Language Based Virtual Machines

... or why speed matters

by Lars Bak, Google Inc

Agenda

- *Motivation for virtual machines*
- *HotSpot*
- *V8*
- *Dart*
- *What I've learned*

Background

- *25+ years optimizing implementations of Object-oriented programming languages*

What I Have Worked On

- *86-91 Beta runtime system*
- *91-94 Self virtual machine and IDE*
- *94-95 Strongtalk virtual machine*
- *95-00 Hotspot JVM **
- *00-02 Monty JVM **
- *02-06 OOVM Smalltalk*
- *06-11 V8 JavaScript engine **
- *11-12 Dart programming platform*

In Case You Forgot Beta

```
productDescription:
  (# name: @text;
     price: @integer;
     noOfSoldUnits: @integer;
     order:
       (# orderDate: @date;
          c: ^customer;
          print:<
            (#
            do name[] -> puttext;
               'Price: '->puttext; price -> putint; ' '->put;
               ' No of units sold: '->puttext;
               noOfSoldUnits->putint; ' '->put;
               orderDate.print;
               C.print;
               inner
            #)
       #)
#);
```

Why Language Based VMs

- *Platform independent execution*
- *Sandboxing*
- *Optimization can take place at runtime*
- *Debugging is possible in production*
- *Loading of third part code at runtime*

VM Side Effect

- *When the VM gets faster*
  - *Existing programs run faster*
  - *Programmers get space for innovation*

The Origin of Hotspot

- *The Self Project, 1989-1995*
  - *Research from Stanford University and Sun Microsystems Labs. Inc.*
  - *Groundbreaking adaptive compilation technology*
  - *Efficient memory management*
  - *Managed by David Ungar and Randall B. Smith*
  - *... but was space traded for speed?*

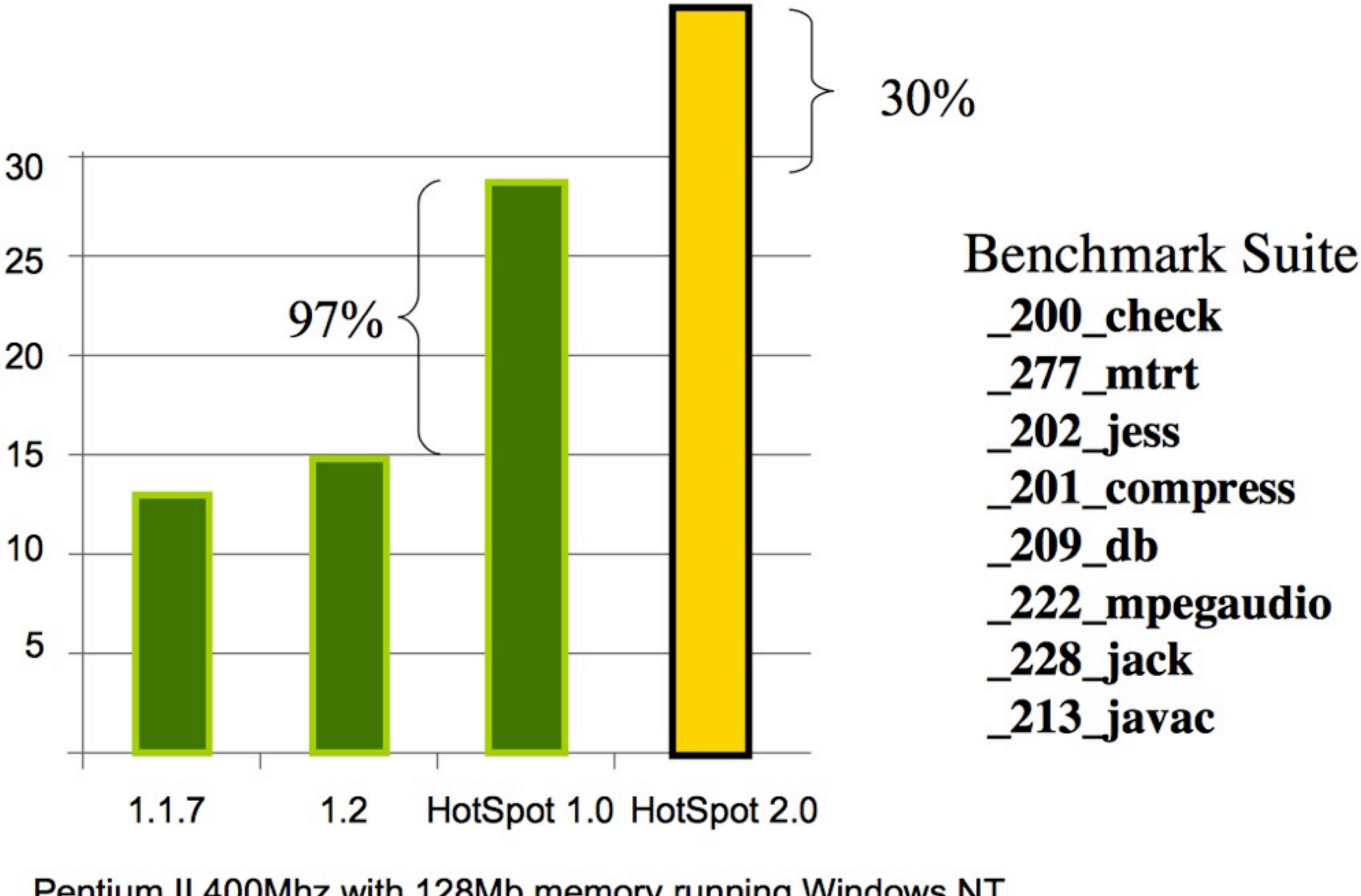
The Startup

- *Longview Technologies, 1994-1997*
  - *David Griswold, Gilad Bracha, Urs Hoelzle, Robert Griesemer, Steffen Grarup, Srdjan Mitrovic, and me*
- *Startup acquired by Sun 1997*

