

# How we made **VisualEditor** faster

Ori Livneh and Roan Kattouw

# “Users really respond to speed.”

- When asked, users unanimously said they preferred more results.
- A/B test of 30 results per page vs. 10 showed 25% fewer searches in less than six weeks.
- Why? 30 results takes .9 seconds to generate; 10 results take .4.
- **“Latency really does matter to users, whether or not they can articulate it.”**

Marissa Mayer (Google),  
[In Search of... A better, faster, stronger Web](#)

# Performance matters

- Strong correlation between speed and engagement observed elsewhere. No reason to believe we're different.
- We use VisualEditor too, and we want it to be fast.
- Editing Wikipedia requires boldness, don't give users time to second-guess themselves or they will!

# Measuring performance

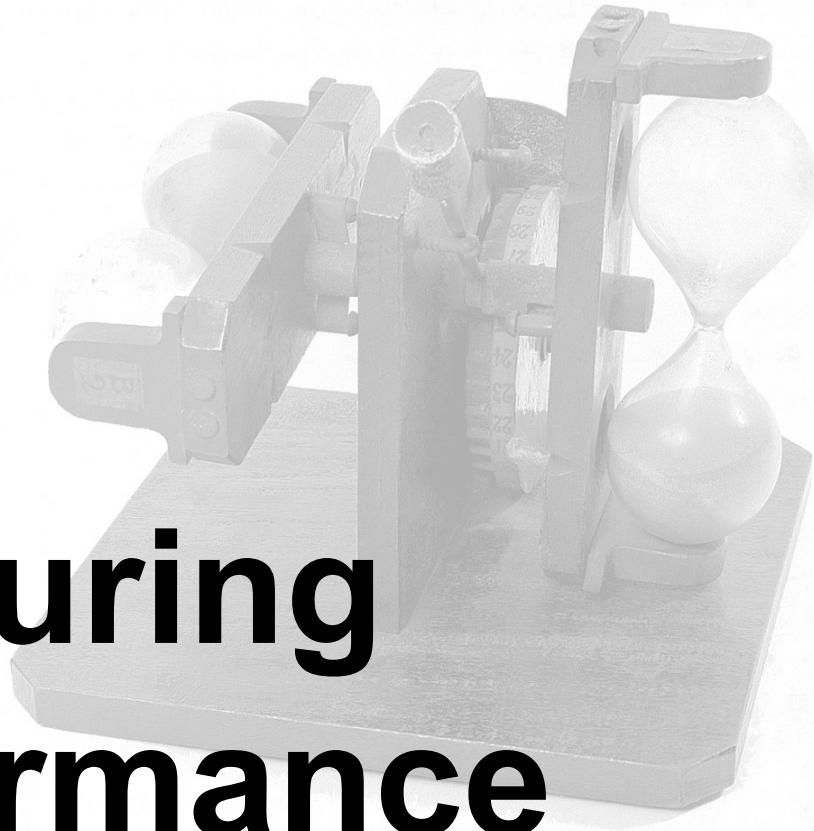


Image by [Wellcome Images](#)



GET /beacon/status?v=load=920...



ve.load:920|ms

Graphite



Graphana



RUM and synthetic (vbench)



