Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator

Pierre-François Dutot[1], Michael Mercier[1,2], Millian Poquet[1], Olivier Richard[1]

DATAMOVE Team, LIG

20th JSSPP Workshop, May 27 2016, Chicago

HPC cluster management

Resources and Jobs Management Systems (RJMS)

- AKA batch scheduler
- Orchestrates resources on HPC clusters
  - Implements scheduling policies
  - Manages parallel jobs
  - Enforces energy policy
- Examples: SLURM, OAR, TORQUE, PBS...

RJMS Facts

- Large scale: from 100 to 100 000 nodes
- Critical production system: downtime is $
- Need to be energy efficient: HPC energy consumption in MW

Improve the RJMS's schedulers

Many parameters to consider:

- scheduling policies
- platform
- workloads
- metrics

We need to experiment on the RJMS but...

Production systems are not available for testing RJMS

- They are already full of users jobs!
- Energy/time cost of experiments is not affordable

Simulation at rescue

- Easy to select workloads, platforms
- Fast comparison of algorithms
  - Real: 33 compute nodes for 3 hours
  - Simu: 1 laptop for 3 seconds
- Closest approches
  - ALEA (Klusáček and Rudová)
  - INSEE based simulator (Pascual et al.)

Common difficulties

- Precision vs. Performance
- Validation: comparison with reality

Batsim: **Bat**ch scheduler **sim**ulator

Focus on realism, facilitate comparison

Batsim's approach

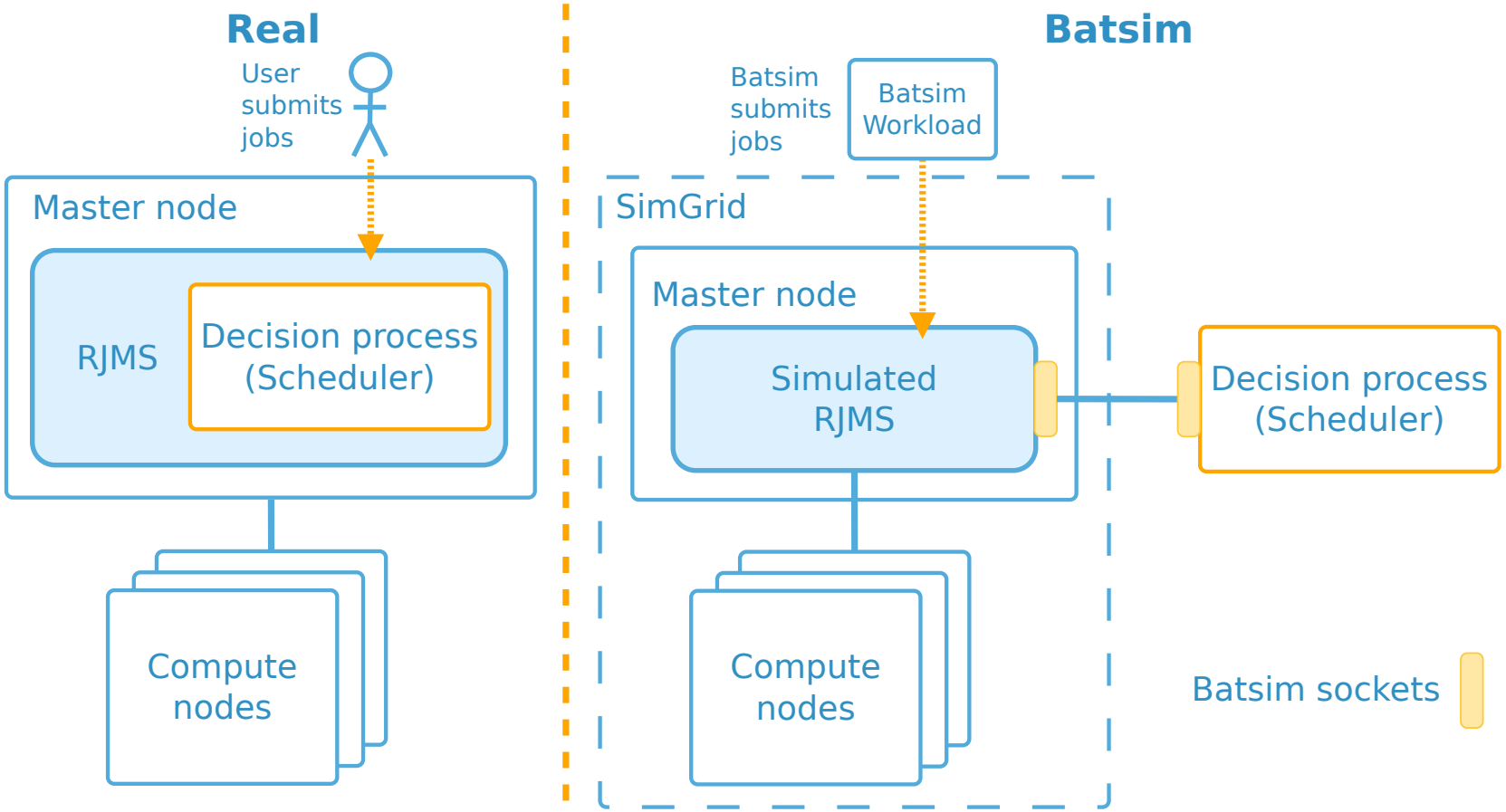
- Realistic network model
  → **SimGrid network models, topologies**
- Multiple job models
  → **From simple to realistic**
- Openness
  → **Open source and documented**
- Modularity
  → **Language-agnostic text-based protocol**
- Easy to evaluate and compare
  → **Any event-based algorithm can be plugged**
  → **Including real RJMS ones!**

