Faster than C?

Parsing binary data in JavaScript

Felix Geisendörfer

7. Oct, 2012 - JSConf.eu

@felixge

transloadit.com

Faster than C?

Sorry about the “title bait”

High performance JavaScript

JavaScript vs C

Good vs Evil

Good Parts vs Evil

Bad Parts vs Evil

early 2010

No MySQL module

for node.js

early 2010

All we had was

NoSQL Libraries

early 2010

Yuichiro MASUI

masuidrive

node-mysql

JavaScript ★ 107 10

Pure JS / No C/C++

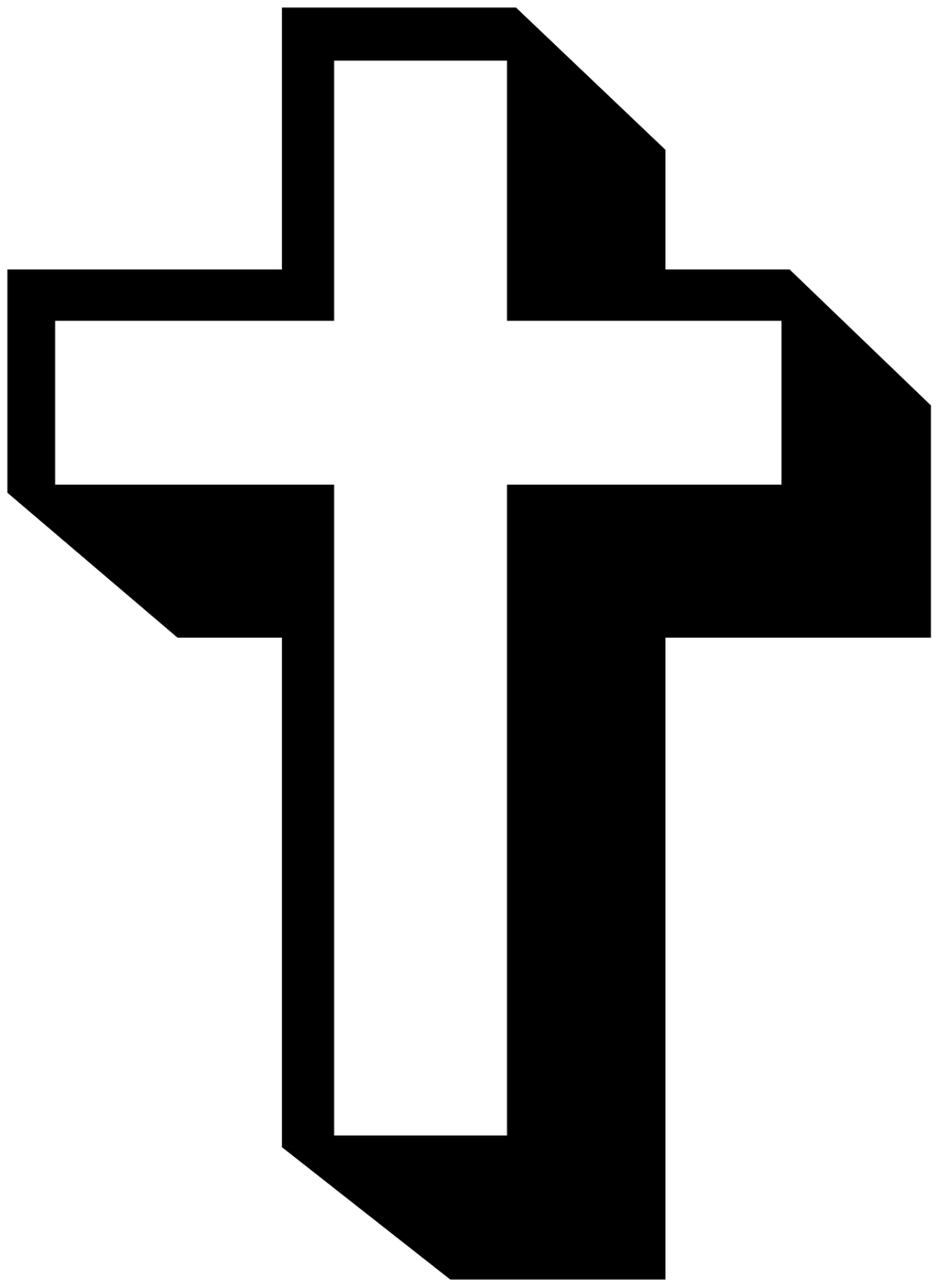
Before Buffers became usable

The Parser was using JavaScript Strings

Node.js Trivia

“Buffers” used to be called “Blobs”