MEDIAWIKI



ve.save:1241|ms



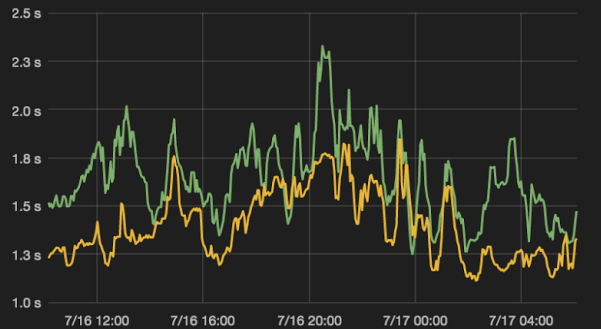
Parsoid



parsoid.serialize:1492|ms

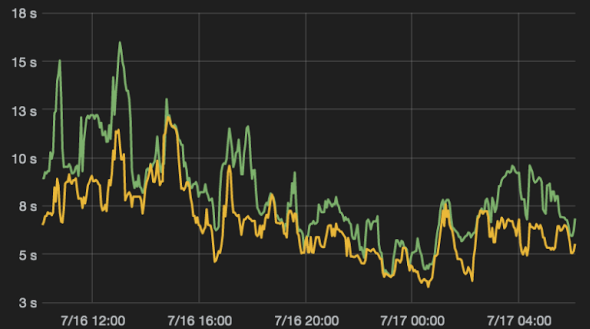
metrics  
data flow

VisualEditor HTML DOM load time, mean



	min	max	avg
mean RESTBase load time	1.250 s	2.327 s	1.644 s
mean PHP API load time	1.112 s	1.843 s	1.397 s

VisualEditor HTML DOM load, 99th percentile



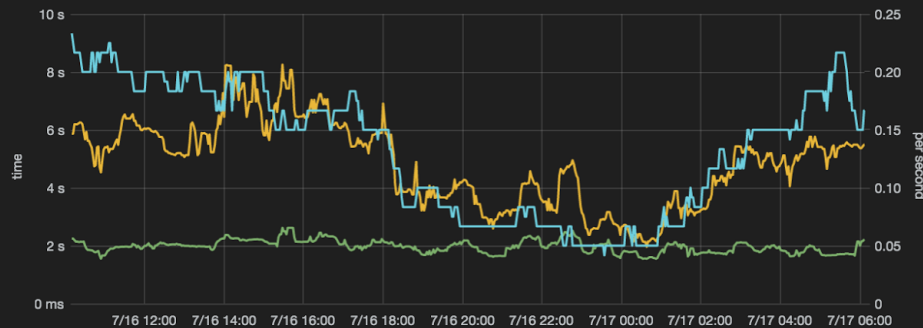
	min	max	avg
99% RESTBase load time	3.81 s	15.96 s	8.22 s
99% PHP API load time	3.30 s	12.09 s	6.61 s

VisualEditor HTML DOM load, rates



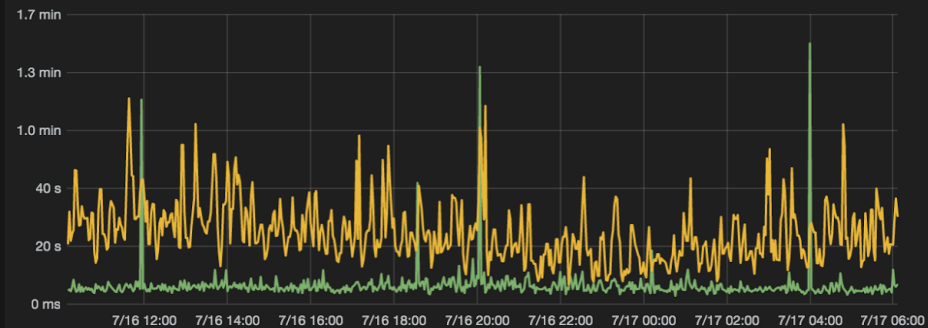
	min	max	avg
RESTBase load rate (req/s)	0.12	0.60	0.38
PHP API load rate (req/s)	0.13	0.63	0.39

VisualEditor HTML DOM serialization



	min	max	avg	current
mean serialization time	1.54 s	2.62 s	1.97 s	2.24 s
99% serialization time	1.96 s	8.25 s	4.84 s	5.49 s
save rate req/s	0.042	0.233	0.133	0.167

VisualEditor activation time



	min	max	avg	current
mean activation time	2.7 s	1.497 min	6.2 s	6.0 s
99% activation time	6.7 s	1.181 min	24.5 s	26.9 s

