Lifecycle use cases

This document enumerates use cases for background tabs, explores existing relevant APIs and outlines ideas for the missing ones.

Relevant documents:

Chromium blog post on throttling — a post announcing budget-based background timer throttling.
Background tabs in Chrome 57 — an article with technical details about budget-based throttling.
Further plans — an approximate roadmap for suspending background tabs.

Background use cases

Background audio

Examples: Spotify, Google Music, Hangouts, YouTube.
Play audio from a backgrounded page while working on something else.
Note there are some sites that have used playing audio to extend their lifecycle:
https://groups.google.com/a/chromium.org/forum/#!topic/blink-dev/XRqy8mIOWps has some details.
Though, like silent audio wouldn't be enough, so you'd need to generate quiet noise.

Solution: allow ServiceWorker to control audio playback (notify when audio track is about to change?) AudioWorker?

Background Sync / Downloading

Examples: GMail?

Solution: ServiceWorkers + unlimited silent push notifications for open tabs + updating tab title/favicon from service worker?

IM clients

Examples: Slack, Hangouts.
Notify user upon a notification: update tab title with a number of unread messages, play sound.

Solution: ServiceWorkers + unlimited silent push notifications for open tabs + playing sound from service worker + updating tab title/favicon from service worker?

Casting/capturing video

Examples: Hangouts

ServiceWorker?

Server hosting

Examples: some p2p games (WebRTC).
Relatively small number of applications use background tab as a server: hosting p2p games, bittorrent, etc. Running an SSH proxy in a background tab.

Allow WebWorker to run freely upon getting an explicit user permission with some Chrome UI indicating an active background tab similar to how android shows a persistent notification for apps running in the background?

Running test suites in the background.

Command-line flag like --disable-background-timer-throttling?

Mining bitcoins or doing some other CPU-heavy simulation.

Same as server hosting — WebWorker + user permission?

Stock ticker monitor

altimin: Isn't it background sync?
P. S. It is a legitimate use case at all?
(=> use a worker + notifications?)

Background WebUSB

go/web-flashstation-design is already running into tab discarding on ChromeOS.

Holding Wake lock

holding system wake lock

Periodic Sync

eg. RSS client updating feed; email feed etc

Upload / Download in background

Letting an upload or download (that was started in foreground) finish in the background.

Loading pages in the background

(e.g., opening several Google Docs -- problem here is IndexedDB transactions that also block the foreground tab)
altimin: Probably it's a good idea to mention it here, but this use case is different from others.

iOS native background use cases

some things are relevant here: eg. bluetooth, voip etc.

End-of-Life use-cases

PageVisibility use-cases (google3)

PageVisibility is used for the following types of work:
- v important for impression tracking in analytics
Update UI State:
- update play state of animation
- tracking when adframe is displayed but page not visible
- redraw (chart) on tab switch
Track UI state for later (when visible):
- chat app - queueing toasts offline
- track pending state (refresh) for later
Misc
- scheduled reporting work

BeforeUnload & Unload use-cases (google3)

A page with a beforeunload handler can return a string, which is used by some sites to cause chrome to prompt the user if they want to leave the page. An example is monorail, where if you've typed into a form, and you leave the page, the site wants to ask "did you really mean to leave?"

I. Analytics & Reporting:

[Handlers: unload; visibility: hidden & visible]
1. Impression tracking: was content viewed
- Google Calendar impression tracking in analytics
- Ads recurring reporting for engagement tracking based on visibility

- punch session impression tracking
2. usage tracking: session length, metrics etc
- Docs reporting on unload to CSI;
- cello final reporting; cello metrics tracking; reports to server periodically and also on unload
- Play: analytics reporting here: update log and periodically report to server

II. Pending state that user has not committed

[Handler: beforeunload]
- Monorail: filing a bug.
- Docs unposted content in comments
- Photos prompts the user if they want to leave before photo upload has finished.
- Photos prompts the user if unsaved changes in editor
- G+ collections prompts if there is stuff in the edit dialog
if beforeunload fires in controller then bail with modal dialog: egs. poll, Square activity , G+ profile open dialog and here
- Admin console - check pending change here
- Docs offline privacy issue: need beforeunload due to privacy implications on untrusted machine -- the app cannot persist state without permissions.

