



Optimizing End-to-End test execution: Unleashing the Resource Dispatcher

Cristian Augusto, Jesús Morán, Claudio de la Riva and Javier Tuya
{augustocristian, moranjesus, claudio, tuya} @uniovi.es



50th Euromicro Conference Series
on Software Engineering
and Advanced Applications (SEAA) 2024


Paris, France 2024



Context

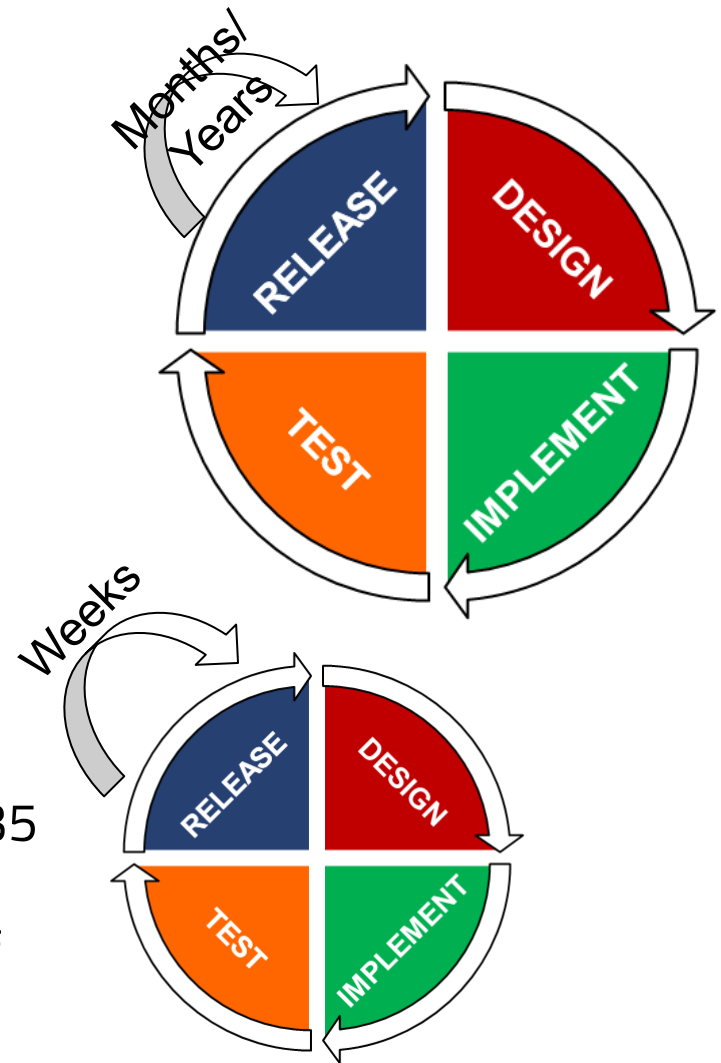
- Nowadays releasing cycles are **shorter** than in the past
- Release speed **increased**, but test suites are **larger** and are executed **frequently** than ever:

 4.2 million of tests running continuously (35 runs/test/day) [1]

 Test suites of 60k test cases, take most of the day to run running [2]

[1] J. Micco, "Continuous Integration at Google Scale," EclipseCon, 2013

[2] How we approach testing VSTS to enable continuous delivery | [Brian Harry's Blog](#).



Motivation

- These test suites improve software **quality** and **reliability**
- As their size **increases**, it's not feasible to execute them at each repository change, due to resource-time constraints.
- Even worse in **resource-consuming** testing levels like **End-to-End (E2E) Testing**

Motivation

- RETORCH [3]: optimizes E2E test suite execution, reducing execution time and unnecessary Resource redeployments



[3] C. Augusto, J. Morán, A. Bertolino, C. de la Riva, and J. Tuya, “RETORCH: an approach for resource-aware orchestration of end-to-end test cases,” *Softw. Qual. J.*, vol. 28, no. 3, pp. 1147–1171, Sep. 2020, doi: 10.1007/s11219-020-09505-2..

