Random Number Generators and Heavy-Tail Distributions

Dror G. Feitelson
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

The mean and standard deviation of heavy-tailed distributions are often infinite, as a result of non-negligible probabilities of sampling very high values. Generating random variates from such distributions may therefore depend on the range of output values of the underlying random number generator. In particular, it may be appropriate to control this range based on the number of samples that are needed.

Keywords: heavy-tail distribution, random variate generation.

Heavy-tailed distributions have been known to model various phenomena such as the distribution of word usage for over fifty years [8]. Their prominance has increased significantly in recent years, with their application to the modeling of WAN traffic [6, 7] and workloads on file servers and web servers [4, 2, 1].

Using such models to drive the simulations used in performance evaluation requires random variates from heavy-tailed distributions to be generated. The simplest (albeit rather extreme) example is the Pareto distribution with shape parameter $a = 1$. This distribution has probability mass function $f(x) = 1/x^2$, and an infinite mean, as $\int x \cdot 1/x^2 \cdot dx = \log x$. Variates from this distribution are generated by selecting u from a uniform distribution on the unit interval, and returning $1/u$ [5]. In particular, samples from the power-law tail of the distribution are generated by those uniform variates that lie very close to zero. The highest value that can be obtained depends on the smallest value that can be generated by the random number generator used for the uniform distribution.

A simple test showing the effect of the random number generator is to generate such variates, and compute their running average. As the mean of this distribution is infinite, we expect the running average to grow with the number of samples; in fact, it should be the natural logarithm of the number of samples. Figure 1 shows actual results for three random number generators that are widely available on Unix systems: rand, random, and drand48.

The left graph, covering the first million samples, indicates that the mean of variates created by the rand generator converges rather quickly, rather than continuing to grow. This is the result of the limited range of only 32768 values generated by this generator.

