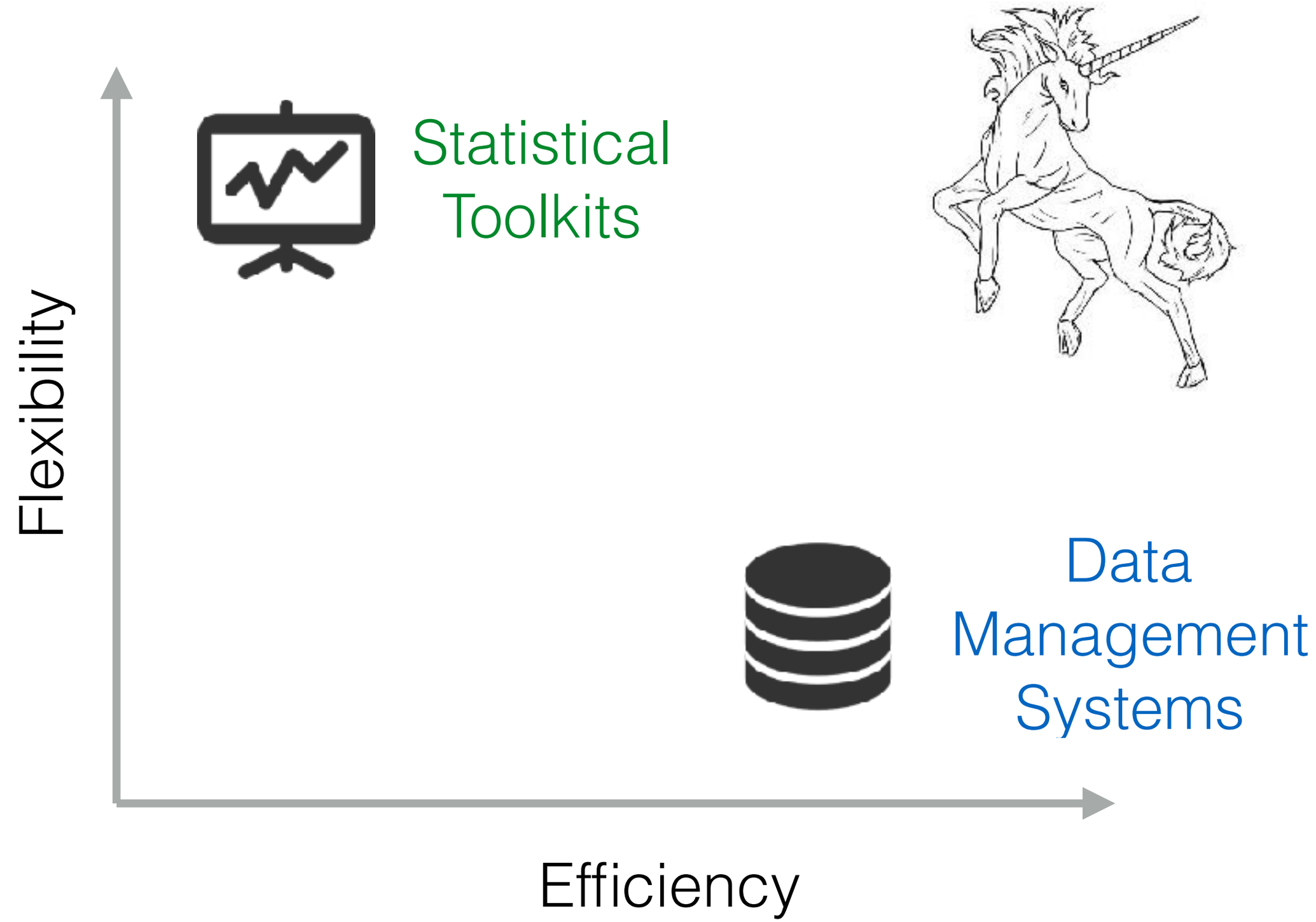
CWI
Centrum Wiskunde & Informatica

VU UNIVERSITY AMSTERDAM

