BlinkOn: Performance

September 24, 2013

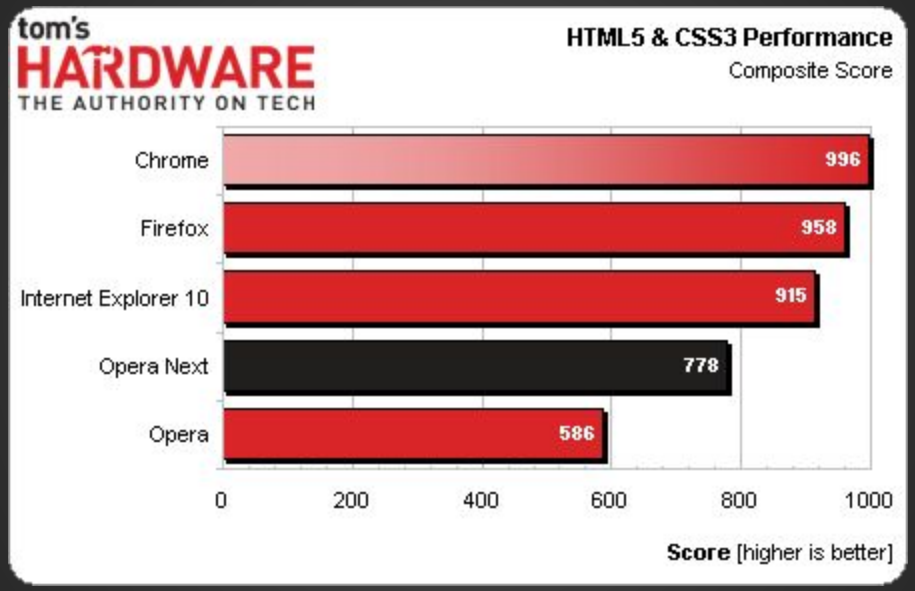
simonjam@chromium.org
tonyg@chromium.org

Moderator Page:

http://goo.gl/hfgp4L

What is Performance?

Performance is Great!



Performance as users see it

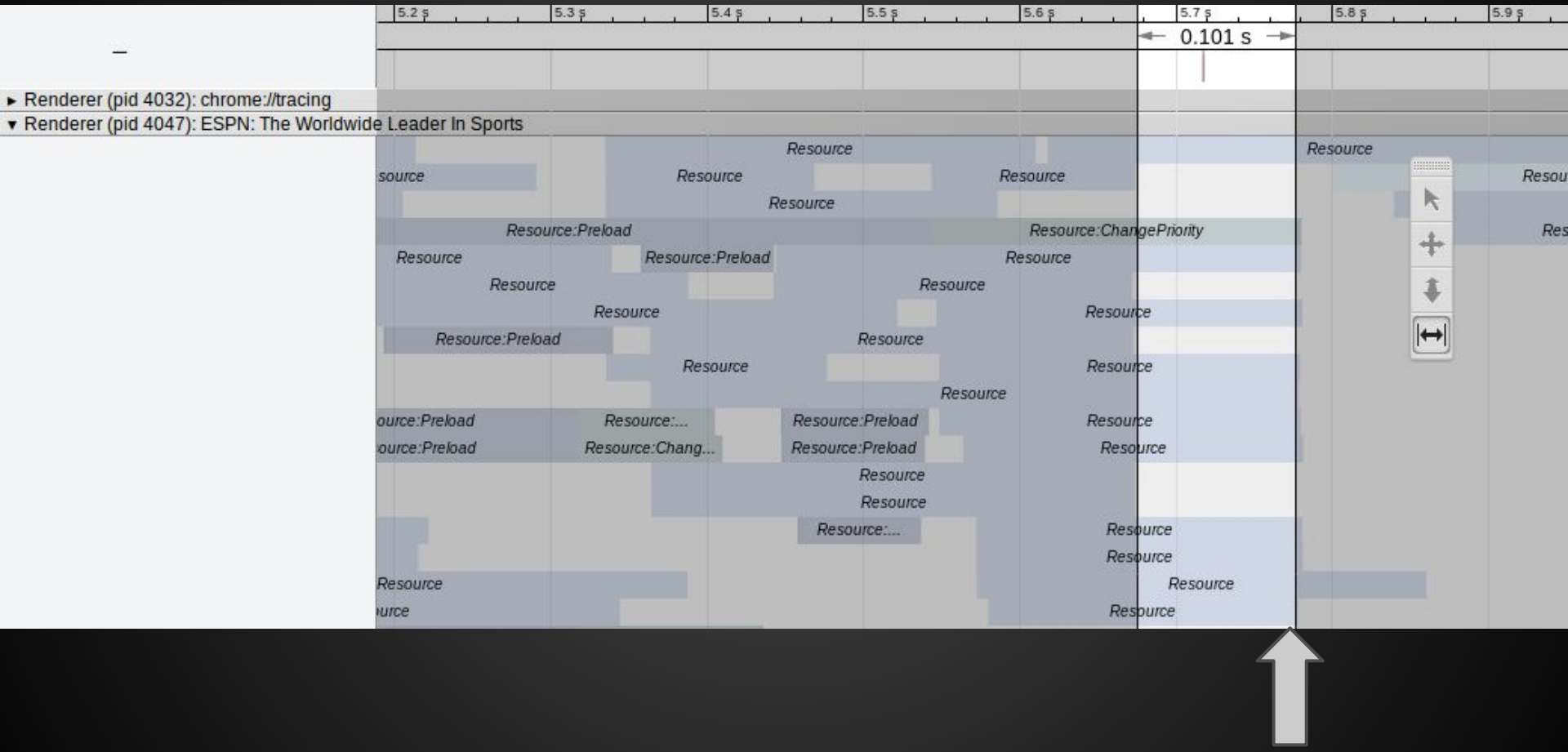
http://www.oilevent.com/

What went wrong?