Hotspot

- *Technology*
  - *Based on the Strongtalk VM internals*
  - *Interpreter + simple JIT*
  - *Generational GC*
  - *Fast synchronization*
  - *First implemented with cooperative threads*

SPEC jvm98 www.spec.org/jvm98



Hotspot Still Going Strong

- *After 17 years, Hotspot is still an active project*
- *Caveat: I left the Java world 10 years ago*

Hotspot Reflections

- *Conservative collectors hide problems*
  - *JSR bytecode caused int/pointer slots*
  - *Raw pointers are likely to leak*
- *Allowing virtual behavior on heap objects from C++ is problematic*
  - *'this' subject to GC*
- *Bug-tail for multi-threaded execution is long*

V8: A JavaScript Engine

- *Hired by Google in fall 2006 to improve performance of JavaScript*
- *Matched strategy for Chrome*
  - *Simple, secure, and fast*

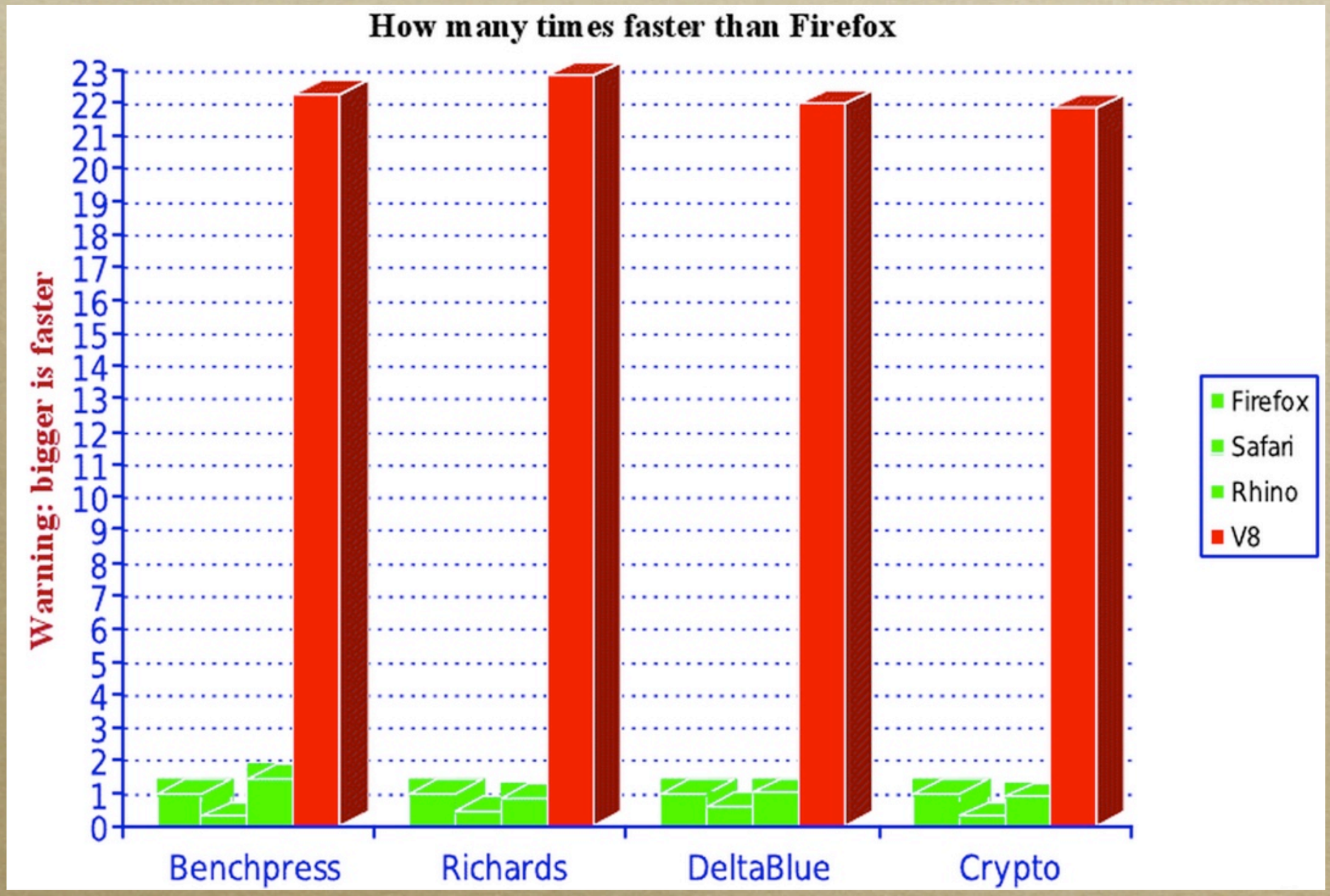
V8 Targets

- *Make JavaScript 10 times faster*
- *Enable JavaScript apps to scale*
- *Open source the project*
- *Raise the industry performance bar*

The Ideas Behind V8

- *Introduced map transitions*
  - *to allow for standard optimizations*
- *Direct from source to machine code*
- *Efficient garbage collection*

