Delaying SwapOut until Commit

Charlie Reis, 8-18-2014

Overview

After a recent optimization by clamy@chromium.org for http://crbug.com/323528, Chrome no longer waits for the SwapOut operation to complete in the old renderer process before allowing a cross-process navigation to proceed in the new renderer process. Instead, RenderViewHostImpl has a state machine and waits for both SwapOut and Commit to complete before destroying the renderer. Running SwapOut asynchronously is not a problem for the web platform because a page's unload handler can no longer do anything visible (e.g., modal dialogs).

This change gives us an opportunity to clean up a substantial amount of complexity in Chrome's navigation logic. Since it is already possible for Commit to finish before SwapOut, we can always wait for Commit before trying to SwapOut. This has several advantages:

- There would be no period of time when a RenderViewHost is swapped out but still visible to the user.
- There is no need to pause a request in the network stack on most cross-process navigations, unless a transfer is required (e.g., server redirects).
- Multiple states can be removed from the RenderViewHostImpl state machine.

Timeline Diagrams

The following timeline diagrams show how much complexity can be removed:

Cross-Process Navigations (no transfer):

- Current timeline - In this diagram, we pause the request in the network stack to start running SwapOut and the unload handler. However, we don't wait for the ack and immediately resume the request. We must wait for both SwapOut and Commit before killing the old renderer process in WasSwappedOut.
- Proposed timeline - After the proposed changes, we don't need to pause the request. We combine SwapOut and WasSwappedOut into a single event that happens after Commit.

Cross-Process Navigations (with transfer):

- Current timeline - This is similar to the current timeline above, but it adds the ability to transfer to a different destination renderer process while the request is paused in the network stack.
- Proposed timeline - The proposed changes keep the ability to transfer processes (e.g., after a server redirect), but no SwapOut logic is necessary until the transfer has completed and the navigation has committed.

Summary of Changes

The proposed changes are implemented in https://codereview.chromium.org/464593003/ for http://crbug.com/402020, and the highlights are listed below.

- **Swap out at commit time:**
  - RFHM::SwapOutOldPage is called from CommitPending.
  - RFHM::SwappedOut goes away.
  - Most tests don't need to simulate OnCrossSiteResponse or SwapOutACK.
- **No pausing for most navigations:**
  - CrossSiteResourceHandler only pauses responses when a transfer is needed.
  - CrossSiteRequestManager and HasPendingCrossSiteRequest go away, since the IO thread doesn't need to know if a cross-process navigation is in progress.
  - RFHM::OnCrossSiteResponse is only used for transfers.
  - RFHM::pending_nav_params_ goes away, and only the CrossSiteTransferringRequest is stored.
  - RDH::ResumeDeferredNavigation is now only used for transfers that don't end up needing a new RenderFrameHost.
- **Fewer states:**
  - STATE_WAITING_FOR_COMMIT and STATE_WAITING_FOR_UNLOAD_ACK are no longer possible.
  - Several tests of these states are no longer relevant.
- **No WasSwappedOut:**
  - RenderFrameImpl::OnSwapOut can directly exit the process rather than waiting for a separate WasSwappedOut message, since the new RenderFrame is already visible.
- **Transfers to and from the same process don't repeat the request:**
  - We can simply resume in NavigatorImpl rather than issuing a duplicate request. This avoids several confusing DidFail, DidStop, and DidStart events.