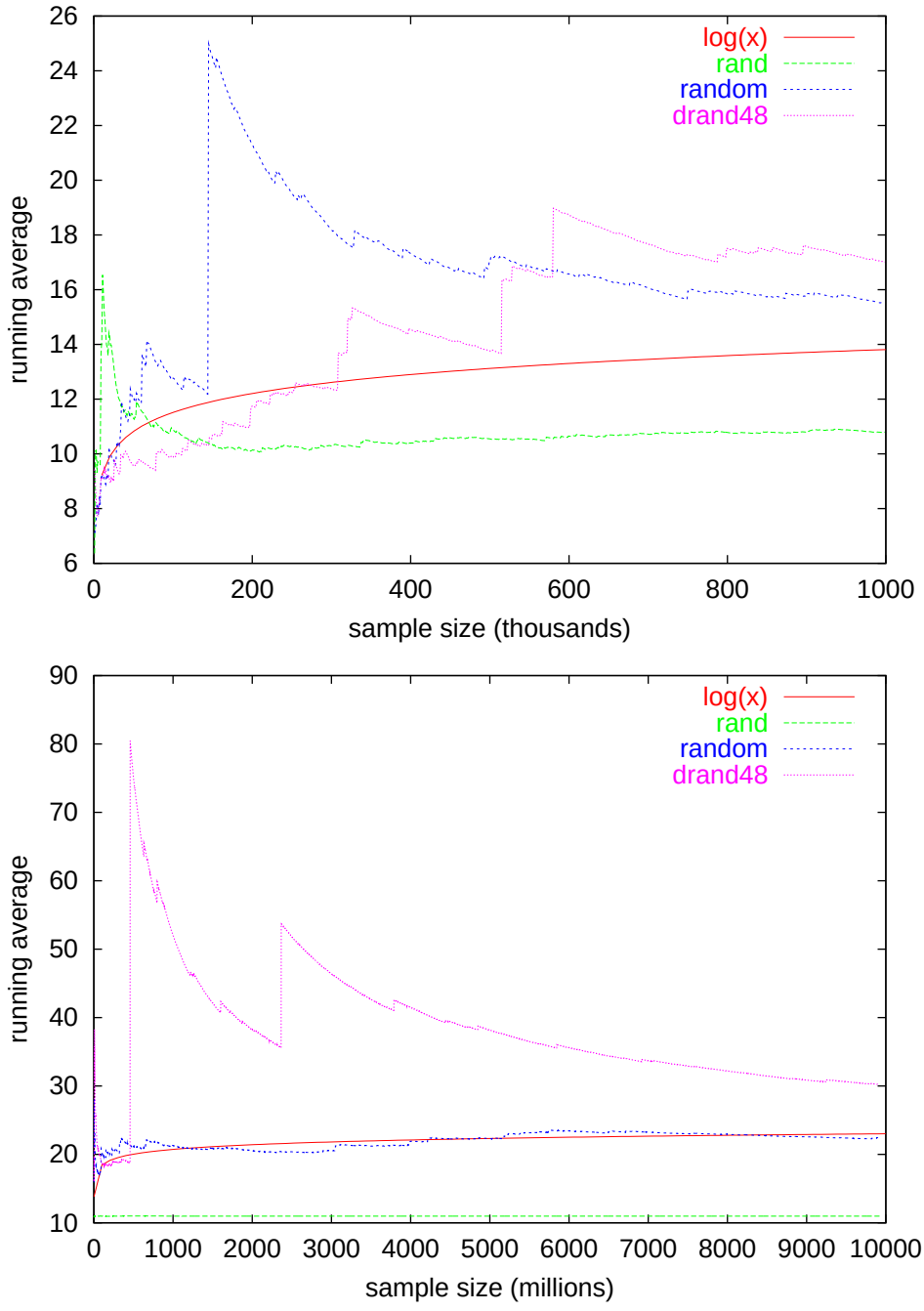

Figure 1: *Running average of samples from a harmonic distribution, as generated by different random number generators. All the results use the default seed.*

The right graph shows that random follows the logarithmic curve rather closely in the range of billions of samples. This is actually a result of the fact that it too is beginning to converge as we approach a number of samples that is commensurate with its range of 2 billion values. On the other hand, the drand48 generator continues to generate extremely large variates due to its use of 48-bit numbers. It is expected to converge to the logarithmic curve near 10^{15} samples, and not to grow any more beyond that point.

The implication for simulations is that, depending on the seed being used, a relatively short simulation might encounter a single value that belongs way out in the tail of the distribution, and has a very low probability of showing up in such a short simulation. Such an extreme value may dominate the results of the entire simulation, rendering the results meaningless.

One should realize that due to such effects, when working with heavy-tailed distirbutions, the simulation actually never converges to a steady state. Instead, it remains in a transient state, as extreme values continue to pop up [3]. A possible solution is therefore to prevent them from popping up too soon. This can be done by setting the range of values, or rather, the minimal value, as a function of the simulation length. Short simlations will set a higher threshold, thus disabling a larger part of the tail of the distribution. Long simulations will delve deeper into the tail.

An example is shown in Figure 2. The methodology used here is to delete those uniform variates whose magnitude is less than one over twice the simulation length. For a simulation using a million variates from the drand48 generator with the default seed, none are deleted[1]. When using 10 million, one variate, about 1.5 million into the sequence, is deleted (top graph). This leads to a significant change in the running average. When using 100 million, this variate is no longer considered extreme enough to delete. However another single variate, about 77 million into the sequence, is deleted (not shown). With a billion, *this* one is OK, but two others are deleted (bottom).

Setting an explicit limit on the tail of the distribution is a reasonable approach, because in many cases it does indeed reflect reality. Consider situations in which Pareto distributions have been used. One is to model the (tail of) the distribution of file sizes, or WWW request sizes. Using drand48 to generate large files may create one that is larger than the total disk capacity in the world, a situation that most modellers would probably regard as unrealistic. Likewise, a Pareto distribution of wealth may in principle create someone whose wealth exceeds the GNP of the United States. The data used to create the models does not include such extreme cases, but with certain random number generators and seeds the models may accidently create them.

There seems to be no "true" solution to the problem raised in this note. When using models with power-law tails, the modeller must make a conscientious decision about how to deal with it. We have suggested a simple approach in which the tail of the distribution is truncated at a place that has a low probability of being sampled, given the number of samples needed. While other approaches are also possible, it seems clear that it is unreasonable to leave the problem up to the random number generator.

[1]This is from a BSDI machine. On a machine running Linux 2.2, the *first* value returned by drand48 has 13 leading zeros after the decimal point, which would get deleted in most practical situations.

