

Integration of an event-based simulation framework into a scientific workflow execution environment for Grids and Clouds

Simon Ostermann, Kassian Plankensteiner,
Daniel Bodner, Georg Kraler and Radu Prodan
University of Innsbruck, Austria,
Institute of Computer Science
simon@dps.uibk.ac.at



Overview

— [Introduction

— Groudsim

— ASKALON

— [Architecture

— [Evaluation

— [Conclusion

Groudsim

— [Event based simulation framework

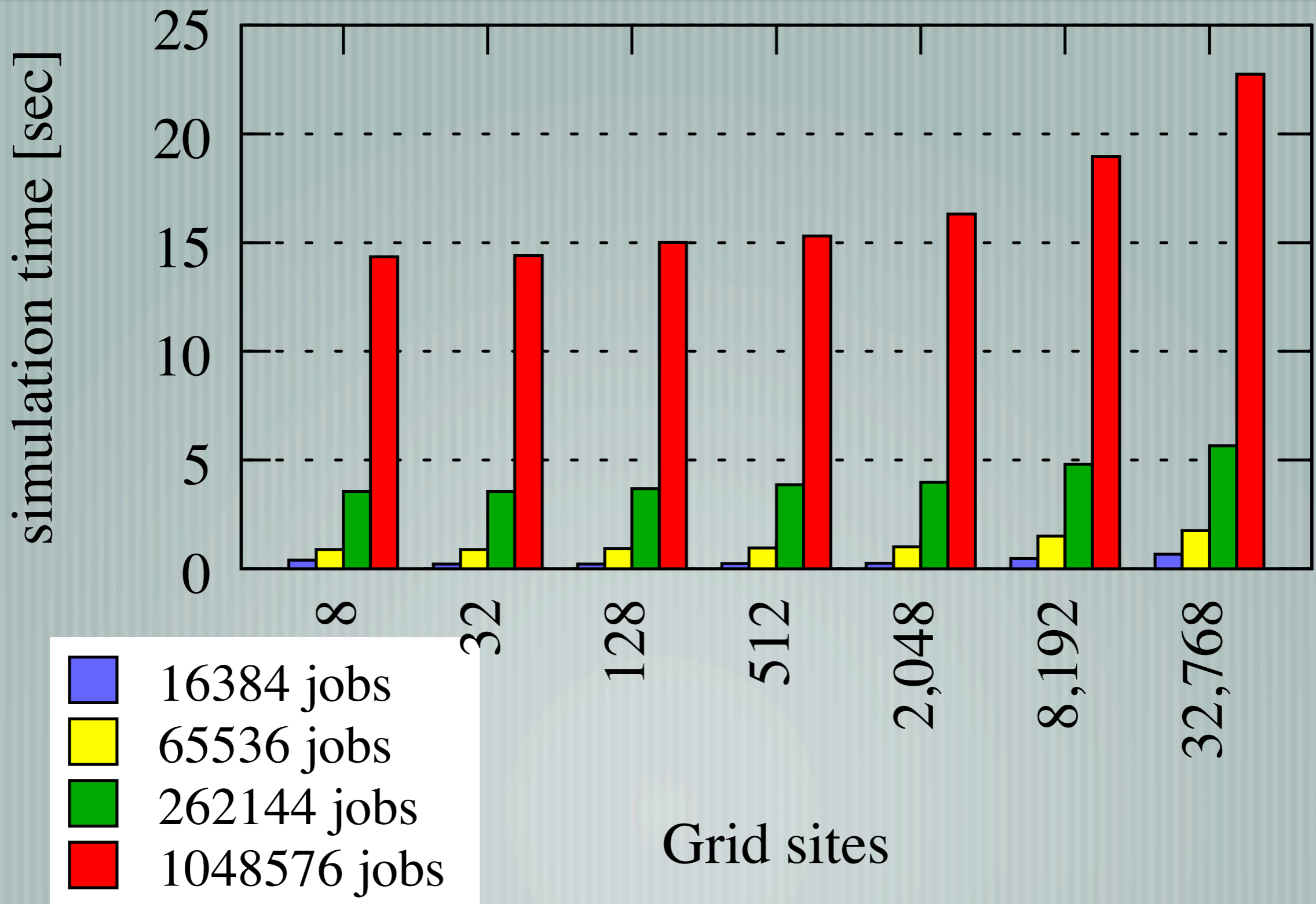
— [File transfer and computational jobs simulation

