Auto-scaling deadline-constrained workloads

in containers in the cloud

Jay DesLauriers
Research Associate,
University of Westminster

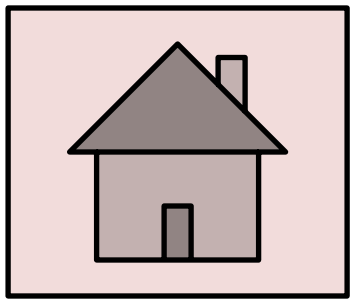
Project COLA

- Horizon 2020
- 33 months
  - Completion September 2019
- 14 Partners in 6 Countries
  - 10 SME/Public Sector
  - 4 HE/Research Institutions

COLA
Cloud Orchestration at the Level of Application

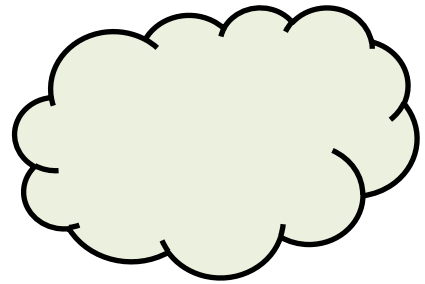
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731574

Head in the clouds



On-Premise

Capital Expense
High Upfront Cost
High Maintenance Cost



Off-Premise

Pay-as-you-go
No Upfront Cost
No Maintenance Cost

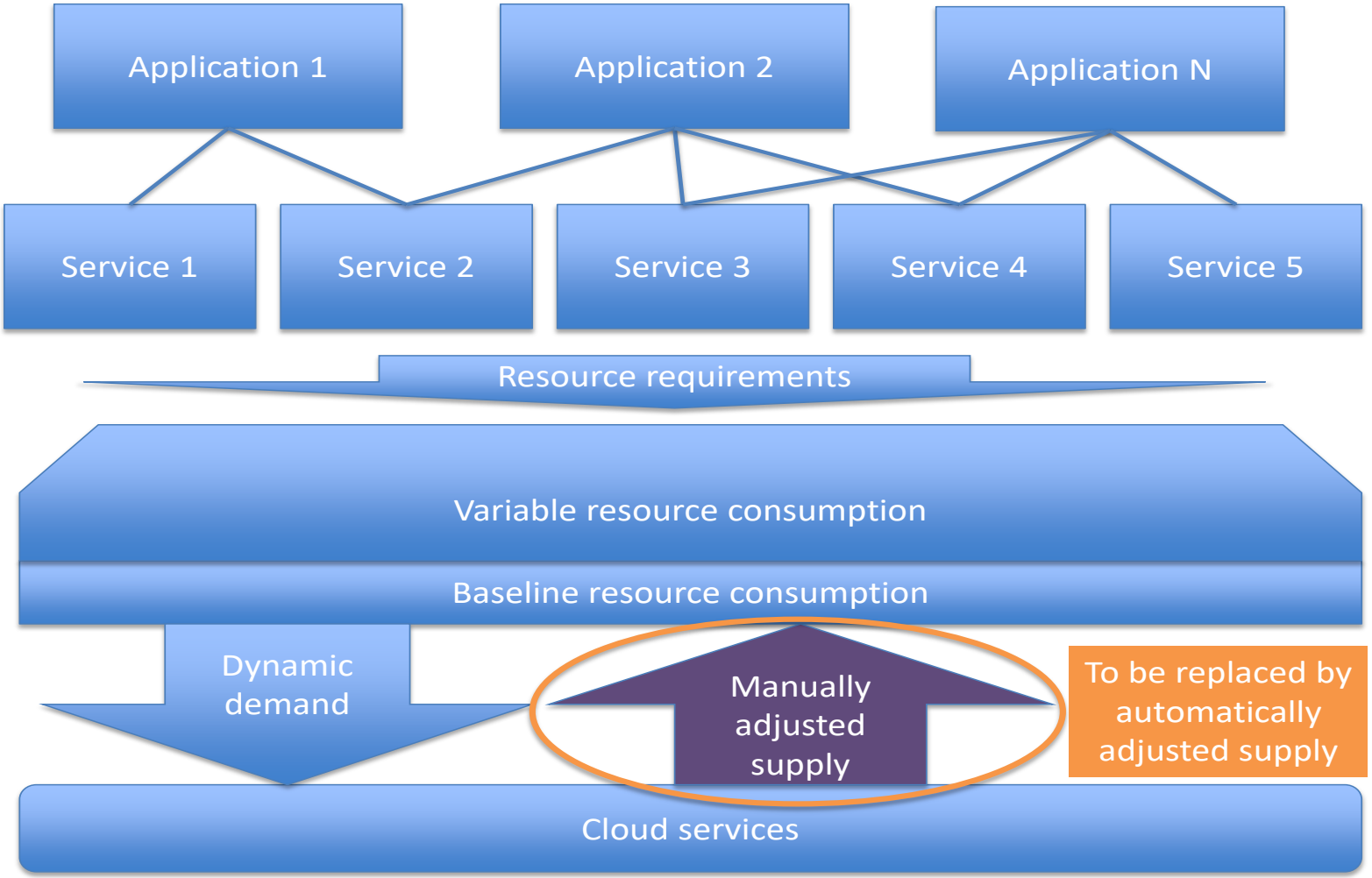
A match made in ...