chrome://tracing

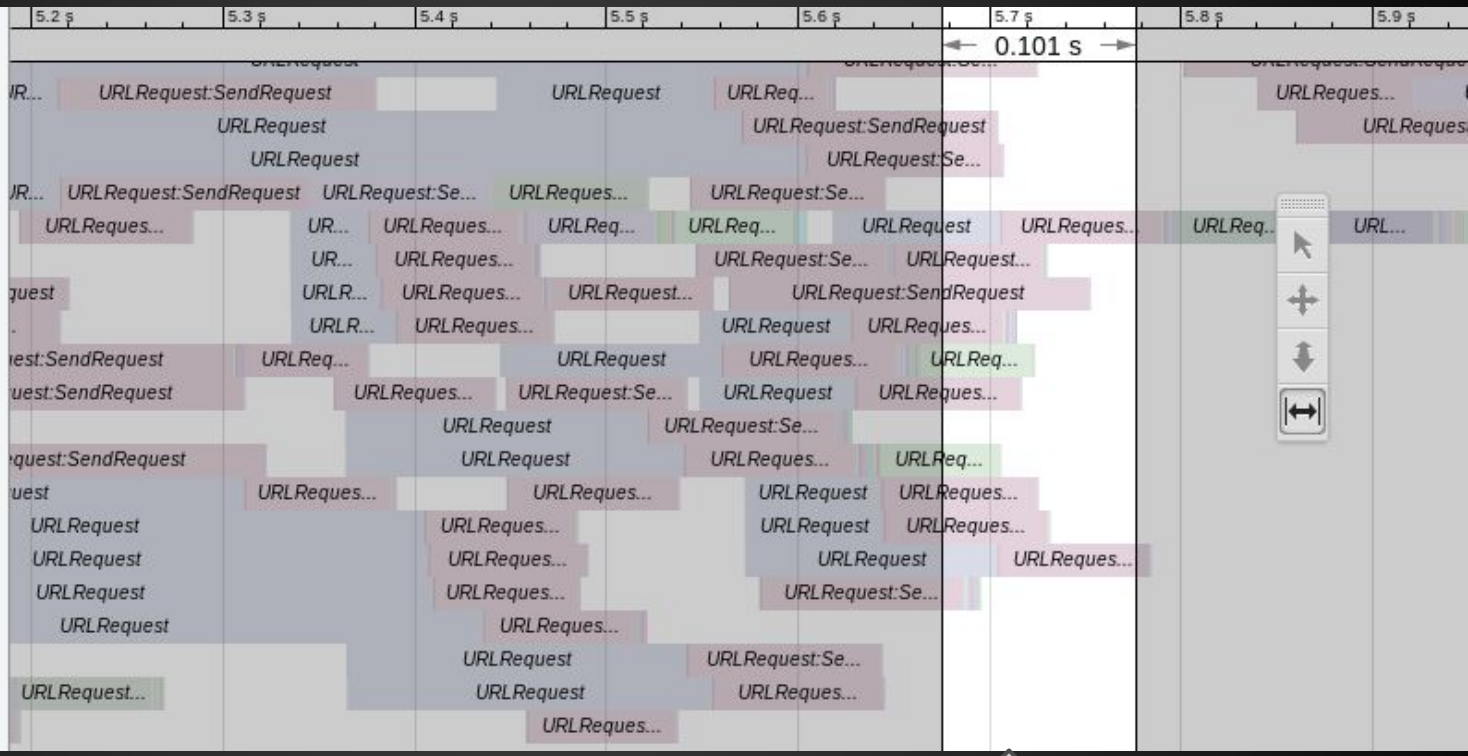
Documentation

Tracing Page Loads



Many resources finish at the same time.

Tracing Page Loads

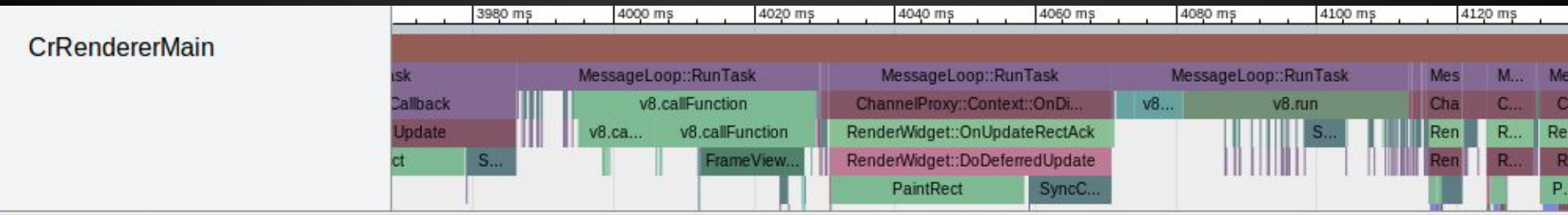


According to the browser process, they finished much sooner.

Tracing Page Loads

The main thread was busy!

