First Interactive and Consistently Interactive

Author: dproy@
Created: 11 April 2017
Last updated: 28 August 2017

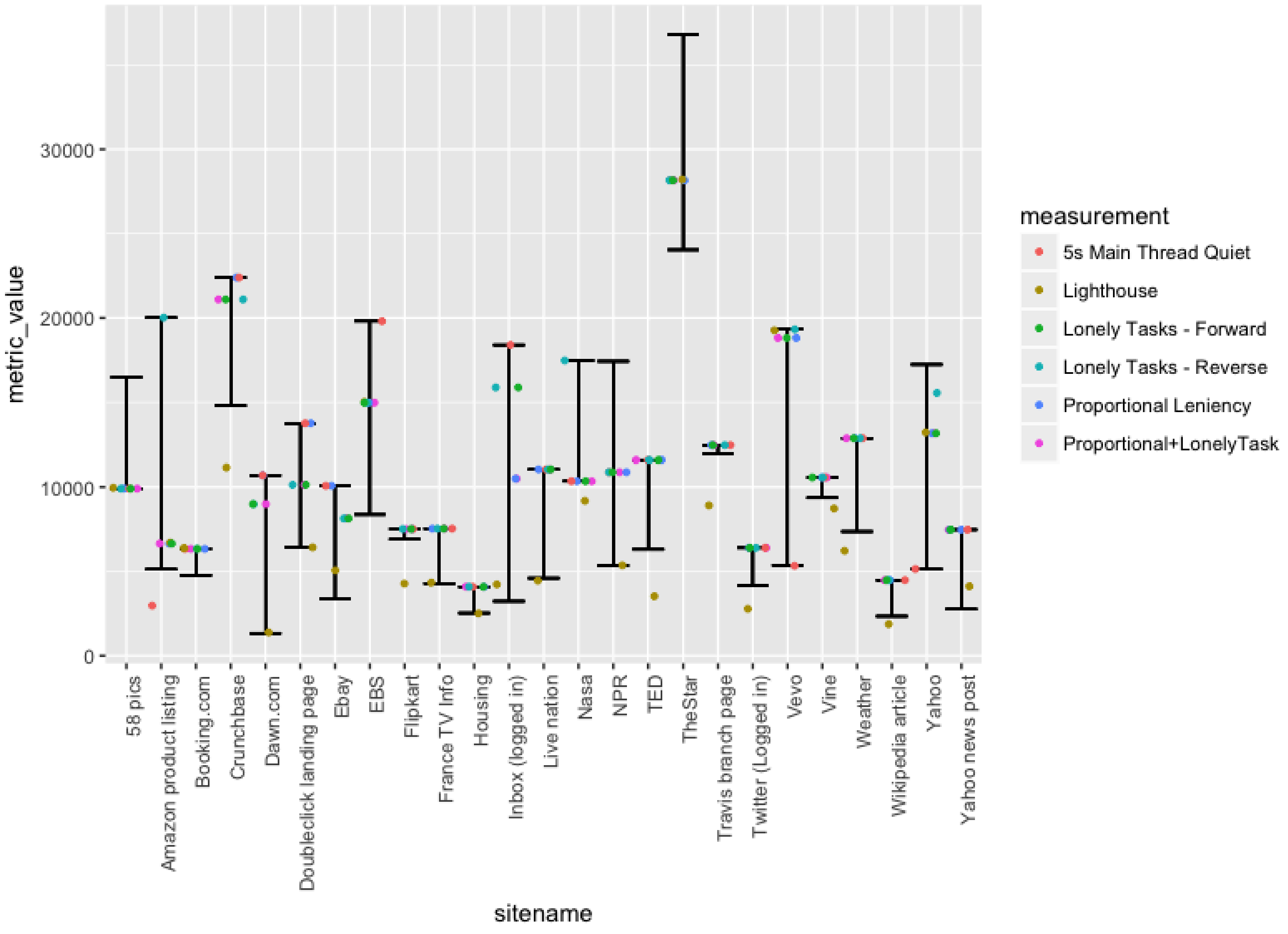
Update [2018 July 25]: We have renamed these metrics to simplify messaging to external developers. First Interactive is now First CPU Idle, and Time to Consistently Interactive is called Time to Interactive (TTI). This doc still uses the old names.

TL;DR: We split First Interactive into First Interactive and Consistently Interactive. We evaluated several candidates for these two metrics, and found a pair of definitions we think is reasonable. This doc motivates the definitions and explains the evaluation procedure in detail.

Recommendations

Our previously recommended reverse search from network-mainThread quiescence definition is staying as our recommended definition of Consistently Interactive . We reran the variability study and it displayed very little variability on 82% of the sites tested.

