University of Bamberg

YeSQL: “You extend SQL”

Jinghao Wu | DT-DB42-M | 28.06.2023

Contents

University of Bamberg

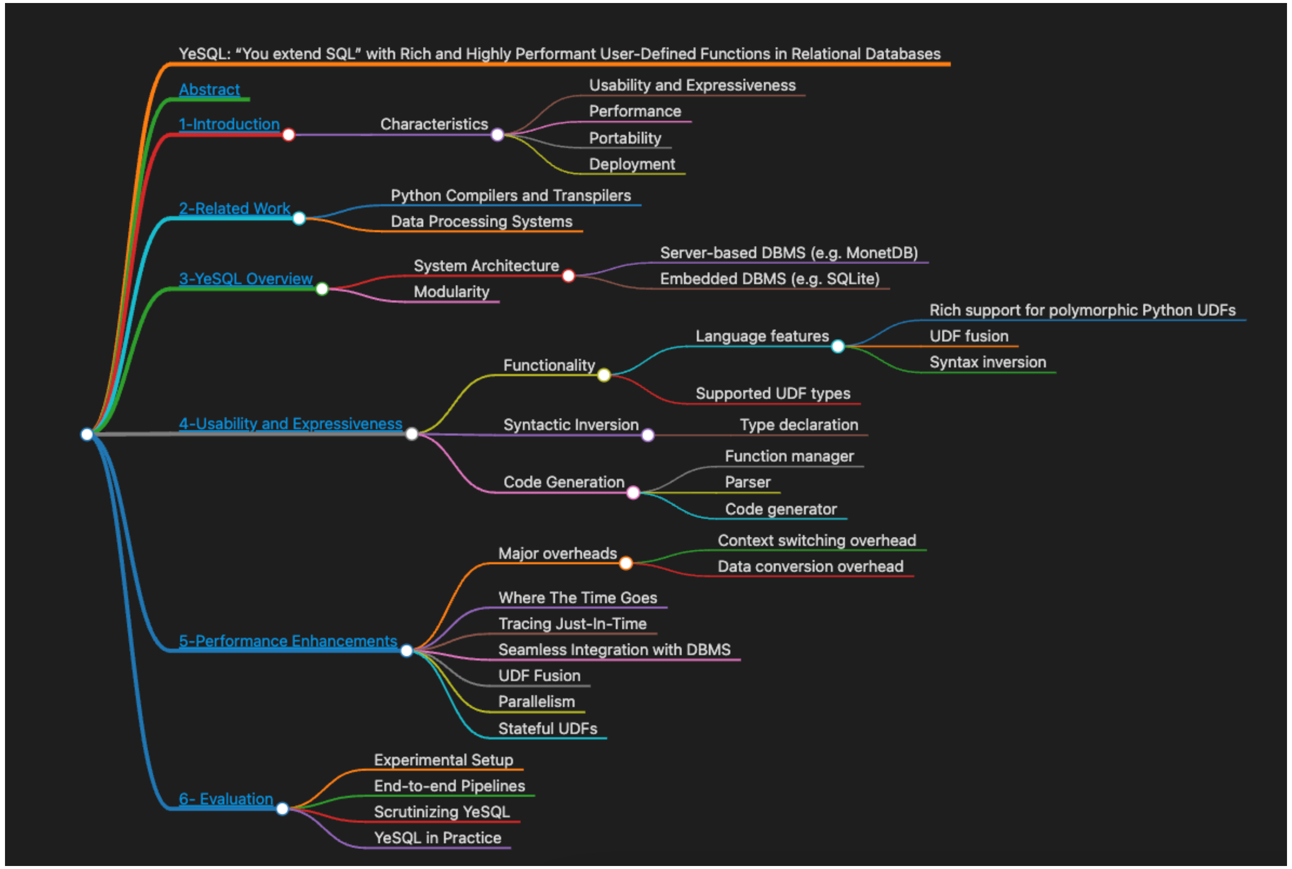
What is YeSQL

Who & Why YeSQL

How does YeSQL perform (Evaluation)

