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, October 19, 2021

1 Description

Triplestores are the database backend of knowledge graphs. They are needed to easily store and query data of knowledge graphs used in the Semantic Web. Since these Triplestores need to handle a lot of different data and access types, it is a good idea to perform benchmarks. 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 a particular type of workload to the system. In the context of Triplestores, a benchmark usually consists of creating a big knowledge graph, perform multiple queries and operations on the data.

—

What general topic are we interested in, why does it matter, where is it used, what problem does it solve? Explain some more background for the topic, help the reader, who may have never heard of the topic, understand what you are talking about.

Current state of the research: - What is a trivial way to solve x? - How is x usually solved in practice? - what are typical ingredients / techniques to construct the systems we are interested in? - How does ? solve the problem? - What does the algo in ? do roughly?

Benchmarking erklären? Triplestores erklären motivierend

- halbe Seite

2 Motivation

Triplestores are the database backend of knowledge graphs. They are typically developed in long iterations and are bench-marked, if at all, in a very late stage of such an development iteration. Typically Benchmarking and evaluation of the results are done manually and bind developers time. Thus, performance regressions are found very late or never.

—

The topic I have to offer has the working title is "Basilisk – Continuous Benchmarking for Triplestores". At the core, it is developing and deploying a CI/CD tool that hooks into github and/or docker registries.

Described in more detail:

Triplestores – the database backend of knowledge graphs – are typically developed in long iterations and are bench-marked – if at all – only in a very late stage of such an iteration. Benchmarking and evaluation of benchmarking results are typically done manually and binds developer's time. Thus, performance regressions are found very late or never.

With Basilisk we started to develop a continuous benchmarking platform for triplestore which hooks into github and docker image registries.

On events like pull requests or newly published versions of triplestores, a benchmarking suite is run automatically.

The first version of Basilisk (https://github.com/dice-group/Basilisk, https://github.com/dice-group/basilisk-frontend) is already implemented. It is based on the benchmarking tool IGUANA (https://github.com/dice-group/IGUANA) and Docker. (It requires triple stores to be dockerized).

The thesis task is to:

The thesis can be extended to a paper in a scientific journal (e.g. ISWC) on the resource track with you as first author.

3 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.

—

Goals of the thesis: - Whats the problem with existing solutions? with the stuff you explained in the motivation - What's the goal of the thesis in a nutshell? solve that problem / analyze the solution / compare potential solutions / ... - Then go into detail. What are the specifics of what you are going to do? Explain intermediate goals

4 Tasks Descriptions

1. describe and review the software architecture 2. deploy Basilisk and its frontend on a publicly available VM 3. benchmark 2 versions of Tentris (https://github.com/dice-group/tentris, via a github hook) and one version from another triple store (via a docker image registry, e.g. https://hub.docker.com/r/ontotext/graphdb/ or https://hub.docker.com/r/openlink/virtuoso-opensource-7). 4. fix critical bugs in 1.-3. and document non-critical

5

Schedule

—

21 weeks of work Gantt chart: pgfgantt package

6

Preliminary Outline of the Thesis

—

Bibliography