Containers

Operating-system virtualisation
and
application packaging

for reusable, portable software

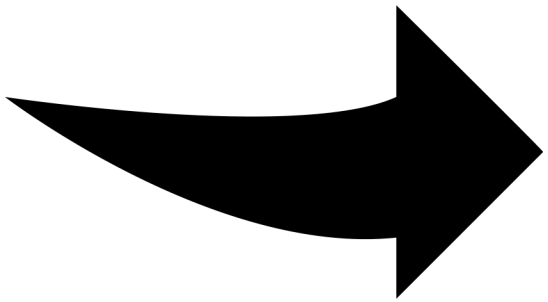
The Problem



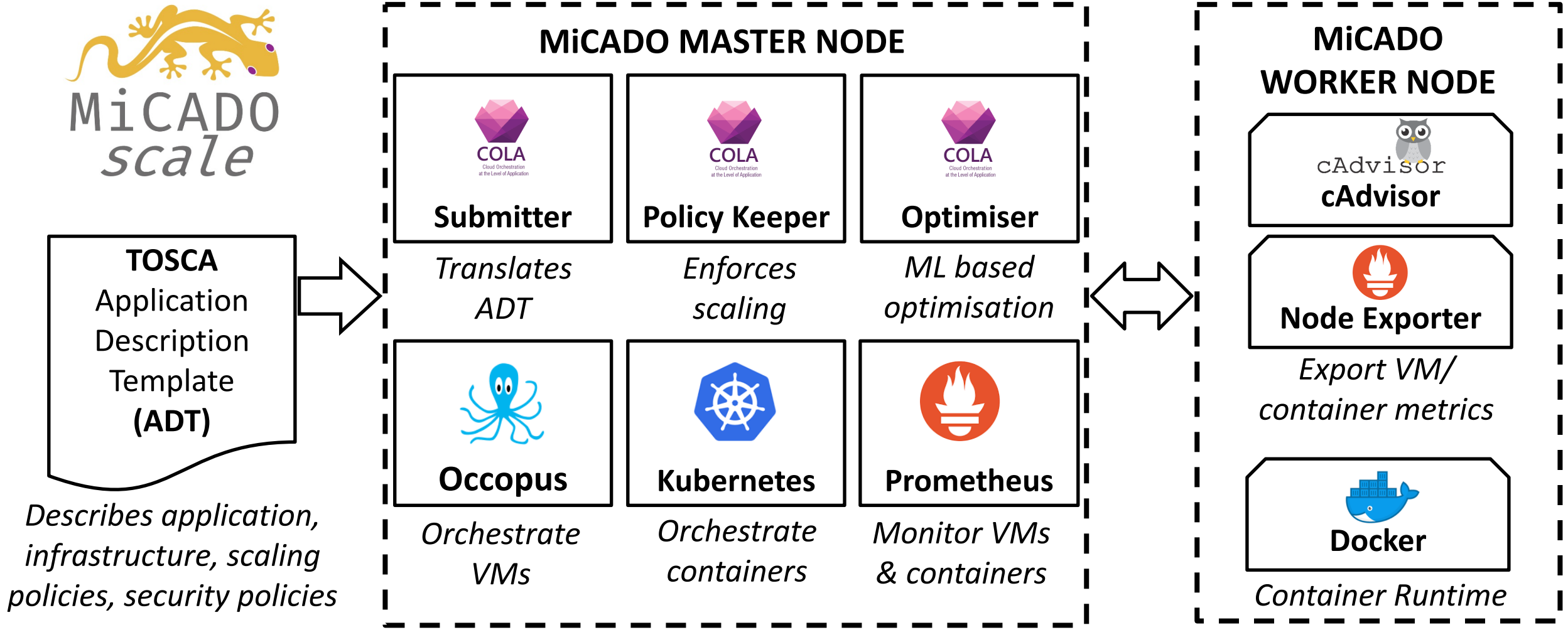
Some requirements:

- Dynamic Supply (auto-scaling)
- Vendor-free
- Modular
- Flexible
- Secure

Finding a solution...

