Expected Queueing Time

Expected Queueing Time, or EQT, measures how responsive a page would be to input, if it was receiving input. In some cases, measuring actual input latency makes more sense, but for cases where synthesizing realistic user input is difficult, or noise caused by the unpredictability of user input is a problem, EQT is an effective solution.

The EQT over a given time window is defined as the expected amount of time input arriving at a random time during the window would be queued before it started being processed.

Off thread input handling

Expected Queueing Time looks at how available the thread executing javascript is. For input handling which is not blocked on this thread, EQT is not a good metric. For example, Chrome's scrolling is generally performed on the Renderer Compositor thread, which is not blocked on the Renderer Main thread.

Assumptions

For EQT, we assume that input is scheduled at the highest possible priority. This means that if there was a pending non-input task, we assume that the input task will always pre-empt the pending task.

Definition

Where$E(W)$is the expected queueing time for a priority task within the window W.

$E(W) \ = \sum_{X \in W} P(X) \times E(X)$, where X includes both tasks and gaps within W.

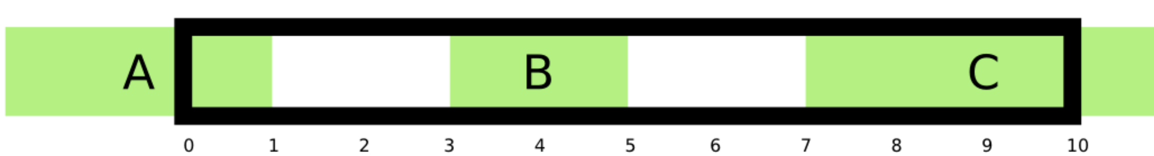
$P(X) \ = \frac{min(X.end, W.end) - max(X.start, W.start)}{W.end - W.start}$ is the probability of a random new task falling within X

.

$E(X) \ = 0$ if X is a gap, or

$E(X) \ = \frac{(X.end - max(X.start, W.start)) + (X.end - min(X.end, W.end))}{2}$ if $X$is a task.

This last formula takes the average of the remaining queueing time at the beginning and the end of the part of the current task within W.



In this example, the probability of a task with random starting time starting within **A** is 1/10, and the expected queueing time for a task which falls within **A** and is within the window is 1/2: the average of the queueing time if the task started at time 0 or at time 1.

The probability of a task with random starting time starting within **B** is 2/10, and the expected queueing time for a task which falls within **B** and is within the window is 1: the average of the queueing time if the task started at time 3 or at time 5.

The probability of a task with random starting time starting within **C** is 3/10, and the expected queueing time for a task which falls within **B** and is within the window is 5/2: the average of the queueing time if the task started at time 7 or at time 10. This is the trickiest case. The queueing time if the task had started at time 7 would have been 4, and then queueing time if the task had started at time 10 would have been 1, leading to an average queueing time of 5/2.
Then the total expected queueing time is $\frac{1}{10} \times \frac{1}{2} + \frac{2}{10} \times 1 + \frac{3}{10} \times \frac{5}{2} = 1$.