Outline

Batsim Overview

- Core program ($\simeq$ 3k lines of C++ code)
- Several schedulers already adapted
  - In Python, Perl, C++
  - More than 15 scheduling and energy policies
  - From production and research
- Simulates a RJMS and the compute nodes
- Scheduler in another process
- Based on the SimGrid framework
  - Reliable: Has been used for 15+ years, strong community
  - Scalable
  - Topology aware network models

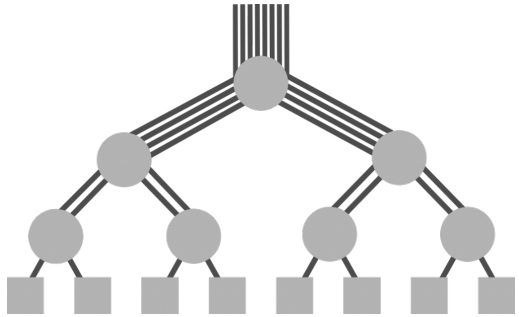
Real vs. Simulated



Batsim inputs

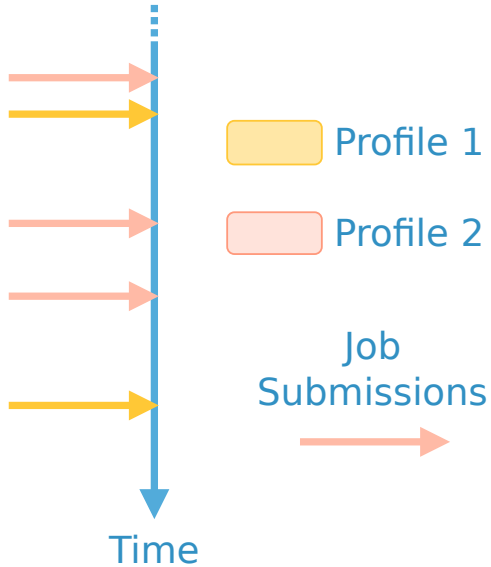
What is a Batsim platform?

- Batsim platform ≃ SimGrid platform



What is a Batsim workload?

- List of jobs
  - Submit time
  - Walltime
    (user-given maximum runtime)
  - Required resources
- Each job is associated to a profile