Conclusion and Future

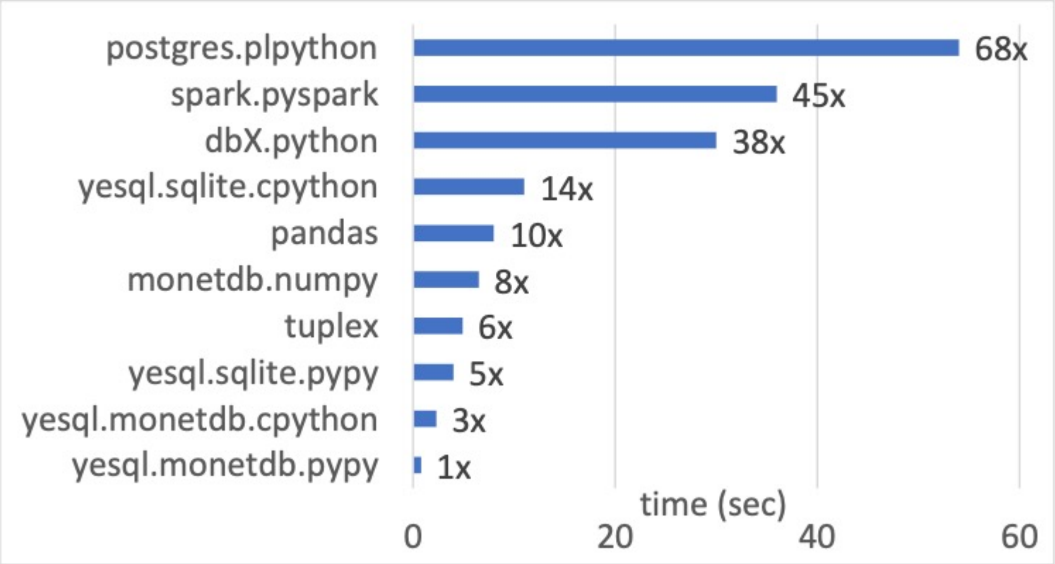
References



What is YeSQL

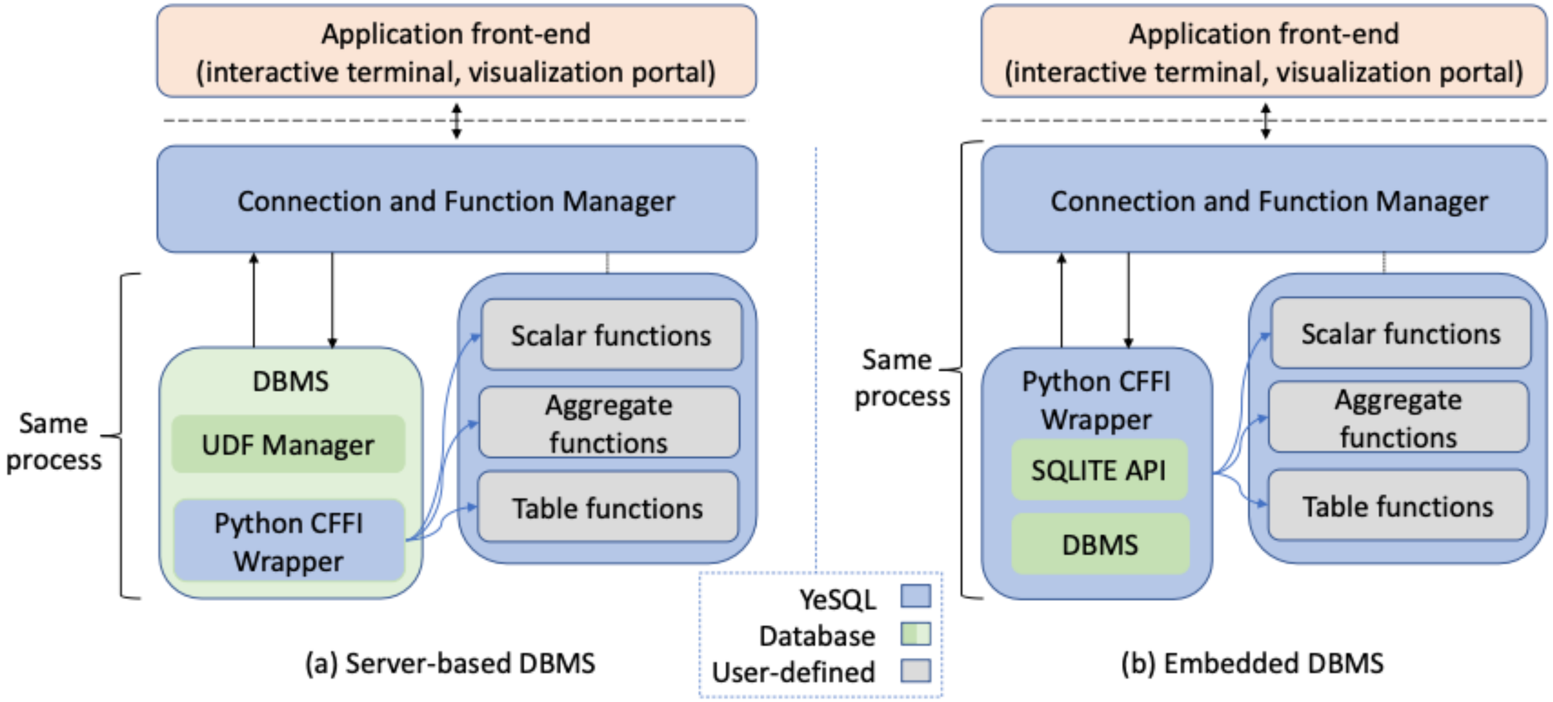
YeSQL is an SQL extension and implementation that enhances the **usability, expressiveness**, and **performance** of Python UDFs in data processing systems.

