Development and evaluation of a Kubernetes cluster simulator based on Batsim

Presented by: Théo Larue
Supervised by: Olivier Richard & Michael Mercier

Université Grenoble Alpes
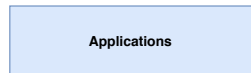
August 31, 2020

Table of contents

Introduction

Resource and Jobs Management System

The RJMS is at the core of the cluster.

Examples of RJMS

- OAR
- SLURM
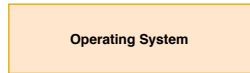- HadoopYARN
- Apache Mesos

Applications

Infrastructure

Runtimes

Resource and Job Management System
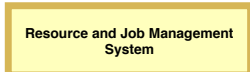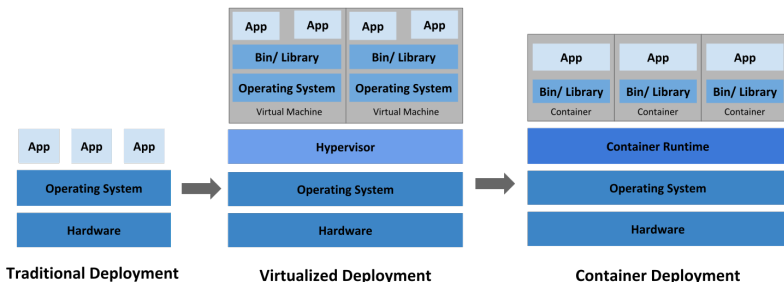
Operating System

Network

Hardware

Kubernetes

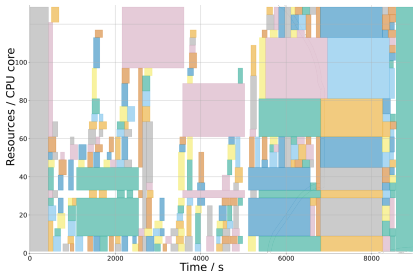## Kubernetes in a nutshell

- Open source resource manager for containerized applications
- About 2M lines of code
- 2.8k contributors

Traditional Deployment   Virtualized Deployment   Container Deployment

source: https://kubernetes.io/docs/

A component of the RJMS: the scheduler

Scheduling is the act of allocating tasks to resources.

Numerous factors

- Workloads
- Applications
- System size
- Network topology
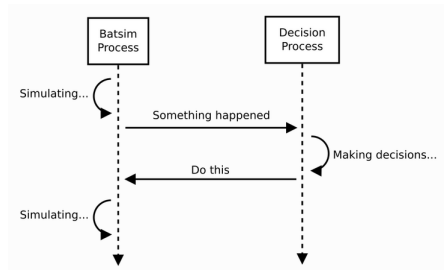- Energy consumption
- Scheduling policies

Complex implementations: Kubernetes default scheduler weighs **47k lines of code**.

Studying RJMS

Different approaches

- Analytical study
- Real experiments
- Emulation
- **Simulation**

Batsim, an infrastructure simulator
aimed at studying RJMS.

Contribution

Batsim
- Event based
- Own protocol
- Dilated time

kubernetes
- Constant API requests
- Own protocol
- (Real) machine time

Batkube supports

- Any Go scheduler
- Any cluster size
- Resource requests
- Non parallel tasks

Batsim ←----→ **Batkube** ←----→ **kubernetes**

Literature review

Infrastructure simulators

	Grid	HPC	Cloud	P2P	Volunteer
SimGrid	✓				
GridSim	✓				
LogGOPSim		✓			
BigSim		✓			
CloudSim			✓		
GroudSim	✓		✓		
PeerSim				✓	
OverSim				✓	
SimBA					✓
SimBOINC					✓

Domain specific simulators

SimGrid

- Framework for building simulators
- Versatile, accurate and scalable
- 20 years of experience
- Simple analytical models

Simulators for the study of RJMS

Often *ad hoc* simulators

"Publish and perish" - Milian Poquet

Some active projects

- YARNSim
- SLURM simulator

	Scheduler	Platform	Job model
Accasim	Internal	Ad hoc	Static duration
Alea	Internal	GridSim	Static duration
Batsim	Custom Protocol	SimGrid	SimGrid models

Simulation of RJMS

Kubernetes cluster simulation

Kubernetes simulation projects

	joySim	k8s-cluster-simulator
Origin	private (JD.com)	student project
Availability	closed source	open source
Focus	service	batch processing
Scheduler	any	user implementation
Models	mock nodes	static job durations