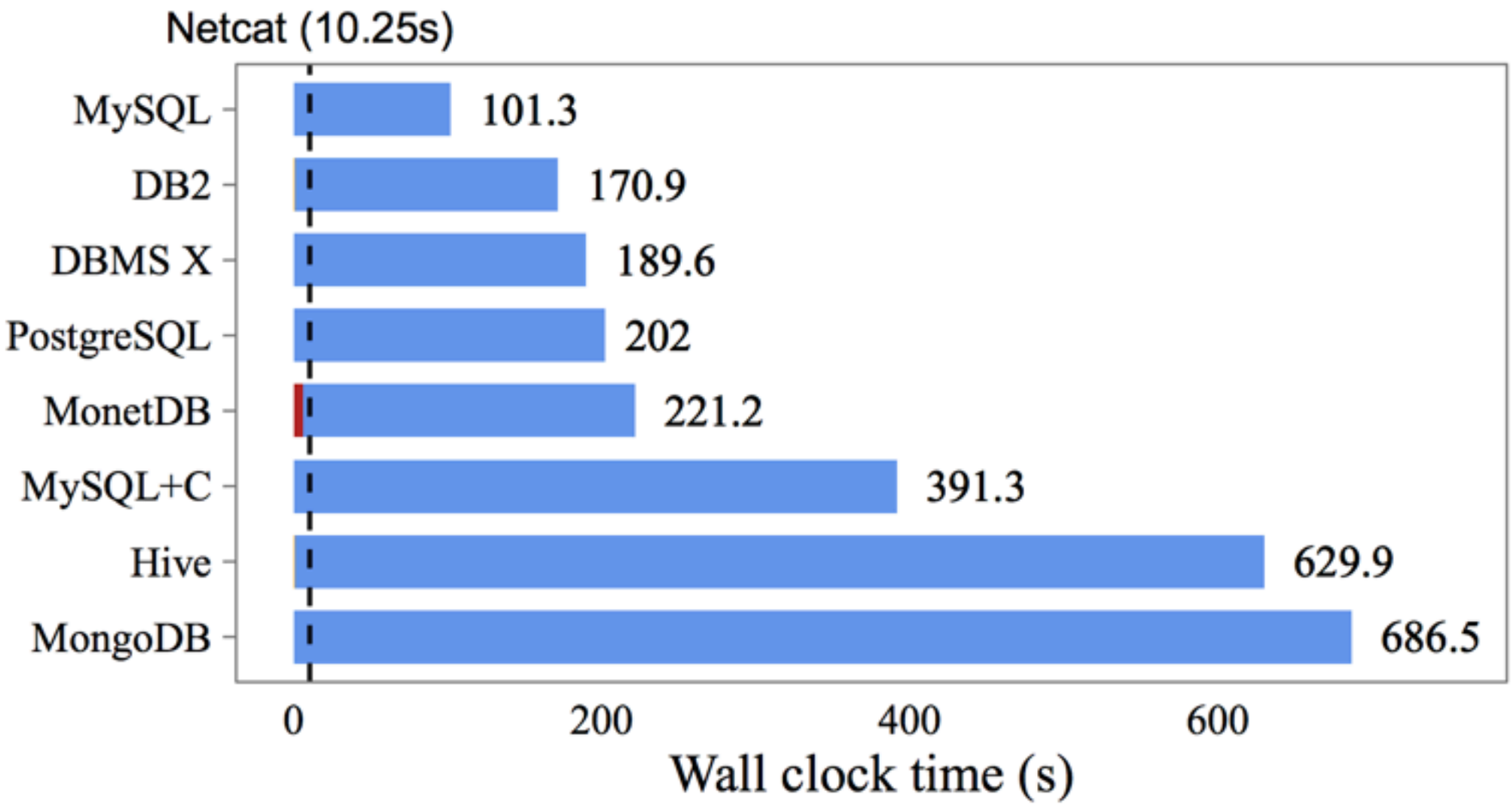
Chimeras are useful - Embedding scripting languages in data management