It provides a more compact and expressive syntax for relational queries and complex compositions of UDFs and relational functions.



System Architecture

YeSQL supports both **Server-based** and **Embedded DBMS** for different application scenarios.



(a) Server-based DBMS

(b) Embedded DBMS

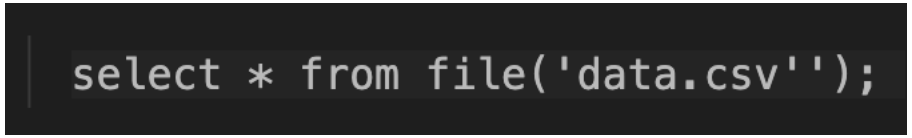
Two user roles:
Application users (data scientists etc.)
UDF developers

Consists of (in Python)
(a) Function manager
(b) Parser
(c) Code generator

Supported UDF types

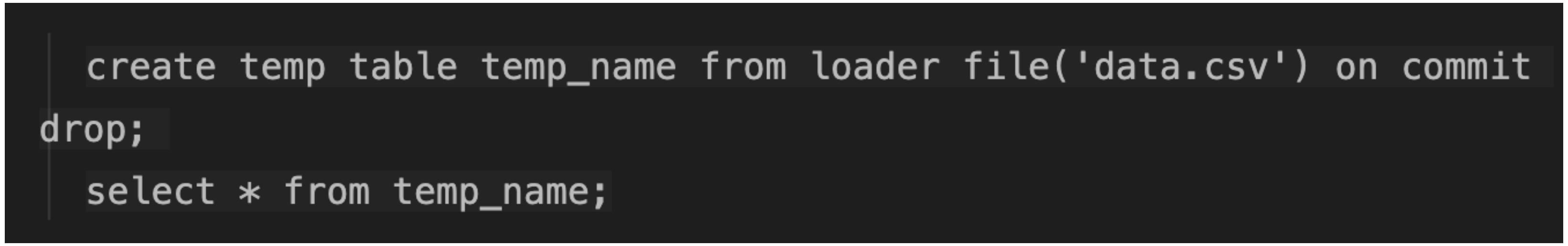
Code Generator

Example polymorphic UDF



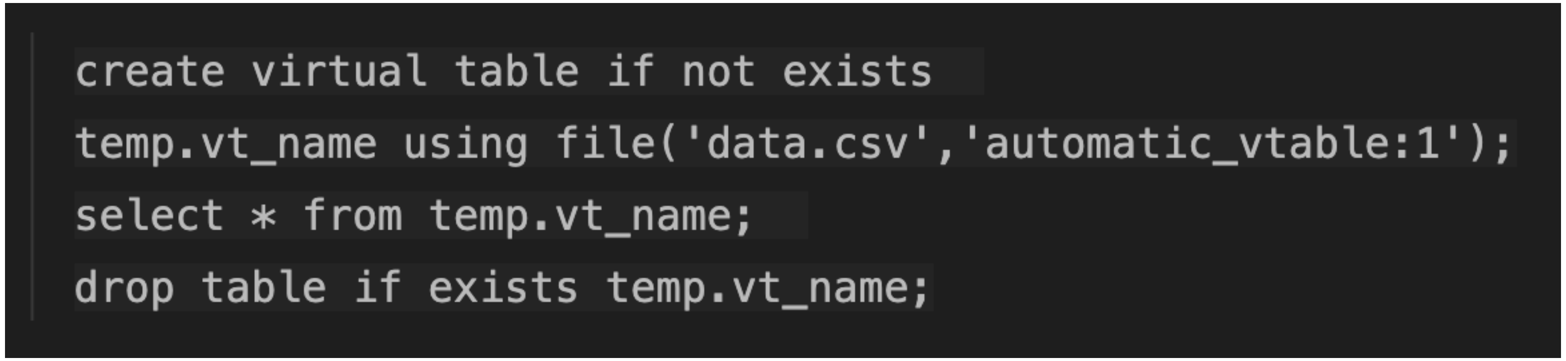
```
select * from file('data.csv'');
```

