealing with real applications and errors should be reduced to the minimum, we use programmable functions for constraining the actions of the agent during its whole lifetime, this ensures the agent will not take actions that will surely hurt the performance of the application during runtime, reducing errors caused by the exploration of the agent.

TRL level: 7

Find the Open Source code here:

• bigdatastack-tasks.ds.unipi.gr/mauricio.fadel/dynamic-orchestrator



 bigdatastack.eu