Progress Report of JerryScript Engine

Zoltan Herczeg
zherczeg.u-szeged@partner.samsung.com

Samsung Research Hub @ University of Szeged

JerryScript Developer Meeting 2016
Staines, UK, April 26, 2016

Overview

Introduction

JerryScript Engine Introduction

- JerryScript is a lightweight ECMAScript 5.1 engine, which is optimized for low-end systems
  - Embedded systems with 32 bit CPU and 64K or less RAM
- Open source: https://github.com/Samsung/jerryscript
- The primary focus of the project has been memory and binary size reduction
  - Performance has also been focus since February 2016

Key Features

- JavaScript is translated to byte code, no intermediate representation (e.g. AST)
- Compact Byte Code: a unique variable length byte code with lightweight data compression
- ECMA values are represented with small objects to reduce memory footprint
- Snapshot: ECMAScript source files can be compiled ahead of time and can be executed from ROM

ECMAScript Conformance

- Test262 is the official ECMAScript conformance test suite
- The es5-tests branch contains the ES 5.1 related tests
- We have achieved **100%** test coverage excluding internalization tests
  - Date support must be enabled
  - Time zone must be set to zoneinfo/America/Los_Angeles
- The last ~20 failures has been fixed this year

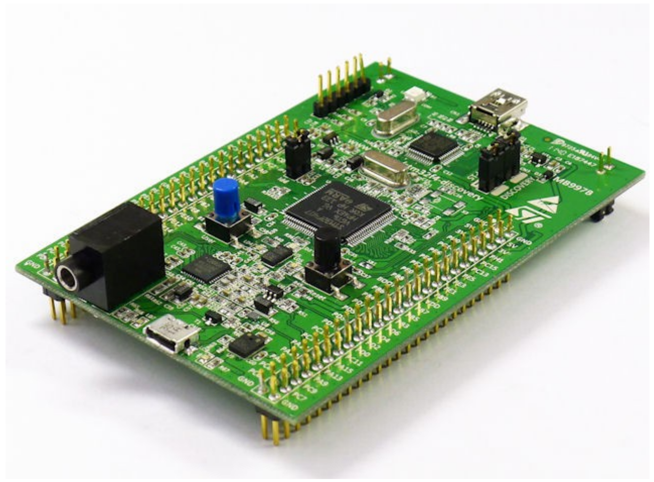
Measurement Overview

ECMAScript Engines

- In the followings three engines are compared
  - Jerry-20-Apr: JerryScript revision a3b1db36
  - Jerry-04-Feb: JerryScript revision db6caf3c
    - Before performance optimizations
  - Duktape 1.4.0 (10.01.16)
- Duktape is a middle level JS engine, which scales moderately towards low-end and high-end systems
  - Balanced between performance and memory consumption

Target Devices

- STM32F4 developer board
  - Cortex-M4F clocked at 168 MHz
  - 192KB of RAM
  - 1MB of flash memory



- Raspberry Pi 2
  - Cortex-A7 clocked at 900MHz
  - 1GB RAM

Benchmarks

- SunSpider 1.0.2
  - https://webkit.org/perf/sunspider/sunspider.html
  - Total of 26 test cases
  - Because of memory limitations, JerryScript can only run 19 test cases
- Ubench
  - https://github.com/WebKit/webkit/tree/master/PerformanceTests/SunSpider/tests/ubench
  - Total of 9 test cases
  - All test run with JerryScript

Average Speedup Computation

- EngineA VS EngineB: Nx (M%) faster/slower is computed as follows
- For test i: a_i = Result_i(EngineA) / Result_i(EngineB)
- The geomteric mean is computed from all a_i values
  - avg = $\sqrt[n]{a_1\ a_2\ a_3 \ldots .a_n}$
- If avg > 1 EngineB is b times faster, and N = avg
- If avg < 1 EngineB is 1/avg times slower, and N = 1/avg
- M = (N-1) * 100

Memory Measurement

Measurement Methods

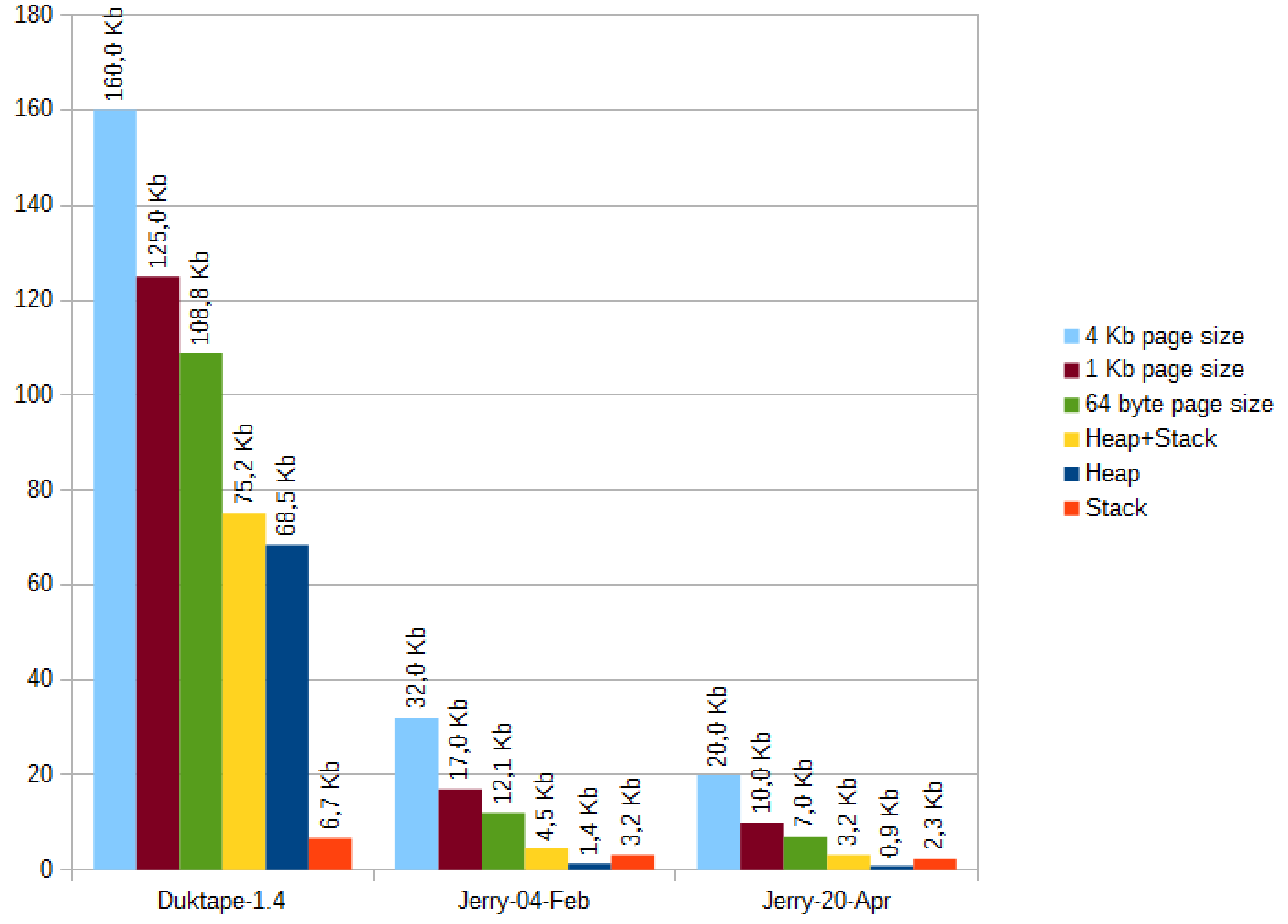
- Memory can be measured in several ways, but none of them is perfect
- Peak heap (malloc) memory consumption
  - excludes allocator, stack and global data memory consumption
- Writable pages allocated by a process (RSS):
  - Results depend on page size, since some pages are only partially used
  - A process may allocate a large memory block but does not use it

