LOAD TOOL TRAFFIC GENERATION FOR LARGE USER POPULATIONS (CMG'12 Presentation Follow-up)

James F Brady
Capacity Planner for the State of Nevada
jfbrady@admin.nv.gov

This is a follow-up to my CMG'12 presentation titled "When Load Testing Large User Population Web Applications the Devil is in the (Virtual) User Details", [BRAD2012]. During my talk I mentioned that most load tools do not offer a Negative-Exponential distribution timer option, even though it is the logical one to draw think times from when attempting to mimic the request pattern of a large user population. I followed that comment with a definition of the simple formula used to generate this type of traffic and some attendees expressed an interest in the formula's origin and the principles surrounding its derivation. The following is intended to satisfy that request by, providing the reason to produce the traffic pattern,stating the timer formula, describing the methodology used to create it, illustrating its implementation with an example, and detailing its derivation mathematically.

Motivation

Motivation for generating Negative-Exponentially distributed times between requests results from the fact that transactions produced by a large population of users, who have no coercion between them when pressing the Enter Key, are independent from each other. This request timing pattern is referred to in the literature as a Poisson process [WIKI01], or random arrivals, and is characterized by times between arrivals being Negative-Exponentially distributed and the number of arrivals in constant length intervals possessing Poisson distribution attributes. If you are unfamiliar with these traffic flow concepts, an intuitive illustration of them using a one meter ruler and two sets of numbered chips is contained in [BRAD2009].

Formula

The formula which produces the desired Negative-Exponentially distributed times between requests is represented by either Eq. 1 or Eq. 2. Using Eq. 2, the time until the next request, t_0, is obtained by multiplying minus the mean time between requests, $-\mu$, by the natural logarithm of a uniformly distributed random number between zero and one, $ln(r_0)$. Eq. 2 is usually used over Eq.1 to obtain t_0 because $t_0 = -\mu\, ln(1 - r_0)$ and $t_0 = -\mu\, ln(r_0)$ yield symmetrical results and Eq. 2 requires one less arithmetic operation than does Eq. 1.

$$t_0 = -\mu\ ln(1 - r_0) \qquad \text{(Eq. 1)}$$

$$t_0 = -\mu\ ln(r_0) \qquad \text{(Eq. 2)}$$

Where:

$t_0 = time\ to\ next\ request \quad ln = natural\ log\ (ln\, e = 1)$

$\mu = mean\ time\ between\ requests \qquad r_0 = random\ number\ 0 \le r_0 \le 1$

Methodology

Eq. 1 results from applying a technique called the Inverse Transformation Method [WIKI02] to the Negative-Exponential distribution. This method, used extensively in load tools and simulation packages, produces samples from the chosen probability distribution by exploiting the fact that the distribution's Cumulative Distribution Function [WIKI03] possesses the same range of values as a (0,1) random number generator.

Eq. 3 illustrates the relationships between the uniformly distributed (0,1) random number, r, the Cumulative

Distribution Function, $F(t)$, and the integral of the Probability Density Function [WIKI04], $f(t)$, within the context of the Inverse Transformation Method.

$$r = F(t) = \int_0^t f(t)\,dt \qquad \text{(Eq. 3)}$$

Where:

$r = random\ number\ 0 \le r \le 1$

$F(t) = Neg\ Exp\ Cum\ Distribution\ Func(CDF)\ \ f(t) = Neg\ Exp\ Prob\ Density\ Func(PDF)$

$t = time\ to\ next\ request$

The expression that yields the desired values of t is derived by performing the integral in, $r = \int_0^t f(t)\,dt$, evaluating it at the limits shown and solving the equation for t.

Eq. 4 is the Negative-Exponential distribution implementation of Eq. 3.

$$r_0 = F(t_0) = \int_0^{t_0} \frac{1}{\mu} e^{-\frac{t}{\mu}}\,dt \qquad \text{(Eq. 4)}$$

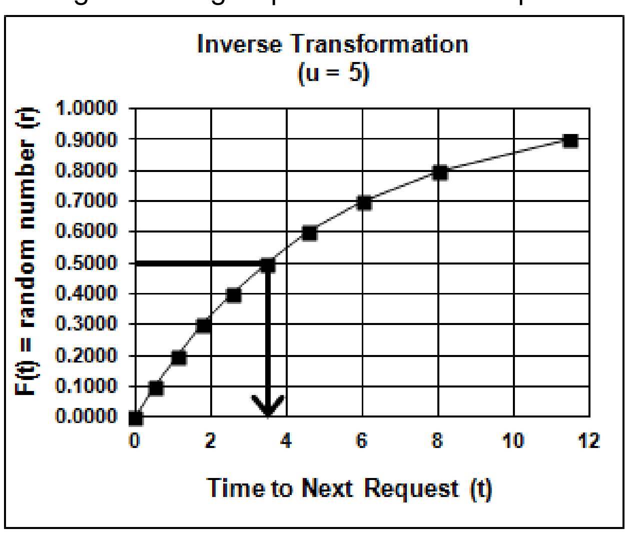
When the integral is performed, evaluated at its limits, and the expression solved for t_0, the result is Eq. 1. The Derivation section of this document contains the step by step mathematical details of the procedure.

Illustration

The graph and associated table in Figure 1 provide an intuitive illustration of the Inverse Transformation Method for the Negative-Exponential distribution using $\mu = 5$ as the mean time between requests.

