State of renderer task throttling

altimin@chromium.org, skyostil@chromium.org

Throttling concepts

Task alignment

Throttled tasks are aligned to multiples of one second, preventing busy-looping and reducing number of page wakeups.

Budget-based throttling

Throttled task queues share a time budget pool, which regenerates at a constant rate. If time budget level is negative, tasks are not allowed to run and blocked until it regenerates. Each time task from throttled task queue is run, time budget level decreases by the task's wall run time.

Throttling policies

Background timer alignment (since time immemorial)

Timers in backgrounded pages are aligned to 1Hz.

Background timer suspension (since time immemorial)

On Android, timers in background pages are completely suspended after 5 minutes.

Audio cancels throttling (M56)

Recently audible (5 seconds) page is exempt from all forms of throttling. Web pages using timer-driven short audio clips would be broken otherwise.

Offscreen timer throttling (expected in M57)

Timers from third-party offscreen iframes are subjected to task alignment. Zero-by-zero iframes are not throttled because they often have a special meaning.

Expensive background timer throttling (expected in M56/57)

Timers in backgrounded pages are throttled on budget-based principle with 1% CPU limit (10ms regeneration every 1s). This policy supplements background timer alignment.
This policy activates 10 seconds after page is backgrounded (this handles the problem of long navigation-related tasks (window.open) consuming all budget).

Plans for the future

Loading task queue throttling

Loading tasks can't be easily throttled when backgrounded because this will introduce page loading time regression (opening a page and switching to another tab until it loads is a common scenario). However, pages should not be loading forever and we should consider throttling loading tasks after a certain period in background (1 minute, 5 minutes, etc).

Increased throttling aggressiveness after time

It may be a good idea for throttling to become more aggressive as time passes (decreased frequency for alignment and decreased regeneration rate for budget-based throttling).

Rendering frame rate throttling

Currently Chrome always attempts to render at the display's native refresh rate. This can be wasteful, especially during loading, so we may want to experiment with lower frame rates.