scheduler-dev performance metrics

skyostil@, brianderson@
January 6th, 2016

This document defines the main performance metrics that are used to measure the effectiveness of scheduling changes.

Unless mentioned otherwise, a smaller number is better for each metric.

Main thread smoothness and responsiveness (i.e., Blink scheduler)

- Key mobile sites (smoothness.key_mobile_sites_smooth)
    - first_gesture_scroll_update_latency (how quickly the browser starts scrolling from an idle state. Note that this can be a lot higher during page load compared to the steady state. Target: <50ms)
    - Note: queuing_durations is deprecated because does not differentiate between cases where the main thread is prioritized vs. when it isn't.
- Tough ad cases (smoothness.scrolling_tough_ad_cases)
    - first_gesture_scroll_update_latency (how quickly the browser starts scrolling from an idle state, Target: <50ms)
- UMA
    - Scheduling.Renderer.BeginMainFrameIntervalCritical (interval between main thread frames when the main thread is on the critical rendering path. Target: 60Hz)
    - Scheduling.Renderer.BeginMainFrameToCommitDuration
    - Missing: latency for first scroll update when we scroll on the compositor but had to go via the main thread (most common case after pure compositor scrolling.) -- essentially a subset of Event.Latency.TouchToFirstScrollUpdateSwapBegin
    - Missing: frame latency for touch handler based interactions (as opposed to scrolling)

Compositor smoothness and responsiveness (i.e., cc::Scheduler)

- Key mobile sites (smoothness.key_mobile_sites_smooth)
    - mean_input_event_latency (average latency of input event handling. Target: <16.6ms)
- Tough ad cases (smoothness.scrolling_tough_ad_cases)
    - mean_input_event_latency (average latency of input event handling. Target: <16.6ms)
- Tough scheduling cases (scheduler.tough_scheduling_cases)
    - frame_times (interval between rendered frames. Target: 60Hz)

    - frame_time_discrepancy (amount of variability in frame rendering times. Target: 0)
- UMA
    - Scheduling.Renderer.DrawInterval (time between successive draws. Target: 60Hz)
    - Scheduling.Renderer.CommitInterval (time between successive draws that resulted in a commit. Target: 60Hz)
    - Amount of checkerboarding: Compositing.RenderPass.AppendQuadData.*
    - Missing: Something about swap throttling
    - Finer grained metrics
        - Scheduling.Renderer.PrepareTilesDuration
        - Scheduling.Renderer.CommitToReadyToActivateDuration
        - Scheduling.Renderer.ActivateDuration
        - Scheduling.Renderer.DrawDuration
        - Scheduling.Renderer.SwapToAckLatency