KONSULTANER®

Better performance with WebWorkers

Konsultaner GmbH & Co. KG
Lugturmstr. 45
01809 Heidenau

Tel: + 49 (0) 3 529 / 52 902 40
Fax: + 49 (0) 3 529 / 52 902 44

www.konsultaner.de
info@konsultaner.de

About us

KONSULTANER®

» Web technology agency

» Java WebSocket-Server + sub protocol = connectanum

» Several JavaScript Modules

» PHP-CMS

» Custom BI-Websoftware

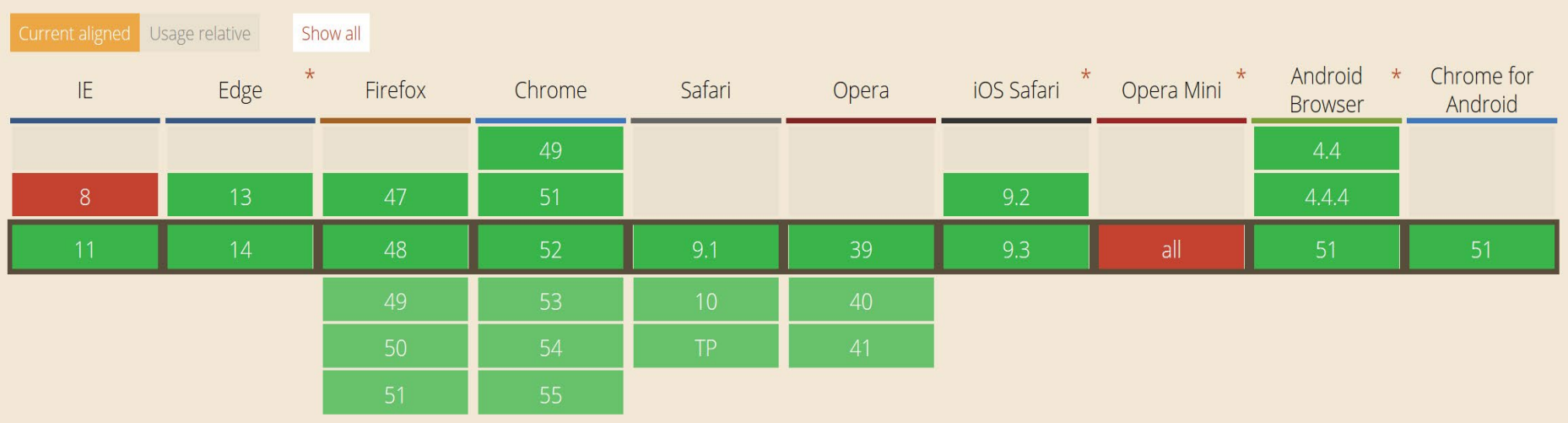
» Apps

What are WebWorkers

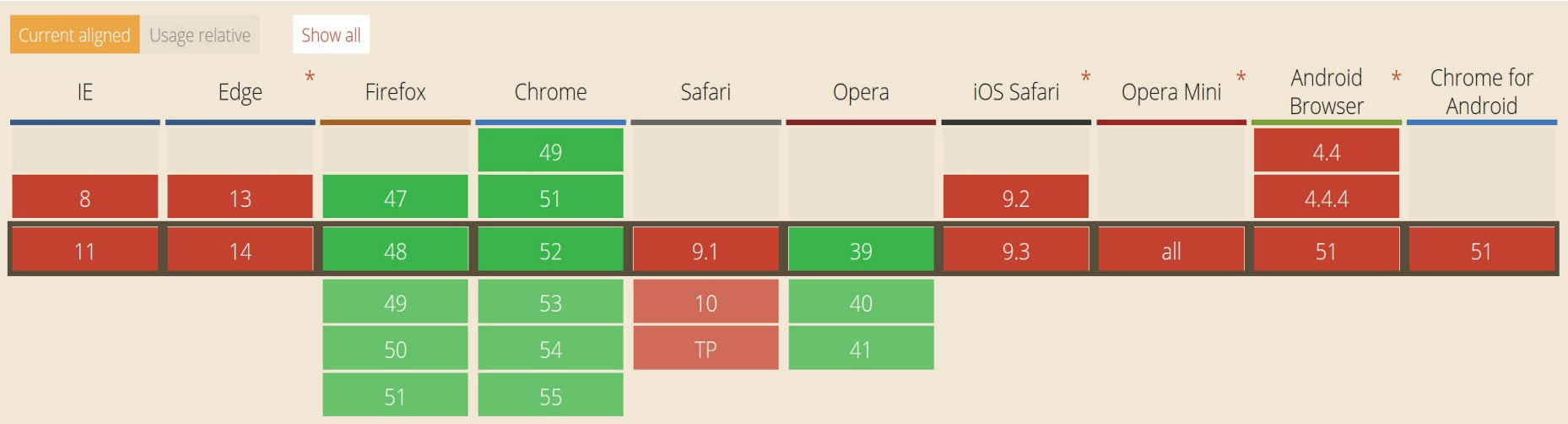
» Isolated Threads in JavaScript

Can i use

caniuse.com/#feat=webworkers



caniuse.com/#search=shared



Why WebWorkers

Why

» Prevent ui-thread blocking
» Enhance performance
» (With) concurrent computation

Easy multithreading

» No locking, no synchronization, no atomics
» Immutable
» Comes with simple message system
» But no shared memory

Implementation

index.js

```
var worker = new Worker('doWork.js');
worker.addEventListener('message', function(e) {
   console.log('Worker said: ', e.data);
}, false);
worker.postMessage('Hello World');
```

doWorker.js

```
self.addEventListener('message', function(e) {
  self.postMessage(e.data);
}, false);
```

Source: html5rocks.com

Easy to solve problems

- XML-parsing
  - Just use one of a milltion tools to compensate it
  - i.e. https://github.com/tobiasnickel/tXml
  - Angular 2 runs in a WebWorker
- The missing load event
  - Write your own
