Why we work on memory

haraken@chromium.org (inspired by nduca@chromium.org)
Last update: 2016 Sep 6
Status: PUBLIC

History

Memory usage in Chromium has increasingly become a focused area. It was raised as the top priority of the web-platform team at the Web Platform Leads in June 2016. For instance, 60 - 70% of the crash reports on foreground renderers on low-RAM Android are OOM.

To address those memory problems, we established TRIM (Targeted Reductions in Memory). The goal is to coordinate a large group of distributed memory experts toward a unified goal, identifying key trouble areas that need more attention, improving and scaling tooling on the Chrome scale, and landing critical optimizations from small fixes to architecture overhauls.

Why memory matters at a product-excellence level

User Pain: Most users notice memory issues indirectly, by experience pain during their day-to-day workflow. As they cycle around apps & tabs through their day, switching from one, to another, to another, they will get all kinds of memory-induced pain:

1. OOM on foregrounded renderers. As mentioned above, 60 - 70% of the crash reports on low-RAM Android, which is strategically important, are OOM.
2. Sluggishness / unusability of Chrome due to high memory usage.
3. Sluggishness / unusability when switching between apps because Chrome is eating lots of memory. Chrome is a bad citizen of its host.
4. State lost when moving to a previously backgrounded tab. Chrome may kill backgrounded tabs on both Android and desktops

Tech press: In addition to the user pain, we also care a lot about the tech press' framing of memory: there are some that simply feel that Chrome is too memory hungry. A good example is here. It is important that we appease these folks.

Low-RAM: For many years to come, it remains strategically important that Chrome performs well on extremely low memory Android devices for emerging markets, those with 512 MB or less. It is critical to recognize that the user pain is huge on those devices. Even in the case of single-tab browsing, it's not rare that the foreground tab gets killed due to high memory pressure. Background renderers are highly likely to get killed and the state is just lost. The

Android team tracks product-excellence issues around memory by simulating a user moving from chrome, to another app, to another app, over and over, and validating that all apps have stayed in memory without state loss. If a user cannot move between a certain number of these apps without a kill, then it is considered as a release blocker.

How to balance a trade-off between performance and memory

Even though memory is super important for product excellence, performance is also the most important thing in Chrome (it's one of the 3S: Speed, Simplicity, Security).

Here is a high-level guideline of how we think about performance and memory:

- On non-low-RAM devices, performance is clearly the most important thing. The goal is to minimize the memory consumption without affecting performance. We should not trade off performance for memory. As far as we've experienced, there're still lots of things (even though they're not low-hanging fruits) we can do to reduce memory consumption without affecting performance.
- On low-RAM devices, memory is likely to be the largest contributor to poor product excellence. Hence we sometimes want to trade off performance for memory. It's hard to define in general how much performance regression is justified for how much memory improvement, so we should make the best judgement on a case-by-case basis.
- Memory Coordinator is going to be a central scheduler that has a global knowledge of all existing tabs in Chrome. Memory Coordinator dispatches notifications when the system is under high memory pressure. All memory components should listen to the notifications and take proper actions.

The existing low-RAM configuration is a static thing. The Memory Coordinator is only for memory. In long term, we should consider generalizing the idea to a "global coordinator", which dynamically balances performance, memory and battery together.