Motivation

- RETORCH is useful when the Execution Plan is executed a single time.
- Continuous Integration (CI) the same or different Execution Plans are executed in a short time (e.g., several updates of Dependabot):
 - **IMPROVEMENT**: several Resources could be potentially shared and reused by the different Execution Plans



Jenkins



Dependabot

Objective

Optimize E2E test execution by enabling Resource sharing across different runs of the same or different Execution Plans.

THROUGH

RESOURCE DISPATCHER

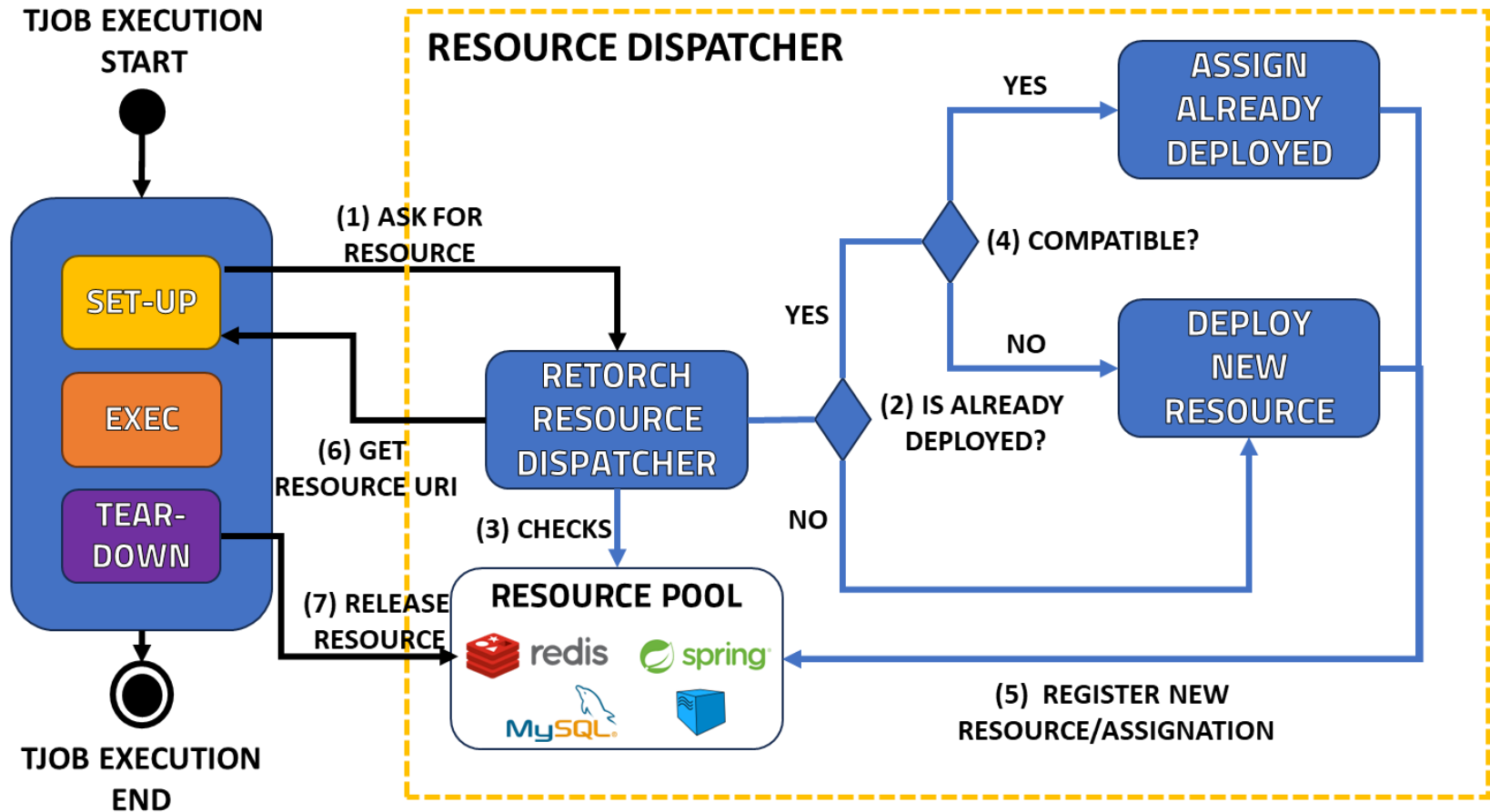
Resource Dispatcher



RETORCH ORIGINAL APPROACH:

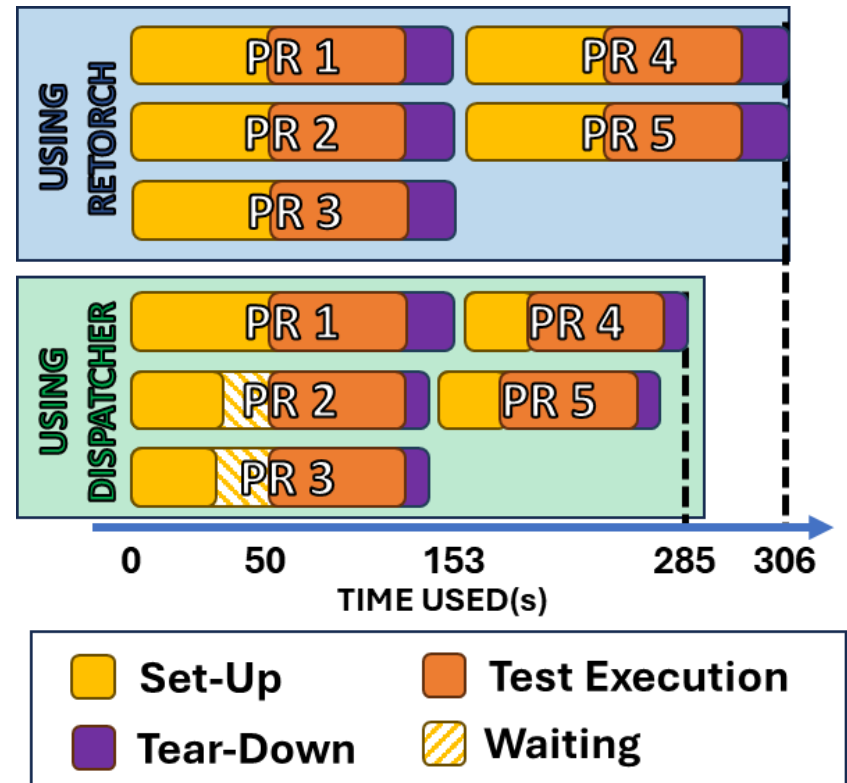
- Decentralized responsibility:
 - **Set-Up**: TJob instantiates its Resources
 - **Execution** (EXEC): executes E2E tests
 - **Tear-down** release of the Resources

Resource Dispatcher



Resource Dispatcher: Proof of Concept

- 5 Dependabot Pull Requests (PR)
- **RETORCH** vs **RETORCH + Dispatcher**
- SAVINGS:
 - **-75%** less Resource redeployments
 - **-7%** less execution time



Conclusions and future work

- The Resource Dispatcher with RETORCH:
 - Addresses open challenges in E2E testing
 - Shows promising reductions in Resource redeployments and execution time.
- As future work we will focus on:
 - Design and implementation of the Dispatcher prototype.
 - Integration into a bot engine that supports the tester as well as the E2E test suite continuous monitorization and improvement.



Optimizing End-to-End test execution: Unleashing the Resource Dispatcher

Cristian Augusto, Jesús Morán, Claudio de la Riva and Javier Tuya
{ [augustocristian](#), [moranjesus](#), [claudio](#), [tuya](#) } @uniovi.es



50th Euromicro Conference Series
on Software Engineering
and Advanced Applications (SEAA) 2024

Paris, France 2024