Selected Measurement Methods

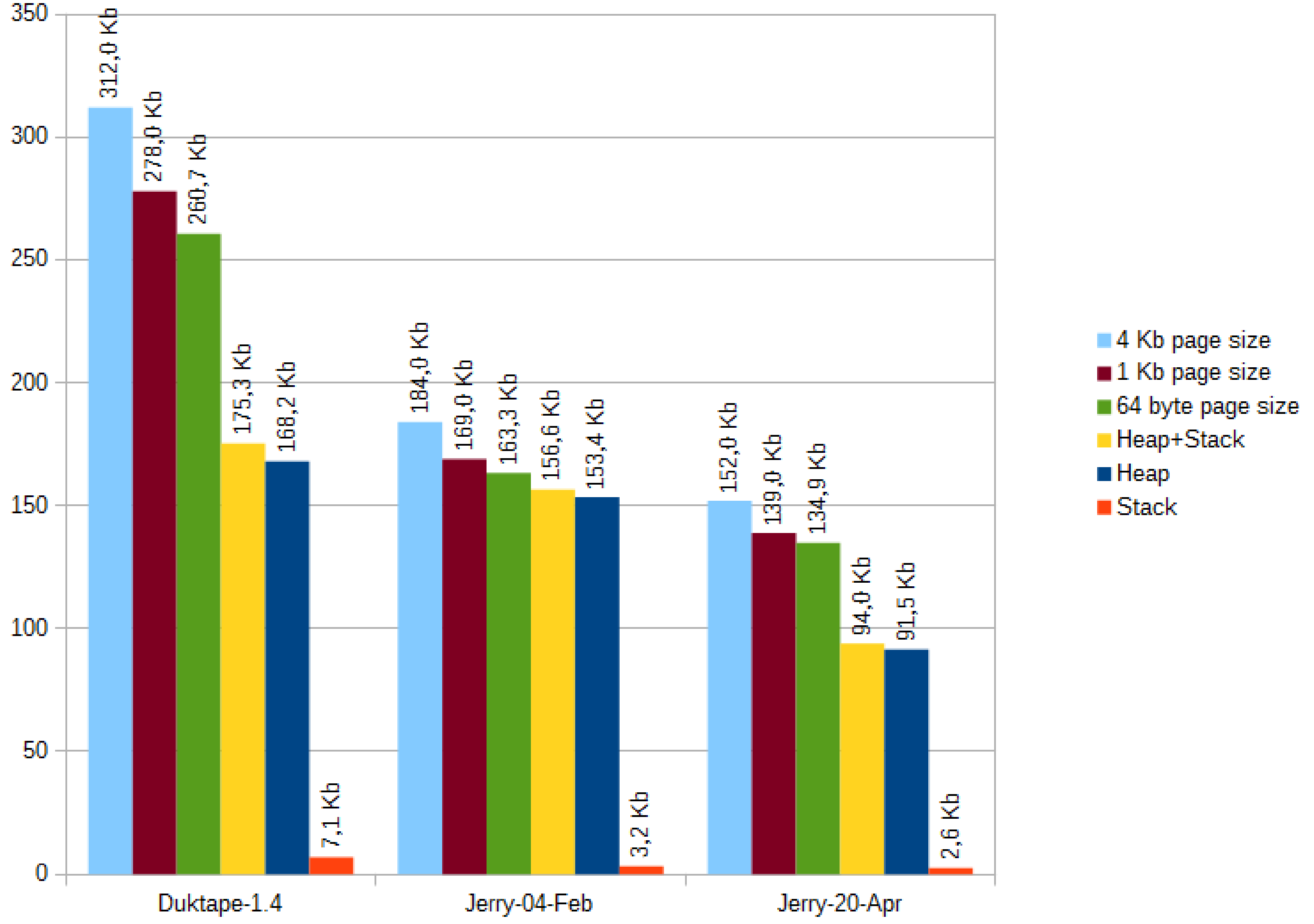
- Peak number of written pages with page size of 64 byte, 1 Kbyte, and 4 Kbyte
  - Measured by Valgrind Heimdall tool
  - Converted to Kbyte for easier comparison
- Peak heap memory consumption
  - JerryScript: Memstats
  - Duktape: Logging allocator
- Peak stack usage
  - Modified main() function

