CWI

Centrum Wiskunde & Informatica

Efficient tabular data ingestion and manipulation with MonetDBLite

Hannes Mühleisen*

useR!2016 2016-05-29

Thanks previous Speakers

- Javier for DBI Intro

- Ben for proposing persistent SQL DBs for medium data

Running a DB is hard

- Installation / Automatic startup difficult
- Configuration for workload often crucial
  - `checkpoint_completion_target`?
  - `effective_cache_size`?
- Maintenance, updates, … workload increase

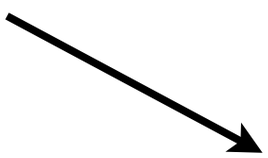
⟶ Most people don’t bother if not forced

What about  ?

- In-process SQL database, data either in memory or in a file, rock-solid, used on every smartphone, browser, OS, ….
- People also use it for large-ish dataset analysis
- Bad idea, SQLite was never built for this
  - e.g. row-based storage model

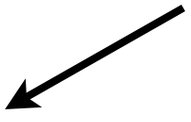
Postgres, MySQL, etc.:

Conceptional



class	speed	flux
NX	1	3
Constitution	1	8
Galaxy	1	3
Defiant	1	6
Intrepid	1	1

Physical (on Disk)



NX		1	3	Constitution		1	8	Galaxy	
1	3	Defiant		1	6	Intrepid		1	1

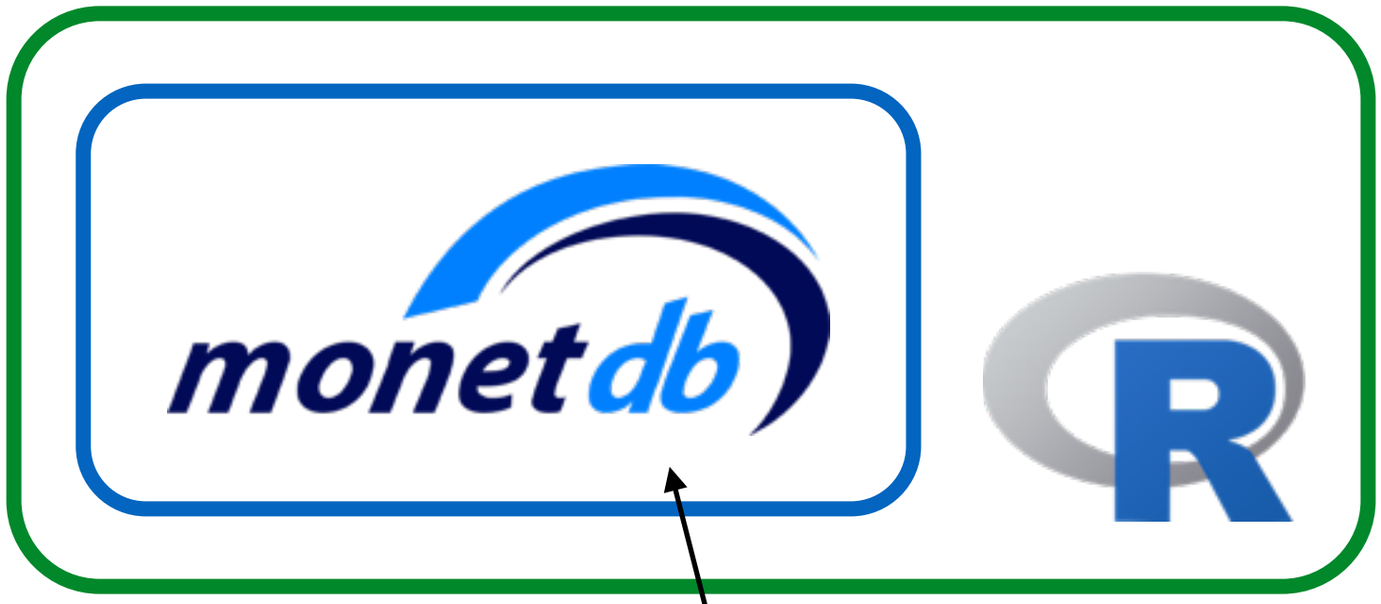
Column Store:

class	speed	flux
NX	1	3
Constitution	1	8
Galaxy	1	3
Defiant	1	6
Intrepid	1	1

NX	Constitution	Galaxy	Defiant	Intrepid
1	1	1	1	1
3	8	3	6	1

Peter A. Boncz, Martin L. Kersten, and Stefan Manegold. 2008. Breaking the memory wall in MonetDB. *Communications of the ACM* 51, 12 (December 2008), 77-85. DOI=10.1145/1409360.1409380

