GNU Taler Scalability

Degree programme : BSc in Computer Science | Specialisation : IT Security
Thesis advisor : Prof. Dr. Christian Grothoff

Currently, central banks all over the globe are investigating possible implementations of a Central Bank Digital Currency (CBDC). Unfortunately, most of today‘s electronic payment systems do either not offer adequate technical privacy assurances to citizens, or are too slow to handle the expected transaction load.

Introduction

According to a personal discussion with Giesecke+Devrient, a payment system that is to support 500 million people (EU) for all payments should be able to handle about 100‘000 transactions per second (TPS). This would be sufficient to handle the combined rate of **all** currently used means of payment, such as credit cards, bank transactions and cash. Naturally, a smaller economy might work well on a much lower transaction rate. For example, the same per-capita use would imply a need of 2‘000 TPS for all of Switzerland.

One of the goals of the GNU Taler project is to become a free software reference implementation of a privacy preserving retail CBDC. The goal of this work was to evaluate and improve the performance of the GNU Taler implementation to assess its scalability.

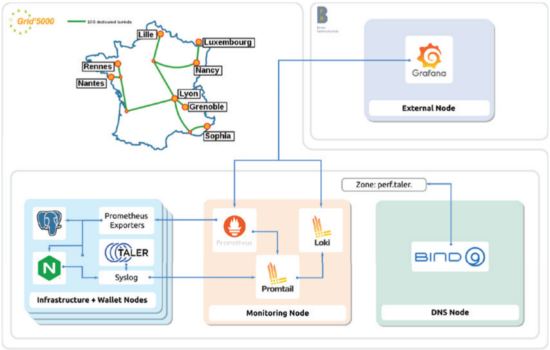
To do this, we created the first distributed GNU Taler deployment using multiple machines on Grid‘5000, a large scale a distributed testbed providing computing and storage resources across France and Luxembourg.

Results

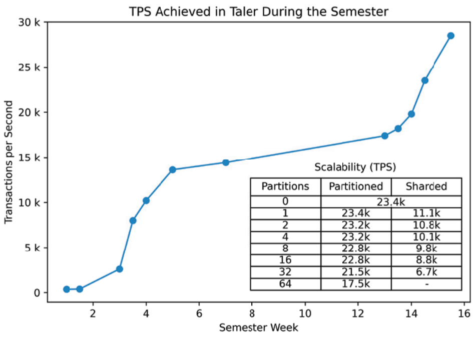
The final result is a dynamic and reproducible experiment setup which can be used for further assessments. This includes data extraction and visualization with a Prometheus, Loki and Grafana stack. Our setup includes the various Taler components, and experiments were performed using a non-uniform load distribution between the merchant accounts for increased realism.

We identified and fixed several bottlenecks in the GNU Taler software, parallelized the execution of the cryptographic frontend so that the PostgreSQL database was left as the natural bottleneck. Here we optimized the queries and modified the schema to allow partitioning of tables. We showed that Taler scales well in a distributed environment, including horizontal distribution of the PostgreSQL database. By making improvements to the database queries, as well as to the source code, we were able to increase performance by a factor of 95, from 300 up to 28.5k TPS. Compared to Visa‘s average load of 1‘667 TPS, this is a respectable result.

Marco Boss



Experiment System Architecture



Scalability (TPS)

Partitions	Partitioned	Sharded
0	23.4k	
1	23.4k	11.1k
2	23.2k	10.8k
4	23.2k	10.1k
8	22.8k	9.8k
16	22.8k	8.8k
32	21.5k	6.7k
64	17.5k	-

TPS achieved on the Grid’5000