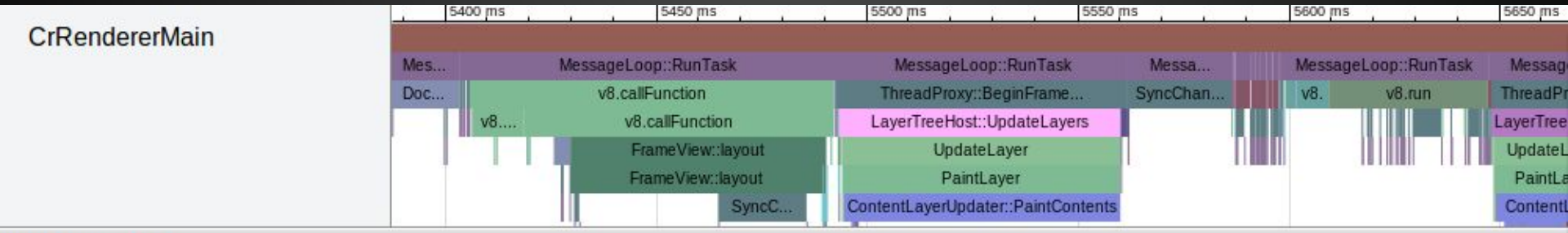
Comparing Traces



Selected slice:

Title	"RenderWidget::DoDeferredUpdate"
Category	"renderer"
Start	"4030.743 ms"
Duration	"40.074 ms"

Software Rendering Path

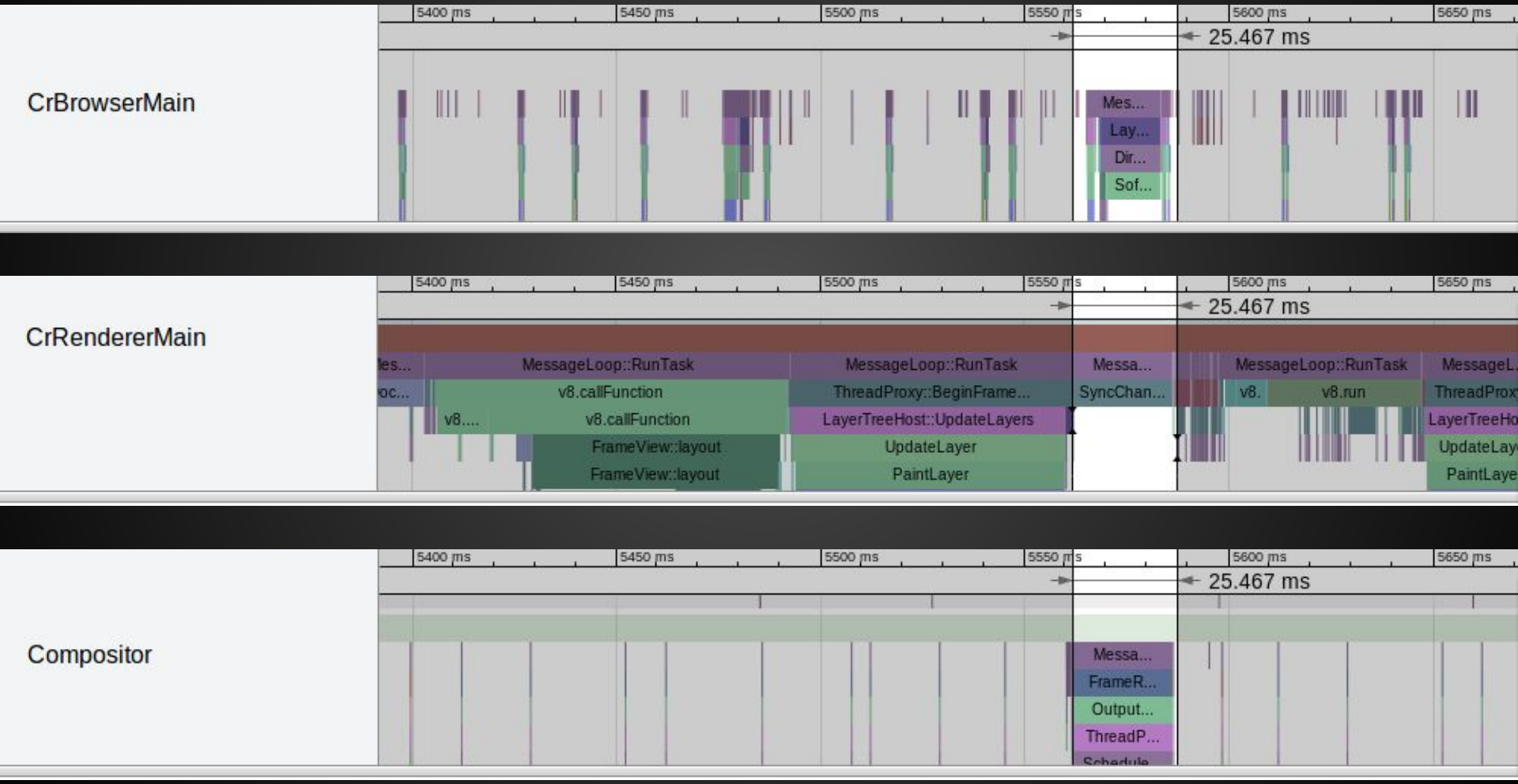


Selected slice:

Title	"LayerTreeHost::UpdateLayers"
Category	"cc,benchmark"
Start	"5493.123 ms"
Duration	"66.841 ms"
Args	
source_frame_number	2

Composited Rendering Path

Tracing Contention



Adding Tracing

Synchronous event (e.g. layout, paint)

- Measures the duration of its scope.

```
 30 #include "core/page/FrameView.h"
 31 #include "core/platform/Logging.h"
 32 #include "core/platform/chromium/TraceEvent.h"
 33 #include "core/workers/WorkerGlobalScope.h"
 34 #include "core/workers/WorkerLoaderProxy.h"
 35 #include "core/workers/WorkerThread.h"
 36 #include "public/platform/Platform.h"
 37 #include "weborigin/SecurityOrigin.h"
 38 #include "weborigin/SecurityOriginHash.h"
 39 #include "wtf/Assertions.h"
 40 #include "wtf/CurrentTime.h"
 41 #include "wtf/MathExtras.h"
 42 #include "wtf/TemporaryChange.h"
efore | Expand all | Expand 10 after
562     for (;;) {
563         ResourceMap::iterator i = m_resources.begin();
564         if (i == m_resources.end())
565             break;
566         evict(i->value);
567     }
568 }
569
570 void MemoryCache::prune()
571 {
572     TRACE_EVENT0("renderer", "MemoryCache::prune()");
573
```

