Custom Display List 🚀 Plan

enne@ (January 2017)

First read: [Blink/CC Custom Display List Explainer]
Also see: https://codereview.chromium.org/2523673004

tl;dr

Replace Blink and ui's use of SkPicture with a new custom display list backing to store recorded paint commands while keeping the painting API mostly unchanged. The larger goals are to move towards Vulkan rasterization and help the compositor find more images in painted content.

LANDING PLAN

1. Sanity check raster+record performance on cluster telemetry
2. Land typedef patch (no logic changes, rename types to new types)
3. Land API patch (backend still Skia, but using new concrete classes)
4. (Optional) Land implementation behind #ifdefs in pieces if too large
5. Turn On! 💡

BIKESHEDDING CLASS NAMES AND FILE LOCATIONS

This patch reimplements SkPicture and friends in Chromium code and so there's a lot of new classes that need to be named. Here's an initial proposal of names, the patches above will use the final names on the right:

SkCanvas -> CdlCanvas -> PaintCanvas
SkSurface -> CdlSurface -> PaintSurface (plan to remove this eventually)
SkPictureRecorder -> CdlPictureRecorder -> PaintRecorder (matches ui::PaintRecorder)
SkPaint -> CldPaint -> PaintFlags
SkPicture -> CdlPicture -> PaintRecord
SkShader -> CdlShader -> Paint(Image|Picture|Wrap|)Shader
SkLiteDL -> CdlPictureBuffer -> PaintOpArray

It's not great to namespace everything in the class name itself with "Paint". However, these names aren't too long and there's too many generic "canvas" or "surface" or "shader" classes in the codebase to not differentiate.

New code will live in cc/paint/, which will have include restrictions via deps to match Blink (e.g. no base, etc). Headers will be added to Blink platform to typedef cc types into the Blink namespace. Blink core can then use these "Blink" classes in paint code. Et voila.

PROTOTYPE CHANGES REQUIRED FOR LANDING

Peephole optimizations in CdlPictureRecorder

SkPicture has a number of SkRecord optimizations that need to be replicated somewhere in the painting chain. In particular, these are SkRecordNoopSaveRestores (which remove save/restores with nothing between them) and SkRecordNoopSaveLayerDrawRestores (which turn a save with opacity + draw into a draw with opacity). These can be accomplished with some additional logic in the CdlPictureRecorder or CdlPictureBuffer itself.

The picture recorder should also calculate op count and whether or not any op has been appended that would veto GPU rasterization.

Fix shader lifetime issues

The prototype generates shaders at draw time, which helps with serialization. However, draw time currently happens on raster worker threads and it's desirable to keep the raster source const during this process. A simplification would be to generate the real SkShader at record time instead. In the future, shaders can just be removed entirely and serializing them can be sidestepped entirely.

Revert PlatformCanvas/CdlSurface uses back to Skia where pixels needed

The prototype changes a number of places from using Skia to using the new custom display list (webrtc, image capture, pepper, browser font resources, transport dib). These really are just trying to manipulate pixels and so do not need custom display lists. If code needs to generate pixels, it can ask Skia directly. If code needs to generate a deferred recording of paint commands, it can use custom display lists.

Write AnalysisCanvas replacement

Currently, the compositor detects solid colors by playing back SkPictures into an overridden SkCanvas class that looks at ops. If the storage format for the paint commands is transparent, then this needs to be rewritten to walk through the ops directly. This would allow us to remove the abort callback and not have to support it in CdlPicture.

Miscellaneous long tail

There a number of other small issues that need to be fixed too:

- Add annotation support to CdlPicture.

- Add a flag to hijack canvas to support skipping images.
- Add approximate bytes used to CdlPicture.

NON-EXHAUSTIVE LIST OF FOLLOW UP WORK

Start on canvas command buffer

The biggest work item that this unlocks is the canvas command buffer work. It's desirable to write the command buffer against custom display lists instead of against SkPictures, so landing custom display lists is a dependency for that work to start.

Let Skia team know about removed features

After this patch, Chromium will no longer need a number of Skia features, including abort callbacks, approximate op count, and number of slow paths. These were all added because the compositor needed more control and information about SkPictures.

Remove GetSkCanvas calls

It is shippable to have CdlCanvas be able to be converted into an SkCanvas where needed to do queries, however I think this blocks future refactoring. It'd be desirable to remove these uses. To do this, drawBitmapRect would need to be implemented and some top level info (color type, bounds) would need to be added to the CdlCanvas. Things that are using CdlCanvas for pixels now should be converted to use Skia instead.

Change refcounting to single ownership, where possible

Most of these Skia data structures are all refcounted with sk_sp but are singly owned, but many of them do not need to be.