Progress After 4 Months



Benchmarking

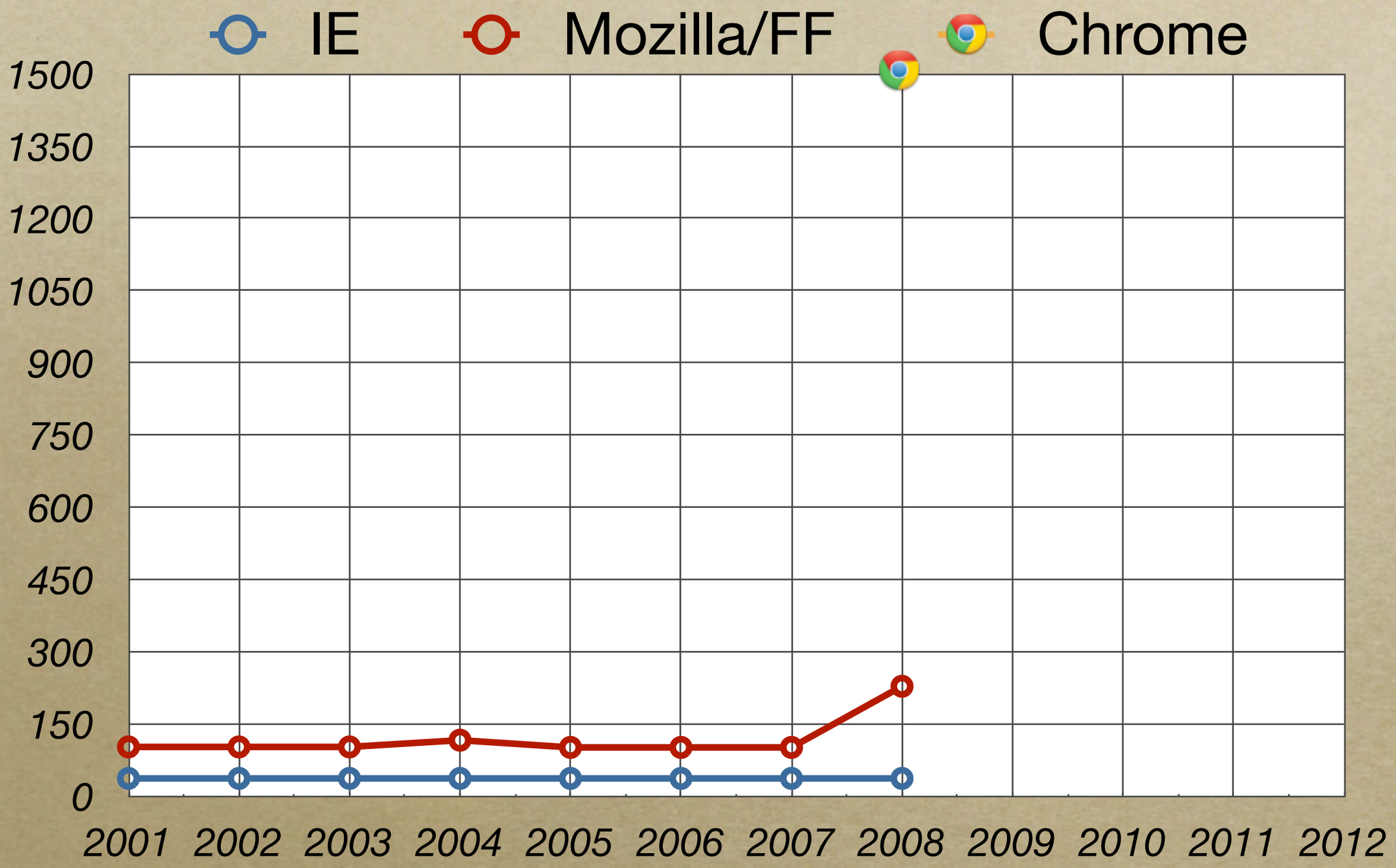
- *Sunspider suite*
  - *Hardly any object allocations*
  - *Hardly any property accesses and calls*
  - *A non OO micro-benchmark*

V8 Benchmark Suite

- *Richards*
- *DeltaBlue*
- *Crypto*
- *RayTrace*
- *EarleyBoyer*
- *RegExp*
- *Splay*
- *NavierStokes*