Enter MonetDBLite



2016 SIGMOD systems award winner

What is MonetDBLite

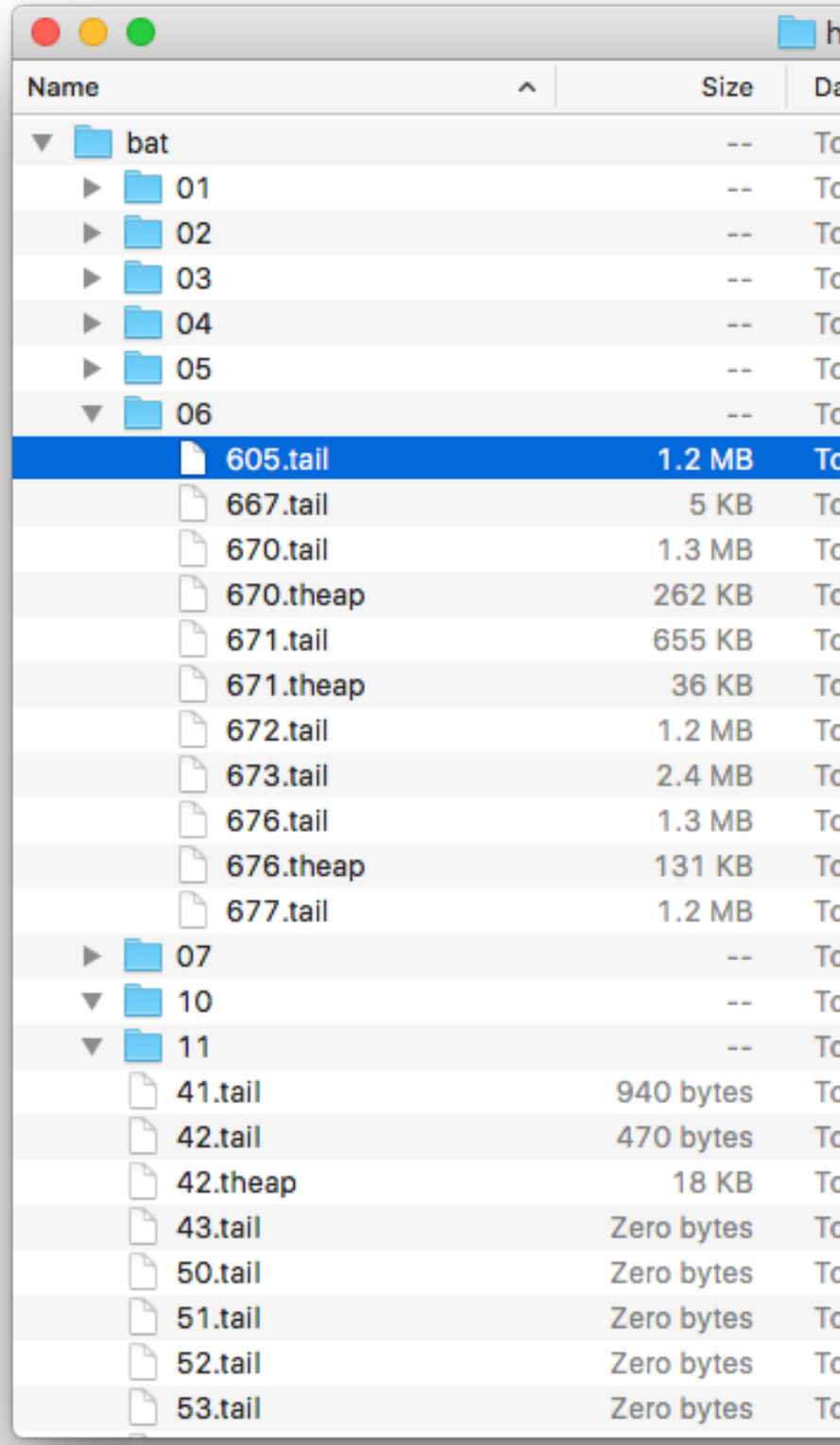
- Embedded & streamlined MonetDB for R
- All of MonetDB: Transactions, complex SQL etc.
- In-Process operation
- Query results are data frames
- Fast data append from data frames
- DBI and dplyr backends

