UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group DICE Group

Bachelor's Thesis Proposal

Submitted to the DICE Group Research Group
in Partial Fullfilment of the Requirements for the Degree of

Bachelor of Science

Basilisk – Continuous Benchmarking for Triplestores

by
Fabian Rensing

Thesis Supervisor:
Prof. Dr. Axel-Cyrille Ngonga Ngomo

Paderborn, November 9, 2021

1 Description and Motivation

In the field of Semantic Web, knowledge graphs are an important structure to represent data and its connections. To easily store and query the data in these knowledge graphs, some data structure or database is needed. The special kind of database developed for knowledge graphs are called Triplestores.

Since knowledge graphs can contain huge amount of data and can also be subject to many changes, Triplestores need to be able to handle many different workloads. To better test and compare Triplestores in these diverse scenarios, benchmarks are performed to allow an appropriate comparison between different Triplestores. Often Triplestores are developed in long iterations and are bench-marked only in a late stage of such an development iteration. Today benchmarks and the evaluation of their results are usually done manually and bind developers time. Thus, performance regressions are found very late or never.

Benchmarks in general are used to measure and compare the performance of computer programs and systems with a defined set of operations. Often they are designed to mimic and reproduce a particular type of workload to the system. In the context of Triplestores, a benchmark usually consists of creating a big knowledge graph and performing multiple queries and operations on the data.

Several benchmarks for Triplestores have been proposed. Iguana is a benchmark-independent execution framework [2] that can measure the performance of Triplestores under several parallel query request. Currently the benchmark execution framework needs to be installed and benchmarks need to be started manually. Basilisk is a continuous benchmarking service for Triplestores which internally uses Iguana to perform the benchmarks. The idea is that the Basilisk service will check automatically for new versions of Triplestores and start benchmarks with the Iguana framework. Further it should be possible to start custom benchmarks on demand. If a new version is found in a provided GitHub- or DockerHub-repository, Basilisk will automatically setup a benchmark environment and starts a benchmarking suite.

This means that developers do not have to worry about performing benchmarks at different stages of development.

2 Formulation of Target Setting

The target of this thesis is to describe the software architecture, deploy the system and perform some benchmarks of different triple stores.

The software architecture of Basilisk was developed through a master thesis beforehand.

and review the software architecture that has been used for the Basilisk platform.

3 Tasks Descriptions

The enumerated tasks for the thesis are defined here and correspond to the schedule shown in chapter 4.

Task 1	Software architecture analysis and review
Time Required	3 Weeks
Description	Analysis and review of the current software architecture for the Basilisk service.

Task 2	Solution Design
Time Required	3 Weeks
Description	The missing parts of the implementation for the Basilisk service will be designed and planned out.

Task 3	Solution Implementation
Time Required	4 Weeks
Description	The previously designed solution will be implemented into the existing architecture.

Task 4	Deploy Basilisk and its frontend
Time Required	2 Weeks
Description	The Basilisk service and the frontend will be deployed on a publicly available VM.

Task 5	Perform benchmarks for Triplestores
Time Required	3 Weeks
Description	Benchmarks for two versions of the Tentris Triplestore[1] and one different Triplestore will be benchmarked.

Task 6	Fix critical bugs, document non-critical
Time Required	2 Weeks
Description	Critical Bugs and errors found in Tasks 1-5 will be fixed and non-critical will be documented.

Task 7	Thesis writing
Time Required	19 Weeks
Description	The thesis will be written parallel to the work on the first Tasks.

4 Schedule

The Schedule for the proposed task in Chapter 3 is shown in figure 4.1

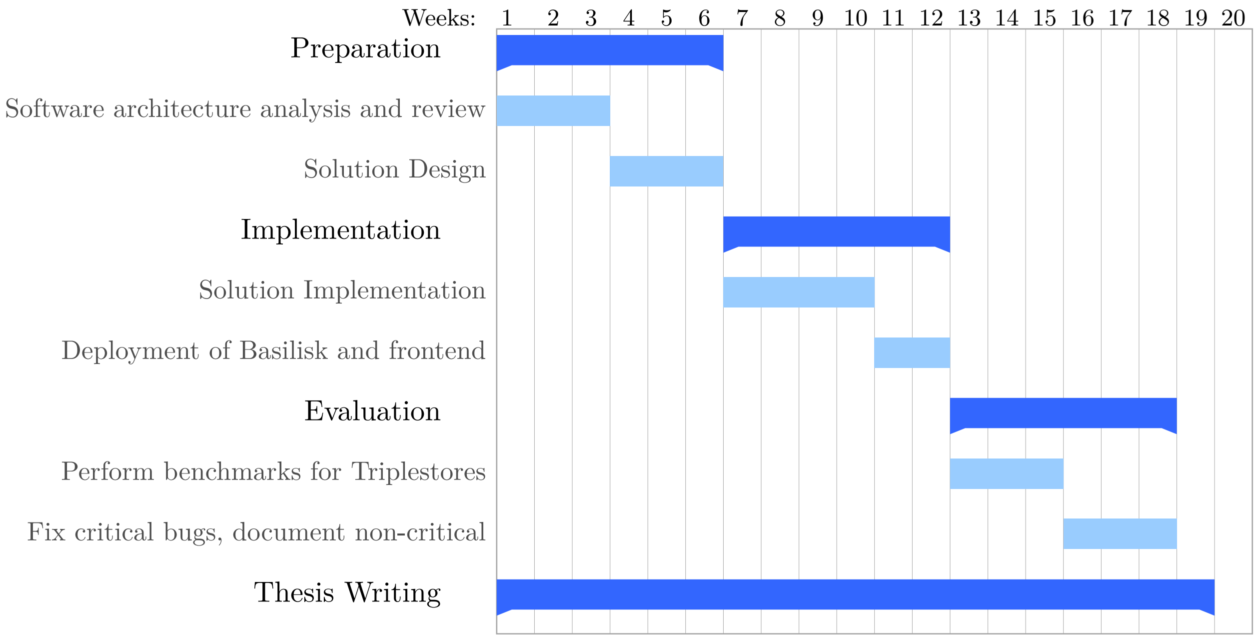

Figure 4.1: Proposed schedule for the work on the thesis.

5 Preliminary Outline of the Thesis

The thesis will comprise the following chapters

1. Introduction:
   - Description of the topic and the context in which it is inserted
   - Topic motivation
2. Related Work
   - Description of existing techniques and approaches
3. Background
   - Description of foundational knowledge required to understand the thesis
4. Approach
   - Description of the designed system
   - Analysis and review of the current software architecture
5. Implementation
   - Solution design and implementation
   - Deployment of the service
6. Evaluation
   - Experiment setup, requirements
   - Performing of benchmarks
   - Result evaluation
7. Summary and Discussion
   - Summary of the work
   - Highlighting the key findings of the evaluation stage

Bibliography

[1] Alexander Bigerl, Felix Conrads, Charlotte Behning, Mohamed Ahmed Sherif, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Tentris – A Tensor-Based Triple Store. In The Semantic Web – ISWC 2020, pages 56–73. Springer International Publishing, 2020.

[2] Felix Conrads, Jens Lehmann, Muhammad Saleem, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. Iguana: A generic framework for benchmarking the read-write performance of triple stores. In Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin, editors, The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II, volume 10588 of Lecture Notes in Computer Science, pages 48–65. Springer, 2017.