Adding Tracing

Async event (e.g. network)

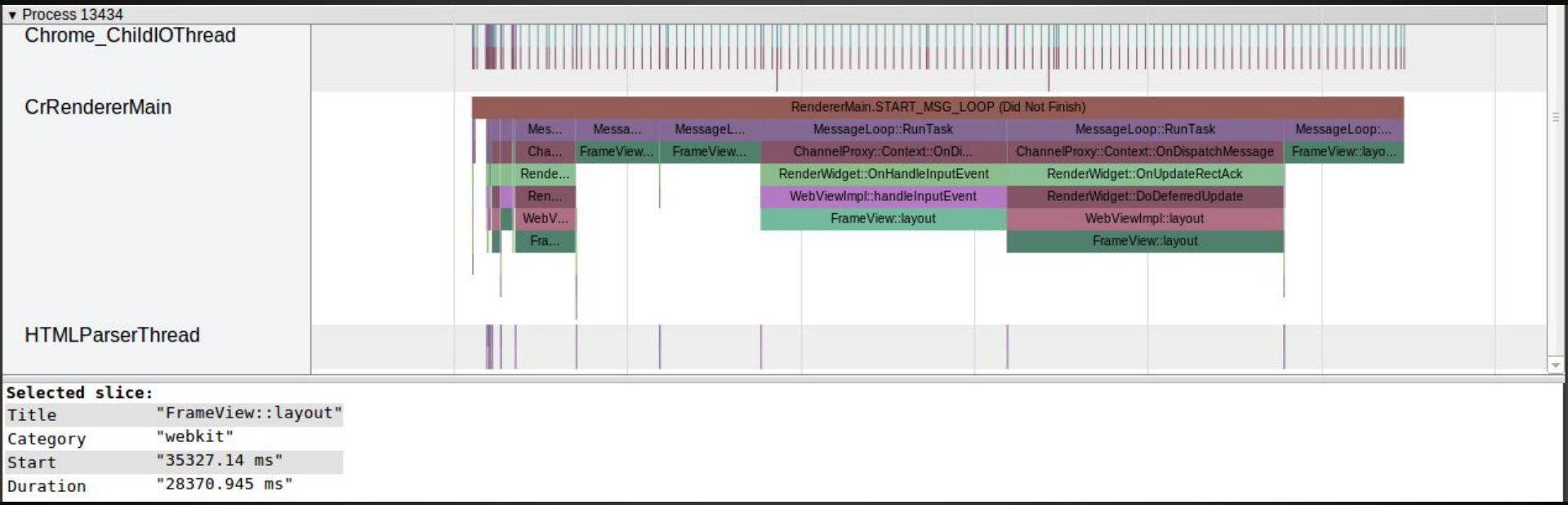
- Measures explicitly defined steps
- Tracks using a unique identifier

```
737     storeResourceTimingInitiatorInformation(resource, request);
738     TRACE_EVENT_ASYNC_BEGIN2("net", "Resource", resource.get(), "url", resource-
    >url().string().ascii(), "priority", resource->resourceRequest().priority());
739     return resource;
740 }
```

```
1070     if (!resource || (m_preloads && m_preloads->contains(resource.get())))
1071         return;
1072     TRACE_EVENT_ASYNC_STEP0("net", "Resource", resource.get(), "Preload");
1073     resource->increasePreloadCount();
```

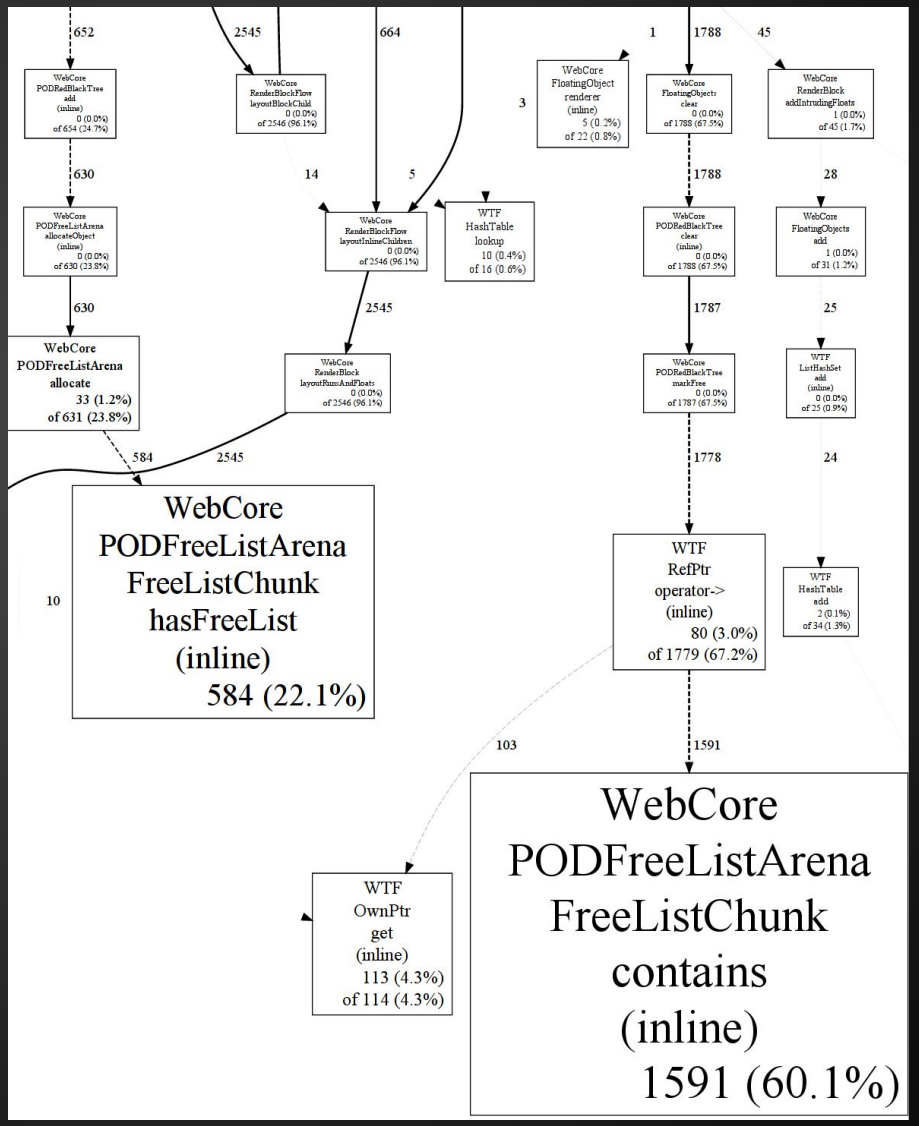
```
1138 {
1139     TRACE_EVENT_ASYNC_END0("net", "Resource", resource);
1140     if (options.sendLoadCallbacks != SendCallbacks)
1141         return;
```

