ohann Sebastian Bach“

Stefan | Home

Johan... | Timeline | Recent

Like

Create Page

Photos | Likes | Events

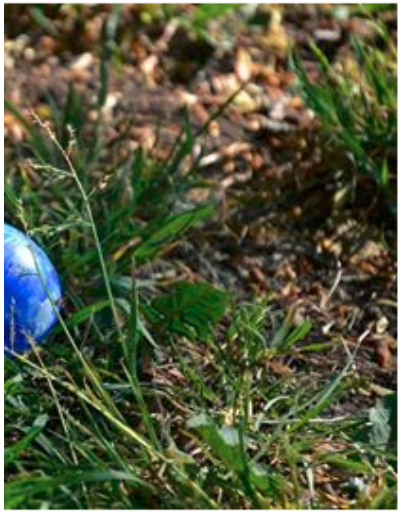
Recent
2010s
Founded

Highlights

ig „Johann Sebastian Bach“?

hann Sebastian Bach“

n!



Invite Your Friends to Like This Page — See All

Type a friend's name... — Invite

[redacted] Invite

[redacted] Invite

[redacted] Invite

Reviews — See All

What do you think of Musikschule Leipzig „Johann Sebastian Bach“?

[redacted]
Die "Schneekönigin-Das Musical" kommt am 04.02.2013 um 17Uhr in die Theaterfabrik. Sichert euch jetzt noch Tickets unter… See More
Like · over a year ago ·

[redacted]
Hallo.
Ich bin Ezchial Nikiema aus Burkina Faso. Ich bin Schlagzeuger.Ich hab gleich nach Ihrer Schule angerufen und hab… See More
Like 2 · over a year ago ·

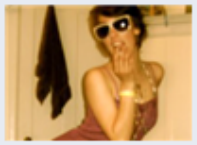
[redacted]

