BlinkGenPropertyTrees

Launch retrospective for GPU Brown Bag

philip rogers (pdr), June 12 2019

BlinkGenPropertyTrees (BGPT)

- Background
- Launch retrospective
  - Correctness
  - Performance
  - Takeaways
- Future

Compositor data structures

Companion talks about property trees:

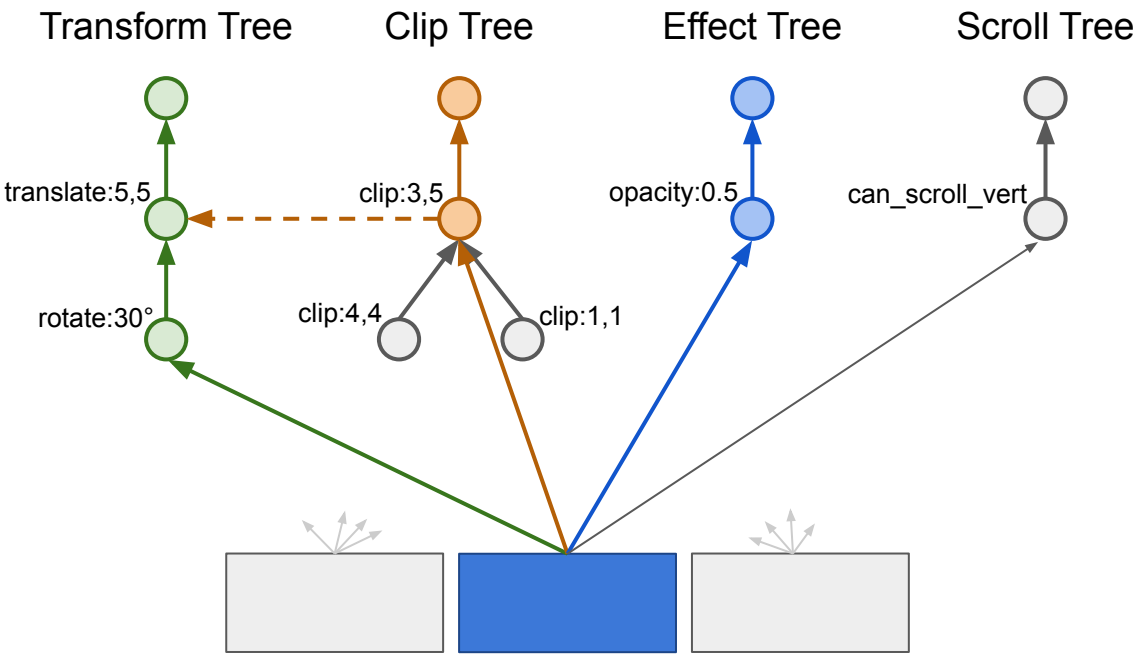
Compositor Property Trees (ajuma, 2017)
Blink Property Trees (pdr, 2017)

How cc works (enne, 2019)