What's wrong with oilevent.com?



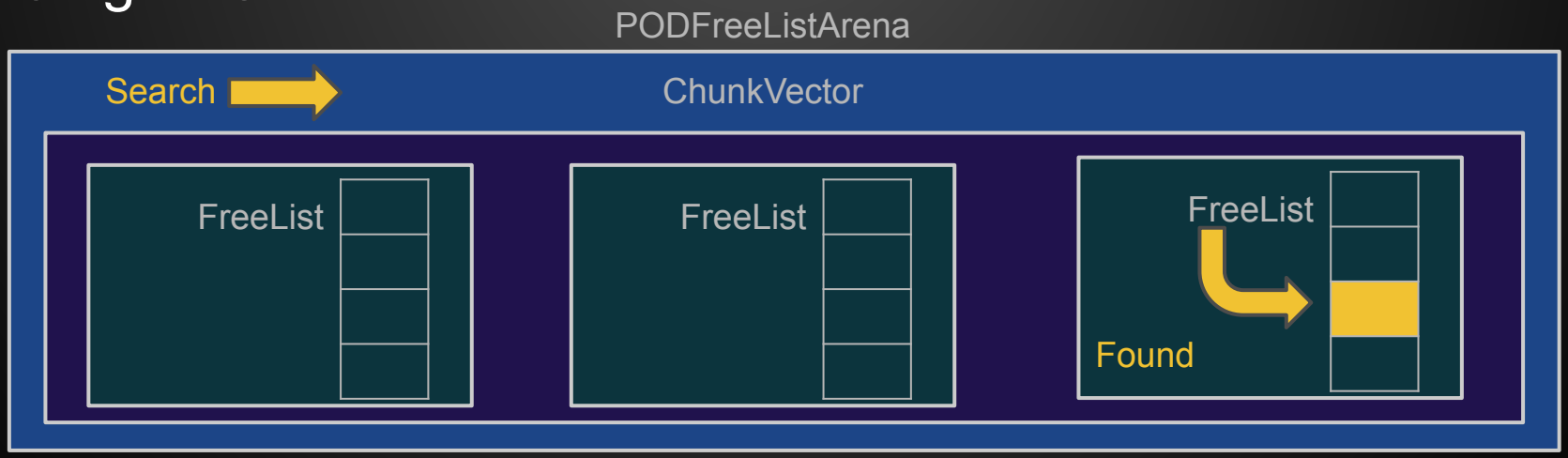
Profiling

[Documentation]



Analysis for oilevent.com

- 6,932 floating elements
- Each has a node in the placedFloatsTree
- Every layout, we free everything in the tree
- PODFreeListArena has multiple freelists in a ChunkVector
- Linearly searches for each freed tree node and adds it to the right list



Fixing oilevent.com

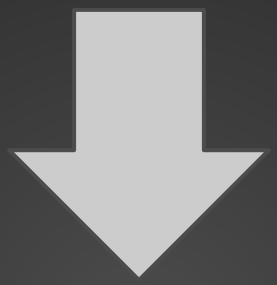
```
60      void freeObject(T* ptr)
61      {
62          ChunkVector::const_iterator end = m_chunks.end();
63          for (ChunkVector::const_iterator it = m_chunks.begin(); it != end; ++it) {
64              FreeListChunk* chunk = static_cast<FreeListChunk*>(it->get());
65              if (chunk->contains(ptr))
66                  chunk->free(ptr);
67          }
68      }
```

```
73      void freeObject(T* ptr)
74      {
75          FixedSizeMemoryChunk* oldFreeList = m_freeList;
76
77          m_freeList = reinterpret_cast<FixedSizeMemoryChunk*>(ptr);
78          m_freeList->next = oldFreeList;
79      }
```

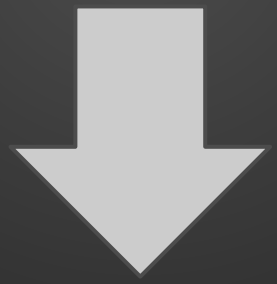
Patch from simonhatch@chromium.org

Telemetry

Microbenchmark



Turn a web page into a benchmark



Turn the internet into a benchmark

Microbenchmark

```
$ tools/perf/run_measurement blink_perf \

third_party/WebKit/PerformanceTests/Parser/url-parser.html
```

