Scalability analysis of cloud-based distributed simulations of IoT systems using HLA

DSC^{2}T Workshop, IEEE ICPADS 2018, December 12, 2018

Thomas Nägele
t.nagele@cs.ru.nl

NWO Applied and Engineering Sciences

iCIS | Software Science
Radboud University

Contents

iCIS | Software Science
Radboud University

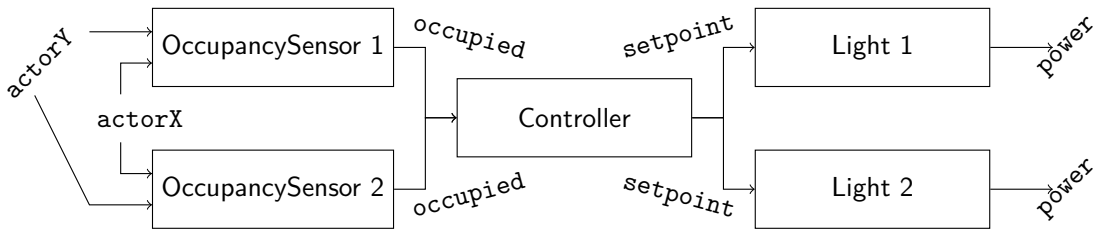
Introduction

- It is hard to verify correct behaviour of large IoT systems
- Errors are difficult to fix after installation
  - Time consuming, thus expensive
- Early simulation of the system allows early verification
- However, simulation is hard
  - Different disciplines
  - Scalability

Introduction

Case study: Smart lighting system

- Smart lighting system of an office building
  - Large IoT system consisting of hundreds of components
  - Based on an existing building
- Allows for different types of areas with different lighting behaviour
- Different models, different disciplines
  - Discrete-time versus continuous-time



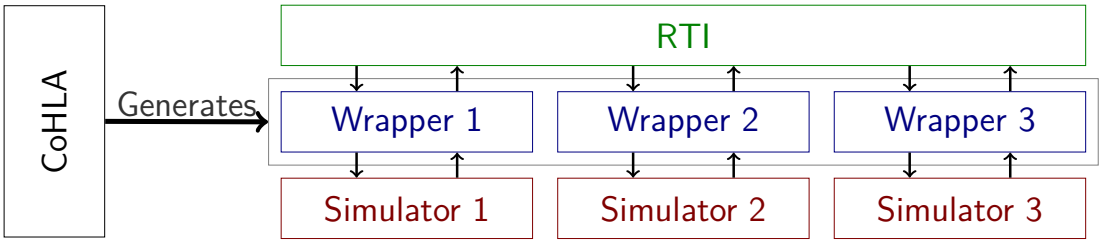
Challenges & Approach

- Rapid co-simulation construction using CoHLA
  - Domain specific language (DSL) to specify co-simulations
  - Using HLA and FMI standards

```
Confederation VerySmallBuilding {
  Instances {
    Instance r1Controller as BasicRoomController
    Instance r1Light1 as DimmableLight
    Instance r1Sensor1 as OccupancySensor
    Instance r2Controller as BasicRoomController
    Instance r2Light1 as DimmableLight
    Instance r2Sensor1 as OccupancySensor
    Instance corridorController as CorridorController
    Instance corridorLight1 as DimmableLight
    Instance corridorSensor1 as OccupancySensor
  }
  Connections {
    Connection { r1Light1.setpoint <- r1Controller.setpoint }
    Connection { r2Light1.setpoint <- r2Controller.setpoint }
    Connection { corridorLight1.setpoint <- corridorController.setpoint }
    Connection { r1Controller.occupied - r1Sensor1.occupied }
    Connection { r2Controller.occupied - r2Sensor1.occupied }
    Connection { corridorController.activity - corridorSensor1.occupied }
    Connection { corridorController.relatedActivity <- r1Sensor1.occupied }
    Connection { corridorController.relatedActivity <- r2Sensor1.occupied }
  }
}
```