VisualEditor save completion time, in milliseconds, 05/17/2015 - 07/17/2015



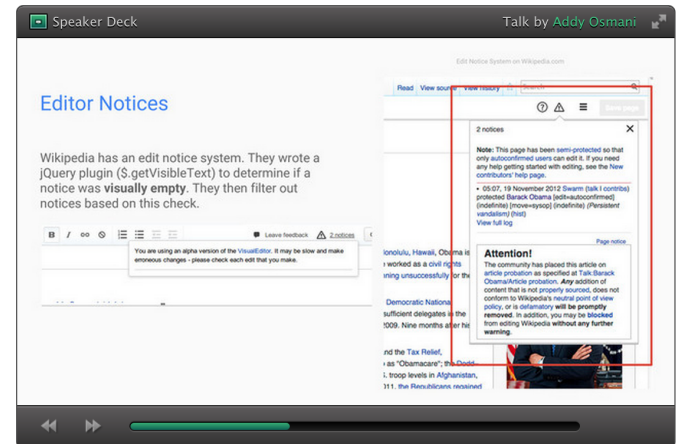


Annotating the timeline recording with post-its and markers to show phases of load with post-its and markers.



# Tooling

- Not mature yet; new frontier
- Reached out to Paul Irish of Google Chrome team.
- Got a free performance audit.
- Takeaways:
  - These are really hard problems. We are pushing boundaries.
  - Our openness and goodwill help builds bridges.



# Upstream issues

jQuery .show() / .hide(): **400ms**

```
var isHidden = function( elem, el ) {  
    // isHidden might be called from jQuery#filter fu  
    // in that case, element will be second argument  
    elem = el || elem;  
    return jQuery.css( elem, "display" ) === "none" |  
};
```

Faster: .addClass( 'hideme' );

# Upstream issues

```
var $a = $( '<p><b>Hello</b></p>' ),
    $b = $( '<p><b>Hello</b></p>', otherDoc );
console.profile();
for ( i = 0; i < 1000; i++ ) {
    $a.children( 'b' );
    $b.children( 'b' );
}
console.profileEnd();
```

otherDoc !== document: **2388ms**

otherDoc === document: **23ms**

# Upstream issues

```
var a = document.createElement( 'img' );
a.src = 'data:image/gif;base64,R0lGODdhAQABAADcACQ=';
console.profile();
for ( i = 0; i < 1000; i++ ) {
    a.cloneNode();
}
console.profileEnd();
```

With src: **196ms**

Without src: **3ms**

# Upstream issues

- “I’m surprised this is the up-to-date version of jQuery”
- Pushing frontiers means finding bugs in infrastructure

# Minor mistakes can be costly

- Redundant data structure: **500ms**
  - Don't copy information out of DOM, just keep DOM
- Change font size after rendering: **250ms**
  - Add CSS class that sets font size *before* attaching
- Workaround for enwiki's edit notices: **200ms**
  - “Is there visible text here” is an expensive question

# Fetching content

- Request for page HTML sent **1200ms** after user clicked edit
- Waterfall requests
  - On edit click, load code module A
  - A loads code module B and content (C)
- Combine and parallelize
  - On edit click, load A+B and load C in parallel
  - All requests sent **immediately** on edit click

# Fetching content

- Page HTML was cached in Varnish
  - Hard to ensure high/consistent hit rate
  - Cache misses are expensive
  - Updates and prepopulation are hard/hacky
- Page HTML is now in RESTbase
  - Cache that doesn't lose data (AKA storage)
  - Easier to populate new HTML on save
  - Cache misses less frequent now



# Perceived Performance

*The experience of waiting...is defined only partly by the objective length of the wait. “Often the psychology of queuing is more important than the statistics of the wait itself,” notes the M.I.T. operations researcher Richard Larson, widely considered to be the world’s foremost expert on lines. **Occupied time feels shorter than unoccupied time.***

**Why Waiting Is Torture.** New York Times, 2012-08-19

# Perceived Performance

Resume editing Save your changes Save page

**Edit summary** (Briefly describe your changes)

Describe what you changed

This is a [minor edit](#)  Watch this page 255

By clicking the "Save page" button, you agree to the [Terms of Use](#) and you irrevocably agree to release your contribution under the [CC BY-SA 3.0 License](#) and the [GFDL](#) with the understanding that a hyperlink or URL is sufficient for CC BY-SA 3.0 attribution.

