

Extreme LUD pinpoint results

2024 Mar-Apr

2024 March-April

Yuki Shiino <yukishiino@chromium.org>

Speedometer2

Question: How much regression do we accept when running finch experiments? What sampling frequency do we have for experiments?

Speedometer3

system_health.common_mobile

Conclusion of the discussion (2024 March)

Appendix

Overhead of the allocator shim trampoline

When applied to the browser process only

Pinpoint raw results

mac-m1_mini_2020-perf/speedometer2 (applied to all processes)

android-pixel4-perf/speedometer2 (applied to all processes)

mac-m1_mini_2020-perf/speedometer2 (applied to the browser process only)

android-pixel4-perf/speedometer2 (applied to the browser process only)

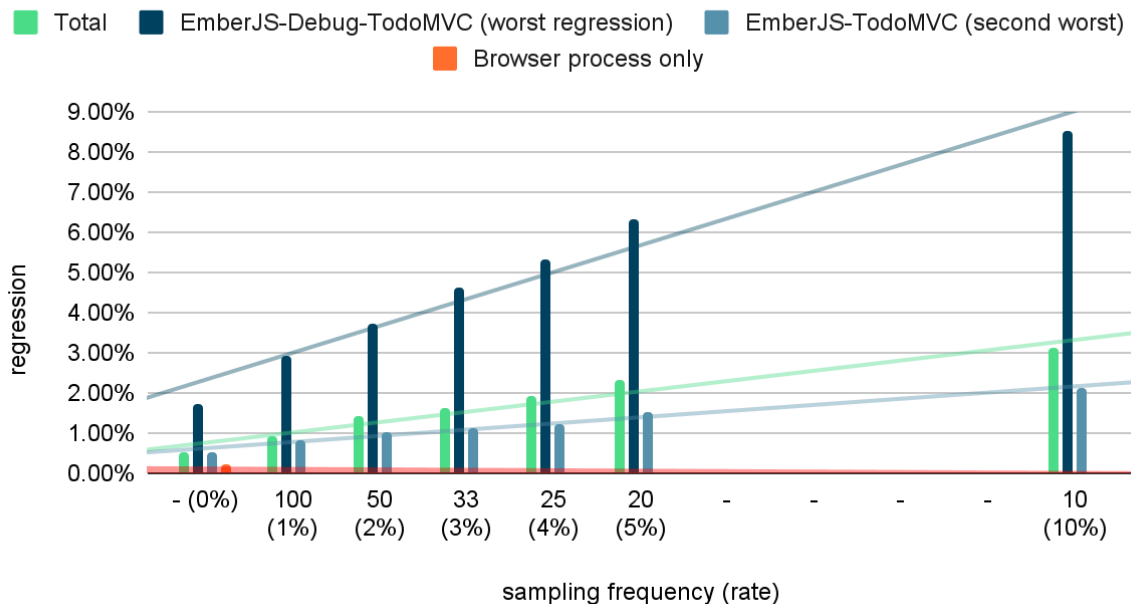
mac-m1_mini_2020-perf/speedometer3 (applied to the browser process only)

android-pixel4-perf/speedometer3 (applied to the browser process only)

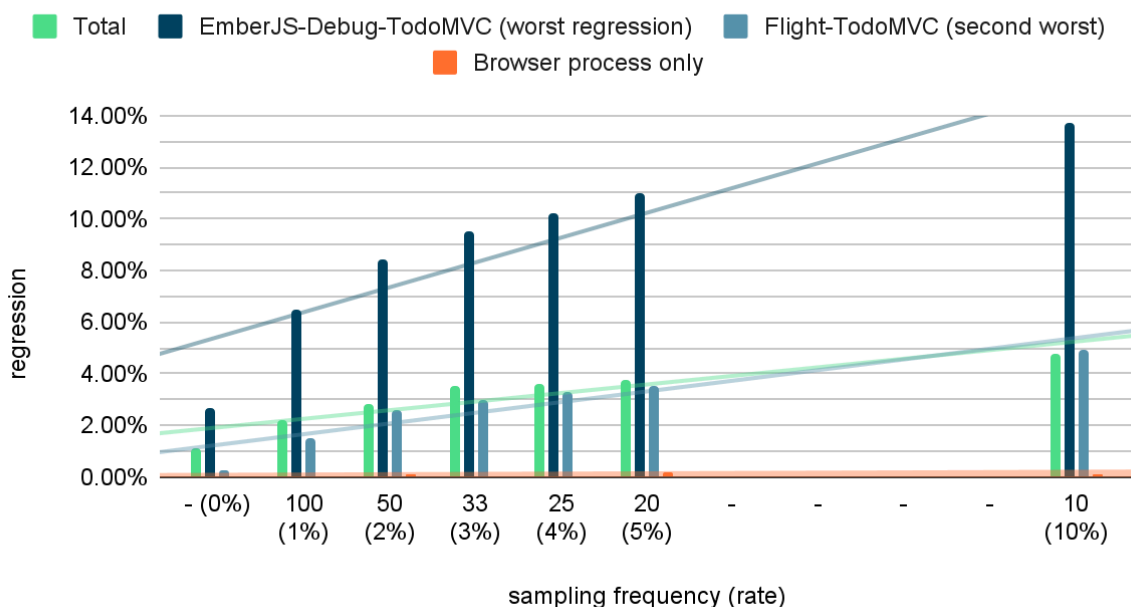
android-pixel4-perf/system_health.common_mobile (applied to the browser process only)