MonetDB



```
 create temp table temp_name from loader file('data.csv') on commit
drop;
 select * from temp_name;
```

SQLITE API



```
create virtual table if not exists
temp.vt_name using file('data.csv','automatic_vtable:1');
select * from temp.vt_name;
drop table if exists temp.vt_name;
```

Who & Why YeSQL (Characteristics)

Usability and Expressiveness

- Small test and Syntax Inversion

Performance (more details in next slide)

- Five techniques

Portability and (Modularity)

- Synergy

Deployment

- OpenAIRE
- Several domains

380 undergraduate student

two algorithms with/out YeSQL

328 completed successfully the task (86.3%)

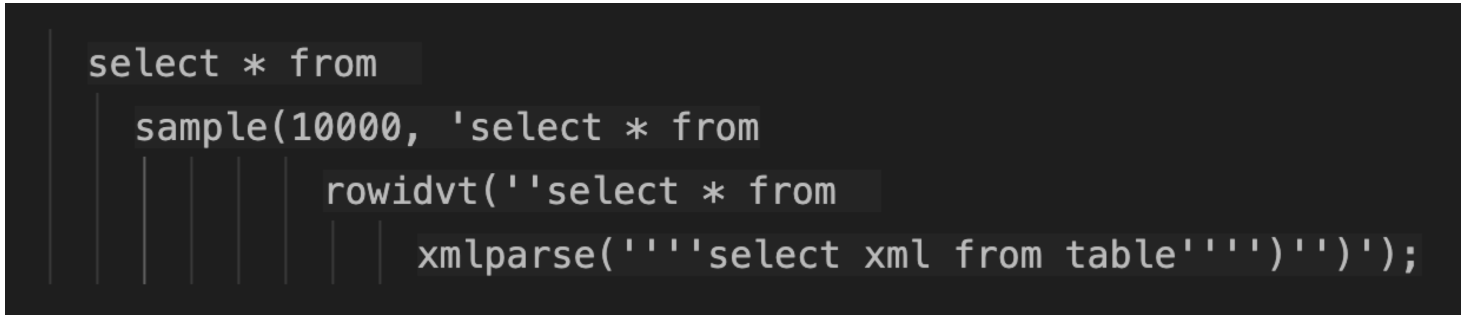
OpenAIRE

A technical infrastructure co-designed and co-developed in the context of a consortium of **65** European universities, research centers, and other institutions, offering services that were invoked **42M** times last year in the context of **1M** visits.

Syntax Inversion

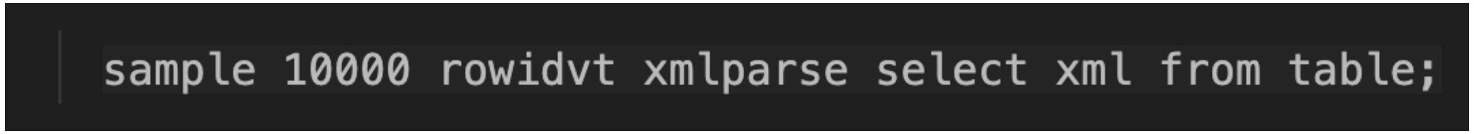
University of Bamberg

PostgreSQL query



```
select * from
  sample(10000, 'select * from
          rowidvt(''select * from
              xmlparse(''''select xml from table'''')'')');
```