Job Profile types

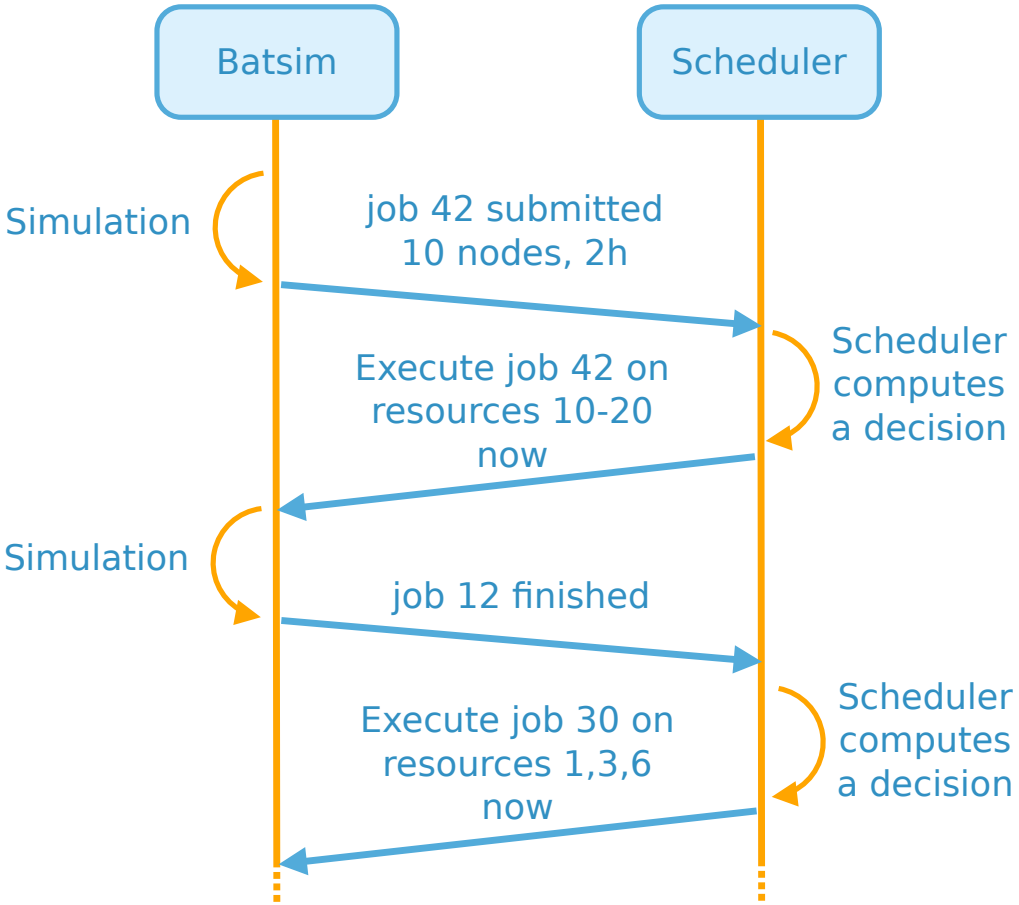
Delay
- Fixed amount of time

MSG
- A computation vector (1D matrix)
- A communication 2D matrix

Sequence
- A sequence of profiles
- Repeated n times
- à la BSP[1]

[1] Bulk Synchronous Parallel model

Communication Protocol



Protocol characteristics

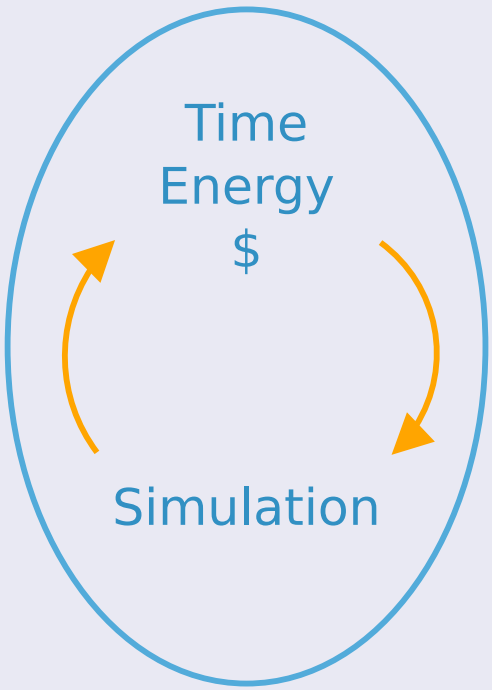
- Textual
- Synchronous
- Request-Reply pattern
- Unix Domain Socket
- More info in the doc

Outline

Evaluation process

Same scheduler run **in vivo** and **in simulo**
→ Compare the metrics

A chicken and egg situation



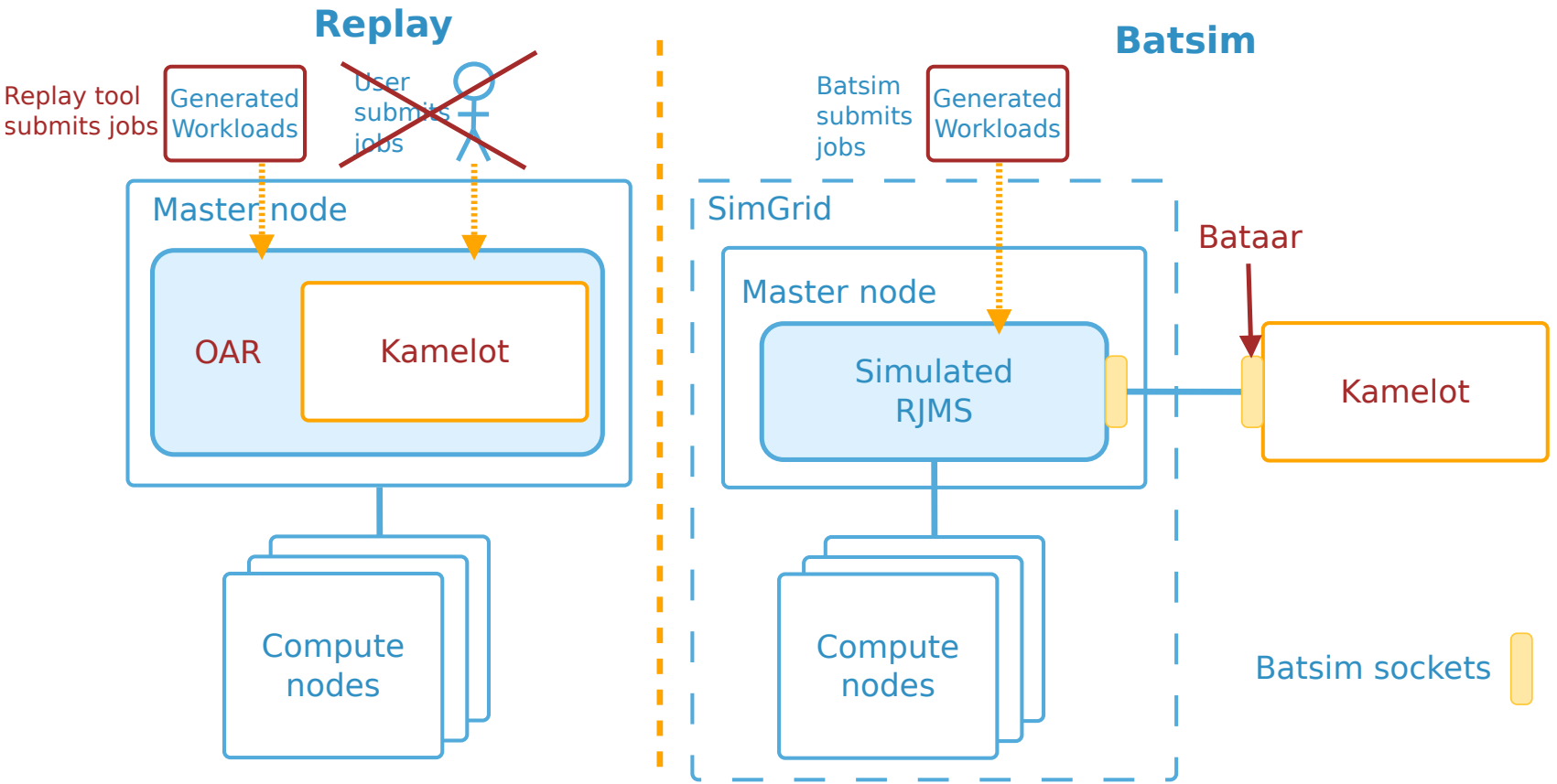
We need simulation to save $
But we need $ to validate the simulator

