Master's Degree in Informatics Engineering

Thesis - Intermediate Report

Observing and Controlling Performance in Microservices

Author:

André Pascoal Bento

Supervisor:

Prof. Filipe João Boavista Mendonça Machado Araújo

Co-Supervisor(S):

Prof. Jorge Cardoso

Jaime Correia???

UNIVERSIDADE D COIMBRA

December 2018

This page is intentionally left blank.

Abstract

In a world of increasingly decoupled microservices ...

Keywords

Microservices, Cloud Computing, Observing, Monitoring.

This page is intentionally left blank.

Resumo

Hoje em dia encontramos-nos num mundo em que a crescente evolução técnológica, exige cada vez mais dos sistemas computacionais e das pessoas que os desenvolvem e mantêm. Com este crescimento, a complexidade e a distribuíção dos sistemas aumenta a passos largos, de modo a que se torna bastante dificil de os gerir e de perceber o seu funcionamento em geral. É neste problema que a presente tese visa prestar soluções. As soluções apresentadas neste documento, têm como principal objectivo permitir ???

Palavras-Chave

Micro-serviços, Computação na núvem, Observação, Monitorização.

This page is intentionally left blank.

Contents

This page is intentionally left blank.

Acronyms

API Application Programming Interface. 6, 7

DEI Department of Informatics Engineering. 1

GDB Graph Database. 4

HTTP Hypertext Transfer Protocol. 4

RPC Remote Procedure Call. 4

TSDB Time Series Database. 4

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

This document represents the *Master Thesis* in *Informatics Engineering*, of the student *André Pascoal Bento* during the school year of 2018/2019, taking place in the *Department of Informatics Engineering (DEI)* of the *University of Coimbra*.

1.1 Context

- Introduce the thesis theme explaining involving the reader about the theme and concept of the thesis.

1.2 Motivation

- Explain and present the motivation about the theme behind this thesis.

1.3 Goals

- Present the main goals of this project.

1.4 Work Plan

- Explain and present the work plan for this project.

Plano de Trabalhos - Semestre 1

- Study the state of the art, namely existing tools, and related technologies like tracing (2 months);

- Integrate the existing work (1 month);

- Define requirements of the monitoring tool (1 month);

- Write intermediate report (1 month).

Plano de Trabalhos - Semestre 2

- Implement the monitoring tool (2 months);

- Test and evaluate the results (2 months);

- Write final report (1 month);

1.5 Document Structure

- Present the document structure of this report.
- Refer the next sections and explain what they contains.

Chapter 2

State of the Art

In the following chapter, we will be discussing the core concepts regarding the project subject and the most modern techniques and technology that's available for the purpose today. All the information that will be presented was the result of a work of investigation through published articles, knowledge exchange and web searching.

The main purpose of the section 2.1 is to introduce and provide a brief explanation about the core concepts. In the second section 2.2, all the important technologies are analysed and discussed using tables and diagrams.

2.1 Concepts

The following concepts represents the baseline to understand the work related to this project.

2.1.1 Microservices

Microservices is "an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities"[5].

2.1.2 Observing

Observing is "to be or become aware of, especially through careful and directed attention; to notice"[6].

The presented definition represents the meaning of the word Observing and it is reflected exactly as it is in the project context. For example, observe the interaction between some microservices to notice a fault.

When we want to understand the working and behaviour of a system, we need to watch it very closely and pay special attention to all information it provides. This information may be in multiple structured text formats, and it can contain lots of information regarding the interaction between microservices and the access to them. Therefore, we may work with this information as a starting point to perceive the characteristics of the system and build a tool that is able to be aware of it.

2.1.3 Controlling Performance

- Explain whats is Controlling with a definition and whats Performance

2.1.4 Traces and Spans

First things first, we can think in a trace as a group of spans. A trace is a representation of a data/execution path through the system and a span represents the logical unit of work in the system. The span has an operation name, the start time of the operation, and the its duration. An example of a span can be an Hypertext Transfer Protocol (HTTP) call or an Remote Procedure Call (RPC) call. With a couple of spans, we might be able to model a graph of a portion of the system, because they represents causal relationships in the system.

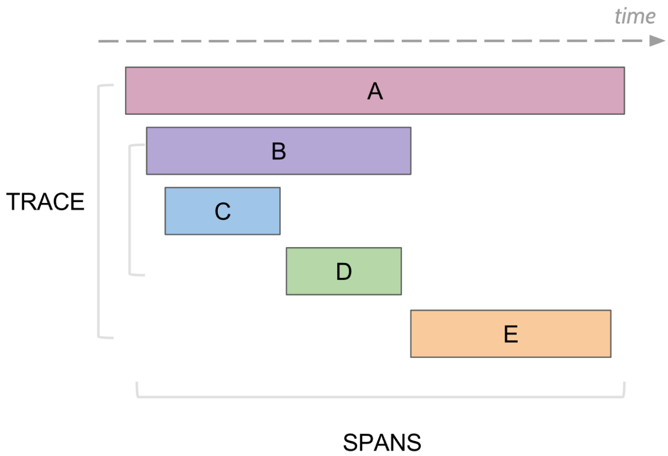

Figure 2.1: Traces and spans disposition over time.

As you can see in the figure 2.1, the spans spread over time, overlapping each other, since nothing prevents the occurrence of multiple calls in very close times.

2.1.5 Graph Database

A Graph Database (GDB) is "a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data"[8].