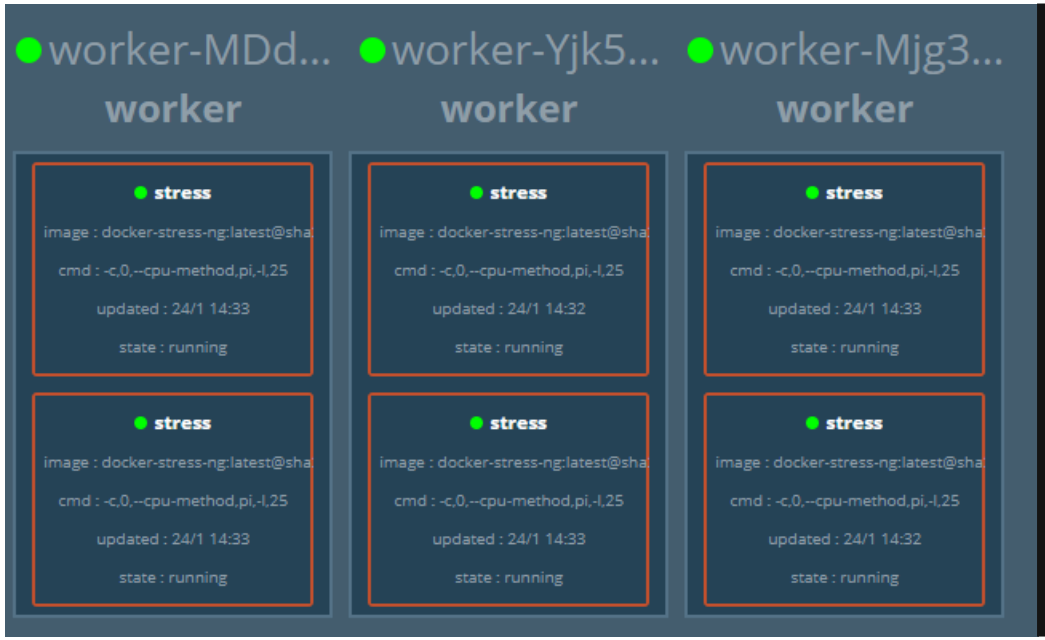


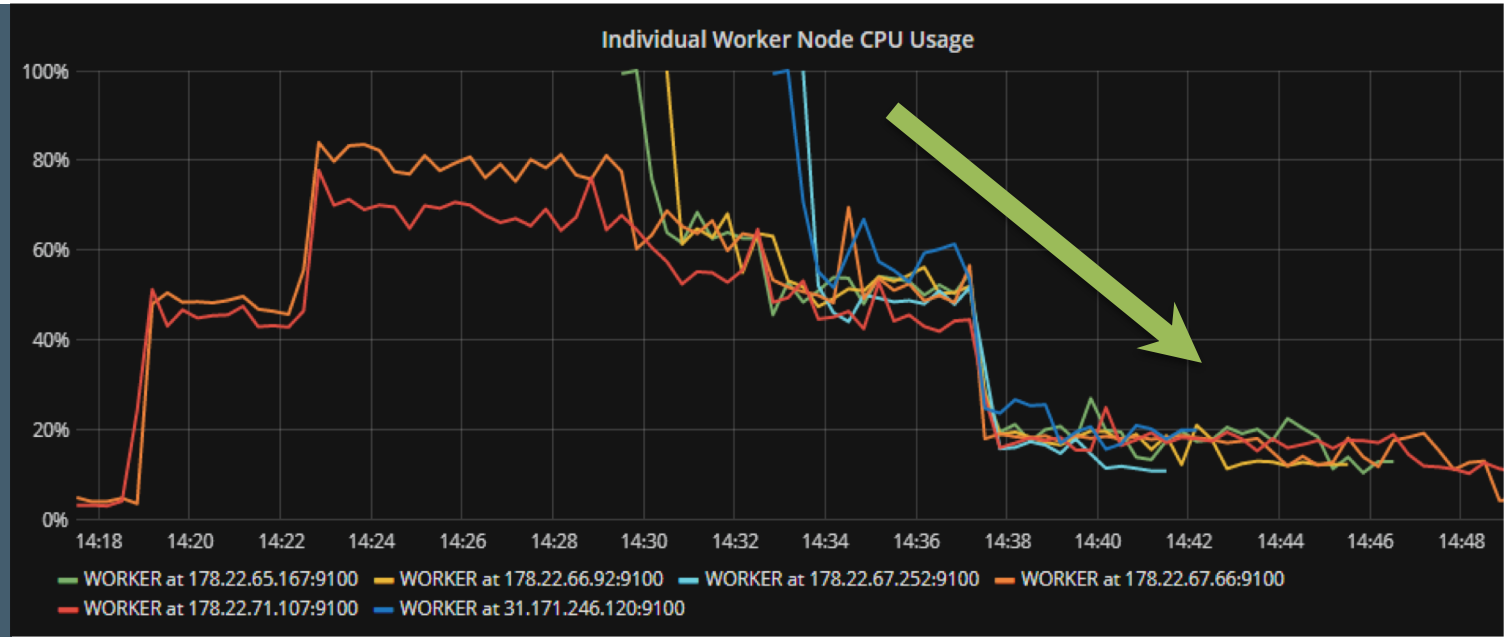
The Solution



Scaling Use-Case No.1

- Resource intensive services