Test: bitops-3bit-bits-in-byte

Samsung Open Source Group

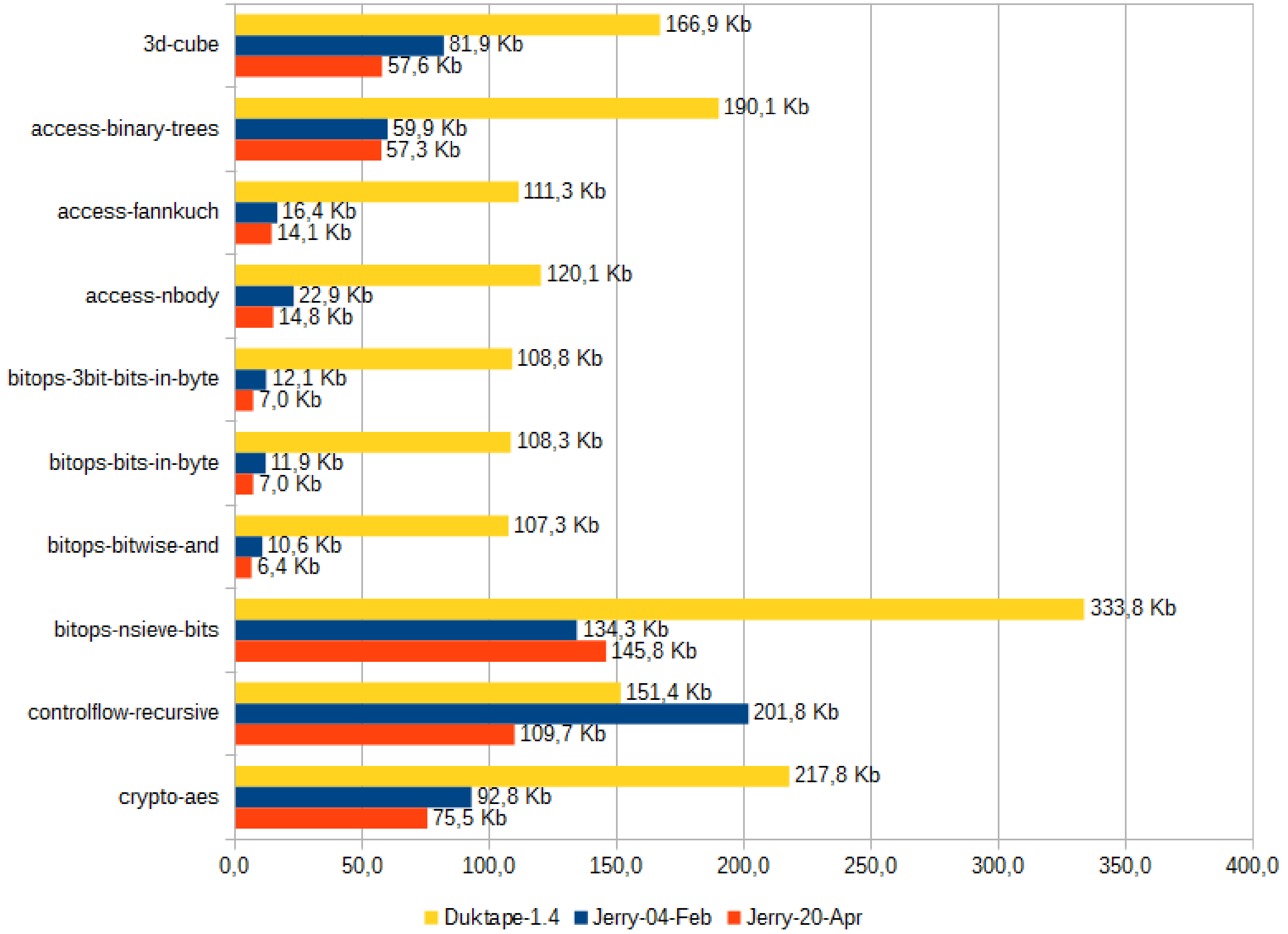


Test: string-base64

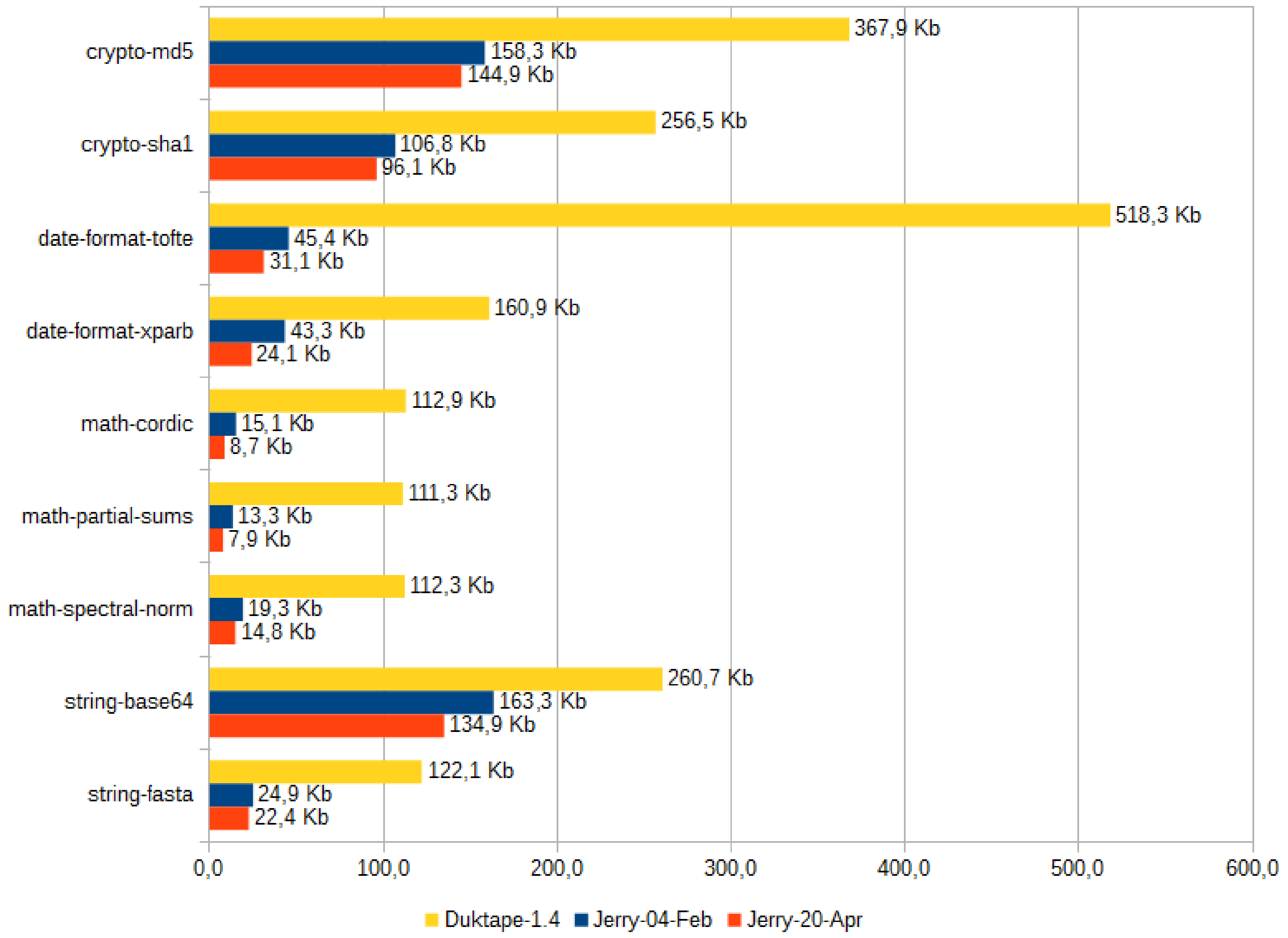


Results With 64 Byte Page Size

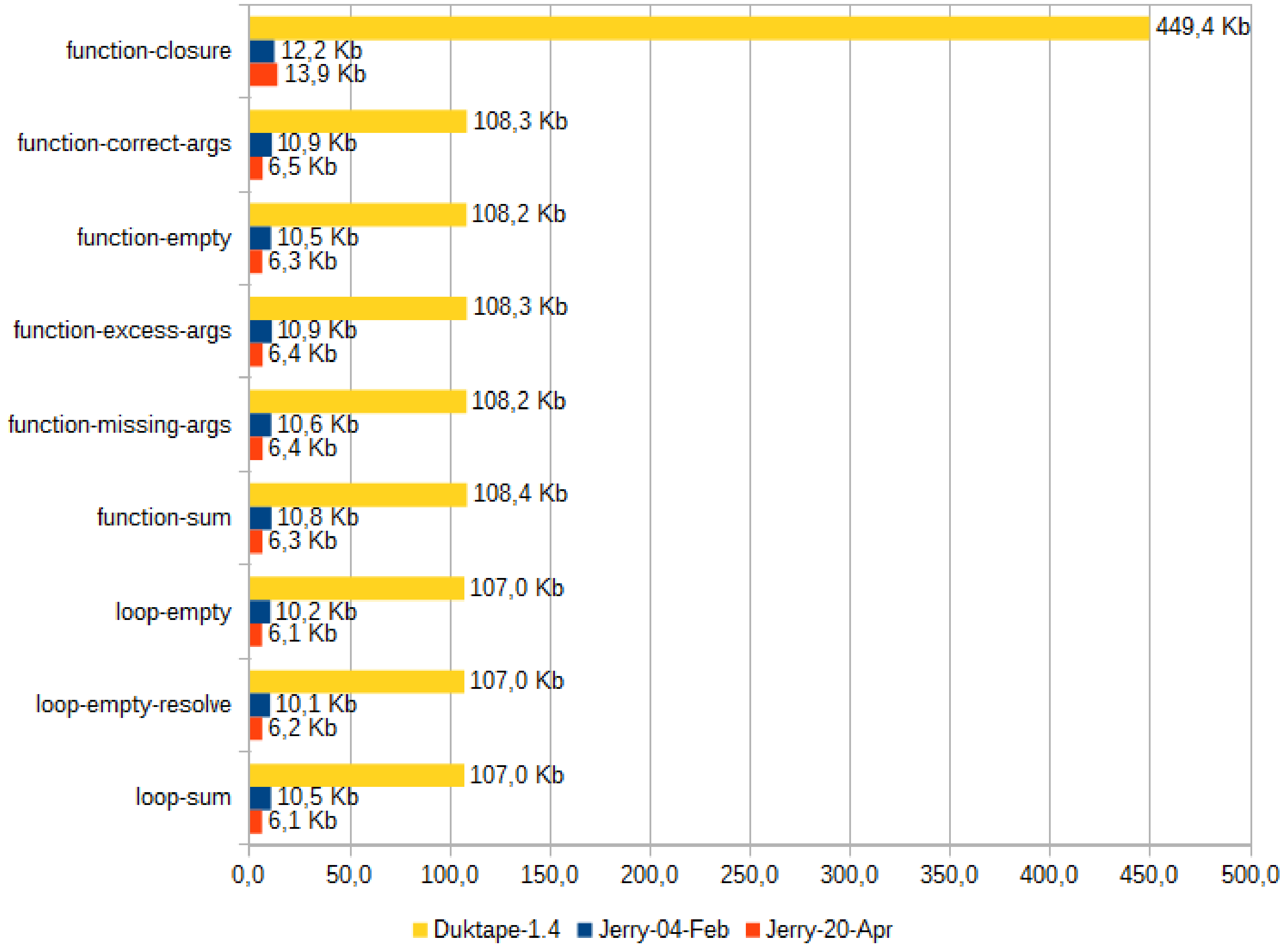
SunSpider Mem. Rpi2, 64 byte Pages



SunSpider Mem. Rpi2, 64 byte Pages (2)



