

Detecting Device Memory Capability

panicker@chromium.org, mariakhomenko@chromium.org

Mar 10, 2017

Motivation

“one-size-fits-all” web experience does not work in a world with widely varying device capabilities. Web apps that delight users on high end devices can be unusable on low end devices, particularly in emerging markets.

To support a diverse web ecosystem that is inclusive of users on low end devices, apps need to be able to tailor their content to the **device constraints**. This allows users to consume content suited to their device, and ensures they have a good experience and fewer frustrations.

Developers are interested in the “**device-class**” for the following known use-cases:

1. Serve “light” version of the site or specific components, for low end devices. Egs.

- Serve Google "search lite" - a 10KB search results page used in EM.
- Serve a light version of video player in Facebook
- Serve lightweight tile images in Google Maps

2. Normalize Metrics: analytics need to be able to normalize their metrics against the device-class. For instance, a 100ms long tasks on a Pixel is a more severe issue vs. on a low end device.

Device memory is an interesting signal in this context. Low memory devices (under 512MB, 512MB - 1GB) are widely used in emerging markets. Our telemetry indicates large number of OOMs on foreground renderer on these devices. In this case, serving a lite version not only improves the user experience, it is *necessary* for the site to be usable at all (as opposed to crashing due to memory constraint).

Proposal

We propose a **header and web exposed API** to surface device capability for memory (RAM). The mechanism should be extensible to other device capabilities such as CPU i.e. number of cores, clock speed etc.

A header will enable the server to deliver appropriate content, eg. a “lite” version of the site.

The JS API will enable clients to make appropriate decisions eg. using more storage vs. making additional requests, requesting appropriate resources from the server etc.

ASIDE: the JS API for CPU Cores is already available via hardwareConcurrency API, but there is no header available.

The Header

Proposed Header for memory: "device-ram"

```
"device-ram" ":" <value>
```

where <value> is the number of GiB of ram (floating point number) rounded down to the nearest power of two.

For example, if the user has the total of 512 MiB of RAM the value would be 0.5. If they have 768 MiB of ram it would also be 0.5. If they have 3 GiB of RAM the value would be 2.

Why separate header and rounding?

HTTP caching doesn't deal well with mixed value headers, therefore separate headers are recommended. Also, rounding down to power of two enables caching and mitigates fingerprinting.

When is the header sent?

The header will be sent after an explicit per-origin opt-in via [Client Hints](#) mechanism. The following new hint will be added: Accept-CH: device-ram

For background, [Client Hints](#) provides a set of HTTP request header fields, known as Client Hints, to deliver content that is optimized for the device. In that sense using Client Hints is a great fit for here.

Client Hints recently [addressed a significant limitation in spec](#): opt-in can now be persisted across browser restarts using "max-age" in header.

The web exposed API

We propose adding the following API to [navigator](#):

```
navigator.deviceRam
```

which returns number of GiB of ram (floating point number) rounded down to the nearest power of two (same as the header).

TODO: Any reason to add a level of indirection similar to [NavigatorConcurrentHardware](#)?

Why not surface Device Class directly?

While exposing a composite "Device Class" would be useful for developers, it has a number of challenges: it's hard to specify in a future-proof manner as it is constantly changing, it requires significant and ongoing work (constantly update algorithm OR classify known devices), it is difficult to reach agreement amongst vendors, and come up with something that works for all web sites etc.

While this is something worth considering down the road, we think we can get most of the benefit by exposing a couple of specific signals device memory and device CPU (cores). In particular device memory is a reasonable proxy for device class.

Security & Privacy

See [Security section in Client Hints](#).

Requiring per-origin opt-in with Accept-CH restricts when the header is advertised.

Restricting to a ceiling value (rounded down to the nearest power of two), as opposed to exact value, mitigates fingerprinting risk. This could be further restricted, if necessary, to returning values only for low memory cases i.e. less than 1GB.

Relevant Links

<https://github.com/facebook/device-year-class>

<https://github.com/igrigorik/http-client-hints>

<https://developers.google.com/web/updates/2015/09/automating-resource-selection-with-client-hints>

<https://github.com/WICG/netinfo/issues/46>

<https://developer.mozilla.org/en-US/docs/Web/API/NavigatorConcurrentHardware>