This document details the APIs provided by various platforms for timer coalescing.

Android/Linux

Slack can be set for individual threads or for a given process using the PR_SET_TIMERSLACK flag to the prctl() call.

Granularity: Can be set per-process or per-thread.

What this affects: "The timer expirations affected by timer slack are those set by select(2), pselect(2), poll(2), ppoll(2), epoll_wait(2), epoll_pwait(2), clock_nanosleep(2), nanosleep(2), and futex(2) (and thus the library functions implemented via futexes, including pthread_cond_timedwait(3), pthread_mutex_timedlock(3), pthread_rwlock_timedrdlock(3), pthread_rwlock_timedwrlock(3), and sem_timedwait(3))."

Windows

MSDN article on timer coalescing.

Calls to SetWaitableTimerEx() - Win7 and SetWaitableTimer() - win8 allow setting slack for individual Windows timers.

Granularity: Per timer.

What this affects: Just the timer in question.

Notes: It's explicitly documented that the OS coalesces timers across processes.

OS X

Power saving technologies in Mavericks

When using a Core Foundation runloop we can use CFRunLoopTimerSetTolerance() to set slack for an individual timer.

The low level API that backs this is using a timer (EVFILT_TIMER) in a kqueue(), specifying the NOTE_LEEWAY flag.

Granularity: Per timer

What it affects: Just the timer in question.

Notes: On OSX 10.9, the background status of a process figures into coalescing, Chrome does not currently background processes on OS X (crbug.com/383613). We should also look into supporting App Nap better.