Challenges & Approach

- Rapid co-simulation construction using CoHLA
  - Domain specific language (DSL) to specify co-simulations
  - Using HLA and FMI standards



iCIS | Software Science
Radboud University

Challenges & Approach

Scalability Challenges

- Building quickly becomes very large
  - CoHLA definition also becomes very large
  - Simulation time increases rapidly
- Manual definition of instances and connections is very error prone
- Size and location information is difficult to specify/modify

Approach

- Deal with scalability challenges
  - Use a DSL to specify lighting system
  - Distribute the co-simulation

Lighting DSL (LDSL)

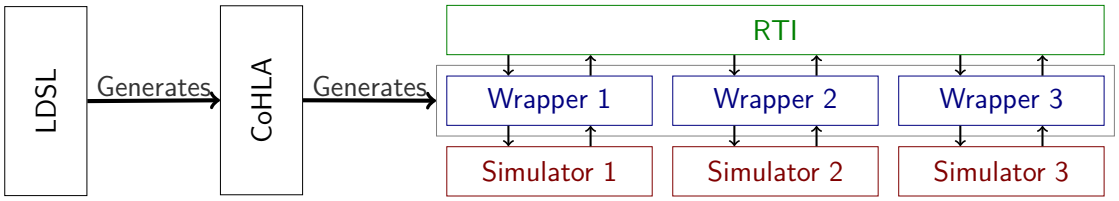
- A DSL to simplify the definition of a co-simulation for a building

```
1  | Building VerySmallBuilding {
2  |   Room room1 {
3  |     Draw: (0,0) (600,0) (600,300) (0,300)
4  |     Devices {
5  |       Light light1 on (300,150)
6  |       Sensor sensor1 on (300,150)
7  |     }
8  |   }
9  |   Room room2 {
10 |     Draw: (0,450) (600,450) (600,750) (0,750)
11 |     Devices {
12 |       Light light1 on (300,600)
13 |       Sensor sensor1 on (300,600)
14 |     }
15 |   }
16 |   Corridor corridor1 {
17 |     Draw: (0,300) (600,300) (600,450) (0,450)
18 |     Rooms: room1 room2
19 |     Devices {
20 |       Light light1 on (300,375)
21 |       Sensor sensor1 on (300,375)
22 |    }
23 |   }
24 | }
```


iCIS | Software Science
Radboud University

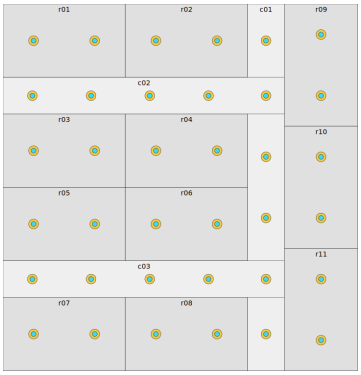
Lighting DSL (LDSL)

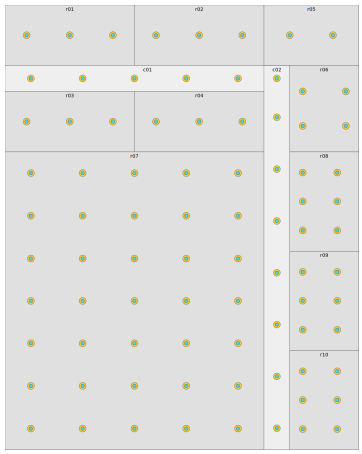
- A DSL to simplify the definition of a co-simulation for a building
- From this building specification, the following is generated
  - A CoHLA definition for the building
  - An image of the building
  - A web page to replay/show resulting log files