Our evaluation process is **reproducible**

Experiment design - in vivo vs. in simulo

	In Vivo	In Simulo
Platform	Grid'5000 testbed cluster	SimGrid XML platform
	33 nodes under 1 switch	
Jobs	NAS Parallel Benchmarks Type: IS, FT and LU	Job profile generated from real execution instrumentation
	Size: from 1 to 32 nodes Total: 47 different jobs	
Workloads	Contains 800 jobs randomly picked Inter arrival times: Weibull dist. ($shape = 2$, $scale = 15$) Job size: $2^{\lfloor u \rfloor}$, $u \sim$ Lognormal dist.($\mu = 0.25$, $\sigma = 0.5$) Total: 9 workloads ≈4h each	
	2 runs/workload	1 deterministic simulation
Scheduler	Kamelot: scheduler from OAR (conservative backfilling)	
	Directly executed by OAR	BatAar: BatSim Adaptor for OAR
Resources	10 000+ hour×cores	Only 30s on a laptop

Evaluation instantiation



Gantt charts is not enough

Use metrics

We need aggregated metrics:

Makespan AKA Cmax

$$max_i(FinishT_i)$$

Mean waiting time

$$\frac{1}{nbJobs} \sum_i StartT_i - SubmitT_i$$

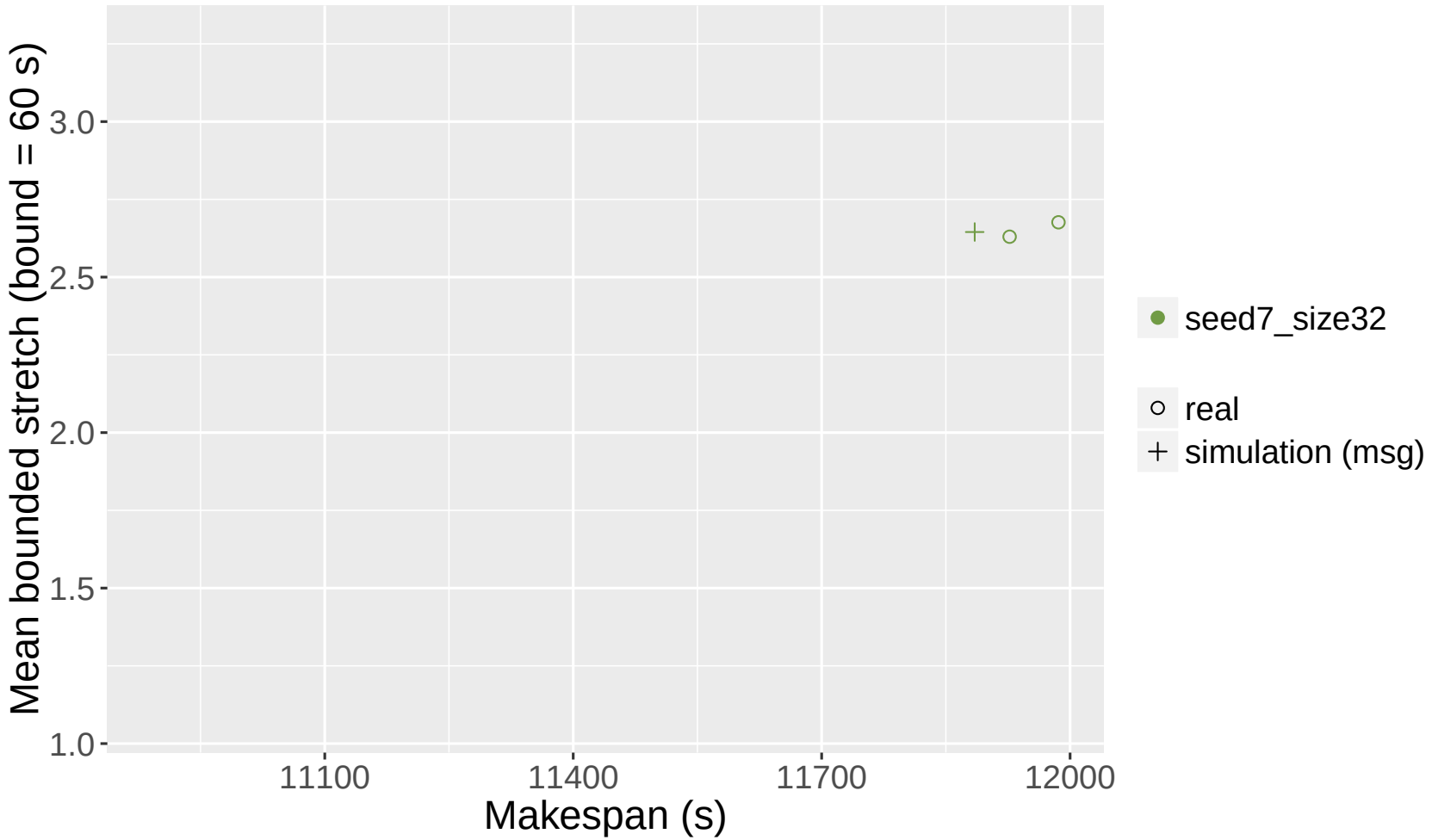
Mean bounded slowdown

$$\frac{1}{nbJobs} \sum_i max(\frac{FinishT_i - SubmitT_i}{max_i(K, FinishT_i - StartT_i)}, 1)$$

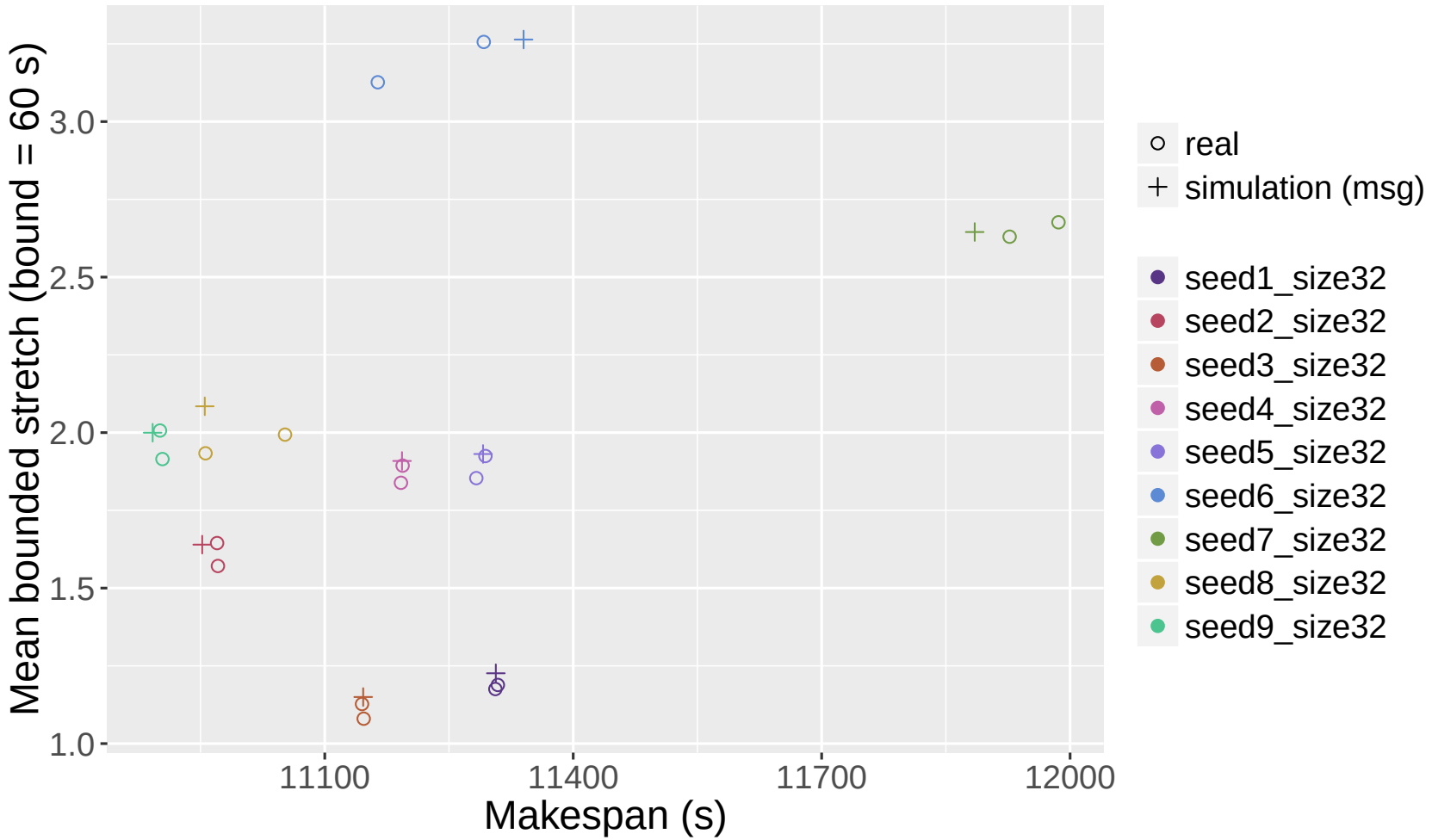
Real life is not deterministic:

- Every real execution different
- Simulation is deterministic
- → It must be representative

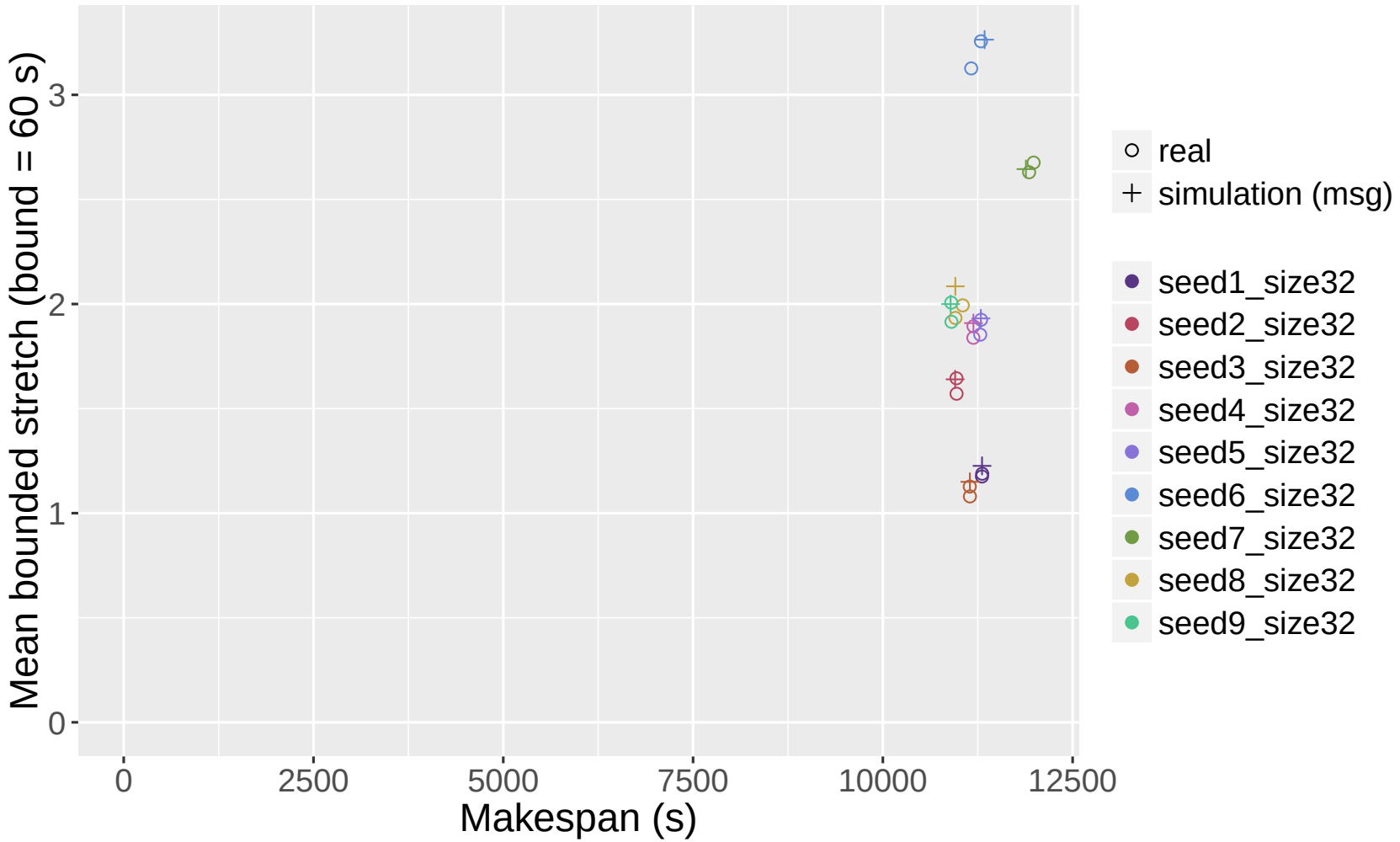
Metrics comparison - One workload



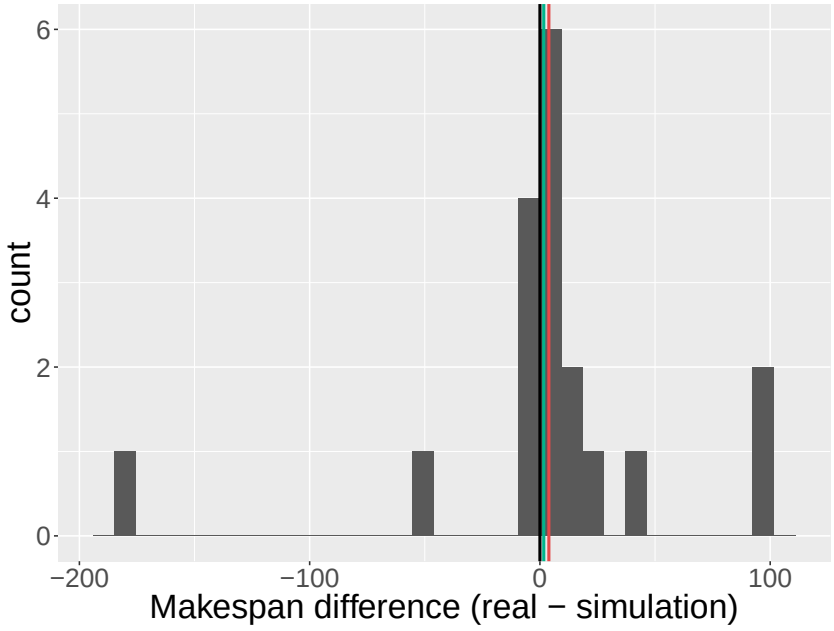
Metrics comparison - All workloads

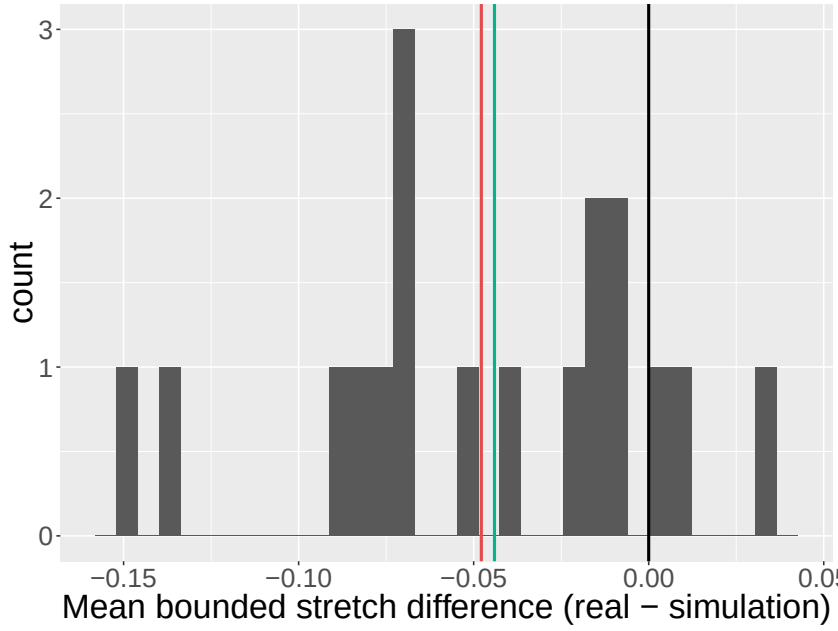


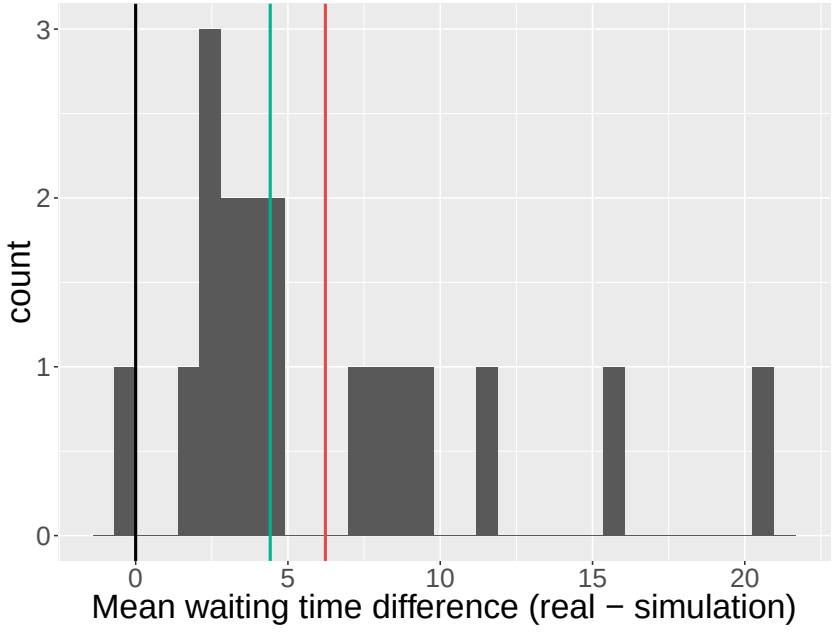
Metrics comparison - Zoom out



Metrics comparison - Difference overview







Differences

- Waiting times are underestimated
- **Beware of overfitting OAR's behaviour**

Outline

Results discussion

Done

- Same scheduler run in vivo and in simulo
  - Delay and MSG model
  - Similar behavior on classic metrics
  - Small workloads
  - Small homogeneous platform

Next steps

- Test other schedulers in vivo and in simulo (e.g. Flux scheduler)
- Characterize real experiment noise ($$) (in progress)
- Heterogeneous platforms ($$)
- Big platforms ($$$$$)
- Validate the energy models
- Include IO simulation (→ big data workload simulation)

Thanks!