JavaScript Performance Timeline

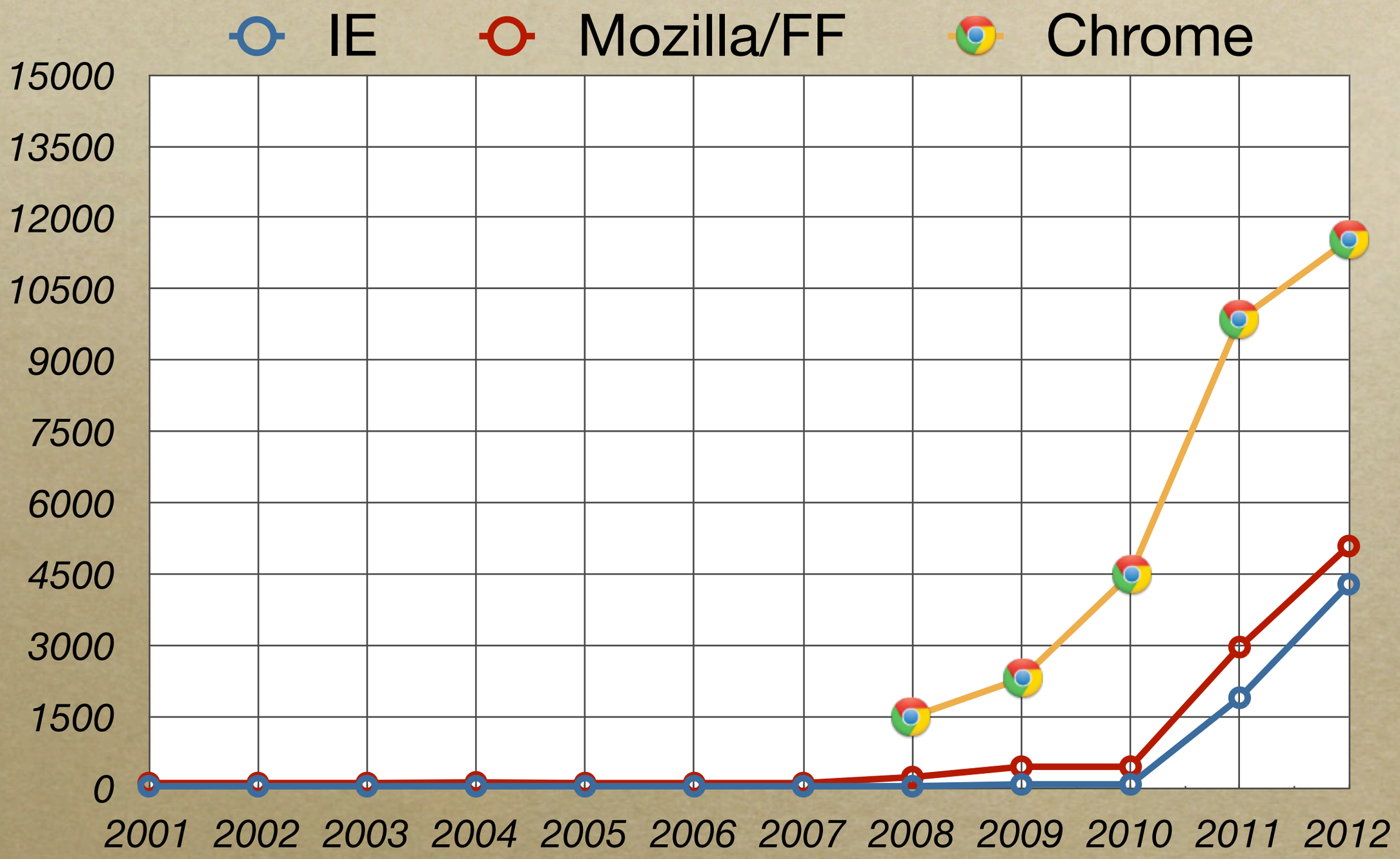
(V8 benchmark v3 - higher is better)



by Jim Hugunin

JavaScript Performance Timeline

(V8 benchmark v3 - higher is better)

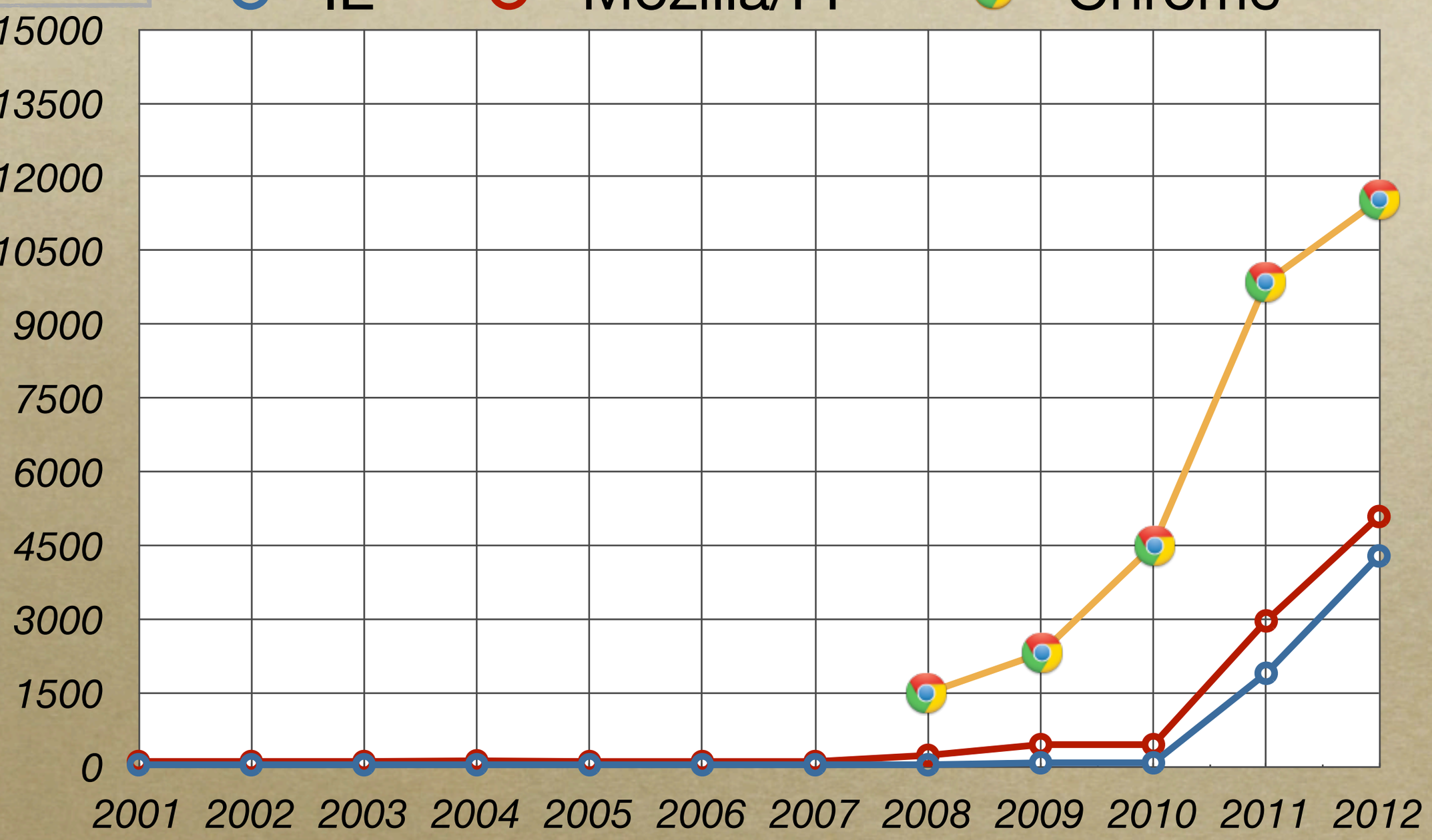


by Jim Hugunin

JavaScript Performance Timeline

(V8 benchmark v3 - higher is better)