  - When all your initialization (i.e. server requests) is done, send a ready message
- Serialization is really fast, everything important works as expected
  - Bigger data can be passed as ArrayBuffer in the transferList
  - BUT: Functions are not serializable!
    - I know eval, but I wouldn't use it
    - You can add functionality during your build process

Non-trivial problems

- WebWorker only provide a simple asynchronous message send and receive logic

- Nice to have features
  - Multiple threads
  - RPC calls
  - Promises
  - Progress events (thread.js)
  - If you prefer the actor pattern (akka.js)

Serialization

» Is serialization too slow?

» http://runspired.github.io/webworker-performance/

» Chrome52 on this Laptop

» ~14kbyte

String => 133ms
{String[]} => 238ms
{String[]} as transferable => 290ms

» ~35kbyte

String => 303ms
{*} => 968ms
{*} => 611ms

Computing

- Google says
  - In many cases you can move pure computational work to Web Workers, [...] Data manipulation or traversal, like sorting or searching, are often good fits for this model, as are loading and model generation

- But the worker has extra
  - Source load time
  - Creation time
  - Data transfer time

Source: https://developers.google.com/web/fundamentals/performance/rendering/optimize-javascript-execution

Computing

» Load the Worker once, not for each computation

» Measure performance
Everything under 3-4ms is not worth thinking about [google]