We experimented with several new definitions of First maybe-not-consistently Interactive. We made a labeled a range of timeline as “reasonable” for 25 sites, and tried to find a definition that always fires in the reasonable range, and then as early as possible. The experiment results can be summarized in this graph, where the black bars show the reasonable ranges, and the data points show where each definition fired:



We recommend “Proportional + LonelyTask”, which gradually shrinks the size of quiet window required, and ignores long tasks if they are isolated enough. We ran a variability study for this definition and it is stable for 87% of the sites.

Table of Contents

Splitting FirstInteractive

There are two schools of thought about how FirstInteractive should be defined:

1) FirstInteractive is the first moment when a website is minimally interactive: *enough* (but maybe not all) UI components shown on the screen are interactive, and the page responds to user input in a *reasonable* time *on average,* but it's ok if this response is not always immediate.
2) FirstInteractive is the first moment when a website is completely and delightfully interactive - not only *everything* shown on the page is interactive, but the page strictly meets the I guideline of RAIL: the page yields control back to main thread at least once every 50ms, giving the browser enough breathing room to do smooth input processing.

Using the same definition of firstInteractive to please both camps is infeasible: a metric that leans more towards latter will make the fans of the first definition think firstInteractive is too pedantic and not worth optimizing for; a metric that leans more towards the former will make fans of the second definition complain firstInteractive is too lax and not meaningful to optimize for.

So we split FirstInteractive into two metrics - for lack of better names we will call them ***First Interactive*** and ***Consistently Interactive.***

Something to clear up before we go further:

- **These definitions are nebulous**: we have a vague intuition about how firstInteractive should behave, but it is often impossible to pinpoint a timestamp in the loading timeline of a page as the "true" value of firstInteractive, especially for the first definition.
- **The definitions are necessarily heuristic driven**: there will always be websites that manage to frustrate our heuristic. Our realistic goal is a metric such that
  - it points us to a timestamp when there was a big jump in interactivity for a high enough number of sites
  - Making this number smaller smaller directly correlates with better user experience.

How do we evaluate a definition of firstInteractive?

There are two criteria:
⇒ **Correctness:** We may not know when the true firstInteractive is, but is the timestamp *reasonable?*
⇒ **Stability:** Does the metric consistently return the same value for the same site, or does it exhibit multimodal behavior?

