ΘPAD:
Online Performance Anomaly Detection with Kieker

Tillmann Bielefeld\(^1\) and André van Hoorn\(^2\)

\(^1\) empuxa GmbH, Kiel
\(^2\) Software Engineering Group, Kiel University

7th Hamburg Web Performance Meetup
October 24, 2012 @ Microsoft, Hamburg
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

Oct. 24, 2012 @ Hamburg
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

Does the searchBook service respond in <= 0.5 seconds in 95% of all cases?
**Motivation: Monitoring/Dynamic Analysis**

Kieker: Framework Overview

Does the *searchBook* service respond in $\leq 0.5$ seconds in $95\%$ of all cases?

What is the system's availability?

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

<<SystemAssembly>>
BookstoreSystem

<<Component>>
Bookstore

<<Component>>
Catalog

<<Component>>
CRM
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

BookstoreSystem
<<SystemAssembly>>

Bookstore
<<Component>>

CRM
<<Component>>

Catalog
<<Component>>

How do the components interact?
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

BookstoreSystem
<<SystemAssembly>>

Bookstore
<<Component>>

CRM
<<Component>>

Catalog
<<Component>>

How do the components interact?

Which component causes the problem?
Motivation: Monitoring/Dynamic Analysis

Kieker: Framework Overview

How are my resources utilized?

Bookstore System

Bookstore

Catalog

CRM
Continuous Monitoring of Software Systems

Kieker: Framework Overview

Self-adaptive Software Performance Monitoring for Anomaly Localization

- Application-level Monitoring
  - Observations in field
    - Extensive infrastructure monitoring, application monitoring not widespread
    - Reactive monitoring probe injection only (after a critical performance drop has occurred)

- Business Processes
  - Key performance indicators, e.g. process throughput, ...
- Services
  - SLO appliance, workload, ...
- Application
  - Response times, operational profile, ...
- Middleware Container
  - Thread/connection pool sizes, ...
- Virtual Machine
  - Heap size, ...
- Operating System
  - CPU/memory utilization, ...
- Hardware
  - Availability, reliability, ...

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

Software system with monitoring instrumentation

Monitoring probe

PAD w/ Kieker
Oct. 24, 2012 @ Hamburg
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

Software system with monitoring instrumentation

Java probes/samplers:

- Manual instrumentation
- AspectJ
- Spring
- Servlet
- CXF/SOAP
- SOAP
- Sigar
- CPU utilization
- Memory usage
- <your interception technology>
- <your technology>
- <your monitoring probe>

+ basic adapters for
  - C#/.NET
  - Visual Basic 6/COM
  - COBOL
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

Measurement

Monitoring records

Monitoring log/stream

Monitoring probe

Software system with monitoring instrumentation
Kieker: Example Workflow and Use Cases

Kieker: Framework Overview

Monitoring records
Measurement
Monitoring log/stream
Monitoring Records
Current time
Resource utilization
CPU utilization
Operation execution
Control-flow events
Memory/swap usage
<your monitoring record type>

Software system with monitoring instrumentation
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

Measurement

Monitoring records

Monitoring log/stream

Monitoring Records

Current time

Resource utilization

CPU utilization

Operation execution

Control-flow events

Memory/swap usage

File system

Java Messaging Service (JMS)

Java Management Ext. (JMX)

Database (SQL)

Named pipe

<your monitoring reader/writer>

<T. Bielefeld (empuxa) and A. van Hoorn (CAU)>

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

- Monitoring records
- Analysis configuration (Web GUI)
- Analysis
- Plugins
- Monitoring log/stream
- Measurement
- Monitoring probe
- Software system with monitoring instrumentation
Kieker: Example Workflow and Use Cases
[van Hoorn et al. 2012]

Kieker: Framework Overview

Analysis configuration (Web GUI)

Analysis

Plugins

Monitoring records

Measurement

Monitoring log/stream

Analysis/Visualization Plugins

Sequence diagrams

Dependency graphs

Call graphs

Architecture reconstruction

Trace analysis

<your visualization>

<your trace analysis>

<your reconstruction plugin>

<your analysis plugin/tool>

Software system with monitoring instrumentation
Core Kieker Framework Components

Kieker: Framework Overview

Instrumented Software System

Kieker.Monitoring

Monitoring
Probe

JMX
Interface

Periodic
Sampling

Logging

Time
Source

Monitoring Controller

Monitoring
Writer

Monitoring Log/Stream

Monitoring Record

Pipe & Filter Configuration

Analysis / Visualization Plugin

Analysis Controller

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Core Kieker Framework Components

Kieker: Framework Overview

Instrumented Software System

Kieker.Monitoring

- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- JMX Interface
- Periodic Sampling
- Logging

