Performance Evaluation with Heavy Tailed Distributions

(Extended Abstract)*

Mark E. Crovella

Department of Computer Science
Boston University
111 Cummington St.
Boston MA USA 02215
crovella@cs.bu.edu

1 Introduction

Over the last decade an important new direction has developed in the performance evaluation of computer systems: the study of *heavy-tailed* distributions. Loosely speaking, these are distributions whose tails follow a power-law with low exponent, in contrast to traditional distributions (*e.g.,* Gaussian, Exponential, Poisson) whose tails decline exponentially (or faster). In the late '80s and early '90s experimental evidence began to accumulate that some properties of computer systems and networks showed distributions with very long tails [7, 28, 29], and attention turned to heavy-tailed distributions in particular in the mid '90s [3, 9, 23, 36, 44].

To define heavy tails more precisely, let X be a random variable with cumulative distribution function $F(x) = P[X \leq x]$ and its complement $\bar{F}(x) = 1 - F(x) = P[X > x]$. We say here that a distribution $F(x)$ is *heavy tailed* if

$$\bar{F}(x) \sim cx^{-\alpha} \quad 0 < \alpha < 2 \tag{1}$$

for some positive constant c, where $a(x) \sim b(x)$ means $\lim_{x\to\infty} a(x)/b(x) = 1$. This definition restricts our attention somewhat narrowly to distributions with strictly polynomial tails; broader classes such as the *subexponential* distributions [19] can be defined and most of the qualitative remarks we make here apply to such broader classes.

Heavy tailed distributions behave quite differently from the distributions more commonly used in performance evaluation (*e.g.,* the Exponential). In particular, when sampling random variables that follow heavy tailed distributions, the probability of very large observations occurring is non-negligible. In fact, under our definition, heavy tailed distributions have *infinite variance,* reflecting the extremely high variability that they capture; and when $\alpha \leq 1$, these distributions have *infinite mean.*

* This is a revised version of a paper originally appearing in *Lecture Notes in Computer Science 1786,* pp. 1–9, March 2000.

2 Evidence

The evidence for heavy-tailed distributions in a number of aspects of computer systems is now quite strong. The broadest evidence concerns the sizes of data objects stored in and transferred through computer systems; in particular, there is evidence for heavy tails in the sizes of:

- Files stored on Web servers [3, 9];
- Data files transferred through the Internet [9, 36];
- Files stored in general-purpose Unix filesystems [25]; and
- I/O traces of filesystem, disk, and tape activity [21, 38–40]

This evidence suggests that heavy-tailed distributions of data objects are widespread, and these heavy-tailed distributions have been implicated as an underlying cause of *self-similarity* in network traffic [9, 30, 35, 44].

Next, measurements of job service times or process execution times in general-purpose computing environments have been found to exhibit heavy tails [17, 23, 28].

A third area in which heavy tails have recently been noted is in the distribution of node degree of certain graph structures. Faloutsos *et al.* [14] show that the inter-domain structure of the Internet, considered as a directed graph, shows a heavy-tailed distribution in the outdegree of nodes. These studies have already influenced the way that Internet-like graph topologies are created for use in simulation [32, 26]. Another study shows that the same is true (with respect to both indegree and outdegree) for certain sets of World Wide Web pages which form a graph due to their hyperlinked structure [1]; this result has been extended to the Web as a whole in [6].

Finally, a phenomenon related to heavy tails is the so-called *Zipf's Law* [45]. Zipf's Law relates the "popularity" of an object to its location in a list sorted by popularity. More precisely, consider a set of objects (such as Web servers, or Web pages) to which repeated references are made. Over some time interval, count the number of references made to each object, denoted by R. Now sort the objects in order of decreasing number of references made and let an object's place on this list be denoted by n. Then Zipf's Law states that

$$R = cn^{-\beta}$$

for some positive constants c and β. In its original formulation, Zipf's Law set $\beta = 1$ so that popularity (R) and rank (n) are inversely proportional. In practice, various values of β are found, with values often near to or less than 1. Evidence for Zipf's Law in computing systems (especially the Internet) is widespread [2, 13, 18, 33]; a good overview of such results is presented in [5].

3 Implications of Heavy Tails

Unfortunately, although heavy-tailed distributions are prevalent and important in computer systems, their unusual nature presents a number of problems for performance analysis.