Hannes Mühleisen

Integrate not Reinvent



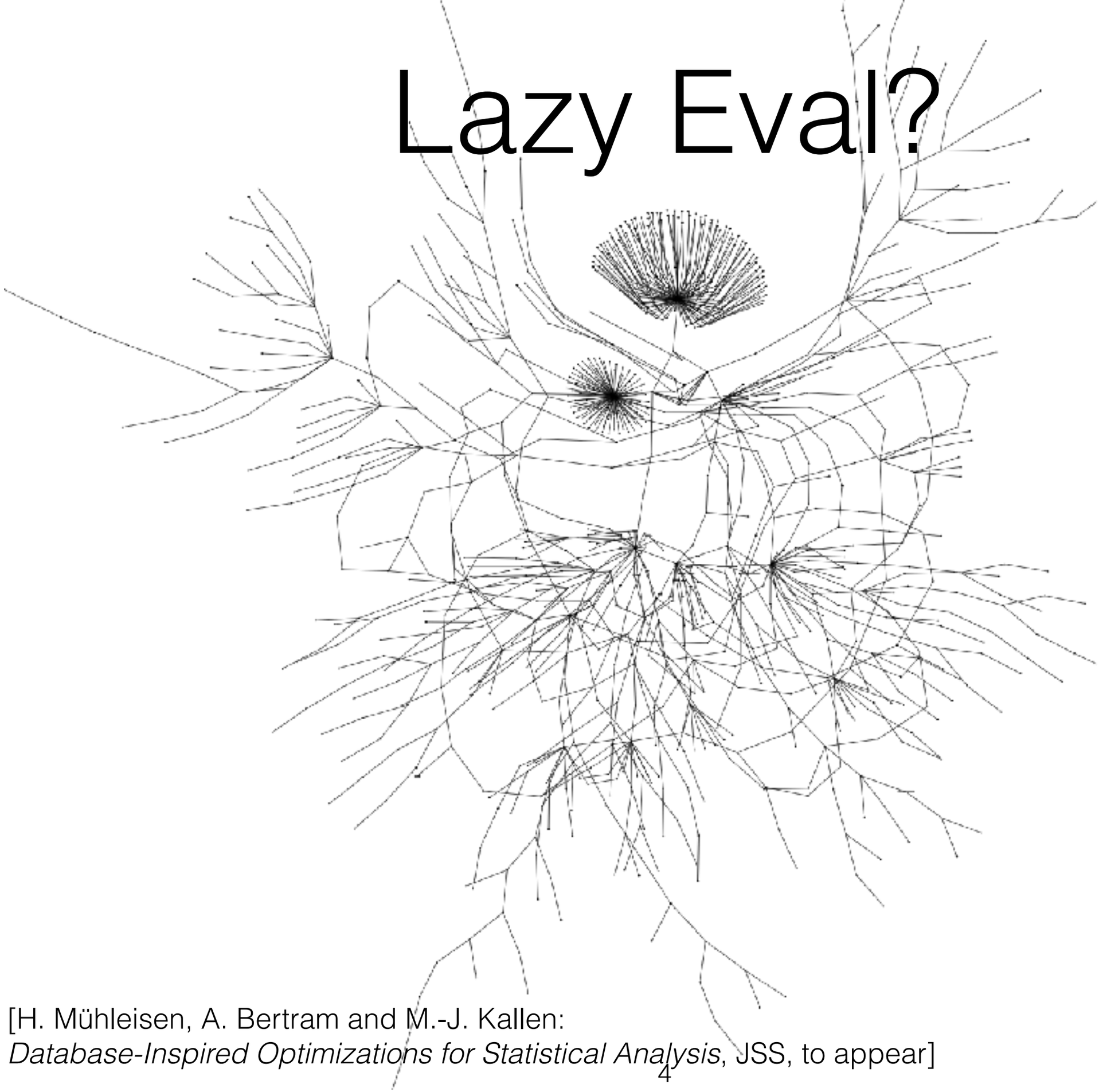
Client protocols?




```
SELECT * FROM lineitem_sf10;
```