The composition of a GDB is based on the mathematics graph theory, and therefore this databases uses three main components called nodes, edges, and properties. This main components are defined and explained in the following list:

- Node:
- Edge:
- Property:

2.1.6 Time Series Database

Time Series Database (TSDB)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sed efficitur magna, et laoreet elit. Praesent sed nulla est. Cras in sem nec sem posuere hendrerit eget eu ligula. Duis tincidunt, lacus quis sollicitudin porttitor, nibh lorem tempor elit, nec tempus nunc ante id felis. Sed volutpat massa nibh, eu tempor ipsum tempor vel. tempor neque.

2.2 Technologies

In this section is presented the technologies and tools that were researched, as well as the discussion considering the objectives defined for this project.

2.2.1 Graph Database Tools

With the concept presented in the section 2.1.5 in mind, we needed to search and find the best technologies available at the time, and that can get the job done. The results of the research are presented in the table 2.1.

2.2.2 Graph Manipulation Tools

Storing data is not enough, we have to be sure that we can handle it. For this purpose, we needed to study the frameworks available for graph manipulation. The table 2.2 presents the main technologies available at the time for graph manipulation.

Table 2.1: Graph databases comparison.

Name	ArangoDB [2]	Neo4J [4]	Facebook TAO [3]
Description	A NoSQL database developed by ArangoDB Inc., that uses a proper query language to access the database.	The most popular open source graph database. It has been developed by Neo4J Inc. and is completely open to the community.	TAO, "The Associations and Objects", is a proprietary database, developed by Facebook Inc., that stores all the data related to the users in the social network.
Licence[9]	Free Apache 2	GPLv3 Community Edition	Proprietary
Supported languages	C++, JavaScript, .NET, Java, Python, Node.js, PHP, Scala, Go, Ruby, Elixir.	Java, .NET, JavaScript, Python, Ruby.	
Pros	Multi data-type support (key/value, documents, graphs). Allows the combination of different data access patterns in a single query. Supports cluster deployment.	Supports ACID(Atomicity, Consistency, Isolation, Durability)[7]. High-availability. Has a visual node-link graph explorer. REST API interface. Most popular open source graph database.	Very fast(=100ms latency). Accepts millions of calls per second. Distributed.
Cons	Needed to learn a new query language called AQL(Arango Query Language). High leaning curve. Has paid version with high price tag.	Can't be distributed (It needs to be vertically scaled).	Not accessible to use.

Table 2.2: Graph manipulation tools comparison.

Name	iGraph	NetworkX	Zen
Description	A library collection for graph creation and manipulation, and for analysing networks.	A Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.	A library that provides a high-speed, easy-to-use API for analysing, visualising and manipulating network graphs.
Licence[9]	GNU GPL2	BSD - New License	BSD
Supported languages	C, C++, Python, R.	Python.	Python.
Pros	Very good performance when used with C/C++. Easy integration with C/C++/R.	Good availability. Easy to install with Python. Mature project. Low performance when there's a tremendous amount of data.	Easy API.
Cons	Lack of documentation.	High learning curve due to the maturity of the project.	Very new project. High number of issues declared/to solve in the repository. Hard installation because of lack of support for Pip [1].

Chapter 3

Research Objectives and Approach

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sed efficitur magna, et laoreet elit. Praesent sed nulla est. Cras in sem nec sem posuere hendrerit eget eu ligula. Duis tincidunt, lacus quis sollicitudin porttitor, nibh lorem tempor elit, nec tempus nunc ante id felis.

This page is intentionally left blank.

Chapter 4

Solution

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sed efficitur magna, et laoreet elit. Praesent sed nulla est. Cras in sem nec sem posuere hendrerit eget eu ligula. Duis tincidunt, lacus quis sollicitudin porttitor, nibh lorem tempor elit, nec tempus nunc ante id felis.

References

[1] Python Software Foundation. Pip documentation. https://pypi.org/project/pip/. Accessed: 2018-10-16.

[2] ArangoDB Inc. Arangodb documentation. https://www.arangodb.com/documentation/. Accessed: 2018-10-16.

[3] Facebook Inc. Tao: Facebook's distributed data store for the social graph. https://cs.uwaterloo.ca/~brecht/courses/854-Emerging-2014/readings/data-store/tao-facebook-distributed-datastore-atc-2013.pdf. Accessed: 2018-10-15.

[4] Neo4J Inc. Neo4j documentation. https://neo4j.com/docs/. Accessed: 2018-10-16.

[5] Chris Richardson. Microservices definition. https://microservices.io/. Accessed: 2018-10-17.

[6] thefreedictionary.com. Observing definition. https://www.thefreedictionary.com/observing. Accessed: 2018-10-13.

[7] Wikipedia. Acid(computer science). https://en.wikipedia.org/wiki/ACID_(computer_science). Accessed: 2018-10-16.

[8] Wikipedia. Graph database. https://en.wikipedia.org/wiki/Graph_database. Accessed: 2018-10-15.

[9] Wikipedia. Software License. https://en.wikipedia.org/wiki/Software_license. Accessed: 2018-10-16.

Appendices

Appendix A

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sed efficitur magna, et laoreet elit. Praesent sed nulla est. Cras in sem nec sem posuere hendrerit eget eu ligula. Duis tincidunt, lacus quis sollicitudin porttitor, nibh lorem tempor elit, nec tempus nunc ante id felis.

This page is intentionally left blank.

Appendix B

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sed efficitur magna, et laoreet elit. Praesent sed nulla est. Cras in sem nec sem posuere hendrerit eget eu ligula. Duis tincidunt, lacus quis sollicitudin porttitor, nibh lorem tempor elit, nec tempus nunc ante id felis.