YeSQL



```
sample 10000 rowidvt xmlparse select xml from table;
```

Two major Overheads

Context switching

Data conversion

Five Techniques

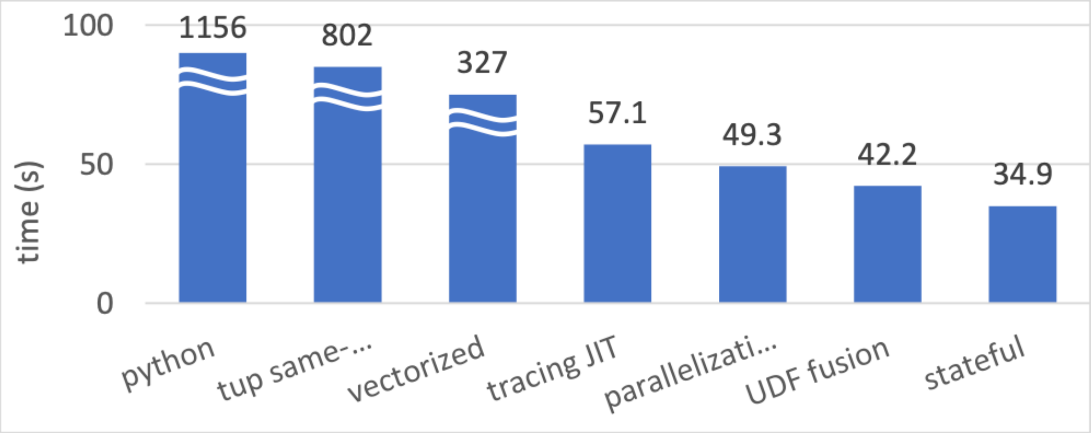
Tracing Just-In-Time (JIT) Compilation

Seamless Integration with DBMS

UDF Fusion

Parallelism

Stateful UDFs



Evaluation - Setup

Ubuntu 20.04, Intel Core (Ivy Bridge E) i7-4930K processor, 3.40GHz (6 cores), 64GB MM

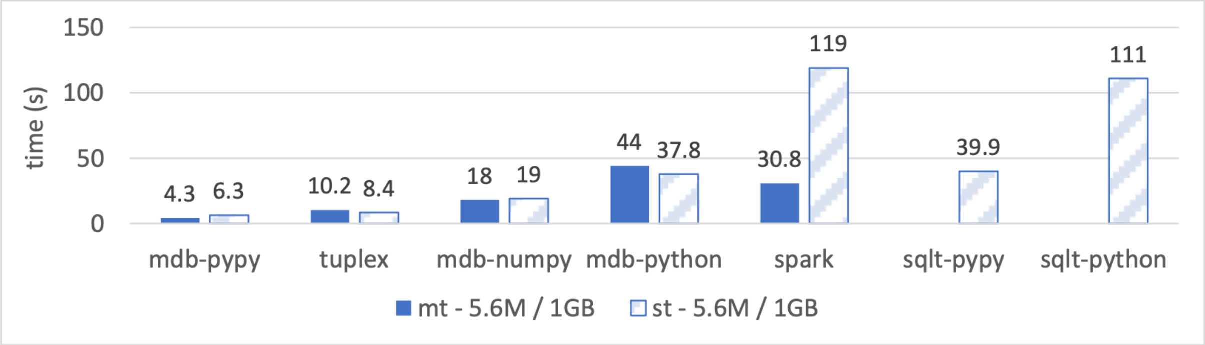
YeSQL against Tuplex, MonetDB, PostgreSQL, dbX, Pandas and Spark

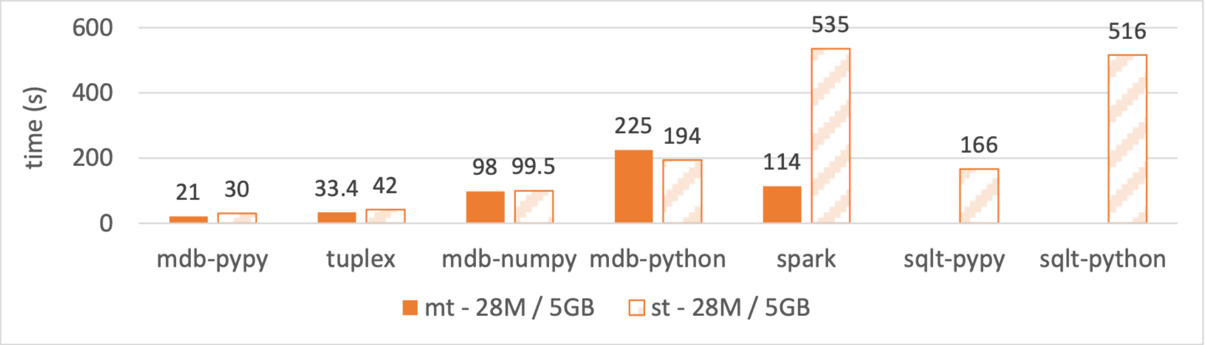
Five runs per test, results were averaged

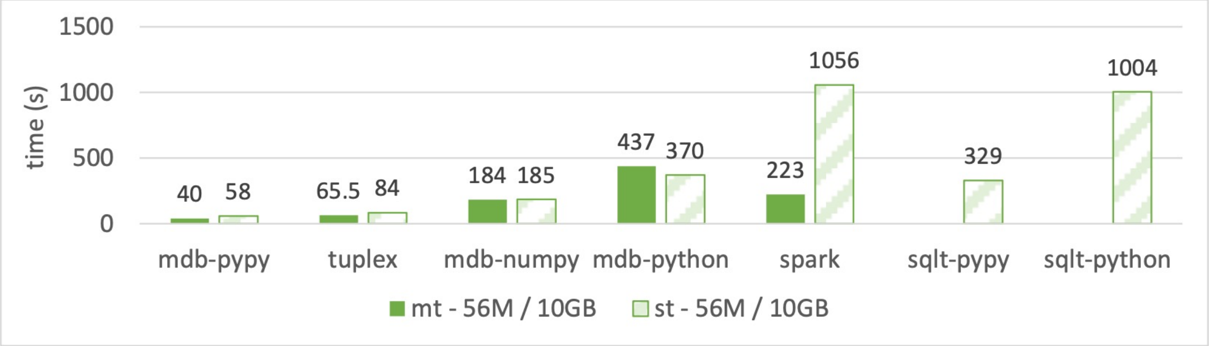
3 datasets: Zillow, Flights and Text mining