For 3 min and 15 sec



RIP Blobs

Sun Dec 13 08:39:20 2009

-

Sun Dec 13 08:42:45 2009

Anyway

mysql can be done without libmysql

Felix Geisendörfer

felixge

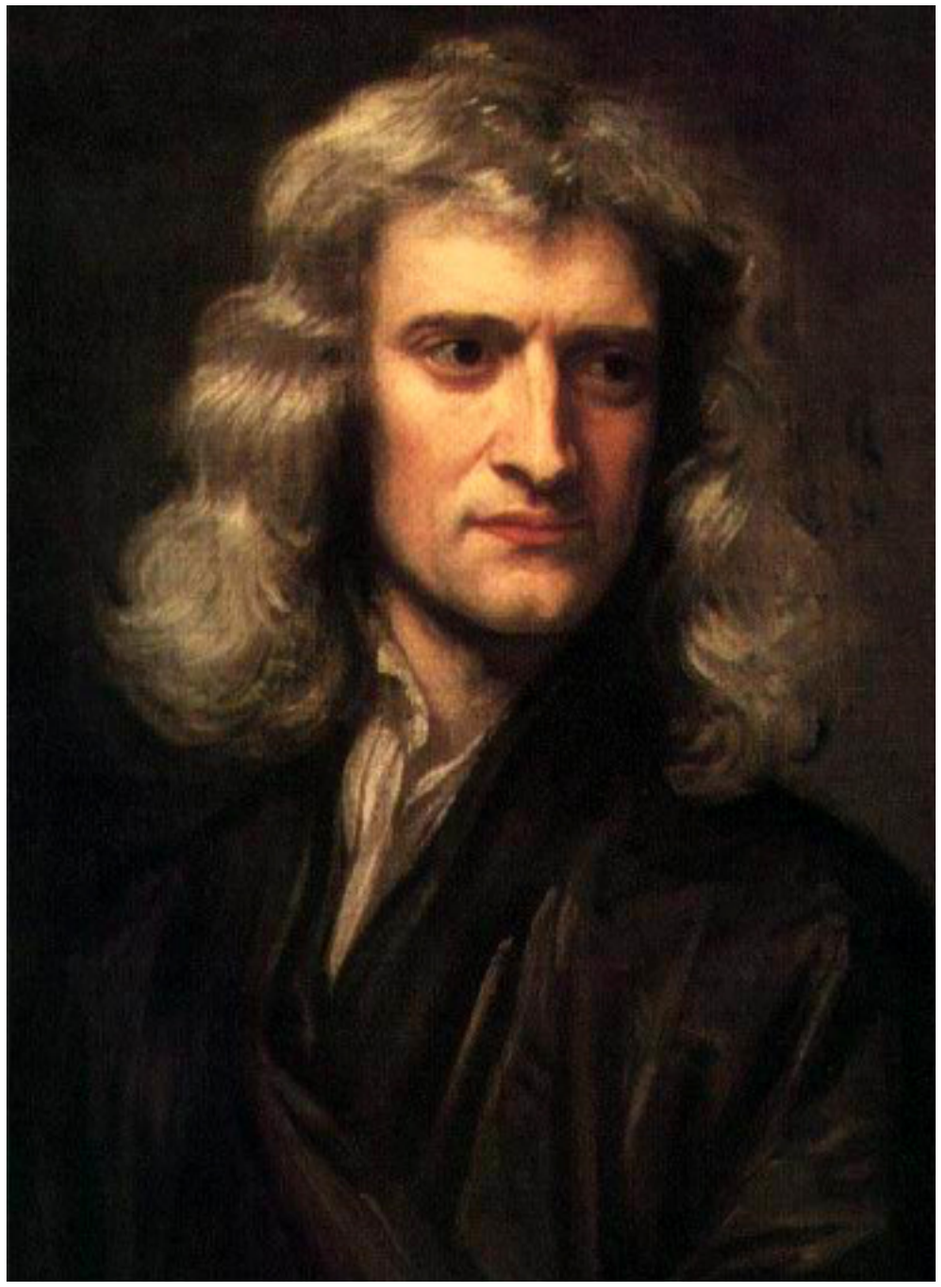
node-mysql

JavaScript ★ 1,364 161

A pure node.js JavaScript Client implementing the MySql protocol.

Last updated 4 minutes ago

No good deed goes unpunished



Sir Isaac Newton

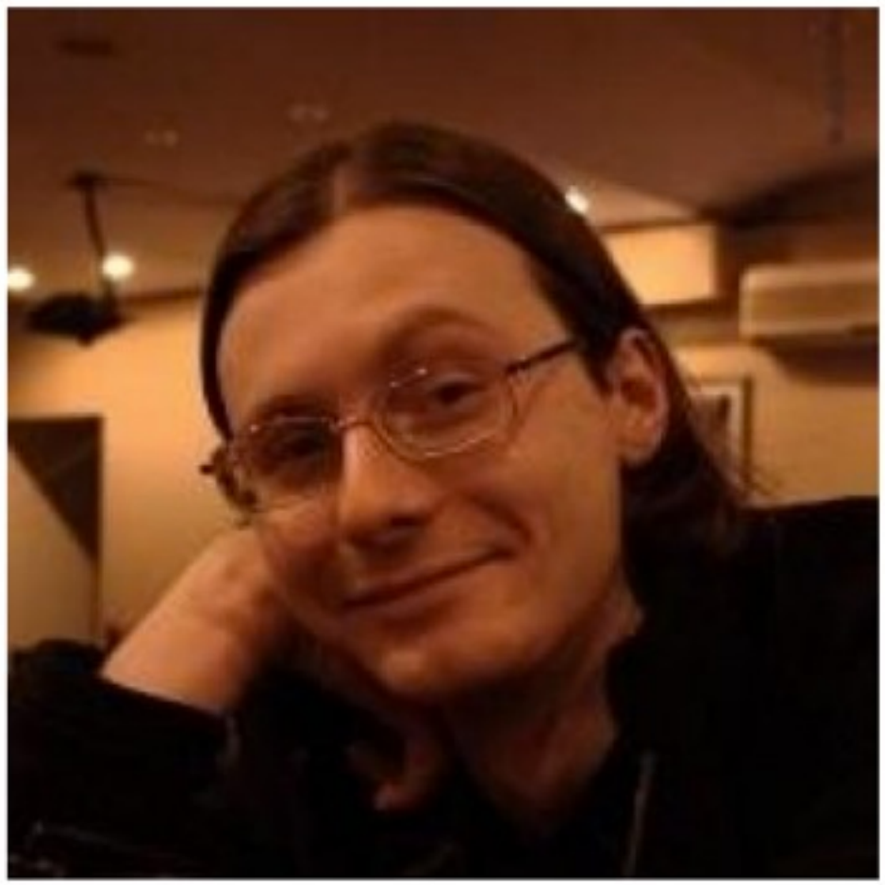
Third Law of Motion

“When a first body exerts a force F1 on a second body, the second body simultaneously exerts a force F2 = –F1 on the first body. This means that F1 and F2 are equal in magnitude and opposite in direction.”

Third Law of Github

“When a first person pushes a library `L1` into a remote repository, a second person simultaneously starts working on a second library L2 which will be equally awesome, but in a different way.”

<3 Github!