Note: Scale shifted by 10x

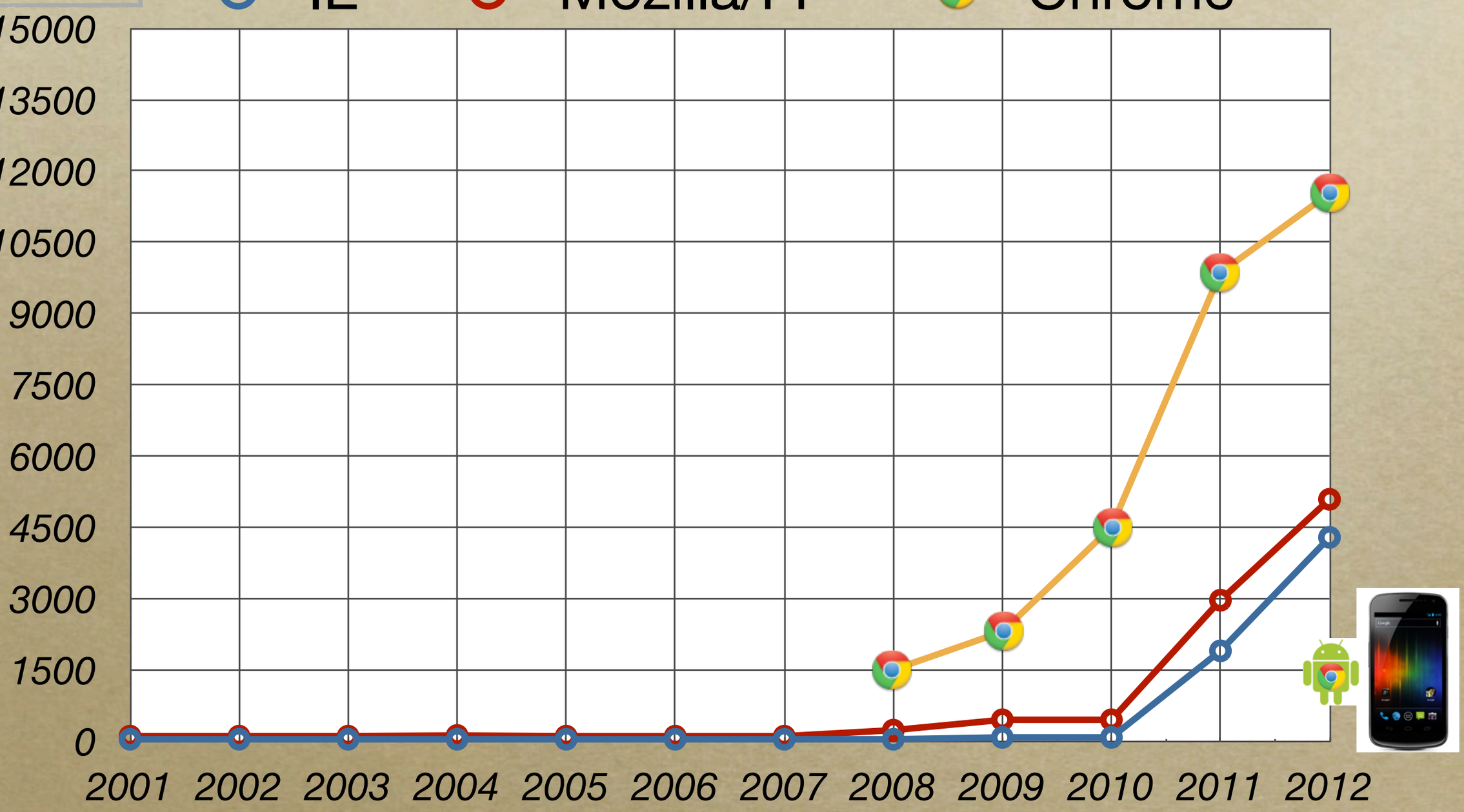


by Jim Hugunin

JavaScript Performance Timeline

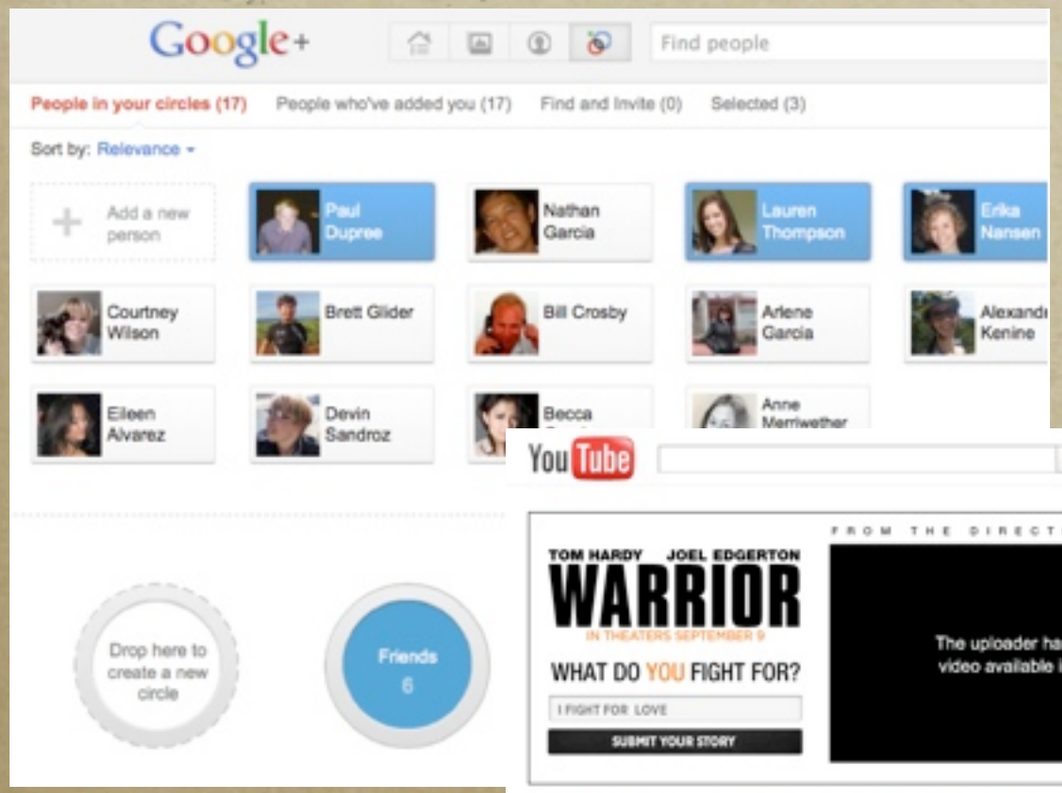
(V8 benchmark v3 - higher is better)

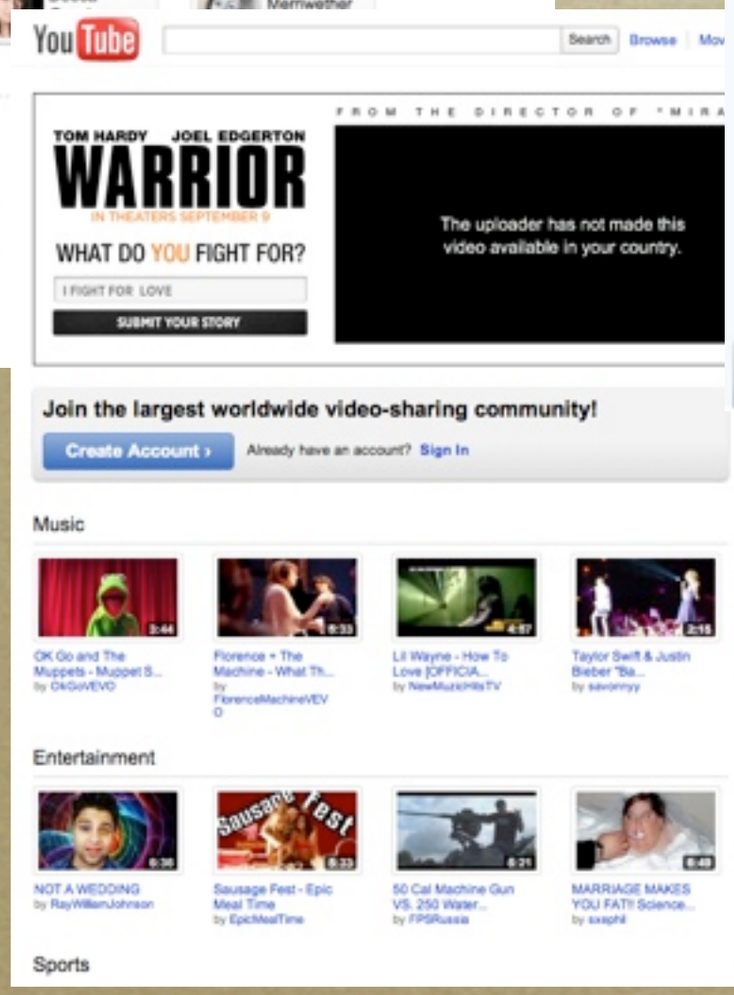
Note: Scale shifted by 10x



by Jim Hugunin

Speed Enables Applications





Using V8

- *Chrome web browser*
- *Android web browser*
- *node.js (server side JavaScript)*
- *Web OS*

Using V8

- *Chrome web browser*
- *Android web browser*
- *node.js (server side JavaScript)*
- *Web OS*

Over 11000 revisions since being open sourced

V8 Reflections

- *Map transitions worked much better that expected*
- *Exposing raw pointers in the C++ runtime system was a mistake*
- *JavaScript speed competition is still going strong*

WAT

- *Presentation by Gary Bernhardt from CodeMash 2012*

JavaScript is Now Faster but ...