— [Grid and Cloud resources with accounting

— [Failure generator to allow realistic analysis of recovery mechanism

— [Highly scalable compared to existing (Java) Simulators

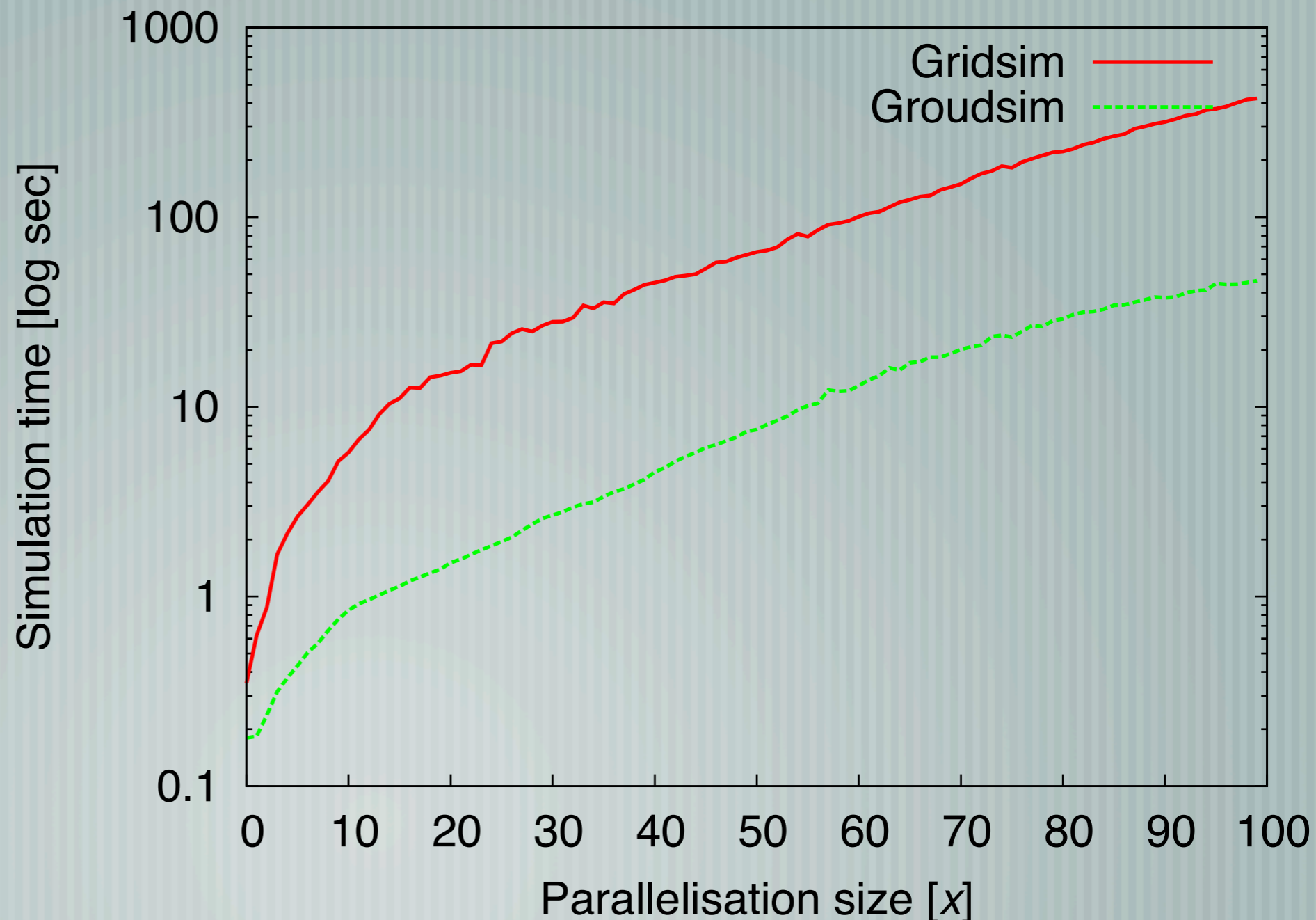
Groudsim scalability



Groudsim comparison

— [Simulating
the same Scenario
on Gridsim and
Groudsim

— [Workflows
with 71 to 6902
tasks ($69 * x + 2$)



ASKALON

— [Environment to execute scientific workflow applications on Grid and Cloud resources

— [Resource & Application management with high level abstraction

— [Execution and performance monitoring and analysis

— [Globus services & Java Webstart GUI

ASKALON Workflow composition

ASKALON Grid Applications Development Environment - /Users/simon/askalon-applications/povray/povray.teuta.mine

File Edit View Insert Format Workflow Resources Tools Provenance Security Help

SubWorkflows

Types Workflows

- Femilab
- grasil
- chemistry
- Montage
- Imaging
 - Function
 - Povray
 - Povray:Convert
 - Povray:Render2
- Astrophysics
- Hydrology
- Meteorology
- Rendering

Quick search

main

```

    graph TD
      Start(( )) --> ParallelFor_1[<<ParallelFor>> ParallelFor_1]
      ParallelFor_1 --> Render2[<<Activity>> Render2]
      Render2 --> Convert[<<Activity>> Convert]
      Convert --> End(((ActivityFinalNode_1)))
    
```

Tree View

- PovrayWorkflow
 - main
 - Control Flow
 - ControlFlow_3
 - ControlFlow_5
 - ControlFlow_7
 - ControlFlow_1
 - ControlFlow_2
 - Data Flow
 - InitialNode1
 - Render2
 - ParallelFor_1
 - Convert
 - ActivityFinalNode_1

