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
RUM and synthetic (vbench)
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

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

Faster: `.addClass( 'hideme' );`
Upstream issues

```javascript
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

```javascript
var a = document.createElement( 'img' );
a.src = 'data:image/gif;base64,R0lGODdhAQABAADcAQAA
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.

Perceived Performance
Perceived Performance
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

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.

Tried and failed to add this to the #OriLivnehMindControlDept project.
Château de la Muette

From Wikipedia, the free encyclopedia

The Château de la Muette (French pronunciation: [ʃato de la möɛt]) 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.
Putting it all together