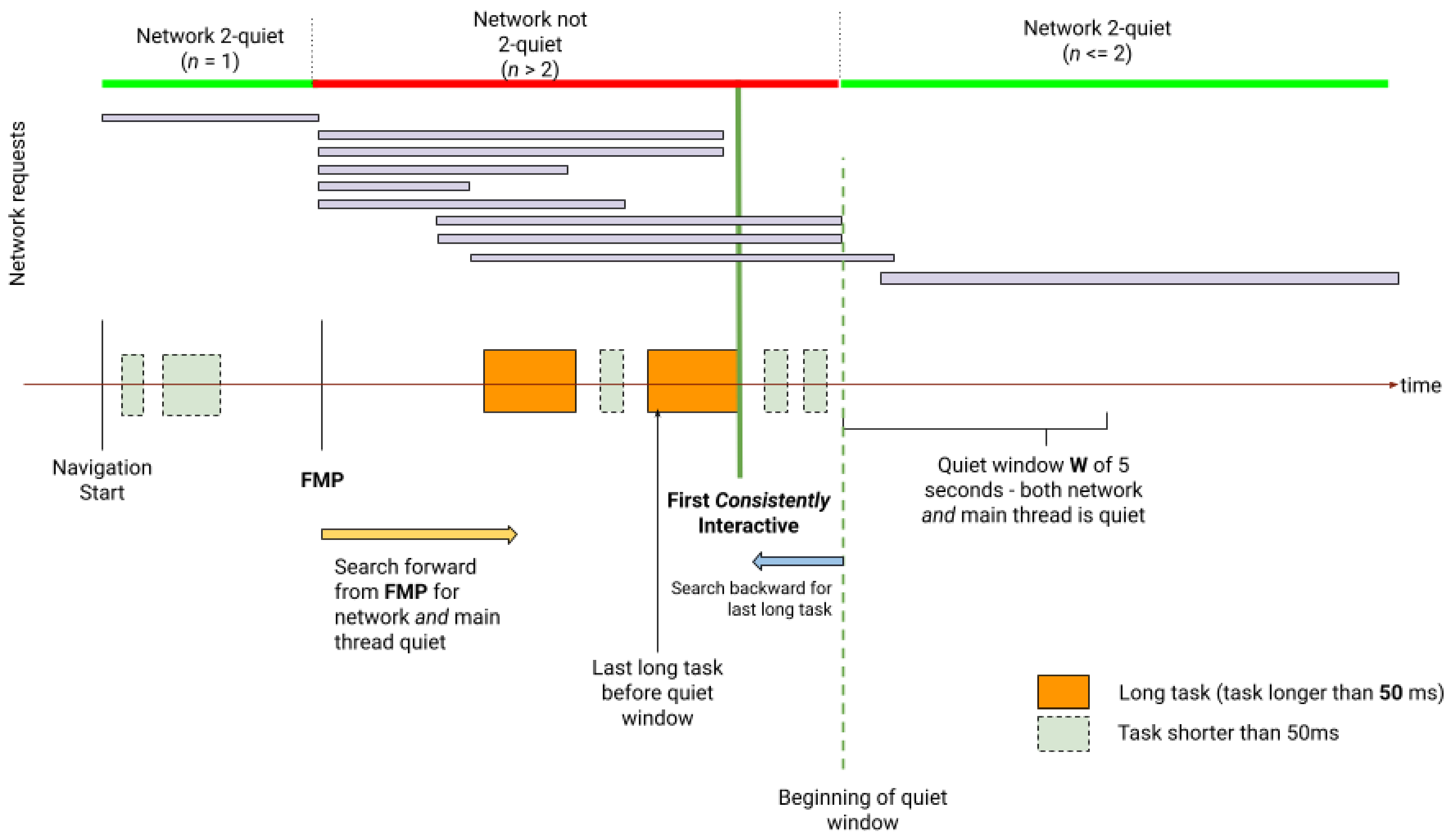
Consistently Interactive :: Correctness

We did a study where we evaluated three definitions. We found:

- Reverse Search from the end of traces and Reverse Search from Network + MainThread quiescence performed exactly the same.
- Both never fired too early, and both fired debatably too late in a number of cases. In contrast, the simple Forward Search definition performed too early a number of times.
- Since we are splitting the metric in two, the "maybe too late" cases now become perfectly acceptable.

The reverse search from network+mainThread quiescence definition does not depend on when we stop recording traces, which is not true for the absolute reverse search definition. With absolute reverse search, stopping recording at time A vs time B can yield two different firstConsistentlyInteractive values. With reverse search from network firstConsistentlyInteractive, while it is possible that we may not be able to discover firstConsistentlyInteractive for one or both of A or B, we will never report two different non-null values of firstConsistentlyInteractive.

This is an important property for stability, which is why **reverse search from network+mainThread quiescence** is the preferred definition.



The basic concept of First Consistently Interactive definition is we look for a 5 second window W where network is mostly quiet (no more than 2 network request in flight at any given time) *and* there are no tasks longer than 50ms in W. We then find the last long task before this window and call the end of that task Consistently Interactive.

See Appendix B for a precise definition.

Consistently Interactive :: Stability

*On a variability study of 100 popular sites, the recommended definition was stable **82%** of the time.*

We previously did a variability study against live sites and found that 65% of the time, the metric had acceptable variability. To recap the evaluation process: We picked around 100 sites and loaded them 25 times each, generated a graph for each of these sites, looked at each graph and made a subjective judgment about whether the metric is stable enough (roughly, we consider the metric stable for a site if there is no more than 3 outlier data points among 25), and counted the number of sites for which it was stable.

The large source of noise in that study was the live internet - we were seeing many cases where the page clearly loaded different content (e.g. different ads) for different loads, so it was not fair to expect firstConsistentlyInteractive to always fire at the same place.

We reran the variability study to use WPR recordings this time to remove that source of noise. Our Consistently Interactive is now stable **82% of the time**.

Data for the new study :

- All graphs
- Spreadsheet with all verdicts.

First Interactive :: Correctness

To evaluate different First Interactive definitions, we reused the 25 annotated traces we used to determine the correctness of Consistently Interactive, and identified a region in the timeline as the *Reasonable Region* (see column I-J on this spreadsheet).

- The end of the *reasonable region* is always the value determined by our Consistently Interactive - it is very unreasonable for our First Interactive to fire later than Consistently Interactive.
- The start of the *reasonable region* is manually annotated, and we've taken a fairly liberal stance on what we would consider interactive - usually at time ***t***, if the user can tap/click most parts of the page, and page does *something*, even if that behavior is not the fully loaded final behavior, we deem it reasonable to have a firstInteractive value of ***t***. As an aside, since we were more accepting this time, most of the classic Forward Search FirstInteractive values we labeled as "Too Early" before were now in the Reasonable zone.

For any definition of First Interactive we come up with, we want:

(1) It produces values in the *Reasonable Region* for all our 25 annotated traces.
(2) The delta between First Interactive and Consistently Interactive is as large as possible.
Any measure of this delta will work - we'll use the sum of deltas over the 25 sites and call it *total delta* below.