Notable Kubernetes schedulers

- kube-batch

- Poseidon (Firmament)
- **kube-scheduler**

Integrating Kubernetes schedulers to Batsim
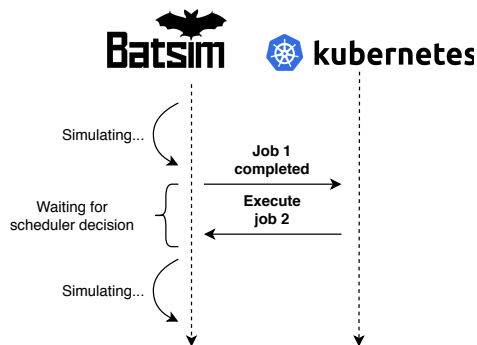
Different communication paradigms

source: https://batsim.readthedocs.io
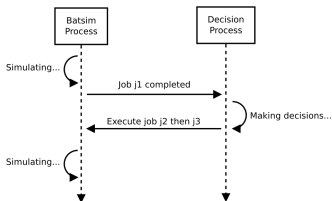
- Event based
- Simulation time

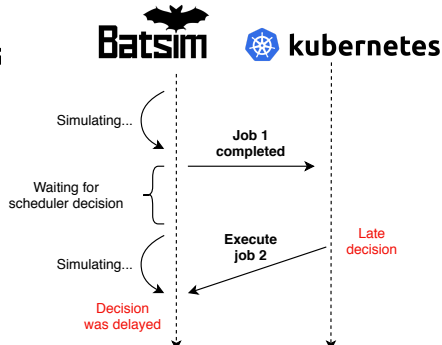source: https://kubernetes.io/docs/concepts/overview/components/
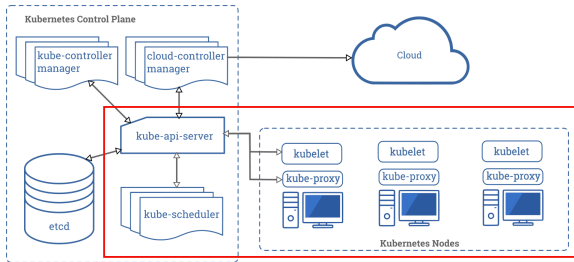
- Central API
- Real time

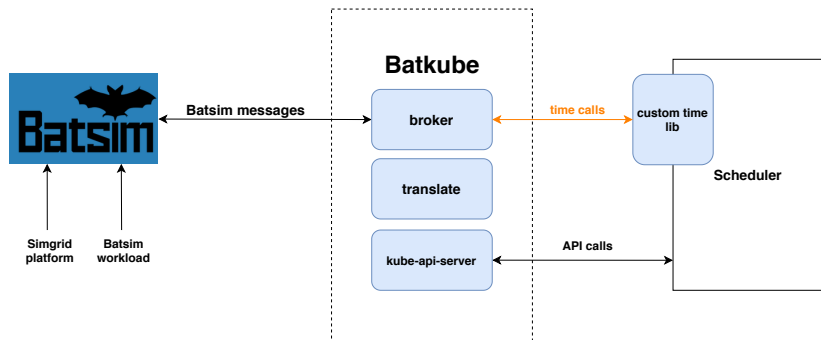# Time synchronization

Scenario 1: correct synchronization

Scenario 2: delayed decision

Technical challenges

Challenges to tackle

1. Integration with Kubernetes
2. Scheduler time interception
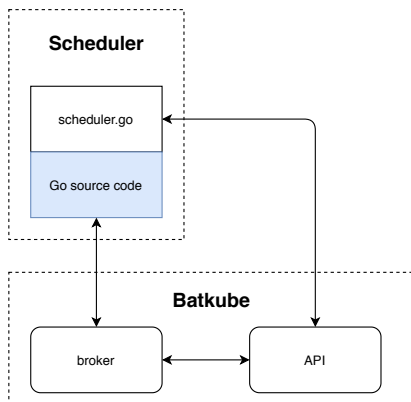3. Time synchronization

