



**FACULTY  
OF MATHEMATICS  
AND PHYSICS**  
Charles University

**MASTER THESIS**

Jakub Háva

**Monitoring Tool for Distributed Java  
Applications**

Department of Distributed and Dependable Systems

Supervisor of the master thesis: Mgr. Pavel Parízek, Ph.D.

Study programme: Computer Science

Study branch: Software Systems

Prague 2017

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....

signature of the author

Title: Monitoring Tool for Distributed Java Applications

Author: Jakub Háva

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Pavel Parízek, Ph.D., Department of Distributed and Dependable Systems

Abstract: TODO: Abstract

Keywords: key words

TODO: Write dedication

# Contents

<b>Introduction</b>	<b>3</b>
<b>1 Project Goals</b>	<b>4</b>
<b>2 Similar Work</b>	<b>5</b>
2.1 Google Dapper . . . . .	5
2.2 Zipkin . . . . .	5
<b>3 Analysis</b>	<b>6</b>
3.1 Instrumentation libraries . . . . .	6
3.1.1 Javassist . . . . .	6
3.1.2 ByteBuddy . . . . .	6
3.1.3 CGlib . . . . .	6
3.1.4 ASM . . . . .	6
3.2 Communication Middleware . . . . .	6
3.2.1 ZeroMQ . . . . .	6
3.2.2 NanoMSG . . . . .	6
3.3 Comparison of Agent Approaches . . . . .	6
3.3.1 Java Agent Solution . . . . .	6
3.3.2 Native Agent Solution . . . . .	6
<b>4 Used Technologies</b>	<b>7</b>
4.1 Java . . . . .	7
4.1.1 Class Initialization Process . . . . .	7
4.1.2 JVMTI . . . . .	7
4.1.3 JNI . . . . .	7
4.1.4 ClassLoaders . . . . .	7
4.2 ByteBuddy . . . . .	7
4.2.1 Main Concept . . . . .	7
4.2.2 Transformers . . . . .	7
4.2.3 Intereptors . . . . .	7
4.2.4 Class File Locator . . . . .	7
4.2.5 Advice API . . . . .	7
4.3 NanoMgs . . . . .	7
4.3.1 C++11 Mapping . . . . .	7
4.3.2 Java Mapping . . . . .	7
4.4 spdlog . . . . .	7
<b>5 Platform Architecture</b>	<b>8</b>
5.1 Architecture Description . . . . .	8
5.2 Communication . . . . .	8
<b>6 Native Agent</b>	<b>9</b>
6.1 Structure Overview . . . . .	9
6.2 Instrumentation API . . . . .	9
6.3 Byte Class Parsing . . . . .	9

6.4	Instrumentation . . . . .	9
6.5	Native Agent Arguments . . . . .	9
<b>7</b>	<b>Instrumentation Server</b>	<b>10</b>
7.1	Instrumentation Handling . . . . .	10
7.2	Communication modes . . . . .	10
7.3	Class Caching . . . . .	10
<b>8</b>	<b>Instrumentation Library</b>	<b>11</b>
8.1	Custom Service Loader . . . . .	11
8.2	Public interfaces . . . . .	11
8.3	Extending the Library . . . . .	11
8.4	ClassLoaders . . . . .	11
8.5	JSON Generation . . . . .	11
<b>9</b>	<b>User Interface</b>	<b>12</b>
9.1	Zipkin Overview . . . . .	12
9.2	Zipkin Data Model . . . . .	12
9.3	Zipkin JSON Format . . . . .	12
<b>10</b>	<b>Collectors</b>	<b>13</b>
<b>11</b>	<b>Deployment Strategies</b>	<b>14</b>
11.1	Instrumentor on the same node with the Application . . . . .	14
11.2	Instrumentor available over the Network . . . . .	14
11.3	Bundling the application classes with the Instrumentor . . . . .	14
<b>12</b>	<b>Platform demonstration</b>	<b>15</b>
12.1	Bulding Monitoring tool on top of Distrace . . . . .	15
12.2	Basic Demonstration . . . . .	15
12.3	Optimizing the instrumentation . . . . .	15
<b>13</b>	<b>Future plans</b>	<b>16</b>
13.1	Integration with well-known data collectors . . . . .	16
13.2	Add support for Flame charts . . . . .	16
<b>14</b>	<b>Docker Support</b>	<b>17</b>
<b>15</b>	<b>Conclusion</b>	<b>18</b>
15.1	Title of the first subchapter of the first chapter . . . . .	18
15.2	Title of the second subchapter of the first chapter . . . . .	18
	<b>Conclusion</b>	<b>19</b>
	<b>List of Figures</b>	<b>20</b>
	<b>List of Tables</b>	<b>21</b>
	<b>List of Abbreviations</b>	<b>22</b>
	<b>Attachments</b>	<b>23</b>