Sponsored

Megastar sucht Nam
migipedia.ch

Pro
blo
ihm

Singles auf Facebook
Zoosk



Alle
seh
ann
Par

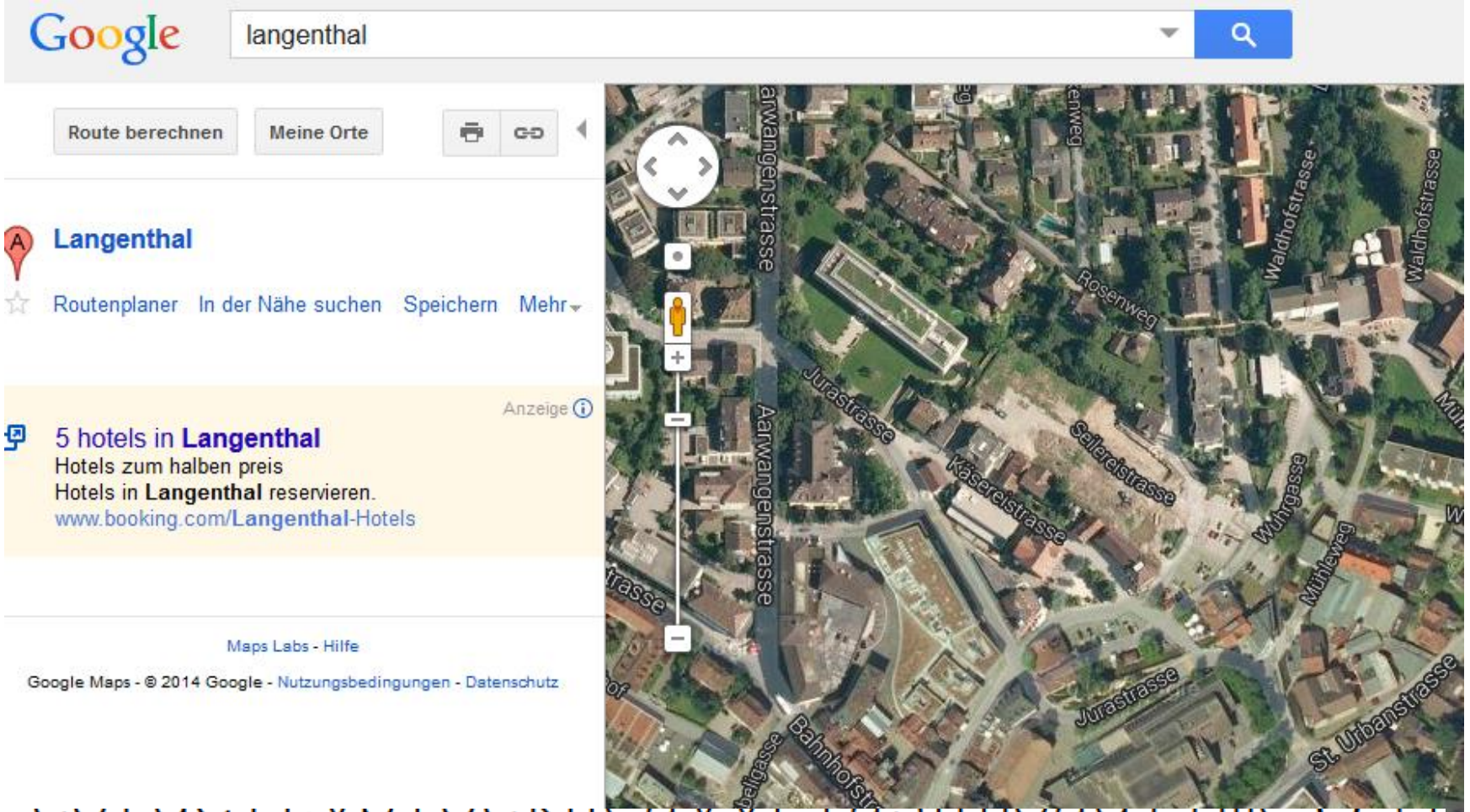
Use Now · 5,000,0

ZAZ live @ MAAG Hal

ZA
Mai
ein
Sou

Tuesday, May 13 at 8:

Join · 98 people are




```
<script type="text/javascript">windo

)ffset = calculateOffsetTop(promoImag
```

River Trail: A Path to Parallelism in JavaScript

Stephan Herhut et al. (Intel Labs), 2013

Stefan Zurfluh, CCC Seminar Talk

May 7, 2014

Web Programming / JavaScript *Today*

- Used more and more for computationally complex, large-scale applications
- The only universal web browser programming language
- Mostly sequential

Requirements, Challenges, Goals for RiverTrail

- Safety and security
- Comfortable API
- Generic and hardware independent
- Dramatic performance improvements

API Components

ParallelArray data type

Parallel **methods**: map, combine, reduce, scan, scatter, filter, flatten, partition, get

Elemental functions

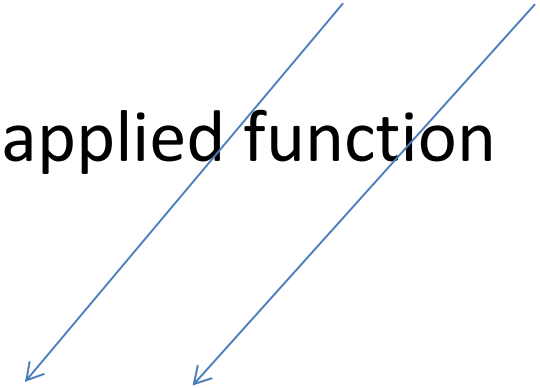
API Components

- ParallelArray
  - Numeric
  - Immutable
  - May be multidimensional
- Parallel methods
  - Compact set of useful, common data-parallel methods
- Elemental function
  - Operates element-wise on parallel arrays
  - Read-only access to global state

Example: Map Function

Element-wise operation on array

```
myArray.map(elementalFunction, arg1, arg2, …)
```



returns new array with applied function

Elemental function:

```
function (element, arg1, arg2, …)
```

```
// Adding one to each element.
var source = new ParallelArray([1,2,3,4,5]);
var plusOne = source.map(function inc(v)
                                { return v+1; });
```