Ubench Mem. Rpi2, 64 byte Pages

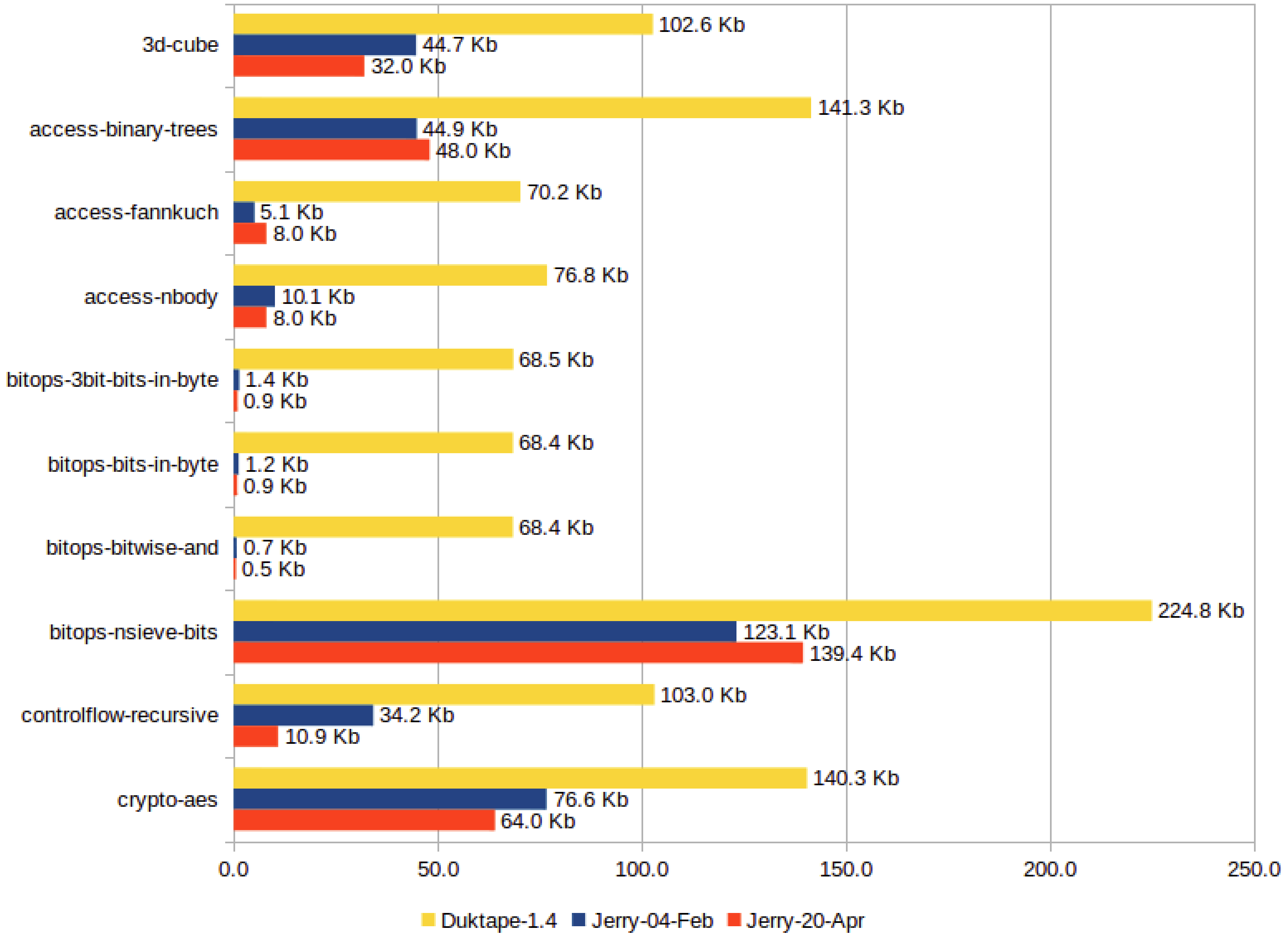


Summary of 64 Byte Page Size

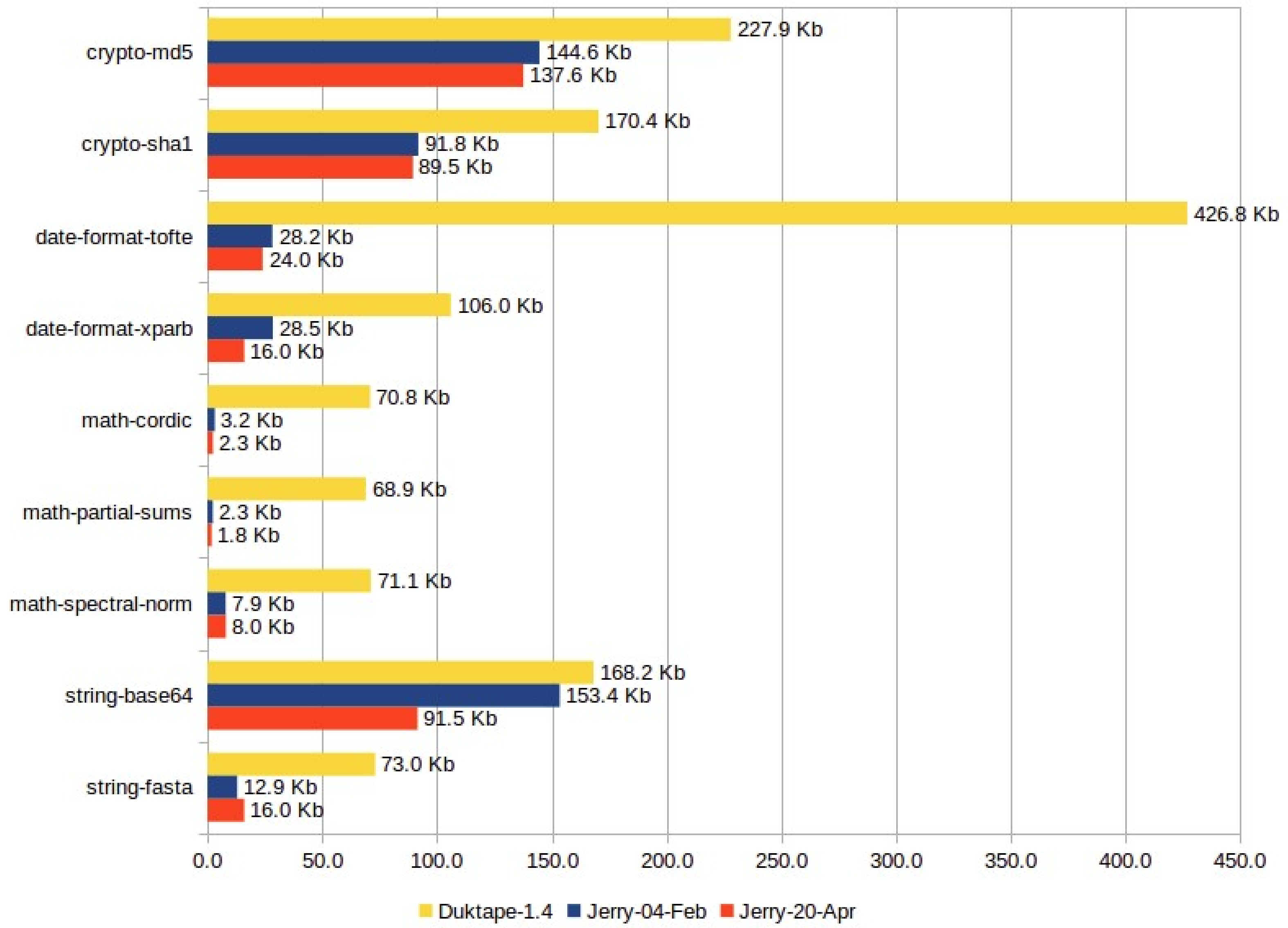
- SunSpider
  - Duktape 1.4 VS Jerry-04-Feb: 76% less memory
  - Duktape 1.4 VS Jerry-20-Apr: 82% less memory
  - Jerry-04-Feb VS Jerry-20-Apr: 27% less memory
- Ubench
  - Duktape 1.4 VS Jerry-04-Feb: 91% less memory
  - Duktape 1.4 VS Jerry-20-Apr: 94% less memory
  - Jerry-04-Feb VS Jerry-20-Apr: 36% less memory

Heap Usage

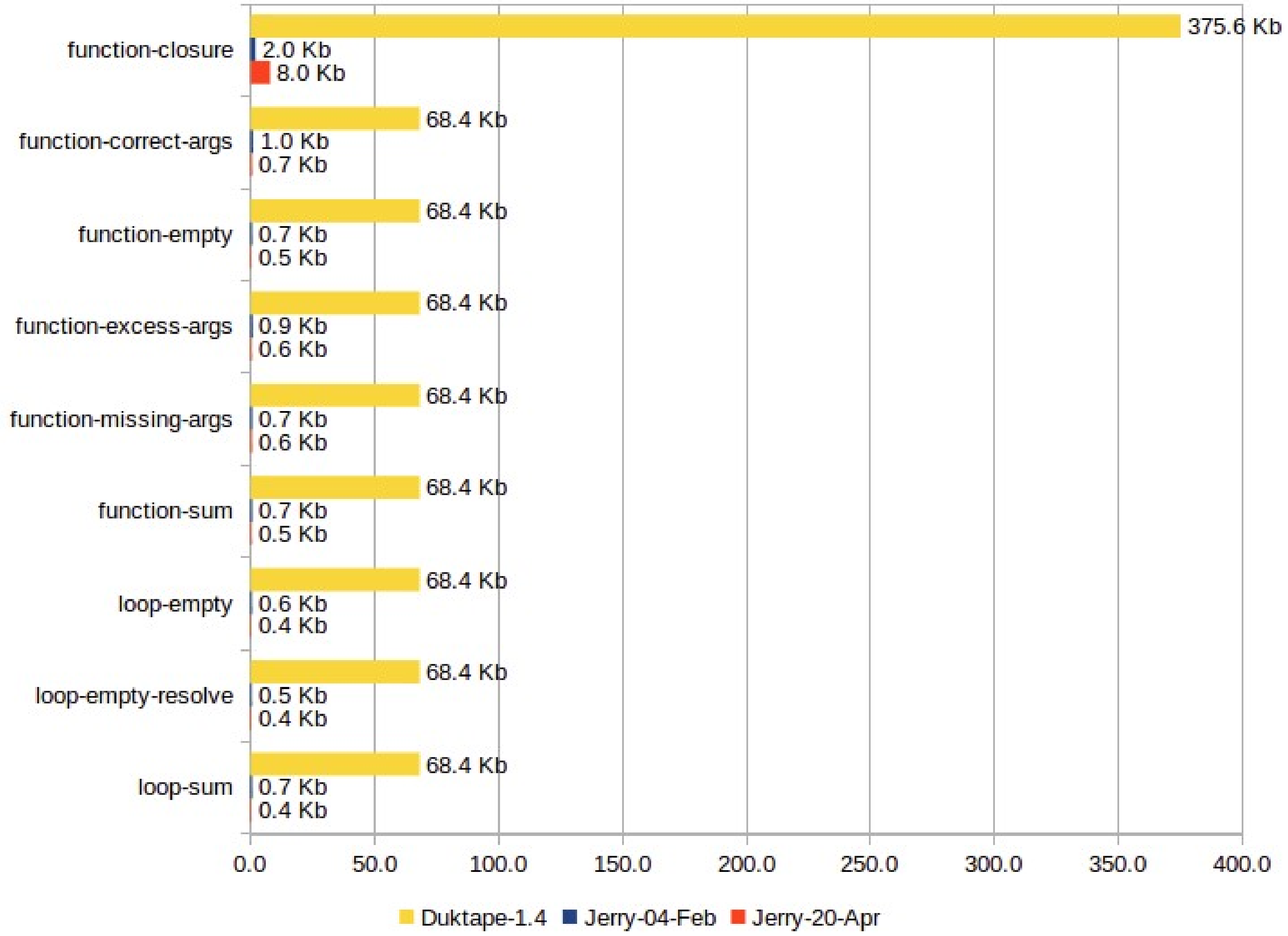
SunSpider Heap Usage on RPi2



SunSpider Heap Usage on Rpi2 (2)