Review your changes

# Perceived Performance

The image shows a screenshot of a web browser window displaying a Wikipedia article in edit mode. A modal dialog box titled "Save page" is open, asking the user to confirm saving their changes. The dialog includes a "Review your change" button and a "Save page" button. The background shows the Wikipedia article content, including a table of contents and a list of items. Below the browser window, a network monitor is visible, showing a list of network requests with columns for Name, Method, Status, Type, Inferred, Size, Time, Content, Start Time, and Size. The network monitor shows three requests: a 200 status for a GET request to /api.php, a 200 status for a POST request to /api.php, and a 200 status for a GET request to /api.php. The network monitor also shows a "No throttling" message.

Name	Method	Status	Type	Inferred	Size	Time	Content	Start Time	Size	Size	Size	Size
/api.php	GET	200	text/html	HTML	10	0.000s						
/api.php	POST	200	application/javascript		1758	0.001s						
/api.php	GET	200	text/html	HTML	344	0.000s						

# Perceived Performance

*“duration has little effect on how pleasant an affective experience is rated (duration neglect). Instead, perception is most heavily influenced by salient features (both good and bad) during the experience and at the conclusion of the experience (peak-and-end effects).”*




**Rethinking the Progress Bar**

# Perceived Performance



The image shows the top portion of a Wikipedia article. At the top, there is a navigation bar with tabs for "Page", "Discussion", "Read", "Edit", "Edit source", and "More". A search box is located on the right side of the navigation bar. Below the navigation bar is a toolbar with various editing tools, including a link icon, a list icon, an "Insert" button, a table icon, and a link icon. A mouse cursor is pointing at the link icon in the toolbar. Below the toolbar, the article title "Barack Obama" is displayed in a large, serif font. To the right of the title, there is a blue progress indicator consisting of several vertical bars of varying heights.

Page Discussion Read Edit Edit source More Search

← → ↕ A ↕   Insert  

Barack Obama 

# Perceived Performance

- Humans do not perceive time in a linear way.
- Users have a strong aversion to pauses, especially towards the end of an operation.
- Solution: progress can be downplayed in the beginning and accelerated towards the end, providing a sense of a rapid conclusion.
- [T95137: The progress bar should accelerate as it nears completion](#)



Esanders added a comment.

Via Web · Apr 6 2015, 3:42 AM



Tried and failed to add this to the #OriLivnehMindControlDept project.

# Perceived Performance

Article

Talk

Read

Edit source



View history



More ▾

Search



## Château de la Muette

From Wikipedia, the free encyclopedia

Coordinates:  48°51′41″N 2°16′10″E

The **Château de la Muette** (French pronunciation: [ʃato də la mɥɛt]) is a [château](#) located on the edge of the [Bois de Boulogne](#) in [Paris](#), France, near the [Porte de la Muette](#).

Three châteaux have been located on the site since a hunting lodge was transformed into the first château for Princess [Marguerite de Valois](#), favorite daughter of [King Henry II](#), sister of Kings [Francis II](#), [Charles IX](#) and [Henry III](#) and the first wife of [King Henry IV](#), in the 16th century. The first château was extended and substantially reconstructed by [Louis XV](#). [Louis XVI](#) and [Marie-Antoinette](#) lived at this second château, and the first manned flight, in a [hot air balloon](#), set off from the château in 1783.



Château de la Muette, Paris in 2013



# Putting it all together