Speedometer2

mac-m1_mini_2020-perf/speedometer2



android-pixel4-perf/speedometer2



Sampling frequency (x-axis) is an experiment parameter, which [is defined as](#)

$$\text{sampling rate} = 1 / \text{sampling frequency}$$

where the sampling rate is of type double and sampling frequency is of type size_t.

The intercept (= regression at 0% sampling) represents the overhead of having the Extreme LUD installed. The overhead comes from the following two points:

1. Having an allocator shim. Even when nothing has been done in the E-LUD, a function call trampoline is needed for all allocator shim functions. E.g. [AllocFn](#). See also [Overhead of the allocator shim trampoline](#).
2. Rolling a dice in `SamplingState::Sample()`. E.g. [FreeFn](#).

Note that although there are a couple of obvious optimization ideas, currently the E-LUD is implemented less invasively to the production code and the existing LUD as the E-LUD is yet just an early experimental idea.

Question: How much regression do we accept when running finch experiments? What sampling frequency do we have for experiments?

Is the sampling frequency 100 (sampling rate 1%) okay for 1% experiments (canary, dev, and possibly beta)? I.e. 1% regression on macOS and 2% regression on Android.

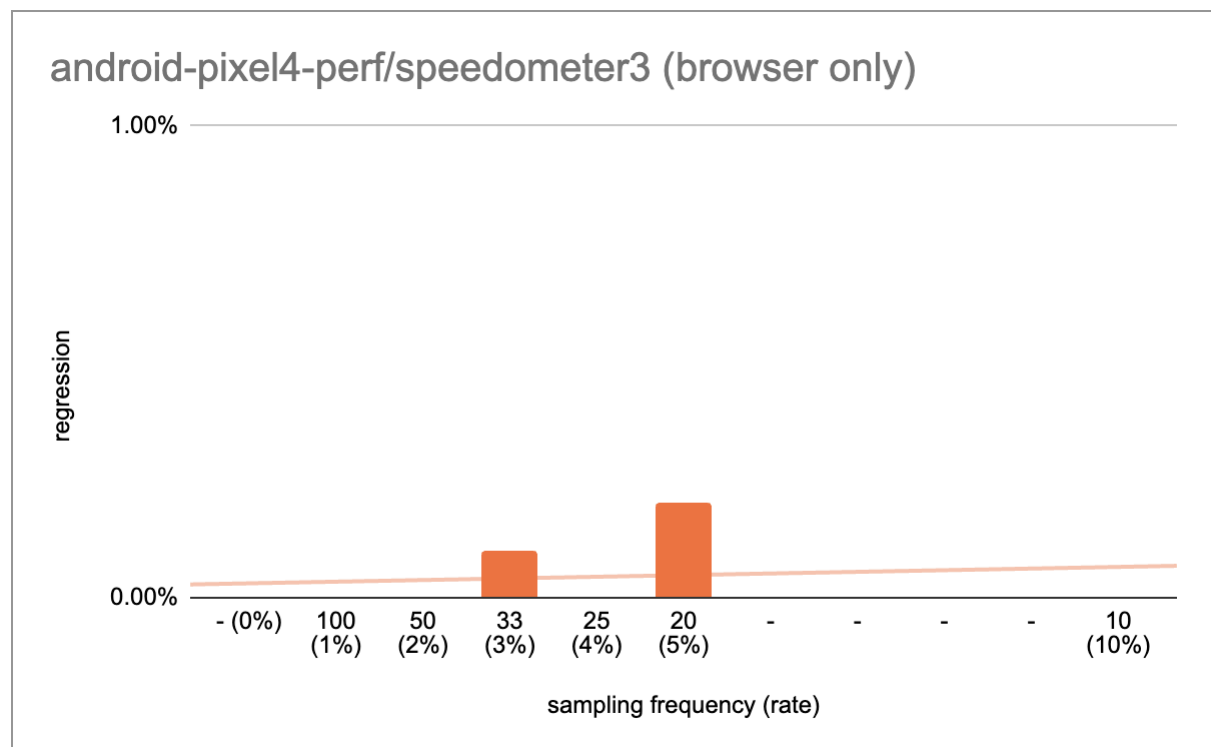
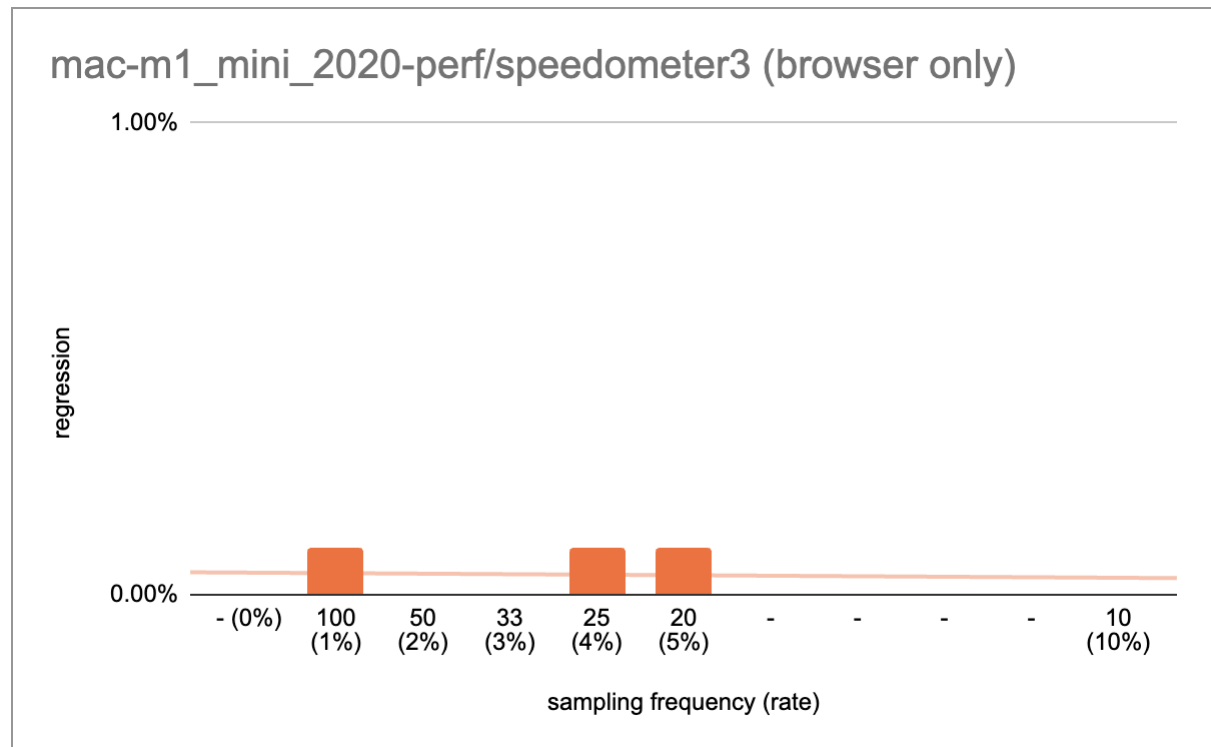
Or less frequently?

If 1% regression on Android is not acceptable, we need to optimize the E-LUD before experimenting.

c.f. default sampling frequencies of GWS-Asan and LUD
([AllocationSamplingFrequency](#) and [default constants](#))

