

# Master's Thesis

## Hardware-assisted Program Tracing for Error Propagation Analysis

### Background

Whenever a system fails, it is important to understand the root cause behind the failure. If it is a software bug, then this bug needs to be identified and removed to prevent the observed failure from reoccurring. If it is a hardware problem or unexpected environmental condition, the probability of its reoccurrence needs to be assessed and, depending on whether this probability is considered acceptably low or not, counter-measures for handling the exceptional case may be added. The process of understanding the cause-effect chain in a system leading to a system failure is called *error propagation analysis* (EPA).

EPA is commonly based on analyses of the execution trace of a failed execution, i.e. a record of the instructions that have been executed and on which data prior to the failure. To spot the point when an execution trace is deviating from a correct execution of the system, the failed execution trace is compared against a previously recorded execution that did not fail. This correct execution is called a *golden run*.

### Objectives

While such trace based error propagation analysis is effective, it has two important limitations. First, execution tracing incurs massive runtime overheads. In addition to their execution, each machine instruction needs to be logged to persistent storage. Second, execution traces are of limited utility for EPA of multithreaded programs, as scheduling non-determinism may cause trace deviations that are unrelated to the system failure. This thesis addresses the two limitations by

1. applying Intel's PT ISA extensions for execution tracing and
2. utilizing static program analyses to differentiate valid from invalid permutations of trace instructions.

### Prerequisites

Candidates should have experience with C/C++ programming. As the usage of PT will likely involve the implementation of a Linux kernel module or the modification of Intel's existing kernel module, any experience with Linux kernel programming is beneficial. The thesis will be written in English.

### Duration/Start

Immediate

### Contact

**Dr. Stefan Winter**  
sw@cs.tu-darmstadt.de  
+49 6151 16 25226  
S2|02 E221

#### Literature



#### Analysis



#### Implementation



#### Awesomeness