Architeture of Batkube

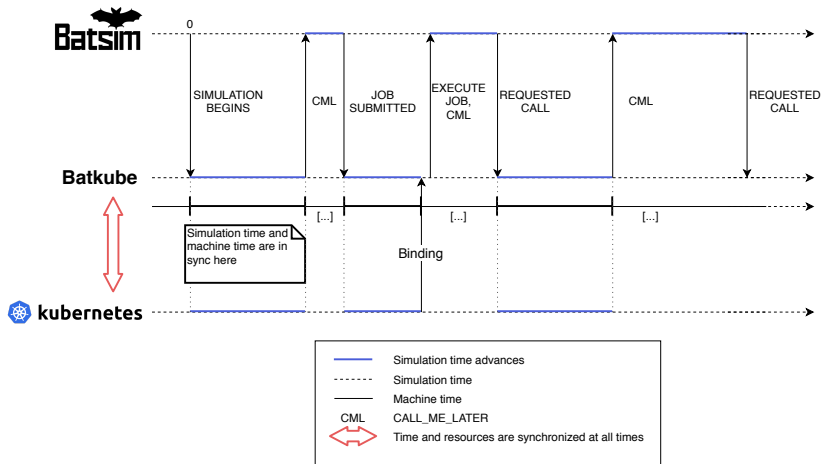

Global architecture of Batkube.

Time interception

Redirection of time calls

- Specific functions are redirected
- Automatic source code manipulation using AST
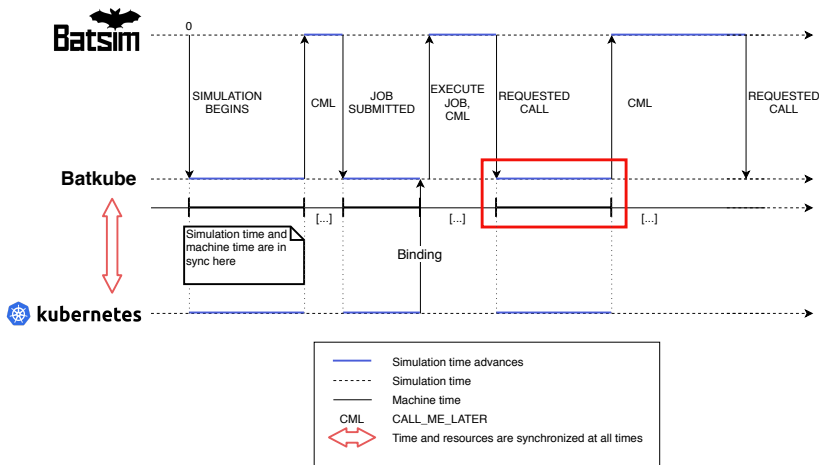- Ensured compatibility with the rest of the code

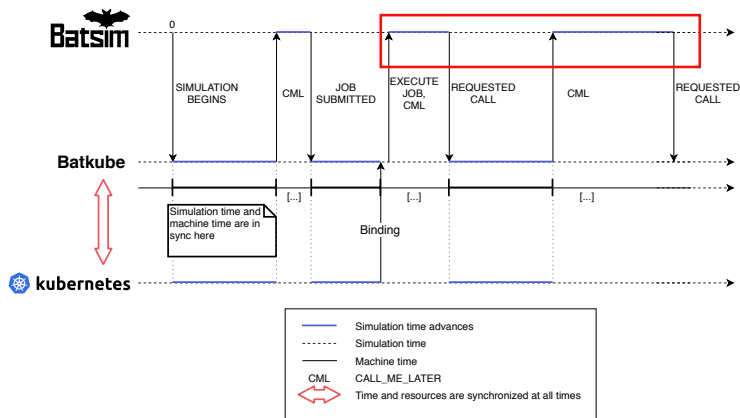Schedulers are patched to redirect their time.

