Backend/Database Co-Design for Rapid Webservice Prototyping: *The FIN Doktorandentag Talk*

Marcus Pinnecke
supervised by Gunter Saake
University of Magdeburg
pinnecke@ovgu.de

This document is located at
www.pinnecke.info/dl/docs/pinnecke-phd-proposal.pdf

January 22, 2019

Abstract

While collections of semi-structured JSON-like records (called *documents*) are commonly managed by dedicated operational database systems (called *document stores*), the nature of such data makes it non-trivial for these systems to support analytical queries on document sets efficiently. This especially holds when data is taken from third-party sources, without beforehand knowledge of the embedded structure. In this context, it is common practice to adopt operational document stores (such as MongoDB) alongside a set of third-party tools, for implementing analytic tasks. Clearly, a consequence of this practice is that a large software stack must be set up, instrumented and maintained - even if queries on the underlying data are quite simple. In our thesis, we research on the problem of *whether building an analytical document store is possible, such that (1) efficient in-memory analytics of read-mostly documents beyond the main-memory capacity is achieved, while (2) making it possible for the user to easily extend those analytics according to business needs.* Thus, we propose Protobase, as a solution. Protobase is our open source analytic NoSQL main-memory document store for prototyping micro-webservices. Our system is instantiated with user-defined code, concealed behind a user-defined REST API, and built around the philosophy of a Protobase instance (user-code + REST API + database) being a micro-webservice.

In this talk, we provide insights into our research, focusing on four cornerstone aspects required to study and evaluate Protobase as a valid solution: i) First, we take the perspective of the backbone of any database system, the storage engine. We report our findings related to the state-of-the-art in storage engine design, with a focus on flexible (tabular) data stores. We determine missing features, and outline our solution for a tabular highly-flexible data storage structure, called *GridTables*. After in-depth analysis towards the question whether tabular highly-flexible storage satisfies the requirements for a semi-structured analytic storage engine, we distill eight compute-intense open research challenges, and show the offspring of this research path. ii) Then, we take lessons learned from our study on storage engines and propose the Cabin file format for columnar binary-encoded storage of JSON-like documents. We show the challenges and engineering solutions behind Cabin, to enable in-memory processing of document data exceeding the main-memory capacity. Namely, we argue that the entire Microsoft Academic Graph (MAG) - a dataset on publication meta data with more than 200GB of JSON files - *fits* into less than 1GB of main-memory if we apply our string caching strategy. iii) Third, we give a detailed requirement analysis and assumptions on the runtime environment of Protobase, and provide insights into its architecture and storage engine design for building a Cabin database. iv) For a solid baseline evaluation of Protobase beyond micro-benchmarking, we suggest an extension of *YCSB*, the popular benchmark for operational systems. In detail, we suggest an extension of *YCSB* (called ycsbX) with analytic queries and workloads, that ultimately aims to extend *YCSB* to be a benchmark for database systems on dynamic hybrid (i.e., operational and analytical) workloads. We show our results on MemSQL, Postgres and Redis to highlight novel insights obtainable from ycsbX. In addition, we describe our plans on how to use ycsbX to evaluate Protobase against MongoDB and alternatives as baseline.

To conclude our talk we summarize our preliminary results, indicating that Cabin is a good fit for in-memory analytics on third-party datasets, that using Cabin as foundation for a database works, and - standing-in for the ongoing final evaluation - we give an informed argumentation on why Protobase with its philosophy is a reasonably good solution to the problem addressed in our thesis.