Oleg Efimov

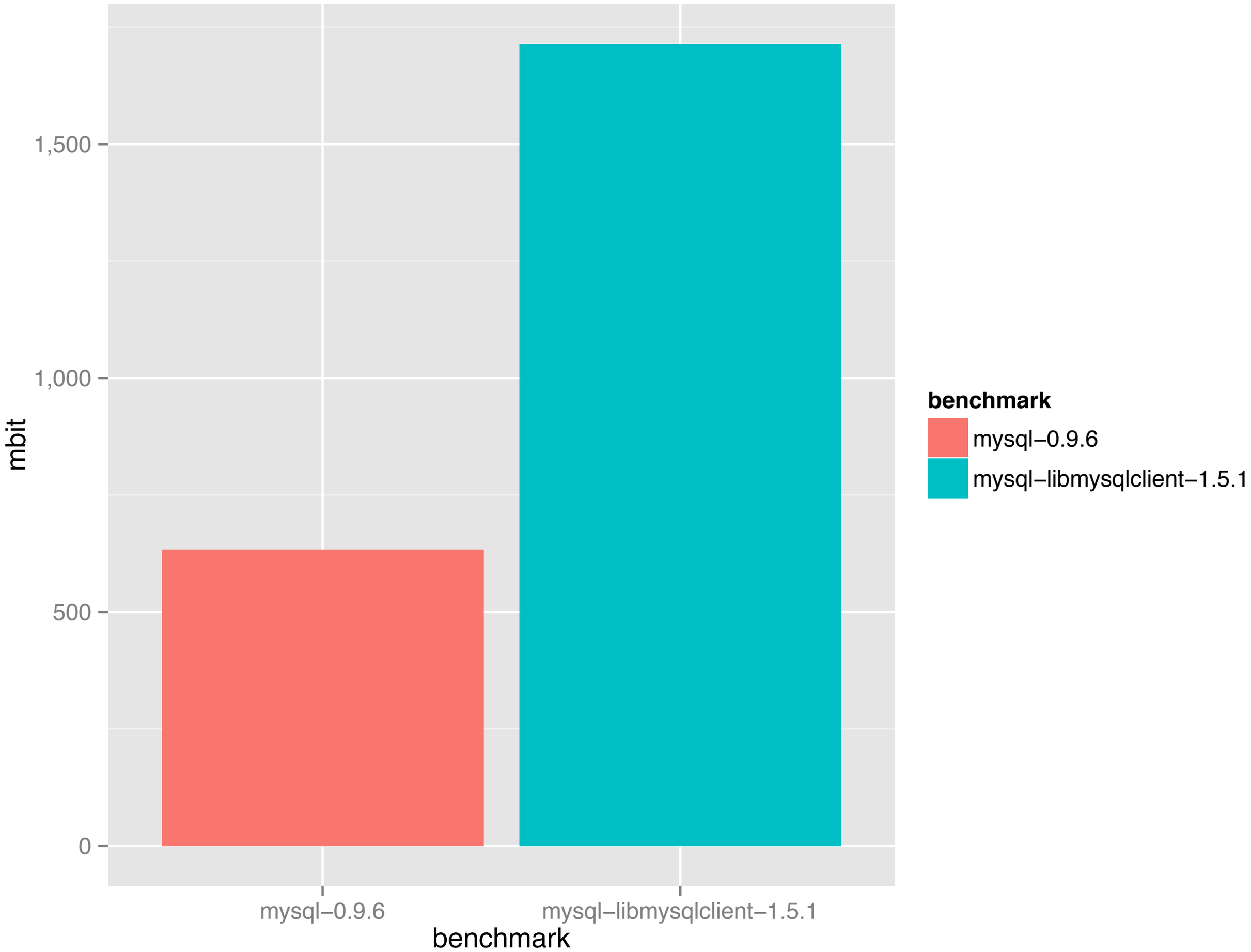
Sannis

node-mysql-libmysqlclient

JavaScript ★ 165 28

Asynchronous MySQL binding for Node.js

Last updated 25 days ago


1,500
1,000
500
0
mbit
mysql–0.9.6
mysql–libmysqlclient–1.5.1
benchmark
benchmark
mysql–0.9.6
mysql–libmysqlclient–1.5.1

Of course.

libmysql = C

my library = JavaScript

C > JS, right?

But V8!

And Crankshaft!!

Node.js !!!1!

Was I living a lie?

Kind of

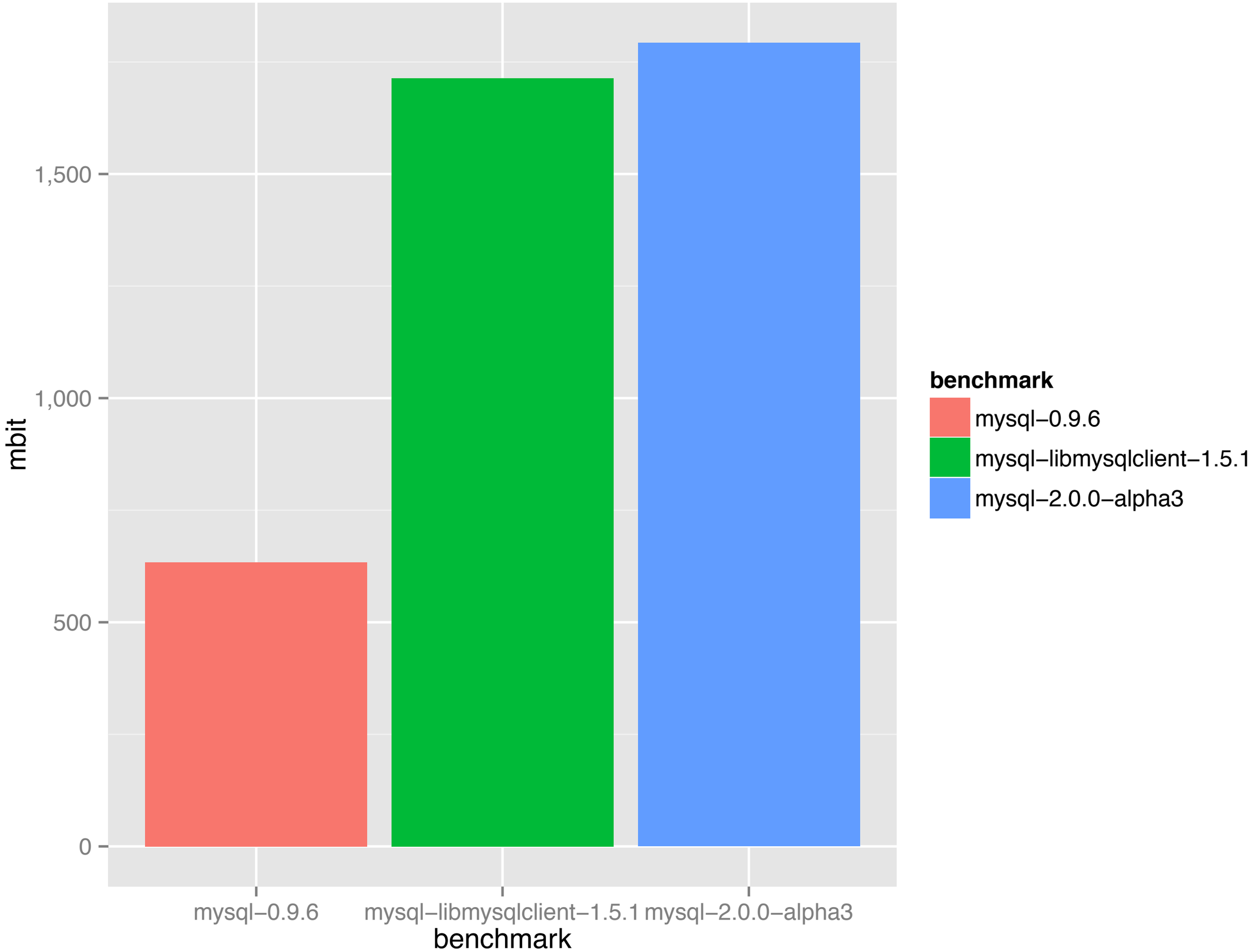
V8 / Node

= Tools

Performance

is not a tool

Performance is hard work & data analysis


mbit
1,500
1,000
500
0
mysql–0.9.6
mysql–libmysqlclient–1.5.1
mysql–2.0.0–alpha3
benchmark
benchmark
mysql–0.9.6
mysql–libmysqlclient–1.5.1
mysql–2.0.0–alpha3