Time synchronization

Time synchronization between Batsim and the scheduler

Parameters of the synchronization I

Timeout value

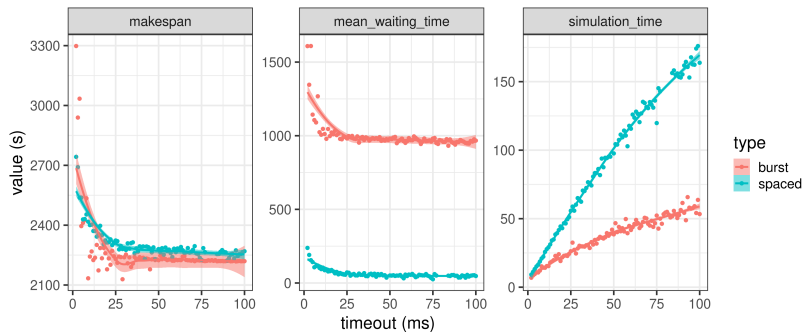# Parameters of the synchronization II

Simulation time step $\in$ [`base-simulation-timestep`, `max-simulation-timestep`]
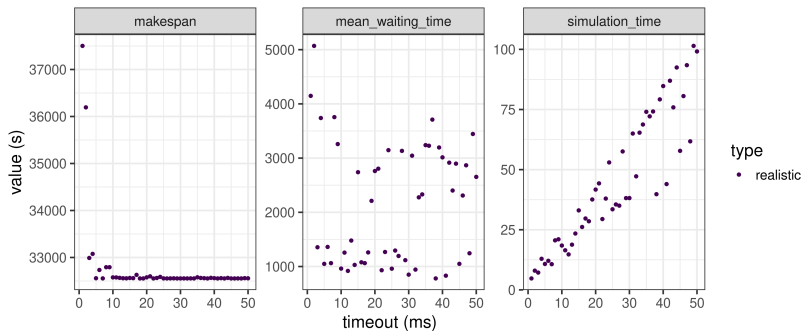
Study of the simulator

Experimental design

TODO: Scheduler used, platforms and workloads tested, what experiments (parameters, metrics studied, repetitions)
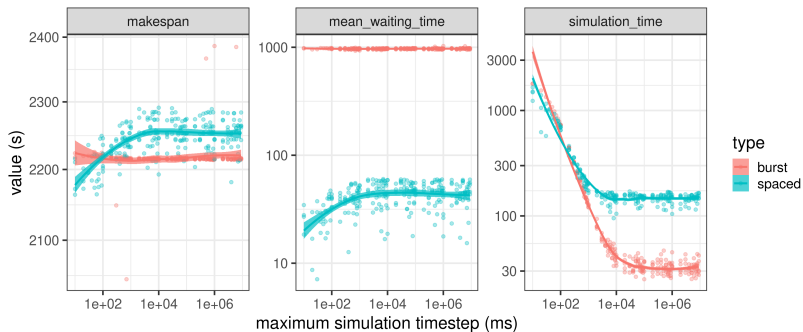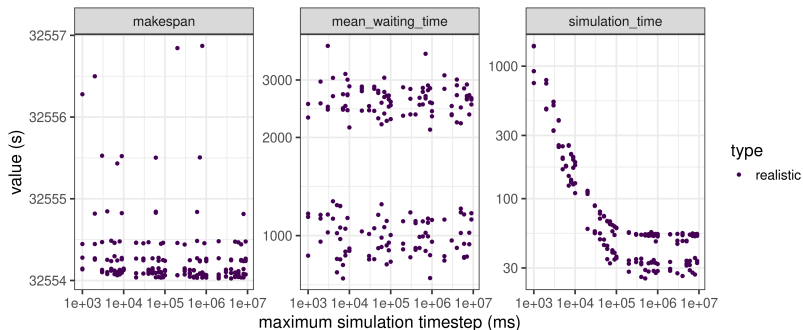
Timeout I

Timeout II

Maximum simulation timestep I

Maximum simulation timestep II

Experimentation on a real cluster

platform specifications

workload.json

start-cluster.sh

submitter.go

Resource watcher | Job submitter

client-go

out0_jobs.csv

out1_jobs.csv

out2_jobs.csv

...