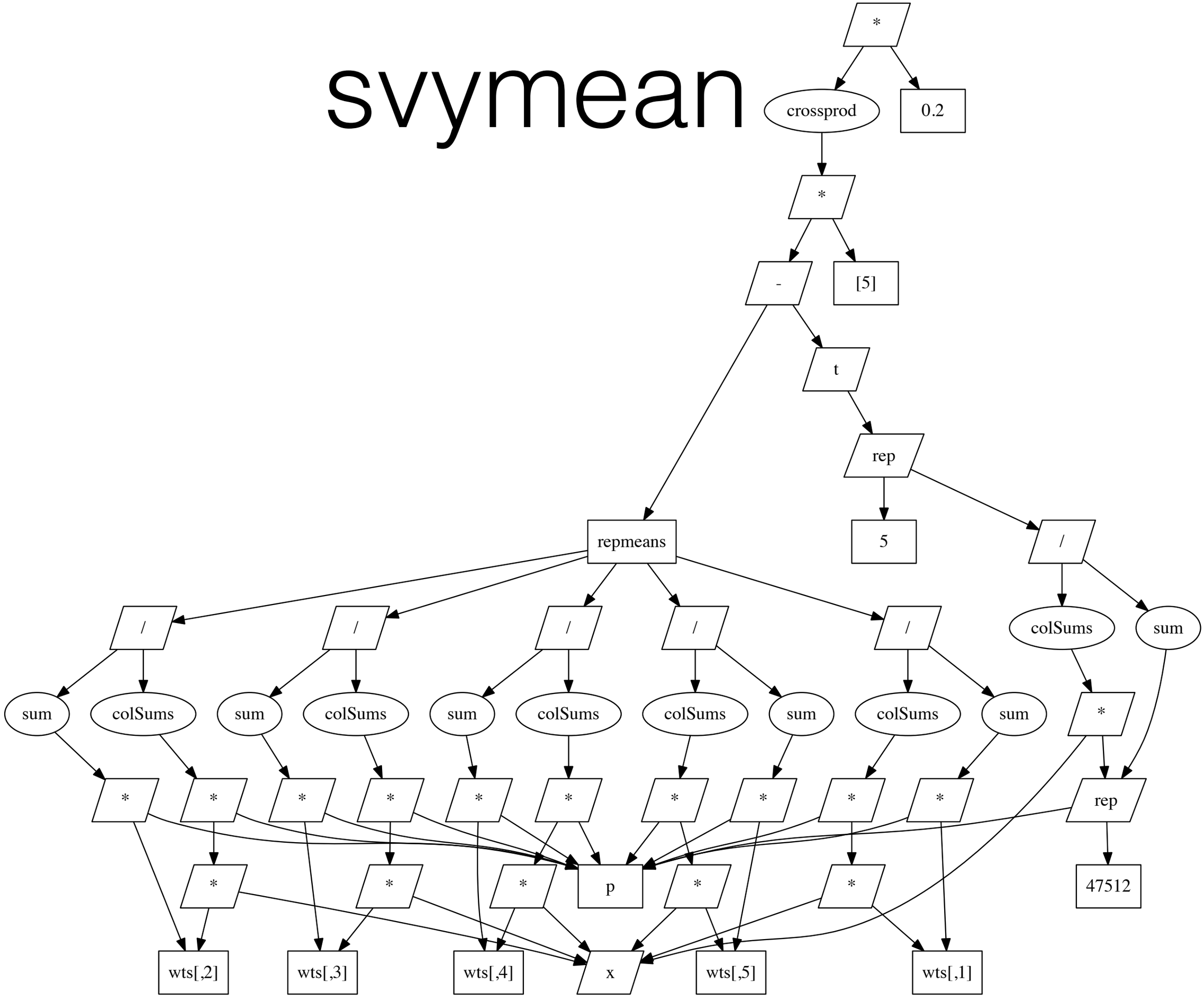
[M. Raasveldt & H. Mühleisen:
Don't Hold My Data Hostage - A Case For Client Protocol Redesign, VLDB 2017]

Lazy Eval?