Quick search

Overview UML Geometry

UML AGWL

General Workflow Input Properties Constraints

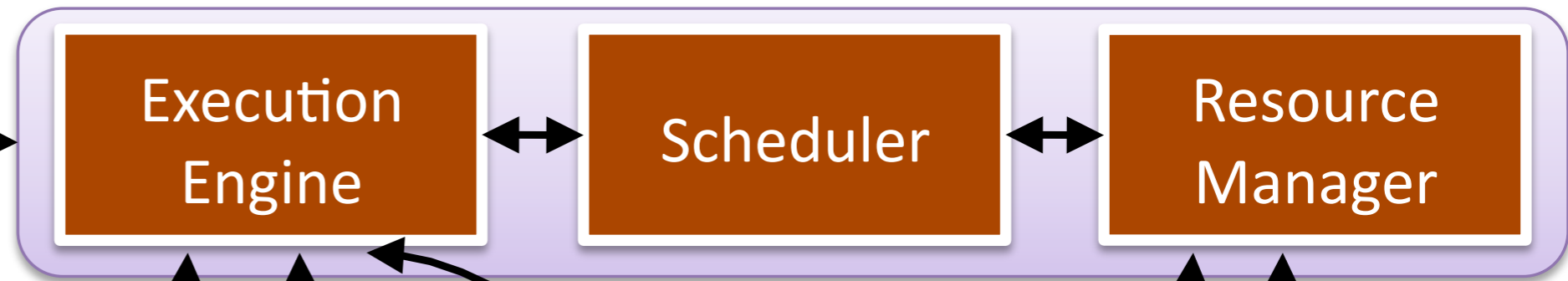
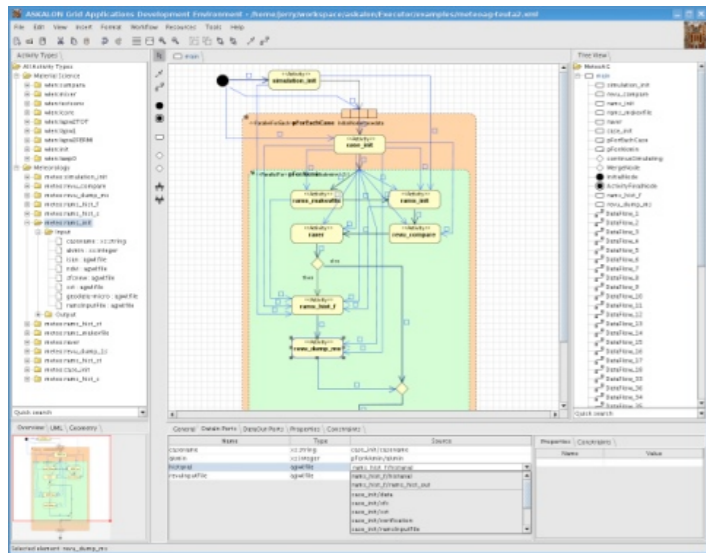
Name	Type	Source
povFile	agwl:file	gsiftp://karwendel.dps.uibk.ac.at//home/kassian...
argFile	agwl:file	gsiftp://karwendel.dps.uibk.ac.at//home/kassian...
iniFile	agwl:file	gsiftp://karwendel.dps.uibk.ac.at//home/kassian...
totalFrames	xs:integer	40
framesPerActivity	xs:integer	2