Emulated cluster
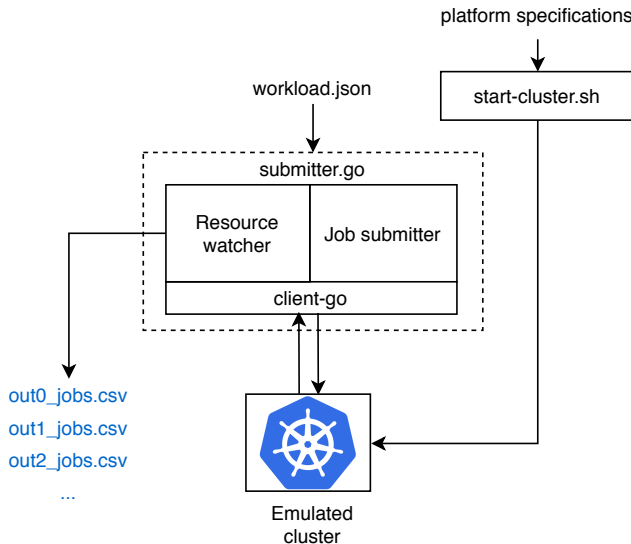
Deviation with reality

workload	makespan				mean waiting time			
	emulated		simulated		emulated		simulated	
	μ	σ	μ	σ	μ	σ	μ	σ
burst	2467	28.3	2215 (-252)	0.508	1077	10.6	970 (-107)	12.6
spaced	2468	5.14	2257 (-211)	16.9	146	1.67	48.1 (-97.9)	9.44
realistic	32556	-	32555 (-1)	1.30	2884	-	2020 (-864)	950

Conclusion

Deviation with reality: can be fixed with some work on the api. Need experiments to measur and quantify this deviation.
max timestep: studying max timestep alone is not enough, need to study it with backoff multiplier.
base time step: need an experiment on it. Too much importance was credited to max timestep, the base timestep might have importance.

Discussion and future work

Capabilities and limitations of Batkube

WIP

Capabilities

- Delay jobs
- Cpu and memory requests
- Can patch any kubernetes scheduler written in Go
- The api only supports the default scheduler

Limitations

- Memory hungry (in fact, the scheduler is memory hungry)
- Some problems with the scheduler
- Not scalable
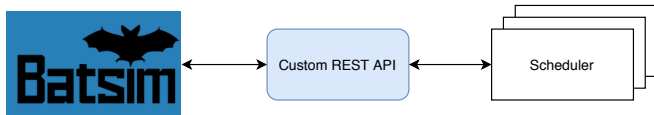
Perspectives for future work

- parallel jobs
- storage
- more complete api: support for more schedulers but also tools (monitoring tools)
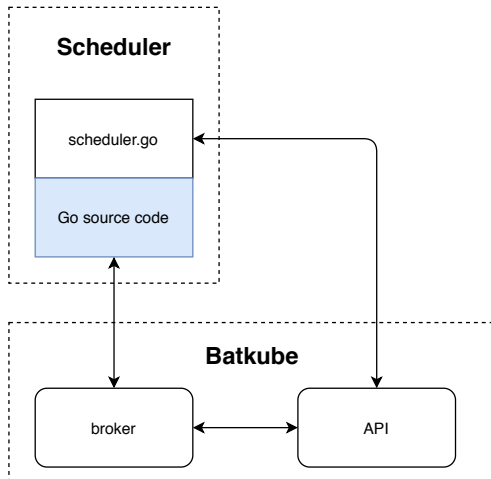
References I

Any questions?

Thank you for your attention!
I am open to any questions.

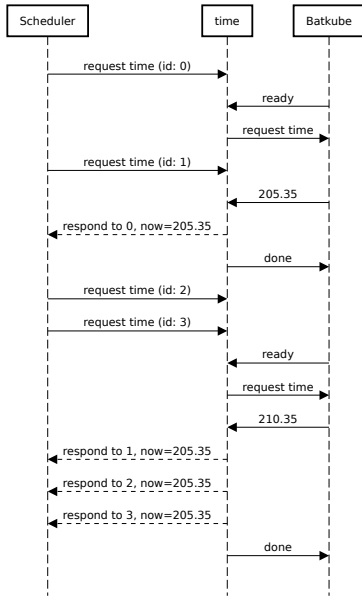# Batkube integration with Kubernetes

Reimplementation of a custom API.
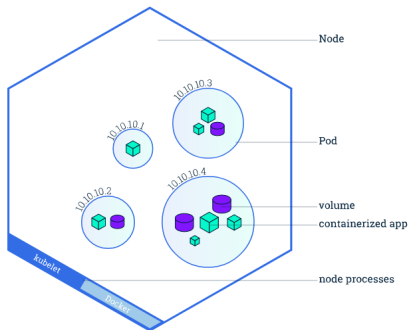
Time interception

Schedulers are patched to redirect their time.

batsky-go

Exchanges between the scheduler, batsky-go ("time") and Batsim

Similar resources

source: `https://kubernetes.io/docs/tutorials/`
`kubernetes-basics/explore/explore-intro/`

Translation between Kubernetes and Batsim

- A Pod = a job.
- A Node = a compute resource.