Monitoring Log/Stream

Monitoring Record

Kieker.Analysis

- Pipe & Filter Configuration
- Analysis / Visualization Plugin
- Analysis Controller

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Core Kieker Framework Components

Kieker: Framework Overview

Instrumented Software System

Kieker.Monitoring
- Monitoring Probe
- Monitoring Controller
- Monitoring Writer

Monitoring Log/Stream
- Monitoring Record

Kieker.Analysis
- Pipe & Filter Configuration
- Monitoring Reader
- Analysis / Visualization Plugin
- Analysis Controller
Core Kieker Framework Components

Kieker: Framework Overview

Instrumented Software System

Kieker.Monitoring
- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- JMX Interface
- Time Source
- Logging
- Periodic Sampling

Monitoring Log/Stream
- Monitoring Record

Kieker.Analysis
- Monitoring Reader
- Analysis / Visualization Plugin
- Pipe & Filter Configuration
- Analysis Controller

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

Kieker.Analysis example pipes-and-filters configuration
Performance anomaly detection and visualization
Architecture and trace reconstruction/visualization

outputPort
<<Reader>>
: FS reader

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
ΘPAD w/ Kieker
Oct. 24, 2012 @ Hamburg
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

Kieker.Analysis example pipes-and-filters configuration

Performance anomaly detection and visualization

Architecture and trace reconstruction/visualization

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

1. Kieker.Analysis example pipes-and-filters configuration
2. Performance anomaly detection and visualization

- <<Reader>>: FS reader
- operationExecutions
- <<Filter>>: Performance anomaly filter
- systemModel
- anomalyRatings

- <<Repository>>: System model repository
- outputPort

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ThetaPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

- Performance anomaly detection and visualization
- System model repository

<<Filter>>
: Performance anomaly filter

<<Filter>>
: Anomaly graph plotter

<<Repository>>
: System model repository

<<Reader>>
: FS reader

operationExecutions

systemModel

anomalyRatings

outputPort

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Kieker: Framework Overview

Example Pipe-and-Filter Configuration

Kieker.Analysis example pipes-and-filters configuration

- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

- Kieker.Analysis example pipes-and-filters configuration
- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization

<<Reader>>
: FS reader

<<Repository>>
: System model repository

<<Filter>>
: Trace reconstruction filter
- traceEvents
- systemModel
- messageTraces
- executionTraces

<<Filter>>
: Performance anomaly filter
- operationExecutions
- systemModel
- anomalyRatings

<<Filter>>
: Anomaly graph plotter
- anomalyRatings

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Example Pipe-and-Filter Configuration

Kieker: Framework Overview

Performance anomaly detection and visualization

Architecture and trace reconstruction/visualization

<<Reader>>
: FS reader

<<Repository>>
: System model repository

<<Filter>>
: Trace reconstruction filter

<<Filter>>
: Sequence diagram visualization

<<Filter>>
: Dependency graph visualization

<<Filter>>
: Anomaly graph plotter

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ThetaPAD w/ Kieker
Oct. 24, 2012 @ Hamburg
Kieker: Framework Overview

**Example Pipe-and-Filter Configuration**

- **Performance anomaly detection and visualization**
- **Architecture and trace reconstruction/visualization**

**Kieker.Analysis example pipes-and-filters configuration**
- **Performance anomaly detection and visualization**
- **Architecture and trace reconstruction/visualization**
Welcome to the Kieker.WebGUI
This is an early beta version of the Kieker WebGUI. Therefore it may contain bugs and some functionality may have not been implemented yet. Just click "Login" to continue.

Welcome to the Kieker.WebGUI
Username: Guest
Password: Guest
Login

Welcome to the Kieker.WebGUI
Project Name: Bookstore-Example
Status: TERMINATED
Last Modification: Wed Oct 17 00:00:50 CEST 2012

Kieker > Bookstore-Example
File > Help >

Analysis Plugins
• Filter
  MyResponseTime(filter)
  MyResponseTime(filter)
  MyResponseTime(filter)
  MyResponseTime(filter)

Properties

Casename: kiekerexamples.userpublic.3anin.bookstore.MyResponseTime(filter)
Name: MyResponseTime(filter)

Analysis

Controller Log
Mon Oct 22 11:12:30 CEST 2012: Starting Analysis for project 'Bookstore-Example'
Mon Oct 22 11:12:33 CEST 2012: Starting Analysis for project 'Bookstore-Example'

Control

Initialize Analysis
Clean Analysis
Start Analysis
Stop Analysis
WebGUI (Beta) Included in Kieker 1.6

Kieker: Framework Overview

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

PAD w/ Kieker
Oct. 24, 2012 @ Hamburg

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Kieker: Framework Overview

- Modular, flexible, and extensible architecture (Probes, records, readers, writers, filters etc.)
- Pipes-and-filters framework for analysis configuration
- Distributed tracing (logging, reconstruction, visualization)
- Low overhead (designed for continuous operation)
- Evaluated in lab and industrial case studies