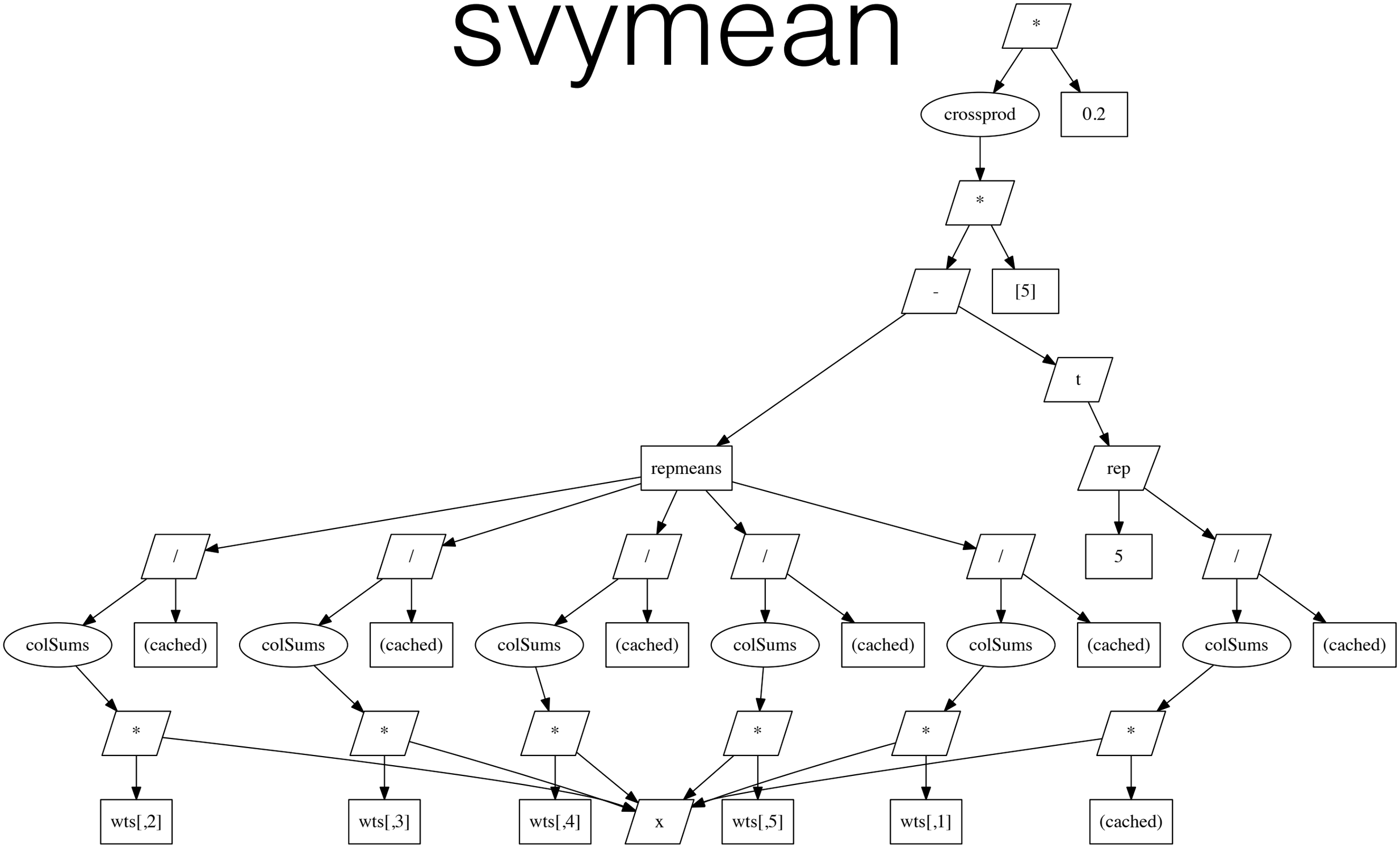


[H. Mühleisen, A. Bertram and M.-J. Kallen:
Database-Inspired Optimizations for Statistical Analysis, JSS, to appear]