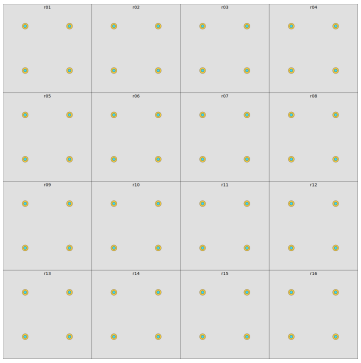


Distributed simulation

- Experiments conducted in commercial cloud
- Up to 16 concurrent computation nodes
- One actor and logger added to each co-simulation
- Multiple sample buildings to experiment with



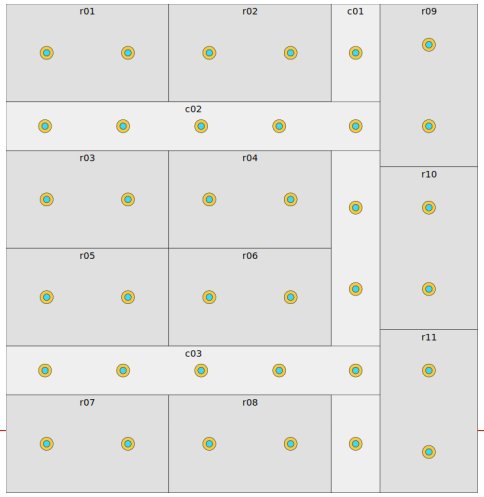


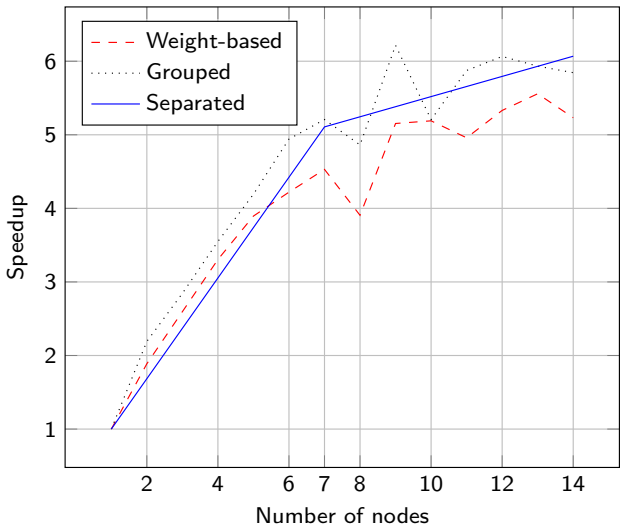


Distributed simulation

Distribution method

- Compare distribution methods with each other
  - Weight-based (automatic) distribution
  - Grouped distribution
  - Separated distribution
- Size: 90 federates