Ubench Heap Usage on Rpi2 (2)

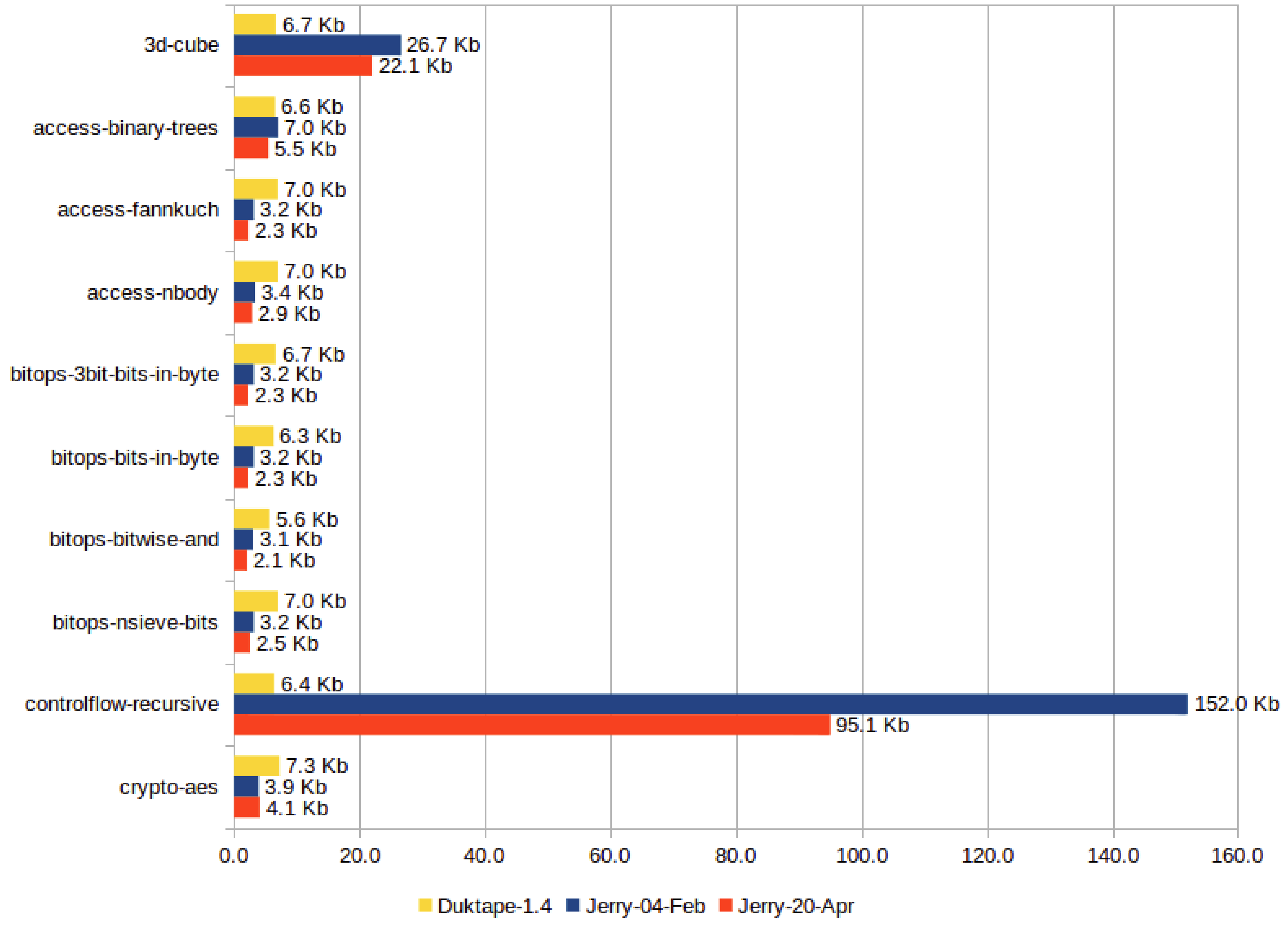


Summary of Heap Usage

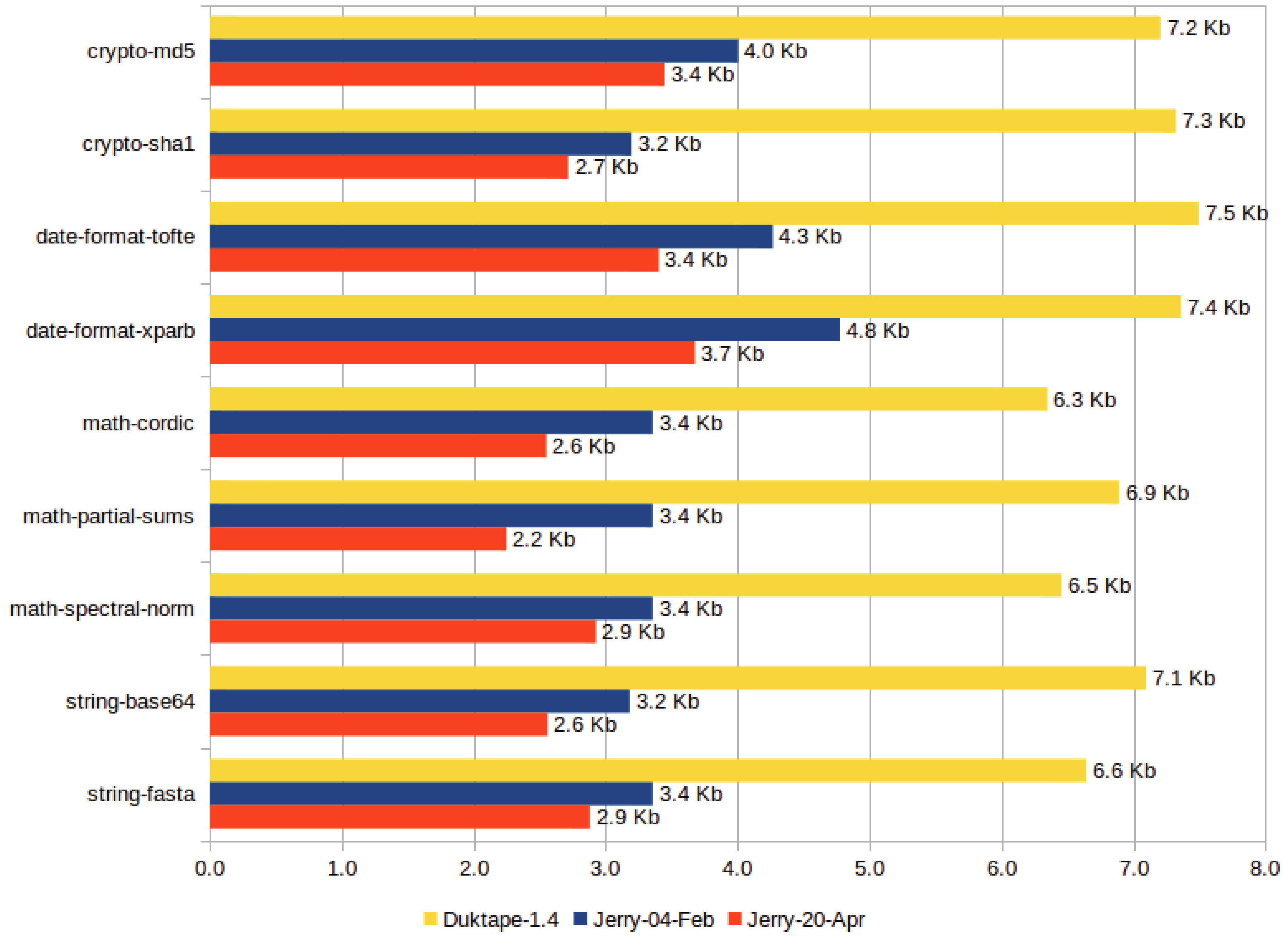
- SunSpider
  - Duktape 1.4 VS Jerry-04-Feb: 85% less heap memory
  - Duktape 1.4 VS Jerry-20-Apr: 88% less heap memory
  - Jerry-04-Feb VS Jerry-20-Apr: 18% less heap memory
- Ubench
  - Duktape 1.4 VS Jerry-04-Feb: 99% less heap memory
  - Duktape 1.4 VS Jerry-20-Apr: 99% less heap memory
  - Jerry-04-Feb VS Jerry-20-Apr: 13% less heap memory