Selected element: InitialNode1

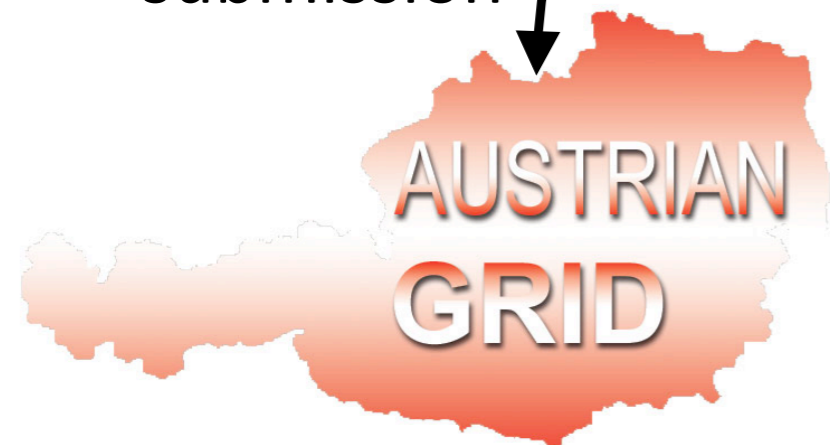
ASKALON Execution

UML Workflow Composition

Runtime Middleware Services



GRAM job submission

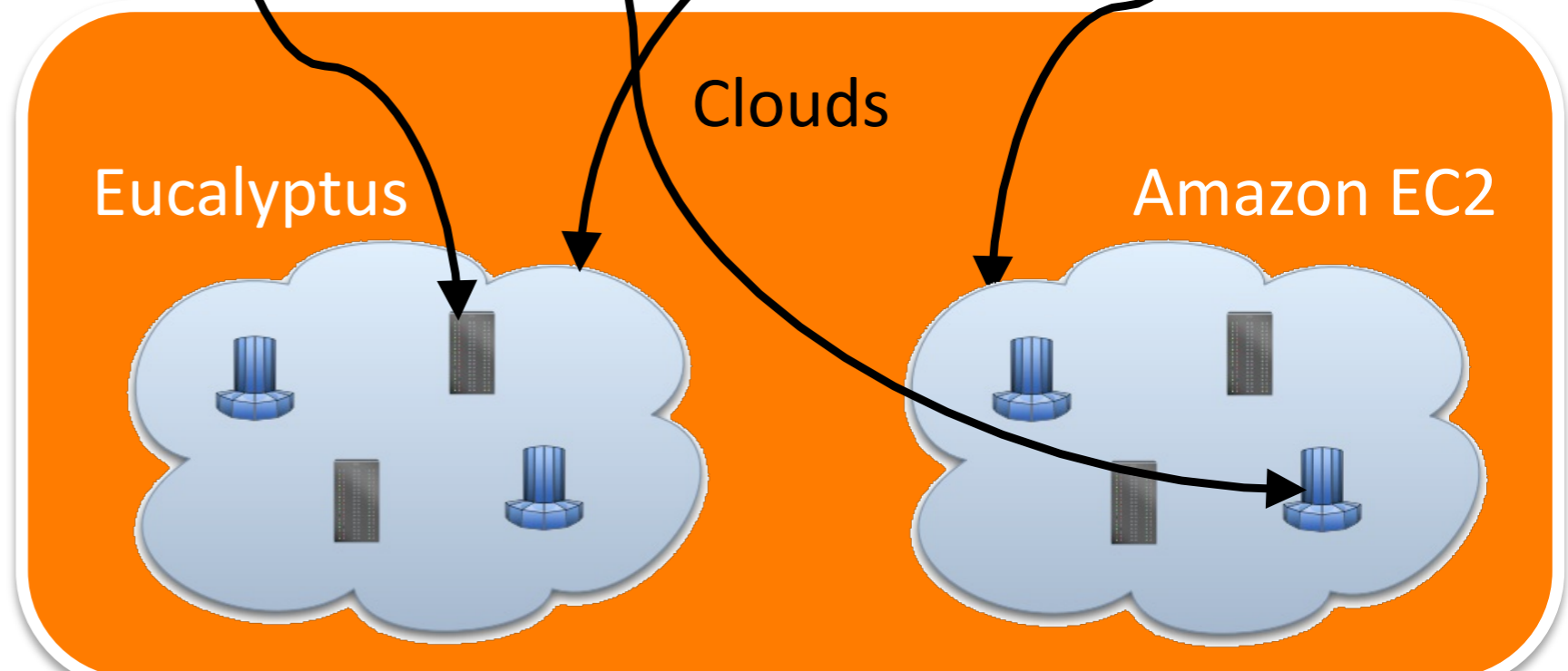


SSH job submission

Schedule jobs

Manage Instances

EC2 API



Architecture

— [Integration of simulator into existing execution environment

— [Resource Management, File transfers and Job executions need to be simulated

— [Context switches between realtime parts of ASKALON and simulation required

— [Control of simulation environment needed

Architecture overview

Workflow Composition
(Teuta)



Simulated Results

Workflow Description

Globus Container

ASKALON

GridARM/GLARE

EE2

GroudSim-ASKALON-Bridge (GAB)