svymean


*
crossprod
0.2
*
-
[5]
t
rep
repmeans
5
/
/
/
/
/
/
colSums
sum
sum
colSums
sum
colSums
sum
colSums
colSums
sum
colSums
sum
*
*
*
*
*
*
*
*
*
*
*
*
rep
*
*
*
p
*
*
47512
wts[,2]
wts[,3]
wts[,4]
x
wts[,5]
wts[,1]

svymean



Maybe…

Lower friction…

R/Python UDFs in DB

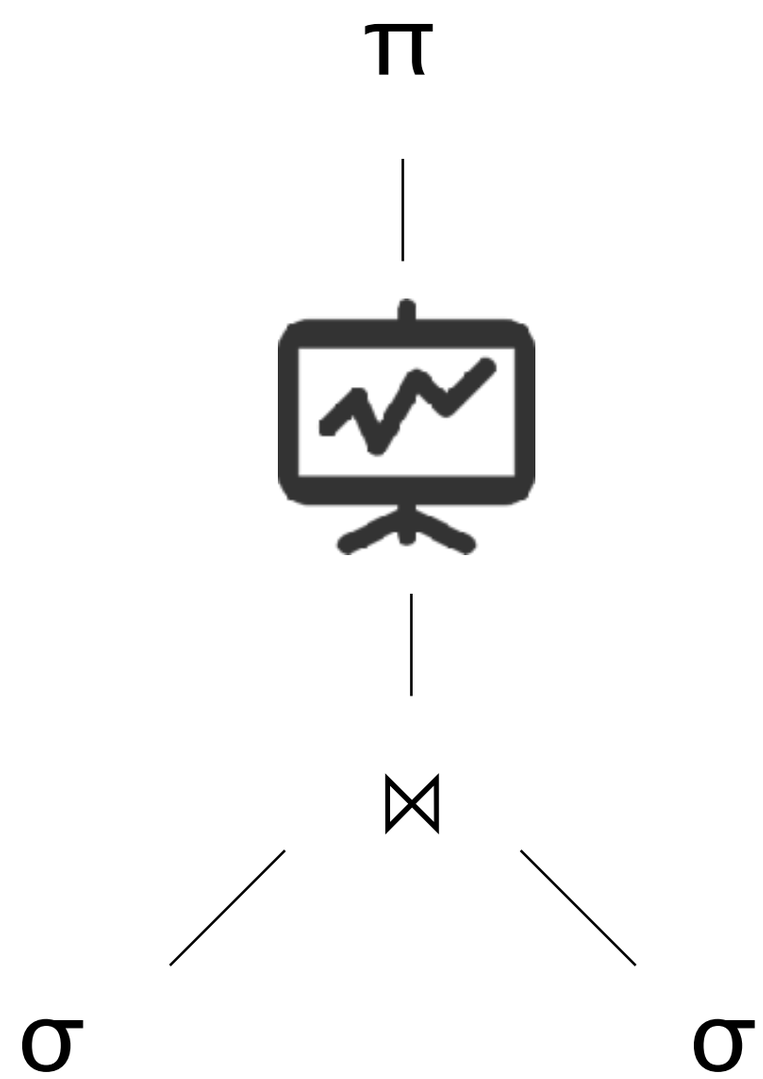
Embedded R in MonetDB

Part of MonetDB since 2014

Embedded NumPy in MonetDB

Released 2016

Relationally Integrated



Statistical analysis as operators in relational queries

σ e.g. Filtering

```
CREATE FUNCTION rapi03(i INTEGER, z INTEGER)
RETURNS BOOLEAN LANGUAGE R { i>z };

SELECT i FROM rval WHERE rapi03(i, 2);
```

Vectorized! Parallel?