Evaluation - Zillow

University of Bamberg



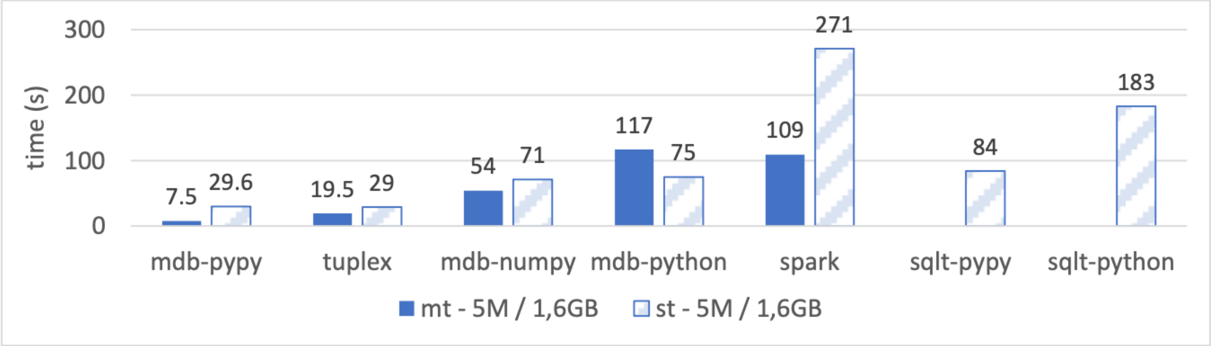


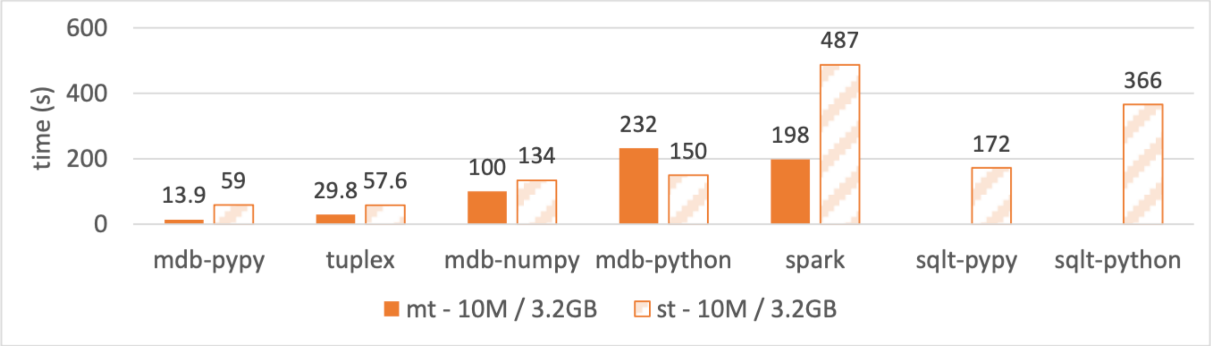


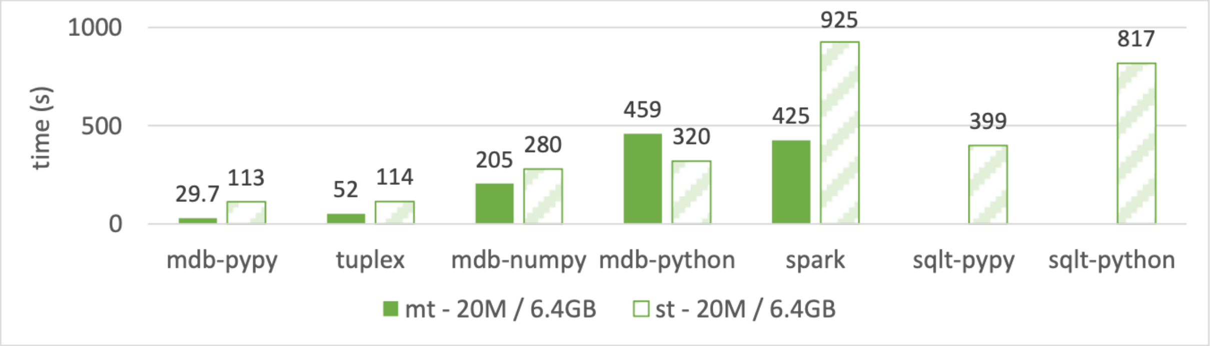
Variables: Data sizes and Degree of parallelism