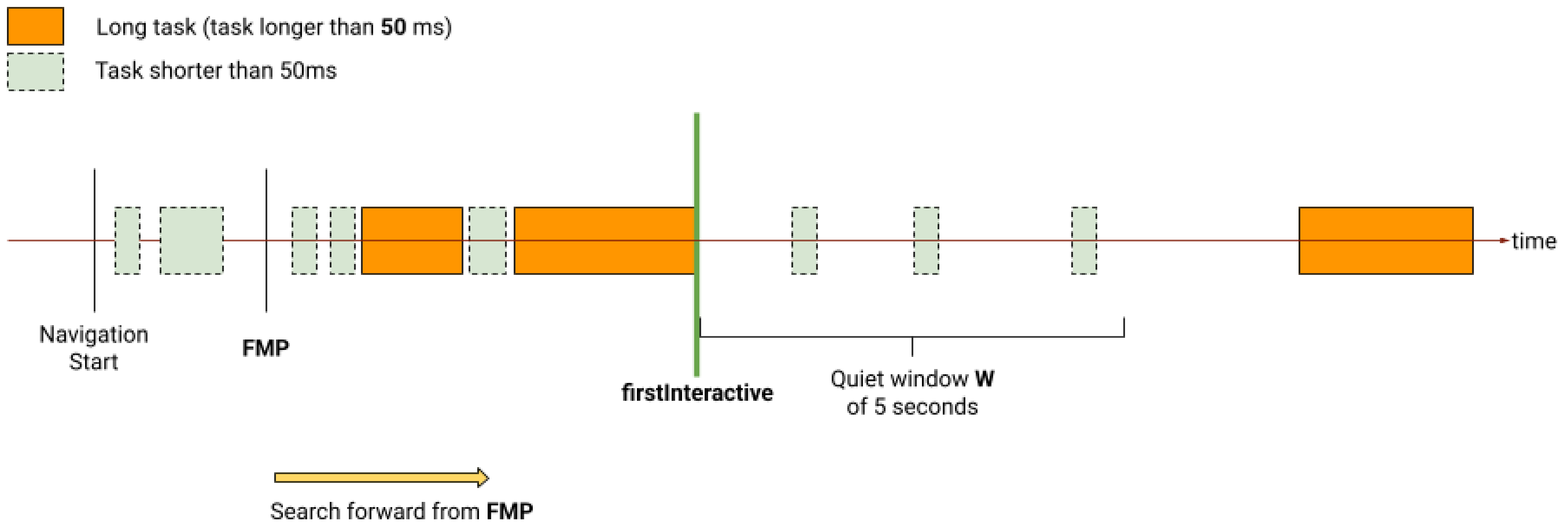
We now introduce several candidates for First Interactive, and examine their behavior on our test traces.

Note: Lower Bounding FirstInteractive at DOMContentLoadedEnd

DOMContentLoadedEnd is the point where all the DOMContentLoaded listeners finish executing. It is very rare for critical event listeners of a webpage to be installed before this point. Some of the firstInteractive definitions we experimented with fired too early for a small number of sites, because the definitions only looked at long tasks and network activity (and not at, say, how many event listeners are installed), and sometimes when there are no long tasks in the first 5-10 seconds of loading we fire FirstInteractive at FMP, when the sites are often not ready yet to handle user inputs. We found that if we take max(DOMContentLoadedEnd, firstInteractive) as the final firstInteractive value, the values returned to *reasonable region.* Waiting for DOMContentLoadedEnd to declare FirstInteractive is sensible, so all the definitions introduced below lower bound firstInteractive at DOMContentLoadedEnd.

(This is different from choosing DCL as the point where we start our forward search. For example, for the classic forward search firstInteractive, we can have a 5s quiet window after FMP where DCL fires somewhere in the middle of the window, and we will pick that DCL timestamp as firstInteractive. If there is a long task within 5s of DCL, starting the search window at DCL will yield a different firstInteractive result.)