III. Imp wrap up: Release locks; disconnect session etc.

[Handler: unload]
- Docs: release lock on unload
- Chromoting: beforeunload: disconnect remote session; record reason for session end
- Youtube: disconnect from remote connection (and logging)
- AMP: notify viewer on unload
- [Ojan's doc]
- Multi window / frame apps need to clear state in the parent window when the child is closed such as to avoid leaking memory, e.g. when a gmail chat window is closed, the parent needs to be notified so that it can delete any associated data.
- Google Docs wants a quick way to notify collaborators when you've stopped working on a document.
- doodle image cleanup to allow GC

IV. Update UI State and track pending UI state for hidden <-> visible

[Handler: Visibility visible & hidden]
- Ads: update play state of animation
- Ads: tracking when adframe is displayed but page not visible
- redraw (chart) on tab switch
Track UI state for later (when visible):
- chat app - queueing toasts offline
- track pending state (refresh) for later

V. Offline handling

[Handlers: offline / connection check & visibility state]
- chat app - queueing toasts: checks offline & visibility state

Other:
- Ads: single / multi-window enforcement
- gmail fires beforeunload dialog when closing main gmail tab with “child mesage tabs” still open (as they will get killed with main tab)
?- Youtube: check queue data
- Drive app’s browser lifecycle manager;
hook for running all registered callbacks on unload; see callers of beforeunload & unload

Weird cases
Notifications extension and popup handling code
search: remove style
unload fires for download case

Docs offline privacy issue:
need for beforeunload arises due to privacy implications on untrusted machine -- the app cannot persist state without permissions.
OTOH native apps have the advantage of being on a trusted device. We may need to consider new mechanisms here: eg. adding user permission, badging as a clue to the user about pending state etc.

Analytics

philipwalton@ is quoted text
Analytics typically sends a ton of data to its servers during unload or on regular timers.

They currently use sendBeacon(with keepalive) but reporting api is potentially going to be used in the future. There is strong desire for reliable delivery even in the case of tabs being fastkilled or unloaded etc.
"anytime metric information might be lost (killed tab, closed tab, navigate away, etc.) that the application has a chance to either report the data (ideally) or store the data for future reporting."

being able to distinguish that a user is closing a tab, knowing if the browser itself is no longer even the active application, etc.

Sw helps on its own:

> Rather than sending a beacon to a server every time something happens, you could send an event to a service worker. The service worker could then hold on to that data, make sense of it in aggregate, and then only send it to an analytics back end once it's ready. And since service worker can do this in the background, it's OK if the user navigates away before the data is sent.

Background sync is going to really be key here

"I think it's ok to not be 100% accurate [snip] because the point is to look at trends rather than single usage instances."

Counterpoint: how do you tell that when you don't know how many beacons are being lost?

Developer Hacks / API Abuse examples

https://twitter.com/theefer/status/872394400529342465 here's another case of using hacks to keep a page open

https://www.thecssninja.com/html5/stop-ios10-safari-auto-locking

Bad User Experience Reports

https://www.reddit.com/r/chromeos/comments/6hb9en/does_anyone_else_with_a_samsung_pro_find_it_has/

The reporter runs a 20min job in a background tab it seems?

"Every morning I also have to run reports from my companies SaaS CRM system and upload CSV files to Google Sheets. Some of the reports take as long as 20 minutes to run. Never had an issue with my Acer 720 (the 4GB model) but the Samsung Pro reloads the report pages which caused me to have to run a 20 minute report twice which eats up 40 minutes of my time."