Anomaly Detection Approach

Matthias Rohr

BTC Business Technology Consulting AG, Germany,
Graduate School TrustSoft*, University of Oldenburg, Germany

March 21, 2011
Software Engineering Group, University of Kiel,
Kiel, Germany

* This work is supported by the German Research Foundation (DFG), grant GRK 1076/1
Context

Primary thesis contribution
- Trace context sensitiv timing behavior analysis (TracSTA)
- Workload intensity sensitive timing behavior analysis (WiSTA)

Secondary thesis contribution
- Application of TracSTA and WiSTA in anomaly-detection-based fault localization scenario ([Rohr, 2006; Rohr et al., 2007])

Primary contribution:
Timing Behavior Analysis Methods
Evaluation:
Quantitative empirical evaluation in industry studies

Secondary contribution: Fault localization approach
Evaluation: Proof-of-concept demonstration in lab-studies and application of the monitoring infrastructure in industry systems
Workload-intensity-sensitive Timing Behavior Analysis

1. Instrumentation and Monitoring
   - Recording of:
     - **Response times** of software operation executions
     - **Execution sequences** corresponding to user requests
     - Host identifier
   - Reconstruction of Traces and Dependency Graphs

2. Trace-Context-Sensitive Timing Behavior Analysis

3. Workload-Intensity-Sensitive Timing Behavior Analysis

4. Anomaly Detection

5. Anomaly Correlation and Fault Localization
1. Instrumentation and Monitoring

2. Trace-Context-Sensitive Timing Behavior Analysis
   1. Identification of trace contexts
   2. Defining classes of observations based on the trace context
   3. (Re-)Merging classes

3. Workload-Intensity-Sensitive Timing Behavior Analysis

4. Anomaly Detection

5. Anomaly Correlation and Fault Localization
Workload-intensity-sensitive Timing Behavior Analysis

1. Instrumentation and Monitoring

2. Trace-Context-Sensitive Timing Behavior Analysis

3. Workload-Intensity-Sensitive Timing Behavior Analysis
   1. Definition of a workload metric by machine learning
   2. Splitting observations according to workload-intensity

4. Anomaly Detection

5. Anomaly Correlation and Fault Localization
Workload-intensity-sensitive Timing Behavior Analysis

1. Instrumentation and Monitoring

2. Trace-Context-Sensitive Timing Behavior Analysis

3. Workload-Intensity-Sensitive Timing Behavior Analysis

4. Anomaly Detection
   - Evaluation of new observations in context of a profile.
   - How normal is a new observation?

5. Anomaly Correlation and Fault Localization
1. Instrumentation and Monitoring

2. Trace-Context-Sensitive Timing Behavior Analysis

3. Workload-Intensity-Sensitive Timing Behavior Analysis

4. Anomaly Detection

5. Anomaly Correlation and Fault Localization
   - Derivation of component ratings from execution ratings
   - Derivation of causes from symptoms
Assumptions: Anomaly propagation model

**Structural architecture**

- Comp1
- Comp2
- Comp3

**Request**

**Response**

**Functional behavior with error propagation**

- An error can create errors along calling dependencies
- Faults and errors *can* cause timing behavior anomalies

**Timing behavior with anomalies**

- Timing behavior anomalies propagate along calling dep.
Anomaly detection

Software timing behavior anomaly detection requirements

- Heavy tails, right skewed (i.e. mode < median < mean), multimodality
- Context information (parameters, workload, state ...)

(a) Mean 34.26

(b) Mean 34.26
Anomaly detection (2/3)

(c) Scatter plot of response times.  
(d) Normalized probability density of the response times.
LOF [Breunig et al., 2000] and LoOP [Kriegel et al., 2009] algorithms provided by ELKI Framework [Achtert et al., 2010].
(g) Anomaly scores for response times using different anomaly detection functions.

LOF [Breunig et al., 2000] and LoOP [Kriegel et al., 2009] algorithms provided by ELKI Framework [Achtert et al., 2010].
Future work

Integration into Kieker or our other ongoing work

- Sharing of anomaly detection algorithms
- Stream-oriented wrappers for non-stream-processing algorithms
- Standardized interfaces and patterns (e.g. annotation?) for Kieker integration


