

The Tale of Child-allocated LocalSurfacelds, Main/Impl Threads, and New Metadata

cblume@

Problem statement:

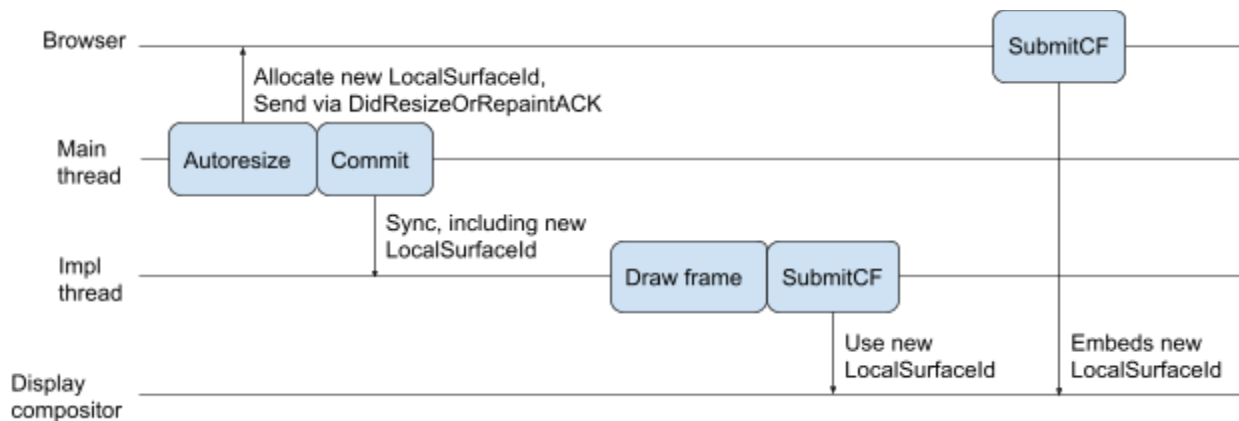
In order to support all of Viz on Clank, we need a method for the renderer process to inform the browser process of scrolling and device orientation changes. This allows the browser process to update its Omnibox and selection handle positions.

Right now, the browser implicitly knows this information because it is the one to allocate LocalSurfacelds. We want to use child-allocated LocalSurfacelds. They offer many benefits like reduced IPC traffic and thus lower latency.

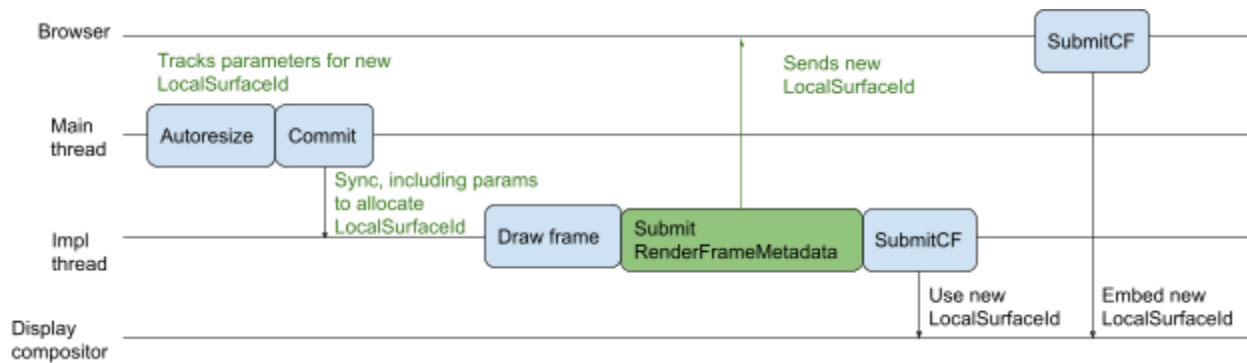
Proposed solution:

Move the child-allocation to the Impl thread & send metadata from there to the browser process.

When child-allocated LocalSurfacelds land, the event sequence will be:



After moving the child-allocation to the Impl thread & adding metadata, the event sequence will be:



Expected regressions:

When using the existing RenderFrameMetadata frequently there are performance regressions. These are inherent to the design as a result of the IPC latency. Since full Viz support requires sending this type of metadata, that same regression is expected.

Work breakdown:

1. Move some of CompositorFrameMetadata into RenderFrameMetadata. Things which should be sent to the Browser such as top/bottom control heights should be moved.
2. Move [RenderFrameMetadata](#) sending to right after the scroll position is set for that frame.
3. Add the LocalSurfaceId to the RenderFrameMetadata and route it to the browser's ParentLocalSurfaceIdAllocator. The RenderFrameMetadata should get the LocalSurfaceId from the LayerTreeHostImpl.
4. Move the child allocation from the Main thread to the Impl thread. This requires Commit sending the params for the LocalSurfaceId's allocation. Those params are parent_sequence_number (uint32_t) and embed_token (base::UnguessableToken). Those are coming from a parent-generated LocalSurfaceId. It may be cleaner to just send the whole LocalSurfaceId.

Follow-up work:

When navigating, it would be nice for the Main thread to be able to allocate and not have to sync with the Impl thread. We may allow the child-allocation to happen on both Main and Impl using our existing optimistic, eventually-consistent system. Although, couldn't it just be a flag to notify the Impl thread we navigated, since the Impl thread still needs to submit the CF?

The RenderFrameMetadata could be sent earlier if we know we aren't scrolling. This would improve the expected performance regression. There will still be a fallback to send RenderFrameMetadata later (after the new scroll position is set for this frame) if either the scroll just began or this frame continues a previous scroll.