# Introduction

# 1. Project Goals



## 2. Similar Work

2.1 Google Dapper

2.2 Zipkin

## **3. Analysis**

### **3.1 Instrumentation libraries**

#### **3.1.1 Javassist**

#### **3.1.2 ByteBuddy**

#### **3.1.3 CGlib**

#### **3.1.4 ASM**

.. just give brief overview what were the instrumentation libraries choices. The selected one will be described in the next section

### **3.2 Communication Middleware**

#### **3.2.1 ZeroMQ**

#### **3.2.2 NanoMSG**

### **3.3 Comparison of Agent Approaches**

#### **3.3.1 Java Agent Solution**

#### **3.3.2 Native Agent Solution**

## 4. Used Technologies

### 4.1 Java

#### 4.1.1 Class Initialization Process

#### 4.1.2 JVMTI

#### 4.1.3 JNI

#### 4.1.4 ClassLoaders

### 4.2 ByteBuddy

#### 4.2.1 Main Concept

#### 4.2.2 Transformers

#### 4.2.3 Intereptors

#### 4.2.4 Class File Locator

#### 4.2.5 Advice API

### 4.3 NanoMgs

#### 4.3.1 C++11 Mapping

#### 4.3.2 Java Mapping

### 4.4 spdlog

logging library used

# 5. Platform Architecture

## 5.1 Architecture Description

## 5.2 Communication

## 6. Native Agent

### 6.1 Structure Overview

### 6.2 Instrumentation API

### 6.3 Byte Class Parsing

### 6.4 Instrumentation

### 6.5 Native Agent Arguments

## 7. Instrumentation Server

### 7.1 Instrumentation Handling

### 7.2 Communication modes

### 7.3 Class Caching

## 8. Instrumentation Library

### 8.1 Custom Service Loader

### 8.2 Public interfaces

### 8.3 Extending the Library

.. instrumentation server can run on the same node or over the network. Instrumentation server can have client code attached or not.

### 8.4 ClassLoaders

### 8.5 JSON Generation

## 9. User Interface

### 9.1 Zipkin Overview

### 9.2 Zipkin Data Model

### 9.3 Zipkin JSON Format



## 10. Collectors

Should I mention the collectors ? It may be sufficient to have send data right to zipkin for demonstration purposes

# 11. Deployment Strategies

- 11.1 Instrumentor on the same node with the Application
- 11.2 Instrumentor available over the Network
- 11.3 Bundling the application classes with the Instrumentor

## 12. Platform demonstration

12.1 Bulding Monitoring tool on top of Distrace

12.2 Basic Demonstration

12.3 Optimizing the instrumentation

## 13. Future plans

13.1 Integration with well-known data collectors

13.2 Add support for Flame charts

## 14. Docker Support

# 15. Conclusion

An example citation: ?

**15.1 Title of the first subchapter of the first chapter**

**15.2 Title of the second subchapter of the first chapter**

# Conclusion

# List of Figures



# List of Tables

# List of Abbreviations

# Attachments