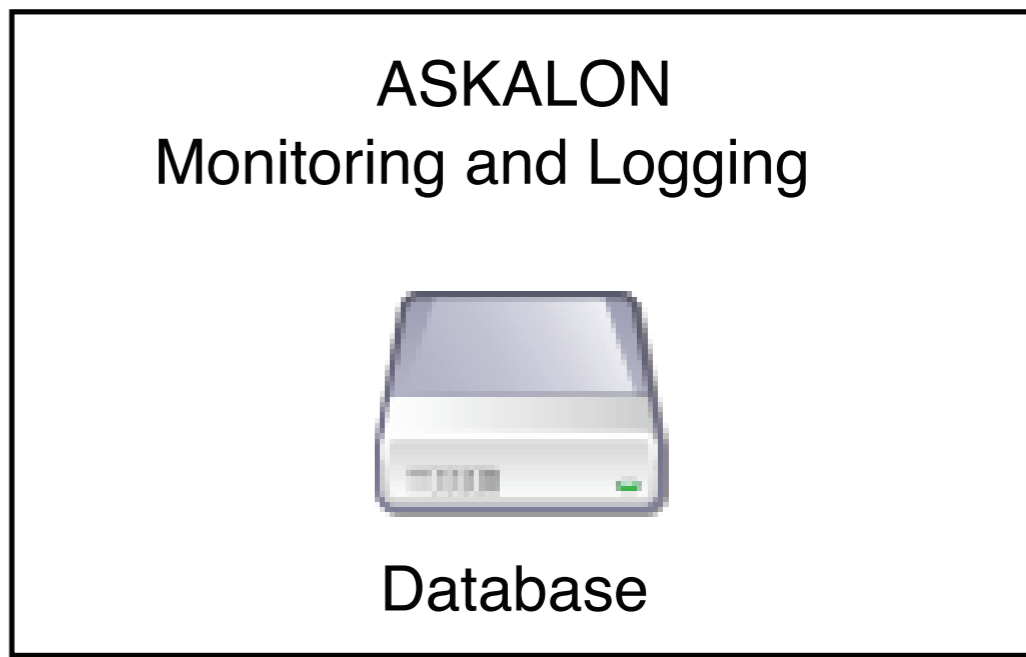
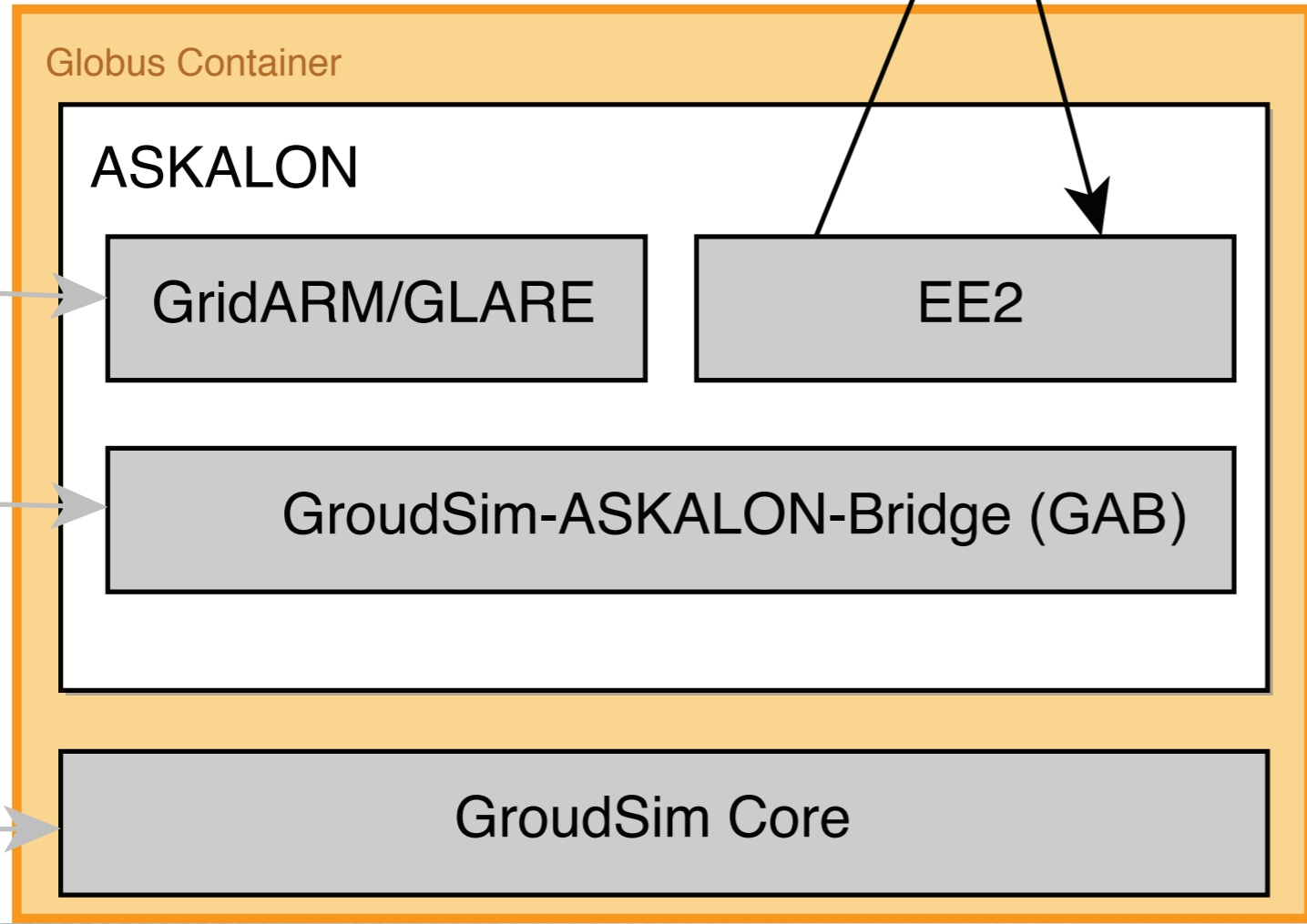
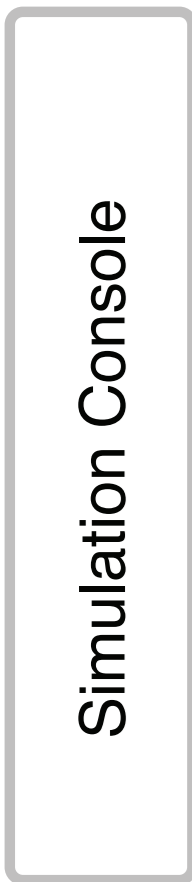
GroudSim Core