Definition 1: Forward Search for five seconds with no long task

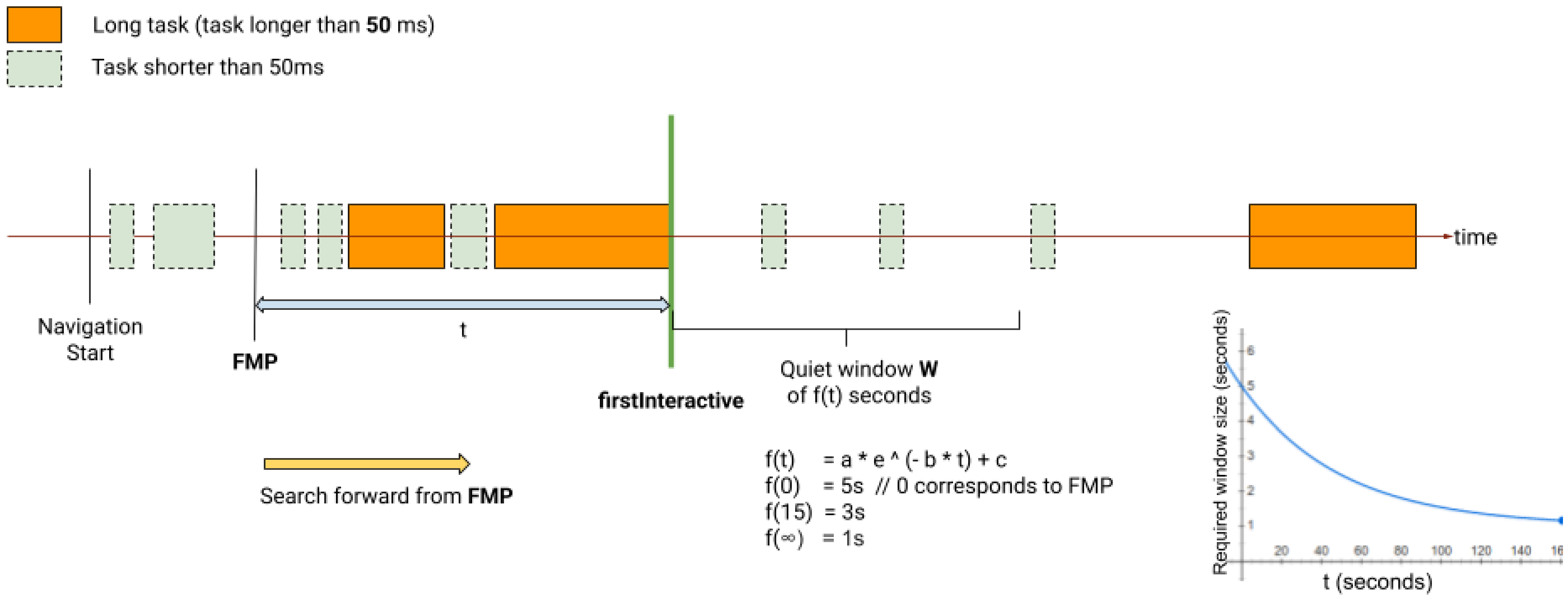


The classic forward search definition, where we look for a 5s window of no tasks longer than 50ms, is a very decent candidate for First Interactive.

Results:

- 25/25 sites reasonable, after clamping at DCL
- Total delta: **46830.87**. Delta breakdown per site. Original traces available here.

Definition 2: Proportional leniency


Long task (task longer than 50 ms)
Task shorter than 50ms
time
Navigation Start
FMP
t
firstInteractive
Quiet window W of f(t) seconds
Search forward from FMP
f(t) = a * e ^ (- b * t) + c
f(0) = 5s // 0 corresponds to FMP
f(15) = 3s
f(∞) = 1s
Required window size (seconds)
6
5
4
3
2
1
20
40
60
80
100
120
140
16
t (seconds)

The constant window size of **5** seconds in the previous definition was arbitrary. Near FMP, this 5 second window makes sense - lots of scripts are still being downloaded and we need a somewhat long window to say with confidence that all the critical js to make the page interactive has executed already. Our goal with First Interactive is to detect the point where the initial flurry of loading activity to get a page minimally interactive is done, and 15 seconds after FMP, having a 3 second quiet window is a good enough signal for that.

