

# Performance analysis of a parallel PDEVS simulator handling both conservative and optimistic protocols

Ben Cardoen<sup>†</sup> Stijn Manhaeve<sup>†</sup> Tim Tuijn<sup>†</sup>  
{firstname.lastname}@student.uantwerpen.be

Yentl Van Tendeloo<sup>†</sup> Kurt Vanmechelen<sup>†</sup>  
Hans Vangheluwe<sup>†‡</sup> Jan Broeckhove<sup>†</sup>  
{firstname.lastname}@uantwerpen.be

<sup>†</sup> University of Antwerp, Belgium  
<sup>‡</sup> McGill University, Canada

## ABSTRACT

With the ever increasing complexity of simulation models, parallel simulation becomes necessary to perform the simulation within reasonable time bounds. The built-in parallelism of Parallel DEVS is often insufficient to tackle this problem on its own. Several synchronization algorithms have been proposed, each with a specific kind of simulation model in mind. Due to the significant differences between these algorithms, current Parallel DEVS simulation tools restrict themselves to only one such algorithm. In this paper, we present a Parallel DEVS simulator, grafted on C++11, which offers both conservative and optimistic simulation. We evaluate the performance gain that can be obtained by choosing the most appropriate synchronization protocol. Our implementation is compared to ADEVS using hardware-level profiling on a spectrum of benchmarks.

## 1. INTRODUCTION

## 2. BACKGROUND

### 2.1 Parallel DEVS

### 2.2 Conservative Synchronization

### 2.3 Optimistic Synchronization

### 2.4 C++11 Parallelism Features

## 3. FEATURES

### 3.1 Based on PythonPDEVS

### 3.2 Different Synchronization protocols

### 3.3 Performance Improvements

## 4. PERFORMANCE

### 4.1 Sequential Simulation

*CPU Usage*  
*Memory Usage*

### 4.2 Parallel Simulation

## 5. RELATED WORK

## 6. CONCLUSIONS

## ACKNOWLEDGMENTS

This work was partly funded with a PhD fellowship grant from the Research Foundation - Flanders (FWO).