RT-SCALER: Adaptive Resource Allocation Framework for Real-Time Containers

Václav Struhár(1), Silviu S. Craciunas(2), Mohammad Ashjaei(1), Moris Behnam(1), and Alessandro V. Papadopoulos(1)

(1) Mälardalen University, Sweden
(2) TTTech Computertechnik AG, Vienna, Austria

Outline

- Motivation & Background
- Problem definition
- RT-SCALER: Adaptive Resource Allocation Framework for Real-time Containers
    - Overview
    - Offline Phase
    - Online Phase
- Practical insight
- Results
- Future Work
- Conclusion

Motivation & Background

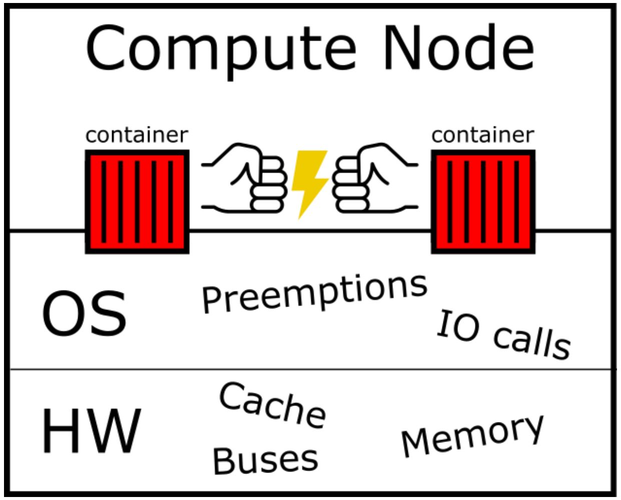
RT containers are gaining tractions:

- PREEMPT_RT + RT Containers
- Hierarchical Scheduling + RT Containers

However, container-based virtualization:

- Interference between containers
- Shared hardware for RT containers + non RT containers
- Unpredictable workloads (IO operations, shared resource usages)

=> Temporal unpredictability

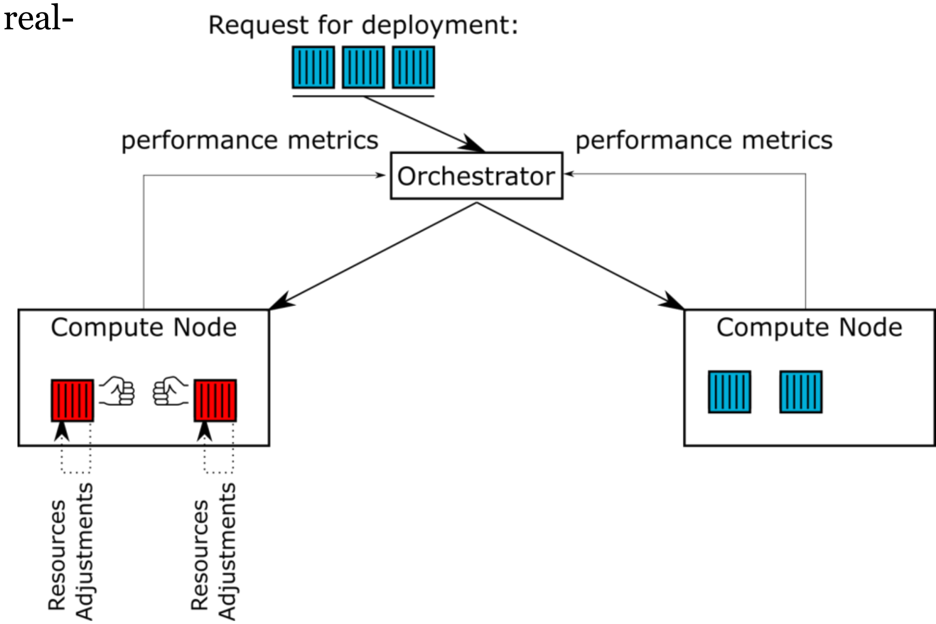


RT-SCALER

High-level idea for an orchestration framework to enable real-time capabilities.

