YeSQL: "You extend SQL" with Rich and Highly Performant User-Defined Functions in Relational Databases

Y. Foufoulas[1,2], A. Simitsis[1], L. Stamatogiannakis[2], Y. Ioannidis[1,2]

[1]Athena Research Center [2]University of Athens

Motivation

- Modern data pipelines involve complex processing tasks
  ↓ Very complicated ecosystem to execute pipelines
- DBs process and manage efficiently large data volumes
  ↓ SQL provides limited expressive power
- UDFs to the rescue
  ↓ Impedance mismatch between their relational (SQL) evaluation and procedural (Python) execution

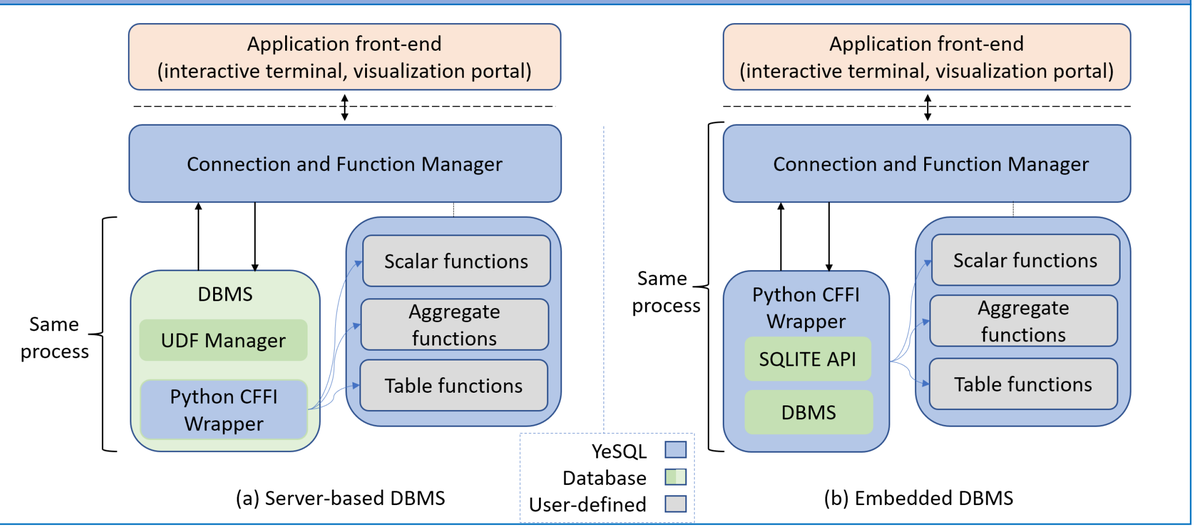
State of the Art

- Python compilers / transpilers
  ↓ Limitations: library support, performance, etc.
- Python UDFs in DBs
  ↓ Limitations: usability, expressiveness, performance
- Python in data engines (UDFs to SQL, UDFs to IR, Tuplex)
  ↓ Lack expressiveness features, support for specific libraries, expensive translation mappings

Our Approach: YeSQL

- Usability and expressiveness
  ↑ Stateful, parametric, polymorphic, dynamically typed, scalar/aggregate/table UDFs
- Performance enhancements
  ↑ Seamless data exchange between UDFs and DB
  ↑ JIT-compiled UDFs and stateful UDFs
  ↑ UDF parallelization and UDF fusion

Architecture



Example

Python UDF

```
@scalar_udf
def remove_punc(text: str): -> str
   return " ".join(["".join([ch for ch in word if ch.isalnum()])
         for word in text.split(' ')]) if text is not None else None
```

what the UDF dev implements

PyPy / CFFI code [db agnostic]

```
/* remove_punc.h  */
extern void remove_punc_wrap(char **input, int count, char **result);
```

