TECHNISCHE UNIVERSITÄT DRESDEN

Chair of Construction Machines and Conveying Technology

OpenModelica Workshop 2011

Performance measurements and Benchmarking of the OpenModelica compiler

A. Pop • J. Frenkel

Outline

Part 1:

1. Introduction
2. Benchmark Models
3. Results
4. Conclusion

Part 2:

Adrian

1. Introduction

Necessity of Performance Measurements and Benchmarks

- Enhance implementation
- Compare different implementation
- Point out inefficient parts of the compiler
- Comparison with other compilers
- Objective measurement for evolution of compiler

2. Benchmarking

The Benchmark

- Synthetic models testing specific aspects of the compiler
- Focus lies on large models
- Benchmarks automated using Python
- will be presented at the Modelica Conference 2011

All benchmarks performed on Windows 7, 64 bit system with Intel Core i7 860, 2.80 GHz and 4.0 GB RAM

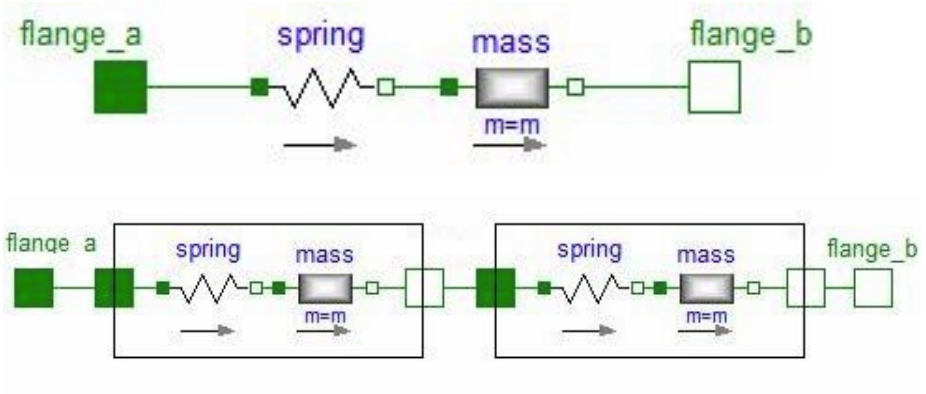
2. Models

Benchmark: evaluate performance regarding large models

Flat Model:

```
model flatclass_n
  input Real inp;
  Real v_1;
  Real v_2;
  Real v_3;
  ...
  Real v_n;
equation
  v_1 = 1 + v_2;
  v_2 = 2 + v_3;
  ..…
  v_(n-1) = (n-1) + v_n;
  der(v_n) = v_1 + inp;
end flatclass_n;
```

Hierarchical Model:

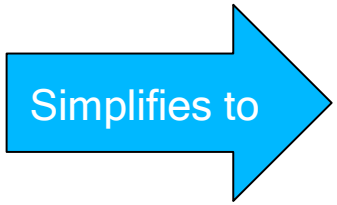


Level	Equations
1	**21**
2	**42**
3	**84**
4	**168**

2. Benchmarking

Benchmark: Remove Alias Equations

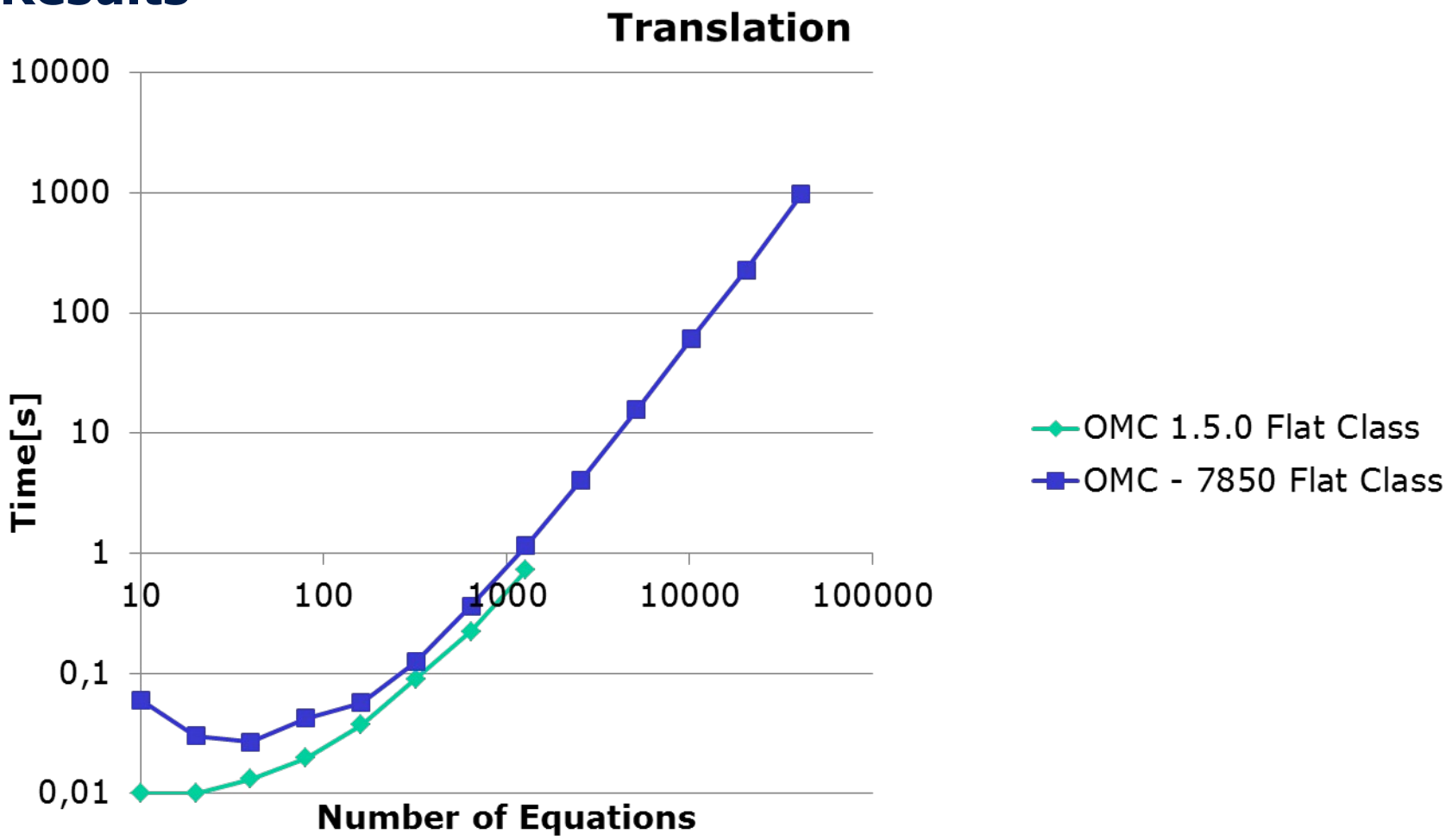
```
model AliasClass_N
  input Real inp;
  constant Integer N=4;
  Real a[2*N+1];
equation
  der(a[1]) = inp;
  a[2] = -a[1];
  a[3] = 2*a[2]+a[1];
  for i in 4:2:2*N loop
    a[i] = a[i-3] + a[i-2] - a[i-1];
    a[i+1] = i*a[i]+(i-1)*a[i-1];
  end for;
end AliasClass_N;
```




```
model AliasClass_N
  input Real inp;
  constant Integer N=4;
  Real a[2*N+1];
equation
  der(a[1]) = inp;
  a[2] = -a[1];
  a[3] = -a[1];
  a[4] = a[1];
  a[5] = a[1];
  a[6] = -a[1];
  a[7] = -a[1];
  …
end AliasClass_N;
```