ASKALON
Monitoring and Logging

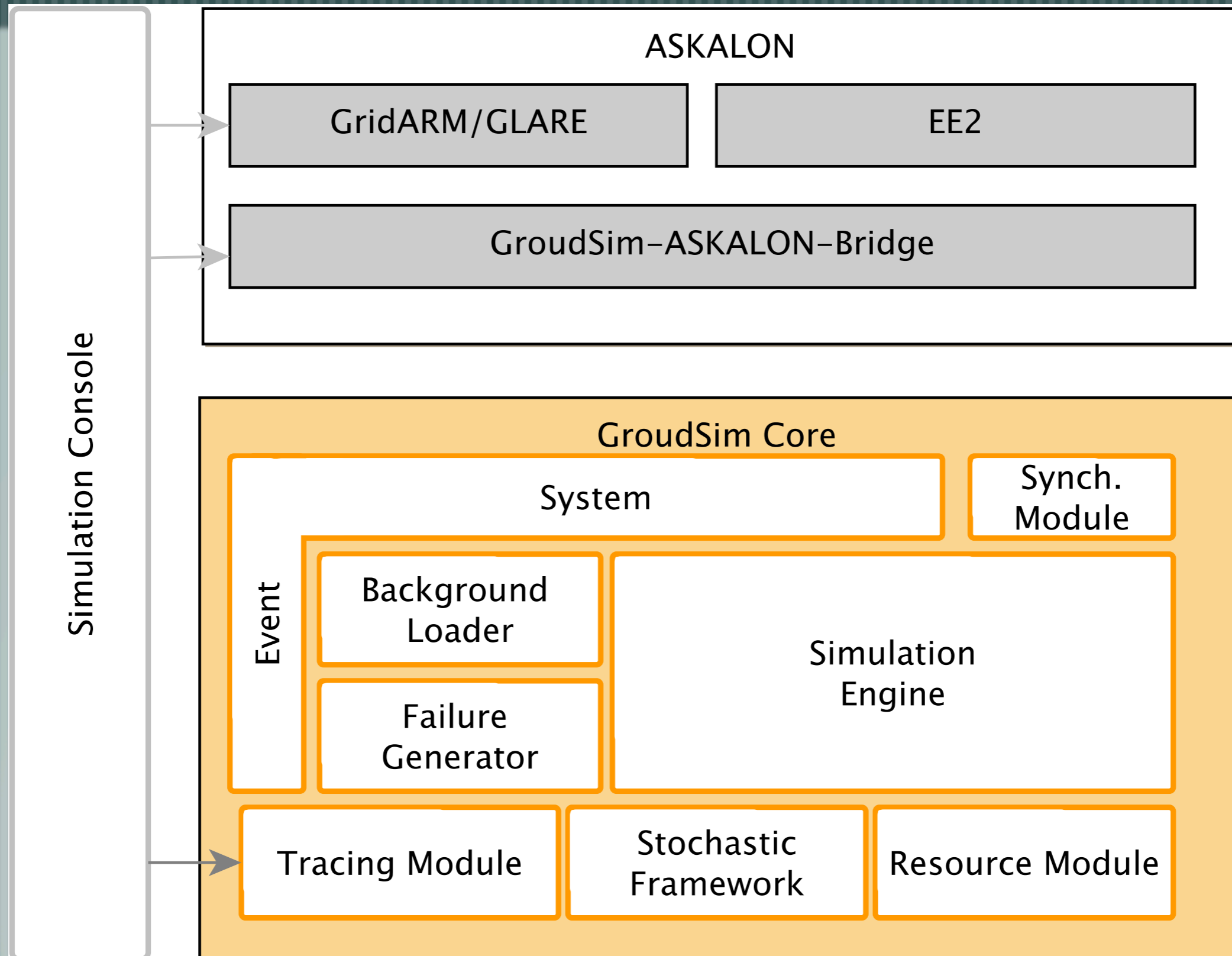


Database

Simulation Console



Architecture details



Synchronization problem: example

— [ASKALON wants to submit 10 jobs to the simulator

— [Limited to 4 parallel job submissions to not flood Grid sites

— [Simulator might start simulating after it gets the first 4 jobs

— [Next 4 jobs arrive at the wrong simulation time resulting in partial sequential execution

— [Solution: ASKALON informs Groudsim when it is finished processing data and the simulation can continue

