Visualization of Performance Anomalies with Kieker
Bachelor’s Thesis

Sören Henning

September 8, 2016
Outline

1. Introduction
2. Foundations
3. Approach
4. Evaluation
5. Conclusion and Future Work
Performance is a significant quality characteristic
- e.g., Amazon: 100 ms delay $\rightarrow$ 1% decrease in sales (Huang 2011)
- e.g., Google: 500 ms delay $\rightarrow$ 20% drop in traffic (Huang 2011)
Performance is a significant quality characteristic

- e.g., Amazon: 100 ms delay → 1% decrease in sales (Huang 2011)
- e.g., Google: 500 ms delay → 20% drop in traffic (Huang 2011)

Detect performance issues to react on them

- As soon as possible
- Use monitoring (e.g., measure execution times)
- Investigate these measurements for anomalies
Performance is a significant quality characteristic
  - e.g., Amazon: 100 ms delay $\rightarrow$ 1% decrease in sales (Huang 2011)
  - e.g., Google: 500 ms delay $\rightarrow$ 20% drop in traffic (Huang 2011)

Detect performance issues to react on them
  - As soon as possible
  - Use monitoring (e.g., measure execution times)
  - Investigate these measurements for anomalies

Visualization can support interpretation of anomalies
θPAD’s Anomaly Detection Approach

Introduction

- Provide anomaly detection
- Part of Kieker
- Only R algorithms
- Problematic anomaly score
- No visualization
- More on this later
Development of an approach to detect performance anomalies with Kieker and visualize them
Goals

Introduction

- G1: Migrate ΘPAD to a TeeTime configuration
Goals

Introduction

▶ G1: Migrate ΘPAD to a TeeTime configuration
▶ G2: Implement multiple forecast algorithms
Goals

Introduction

- G1: Migrate ΘPAD to a TeeTime configuration
- G2: Implement multiple forecast algorithms
- G3: Provide a visualization of measured time series and detected anomalies
Goals

Introduction

- G1: Migrate ΘPAD to a TeeTime configuration
- G2: Implement multiple forecast algorithms
- G3: Provide a visualization of measured time series and detected anomalies
- G4: Evaluate the Implementation
  - G4.1. Feasibility evaluation
  - G4.2. Scalability evaluation
1. Introduction

2. Foundations

3. Approach

4. Evaluation

5. Conclusion and Future Work
Performance Metrics (Koziolek 2008)
- Time behavior and resource efficiency

Time Series (Mitsa 2010)
- Sequence of measurements at regular temporal intervals

Anomaly Detection (Chandola, Banerjee, and Kumar 2009)
- Anomaly: Abnormal data patterns
- Detection: Compare measured values with reference model
**Figure**: Based on Bielefeld (2012) and Frotscher (2013)
Figure: Based on Bielefeld (2012) and Frotscher (2013)
$\Theta$PAD’s Anomaly Detection Approach

Anomaly Score Calculation

Foundations

$$A(a,p) = 0.33$$

anomaly score calculation

anomaly decision

threshold $t = 0.2$

$0.33 \geq 0.2$
▶ Microservices Architectural Pattern (Wolff 2015)
▶ Kieker Monitoring Framework (Hoorn et al. 2009)
▶ TeeTime Pipe and Filter Framework (Wulf, Ehmke, and Hasselbring 2014)
▶ And further technologies (see next slides)
Outline

Approach

1. Introduction
2. Foundations
3. Approach
4. Evaluation
5. Conclusion and Future Work
Graphical Overview of our Approach

Approach

Application Monitoring

Anomaly Detection

Anomaly Visualization

measurements

time series
Architecture of our Approach

Approach

Visualization

Provider

Analysis

R-based Forecast

Database

Visualization

Server

Client

Sören Henning

Visualization of Performance Anomalies

September 8, 2016

15 / 41
Architecture of our Implementation

Approach

Server

Client

Containerized with

CanvasPlot

TeeTime

spring

cassandra

Containerized with

docker

Sören Henning
Filter:

- Operation signature
- Class signature
- Host name
- ...
Performance Anomaly Detection

TeeTime Configuration

Approach

TCP Reader → Flow Record Filter → Record Reconstructor → Record Distributor → Record Converter → Anomaly Detector