Turn a web page into a benchmark

```
$ tools/perf/record_wpr http://google.com/
```

Run that benchmark

```
$ tools/perf/run_measurement <MEASUREMENT> \
  tools/perf/page_sets/www.google.com.json
```

Available Measurements:

loading_profile, loading_timeline, loading_trace, media, memory, page_cycler, smoothness

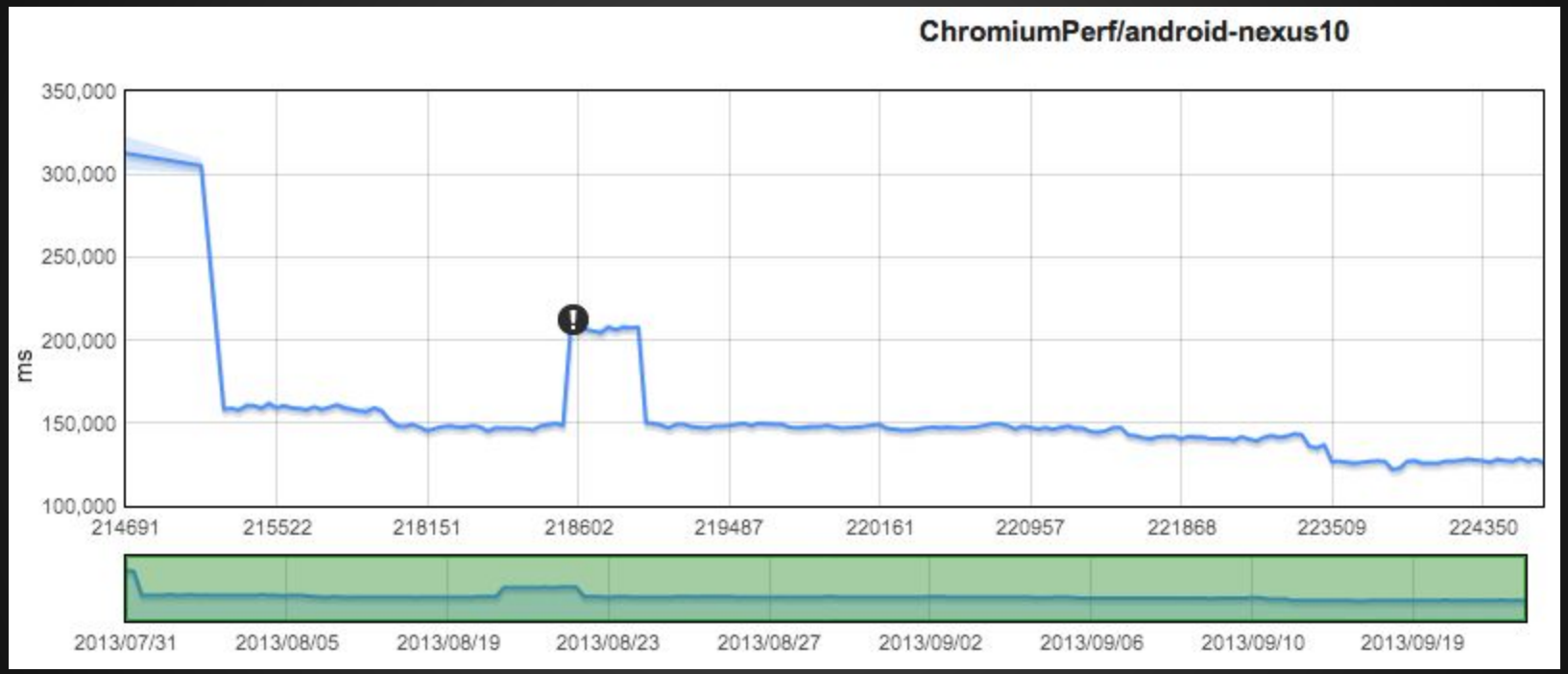
Profile that benchmark

```
$ tools/perf/run_measurement page_cycler \
  --profiler=<PROFILER> \
  tools/perf/page_sets/www.google.com.json
```

Available Profilers:

android-memreport, iprofiler, java-heap, monsoon, perf, sample, strace, tcmalloc-heap, tcpdump, trace, vtune