  - Typically CPU/memory –bound apps/services
  - Containers & underlying VMs scale to meet demand

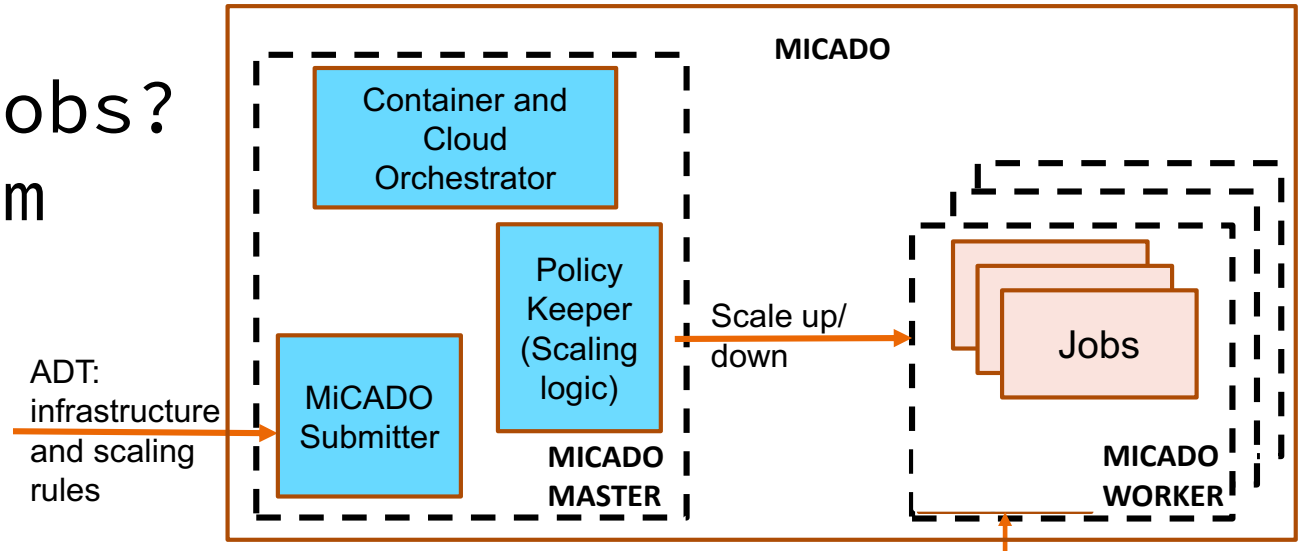




Scaling Use-Case No.2 ... ?