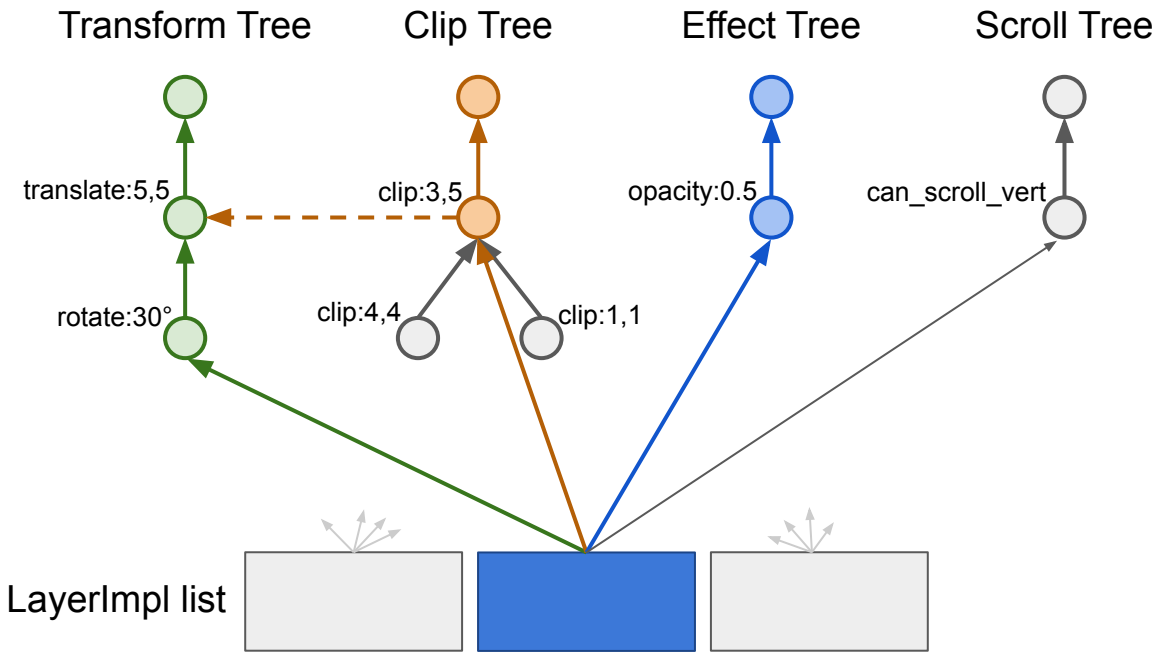
Compositor data structures

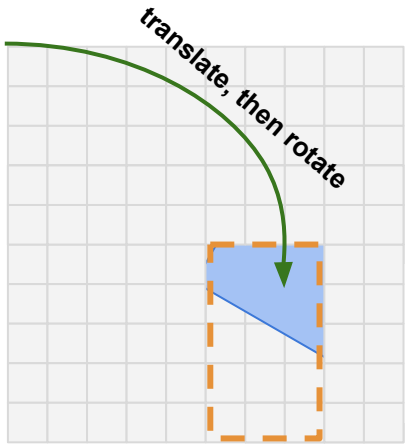
Property trees

LayerImpl list

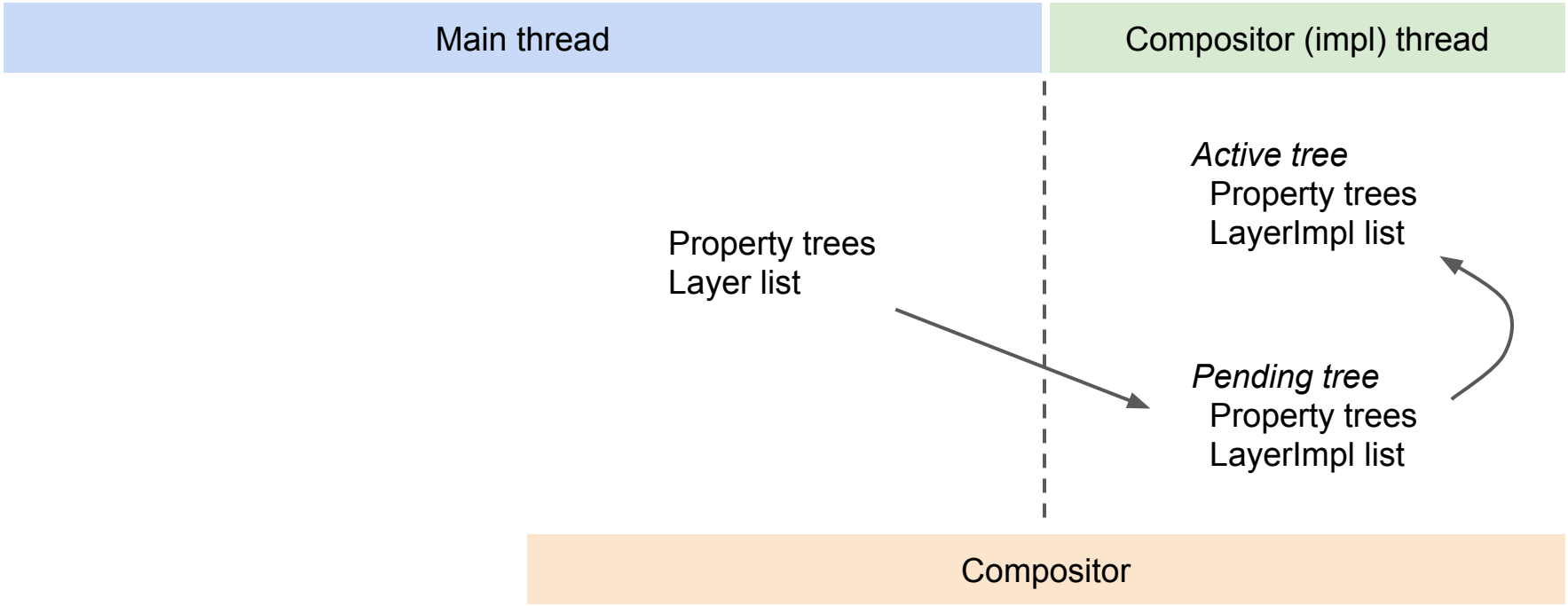


Compositor data structures

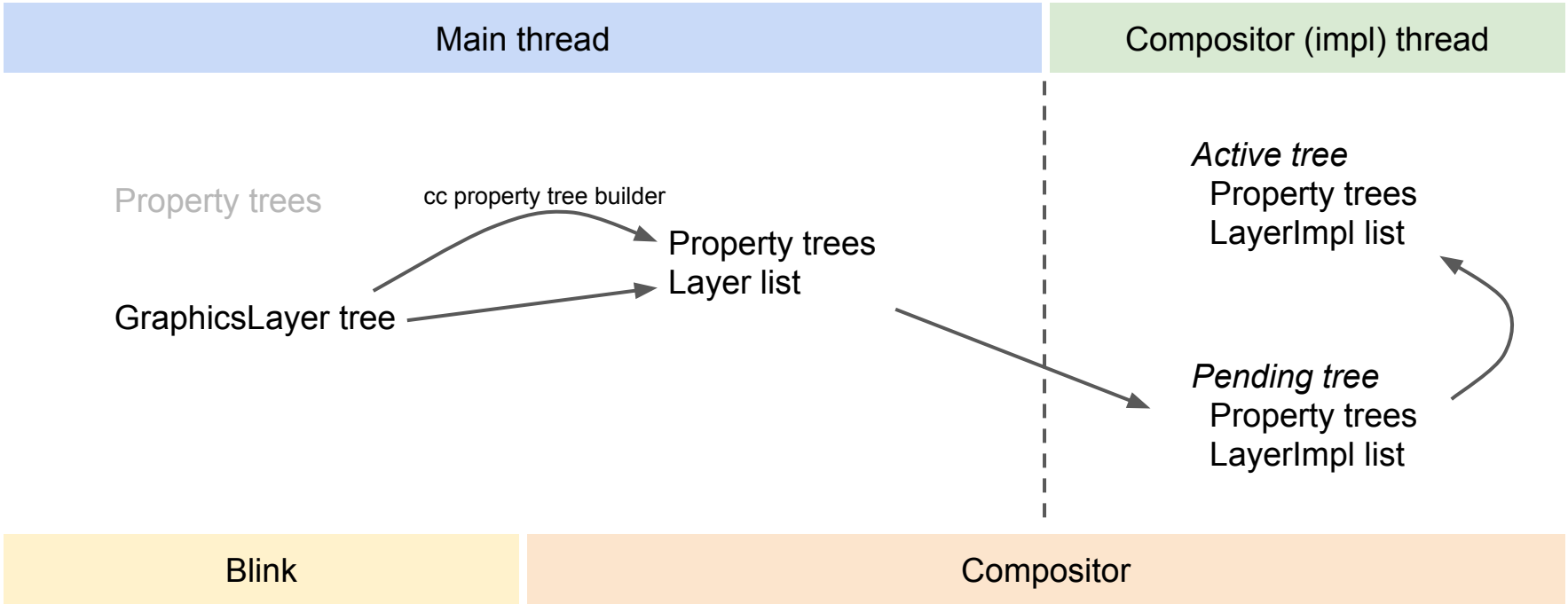




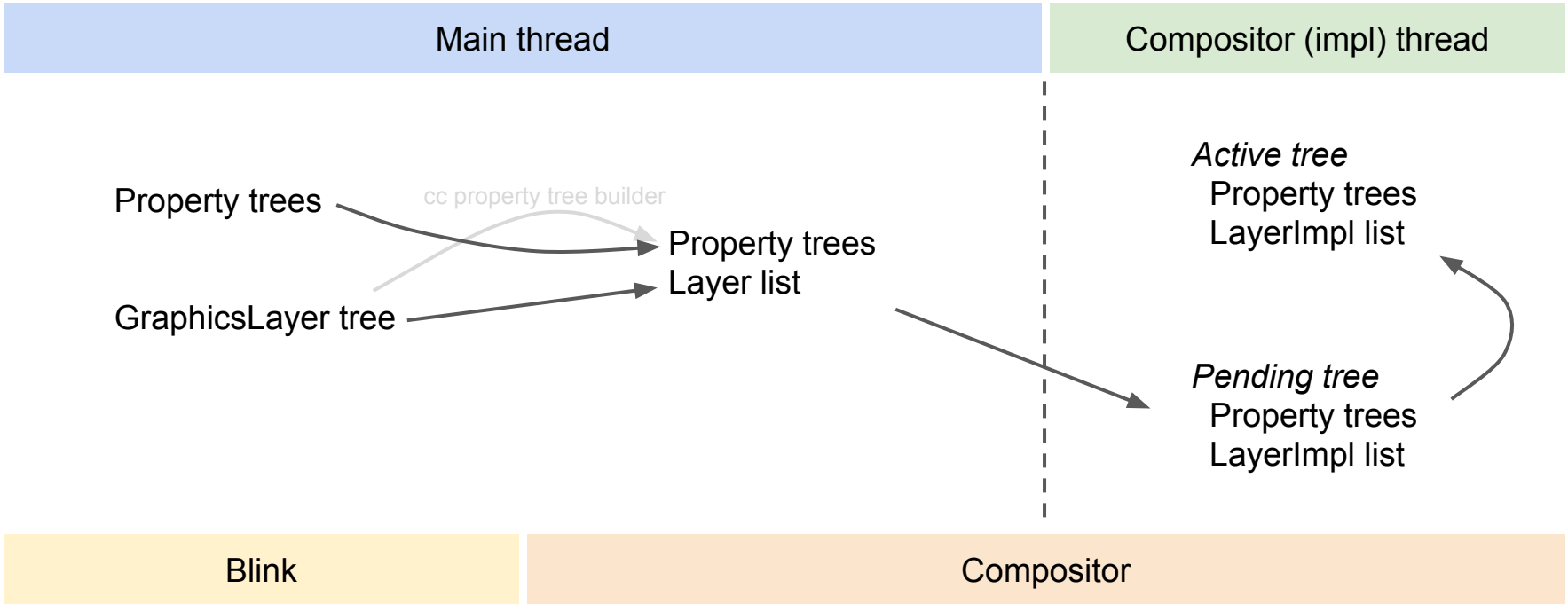
Data flow


Main thread
Compositor (impl) thread
Property trees
Layer list
Active tree
Property trees
LayerImpl list
Pending tree
Property trees
LayerImpl list
Compositor

Data flow (pre-BGPT)



Data flow (post-BGPT)



BlinkGenPropertyTrees

Skip the cc property tree building step

Blink sends cc the final cc::Layer list and cc::PropertyTrees

Launch process

1) Correctness
2) Performance

Strategic mistake to treat these as independent

Used blink RuntimeEnabledFeature controlled via a finch configuration.

Launch process

		web (layout) tests	blink perf tests	blink unit tests[1]	browser perf tests	browser unit tests[1]	Cluster fuzz
Field trial testing config disabled	**RuntimeEnabledFeature Status = test**	On					
	RuntimeEnabledFeature Status = experimental	On	On				
	RuntimeEnabledFeature Status = stable	On	On	On			
Field trial testing config enabled	**RuntimeEnabledFeature Status = stable**	On	On	On	On	On	On

[1] "blink unit tests" include, blink_platform_unittests, blink_unittests, others..
"browser unit tests" include content_browsertests, others..
There is a third category of unit tests that use the default browser-side flag value.

Launch process: RuntimeEnabledFeature

(April 28, 2018): Added BlinkGenPropertyTrees RuntimeEnabledFeature

... got web tests passing ...

(Oct 21, 2018): Promoted BlinkGenPropertyTrees to experimental

Launch process: RuntimeEnabledFeature

(April 28, 2018): Added BlinkGenPropertyTrees RuntimeEnabledFeature

... got web tests passing ...