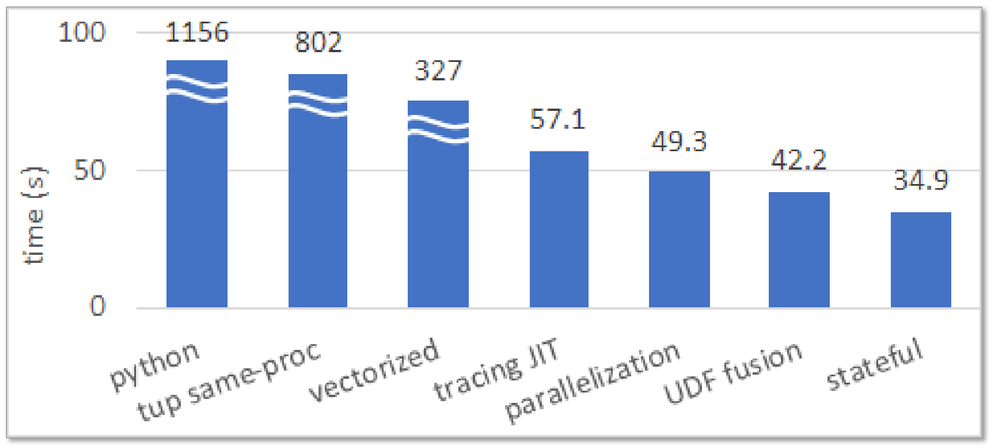
```
import udf_funcs as uf
@ffi.def_extern()
def remove_punc_wrap(input, count, result:)
   for i in range(count):
      _tmp = ffi.string(input[i]).decode()
      _res = uf.remove_punc(_tmp).encode()
      result[i] = lib.strdup(ffi.from_buffer(memoryview(_res))
```

MonetDB C-UDF [db specific]

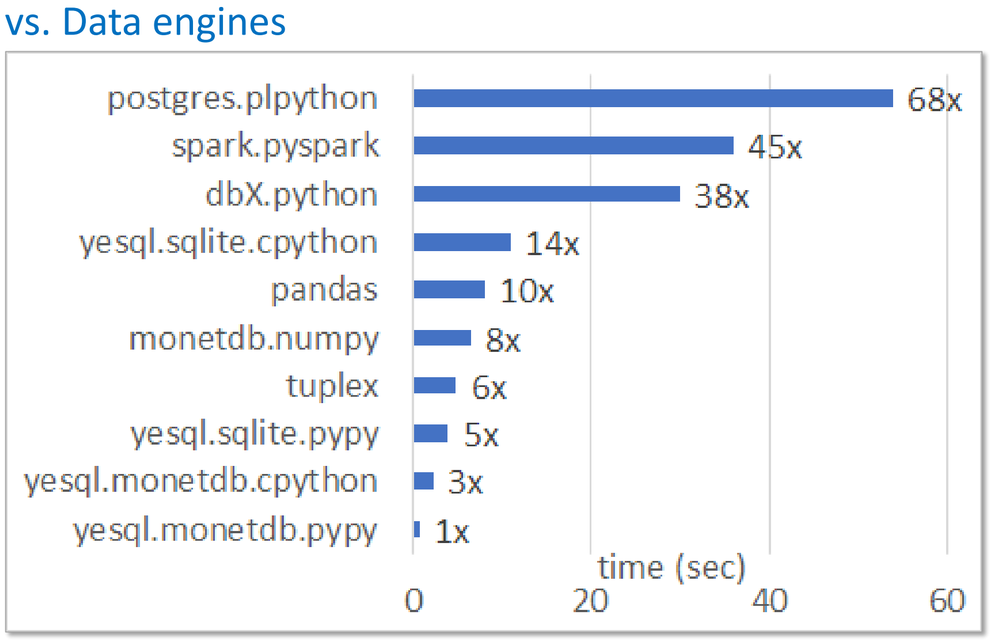
```
CREATE OR REPLACE FUNCTION remove_punc(input STRING)
RETURNS STRING
LANGUAGE C {
#pragma CFLAGS -I<path>
#pragma LDFLAGS -L<path> -l<lib>
#include "remove_punc.h"
  result->initialize(result, input.count);
  remove_punc_wrap(input.data, input.count, result->data);
};
```

Evaluation Results

Where does the time go?



YeSQL vs. Data engines



Info

- **Y. Foufoulas, A. Simitsis, L. Stamatogiannakis, Y. Ioannidis. Y. YeSQL: "You extend SQL" with Rich and Highly Performant User-Defined Functions in Relational Databases - PVLDB, 2022**
- **Y. Foufoulas, A. Simitsis, Y. Ioannidis. YeSQL: Rich User-Defined Functions without the Overhead - PVLDB, 2022 (demo)**