Example: Reduce Function

Reduce a dimension to one element

```
myArray.reduce(elementalFunction, arg1, arg2, ...)
```

returns last element

Elemental function:

```
function (a, b, arg1, arg2, …)
```

should be commutative and associative, as reducing order is arbitrary

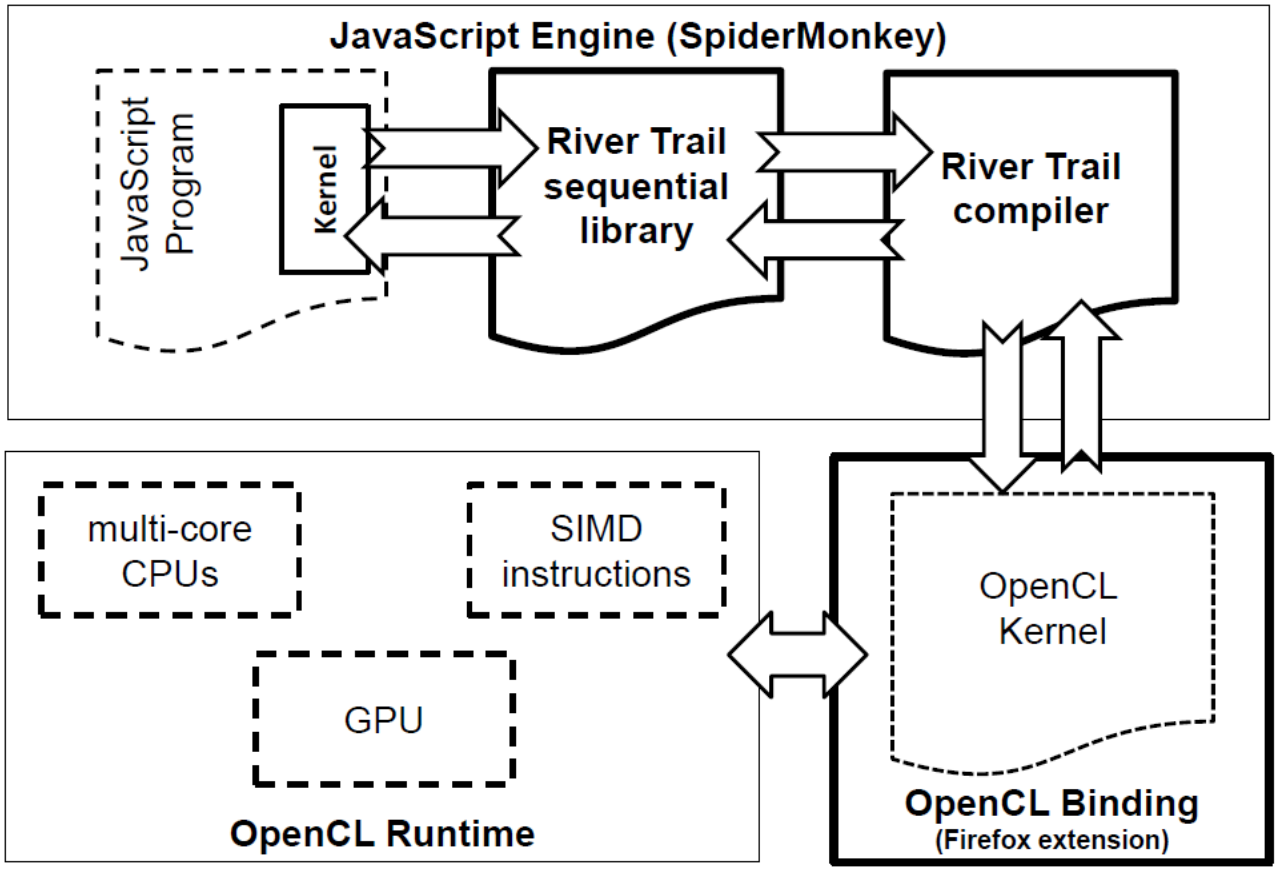
```
// Calculate the sum of the elements
var source = new ParallelArray([1,2,3,4,5]);
var sum = source.reduce(
                function plus(a,b) { return a+b; });
```