Third Law of Github

Brian White

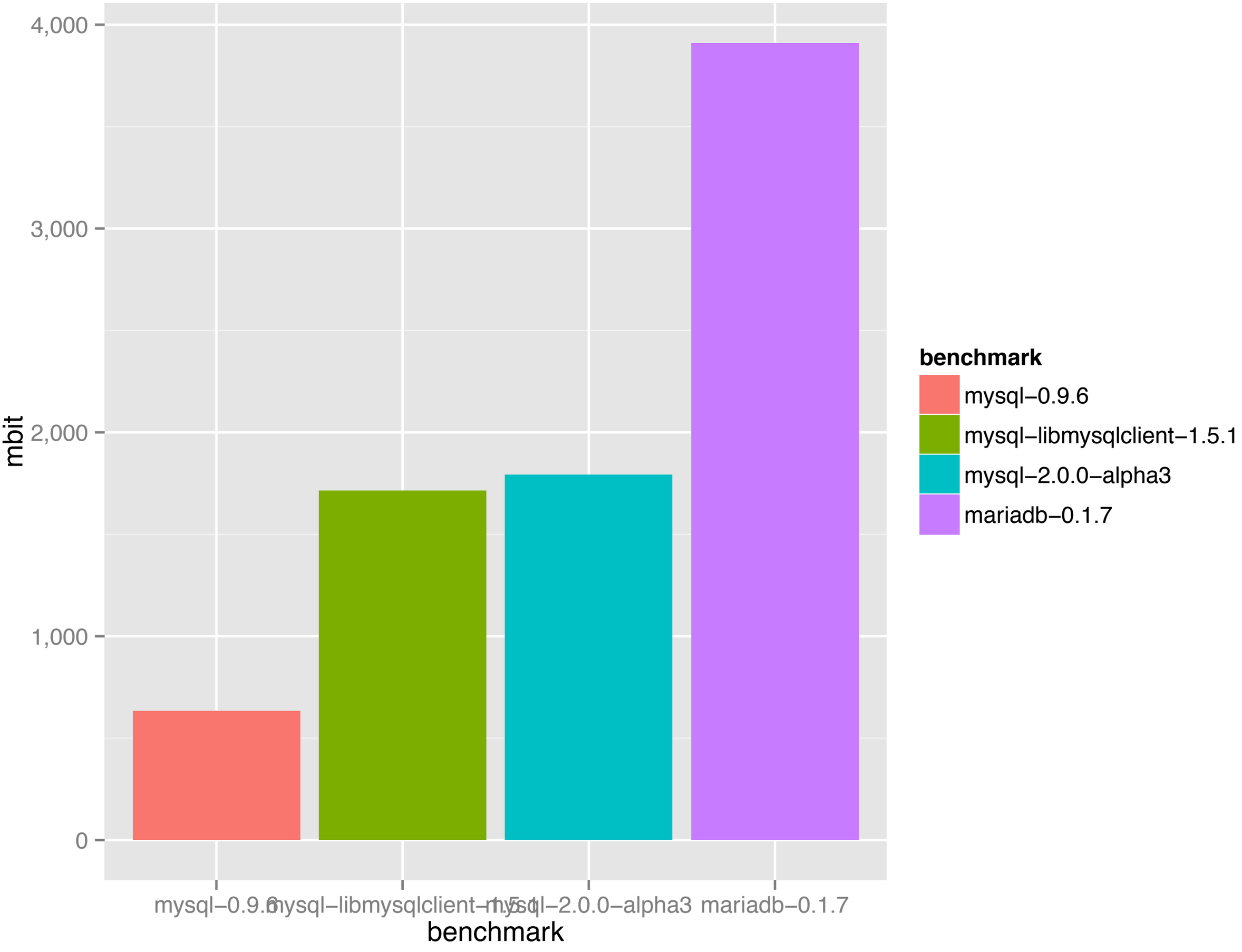
mscdex

node-mariasql

C++ ★ 23 0

A node.js binding to MariaDB's non-blocking (MySQL-compatible) client library

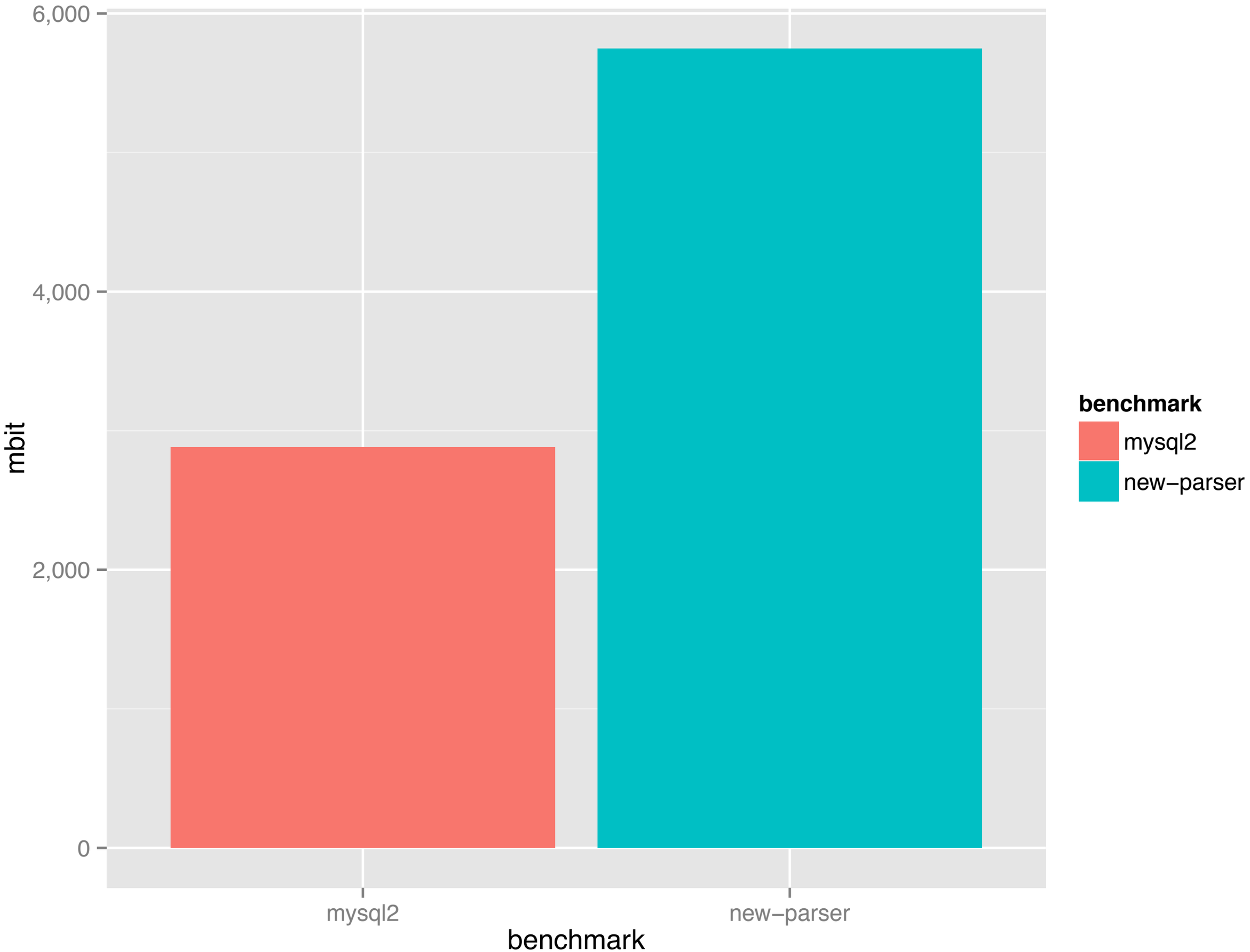
Last updated 4 days ago


4,000
3,000
2,000
1,000
0
mbit
mysql−0.9.6
mysql−libmysqlclient−1.5.1
mysql−2.0.0−alpha3
mariadb−0.1.7
benchmark
benchmark
mysql−0.9.6
mysql−libmysqlclient−1.5.1
mysql−2.0.0−alpha3
mariadb−0.1.7

Time to give up?

NEVER!

New Parser


6,000
4,000
2,000
0
mbit
mysql2
new–parser
benchmark
benchmark
mysql2
new–parser

Third law of Github?

Endgame

Last bottleneck:

Creating JS Objects

Also: MySQL Server saturated

Anyway

How to write fast JS

Does not work

Profiling

- Good for spotting small functions with stupid algorithms performing many iterations
- Bad for complex functions with many primitive operations

Taking performance advice from strangers

- Good for ideas & inspiration

- But useless when applied cargo-cult style

Does Work

BDD

~~Behavior Driven~~

~~Development~~

Benchmark Driven Development

Benchmark Driven Development

- Similar to test driven development
- Use it when performance is an explicit design goal
- Benchmark first > benchmark after !

```
function benchmark() {
    // intentionally empty
}
```

```
while (true) {
    var start = Date.now();
    benchmark();
    var duration = Date.now() - start;
    console.log(duration);
}
```

Benchmark Driven Development

- Next step: Implement a tiny part of your function
- Example: Parse headers of MySQL packets
- Look at impact, tweak code, repeat

Example Results

- try...catch is ok
- big switch statement = bad
- function calls = very cheap
- buffering is ok

Favorite

```
function parseRow(columns, parser) {
  var row = {};
  for (var i = 0; i < columns.length; i++) {
    row[columns[i].name] = parser.readColumnValue();
  }
  return row;
}
```

Make it faster!

```
var code = 'return {\n';

columns.forEach(function(column) {
  code += '"' + column.name + '":' + 'parser.readColumnValue(),\n';
});

code += '};\n';

var parseRow = new Function('columns', 'parser', code);
```

->

```
function parseRow(columns, parser) {
  return {
    id      : parser.readColumnValue(),
    title   : parser.readColumnValue(),
    body    : parser.readColumnValue(),
    created : parser.readColumnValue(),
    updated : parser.readColumnValue(),
  };
}
```