- *Promotes spaghetti style programming*
- *Object-oriented programming is hard*
- *Objects can be changed on-the-fly*
- *No support for libraries*
- *Tool support is weak*
- *Slow application startup*
- *Runtime performance is unpredictable*

The Web is Great

- *Developing small applications is easy*
- *Platform independence*
- *No installation of applications*
- *Supports incremental development*

The Web is Great

- *Developing small applications is easy*
- *Platform independence*
- *No installation of applications*
- *Supports incremental development*

... but innovation is crucial for survival

Dart: A New Web Platform

- *Support for programming in the large*
- *Ultra-fast startup*
- *Predictable performance*
- *Backwards compatibility*

The Dart Language

- *A simple and unsurprising OO language*
  - *Class-based single inheritance*
  - *Interfaces with default implementation*
  - *Optional static types*
  - *Real lexical scoping*
  - *Single-threaded*
- *... and it is readable*

Designed for a VM

- *Straight forward semantics*
- *Simple object model*
- *No class initialization*
- *Application are declared*

Dart Sample

```
class Point {
  Point(this.x, this.y);
  var x, y;
  operator +(other) => new Point(x + other.x, y + other.y);
  scale(factor) => new Point(x * factor, y * factor);
  distance() => Math.sqrt(x*x + y*y);
}

main() {
  var a = new Point(10, 10);
  var b = new Point(2, 3).scale(10);
  print("distance=${(a+b).distance()}");
}
```

Dart Sample With Types

```
class Point {
  Point(this.x, this.y);
  num x, y;
  Point operator +(Point other)
     => new Point(x + other.x, y + other.y);
  Point scale(num factor) => new Point(x * factor, y * factor);
  num distance() => Math.sqrt(x*x + y*y);
}

main() {
  Point a = new Point(10, 10);
  Point b = new Point(2, 3).scale(10);
  print("distance=${(a+b).distance()}");
}
```

Dart Optional Type System

- *Static types convey the intent of the programmer*
- *Checkable documentation for code and interfaces*
- *Type annotations have no effect on runtime semantics*
- *Types can be added later*

Dart Isolates

- *Erlang inspired execution model*
- *Lightweight units of execution*
- *Nothing is shared - no object locking*
- *All communication takes place via message passing*
- *Supports concurrent execution*

Dart Components

- *Programming language with spec.*
- *Libraries*
- *Virtual machine*
  - *Integrated into build of Chrome*
  - *Standalone for server execution*
- *Translator to JavaScript*

Dart Project

- *The Dart web site: http://dartlang.org*
  - *Dart language specification*
  - *Dart language tutorial*
- *The Dart project: http://dart.googlecode.com*
  - *Libraries and code samples*
  - *Dart virtual machine*
  - *Dart-to-JavaScript compiler*
- *Expect beta version of Dart SDK later this year*

Credits

- *Object model inspired by* **Smalltalk**
- *Compilation strategy inspired by* **Self**
- *Optional types inspired by* **Strongtalk**
- *Isolates design inspired by* **Erlang**
- *Syntax inspired by* **JavaScript** *&* **C**

Things I Have Learned

- *Always start small with a small team*
- *Focus on solving the hardest problem first*
- *Competitive situation fuels motivation*
- *Only hire people that are smarter...*

Things I Have Learned...

- *Open source projects work*
- *Help industry*
- *Keep your work honest*

Things I Have Learned...

- *So far, I have not needed any Scrum, Agile, pair programming, or any other software development methodology*
- *Common sense has been sufficient*

Things I Have Learned...

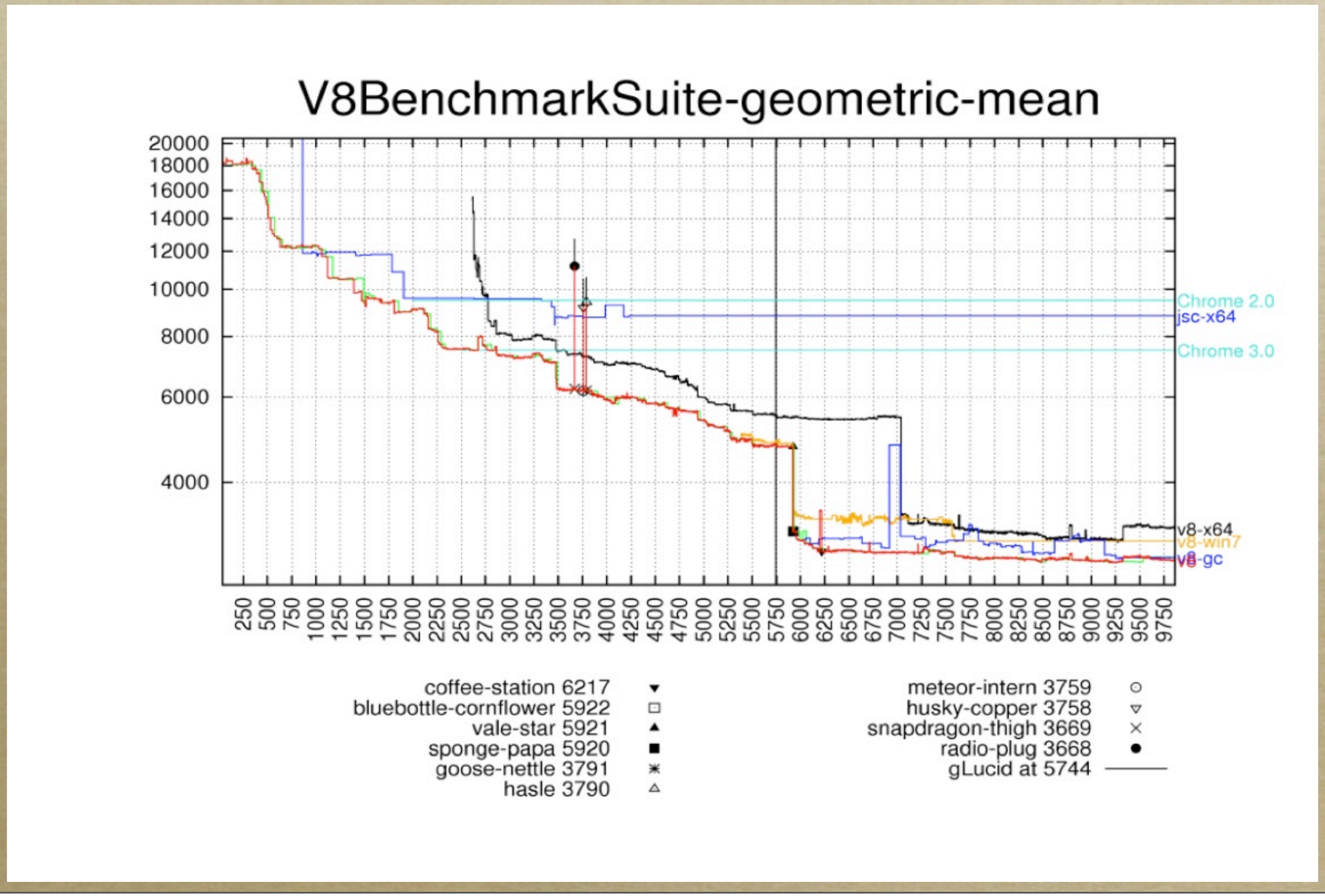
- *Track performance from day one*
  - *Run all benchmarks ...*
    - *on all revisions ...*
    - *on all platforms*