Other Functions

- combine
  - like map, but exposes element index instead of element value to elemental function
- scan
  - reduce n times from 0 to i
- scatter
  - element redistribution with indices – similar to reduce from MapReduce
- filter
  - remove elements according to boolean function
- flatten, partition
  - change array dimensions
- get
  - return element

Implementation

There are both *parallel* and *sequential* implementations



Parallel and Sequential Implementations

- All functions have sequential versions
- If *compiler* and *OpenCL* are present: parallel versions of *map*, *combine*, and *comprehension constructor* are used instead
- Parallel version := elemental function translated from JavaScript to OpenCL

River Trail Compiler

JavaScript

- High level
- Dynamically typed
- Implicit memory mgmt / GC
- (restricted) Shared memory model

OpenCL

- Hardware specific
- Statically typed (C-like)
- Explicit memory (de)allocation
- Distributed memory model

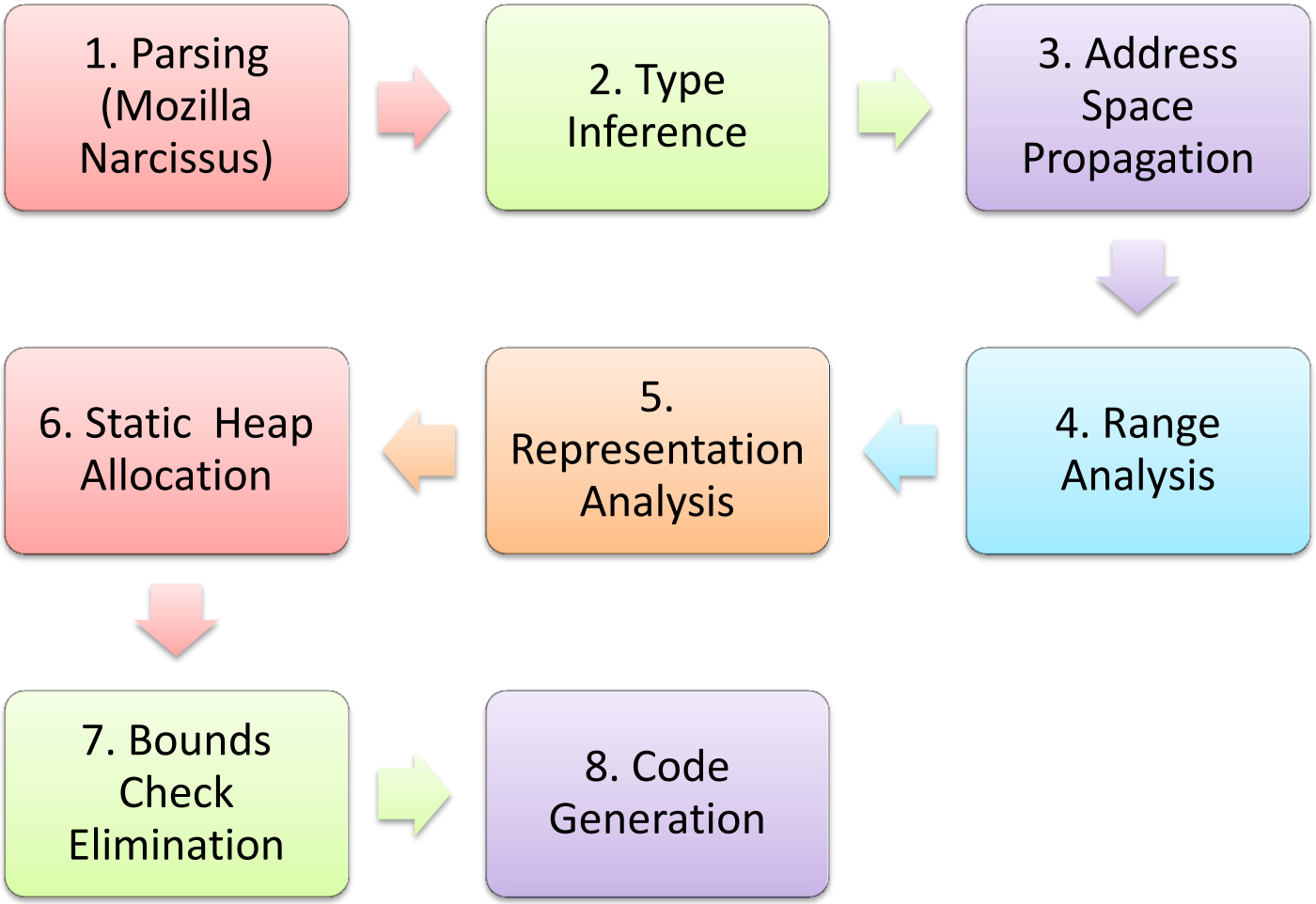
Compiler Restrictions

Don't:

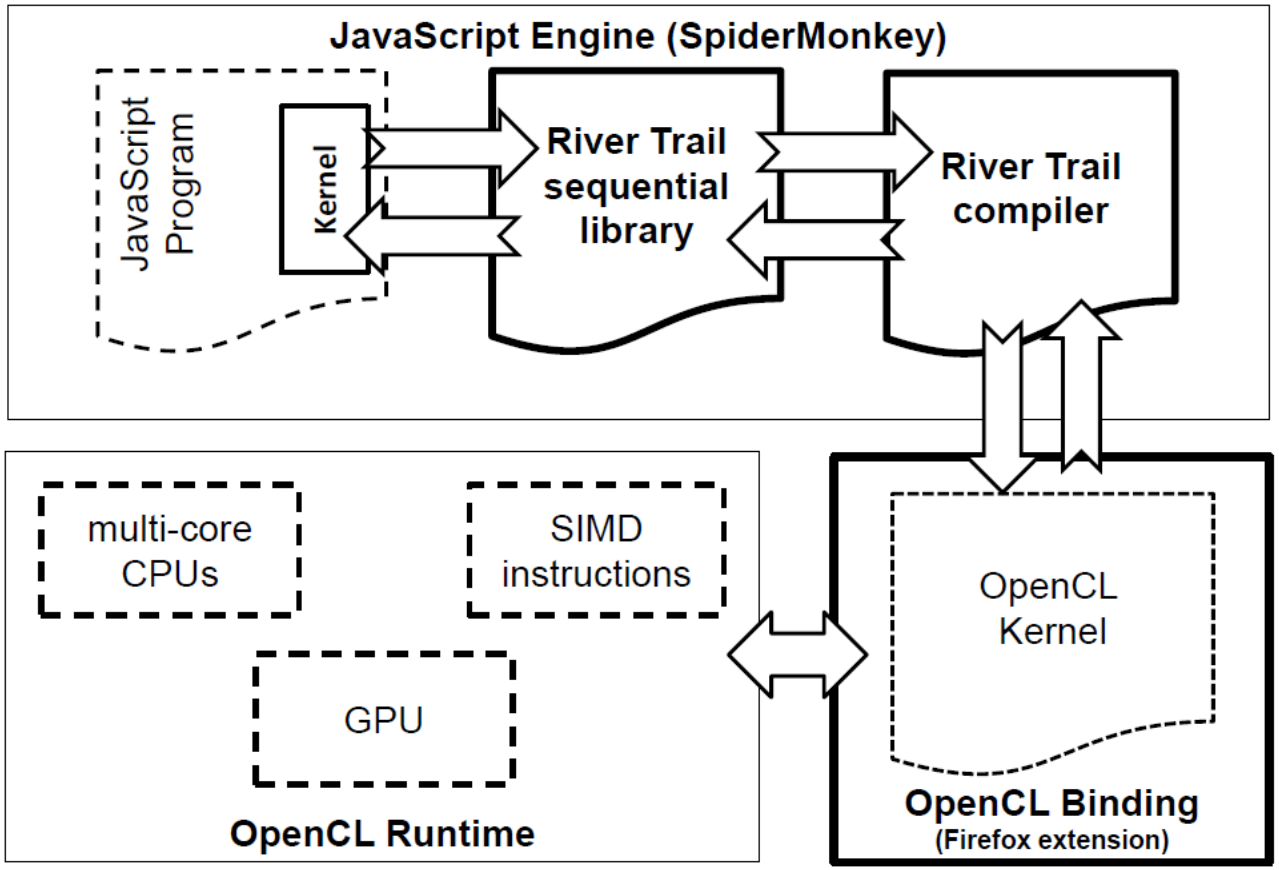
- use closures
- throw exceptions
- use objects
  - except for homogeneous arrays, multiple return types, and Math
- use polymorphism
- use strings
- use null