The fact that even low-order distributional moments can be infinite means that many traditional system metrics can be undefined. As a simple example, consider the mean queue length in an $M/G/1$ queue, which (by the Pollaczek-Khinchin formula) is proportional to the second moment of service time. Thus, when service times are drawn from a heavy-tailed distribution, many properties of this queue (mean queue length, mean waiting time) are infinite. Observations like this one suggest that performance analysts dealing with heavy tails may need to turn their attention away from means and variances and toward understanding the full distribution of relevant metrics. Most early work in this direction has focused on the shape of the tail of such distributions (*e.g.,* [34]).

Some heavy-tailed distributions apparently have no convenient closed-form Laplace transforms (*e.g.,* the Pareto distribution), and even for those distributions possessing Laplace transforms, simple systems like the the $M/G/1$ must be evaluated numerically, and with considerable care [41].

In practice, random variables that follow heavy tailed distributions are characterized as exhibiting many small observations mixed in with a few large observations. In such datasets, most of the observations are small, but most of the contribution to the sample mean or variance comes from the rare, large observations. This means that those sample statistics that are defined converge very slowly. This is particularly problematic for simulations involving heavy tails, which many be very slow to reach steady state [12].

Finally, because arbitrarily large observations can not be ruled out, issues of scale should enter in to any discussion of heavy tailed models. No real system can experience arbitrarily large events, and generally one must pay attention to the practical upper limit on event size, whether determined by the timescale of interest, the constraints of storage or transmission capacity, or other system-defined limits. On the brighter side, a useful result is that it is often reasonable to substitute finitely-supported distributions for the idealized heavy-tailed distributions in analytic settings, as long as the approximation is accurate over the range of scales of interest [16, 20, 22].

4 Taking Advantage of Heavy Tails

Despite the challenges they present to performance analysis, heavy tailed distributions also exhibit properties that can be exploited in the design of computer systems. Recent work has begun to explore how to take advantage of the presence of heavy tailed distributions to improve computer systems' performance.

4.1 Two Important Properties

In this regard, there are two properties of heavy tailed distributions that offer particular leverage in the design of computer systems. The first property is related to the fact that heavy tailed distributions show declining hazard rate, and is most concisely captured in terms of conditional expectation:

$$E[X|X > k] \sim k$$

when X is a heavy tailed random variable and k is large enough to be "in the tail." We refer to this as the *expectation paradox,* after [31, p. 343]; it says that if we are making observations of heavy-tailed interarrivals, then the longer we have waited, the longer we should expect to wait. (The expectation is undefined when $\alpha \leq 1$, but the general idea still holds.) This should be contrasted with the case when the underlying distribution has exponential tails or has bounded support above (as in the uniform distribution); in these cases, eventually one always gets to the point where the longer one waits, the less time one should expect to continue waiting.

The second useful property of heavy tailed distributions we will call the *mass-count disparity.* This property can be stated formally as [19]:

$$\lim_{x \to \infty} \frac{P[X_1 + ... + X_n > x]}{P[\max(X_1, ..., X_n) > x]} = 1 \text{ for all } n \geq 2$$

which is the case when the X_i are i.i.d. positive random variables drawn from a heavy-tailed distribution. This property states that when considering collections of observations of a heavy-tailed random variable, the aggregated mass contained in the small observations is negligible compared to the largest observation in determining the likelihood of large values of the sum.

In practice this means that the majority of the mass in a set of observations is concentrated in a very small subset of the observations. This can be visualized as a box into which one has put a few boulders, and then filled the rest of the way with sand. This mass-count disparity means that one must be careful in "optimizing the common case" [27]. The typical *observation* is small; the typical *unit of work* is contained in a large observation.

This disparity can be studied by defining the mass-weighted distribution function:

$$F_w(x) = \frac{\int_{-\infty}^{x} u \, dF(u)}{\int_{-\infty}^{\infty} v \, dF(v)} \tag{2}$$

and comparing $F_w(x)$ with $F(x)$. Varying x over its valid range yields a plot of the fraction of total mass that is contained in the fraction of observations less than x. An example of this comparison is shown in Figure 1. This figure shows $F_w(x)$ vs. $F(x)$ for the Exponential distribution, and for a particular heavy-tailed distribution. The heavy-tailed distribution is chosen to correspond to empirical measurements of file sizes in the World Wide Web [4]; it has $\alpha = 1.0$. Since the denominator in (2) is infinite for heavy tailed distributions with $\alpha \leq 1$, the actual distribution used has been truncated to span six orders of magnitude — which is reasonable for file size distributions (which can range in size from bytes to megabytes).