(Oct 21, 2018): Promoted BlinkGenPropertyTrees to experimental

(Oct 23, 2018): Promoted BlinkGenPropertyTrees to experimental

(Dec 6, 2018): Promoted BlinkGenPropertyTrees to experimental

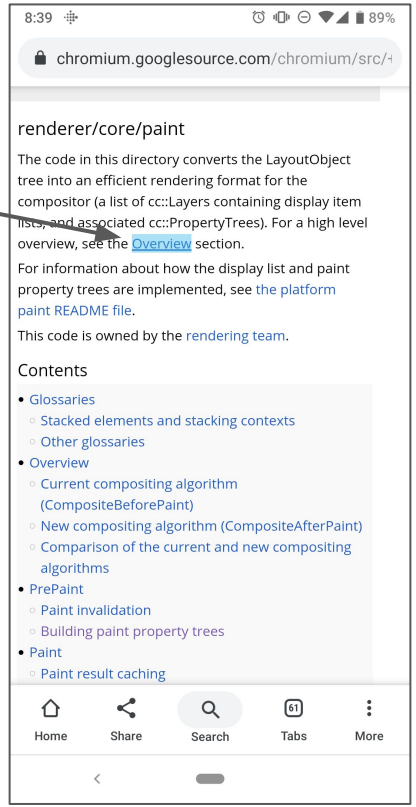
(Jan 12, 2019): Promoted BlinkGenPropertyTrees to experimental

Launch process: RuntimeEnabledFeature

Examples of difficult issues: link highlights

- Android-only highlight drawn around links
  - Immediate feedback that navigation is occuring
  - Each GraphicsLayer has a vector of highlights, each with quads
  - Fades out using compositor-driven opacity animation
  - Was managed outside of blink lifecycle

- BlinkGenPropertyTrees required refactoring feature
  - Moved into paint lifecycle step
    - Needed to be before blink->cc cc::Layer list push
    - Needed to be after PrePaint for correct geometry
  - Explicitly specified effect state: now draws on top of everything



Launch process: RuntimeEnabledFeature

Examples of difficult issues (continued):

- Blink and cc property tree hierarchical differences
  - Blink did not special-case the root scroller (see: RootLayerScrolling)
  - Cc skipped the clip for the root scroller
  - Introducing an extra clip had no user-visible effects
  - Broke google books, headless chrome, and large internal google customer

- Devtools, pinch-zoom, android url bar hiding, animations, extensions color overlay ...

Launch process

		web (layout) tests	blink perf tests	blink unit tests[1]	browser perf tests	browser unit tests[1]	Cluster fuzz
Field trial testing config disabled	**RuntimeEnabledFeature Status = test**	On					
	RuntimeEnabledFeature Status = experimental	On	On				
	RuntimeEnabledFeature Status = stable	On	On	On			
Field trial testing config enabled	**RuntimeEnabledFeature Status = stable**	On	On	On	On	On	On

[1] "blink unit tests" include, blink_platform_unittests, blink_unittests, others..
"browser unit tests" include content_browsertests, others..
There is a third category of unit tests that use the default browser-side flag value.

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed
- (Jan 29): Landed field-trial testing configuration
  - No more coverage of non-BGPT codepath for perf tests

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed
- (Jan 29): Landed field-trial testing configuration
  - No more coverage of non-BGPT codepath for perf tests
- (Jan 29): crbug.com/926327 filed with 13 perf regressions on desktop

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed
- (Jan 29): Landed field-trial testing configuration
  - No more coverage of non-BGPT codepath for perf tests
- (Jan 29): crbug.com/926327 filed with 13 perf regressions on desktop
- (Jan/Feb): More regressions filed and merged into metabugs

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed
- (Jan 29): Landed field-trial testing configuration
  - No more coverage of non-BGPT codepath for perf tests
- (Jan 29): crbug.com/926327 filed with 13 perf regressions on desktop
- (Jan/Feb): More regressions filed and merged into metabugs
- (Feb 14): GPU perf bot outage ends

Launch process: performance

- (Jan 12): Final promotion of BlinkGenPropertyTrees to experimental
  - Begin limited coverage of the non-BGPT codepath using linux-only BGPT-disabled trybot
  - No more coverage of non-BGPT codepath for blink perf tests
- (Jan 18): GPU perf bot outage begins, but goes unnoticed
- (Jan 29): Landed field-trial testing configuration
  - No more coverage of non-BGPT codepath for perf tests
- (Jan 29): crbug.com/926327 filed with 13 perf regressions on desktop
- (Jan/Feb): More regressions filed and merged into metabugs
- (Feb 14): GPU perf bot outage ends
- (Feb): Even more regressions filed and merged into metabugs