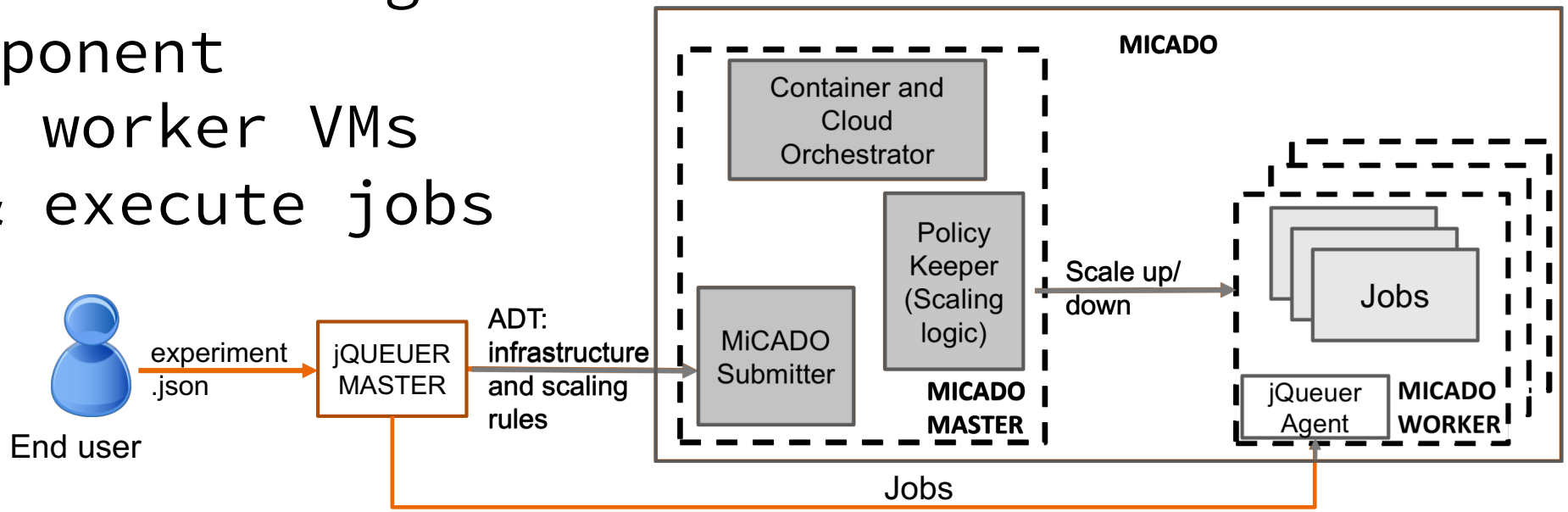
- Multi-job experiments
  - Typically batch/parameter sweep jobs
  - Containers/VMs scale to complete jobs by deadline
  - Where do we put the jobs?
  - How do we execute them (in containers!)

<insert queue here>



JQueuer

- Asynchronous Distributed Task Queue
  - Master Component
    - Runs externally
    - Queue & monitoring
  - Agent Component
    - Runs on worker VMs
    - Fetch & execute jobs



JQueuer Metrics

Metrics exported to MiCADO for scaling:

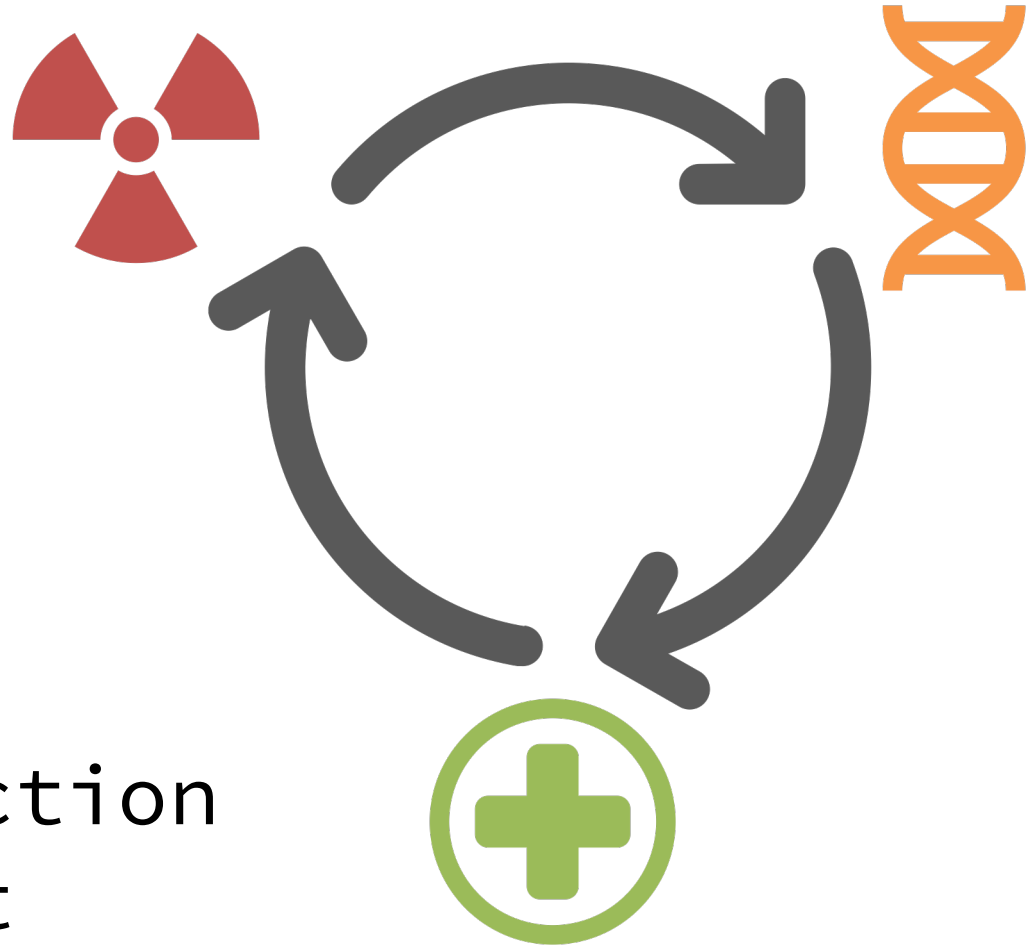
Queue length
Jobs completed
Jobs failed
Jobs running
Jobs remaining
Time elapsed
Average job length
Time to deadline