eval = awesome

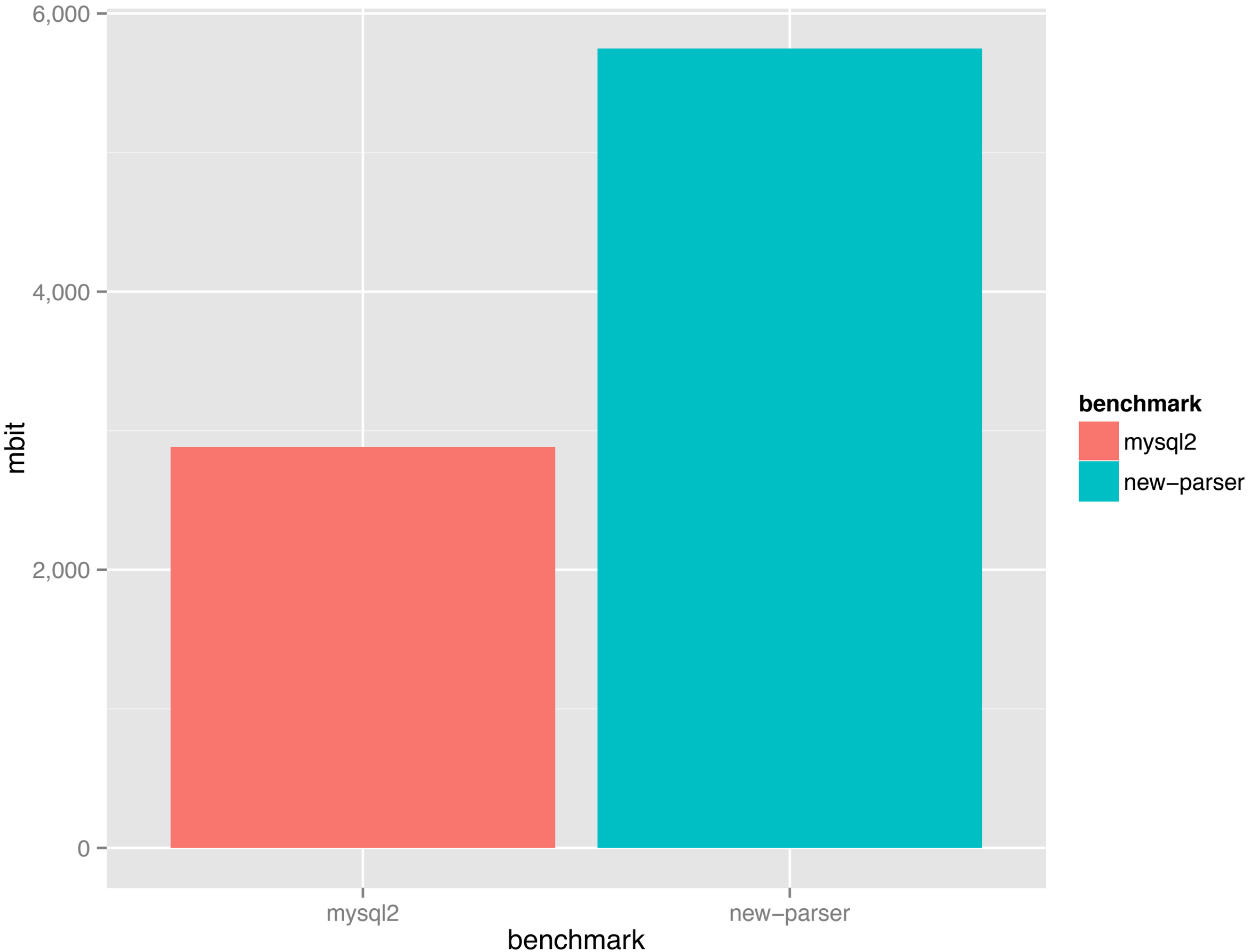
Data analysis

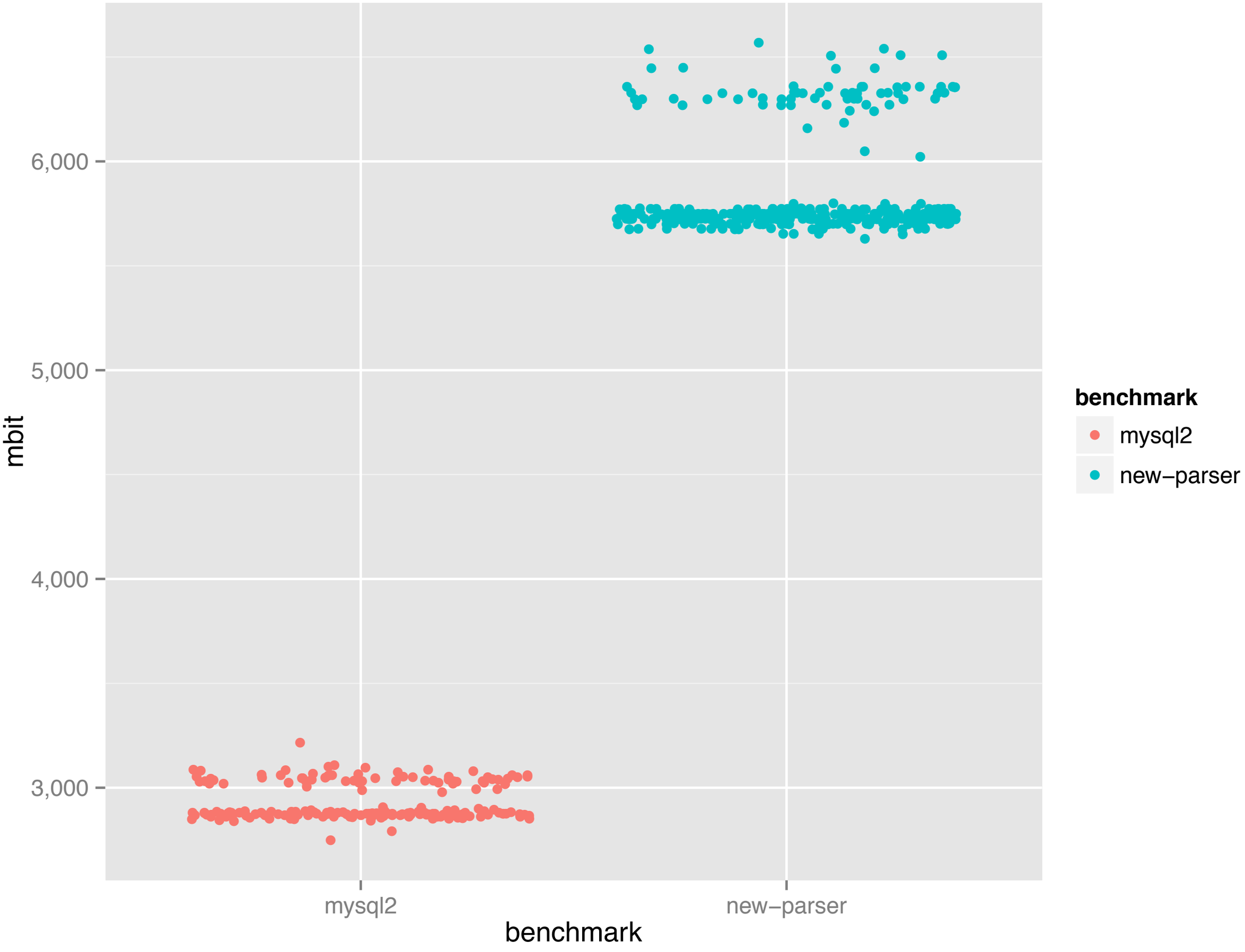
- Produce data points as tab separated values
- Add as many VM/OS metrics as you can get to every line
- Do not mix data and analysis !!