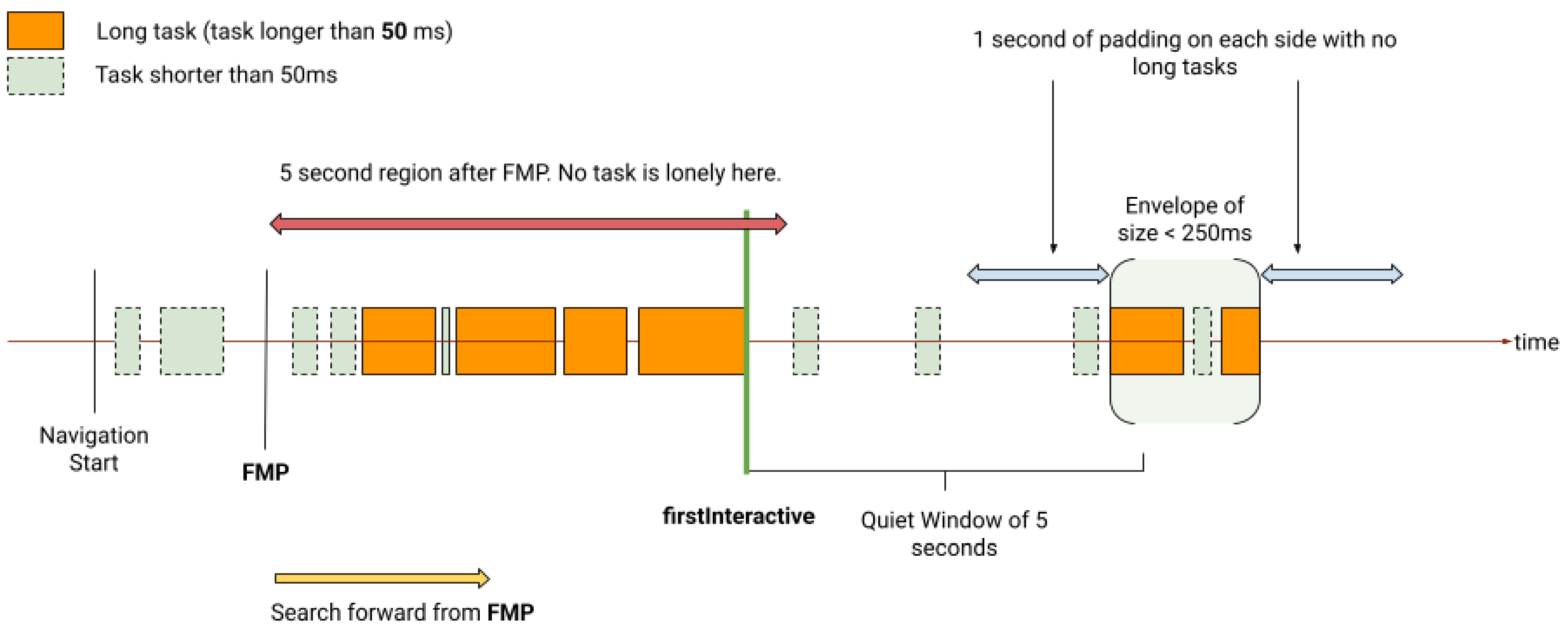
To obtain Definition 2, we therefore modify Definition 1 so that the required size of window gradually shrinks the further away we are from FMP, but never drops below 1s. For the curve we chose, the required window size falls at a rate such that it's 3s when we are 15s away from FMP.

In more precise terms, the required window size is now f(t) instead of being a constant, where t is the time between the start of the window and FMP. We model f(t) as a negative exponential function - f(0) is 5s, f(15) is 3s, and as t → ∞, f(t) → 1. Working out the math, we approximately get f(t) = 4 * e^(-0.045 * t) + 1.

Results for Definition 2:

- 25/25 sites reasonable, after clamping at DCL
- Total delta: **59568.44**. Delta breakdown per site. Original traces available here.

Definition 3: Forgiving lonely tasks


Long task (task longer than 50 ms)
Task shorter than 50ms
1 second of padding on each side with no long tasks
5 second region after FMP. No task is lonely here.
Envelope of size < 250ms
time
Navigation Start
FMP
firstInteractive
Quiet Window of 5 seconds
Search forward from FMP

When the page is doing the most critical loading related tasks, the long tasks are usually densely packed. The isolated tasks are usually some third party ads or analytics script (and sometimes V8 GC tasks), and these should not block First Interactive - calling out the effect of these is the job of Consistently Interactive.

We will call a set of long tasks ***lonely*** if they can be enveloped in a window ***L*** of size less than 250ms, such that there is no long task overlapping with [*L.start* - 1s, *L.start*] and [*L.end*, *L.end* + 1s] regions. We add the additional condition that within 5 seconds of FMP, no task is lonely: all tasks that close to the FMP have a high probability of being critical.

To obtain Definition 3, we now modify Definition 1 such that lonely tasks are no longer considered long tasks.

Another version of this definition (let's call it Definition 3.1) is we search backward like our Consistently Interactive, and ignore lonely tasks. This has a lower total delta, but is still pretty good.