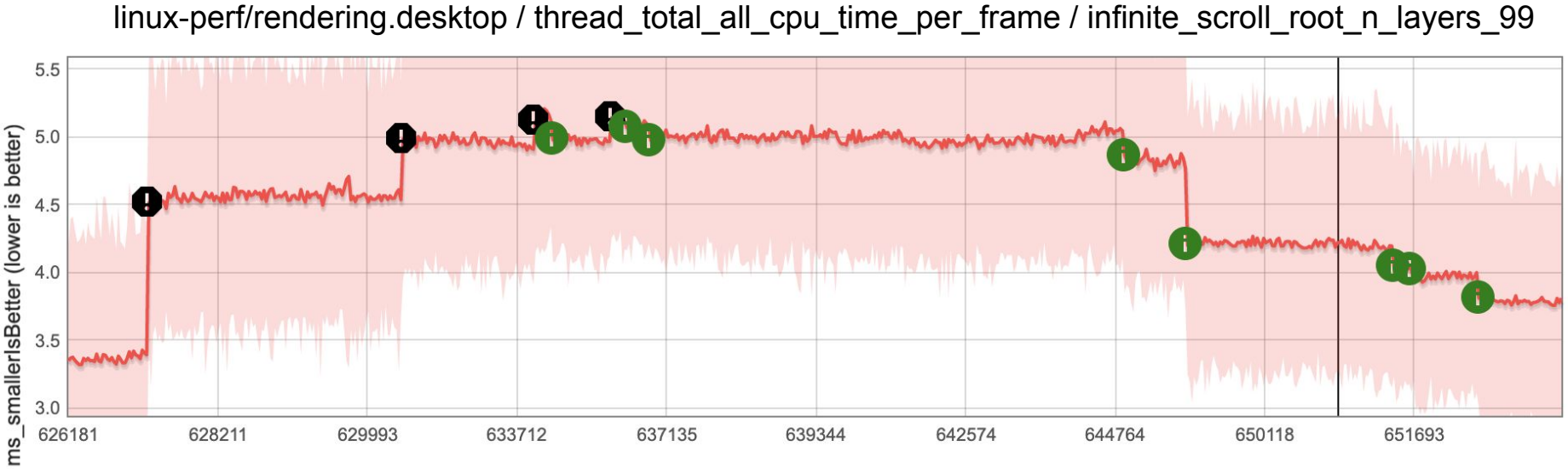
Launch process: performance

Manual triage:

- Spreadsheet with 300 tests
- 175+ pinpoint jobs run

Perf graph

Launch process: performance



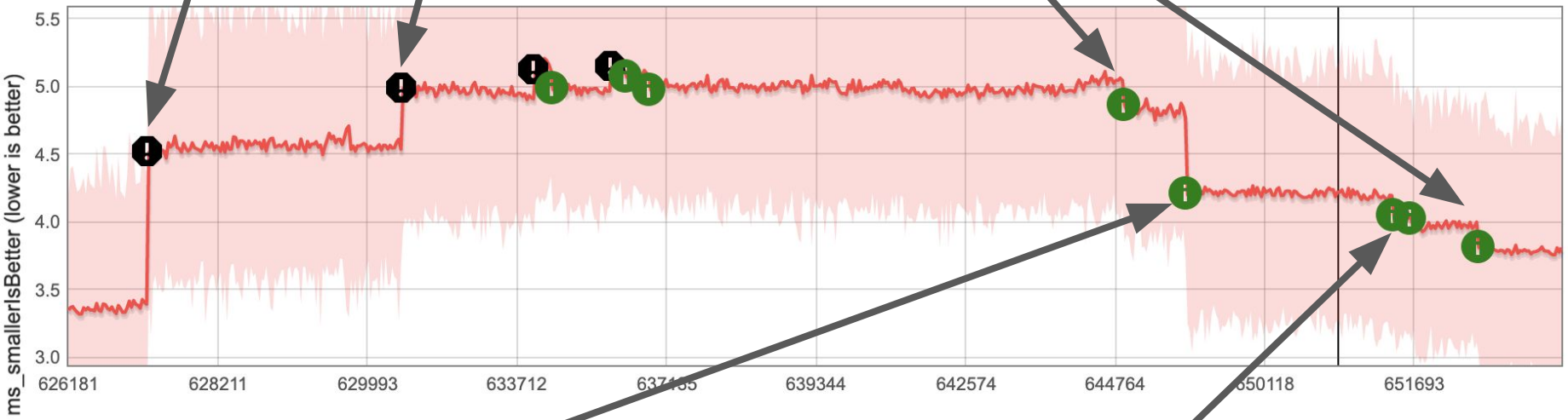
BlinkGenPropertyTrees regression

Perf graph

r645372: Don't collect element ids of scrollbar layer effect for animation (wangxianzhu)

r652620: Make BGPT de-composite trivial transforms (masonfreed)