As shown, a uniformly distributed (0, 1) random number is drawn and its location found on the $F(t)$ axis. Then, a horizontal line is drawn from the $F(t)$ axis until it intersects the Eq. 1 curve. The time to delay until making the next request is the t axis value at that intersection point.

Figure 1: Neg-Exp Time to Next Request



F(t)	t (u=5)
r	t = -u*ln(1-r)
0.0000	0.0
0.1000	0.5
0.2000	1.1
0.3000	1.8
0.4000	2.6
0.5000	3.5
0.6000	4.6
0.7000	6.0
0.8000	8.0
0.9000	11.5

The shaded example shows that if r = .5000 is drawn the time to wait until making the next request is t = 3.5.

Derivation

What follows is a detailed derivation of Eq. 1 and Eq. 2 performed by applying the Inverse Transformation Method to the Negative-Exponential distribution.

The first step in this process is to integrate the Negative-Exponential Probability Density Function (PDF), Eq. 5 through Eq. 9, and evaluate the resulting Cumulative Distribution Function (CDF) over the range $(0, t_0)$, Eq. 10 through Eq. 13.

The second step, shown in Eq. 14 through Eq. 20, is to set the results of this integral equal to a (0,1) random number, r_0, and solve for t_0, the time to wait before making the next request.

Step 1: Integrate the PDF over the range $(0, t_0)$:

$$f(t) = \frac{1}{\mu} e^{-\frac{t}{\mu}} \qquad \text{(Eq. 5)}$$

$$F(t_0) = \int_0^{t_0} f(t)\, dt \qquad \text{(Eq. 6)}$$

$$F(t_0) = \int_0^{t_0} \frac{1}{\mu} e^{-\frac{t}{\mu}}\, dt \qquad \text{(Eq. 7)}$$

$$F(t_0) = \frac{1}{\mu} \int_0^{t_0} e^{-\frac{t}{\mu}}\, dt \qquad \text{(Eq. 8)}$$

From Calculus:

$$\int e^{at}\, dt = \frac{e^{at}}{a} \qquad \text{(Eq. 9)}$$

Evaluate the CDF at its limits $(0, t_0)$:

$$F(t_0) = \frac{1}{\mu} \left[-\mu e^{-\frac{t}{\mu}} \right]_0^{t_0} \qquad \text{(Eq. 10)}$$

$$F(t_0) = \frac{1}{\mu}\left[\left(-\mu e^{-\frac{t_0}{\mu}}\right) - \left(-\mu e^{0}\right)\right] \qquad \text{(Eq. 11)}$$

$$F(t_0) = \frac{1}{\mu}\left[\mu - \mu\, e^{-\frac{t_0}{\mu}}\right] \qquad \text{(Eq. 12)}$$

$$F(t_0) = 1 - e^{-\frac{t_0}{\mu}} \qquad \text{(Eq. 13)}$$

Step 2: set $r_0 = F(t_0)$ and solve for t_0:

$$r_0 = F(t_0) = 1 - e^{-\frac{t_0}{\mu}} \qquad \text{(Eq. 14)}$$

$$r_0 = 1 - e^{-\frac{t_0}{\mu}} \qquad \text{(Eq. 15)}$$

$$e^{-\frac{t_0}{\mu}} = 1 - r_0 \qquad \text{(Eq. 16)}$$

$$ln\left(e^{-\frac{t_0}{\mu}}\right) = ln(1 - r_0) \qquad \text{(Eq.17)}$$

$$-\frac{t_0}{\mu} = ln(1 - r_0) \qquad \text{(Eq. 18)}$$

$$t_0 = -\mu\ ln(1 - r_0) \qquad \text{(Eq. 19)}$$

Given the symmetry of $(1 - r_0)\ and\ r_0$:

$$t_0 = -\mu\ ln(r_0) \qquad \text{(Eq.20)}$$

Summary

When load testing large user population applications Negative-Exponentially distributed think times are the logical choice but that “think time” option is often not on the load tool’s list of available timers. Eq. 20 clearly illustrates its absence isn’t due to mathematical complexity. The formula is simple and, as the previous section demonstrates, its derivation is straight forward.

Those who attended my CMG’12 presentation heard that other think time probability distributions can effectively mimic a large user population request pattern if a sufficient number of load tool threads are implemented. The traffic quality evaluation effort associated with this approach, I described in my talk, can be mitigated by integrating Eq. 20 into the load tool’s framework because this timer algorithm produces think times that are independent from thread count.

References

[BRAD2009] J. F. Brady, "The Rosetta Stone of Traffic Concepts and Its Load Testing Implications," CMG MeasureIT, (September 2009).
http://www.cmg.org/measureit/issues/mit63/m_63_4.html

[BRAD2012] J. F. Brady, "When Load Testing Large User Population Web Applications The Devil Is In the (Virtual) User Details", CMG'12 International Conference, (December 2012).

[WIKI2001] Wikipedia, "Poisson process", (2010)
http://en.wikipedia.org/wiki/Poisson_process.

[WIKI2002] Wikipedia, "Inverse transform sampling", (2013) http://en.wikipedia.org/wiki/Inverse_transform_sampling .

[WIKI2003] Wikipedia, "Cumulative distribution function", (2013)
http://en.wikipedia.org/wiki/Cumulative_distribution_function

[WIKI2004] Wikipedia, "Probability Density function", (2012)
http://en.wikipedia.org/wiki/Probability_density_function

Copyrights and Trademarks