Stack Usage

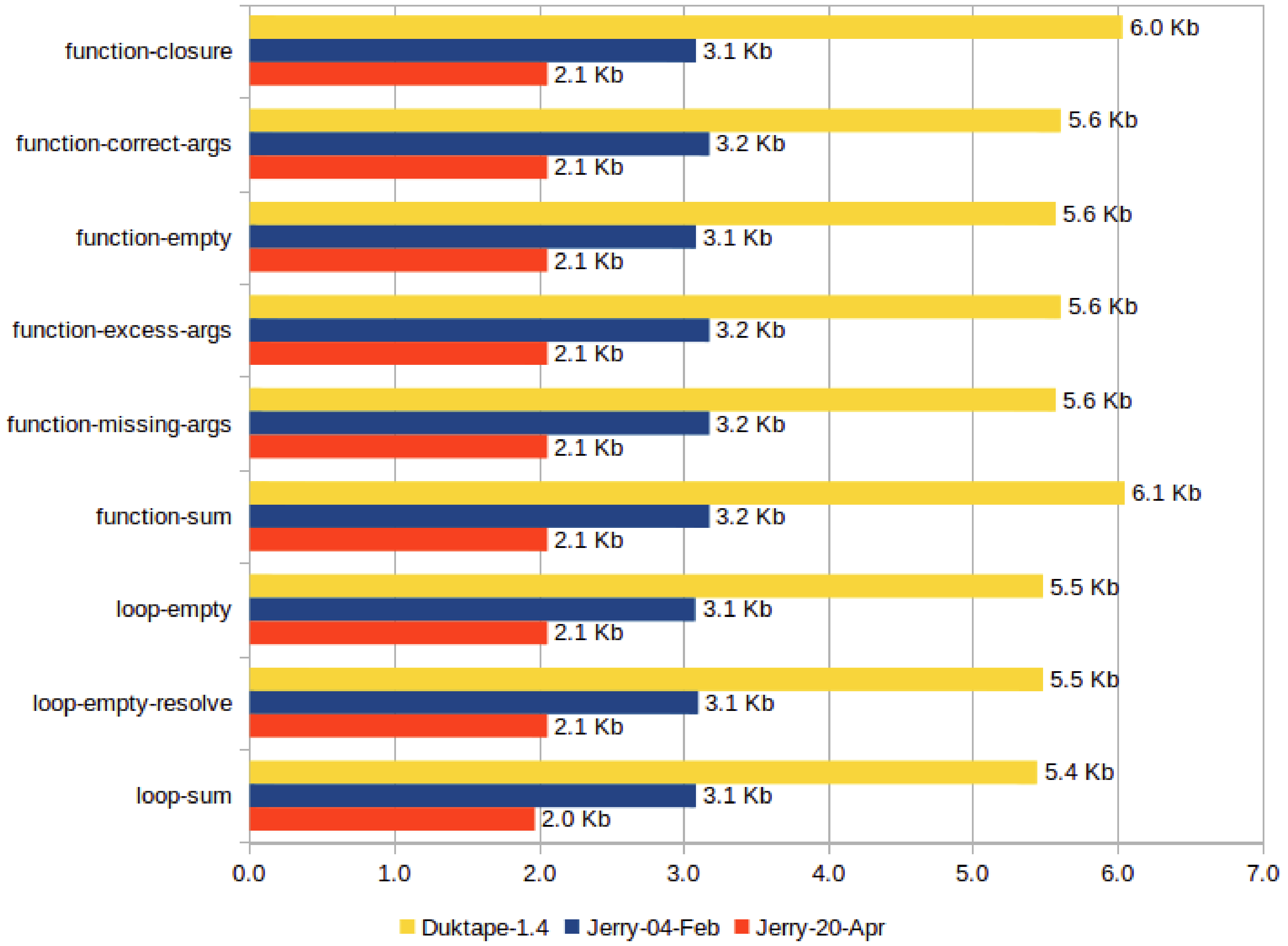
SunSpider Stack Usage on RPi2



SunSpider Stack Usage on RPi2 (2)