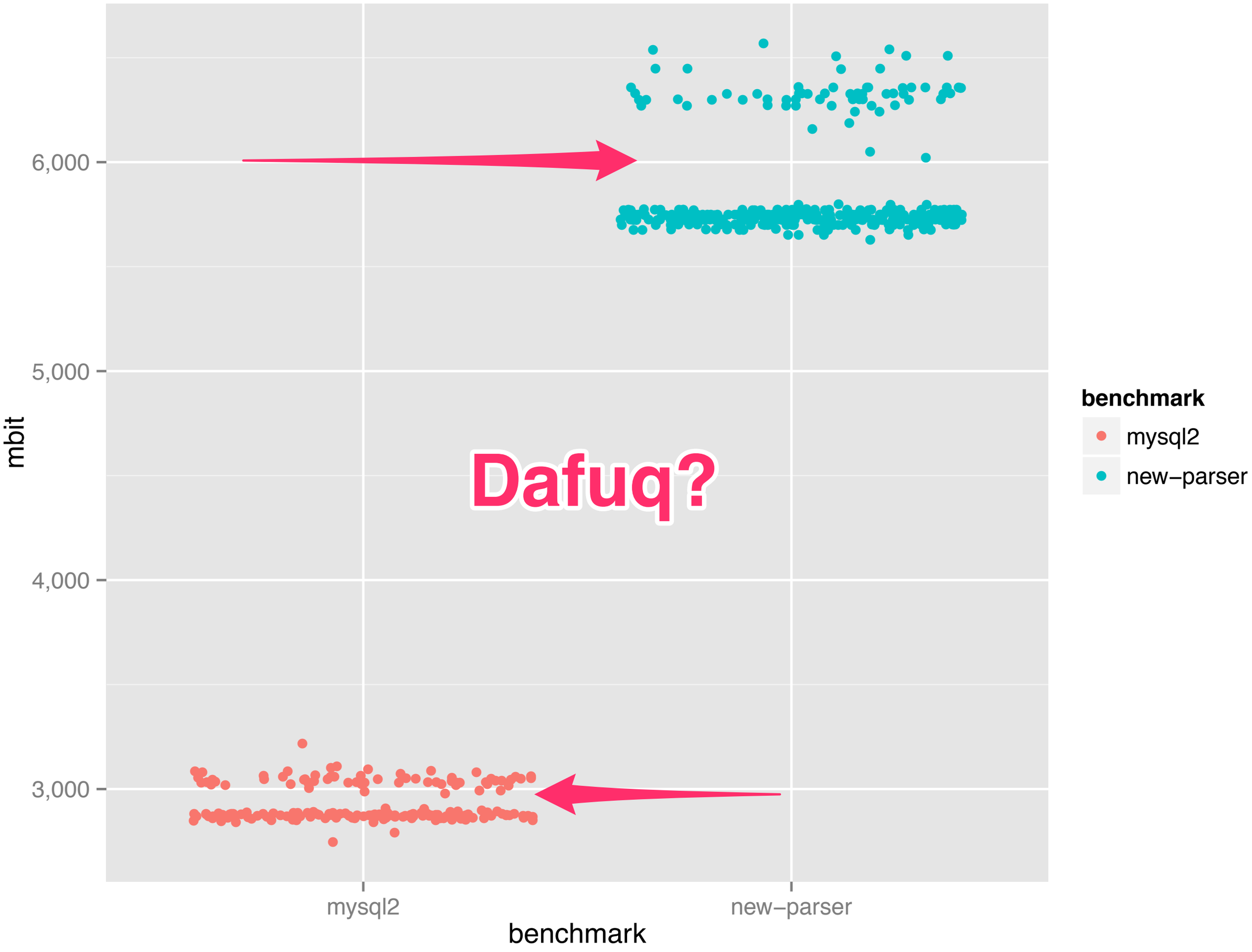
Recommended Tools

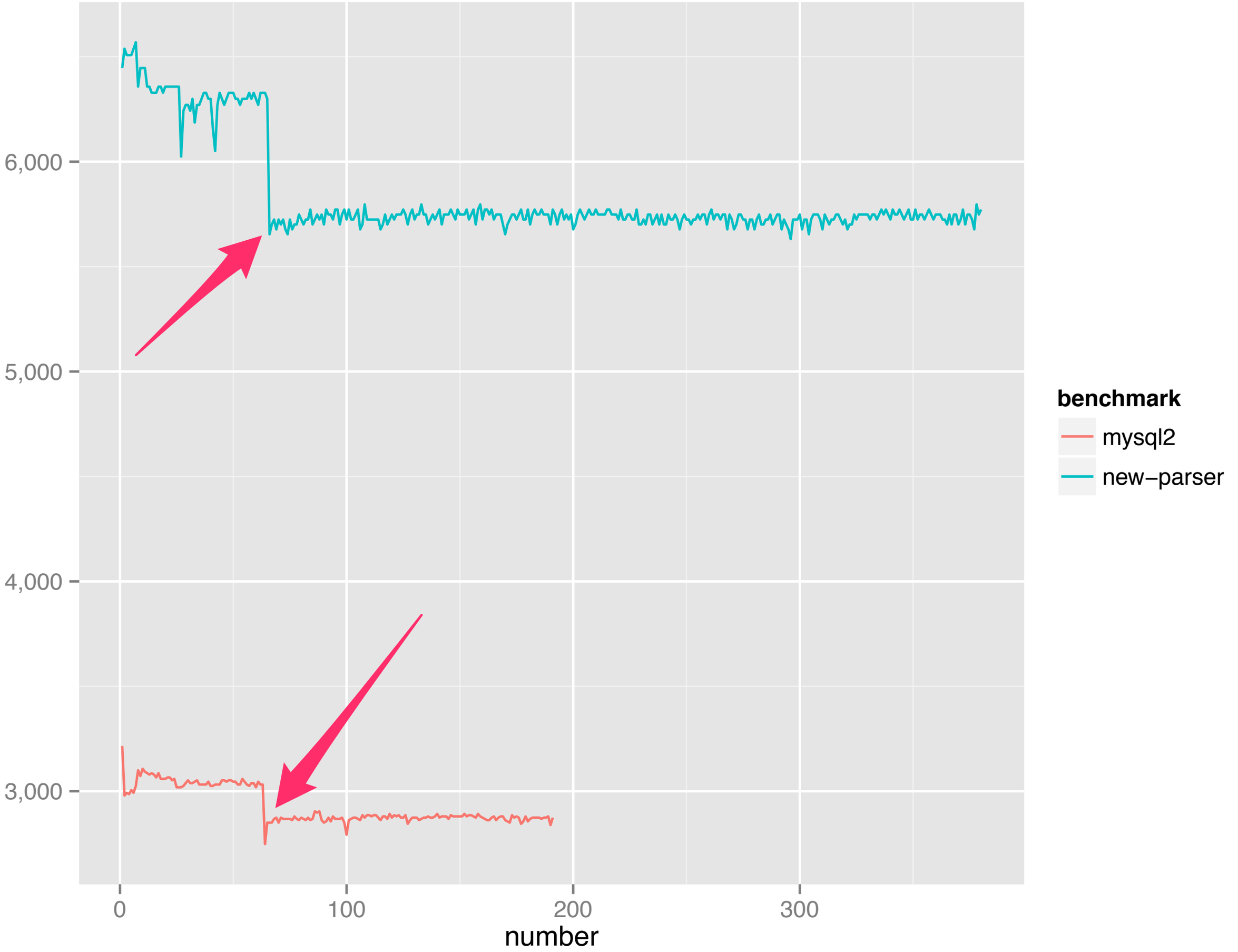
- node benchmark.js | tee results.tsv
- R Programming language (with ggplot2) !
- Makefiles, Image Magick, Skitch

