Towards Model-based Performance Predictions of SAP Enterprise Applications

Adrian Streitz, Maximilian Barnert, Johannes Rank, Harald Kienegger, Helmut Krcmar
Agenda

1. Motivation
2. Context: SAP Ecosystem
3. Conceptual Approach
4. Experiment
5. Conclusion
Motivation

Long running software applications affect a **loss of productivity** for each user and leads eventually to **financial damage** for the company (unused working hours, delays in production lines, etc.)

(Compuware, 2010)
## Motivation

<table>
<thead>
<tr>
<th>Situation</th>
<th>Complication</th>
<th>Resolution</th>
</tr>
</thead>
</table>
| - Performance has become an important key factor for today's application systems (Willnecker, 2018) | - traditional software engineering methods  
  - late performance evaluations after development  
  - test scenarios are complex & time-consuming  
  - Previous work (Streitz et al., 2018) reveals  
  - lack of dedicated performance skills  
  - strong occupation by providing new functionality  
  - no extra time for complex performance evaluations | - Performance Awareness (Tuma, 2014)  
  - observation of software performance  
  - react upon problems during development  
  - model-based performance predictions |
| - efficiency of software systems correlates with the company's revenue (Heger et al., 2017) | | |
| - ERP software is highly used in industry (Statistisches Bundesamt, 2017) | | |
| - most common representative: SAP ERP | | |

Symposium on Software Performance 2018, Hildesheim, Germany
Current Situation: ERP Systems

- success of companies in a competitive environment associates that all *business processes run effectively*
- most business processes in industry are supported by Enterprise Resource Planning (ERP) systems
- SAP SE (market leader for ERP systems)
  - 23% market share (worldwide)
  - 46% market share (Germany)

companies rely on **stable and efficient** SAP ERP applications (*time-sensitive* transactional operations)
SAP Ecosystem
Conceptual Approach

- **model-based performance evaluations** help
  a) late performance evaluations
  b) lack of performance skills
  c) time pressure of software developers
- automatic model transformation necessary

---

**Procedure**

1. Abstract Syntax Tree Generation from ABAP source code & cleanup post-processing
2. Transformation into PCM instances
3. Model parameterization by considering performance records from SAP ERP system
4. Conduction of model simulations → receive performance predictions
Abstract Syntax Trees

```plaintext
x := a + b;
y := a * b;
while (y > a) {
    a := a + 1;
    x := a + b;
}
```
Prototype

SAP Test System

SAP Development System

- SAP Metrics
- SAP Metrics Provider
- SAP DB
- ABAP Source Code Package
- AST Generation
- AST XML files

Java Environment

- Performance Records Provider
- Performance Knowledge DB
- Performance Records Adapter
- SAP Performance Adapter

- AST Preprocessing
- Clean AST
- PCM Transformation
- Simulation Engine
- Performance Results
- PCM Instance
Example: Iterative Signature Tracing

1. Run through all ASTs
2. Bottom up search for "var1"
3. Found type for "param1"
4. Collect found parameters
5. Generate Signature after (1) is completed
Experiment

- ABAP program for determining code clones
- covers procedural and object-oriented ABAP instructions
- two execution variants with different response time behavior
Experimental Results

### Performance Prediction

<table>
<thead>
<tr>
<th>Variant</th>
<th>Response time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AST-based</td>
<td>5.10</td>
</tr>
<tr>
<td>Token-based</td>
<td>2.10</td>
</tr>
</tbody>
</table>

### Performance Measurement

<table>
<thead>
<tr>
<th>Variant</th>
<th>Response time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AST-based</td>
<td>5.21</td>
</tr>
<tr>
<td>Token-based</td>
<td>2.96</td>
</tr>
</tbody>
</table>

![Chart showing response times for Variant A and B]
Future Work

+ bigger evaluation
+ IDE integration
Conclusion

- consideration of model-based performance predictions for ERP software
  - approach to transform ABAP into PCM
- easy to use – fully automated & fast
  - future work: automatic parameterization + IDE integration
- goal: no knowledge in the performance modelling domain required

Enable **Performance Awareness** of ABAP developers
Thank you for your attention!

Adrian Streitz, M.Sc.
adrian.streitz@in.tum.de

Technical University of Munich
Department of Computer Science
Boltzmannstr. 3
85748 Garching