



1.3k

More

Next Blog»

Create Blog Sign In



Compiling in the background for a smoother user experience

Thursday, February 13, 2014

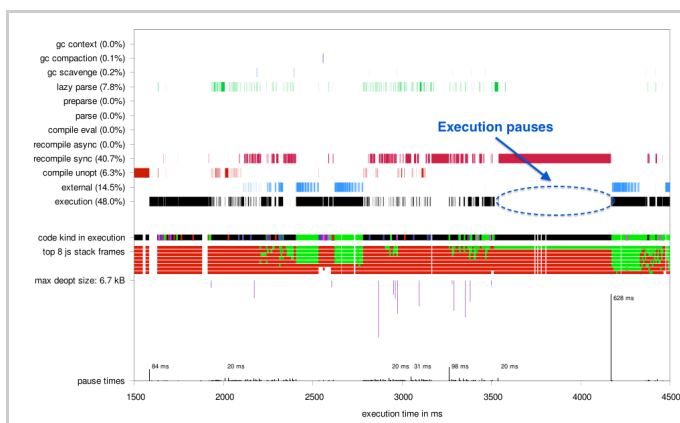
We aim to make Chrome the fastest browser possible. One way [V8](#) helps with this goal is by compiling JavaScript into native machine code to execute scripts quickly. Historically, Chrome compiled JavaScript on the main thread, where it could interfere with the performance of the JavaScript application. In the latest [Chrome Beta](#) we've enabled concurrent compilation, which offloads a large part of the optimizing compilation phase to a background thread. The result is that JavaScript applications remain responsive and performance gets a boost.

Let's have a look under the hood of V8 to understand how this works. To reduce the overall time spent compiling, V8 defers compilation of JavaScript functions until immediately before they are executed the first time. This compilation phase is fast but doesn't focus on optimizing the code, just on getting it done quickly. In V8, pieces of code that are executed very often are compiled a second time by a specialized [optimizing compiler](#). This second compilation pass makes use of many advanced optimization techniques, meaning it takes more time than the first pass but delivers much faster code.

Until now, V8 took turns compiling optimized Javascript code and executing it. For large pieces of code this could become a nuisance, and in complex applications like games it could even lead to stuttering and dropped frames. Concurrent compilation tackles this issue.

The graphs below show excerpts of V8's vital signs when running [Mandree!](#), part of the [Octane benchmark suite](#), on the Nexus 5 phone. The graphs have been created using our new [profiling visualization tool](#). The black bar represents execution of JavaScript, and ideally would be solid.

The first graph shows V8 running without concurrent compilation. V8 is fully occupied with optimizing a large piece of code, causing an execution pause of more than 600ms.



By enabling concurrent compilation, V8 still optimizes a large piece of code, but does it in a background thread. Execution continues alongside compilation, providing for a smoother user experience. For instance, concurrent compilation improved the [Mandree!](#) score of [Octane 2.0](#) by 27% on a Nexus 5 and made graphic-intensive applications such as the [Epic Citadel Demo](#) run even smoother in Chrome.

Search our Blog

Archive

February (2)

Subscribe



Google Chrome Deve...



Follow

+1

More Blogs from Google

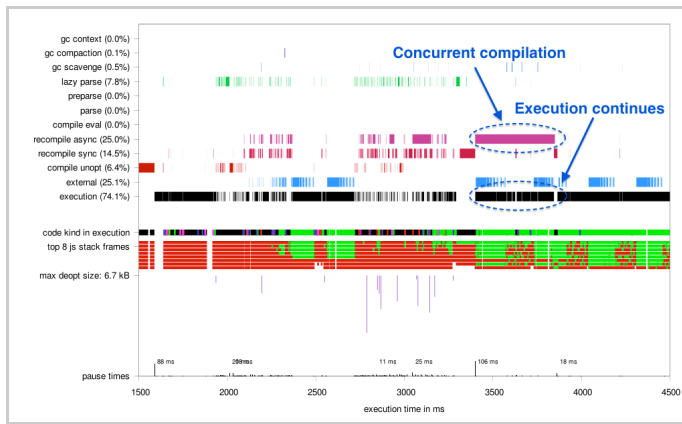
Visit our [directory](#) for more information about Google blogs.

Useful links

[Chromium Homepage](#)
[Google Chrome](#)
[Google Chrome Blog](#)
[Google Chrome Release Notes](#)
[Google Open Source Blog](#)
[Google Code Blog](#)
[WebKit Blog](#)

Labels

[accessibility](#) (1)
[benchmarks](#) (1)
[beta](#) (1)
[blink](#) (1)
[chrome apps](#) (3)
[Chrome Frame](#) (1)
[chrome web store](#) (26)
[chromeframe](#) (3)
[chromeos](#) (3)
[chromium](#) (3)
[cloud print](#) (1)
[dart](#) (8)
[devtools](#) (11)
[extensions](#) (23)
[gdd](#) (1)



Concurrent code compilation is another step towards [reducing latency](#) in Chrome and is part of various [ongoing efforts](#) to deliver more responsive, smoother web applications.

Posted by Yang Guo, Multi-threaded V8 Engineer

 +1372 Recommend this on Google

[googlechrome](#) (12)

[html5](#) (11)

[incognito](#) (1)

[javascript](#) (3)

[linux](#) (2)

[mac](#) (1)

[mobile](#) (2)

[na](#) (1)

[native client](#) (8)

[New Features](#) (5)

[octane](#) (1)

[open web](#) (2)

[releases](#) (2)

[rlz](#) (1)

[security](#) (23)

[spdy](#) (2)

[ssl](#) (2)

[v8](#) (5)

[web intents](#) (1)

[webaudio](#) (3)

[webgl](#) (7)

[webkit](#) (5)

[webp](#) (5)

[webrtc](#) (4)

[websockets](#) (5)

[webtiming](#) (1)



Pages

[Home](#)

113 comments



Add a comment as Alex Brooks

Top comments

**Szabolcs Péter** 3 days ago - Shared publicly

Is background compilation now live on stable Chrome 33?

+1 1 · Reply

**Marc d'Entremont** 1 week ago - Shared publicly

That's pretty cool.

Will this approach end up filtering to Node.js?

1 · Reply

**Vyacheslav Egorov** via Google+ 1 week ago - Shared publicly

+15 1 · Reply

Hide replies

**Egor Pasko** 1 week ago (edited) +2

Chrome people never stop to amaze me. Report performance results on Nexus 5, just like it's reproducible. Show PNG, hide raw data. Fukyea.

**Vyacheslav Egorov**

1 week ago +5

+**Egor Pasko** it's a demo, not a scientific paper after all.**Vaidotas Didžbalis**

10 hours ago (edited)

Old paper on the related topic, "Self includes: Smalltalk"

<http://www.merlinter.com/download/mario.pdf>**Vyacheslav Egorov**

8 hours ago

+**Vaidotas Didžbalis** I fail to see how it is related topic. There is nothing about concurrent compilation in that paper.

Add a comment...

[Post a Comment](#)

[Older Post](#) [Home](#)

Subscribe to: [Post Comments \(Atom\)](#)