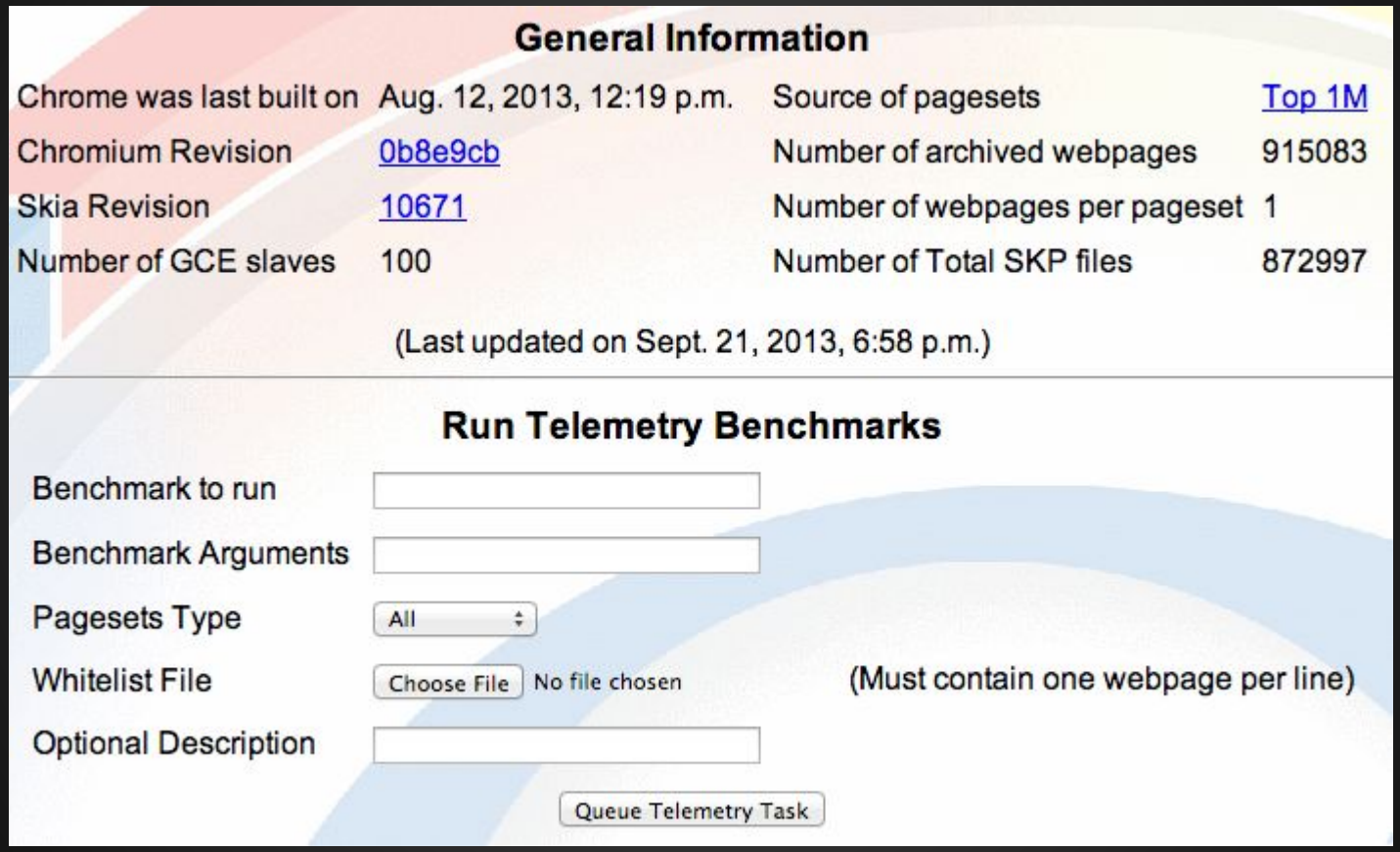
Monitor that benchmark



“Tough Layout Cases”

http://chromeperf.appspot.com/

Turn the internet into a benchmark



http://skia-tree-status.appspot.com/skia-telemetry

What to optimize?

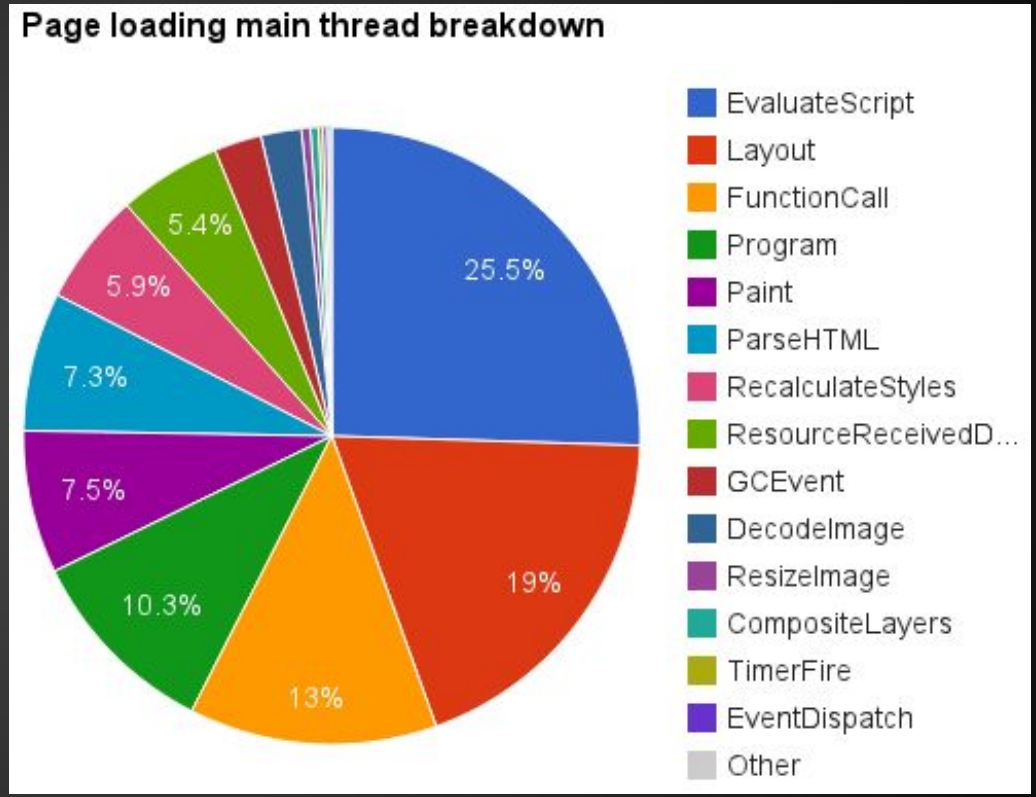
Give the main thread to developers

- Improves page load time
- Reduces jank
- Improves responsiveness

Top million sites - Linux

#1, #3 are JS
Good! Devs have 39%
of the main thread.