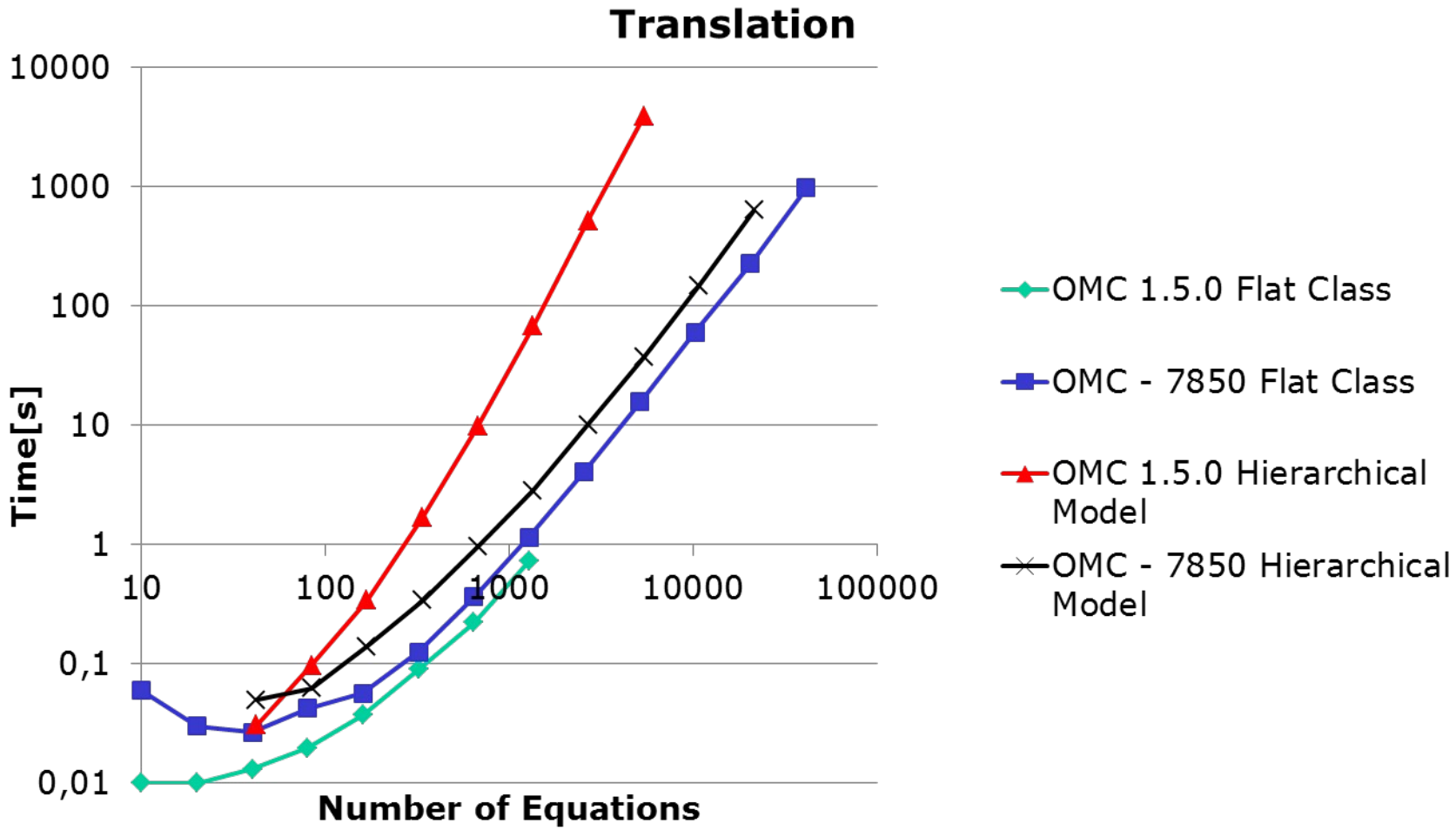
3. Results

Results



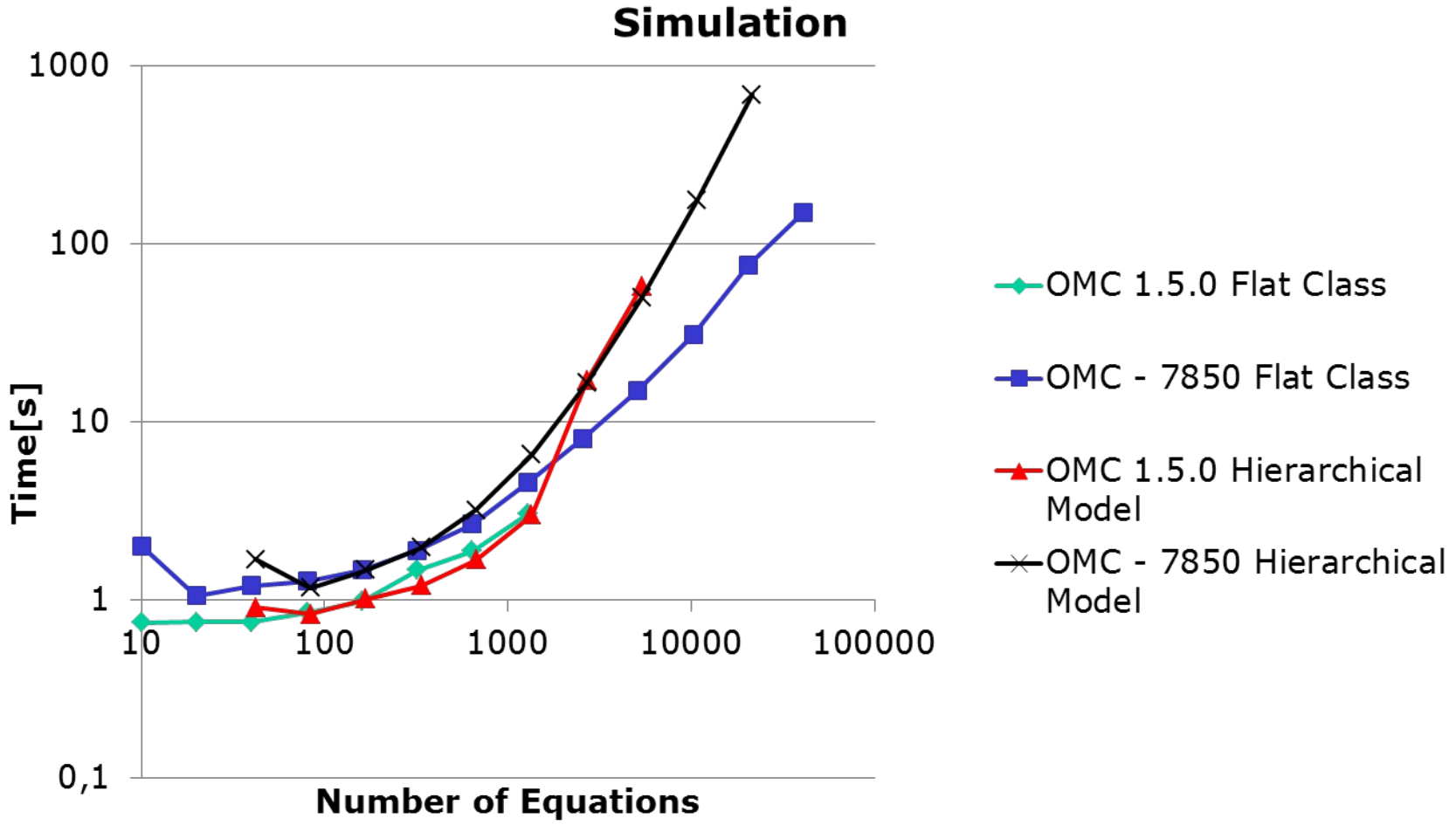
3. Results

Results



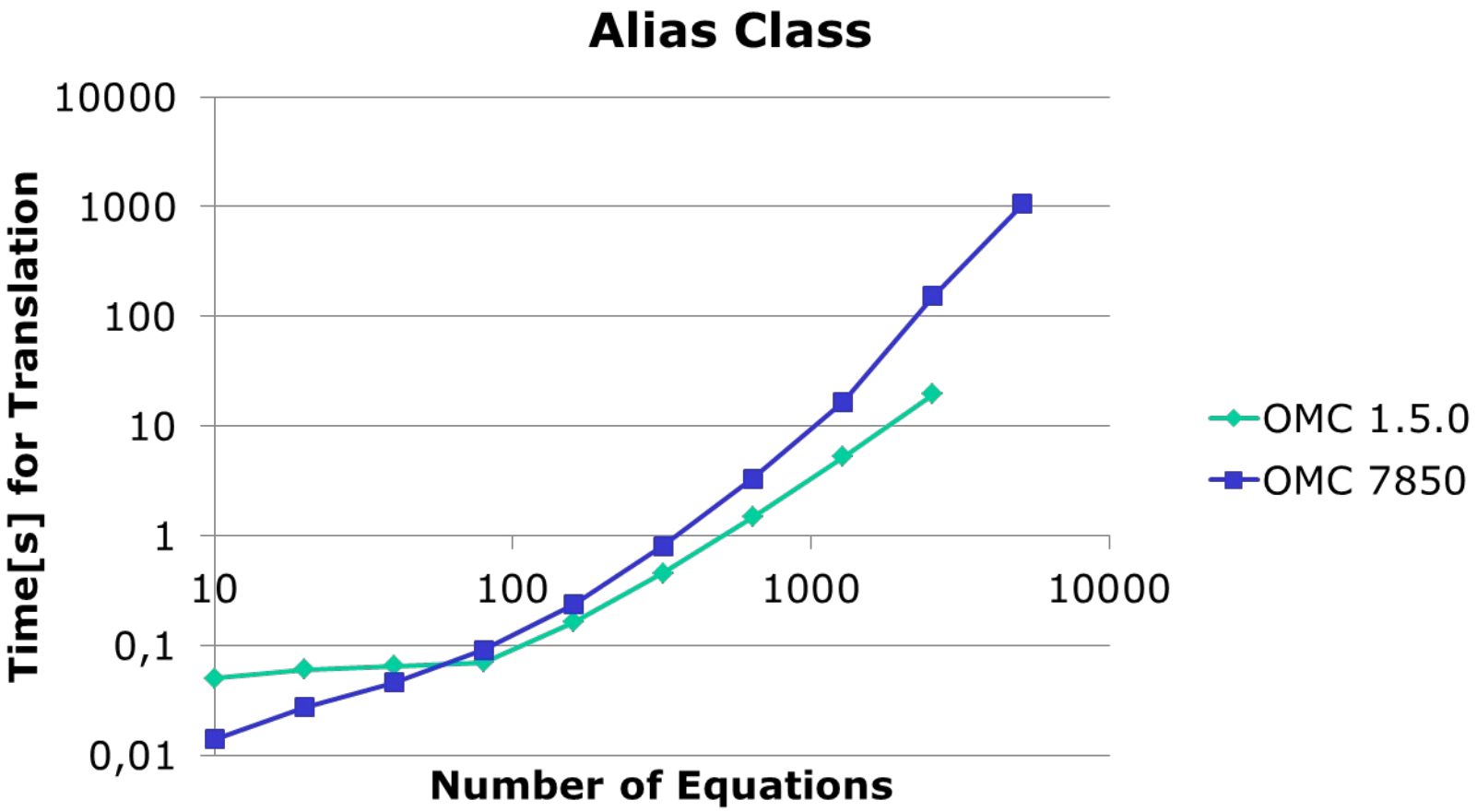
3. Results

Results