Two Phases:

- Offline (Static allocation)
  - Deployment of container
  - Where to place the container?
- Online (Dynamic allocation)
  - Continuous adjusting resources of containers

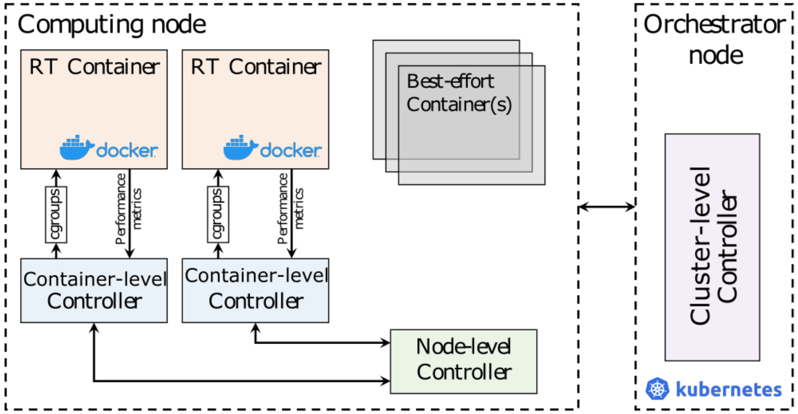


System Overview

The main components:

- RT Containers
- Best-effort containers

- Container-level Controller
  - independently controls resources allocated to the corresponding container
- Node-level Controller
  - maintains the distribution of system resources amongst the real-time containers deployed in a single computing node
- Cluster-level Controller
  - The cluster-level controller has a holistic view of the system and can decide to re-allocate containers to nodes or allocate newly arriving containers to initial nodes.

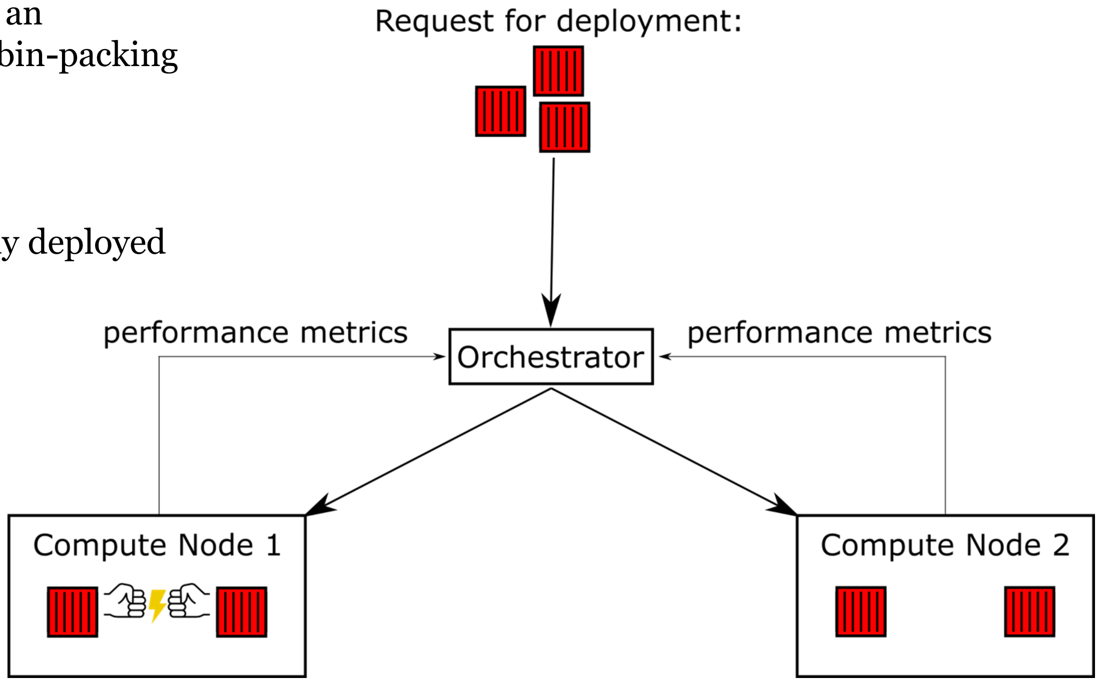


Offline Phase

Where to place the container?