Ubench Stack Usage on RPi2

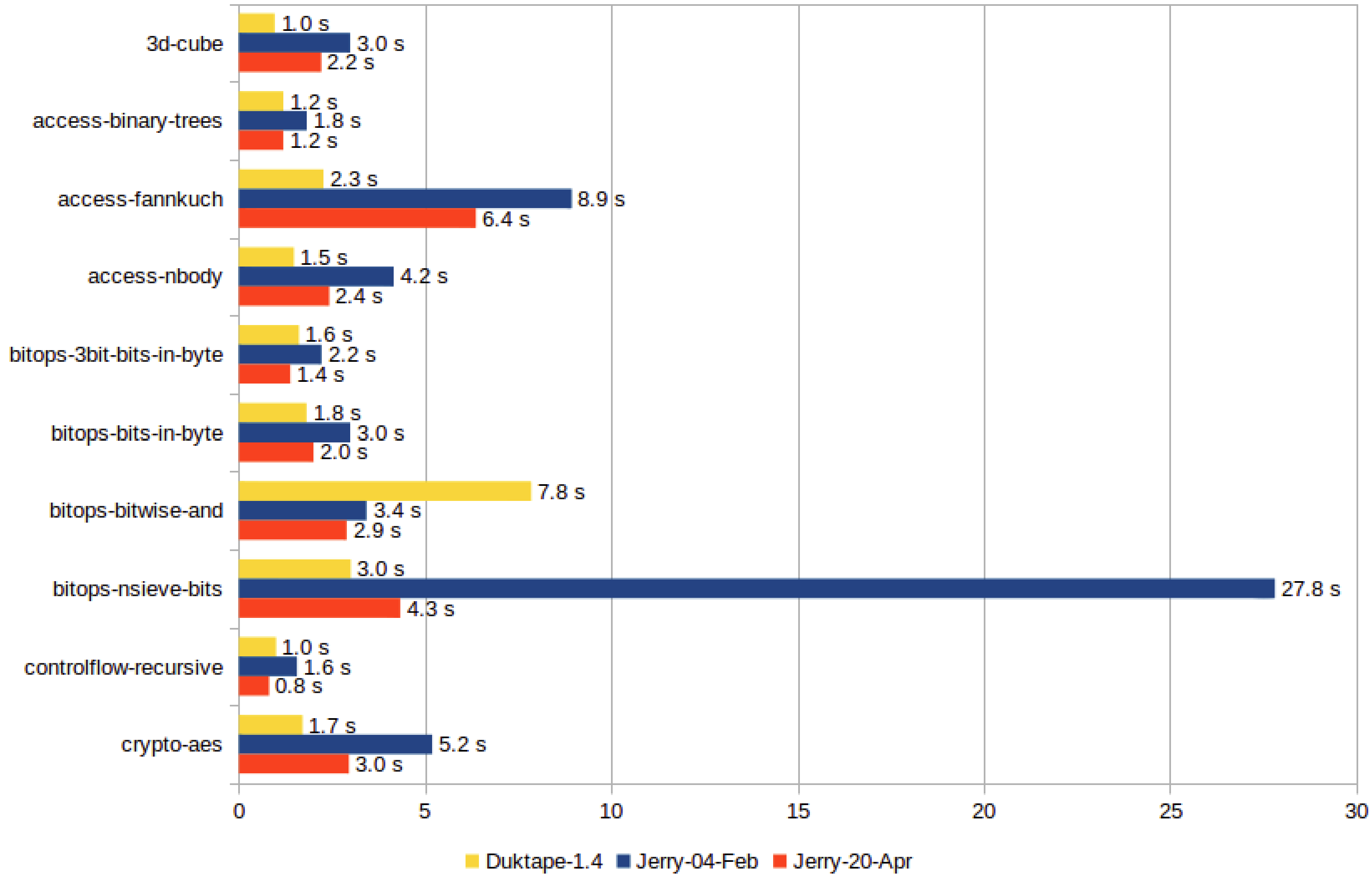


Summary of Stack Usage on Rpi2

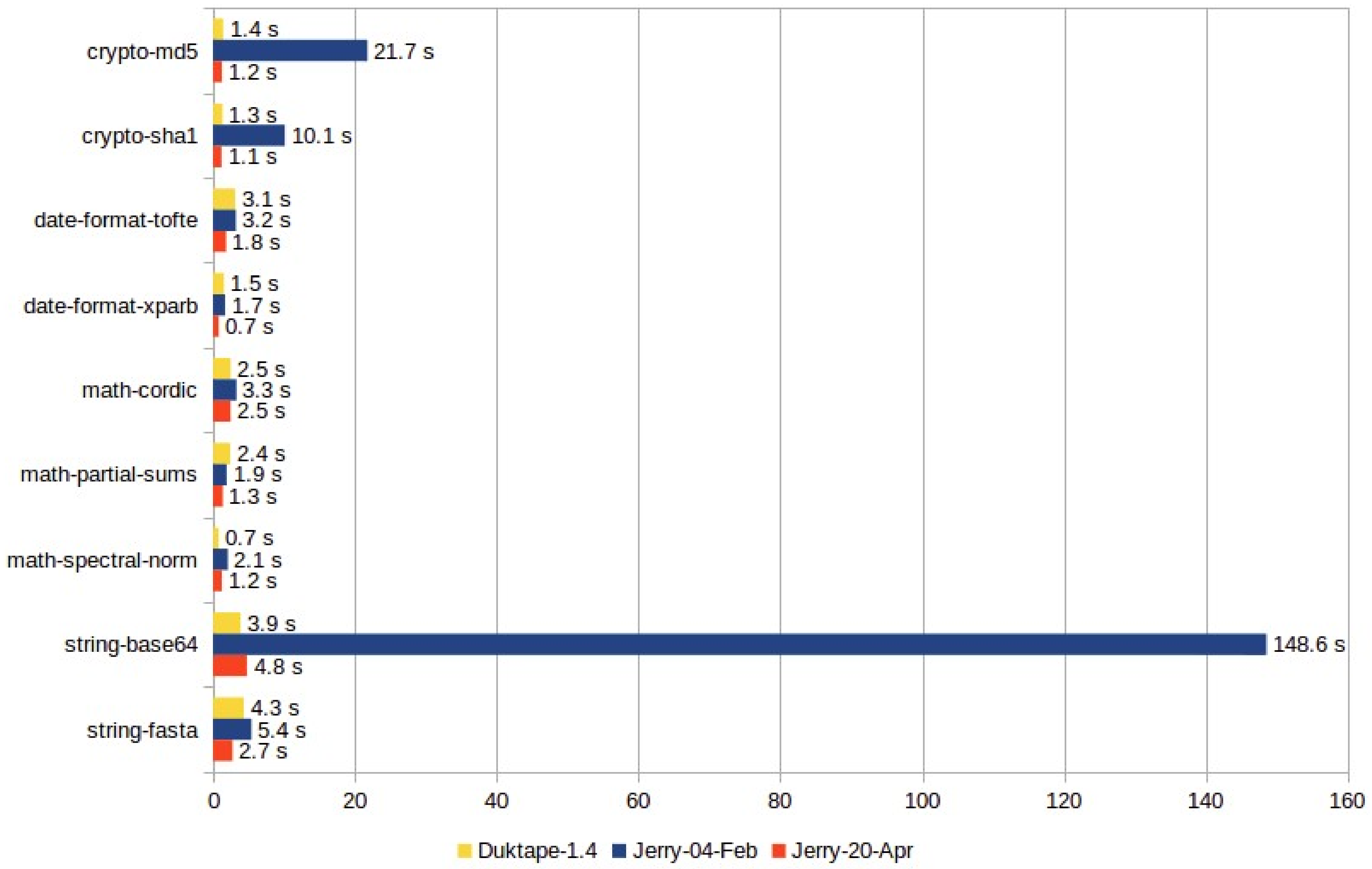
- Duktape uses a fixed stack
  - JavaScript functions use heap for recursion
  - Disadvantage: a large amount of heap is reserved for ECMAScript call stack
- JerryScript stack usage is reduced by 21% on SunSpider and by 34% on Ubench

Performance Comparison

SunSpider Performance on RPi2



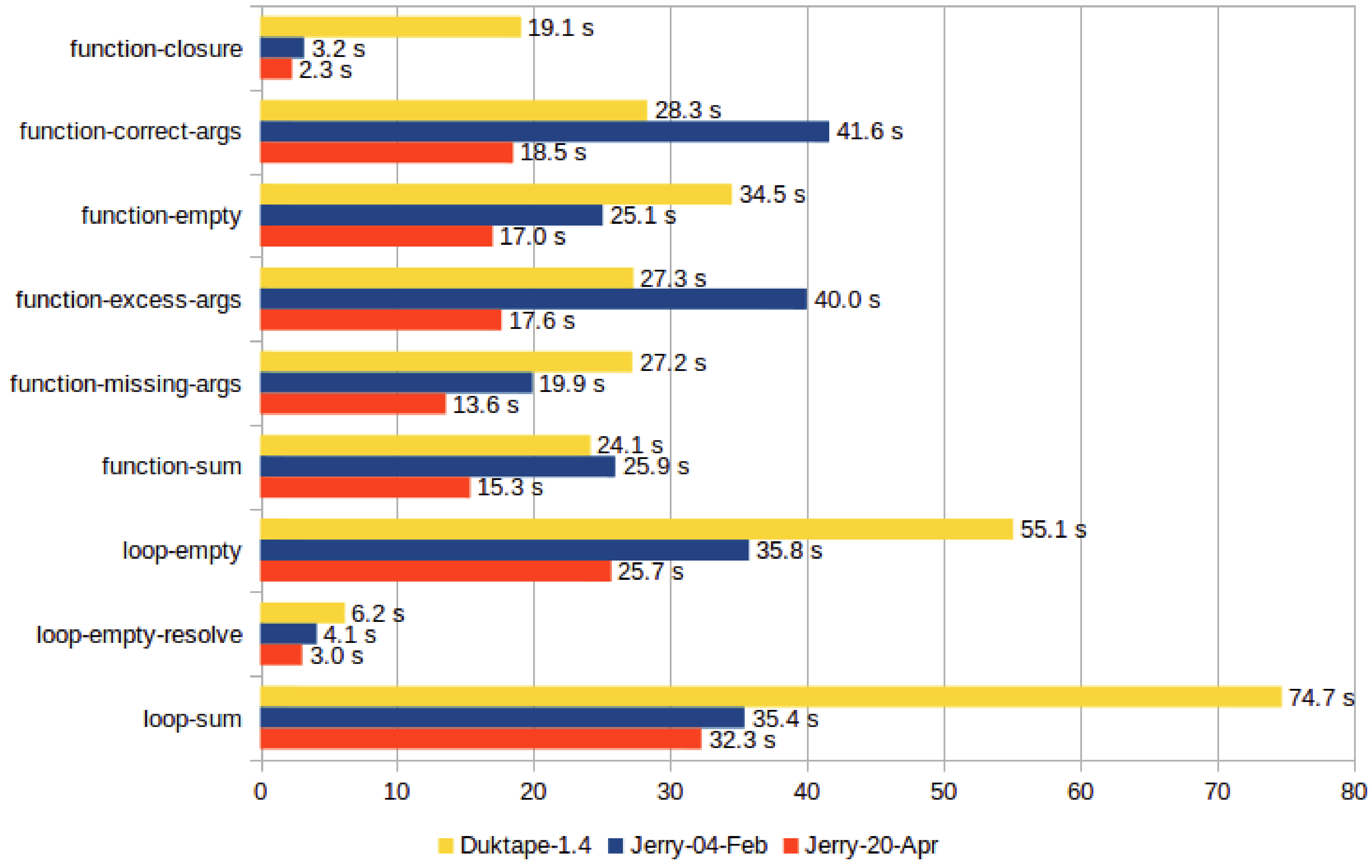
SunSpider Performance on RPi2 (2)



SunSpider RPi2 Statistics

- Speedup
  - Duktape 1.4 VS Jerry-04-Feb: 2.52x (152%) slower
  - Duktape 1.4 VS Jerry-20-Apr: 1.01x (1%) slower
  - Jerry-04-Feb VS Jerry-20-Apr: 2.5x (150%) faster
- 9 tests are faster with Jerry-20-Apr
- 8 tests are faster with Duktape 1.4
- 2 tests have the same runtime