The experiment

Determining the impact of changes in behavior on the spread of a disease across a population

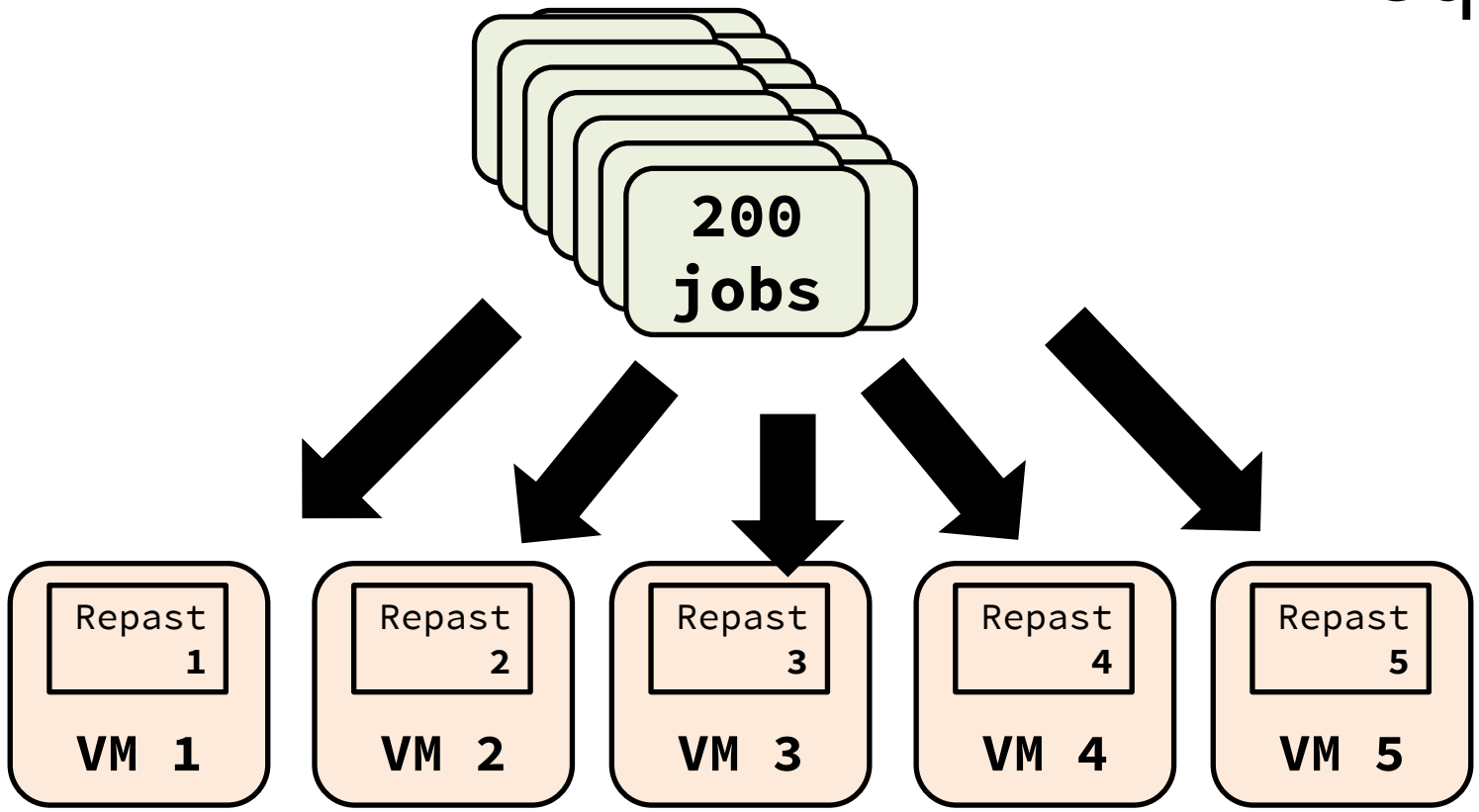
Experiment design

- Agent-based simulation
  - Repast Simphony
- Three agents
  - Infected
  - Susceptible
  - Recovered
- Simulate movement & interaction of agents in an environment