Batsim:
https://github.com/oar-team/batsim

Contacts

michael.mercier@inria.fr
millian.poquet@inria.fr

References:

- Dalibor Klusáček, Hana Rudová. **Alea 2 - Job Scheduling Simulator.** In proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST, 2010.
- Jose A. Pascual, Jose Miguel-Alonso, Jose A. Lozano. **Locality-aware policies to improve job scheduling on 3D tori.** The Journal of Supercomputing, 2015, vol. 71, no 3, p. 966-994.
- D.G. Feitelson. **Workload Modeling for Computer Systems Performance Eval- uation.** Cambridge University Press, Cambridge, 2015.

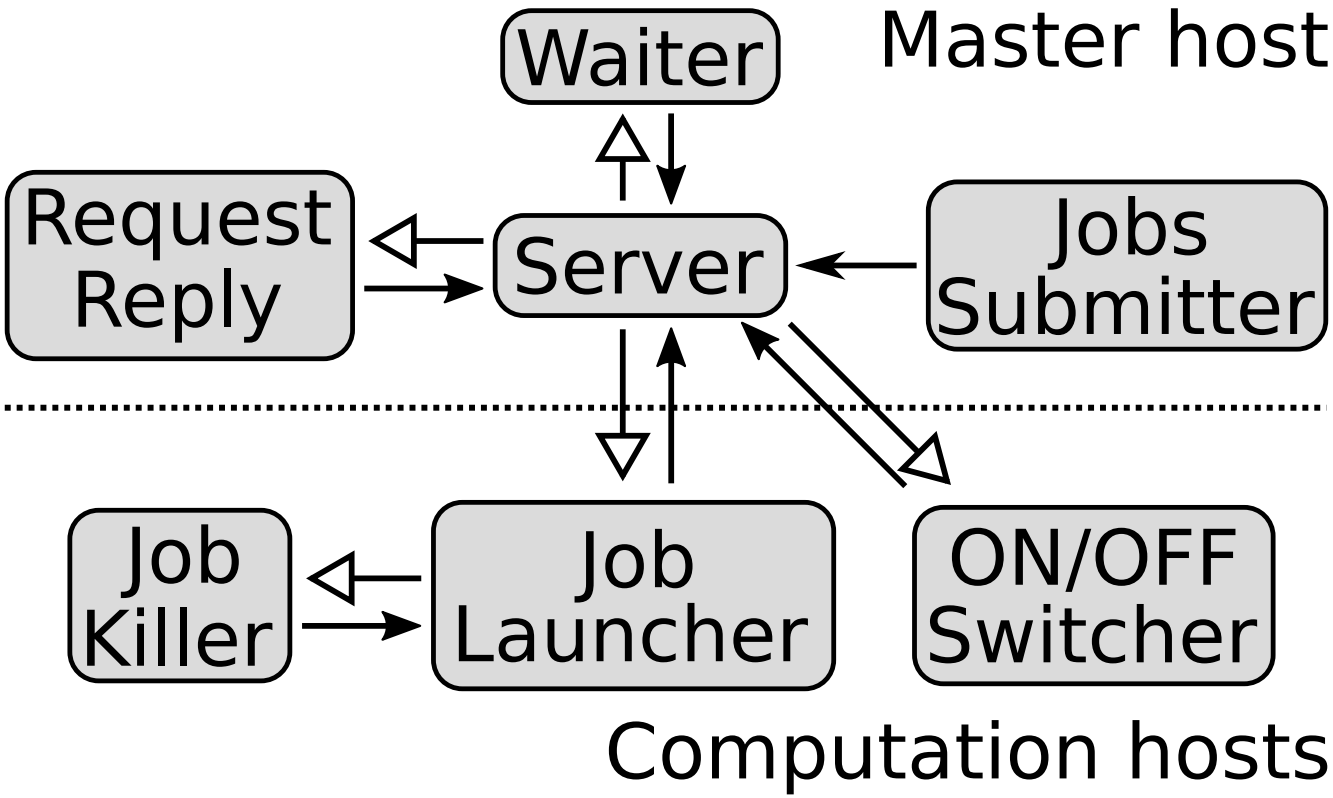
Workload File Example

```
{
  "nb_res": 256,
  "jobs": [
    {"id": 1, "subtime": 10, "walltime": 120, "res": 4, "profile": "is.B.4"},
    {"id": 2, "subtime": 20, "walltime": 100, "res": 4, "profile": "is.B.4"}
  ],
  "profiles": {
    "is.B.4": {
      "type": "msg_par",
      "cpu": [3.9e10, 3.8e10, 3.8e10, 3.8e10],
      "com": [163195200.0, 54499200.0, 54499200.0, 54499200.0,
              163195200.0, 54499200.0, 54499200.0, 54499200.0,
              163195200.0, 54499200.0, 54499200.0, 54499200.0,
              163195200.0, 54499200.0, 54499200.0, 54499200.0],
      "command": "mpirun is.B.4"
    }
  }
}
```

Inner Mechanics overview

What is inside Batsim?

- Batsim is NOT a simple event loop
- Batsim = many *processes* which interact with each other



Inner Mechanics details

Why does Batsim use such a design?

- Permit fine-grained RJMS simulation
- More complex, but more modular
- Individual parts are well separated and remain simple
- Specific parts can be detailed, without modifying the other parts (e.g. scheduling time, job launchs...)
- More complex designs can be done (e.g. distributed RJMS...)