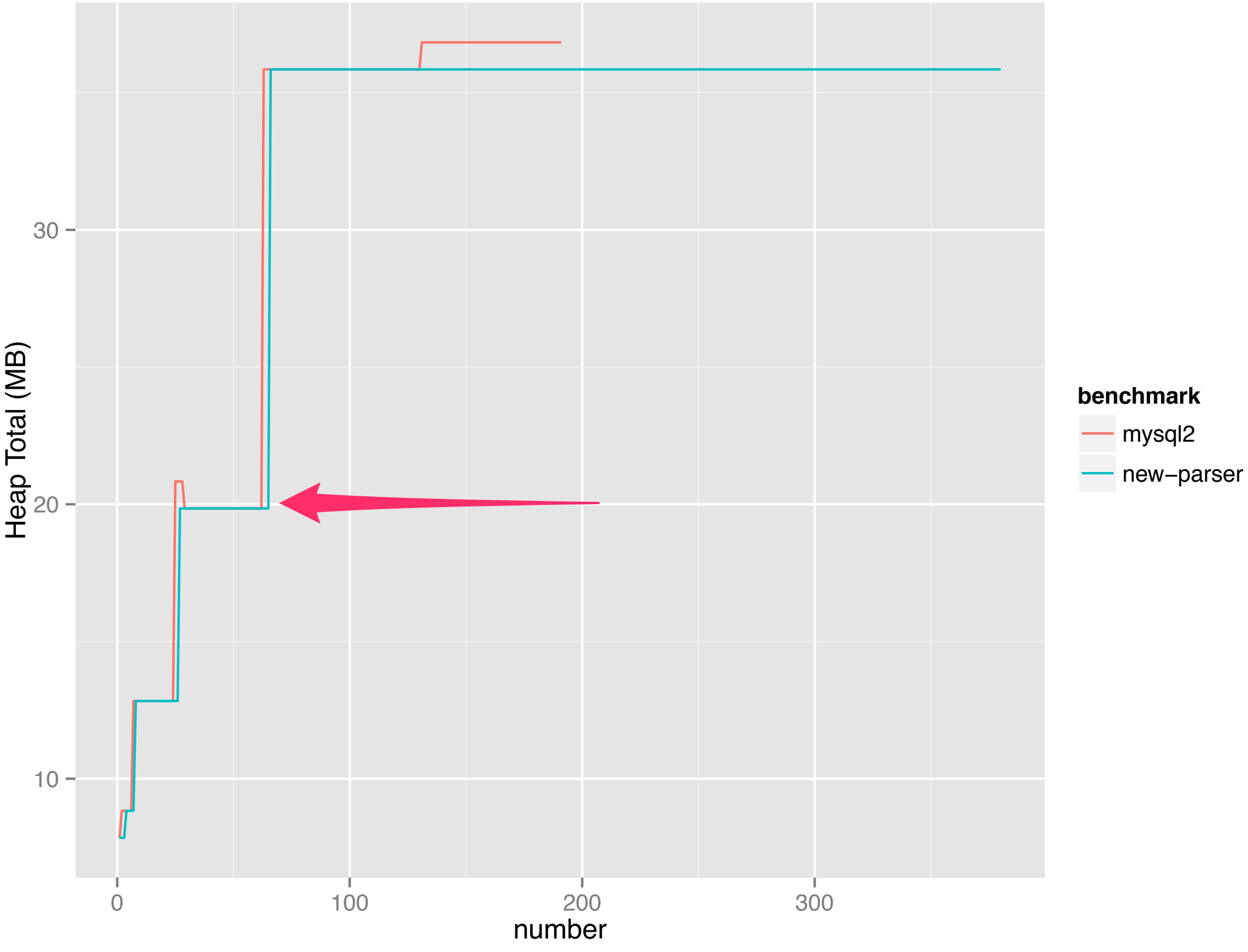
Why?

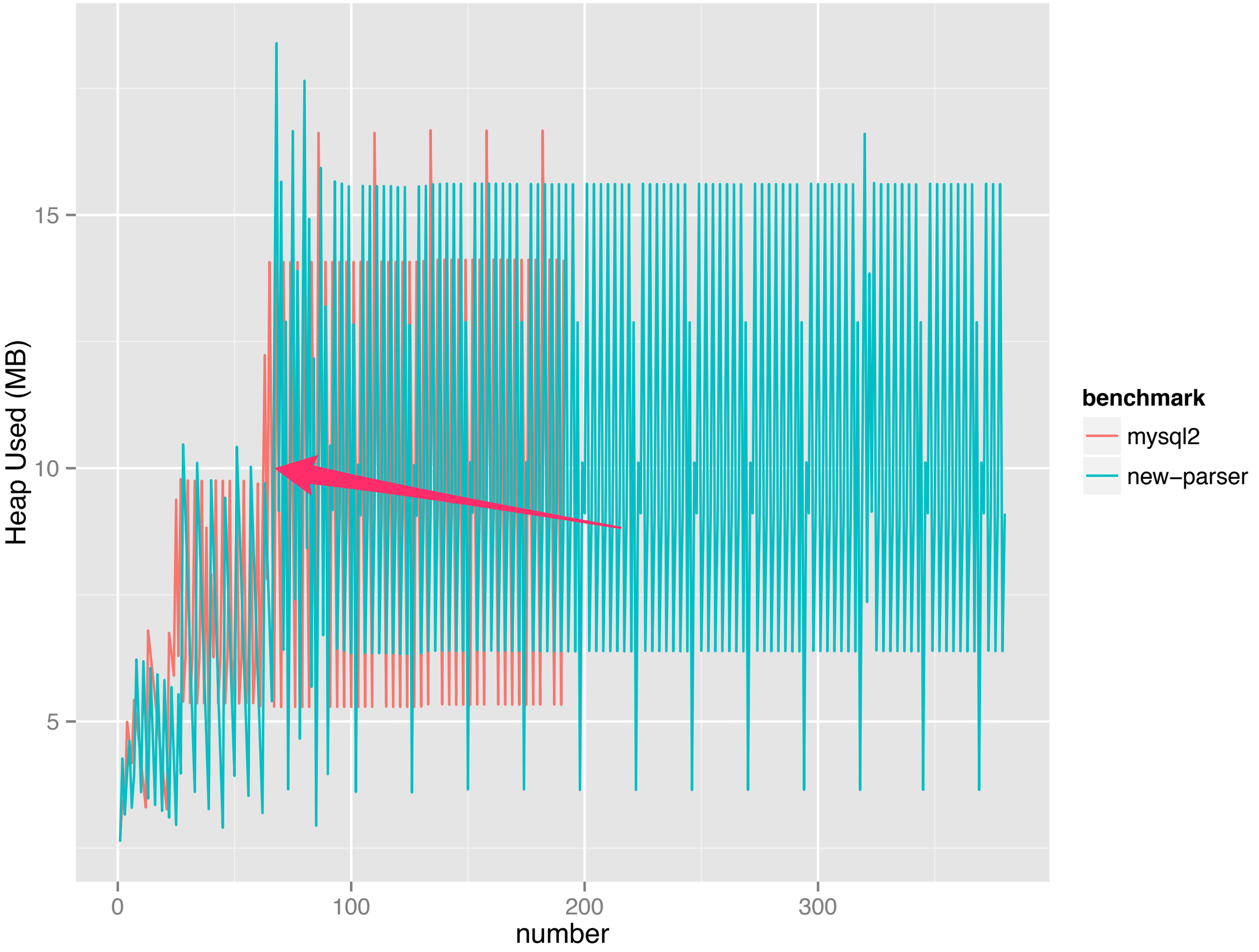

6,000
4,000
2,000
0
mbit
mysql2
new–parser
benchmark
benchmark
mysql2
new–parser


6,000
5,000
4,000
3,000
mbit
mysql2
new−parser
benchmark
benchmark
mysql2
new−parser


6,000
5,000
4,000
3,000
mysql2
new–parser
benchmark
Dafuq?
benchmark
mysql2
new–parser


6,000
5,000
4,000
3,000
mbit
0
100
200
300
number
benchmark
mysql2
new−parser


Heap Total (MB)
30
20
10
0
100
200
300
number
benchmark
mysql2
new−parser


Heap Used (MB)
15
10
5
0
100
200
300
number
benchmark
mysql2
new−parser

Collect data ->

Analyze it ->

Find problems ->

Tweak the code ->

Repeat

Thank you

Thank you !

github.com/felixge/faster-than-c

All benchmarks, results and analysis scripts