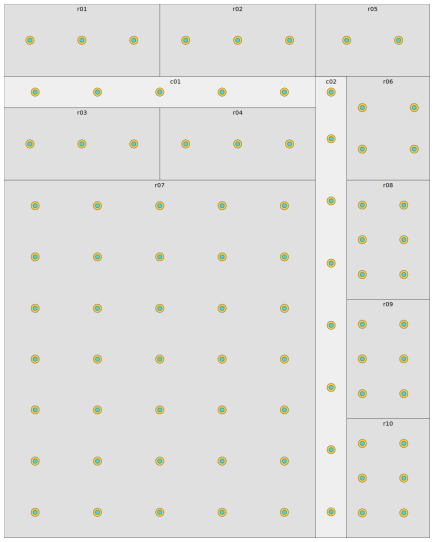


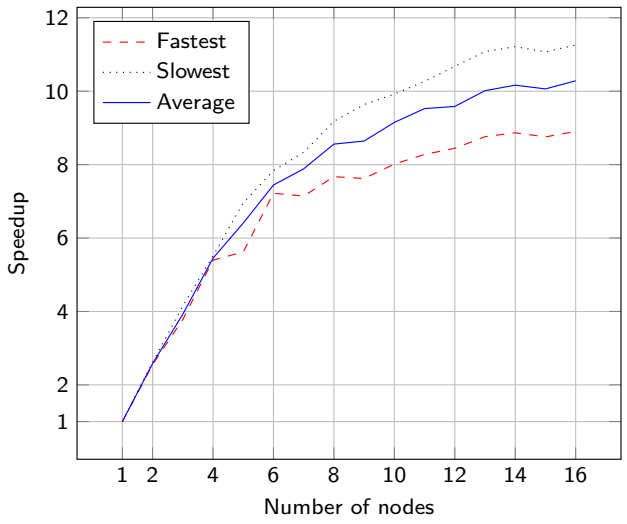


Distributed simulation

Scalability

- Performance scalability of the distribution
- Size: 184 federates



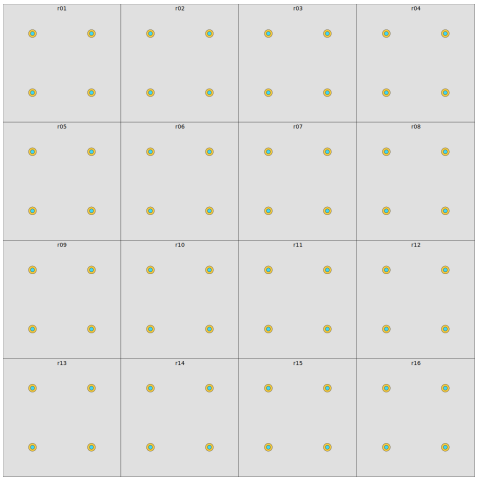


Distributed simulation

Optimising HLA distribution

- Publish-subscribe mechanism in HLA is class-based
  - Every controller receives **all** sensor updates!
  - Filtering is done in federate implementation

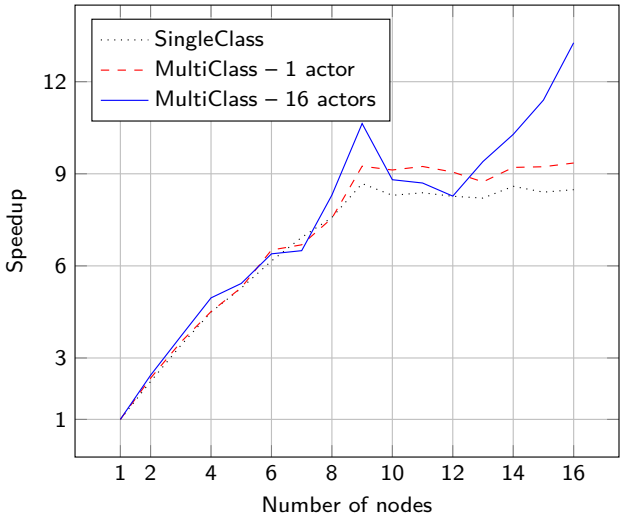
Area	Instance	Class	
		SingleClass	**MultiClass**
Room 1	Controller	LightController	LightControllerRoom1
	Light 1	Light	LightRoom1
	Light 2		
	Sensor 1	Sensor	SensorRoom1
	Sensor 2		
Room 2	Controller	LightController	LightControllerRoom2
	Light 1	Light	LightRoom2
	Light 2		
	Sensor 1	Sensor	SensorRoom2
	Sensor 2		
Corridor	Controller	CorridorController	CorridorControllerCorridor
	Light 1	Light	LightCorridor
	Sensor 1	Sensor	SensorCorridor



Distributed simulation

Optimising HLA distribution

- Scalability when using the MultiClass approach
- Size: 145 federates



Conclusion

- Distributed co-simulation can significantly decrease execution time
- The distribution method often does not impact the performance
- Co-simulation allows analysis of lighting/IoT systems during design
  - Design space exploration
    - System-level lighting behaviour, such as time-outs
    - Consequences for energy consumption
  - Early issue detection
- CoHLA simplifies the steps to distribute a co-simulation

Questions

Thanks for your attention. Are there any questions?

iCIS | Software Science
Radboud University