DevTools benchmark: Chromium Speed Infra or its own infra?

Note: this document is public

Author: crouleau@, TL Chrome Benchmarking
Date: Nov 21, 2019

Context

Yang wrote a document DevTools frontend startup benchmark that explains the need for a devtools Frontend benchmark. I think that this is a good idea, but the infrastructure for it should be separate from the infrastructure that we use for Chromium. I explain below.

Disadvantages of Running On Chromium Infra

There are disadvantages to running performance testing for DevTools inside of Chromium Speed Infra rather than as part of its own infra.

1. Frequency of commits. Chromium Speed Infra handles Chromium-scale commits. This seems to be one commit every 2 minutes roughly (as opposed to one commit every 2 hours roughly on devtools frontend (https://chromium.googlesource.com/devtools/devtools-frontend.git/+log/refs/heads/master): https://chromium.googlesource.com/chromium/src/+log/refs/heads/master. Because of this frequency of commits, we can only run the full suite of performance tests once every 100 commits or so (depending on the time of day). This means that when we notice a performance regression, the list of potential culprits is huge. We have a bisection tool called Pinpoint to figure out which commit caused the regression, but it frequently doesn't work for many reasons:
    a. Chrome fails to compile at certain commits.
        i. CQ compiles Chromium without release optimizations. Every few weeks, a changelist slips in that breaks Chrome's release build. Performance tests are run against a Chrome binary that mimics release and has all optimizations on. For revisions at which Chrome does not compile, Pinpoint will fail.
    b. Capacity issues. Pinpoint is frequently overrun with too many job requests. Since it is one large pool, requests from one team cause another team to become starved.
    c. Statistics problems. Sometimes Pinpoint identifies lots of potential culprits and many of them don't make sense. Other times it misses obvious culprits.
2. Flakiness.

    a. Changes to Chromium frequently break various Telemetry features. You cannot catch regressions if you cannot run the tests. If you pinned a specific version of Chrome and only updated it for new releases, you would be shielded from this flakiness.

3. Roll failures will cause DevTools frontend changes to become bunched up together, so the performance graphs will no longer show the granularity that you would want: you won't be able to tell which of your changes caused a regression.

Alternative

Instead, DevTools frontend could simply have its own performance infrastructure. It could run performance tests on CQ or waterfall. It could use Telemetry or it could use its own separate system. (Many open source and closed source products exist for testing website performance.) Since DevTools already uses LUCI, it could easily mimic our configurations.

However, you can mix and match various tooling that you want. Hablich@ explained that you could instead of using what we have in Telemetry/Speed Infra do this:

1.) Open DevTools frontend via puppeteer - Run the tests on special bots (machenbach@ should know more here)
2.) Use tracing to measure performance (thanks +mathias@chromium.org)
3.) Send results to Chromeperf
4.) Display graphs on Chromeperf

I'll add that you can also set up Pinpoint for custom systems if needed. However, given the infrequent commits, I think you may not need it. (Regressions will be obvious.)

Downsides of Having Separate Infra

You will need to find someone to write and maintain this infra. ji@ may be able to help find someone. I will be able to help advise.

Depending on the benchmarking setup, DevTools resources will be loaded from file system rather than from a .pak file. This may skew the results a bit.