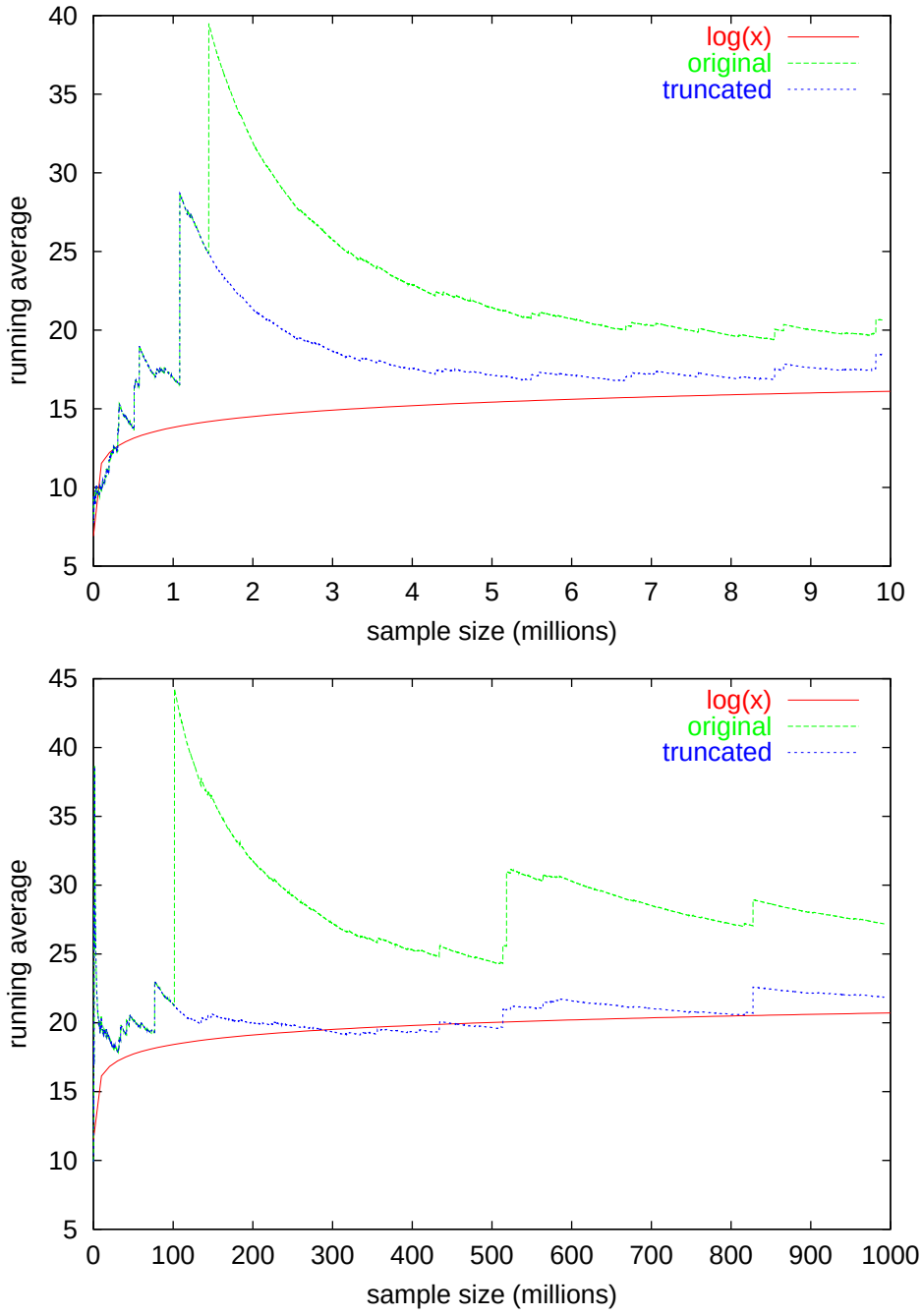

Figure 2: *Effect of truncating a single value out of 10000000 from the drand48 output (top), and two out of 1000000000 (bottom).*

Acknowledgement

This research was supported by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities. The observations in this note were triggered by trying to write a Perl script that computes the mean of Pareto variates. Perl only provides the rand random number generator.

References

[1] P. Barford and M. Crovella, "*Generating representative web workloads for network and server performance evaluation*". In *SIGMETRICS Conf. Measurement & Modeling of Comput. Syst.*, pp. 151–160, Jun 1998.

[2] M. E. Crovella and A. Bestavros, "*Self-similarity in world wide web traffic: evidence and possible causes*". In *SIGMETRICS Conf. Measurement & Modeling of Comput. Syst.*, pp. 160–169, May 1996.

[3] M. E. Crovella and L. Lipsky, "*Long-lasting transient conditions in simulations with heavy-tailed workloads*". In *Winter Simulation conf.*, Dec 1997.

[4] S. D. Gribble, G. S. Manku, D. Roselli, E. A. Brewer, T. J. Gibson, and E. L. Miller, "*Self-similarity in file systems*". In *SIGMETRICS Conf. Measurement & Modeling of Comput. Syst.*, pp. 141–150, Jun 1998.

[5] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.

[6] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "*On the self-similar nature of Ethernet traffic*". *IEEE/ACM Trans. Networking* **2(1)**, pp. 1–15, Feb 1994.

[7] V. Paxon and S. Floyd, "*Wide-area traffic: the failure of Poisson modeling*". *IEEE/ACM Trans. Networking* **3(3)**, pp. 226–244, Jun 1995.

[8] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.