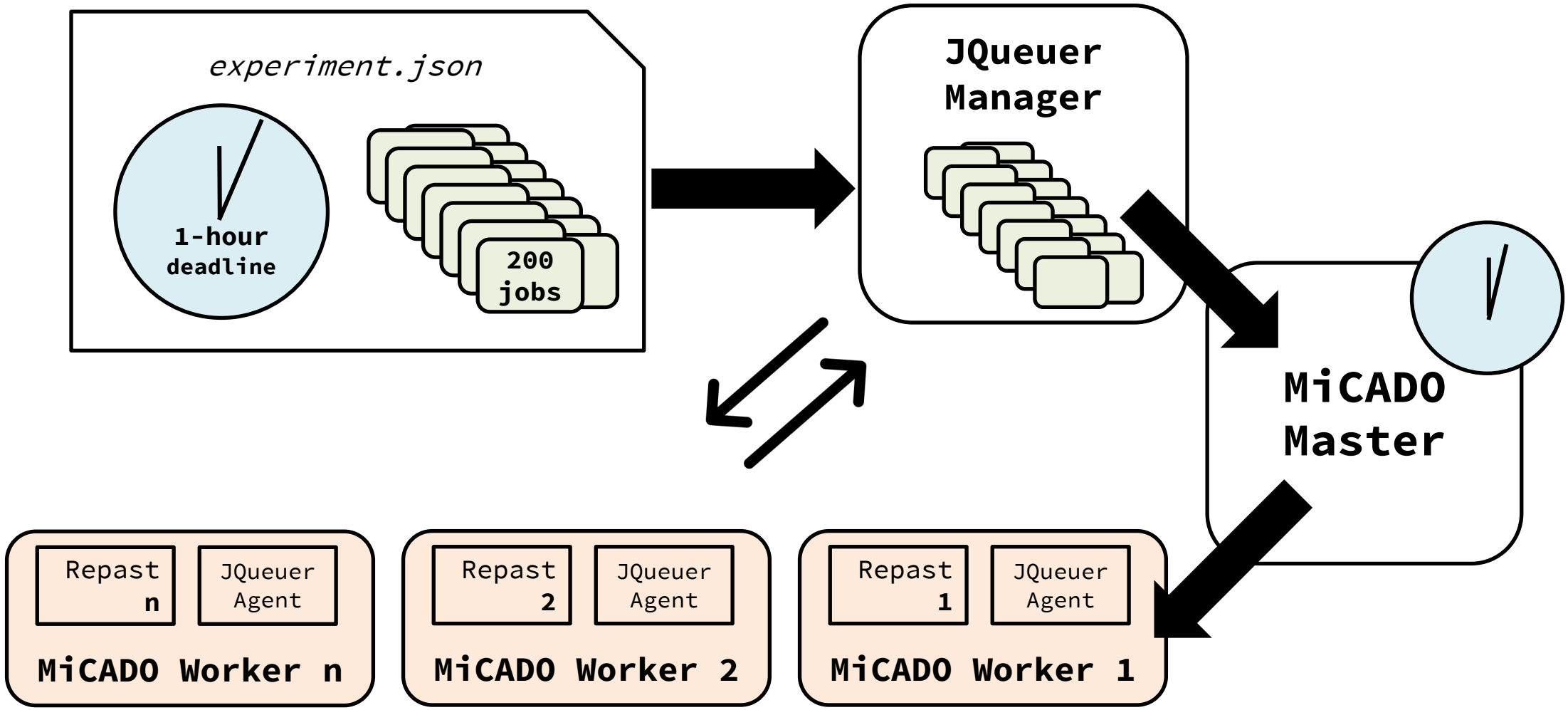


Manual job allocation (baseline)

