



### Interactive Production Performance Feedback in the IDE

### Jürgen Cito

2nd Vienna Software Seminar, August'19



### No continuous feedback loop from production to development



### PerformanceHat

Instant performance feedback prediction on code changes implemented as Eclipse plugin with Java



### Continuous and Incremental Feedback Algorithm

```
Algorithm 1: Matching, Inferring, and Propagating Run-
time Information to AST Nodes in Method m
 Data: A method m \in M(p), a dataset \mathcal{D}, a specification function \mathcal{S},
       an inference function \Gamma
 Result: All relevant AST nodes in m annotated with data in \mathcal{D} or with
        a prediction inferred through \Gamma
 toInferNodes \leftarrow \emptyset:
 // Iterator goes through method AST through BFS
     (i.e. outside-in)
                                                                       [Pass 1: Specification Mapping]
                                                                     [Pass 2: Inference + Propagation]
```

# Inference on code changes





#### [Pass 1: Specification Mapping]

## Inference on code changes

![](_page_5_Figure_1.jpeg)

![](_page_5_Figure_2.jpeg)

[Pass 2: Inference + Propagation]

![](_page_6_Figure_0.jpeg)

"Enhancing Performance Prediction Robustness by Combining Analytical Modeling and Machine Learning", D. Didona et al., ICPE'17

Transfer learning uses knowledge on the source to learn a cheaper target model

![](_page_6_Figure_3.jpeg)

"Transfer Learning for Software Performance Analysis: An Exploratory Analysis", P. Jamshidi et al., ASE'17

### PerformanceHat (Open Source Case Study: Agilefant)

```
@Transactional(readOnly = true)
public List<StoryT0> retrieveLeafStories(int projectId, StoryFilters filters) {
    Project original = this.retrieve(projectId);
```

```
List<Story> leafStories = this.storyRankBusiness.
    retrieveByRankingContext(original);
```

```
leafStories = storyFilterBusiness.
    filterStoryList(leafStories, filters);
List<StoryT0> leafStoriesWithRank = new ArrayList<StoryT0>();
int rank = 0;
for(Story leafStory : leafStories) {
    StoryT0 tmp = new StoryT0(leafStory);
    tmp.setRank(rank++);
    Set<Task> tasks = new HashSet<Task>();
    for (Task task : tmp.getTasks()) {
        TaskT0 taskT0 = new TaskT0(task);
        long sum = 0;
        (/TDDD: sum up minutes from taskT0 aptHounEntries and
```

### Interactive PerformanceHat

Enable interactive model and uncertainty reasoning

![](_page_8_Picture_3.jpeg)

### Challenges

Method-level sampling not available

Inferring Latent Variables for Performance Models

**Dealing with Uncertainty** 

**Premature Optimization** 

![](_page_10_Picture_0.jpeg)

#### Integration in HANA Web IDE

![](_page_10_Figure_2.jpeg)

![](_page_10_Figure_3.jpeg)

Number of Calls

297

108

 $2^{-6}$ 

"Monitoring-aware IDEs" @ FSE'19

![](_page_10_Figure_5.jpeg)

![](_page_10_Picture_6.jpeg)

Log awareness - field study w/ Jos Winter, Mauricio Aniche, Arie van Deursen @ TU Delft

Integration with Infer for Code Review

## Interactive Production Performance Feedback in the IDE enables developers to *contextualize* production perspectives and **prevent** performance regressions

![](_page_11_Figure_1.jpeg)

![](_page_11_Picture_2.jpeg)

http://sealuzh.github.io/PerformanceHat/

### Jürgen Cito <jcito@mit.edu>

![](_page_11_Picture_5.jpeg)

![](_page_11_Picture_6.jpeg)