Metrics change (ignore this)



r647866: Directly update ScrollTranslation nodes. (vmpstr)

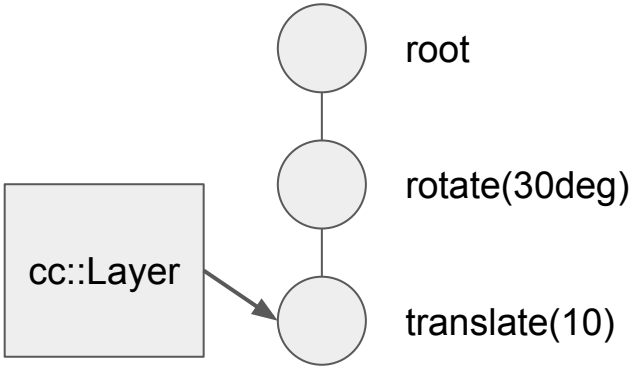
r648068: Remove no-op compositing scroll offset updates (bokan)

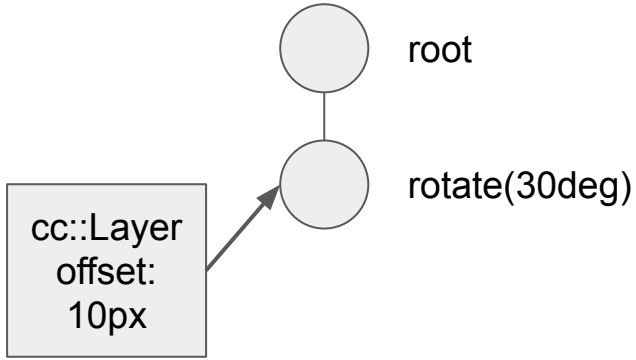
r651338: Add a fast-path for simple value transform changes. (vmpstr)

Launch process: performance

Examples of interesting performance optimizations:

- Direct property node updates skip major steps of the blink lifecycle
  - E.g., changing a CSS transform value from javascript skips everything after PrePaint
  - Implemented by tracking cc::PropertyNode in blink::PaintPropertyNode
- Simple transforms nodes associated with only one cc::Layer are "decomposited" by baking the offset into the cc::Layer:





Launch process: performance

Examples of interesting performance optimizations (continued):

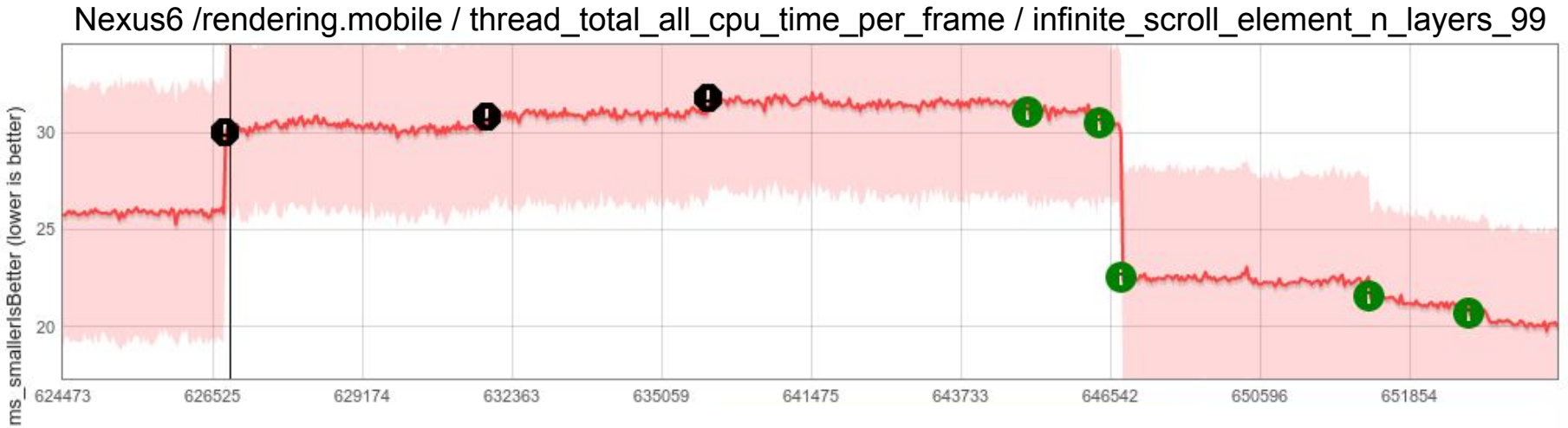
- BlinkGenPropertyTrees created more render surfaces
  - UMA data showed this, but difficult to find locally
  - Ran all web_tests with and without BGPT using a script, compared render surface counts
  - Breakthrough when Xianzhu created ad-hoc metrics which compared counts on top 10k pages
  - Due to: border-radius differences, directly-composited images, non-drawable layers, and an optimization to avoid render surfaces for grouping when there is only one layer in the group
  - 50th: +1.00%, 95th: +2.83%