1. Calculate a set of ideal RT interfaces
2. Allocation of containers to nodes => an optimization problem similar to the bin-packing problem

- Additional hints for the orchestrator
  - Performance Metrics of already deployed containers
  - …



Online Phase

- To react on unforeseen changes on temporal behaviour

- Monitoring + Adjusting System Resources + Container Migration

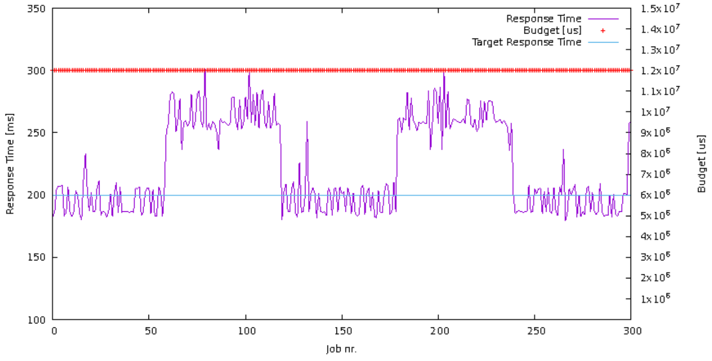
- **Monitoring**
    - Real-time related metrics
        - Deadline miss rate
        - Lateness of the tasks
        - Response time of the tasks
- **Adjusting resources**
    - CPU reservation
- **Relocation**
    - If there is not enough resources in the computing node

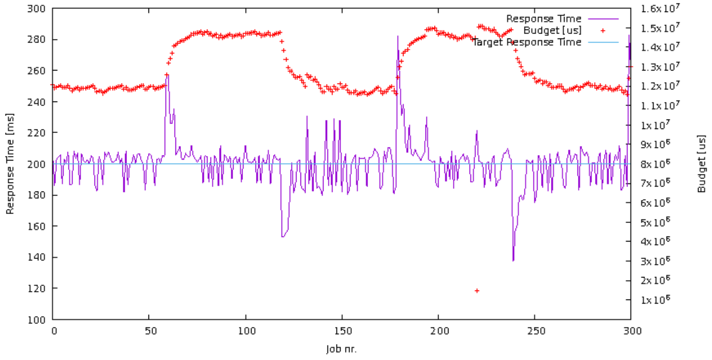
Practical insight

- Linux + Hierarchical Scheduling Patch by Abeni et al. + Docker

- **Monitoring module**
  - Response times

- **Resource adjustment module**
  - PID controller
    - Target response time
    - Adjusting CPU allocation in order to reach the target response time





Future Work

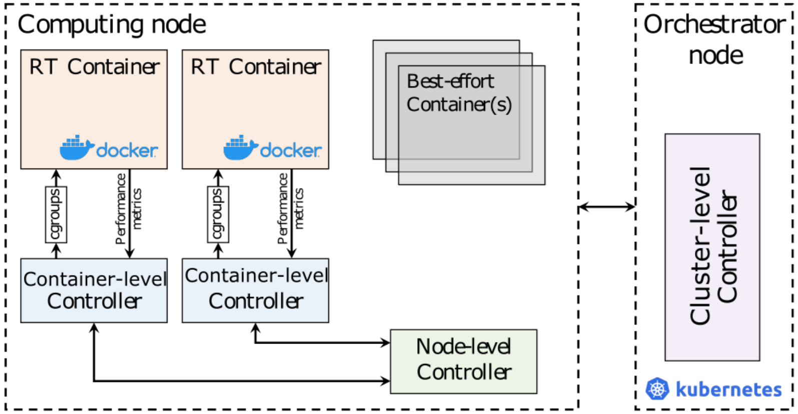
- investigate various adaptation strategies of real-time containers
  - Employ machine learning & predictions
  - Decide what parameters to change (e.g., container period/budget, migration between cores)

- experimentally evaluate the complex control loop across different hierarchical levels in distributed edge computing applications

Conclusion

High-level idea for an orchestration framework to enable real-time capabilities.

- reacting to unforeseen situations
- adaptation of container resources
- migration of containers



Thank You!

MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS

Václav Struhár
vaclav.struhar@mdh.se