CWI

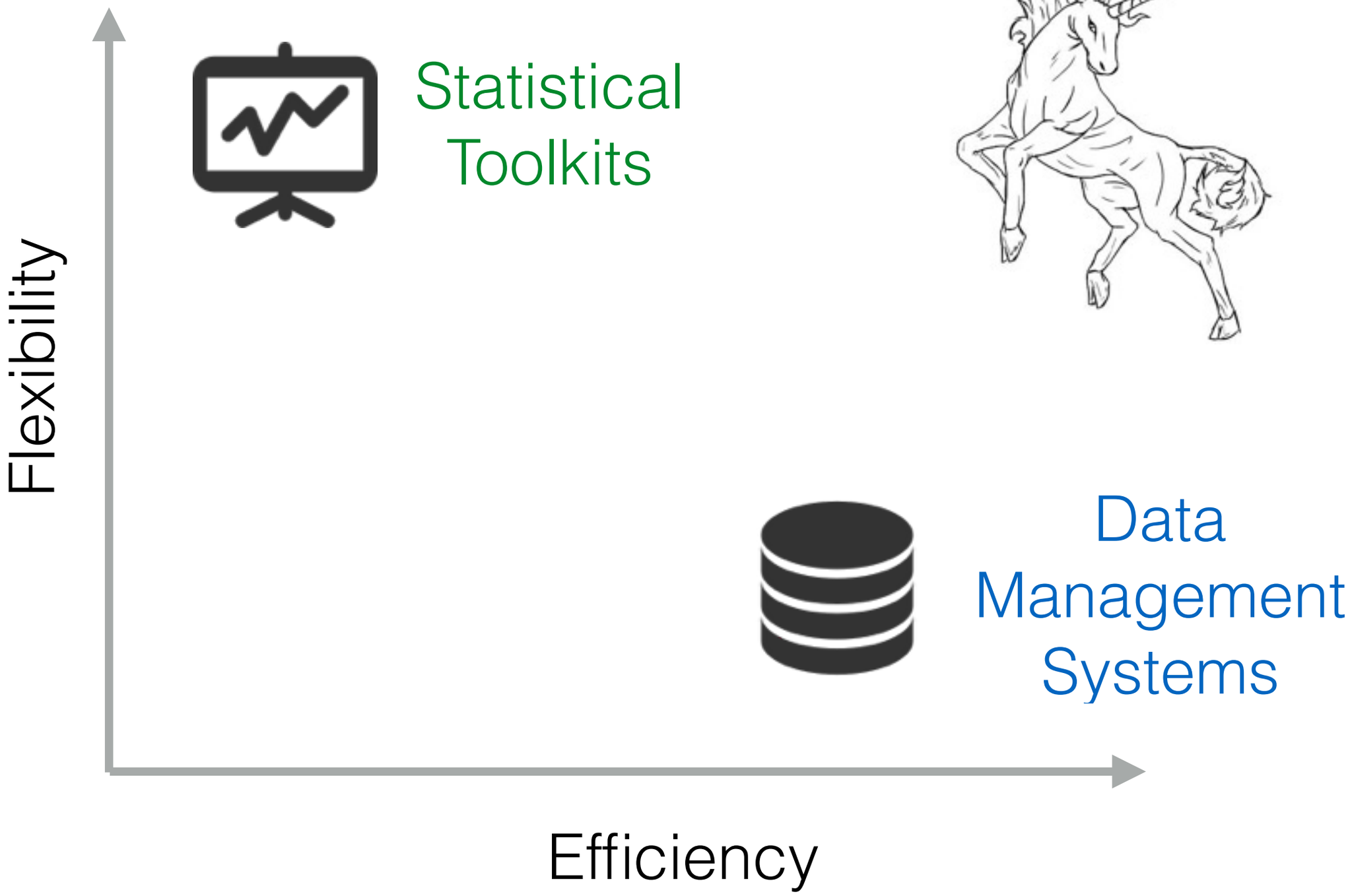
Centrum Wiskunde & Informatica

# Ad-Hoc User-Defined Functions for MonetDB with R

Hannes Mühleisen

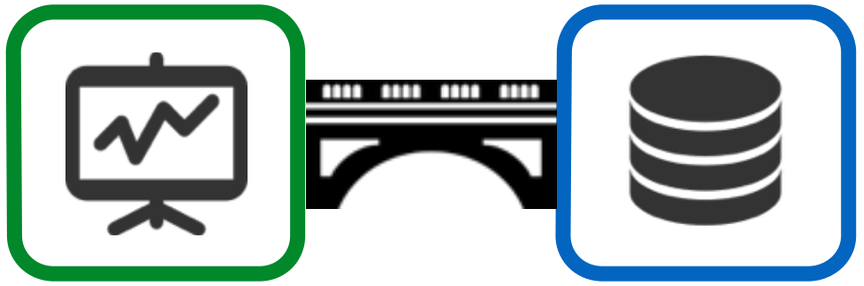
useR! 2015, 2015-07-01

# Which Systems?



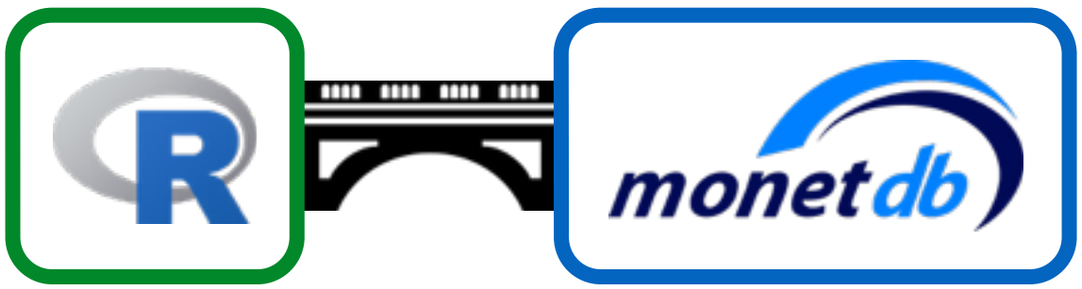
# Bridge the Gap

+ Native operators, lazy evaluation



+ Cheap data transfer

# Previously




MonetDB.R connector

DBI, dplyr backend

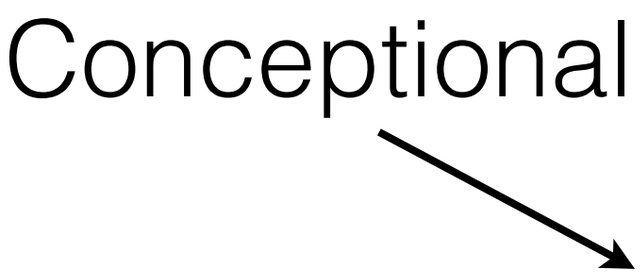
# Now


Embedded R in MonetDB

Part of MonetDB distribution since 2014

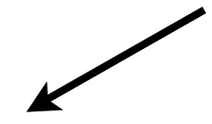
# Postgres, Oracle, DB2, etc.:

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 |

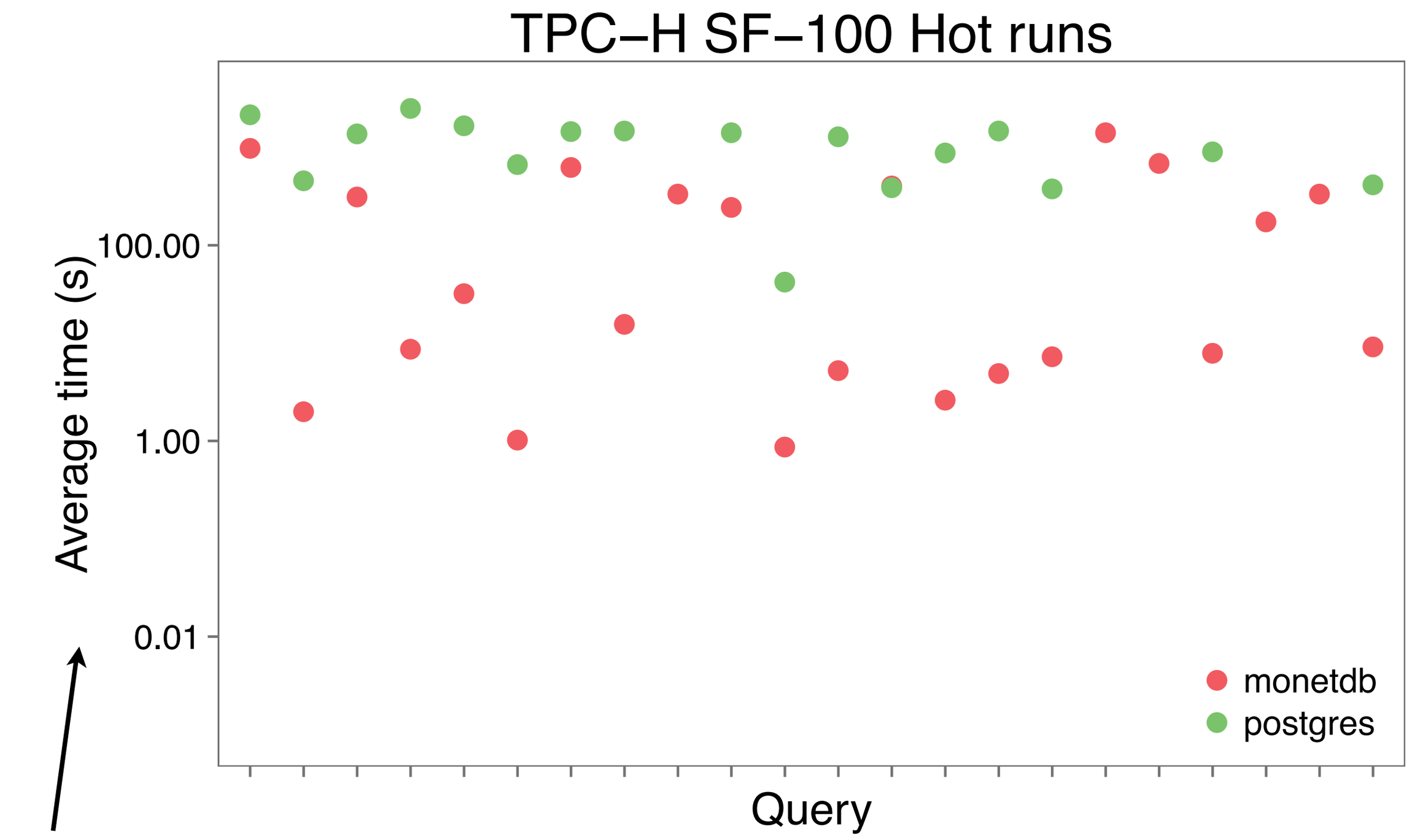
# Column Store:

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

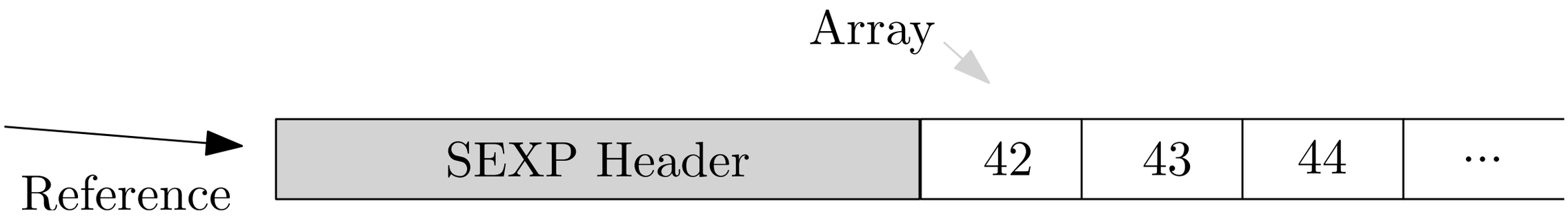
| NX | Constitution | Galaxy | Defiant | Intrepid |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 3 | 8 | 3 | 6 | 1 |