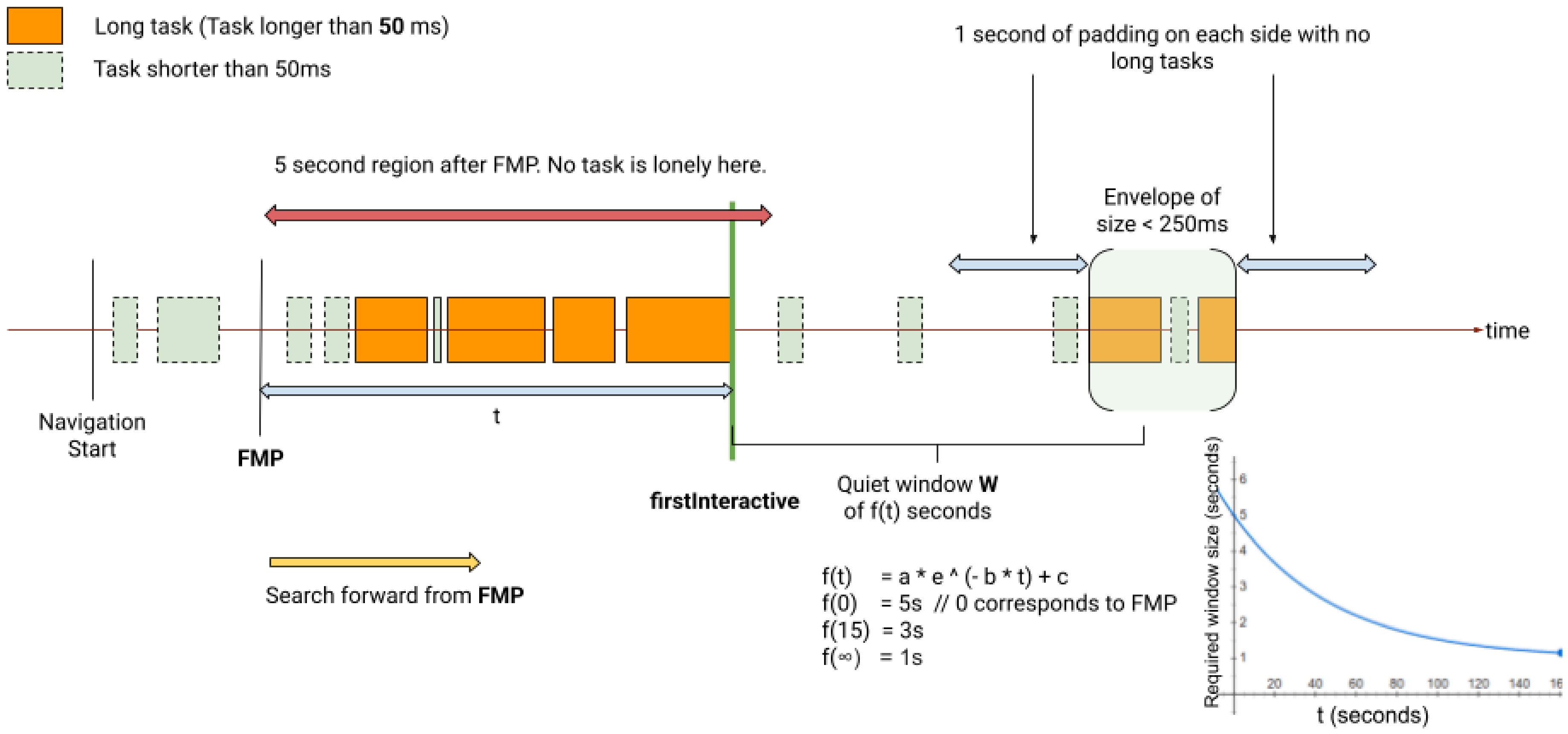
Results for Definition 3:

- 25/25 sites reasonable, after clamping at DCL
- Total delta: **62753.27**. Delta breakdown per site. Original traces available here.

Results for Definition 3.1:

- 25/25 sites reasonable, after clamping at DCL
- Total delta: **39353.46**. Delta breakdown per site. Original traces available here.

Definition 4: Combining proportional leniency and lonely tasks


Long task (Task longer than 50 ms)
Task shorter than 50ms
1 second of padding on each side with no long tasks
5 second region after FMP. No task is lonely here.
Envelope of size < 250ms
time
Navigation Start
FMP
t
firstInteractive
Quiet window W of f(t) seconds
Search forward from FMP
f(t) = a * e ^ (- b * t) + c
f(0) = 5s // 0 corresponds to FMP
f(15) = 3s
f(∞) = 1s
Required window size (seconds)
6
5
4
3
2
1
20
40
60
80
100
120
140
16
t (seconds)

Of course, these two ideas are cleanly composable; why don't we have both? This is exactly was Definition 4 is - we gradually shrink the required quiet window size, and also forgive isolated long tasks.

Results:

- 25/25 sites reasonable, after clamping at DCL
- Total delta: **68137.32**. Delta breakdown per site. Original traces available here.

Recommended First Interactive Definition

Definition 4 is the best we have right now. It combines both of our intuition about what First Interactive looks like in a trace, and has the highest total delta.

See this appendix section for self contained definition if you want to implement FirstInteractive. Also, there is an equivalent FirstInteractive definition in terms of heavy and light task clusters instead of task envelopes and lonely tasks that leads to a cleaner implementation. See First Interactive - Task Cluster Definition.See this appendix section for self contained definition if you want to implement FirstInteractive. Also, there is an equivalent FirstInteractive definition in terms of heavy and light task clusters instead of task envelopes and lonely tasks that leads to a cleaner implementation. See First Interactive - Task Cluster Definition.

Caveats

Notes on the possible risk of overfitting:

- The proportional leniency curve was the first curve we tried with parameters based on our intuitions, so that curve is likely not overfitted.
  We did look at some other curves afterwards, but they provided no improvement and we stuck with our first definition.
- The parameters for identifying lonely tasks (1s padding, 250ms envelop) *was* determined by doing a brute force search over a large space of possible parameters. The parameters still make sense intuitively, but the risk of overfitting here is much higher.