mdb-pypy: YeSQL with tracing JIT on MonetDB

Evaluation - Flights



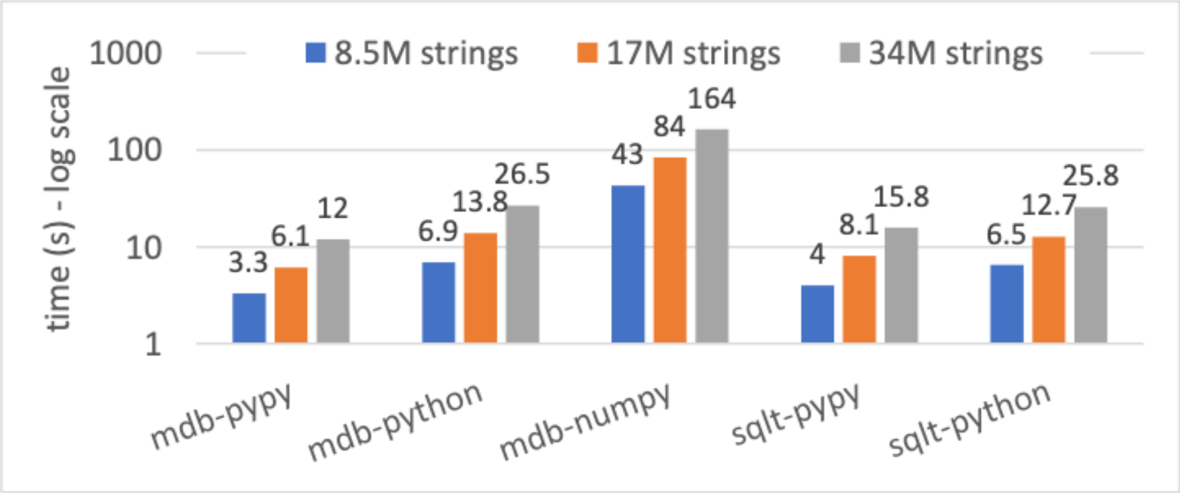




Variables: Data sizes and Degree of parallelism.

This dataset differs from zillow, as it contains many columns (110), two small tables used in joins, and more operators, 23 operators on the larger table, 3 joins, and 1 filter.

Evaluation – Text Mining

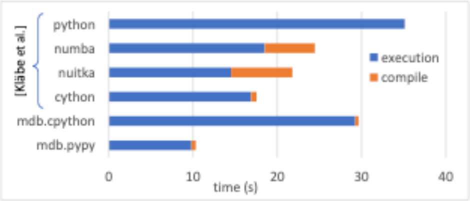


Variables: Data sizes and with various implementations (CPython, Numpy, and PyPy)

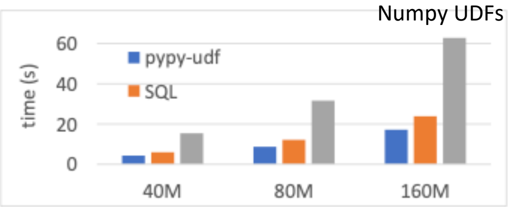
on a server-based engine(MonetDB), and an embedded one. (SQLite)