Evaluation

- [Verify that simulated workflow behaves similar to real execution

- [Analyze execution time spent in

- ASKALON

- Simulation execution

- Synchronization and Context switching overheads

Eval: Real vs. Simulation

— [Using the current available Grid as the simulation input

— [Switching to simulation mode

— [Running (=simulating) the execution

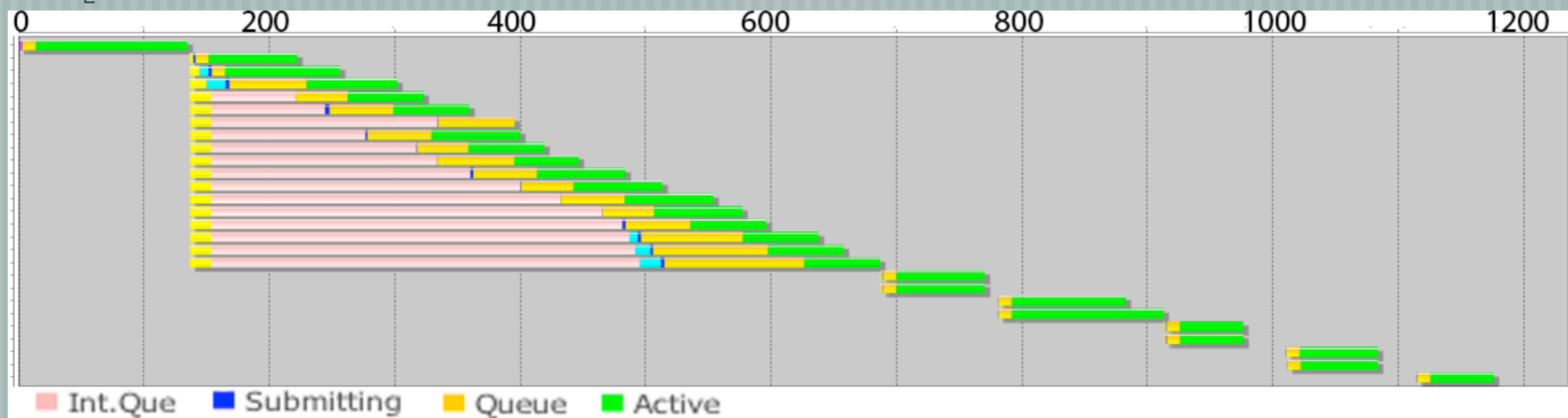
— [Switch back to normal execution

— [Running a real execution

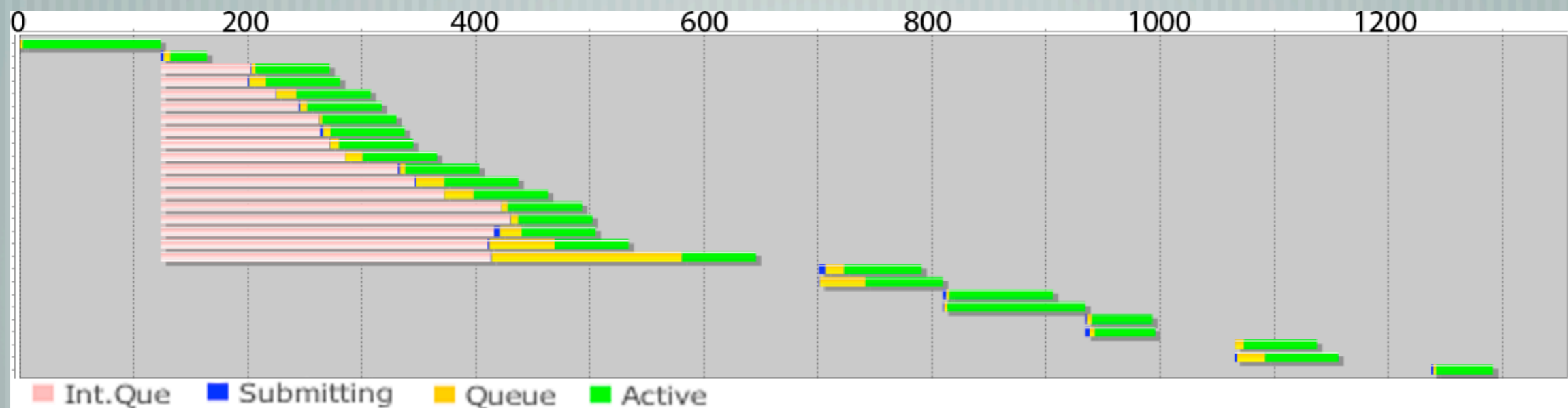
— [Compare performance traces of runs

Eval: Real vs. Simulation II

Simulated results of a workflow



Similar real execution of the same workflow



Eval: Real vs. Simulation II

— [Comparison of the runtime of workflow execution and simulation

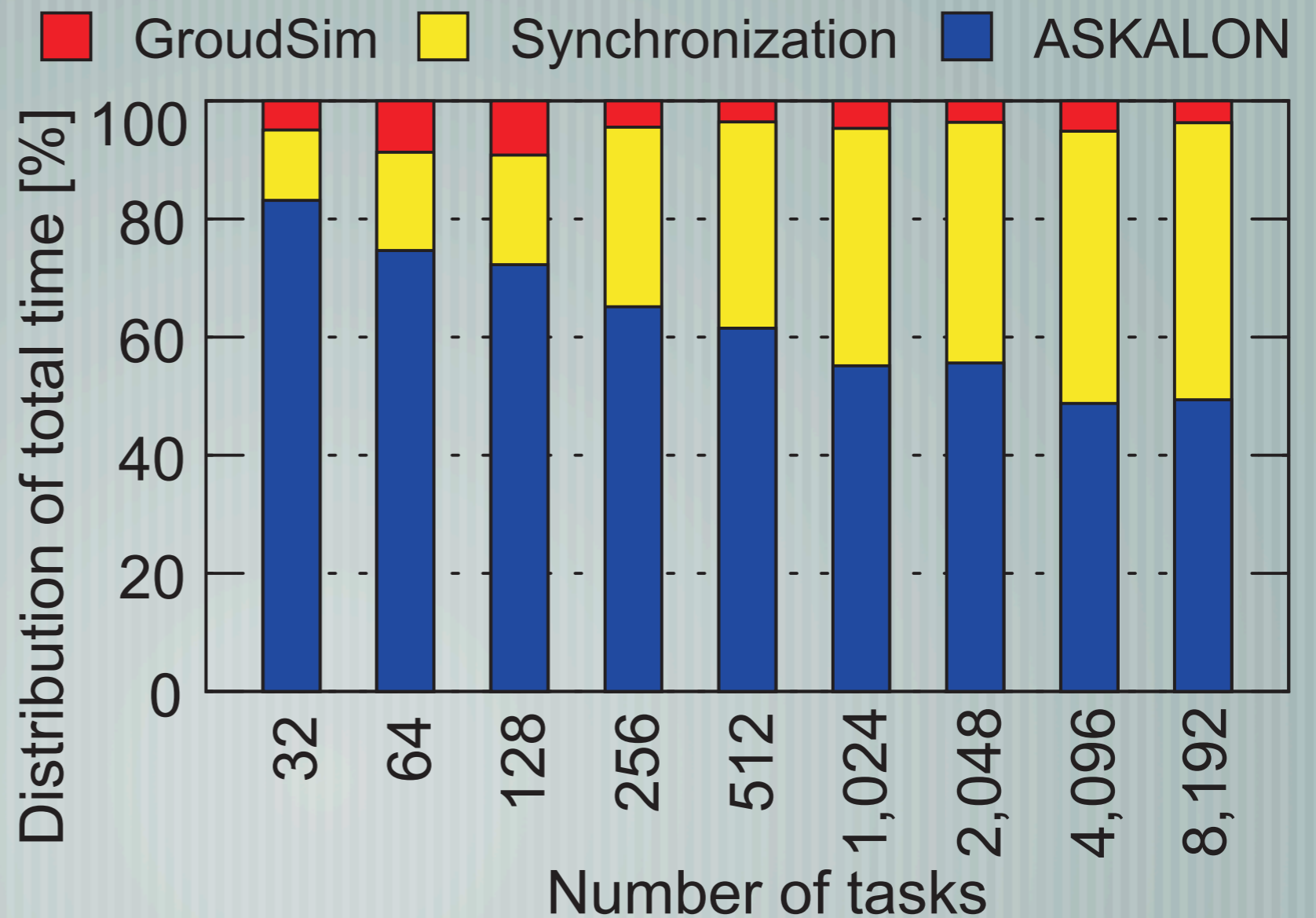
— [Real executions redone using simulator in ASKALON