Record Converter → Anomaly Detector

Record Converter → Anomaly Detector

...
Performance Anomaly Detection

TeeTime Configuration

Approach

TCP Reader → Flow Record Filter → Record Reconstructor → Record Distributor → Record Converter → Anomaly Detector

...
### Time Series Analysis and Anomaly Detection

#### TeeTime Configuration

**Approach**

- **Database Adapter**
  - Interface
  - on startup

**Anomaly Detection Stage**

- **Sliding Window**
  - In Memory

- **Time Series Loader**
- **Normalizer**
- **Forecaster**

- **Measurement Forecast Decorator**

- **Anomaly Score Calculator**

- **Distributor**
  - **Threshold Filter**
  - **Storager**

---

**Sören Henning**

Visualization of Performance Anomalies

September 8, 2016
Demo of Visualization

Approach

KiekPAD | Anomaly Detection

demo-method

Visualization of Performance Anomalies

Sören Henning

September 8, 2016
Usage of Arne Johanson’s CanvasPlot (Johanson 2016)
Outline

1. Introduction
2. Foundations
3. Approach
4. Evaluation
5. Conclusion and Future Work
Feasibility Evaluation

Scenarios Evaluation

Response Time in ms

Time in ms

Response Time in ms

Time in ms

Response Time in ms

Time in ms

Response Time in ms

Time in ms
Feasibility Evaluation
Scenario: Seasonal with Anomaly

Response Time in ms vs Time in ms

Anomaly Score vs Time in ms

- ARIMA Forecaster
- Exponential Weighted Forecaster
- Linear Weighted Forecaster
- Logarithmic Weighted Forecaster
- Mean Forecaster
- Regression Forecaster
Take time for record processing in analysis
Take time for record processing in analysis
Evaluate: Execution time $\leq$ measurement frequency?
Scalability Evaluation

Configuration

Evaluation

- Take time for record processing in analysis
- Evaluate: Execution time $\leq$ measurement frequency?
- For all parameter combinations:

  **Measurement frequencies**  2 ms, 5 ms, 10 ms, 50 ms, 100 ms, 150 ms, 200 ms

  **Sliding window**  10,000 ms, 50,000 ms, 100,000 ms, 150,000 ms, 200,000 ms, 400,000 ms

  **Normalization interval**  10 ms, 20 ms, 100 ms, 200 ms, 500 ms, 1000 ms, 2000 ms

  **Forecast algorithm**  ARIMAForecaster, RegressionForecaster

  **Normalization algorithm**  MeanAggregator
### Some examples:

<table>
<thead>
<tr>
<th>freq.</th>
<th>sld. window</th>
<th>norm. intvl.</th>
<th>forecaster</th>
<th>(\emptyset) exec. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10,000</td>
<td>200</td>
<td>Regression</td>
<td>1.64</td>
</tr>
<tr>
<td>5</td>
<td>400,000</td>
<td>20</td>
<td>Regression</td>
<td>4.35</td>
</tr>
<tr>
<td>50</td>
<td>50,000</td>
<td>200</td>
<td>ARIMA</td>
<td>69.98</td>
</tr>
<tr>
<td>100</td>
<td>100,000</td>
<td>500</td>
<td>ARIMA</td>
<td>78.24</td>
</tr>
<tr>
<td>150</td>
<td>200,000</td>
<td>2,000</td>
<td>ARIMA</td>
<td>187.21</td>
</tr>
</tbody>
</table>

... all values in ms
Outline

1. Introduction
2. Foundations
3. Approach
4. Evaluation
5. Conclusion and Future Work
Further development of the ΘPAD Approach

- Proving infrastructure via Docker containers
- Immediately record processing

All implementations available as open source: github.com/SoerenHenning
Conclusion and Future Work

- Further development of the ΘPAD Approach
  - Proving infrastructure via Docker containers
  - Immediately record processing
- Native Java algorithms for anomaly detection
Conclusion and Future Work

- Further development of the ΘPAD Approach
  - Proving infrastructure via Docker containers
  - Immediately record processing
- Native Java algorithms for anomaly detection
- Providing an interactive, real time visualization
Conclusion and Future Work