Persistence

- Table data is stored on disk in a native *columnar* format

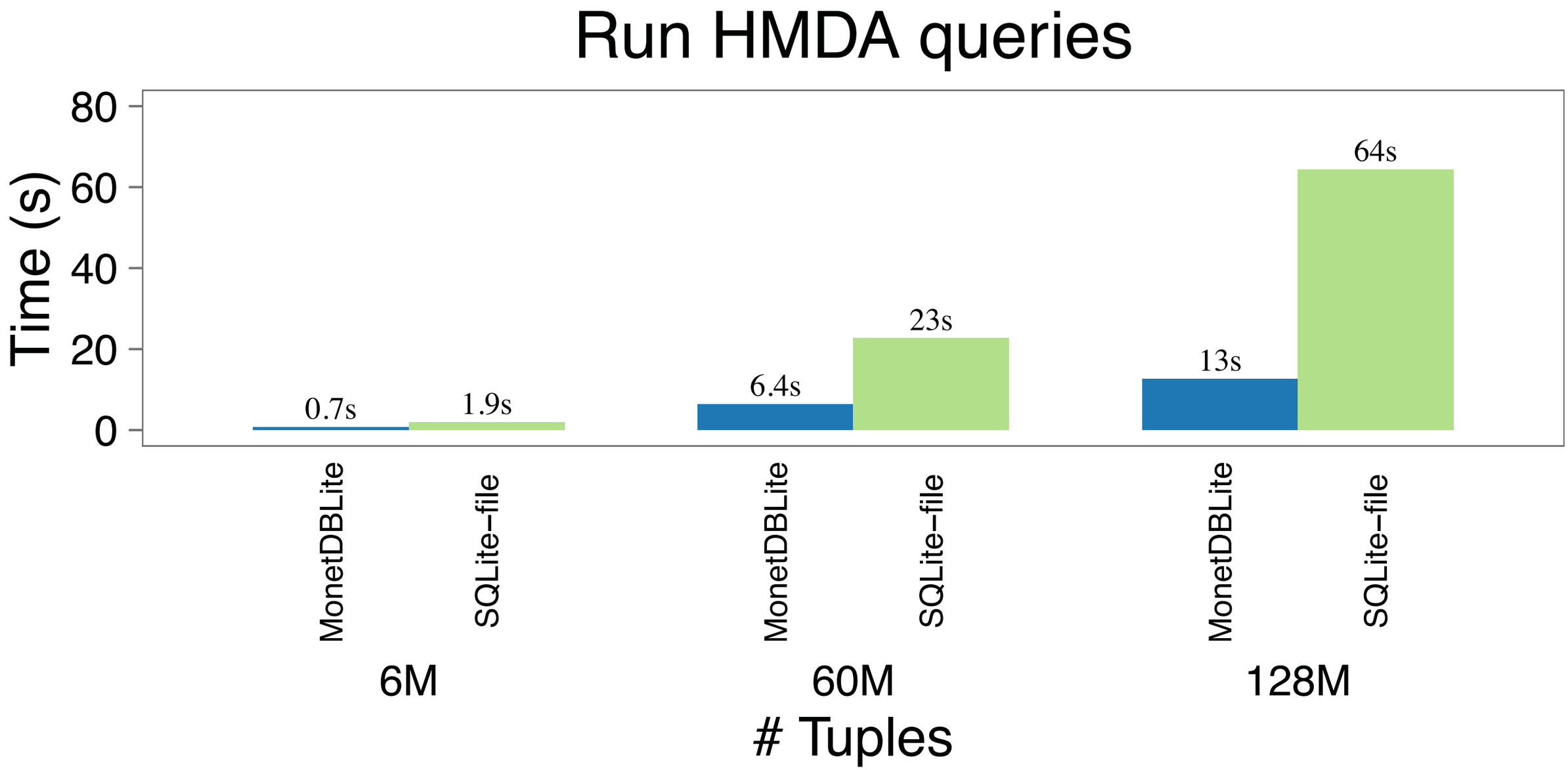
- Persistent, super-fast access

- No more waiting at beginning of script to load data…

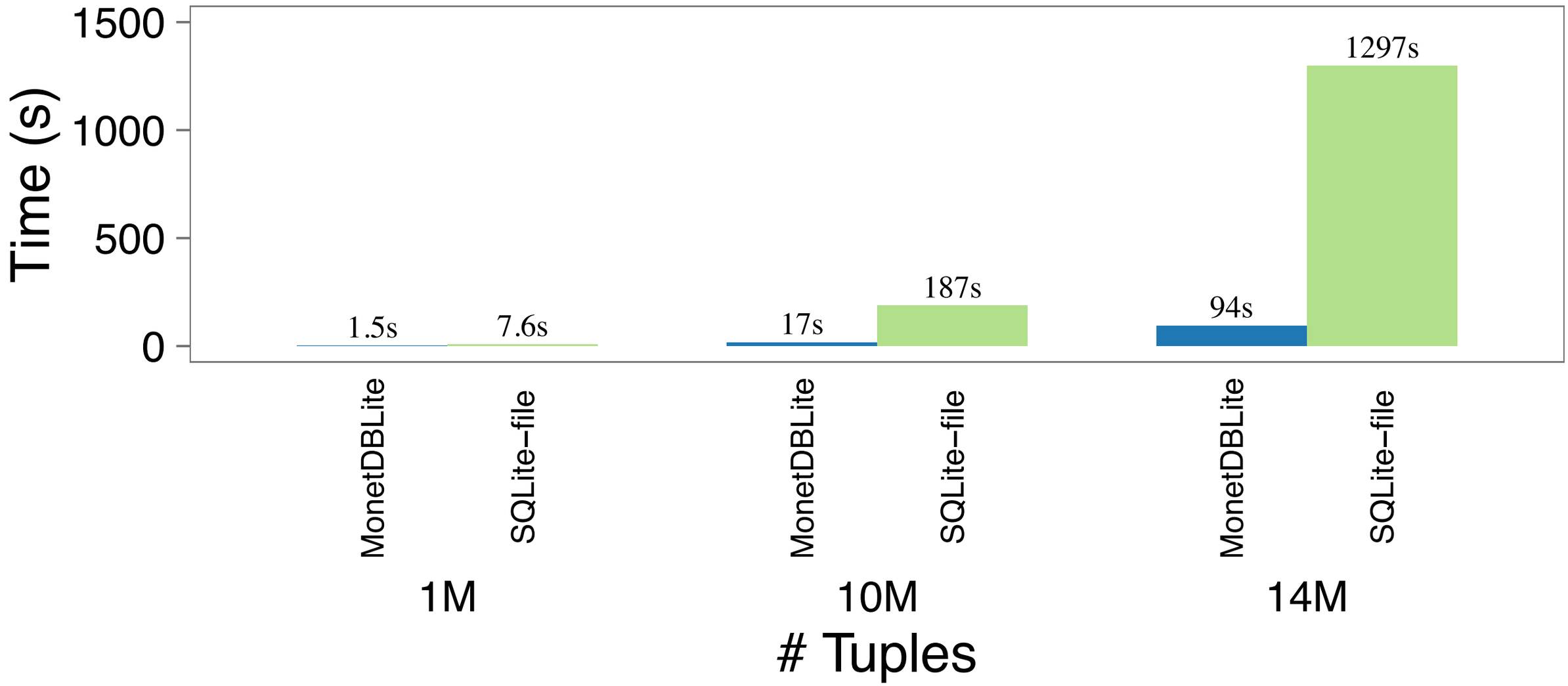


Performance

- Home Mortgage Disclosure Act dataset & queries
  - 128M records, 71 fields each, 56 GB CSV
  - Sampled down to 6M and 60M for testing
- Contenders: MonetDBLite & SQLite with a file backend
- Experiments:
  - HMDA queries
  - Table transfer