within elemental functions

Compiler Stages



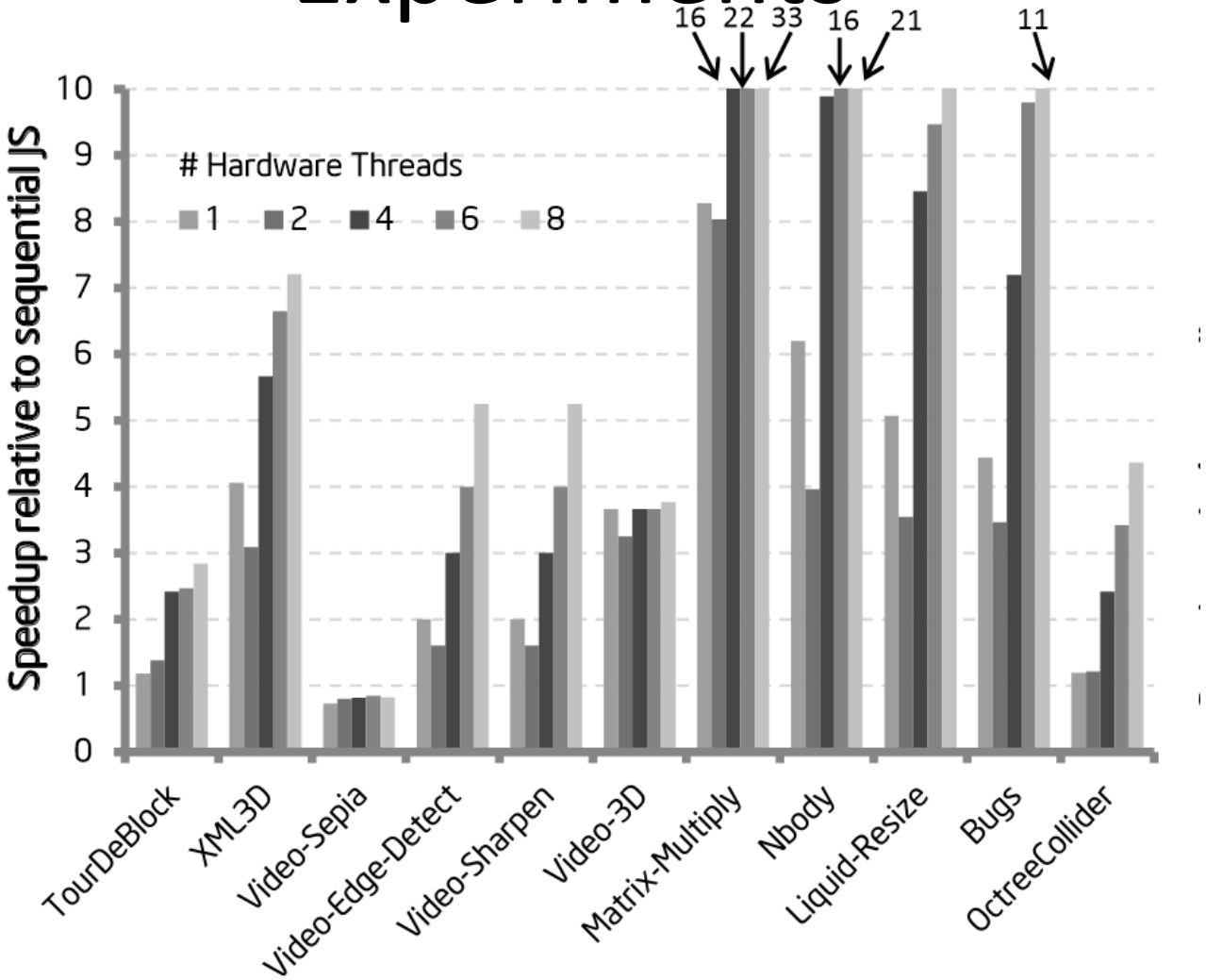
Implementation



After Compilation?

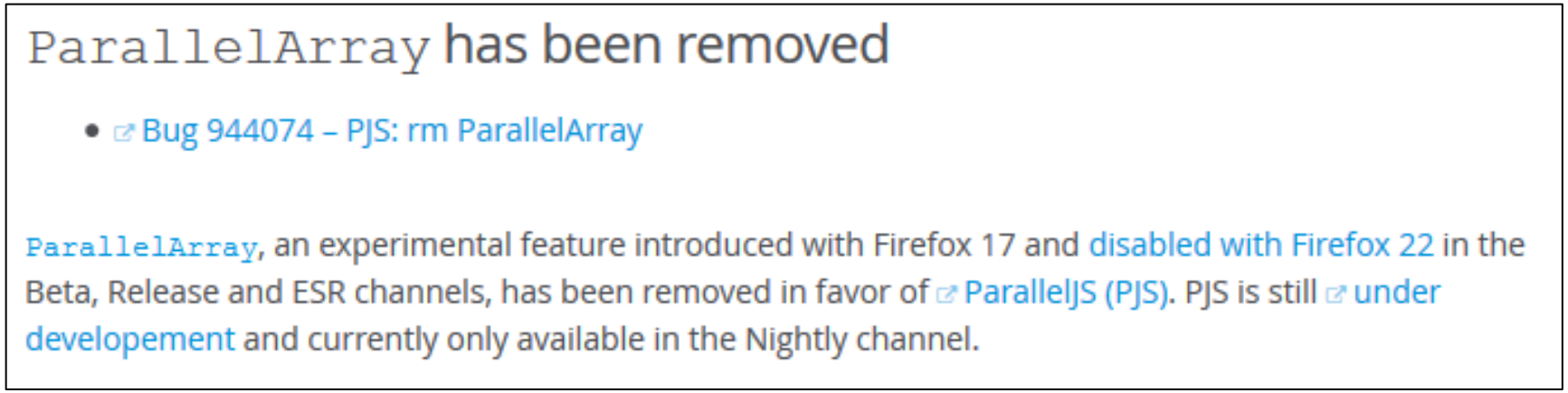
- OpenCL embedding into SpiderMonkey (written in C++)
- Optimizations:
  - cache compiled functions
  - cache mapped ParallelArrays
  - result is not mapped back until read in JavaScript
  - memory alignment
  - dynamically set CPU/GPU distribution factor (“hybrid execution”)

Experiments



Firefox

- Authors claim joint work with Mozilla on production version
- Firefox 29 (April 29, 2014):



`ParallelArray` has been removed

- Bug 944074 – PJS: rm ParallelArray

`ParallelArray`, an experimental feature introduced with Firefox 17 and disabled with Firefox 22 in the Beta, Release and ESR channels, has been removed in favor of ParallelJS (PJS). PJS is still under developement and currently only available in the Nightly channel.

- ECMAScript Proposal

Personal Assessment

- First data-parallelism library for JavaScript
- Minimalistic approach
- Integrates well with existing technologies
- Experimental results are promising
- Enables new kinds of web applications
- Other parallelizable routines
- Additional web browser components needed
- Browser Support?