DB in C/R/NumPy/Java/…

MonetDBLite for R
Released 2015

MonetDBLite for Python
Released 2017

MonetDBLite for Java[R ™ C]
Preview

What is MonetDBLite

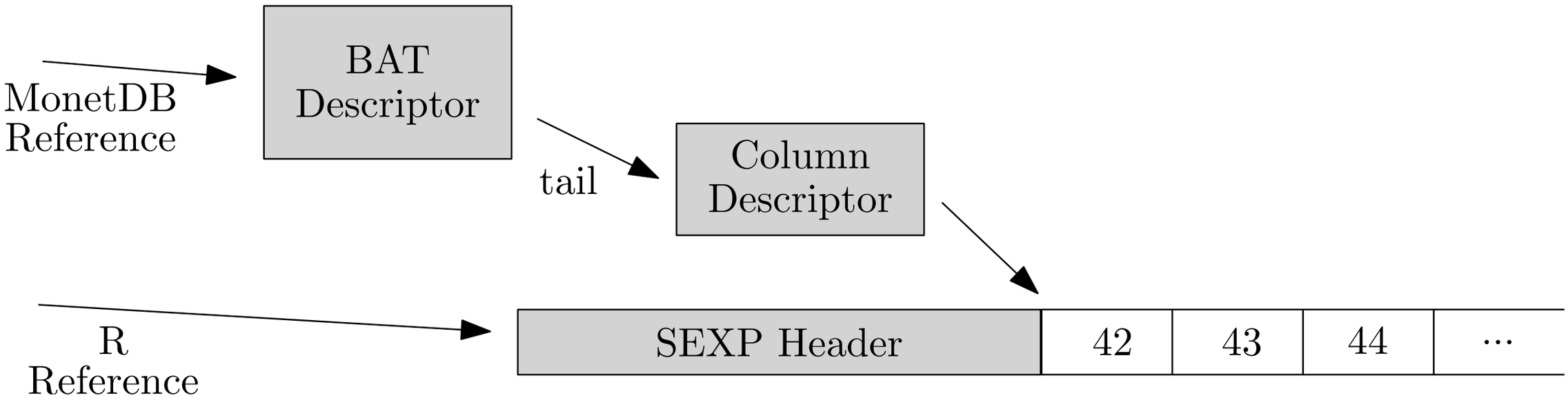
- Embedded & streamlined MonetDB for other languages
- All of DB: Transactions, complex SQL etc.
- In-Process operation
- Query results are data frames
- Fast data append from data frames
- Zero-Copy (Beta)

Zero-Copy

[J. Lajus and H. Mühleisen:
Efficient Data Management and Statistics with Zero-Copy Integration, SSDBM2014]

[M. Raasveldt and H. Mühleisen:
Vectorized UDFs in Column-Stores, SSDBM2016]

Dress-up



+ mmap / Garbage Collection Fun

Design Challenges

- Zero copy vs. Data Integrity
  - R: shared markers
  - NumPy: readonly flag
  - Native code: No guarantees :(
- Garbage Collector hacks
  - who calls `free()`?

Design Challenges II

- Systems/languages never designed to run embedded
  - Fatal errors `exit()`
  - Relative paths relying on `setwd()`
  - Symbol clashes, e.g. `error()` etc.
  - Global variables galore (GIL, restartability?)
  - `stdout`/`stderr` & signal hijacking

Conclusions?

- Clean (re-)implementations won’t happen. Trying to make sense of user scripts rarely works.
- People don't use R/Python (the languages), people use the ecosystem, which we can’t optimize
- We need in-process integration for speed
- Design for embedding!
- Nicer architecture “for free”

http://hannes.muehleisen.org
@hfmuehleisen