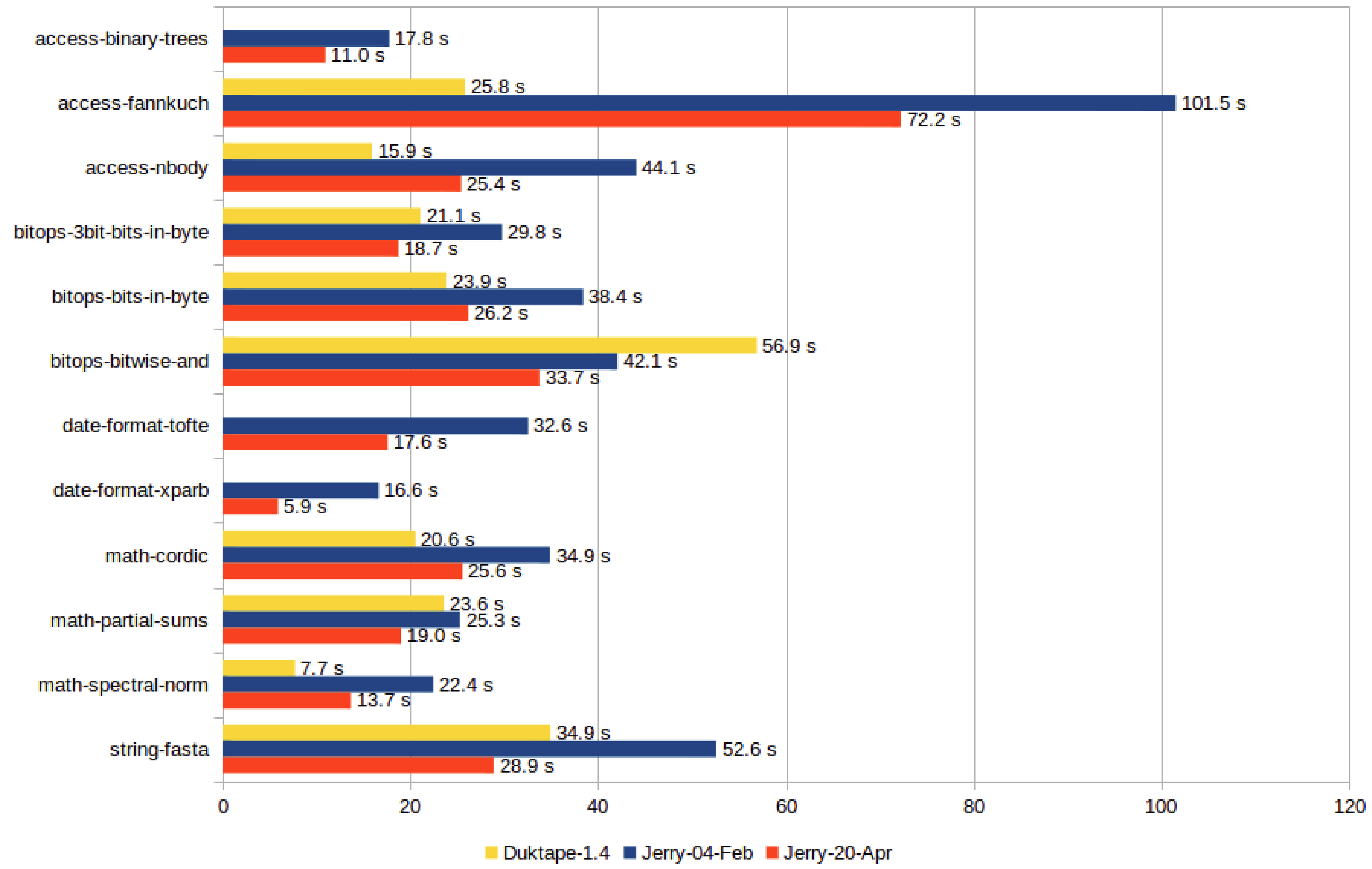
Ubench Performance on RPi2



Ubench RPi2 Statistics

- Speedup
  - Duktape 1.4 VS Jerry-04-Feb: 1.42x (42%) faster
  - Duktape 1.4 VS Jerry-20-Apr: 2.21x (121%) faster
  - Jerry-04-Feb VS Jerry-20-Apr: 1.55x (55%) faster
- All tests are faster with Jerry-20-Apr

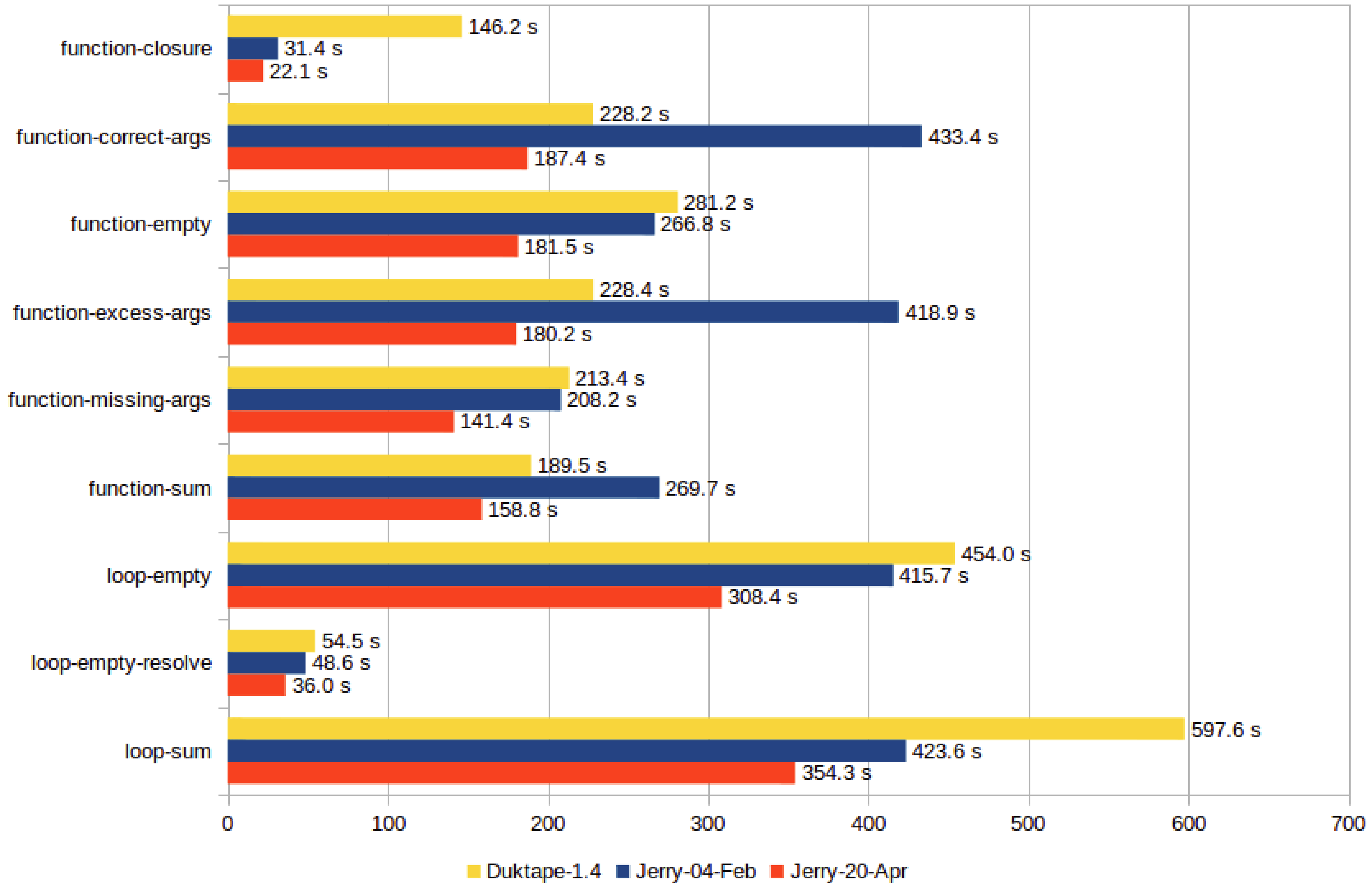
SunSpider on STM32F4



SunSpider STM32F4 Statistics

- Duktape comparisons only includes those tests whose run with Duktape
- Speedup
  - Duktape 1.4 VS Jerry-04-Feb: 1.73x (73%) slower
  - Duktape 1.4 VS Jerry-20-Apr: 1.15x (15%) slower
  - Jerry-04-Feb VS Jerry-20-Apr: 1.61x (61%) faster
- Speedup when these tests are selected on Rpi2
  - Duktape 1.4 VS Jerry-04-Feb: 1.38x (38%) slower
  - Duktape 1.4 VS Jerry-20-Apr: 1.01x (1%) slower
  - Jerry-04-Feb VS Jerry-20-Apr: 1.37x (37%) faster

Ubench on STM32F4



Ubench STM32F4 Statistics

- Speedup
  - Duktape 1.4 VS Jerry-04-Feb: 1.06x (6%) faster
  - Duktape 1.4 VS Jerry-20-Apr: 1.68x (68%) faster
  - Jerry-04-Feb VS Jerry-20-Apr: 1.57x (57%) faster
- All tests are faster with Jerry-20-Apr

Binary Size Comparison

Binary Size Comparison

- ARM 32 bit Thumb-2 stripped binary size
  - Duktape 1.4: 204,428 (non-static)
  - Jerry-04-Feb: 200,668 bytes (static)
  - Jerry-20-Apr: 174,988 bytes (static)
- Engines support reduced modes where certain features (e.g. regular expressions) can be disabled to reduce binary size

Summary

Summary

- JerryScript consumes considerably less memory than Duktape
- JerryScript and Duktape has similar performance on Raspberry Pi 2
- JerrryScript has a bit lower performance on STM32F4 than Duktape

Thank you.