Perf graph

Launch process: performance

Examples of interesting performance optimizations (continued):

- BlinkGenPropertyTrees can scroll faster
  - No longer needs a scroll-relative offset in cc::Layer
  - Can skip compositing updates when scroll offsets change



Launch process: performance

BlinkGenPropertyTrees did ~8% more page loads.

Finch population skew: see finch-users post for more information.

The "Actives" tab can be used to see population sizes if you suspect this.

Launch process: takeaways

Correctness and performance are not independent problems: test both early.

Lean towards treating unexplained perf failures like unit test failures and reverting.

Minimize difference between what users run and what bots test.

Launch process: ideas for better launches

- Experiment with benchmark prioritizations
  - How to proceed with some progressions and some regressions?
    Prioritize large regressions?
    Try composite benchmarks (see: tech talk)
- Field trial guidance is not recommended for this type of change
  - Independently enable whether change runs on perf bots
  - Consider running pinpoint jobs with feature enabled for major suites (rendering.mobile, etc)
  - Setup custom perf bot for non-default-tested configuration to de-risk flag flipping
  - Experimental waterfall (vmiura, dpranke)
- Independently enable clusterfuzz
  - Update (July 2, 2019): this can now be done, email mbarbella@google.com to set this up.

BlinkGenPropertyTrees

Launching in M75

BlinkGenPropertyTrees one pager

BlinkGenPropertytrees performance analysis

LayoutNG block-flow launch plan

Check if your browser has BlinkGenPropertyTrees
https://bgpt.rentals

BlinkGenPropertyTrees UMA results

Blink.MainFrame.UpdateTime: flat

RenderSurface regression: FastBorderRects finch trial landed today

Compositing.Renderer.LayersUpdateTime:
Sum: -23.06%, Count: -15.78%, 50th: -23.14%, 95th: 0.87%

Compositing.Renderer.NumActivePictureLayers:
Sum: -34.78%, Count: -4.28%, 50th: -32.69%, 95th: -32.49%

Post-launch tasks

Simplify cc::Layer, cc::LayerImpl, cc property trees
  Unify property tree structures

Large dependency on cc property tree builder for unittests (275 tests)
    auto-convert using c++?

ui/views will still use cc property tree builder

Long term code health?

Future

CompositeAfterPaint

Fixes fundamental bug in architecture

Make compositing decisions on compositor thread

Moves work off main thread

Raster inducing scroll

Separate threaded scrolling from composited scrolling

BlinkGenPropertyTrees

Chris Harrelson (chrishtr), David Bokan (bokan), Jianpeng Chao (chaopeng), Mason Freed (masonfreed), Majid Valipour (majidvp), Peter Mayo (petermayo), Philip Rogers (pdr), Robert Flack (flackr), Sahel Sharifymoghaddam (sahel), Stephen McGruer (smcgruer), Tien-Ren Chen (trchen), Vladimir Levin (vmpstr), Xianda Sun (sunxd), Xianzhu Wang (wangxianzhu), Xida Chen (xidachen)

BlinkGenPropertyTrees one pager

BlinkGenPropertyTrees performance analysis

LayoutNG block-flow launch plan