Workflow name	Number of runs	Total number of activities	Average activities per run	Real execution time [hours]	Simulation time [min.]
Blender	432	131,501	305	41.20	17.2
LinMod	62	100,042	1,620	24.89	12.1
Montage	158	359,946	2,300	78.40	58.5
PovRay	182	78,402	430	1,156.98	9.4
WIEN2k	673	207,256	310	108.21	29.3
Total	1,507	877,293	580	1,409.68	126.0

Overhead analysis

Scenario with 1 Grid CPU -> Context switch for each submitted job

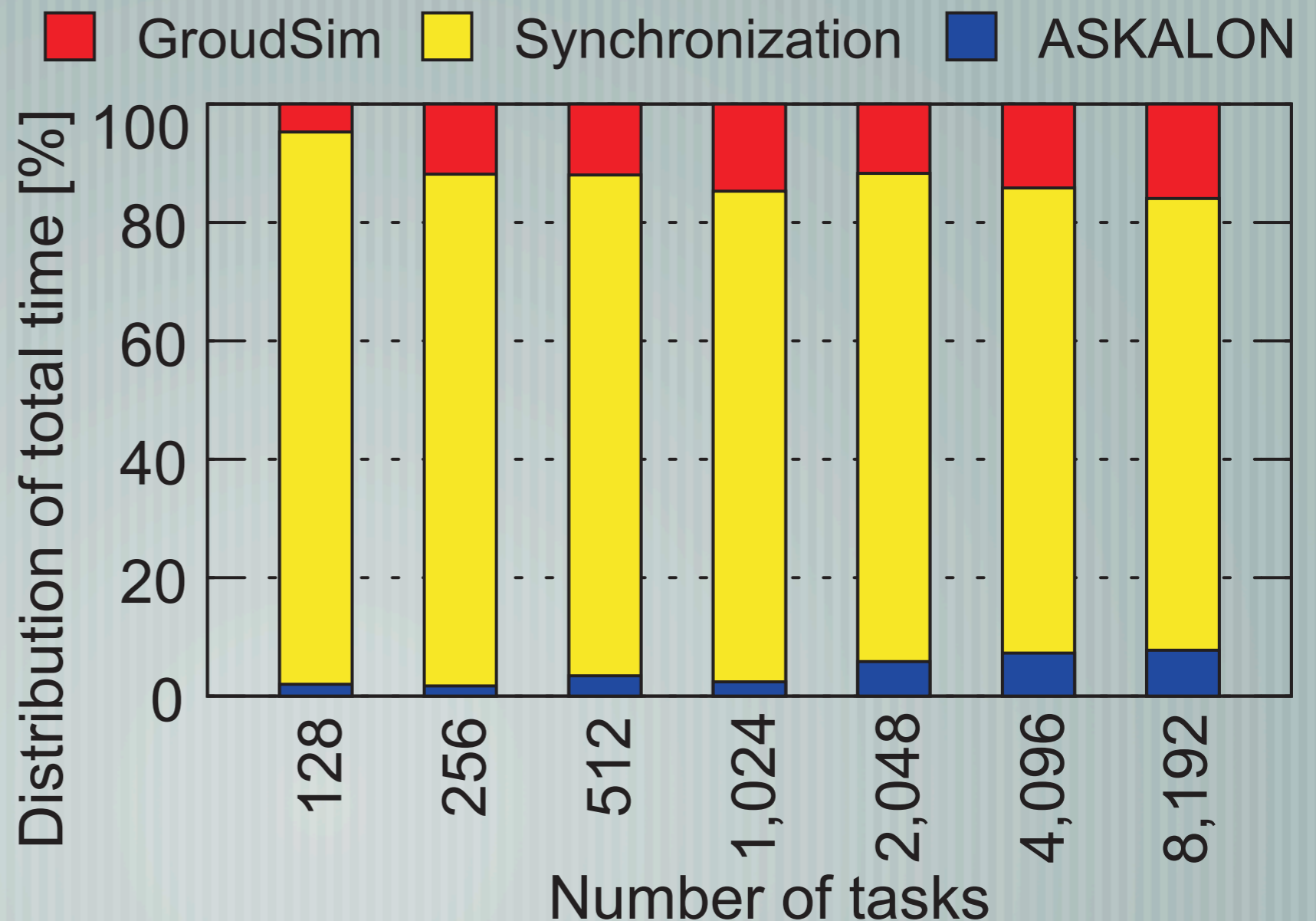
ASKALON does not work very efficient in this case, as the amount of work it can process is limited to 1 task



Overhead analysis II

Scenario with more Grid CPUs than jobs to submit -> all jobs are submitted at once

Synchronization hides part of the work done in ASKALON, when jobs are still being submitted but the simulation waits



Conclusion

- [Integration of simulation environment into a scientific workflow engine
- [Allows easy testing and simulation of optimizations in scheduling, prediction, file transfer and general workflow optimization
- [Allows scientists to test drive their workflow with our system with minimum overhead

Conclusion II

— [Good to demonstrate the functionality of ASKALON even when no Grid reachable

— [Fast execution compared to real runtimes with all features of the real execution

— [Cheap way to evaluate cloud provisioning techniques before going to the real resource

Thank you!

Questions?