Kieker is open-source software (Apache License, V. 2.0)

http://kieker-monitoring.net

Kieker is distributed as part of SPEC® RG's repository of peer-reviewed tools for quantitative system evaluation and analysis

http://research.spec.org/projects/tools.html
Agenda

1. Kieker: Framework Overview

2. OPAD: Online Performance Anomaly Detection

3. Conclusion
Thesis Goals

1. Design of online performance anomaly detection concept (ΘPAD)
2. ΘPAD implementation as Kieker plugin
3. ΘPAD integration with case study system
4. Evaluation @ XING

Tillmann C. Bielefeld:
“Online performance anomaly detection for large-scale software systems”
Existing Logjam-based Monitoring @ XING

Logjam -based monitoring already in place @

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

Logjam-based monitoring already in place @ XING
Integration of ΘPAD in XING’s Architecture

OPAD: Online Performance Anomaly Detection

XING’s logging/monitoring architecture

Servers
- App
- Support
- XWS (API)
- DB
- Background

Importer
Log Database
Logjam

mongoDB
(name: "mongodb", type: "db")

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Integration of ΘPAD in XING’s Architecture

TPAD: Online Performance Anomaly Detection

Servers

App

Support

XWS (API)

DB

Background

Importer

Log Database

Logjam

mongoDB

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Example JSON Logging Message

```
{
    "count": 5204.903527993169,
    "memcache_time": 6505.196318140181,
    "api_time": 2207.0271495891297,
    "db_time": 5004.8727338680155,
    ...
    "view_time": 3936.1623304929153,
    "total_time": 1586.8188192888886,
    "api_calls": 5546.250545491678
}
```

Input data received via AMQP and processed by ΘPAD
Chapter 4. Design and Implementation of the ΘPAD System

For this implementation, Kieker is used as a base only without additional analysis means. However, it is possible to deploy ΘPAD alongside other performance analysis plugins.

Following are the previously introduced libraries and technologies woven together in Kieker (Section 4.4.1). The plugin's internal structure is described in Section 4.4.2. Ultimately, Section 4.4.3 explains how good code quality was achieved in the development process.

4.4.1 Integration into Kieker

ΘPAD uses the Kieker monitoring framework written in Java as a base for its implementation. The connection to the anomaly detection code is made by a plugin container Kieker offers. The plugin pattern is used when certain behavior is based on different implementations that have to be configured in a central runtime environment [Fowler 2003, p. 500]. To start ΘPAD in a Kieker environment, all necessary depending libraries, the kieker.jar and the plugin code have to be accessible in the same class path.

ΘPAD gets instantiated at startup of the Kieker server. Both main architectural components of Kieker, Monitoring and Analysis are used to route the measurements to the plugin. The data flow from input to output is illustrated in Figure 4.4.

Figure 4.4: The coarse-grained architecture follows the linear data flow of the approach (see Chapter 3). The AMQPBridge adapter translates the monitored system’s measurements to Kieker records and therefore makes ΘPAD reusable in other environments (NFR4).

This graphic uses the AMQP notation of Figure 2.19.

Raw measurements, encoded as JSON strings, are sent to the measurement queue by the system under monitoring in a temporal sequence fashion as described in Section 3.4. The Kieker Monitoring component, then deserializes these strings and transforms them into measurement records as defined in Equation 3.4. Subsequently, these records are put into a Java Pipe forwards data in memory according to the FIFO principle. Figure 4.5 shows the corresponding class instantiation.

1. AMQP messages transformed into Kieker monitoring records
2. ΘPAD: pipes-and-filters processing of records
3. ΘPAD results passed to alerting queue and time-series storage
OPAD Processing Steps

OPAD: Online Performance Anomaly Detection

<<Reader>>
<<Filter>>
Time Series Extraction

<<Filter>>
Time Series Forecasting

<<Filter>>
Anomaly Score Calculation

<<Filter>>
Anomaly Detection

<<Filter>>
Alerting (e.g., AMQP)
Step 1: Time Series Extraction

ΘPAD Processing Steps (cont’d)

OPAD: Online Performance Anomaly Detection

Time Series Extraction
Time Series Forecasting
Anomaly Score Calculation
Anomaly Detection
Alerting (e.g., AMQP)

Continuous Time

Discrete Time Series

Event on ES
Discretization Function
Time Series X
Current Time

select sum(value) as aggregation
from MeasureEvent.win:time_batch( 1000 msec )
Step 2: Time Series Forecasting

OPAD Processing Steps (cont’d)

OPAD: Online Performance Anomaly Detection

- Time Series Extraction
- Time Series Forecasting
- Anomaly Score Calculation
- Anomaly Detection
- Alerting (e.g., AMQP)

\[ \Delta W \]