As we try out our definitions in the real world we may need to revise our parameters if it becomes apparent the parameters are not optimal.

Note: Using FCP instead of FMP for start of window

All the definitions presented above uses FMP as the start of window. We experimented with using FCP as the start of window. Since FCP is easier to standardize, dropping the dependency on FMP can yield a easier path to standardization for TTI. Unfortunately, using FCP at the start of window tend to make our definition fire too early. For example, if we substitute FCP for FMP in definition 4, we can 3/25 sites firing too early.

- Metric values for each site, with too early cases marked.
- Annotated traces for too early cases:
  - Crunchbase
  - TED
  - Vine

First Interactive :: Stability

We did a similar variability study as the Consistently Interactive - about 100 sites, 25 runs each, plotted graphs, and declared metric stable for a site if <= 3 outliers. Definition 4 was stable **87%** of the time.

Data:

- All graphs
- Spreadsheet with all the verdicts

For comparison we did the variability also on Definition 2 and Definition 3.1, and they were both stable 85% of the time. Here is a list of slightly messier graphs that plots Definition 2, 3.1, and 4 for each site.

Appendix A: Results from evaluating other definitions

In course of finding First Interactive we tried an array of other ideas. For completeness, we are including the list of these ideas here:

- **EQT based FirstInteractive:**
  - **Lighthouse FirstInteractive:** The definition previously implemented in lighthouse looked for a 500ms window with 90th percentile EQT less than 50ms. In our tests, it fired too early 12 times: Link to data. (If you're interested, link to raw traces and link to annotations for determining reasonable ranges.)
    Changing the parameters is unlikely to fix this metric. EQT based approaches have theoretically pleasing formulations, but in practice sites go through periods of low EQT too often and it's extremely difficult to identify the real bulk of loading activity, especially if don't want a series of back to back 49 ms tasks to not block FirstInteractive.
  - **Reverse search for periods of High EQT:** Similar to the lighthouse approach, but doing the search backwards from a certain point of interest (say network quietness.) We only tried using mean EQT instead of EQT percentiles, but tried many different parameters for window size and EQT threshold. It showed very little promise.
- **Event listener based FirstInteractive:**
  - We briefly played around with a metric that looks for stability in the raw count of event listeners on the page. The data was so unpromising that we did not proceed to combining event listener data with other heuristics.

Appendix B: Precise definitions of presented metrics

FMP = First Meaningful Paint

Consistently Interactive:

Find a the first 5 second window W after FMP such that

- W overlaps no tasks longer than 50ms
- For all timestamp t in W, number of resource requests in flight at t is no more than 2.

Now find the last long task L before W.

- Consistently Interactive Candidate is the end of L
- In the case there is no long task before L, Consistently Interactive Candidate = FMP.

Now, Consistently Interactive = max(Consistently Interactive Candidate, DOMContentLoadedEventEnd)

New Edit (28 April 2017): Added lower bounding at DCL. We should also lower bound this metric so that we always have First Interactive ≤ Consistently Interactive.

See this doc for motivations and pretty diagrams.

First Interactive:

Note (August 28, 2017): There is an equivalent FirstInteractive definition in terms of heavy and light task clusters instead of task envelopes and lonely tasks that leads to a cleaner implementation. See First Interactive - Task Cluster Definition. We have the original definitions in terms of Lonely Tasks below:

Let $b = \frac{1}{15000}\log(\frac{1}{4000}(3000 - 1000))$,
and $f(t) = 4000\,e^{bt} + 1000$

Note that

- f(0) = 5000
- f(15000) = 3000
- f(x) = 1000 as x → infinity.

[We're measuring time in ms here which is why there are so many trailing 0s.]

Define a task T as *lonely* if there exists a window E of size at most 250 ms such that

- T is completely contained in E.
- E.start is at least 5 seconds away from FMP
- The windows [E.start - 1 second, E.start] and [E.end, E.end + 1 second] overlaps no tasks longer than 50 ms.

- (just to make things clear) E is allowed contain however many long tasks it can fit.

(You can refer to E as the lonely window)

Now to get First Interactive Candidate, find the first window W after FMP such that

- If W overlaps a task T, either duration of T is less than 50ms, or T is lonely.
- W.duration <= f(W.start - FMP)

First Interactive is max(First Interactive Candidate, DOMContentLoadedEventEnd).

Change Log

28 August 2017: Add link to FirstInteractive definition in terms of task clusters.
25 July 2017: Correction: ConsistentlyInteractive meets ***R*** guideline of RAIL ⇒ Strictly meets ***I*** guideline of RAIL.
28 April 2017: Added lower bounding at DCL.