equal distribution
40x jobs per VM

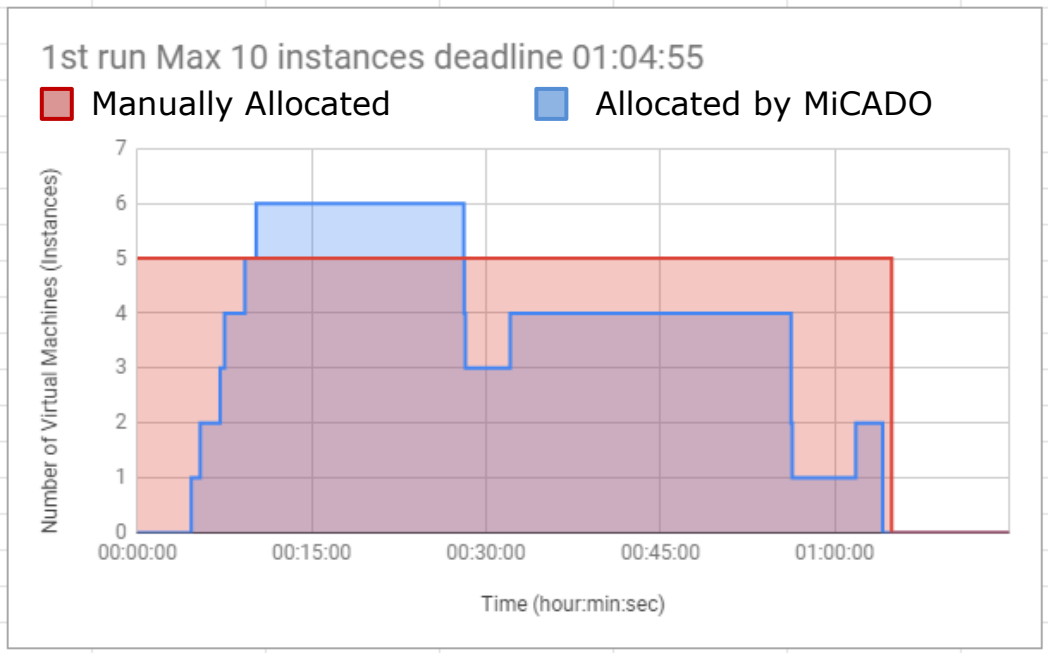


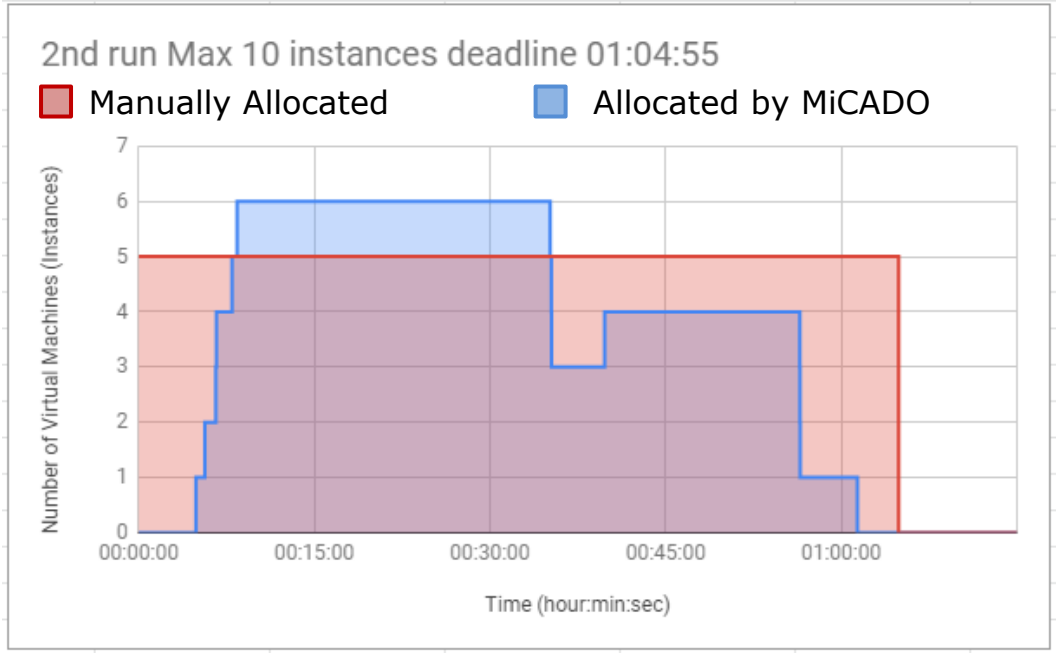
Automatic job allocation (MiCADO)



Results

- Dynamic allocation of variable length jobs results in a better use of cloud resources





5 VMs
Manual allocation (baseline)

3.86 VMs
Dynamic allocation (MiCADO)

Thanks!

- github.com/micado-scale/ansible-micado
- project-cola.eu/
- T. Kiss, J. DesLauriers, G. Gesmier et al., A cloud-agnostic queuing system to support the implementation of deadline-based application execution policies, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.05.062

COLA
Cloud Orchestration at the Level of Application

Project Director: Dr. Tamas Kiss, University of Westminster, UK

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731574