Things I Have Learned...

- *Track performance from day one*
  - *Run all benchmarks ...*
    - *on all revisions ...*
    - *on all platforms*

... if not, performance deteriorates

Performance Tracking



Things I Have Learned...

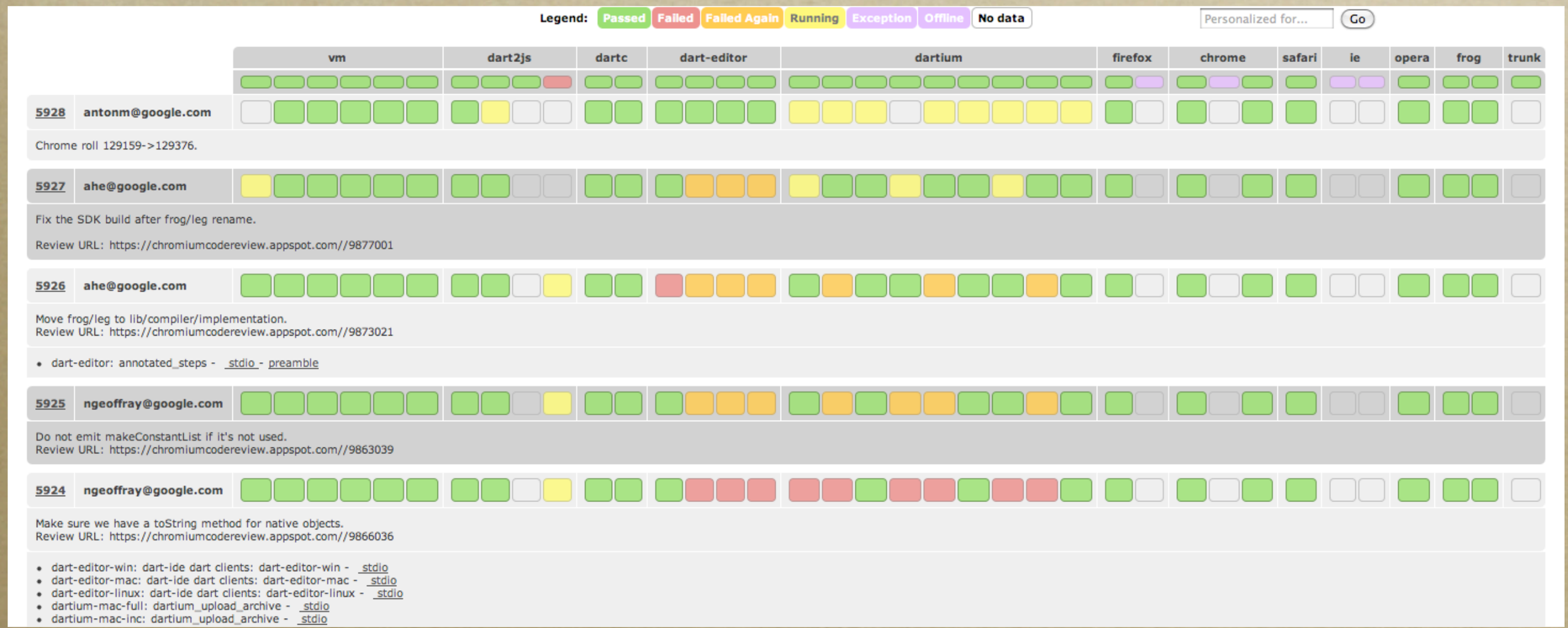
- *Testing must run from day one*
  - *Execute all tests ...*
  - *for all revisions ...*
  - *on all platforms*

Things I Have Learned...

- *Testing must run from day one*
  - *Execute all tests ...*
  - *for all revisions ...*
  - *on all platforms*

... if not, quality deteriorates

Testing Framework



Things I Have Learned...

- *I'm glad I never caved in and designed a general multi language virtual machine*

Things I Have Learned...

- *Technology transfer from research/ academia to industry does not happen through osmosis*
- *If you have great ideas, please go do a startup*

Technical Summary

	Precise GC	*Adaptive optimizations*	*Incremental execution*	*Multi-threaded*	*Bytecodes*	*in C++*	*Open source*
Beta	●			|	|		
Self	●	●	●	|	●	●	●
Strongtalk	●	●	●	|	●	●	●
Hotspot	●	●	|	●	●	●	|
Monty	●	●		|	●	●	|
OOVM	●		●	|	●	●	
V8	●	●	●			●	●
Dart	●	●	|	/		●	●