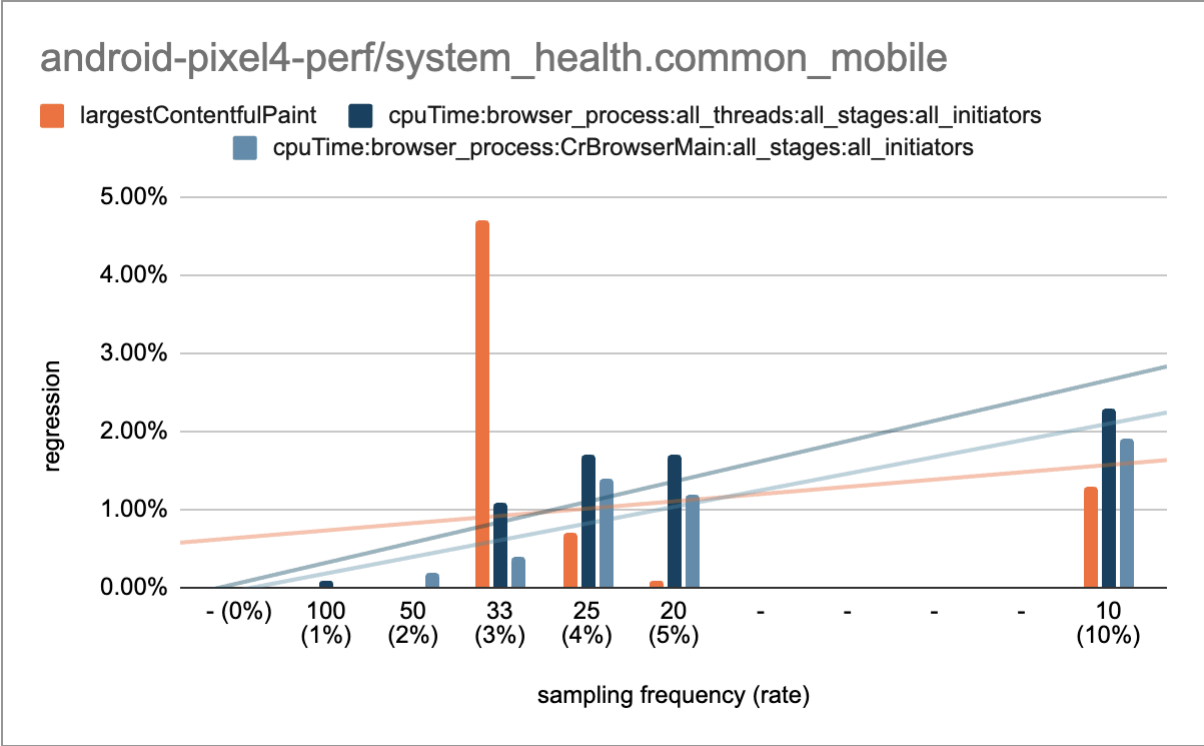
platform	sampling frequency	sampling rate
Android	2000 to 20*2000	0.0025% to 0.05%
Linux/CrOS/Fuchsia	1500 to 16*1500	0.0042% to 0.067%
Others	1000 to 16*1000	0.0063% to 0.1%

Speedometer3



No regression is observed when applied to the browser process only.

system_health.common_mobile



When applied to the browser process only and the sampling rate is 1%, no regression is observed about "largestContentfulPaint" and "cpuTime:browser_process".

Conclusion of the discussion (2024 March)

We're going to run finch experiments only on the browser process. The regression of the speedometer is almost zero when applied only to the browser process (orange line in the graphs above).

Appendix

Overhead of the allocator shim trampoline

When noop allocator shim functions (just as trampolines to the next allocator shim) are inserted to the allocator shim chain [\[CL\]](#), there is almost zero regression on mac-m1_mini_2020-perf [\[pinpoint\]](#) while there are noticeable regressions on android-pixel4-perf [\[pinpoint\]](#).

benchmark	regression
Total	1.5%
EmberJS-Debug-TodoMVC	4.3%
Flight-TodoMVC	0.9%

Compared to the regression of the nearly-zero sampling rate case [[pinpoint](#)], we can conclude that the cost of rolling a dice (`SamplingState::Sample()`) is very close to zero.

P.S. The overhead might be flaky. More experiment pinpoints were run but the overhead doesn't reproduce stably. [pinpoint: [no optimization](#), [optimized](#)] and yet another [pinpoint: [no optimization](#), [optimized](#)]

When applied to the browser process only

No regression is observed on mac M1 [[pinpoint](#)]. Only very small regression is observed on Android Pixel 4 [pinpoint with sampling rate [100%](#), [5%](#), [1%](#)].

c.f. memory regression [pinpoint: [memory_desktop](#)]

Pinpoint raw results

mac-m1_mini_2020-perf/speedometer2 (applied to all processes)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[noop allocator shim functions](#)

android-pixel4-perf/speedometer2 (applied to all processes)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[noop allocator shim functions](#)

mac-m1_mini_2020-perf/speedometer2 (applied to the browser process only)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)

[sampling frequency 10](#)
[sampling frequency 1](#)

android-pixel4-perf/speedometer2 (applied to the browser process only)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[sampling frequency 1](#)

mac-m1_mini_2020-perf/speedometer3 (applied to the browser process only)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[sampling frequency 1](#)

android-pixel4-perf/speedometer3 (applied to the browser process only)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[sampling frequency 1](#)

android-pixel4-perf/system_health.common_mobile (applied to the browser process only)

[sampling frequency 1000000](#)
[sampling frequency 100](#)
[sampling frequency 50](#)
[sampling frequency 33](#)
[sampling frequency 25](#)
[sampling frequency 20](#)
[sampling frequency 10](#)
[sampling frequency 1](#)