Several Micro-experiments - Flights

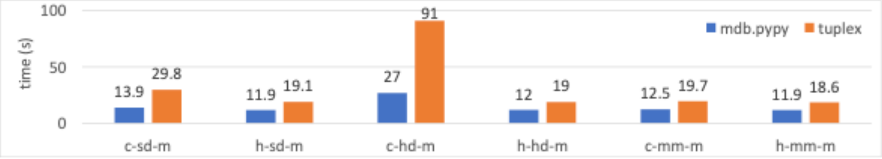
University of Bamberg



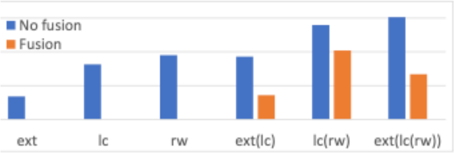
(a) Comparison of a single UDF



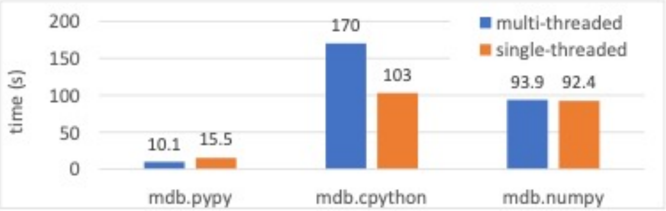
(b) Seamless integration



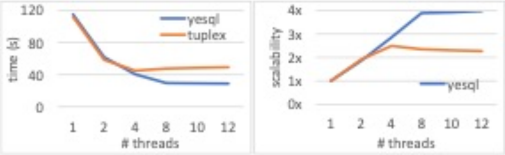
(c) Cold cache vs. Hot cache on SDD/HDD/tmpfs (multi-threaded)



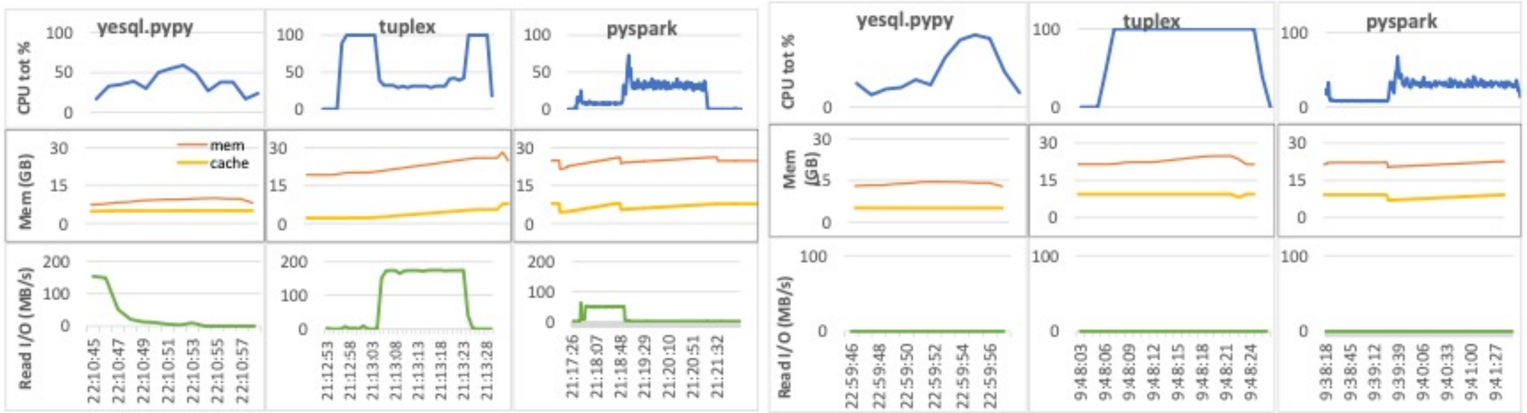
UDF fusion



(a) Multi-threaded vs. single-threaded



(b) Scaling degree of parallelism



(c) Resource usage: YeSQL, Tuplex, PySpark with cold and hot caches

Conclusion

- An SQL extension with rich UDF support along with a pluggable architecture to easily integrate it with either server-based or embedded database engines.
- Supports Python UDFs fully integrated with relational queries as scalar, aggregator, or table functions.
- Outperforms alternative implementations due to its several performance enhancements, including tracing JIT compilation of Python UDFs, parallelism and fusion of UDFs, stateful UDFs, and seamless integration with a database engine.
- **Future work:** Extensions to federated, heterogeneous systems or optimization opportunities in UDF fusion and query rewriting.

> *“YeSQL is a **significant step forward** in the direction of enhancing the usability and improving the performance of user-defined functionality inside DBMSs. ”*

References

1. YeSQL: “You extend SQL” with Rich and Highly Performant User-Defined Functions in Relational Databases
https://www.vldb.org/pvldb/vol15/p2270-foufoulas.pdf

2. YeSQL: Rich User-Defined Functions without the Overhead
https://www.vldb.org/pvldb/vol15/p3730-foufoulas.pdf

Website: https://www.openaire.eu

Thank you for your attention!

Jinghao Wu | DT-DB42-M | 28.06.2023

Related Work From Our Professor

Schüle et al. [63] extends the PostgreSQL grammar to allow lambda expressions and subqueries as table function's arguments and presents a modification of PostgreSQL's JIT compiler framework to inline lambda expressions in table functions.

——Maximilian E. Schüle, Jakob Huber, Alfons Kemper, and Thomas Neumann. 2020.

Freedom for the SQL-Lambda: Just-in-Time-Compiling User-Injected Functions in PostgreSQL.