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

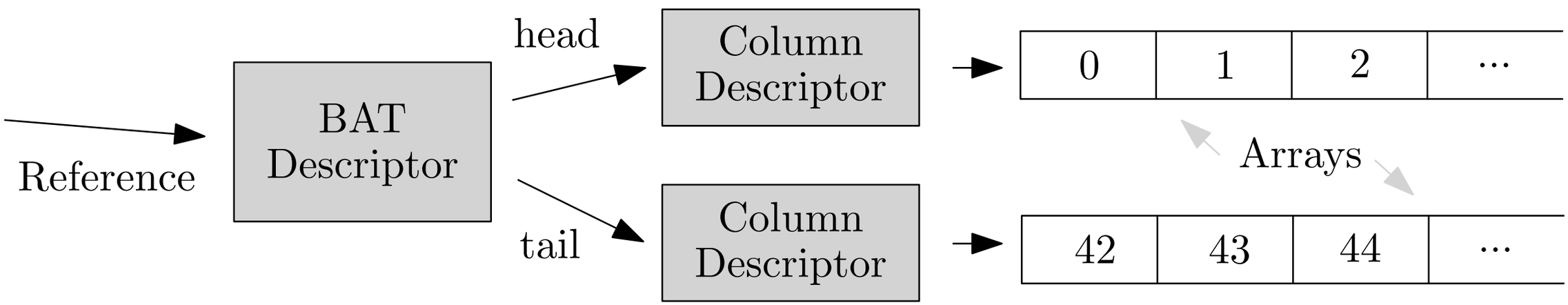
# Performance...



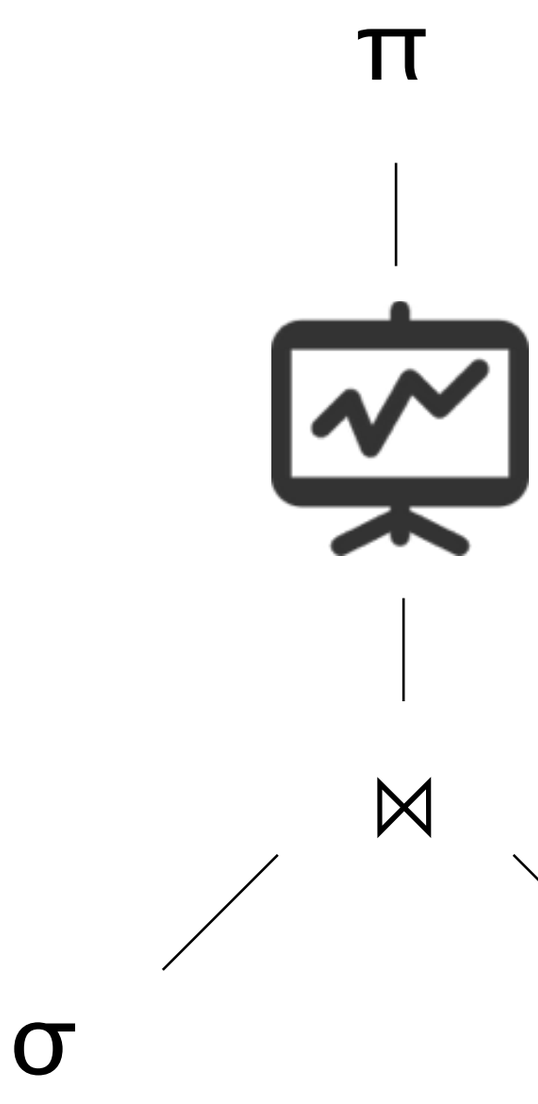
# R SEXP




# MonetDB BAT



# Relationally Integrated




+

Statistical analysis as operators in relational queries

# Table-producing

```
CREATE FUNCTION rapi01(i INTEGER)
RETURNS TABLE (i INTEGER, d DOUBLE)
LANGUAGE R { data.frame(i=seq(1,i),d=42.0) };

SELECT i,d FROM rapi01(42) AS r WHERE i>40;
```

# π Transformations

```
CREATE FUNCTION rapi02 (i INTEGER,
  j INTEGER, z INTEGER) RETURNS INTEGER
LANGUAGE R { i*sum(j)*z };

SELECT rapi02(i,j,2) AS r02 FROM rval;
```