» Use worker when ever javascript blocks the ui-thread

Example with Ionic

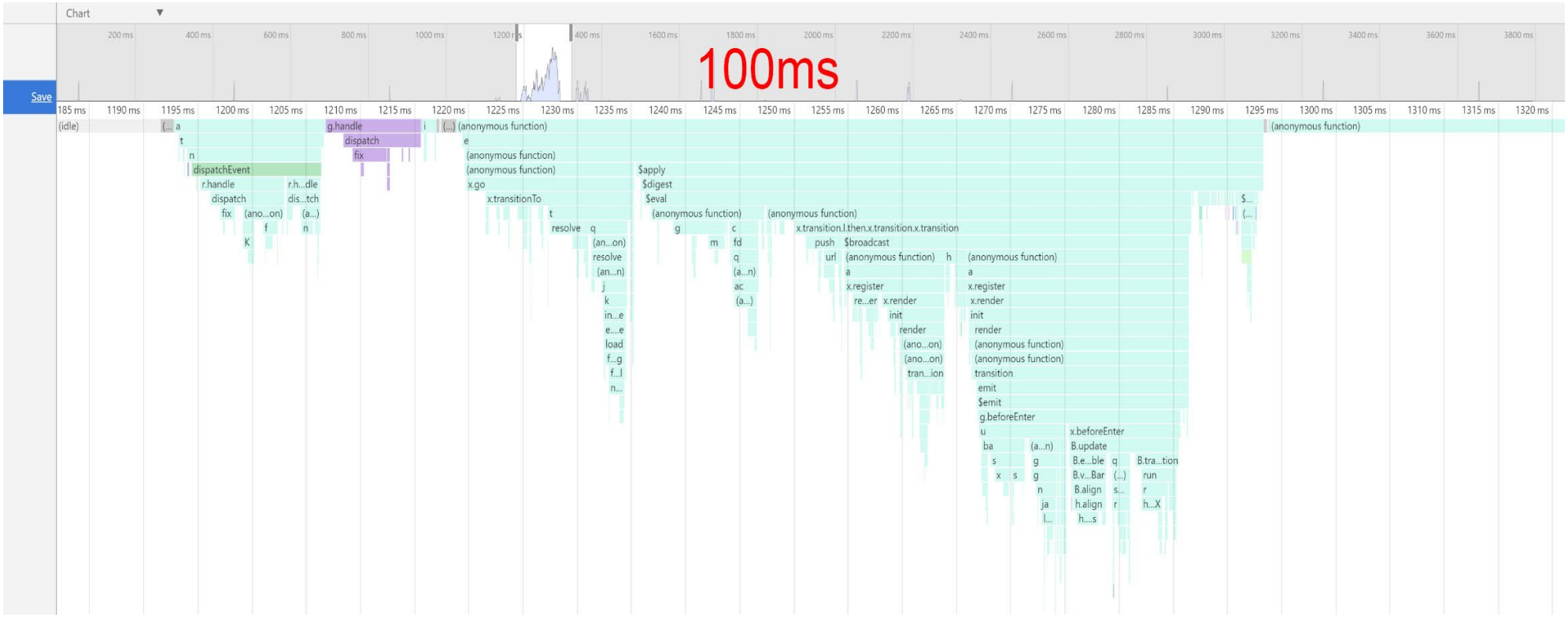
» Ionic app, page transition optimization
  » Transition from dashboard to a heavy list view
  » Ng-repeat is very slow
  » Manual DOM creation with angular services not too fast

» 3-4 Seks processing time on samsung galaxy S5 with crosswalk to finish the transition with ng-repeat