- Further development of the ΘPAD Approach
  - Proving infrastructure via Docker containers
  - Immediately record processing
- Native Java algorithms for anomaly detection
- Providing an interactive, real time visualization
- All implementations available as open source: github.com/SoerenHenning
Handling of fast incoming measurements
  ▶ Aggregate before analysis (ΘPAD)
  ▶ Cache time series operations
Future Work

Conclusion and Future Work

- Handling of fast incoming measurements
  - Aggregate before analysis ($\Theta$PAD)
  - Cache time series operations
- Parallelized and distributed analysis
  - Is or will be supported by TeeTime

Sören Henning
Visualization of Performance Anomalies
September 8, 2016

28 / 41
Future Work

Conclusion and Future Work

- Handling of fast incoming measurements
  - Aggregate before analysis (ΘPAD)
  - Cache time series operations
- Parallelized and distributed analysis
  - Is or will be supported by TeeTime
- Take advantage of Cassandra’s features for data storage
Future Work

Conclusion and Future Work

- Handling of fast incoming measurements
  - Aggregate before analysis (ΘPAD)
  - Cache time series operations
- Parallelized and distributed analysis
  - Is or will be supported by TeeTime
- Take advantage of Cassandra’s features for data storage
- Configuration via Rest/GUI


Feasibility Evaluation
Scenario: Constant with Anomaly

Feasibility Evaluation

Response Time in ms

Time in ms

Anomaly Score

Time in ms

ARIMAForecaster
LinearWeightedForecaster
MeanForecaster
ExponentialWeightedForecaster
LogarithmicWeightedForecaster
RegressionForecaster
Feasibility Evaluation

Scenario: Constant with Anomaly - Detail

Feasibility Evaluation

![Graph showing response time and anomaly score over time]

- ARIMA Forecaster
- Exponential Weighted Forecaster
- Linear Weighted Forecaster
- Logarithmic Weighted Forecaster
- Mean Forecaster
- Regression Forecaster

Sören Henning
Visualization of Performance Anomalies
September 8, 2016
Feasibility Evaluation
Scenario: Linearly Increasing with Anomaly

Response Time in ms

Anomaly Score

ARIMA Forecaster
Exponential Weighted Forecaster
Linear Weighted Forecaster
Logarithmic Weighted Forecaster
Mean Forecaster
Regression Forecaster
Feasibility Evaluation

Scenario: Linearly Increasing with Anomaly - Detail

Feasibility Evaluation

- ARIMAForecaster
- ExponentialWeightedForecaster
- LinearWeightedForecaster
- LogarithmicWeightedForecaster
- MeanForecaster
- RegressionForecaster

Time in ms

Anomaly Score

Response Time in ms
Feasibility Evaluation
Scenario: Seasonal with Anomaly

Response Time in ms

Anomaly Score

Time in ms

ARIMAForecaster
LinearWeightedForecaster
MeanForecaster
ExponentialWeightedForecaster
LogarithmicWeightedForecaster
RegressionForecaster

Sören Henning
Visualization of Performance Anomalies
September 8, 2016
Feasibility Evaluation

Scenario: Seasonal with Anomaly - Detail

Response Time in ms vs. Time in ms

Anomaly Score vs. Time in ms

Comparison of different forecasting methods:
- ARIMA Forecaster
- Exponential Weighted Forecaster
- Linear Weighted Forecaster
- Logarithmic Weighted Forecaster
- Mean Forecaster
- Regression Forecaster
Feasibility Evaluation

Scenario: Exponential Increasing

Feasibility Evaluation

Response Time in ms

Time in ms

Anomaly Score

Time in ms

ARIMAForecaster | LinearWeightedForecaster | MeanForecaster
ExponentialWeightedForecaster | LogarithmicWeightedForecaster | RegressionForecaster
Screenshots of Visualization

Feasibility Evaluation

KiekPAD   Anomaly Detection

demo-method

Graph showing anomalies in response time with peaks indicating potential performance issues.
Screenshots of Visualization

Feasibility Evaluation

KlekPAD | Anomaly Detection

[Graph showing response time and anomaly scores over time]

Predictions | Anomaly Scores | Thresholds | -0.3 | 0.3 | Refresh Interval | 500

Sören Henning

Visualization of Performance Anomalies

September 8, 2016