#2 Layout (19%)
#4 Uncategorized (10%)
#5 Paint (8%)
#6 HTML Parser (7%)
#7 Style Recalc (6%)
#8 Resource Receive Data (5%)



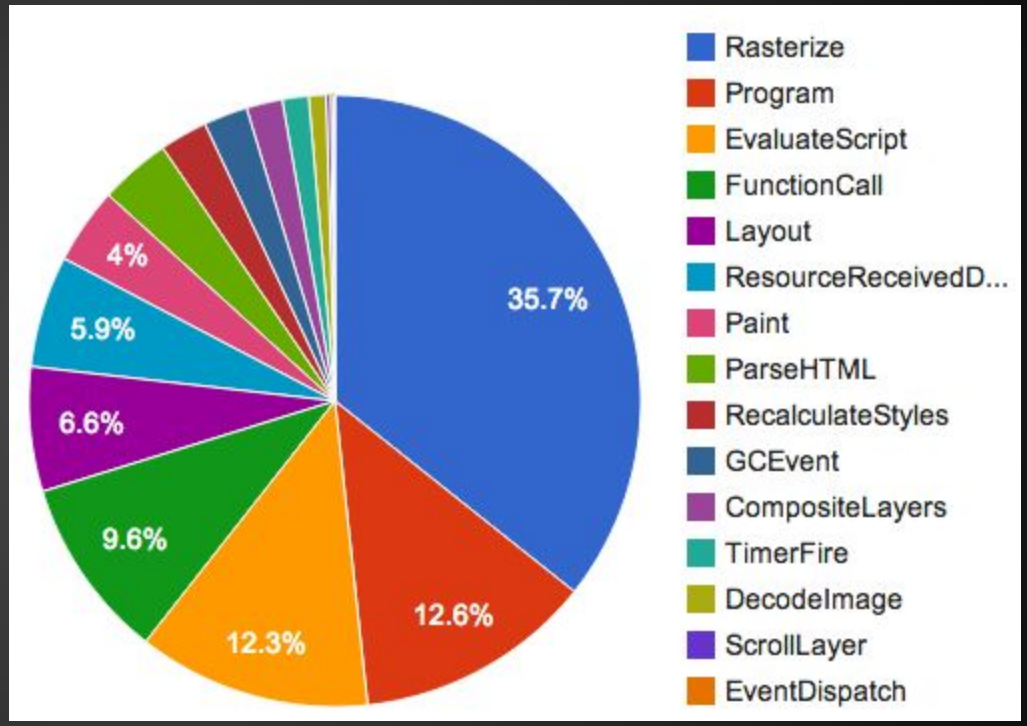
Problems with studying Linux

- Android is our highest priority platform
- Linux had inflated layout due to pathological font loading (fixed by Eric Seidel)
- Linux uses a 2 year old graphics stack

Top 25k sites - Android

Is Rasterize our highest priority? Maybe not…

- Runs on new thread
- Includes desched
- Includes animations running while blocked on network

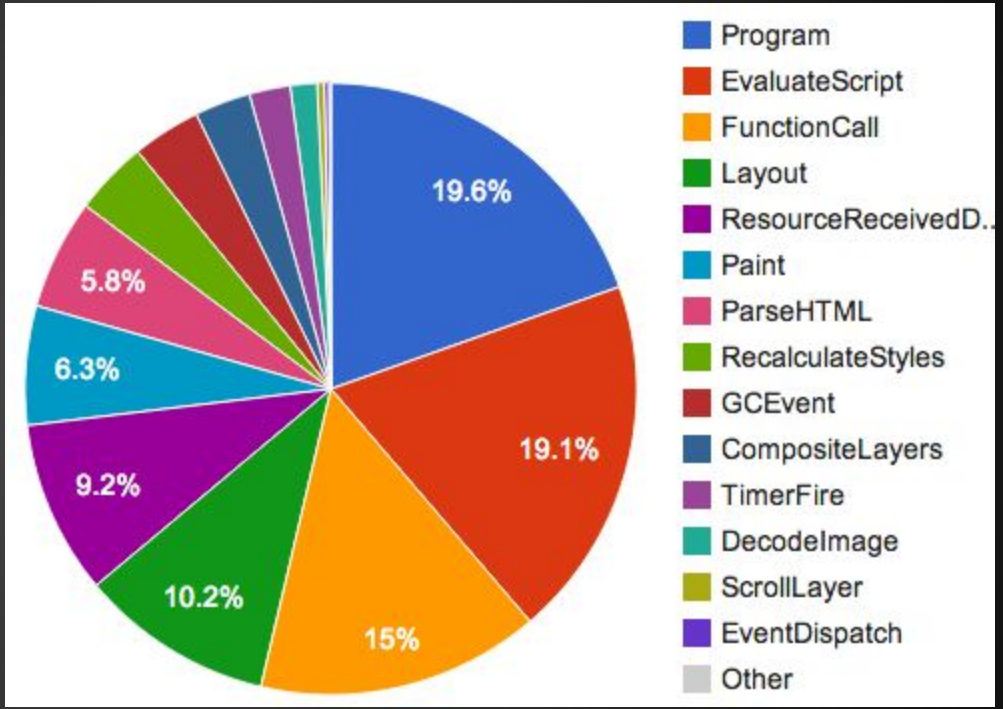


High priority to understand how threads contend

Top 25k sites - Android

#2, #3 are JS, here
devs only have 34%
of the main thread

#1 Uncategorized (20%)
#4 Layout (10%)
#5 Receive Data (9%)
#6 Paint (6%)
#7 Parse HTML (6%)



High priority to understand uncategorized time

Some priorities

- Understand thread contention better
- Categorize the uncategorized
- Drive down layout time
- Understand Receive Data contribution
- Drive down Parse HTML contribution
- Recalc Style will poke its head into the profile when some of the above are fixed.

Discussion

Moderator Page:

http://goo.gl/hfgp4L

Ideas?

Questions?

Future direction?