» ~1Sek processing time in browser

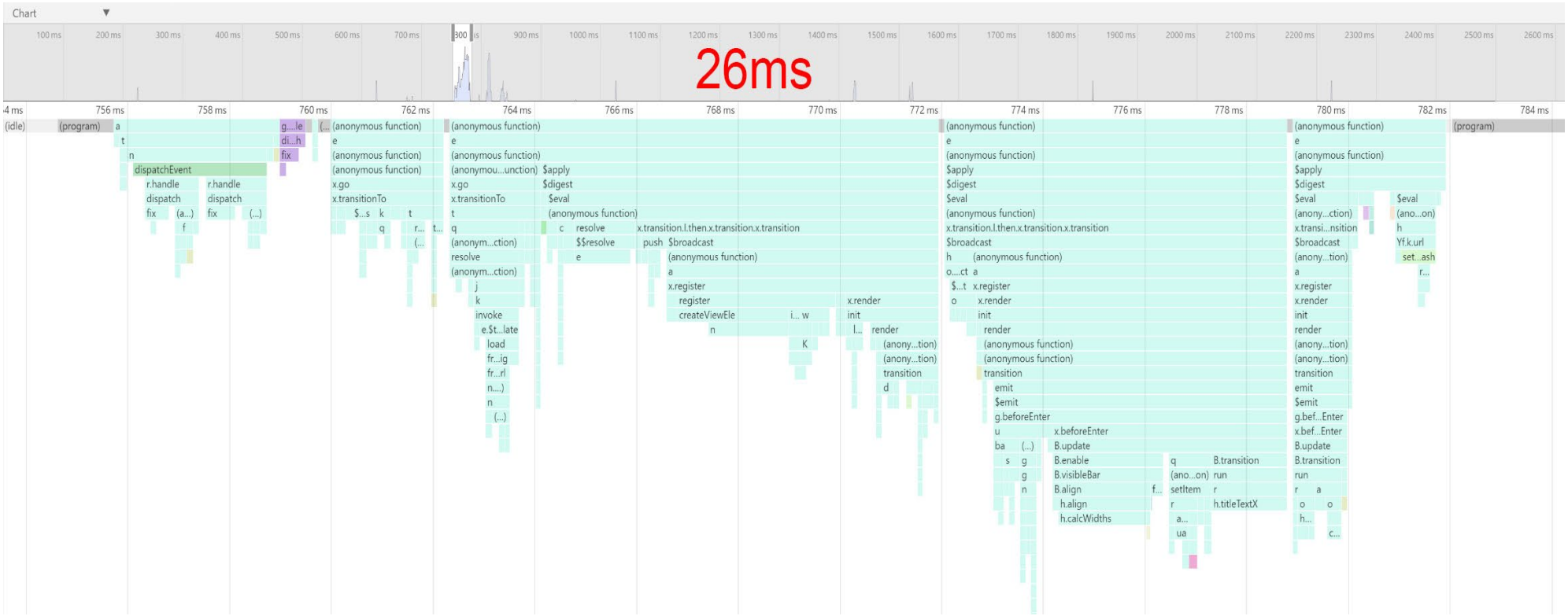
Profile not optimized

» DOM generated in directive



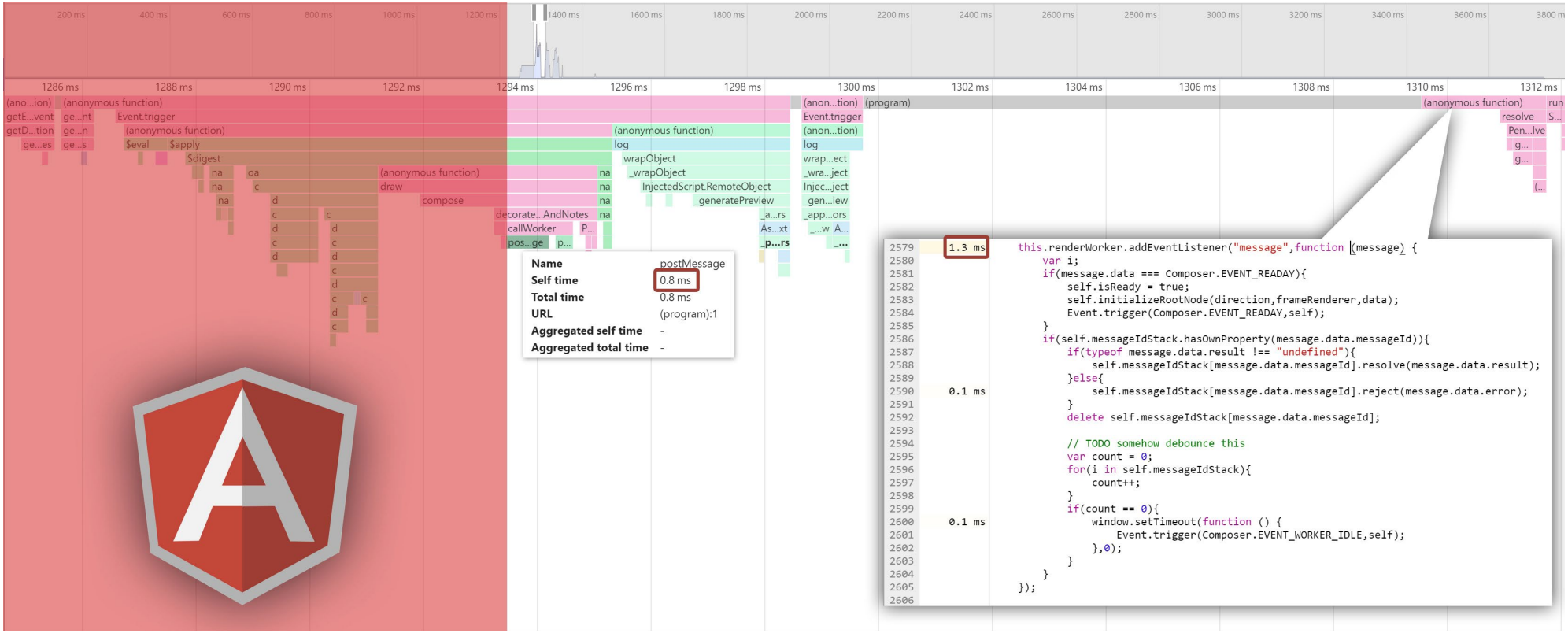
Profile not optimized

» DOM generated in worker



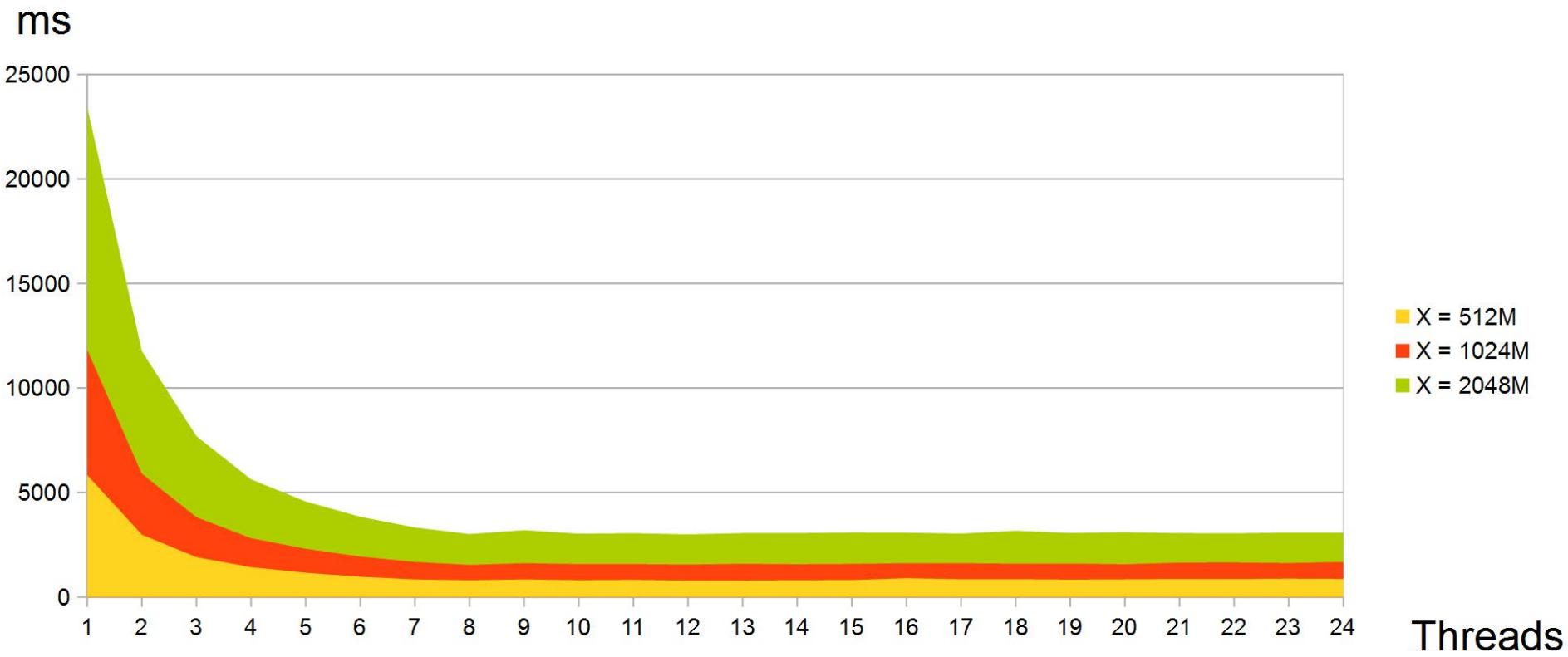
Example Calendar

» Approves performance of ionic example (~26ms)



Multicore

» Calculate 2^x mod 97777 using n threads
on a Intel i7 6700 Quadcore (8 logical cores) @2.6GHz



Source: http://pmav.eu/stuff/javascript-webworkers/

Thank You!