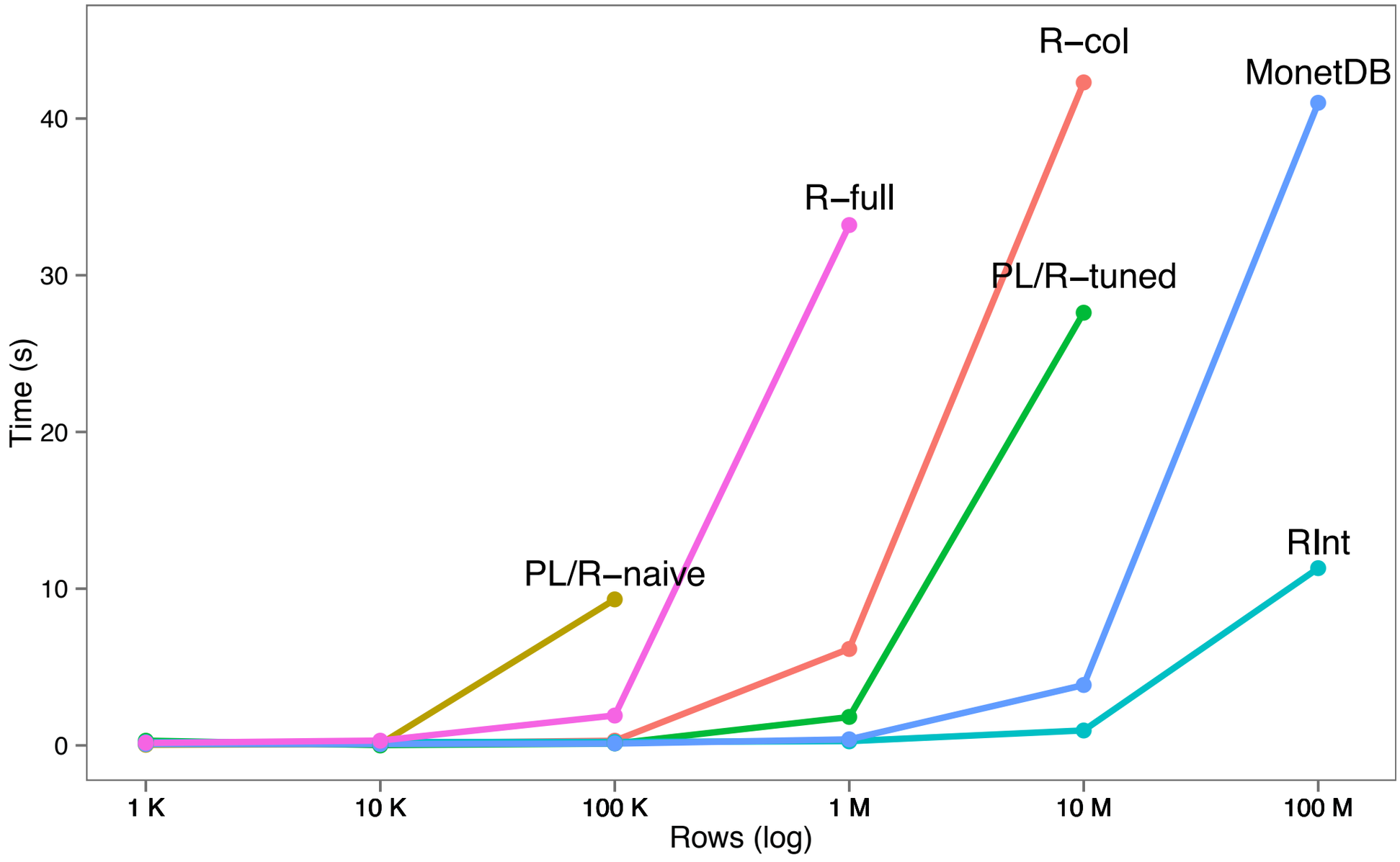
# σ Filtering

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

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

# Aggregation

```
CREATE AGGREGATE kmeans(data FLOAT, ncluster
INTEGER) RETURNS INTEGER
LANGUAGE R { kmeans(data,ncluster)$cluster };

SELECT cluster FROM (SELECT MIN(x) AS minx,
MAX(x) AS maxx, kmeans(x,5) AS cluster FROM
xdata GROUP BY cluster) as cdata ORDER BY
cluster;
```

# Performance…



# Code Shipping

```
> rf.fit <- randomForest(income~.,
data=training, mtry=2, ntree=10)
```

```
> predictions <- mdbapply(con, “t1",
function(d) {
  p <- predict(rf.fit, type=“prob",
    newdata=d)[,2]
  p[p > .9]
})
```

MonetDB.R 0.9.8, soon

# Demo

```
> system.time(mdbapply(con, "flights", summary))
   user  system elapsed
  0.013   0.000   0.654
> system.time(summary(dbReadTable(con, "flights")))
   user  system elapsed
  1.756   0.165   3.419
```

CWI
Centrum Wiskunde & Informatica

monetdb

# Thank You

# Questions?

http://www.monetdb.org

@hfmuehleisen