T. Bielefeld (empuxa) and A. van Hoorn (CAU)

ΘPAD w/ Kieker

Oct. 24, 2012 @ Hamburg
Step 3: Anomaly Score Calculation

ΘPAD Processing Steps (cont’d)

OPAD: Online Performance Anomaly Detection

- Time Series Extraction
- Time Series Forecasting
- Anomaly Score Calculation
- Anomaly Detection
- Alerting (e.g., AMQP)
Step 4: Anomaly Detection

ΘPAD Processing Steps (cont’d)

OPAD: Online Performance Anomaly Detection

Time Series Extraction

Time Series Forecasting

Anomaly Score Calculation

Anomaly Detection

Alerting (e.g., AMQP)

Abnormal Score

Normal Score

Anomaly Threshold

Anomaly Detected
ΘPAD Web Interface

OPAD: Online Performance Anomaly Detection

Query MongoDB:

DB: opad_development  Collection: evaluation_2w2011_2

Sort: {"time": 1}  Limit: 2000

From - to: 1324285440000 1324317000000

Zoom: out  left  right

Anomaly Score

0.75
0.5
0.25
0

Measures

81697367
72619882
63542396
54464911
45387426
36309941
27232456
18154970
90774858

Measure: 32274036.4
Forecast: 72931402.17
Anomaly Score: 0.3057
2011-12-19 15:15:00

1  2  3  4
Evaluation Methodology: GQM

OPAD: Online Performance Anomaly Detection

Goal
Assess Practicality of Approach

Questions
- How precise is the detection?
- How accurate is the detection?

Metric
- Number of true positives
- Number of false negatives

Goal/Question/Metric (GQM) plan (excerpt)
Manual detection using the visualization tool
8 anomalies were detected
PAD Results

Evaluation

OPAD: Online Performance Anomaly Detection

aptt.Fses.D1min.L15min

aptt.Fets.D1min.L1h

aptt.Fmean.D5min.L1h
ROC Curves (Introduction)

Evaluation (cont’d)

OPAD: Online Performance Anomaly Detection

\[ TPR = \frac{TP}{TP + FN} = \frac{TP}{F} \]

\[ FPR = \frac{FP}{FP + TN} = \frac{FP}{NF} \] (1)
ROC Curves (ΘPAD Results)

Evaluation (cont’d)

OPAD: Online Performance Anomaly Detection

![ROC Curves Diagram]

- mean.D1min.L1h
- ses.D1min.L15min
- mean.D20min.L2h
- aptt.Fws.D1h.L24h

T. Bielefeld (empuxa) and A. van Hoorn (CAU)
Accuracy and Precision
Evaluation (cont’d)
OPAD: Online Performance Anomaly Detection

\[
\text{ACC} = \frac{TP + TN}{N} = \frac{TP + TN}{TP + FP + FN + TN} \quad \cdot \quad (3)
\]

\[
\text{PREC} = \frac{TP}{POS} = \frac{TP}{TP + FP} \quad \cdot \quad (2)
\]

Outlook

- ΘPAD to be released as part of Kieker
- Follow-up theses on ΘPAD

Contact Us

- till@empuxa.com
- avh@informatik.uni-kiel.de
### KoSSE Symposium on Application Performance Management

Nov. 29/30, 2012 (Thu/Fri) @ Wissenschaftszentrum Kiel

### Invited talks by (see http://kosse-sh.de for details)

- b+m Informatik AG, Melsdorf
- Consist Software Solutions GmbH, Kiel
- empuxa GmbH, Kiel
- Karlsruhe Inst. of Technology (KIT)
- Kiel University
- QAware GmbH, Munich
- RWTH Aachen
- SAP Research, Karlsruhe
- XING AG, Hamburg

- Social event: Dinner at Forstbaumschule (Nov. 29, 18:30h)
- Registration required (no fee!): mail@diwish.de
T. C. Bielefeld. Online performance anomaly detection for large-scale software systems.  

http://kieker-monitoring.net/documentation/.


1. **Sequence diagrams**
2. Dynamic call trees
3. Hierarchical calling dependency graphs
4. System model

(a) Assembly-level view
(b) Deployment-level view
 bonus

1. Sequence diagrams
2. Dynamic call trees
3. Hierarchical calling dependency graphs
4. System model

(a) Dynamic call tree (single trace)

(b) Aggregated deployment-level call tree
1. Sequence diagrams
2. Dynamic call trees
3. **Hierarchical calling dependency graphs**
4. System model

(a) **Assembly-level component** dependency graph

(b) **Deployment-level operation** dependency graph
1. Sequence diagrams
2. Dynamic call trees
3. Hierarchical calling dependency graphs
4. **System model** (here: HTML representation)