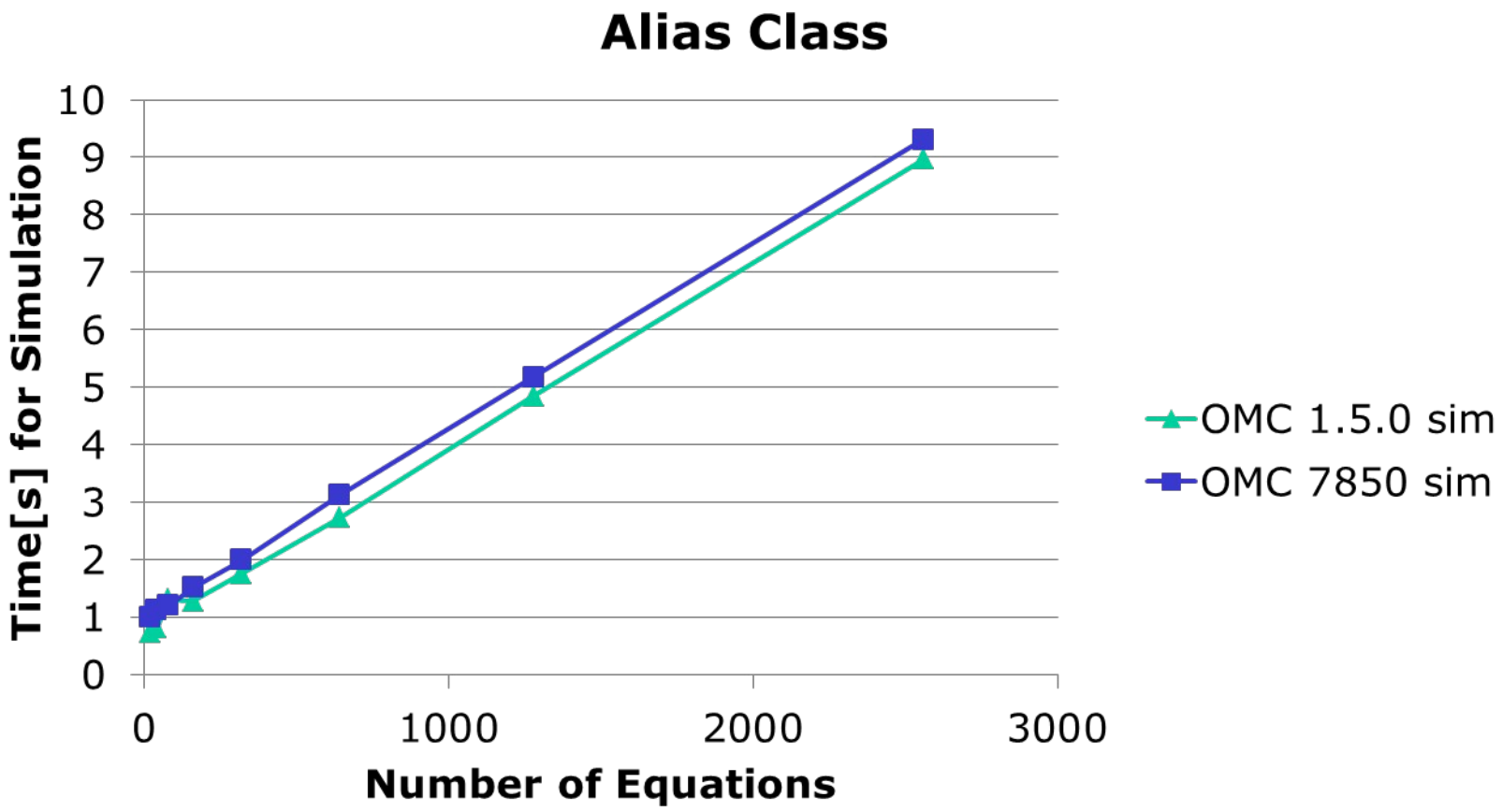
3. Results

Results



3. Results

Results



4. Conclusion

- Maximum treatable model size increased from around 10 000 to 40 000 equations
- Faster compilation process for models with fine grained class structure
- more work during translation done, time for translation remained equal
- Method available to test the influence of new features/implementations
- Method to check specific parts of the compiler

The End Part 1

Jens Frenkel
Dresden University of Technology
jens.frenkel@tu-dresden.de
http://tu-dresden.de/bft