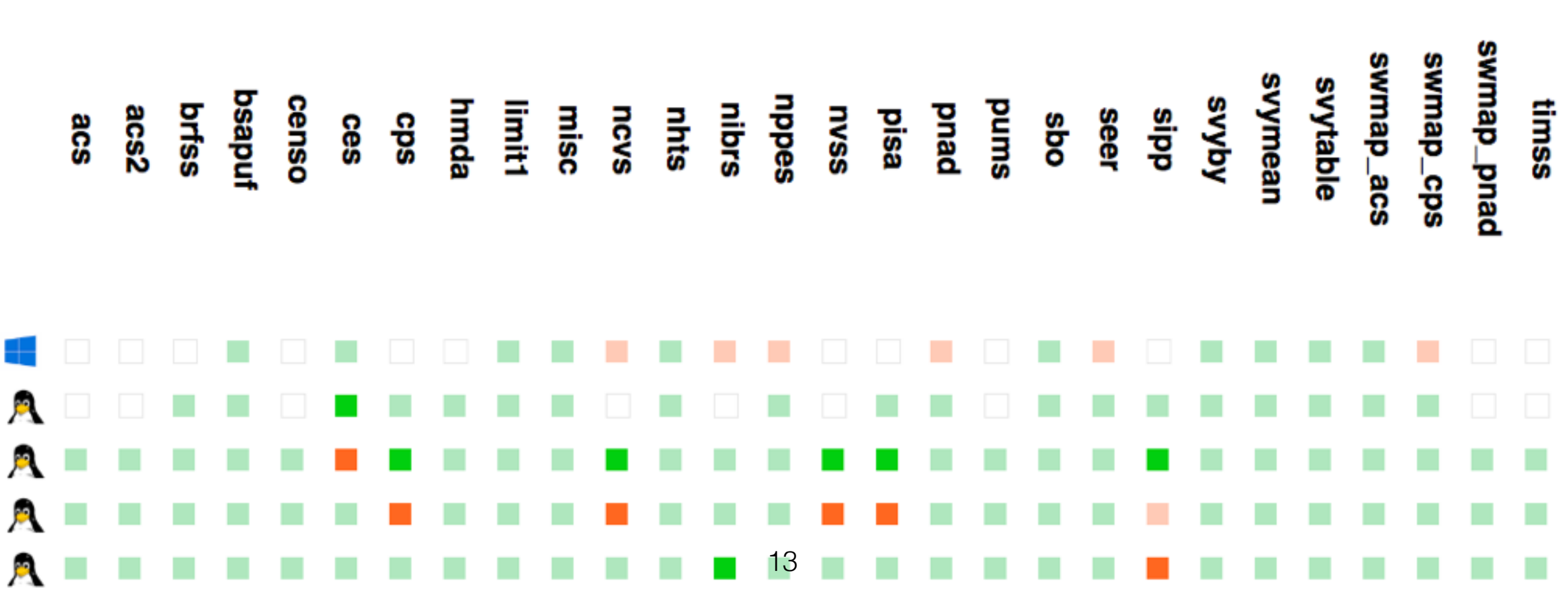

Run HMDA queries
Time (s)
80
60
40
20
0
0.7s
1.9s
6.4s
23s
13s
64s
MonetDBLite
SQLite−file
MonetDBLite
SQLite−file
MonetDBLite
SQLite−file
6M
60M
128M
Tuples

Convert table to data.frame



End-to-end-testing

- “Sisyphus”
  - http://www.asdfree.com , survey data with R
- Making sure analyses still runs with MonetDBLite



Live Demo

Script at https://gist.github.com/hannesmuehleisen/f6d590b4efcda539d0e8a27c420764dc

MonetDBLite

Fast SQL, fast startup

https://github.com/hannesmuehleisen/MonetDBLite

Lazy typers: http://ml.h4.ms

@hfmuehleisen