The figure shows that for the Exponential distribution, the amount of mass contained in small observations is roughly commensurate with the fraction of total observations considered; *i.e.,* the curve is not too far from the line $y = x$. On the other hand, for the heavy tailed distribution, the amount of mass is not at all commensurate with the fraction of observations considered; about 60% of the mass is contained in the upper 1% of the observations! This is consistent

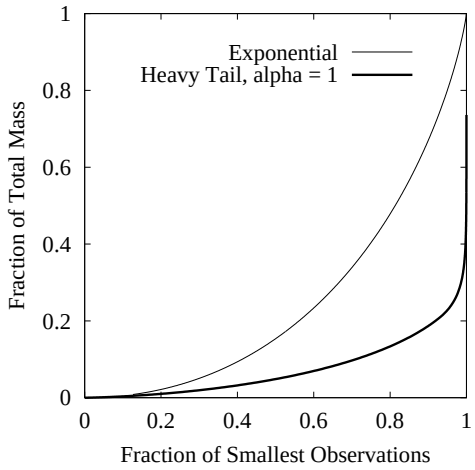

Fig. 1. Total Mass as a Function of Smallest Observations

with results in [37] showing that 50-80% of the bytes in FTP transfers are due to the largest 2% of all transfers.

4.2 Exploiting The Heavy Tail Properties

Once these properties are understood, they can be exploited in a number of ways to improve system performance. This section summarizes some (though not all) recent attempts to do this.

Load Balancing in Distributed Systems In some distributed systems, tasks can be pre-empted and moved from one node to another, which can improve load balance. However, the cost of migration is not trivial and can outweigh performance gains from improved load balance if not used carefully. In [23], the authors show that previous assessments of the potential for pre-emptive migration had mainly used exponential tasks size assumptions and concluded that the potential gains from task migration were small. However, once the task size distribution is understood to be heavy-tailed, two benefits emerge: 1) the mass-count disparity means that relative few tasks need to be migrated to radically improve load balance; and 2) the expectation paradox means that a task's lifetime to date is a good predictor of its expected future lifetime. Taken together, these two benefits form the foundation for a enlightened load balancing policy that can significantly improve the performance of a wide class of distributed systems.

When pre-emption is not an option, understanding of heavy tailed distributions can still inform load balancing policies. The question in these systems is "which queue should an arriving task join?" In the case when service at the nodes is FCFS, and knowledge is available about the size of the arriving task, the best policy is commonly assumed to be joining the queue with the shortest expected

delay [43] although this is known to be best only for task size distributions with increasing failure rate. In [24], the authors show a better policy for the case in which task sizes have a heavy-tailed distribution, which they call SITA-E. The idea is to assign an incoming task to a queue based on the incoming task's size. Each queue handles tasks whose sizes lie in a contiguous range, and ranges are chosen so as to equalize load in expectation. This policy is shown to significantly outperform shortest-expect-delay assignment, when $1 < \alpha \leq 2$. The benefits of the policy accrue primarily from the the mass-count disparity in task sizes: grouping like tasks together means that the vast majority of tasks are sent to only a few queues; at these queues, task size variability is dramatically reduced and so FCFS service is very efficient.

Finally, in another paper [8,11], the authors show that in the same setting (distributed system of FCFS servers, task sizes are heavy tailed, and incoming task sizes are known) the *expected slowdown* metric is optimized by policies that do *not* balance load. (Slowdown is defined as a job's waiting time in queue divided by its service demand.) This is possible because of the mass-count disparity; when most tasks are sent to only a few queues, reducing the load at those queues decreases the slowdown experienced at those queues. In this case, most tasks experience decreased slowdown, while the relatively few large tasks experience only slightly increased slowdown. In expectation, slowdown is decreased.

Scheduling in Web Servers In single-node systems, attention has been given to the scheduling issue. Most systems use a variant of timesharing to schedule tasks, possibly incorporating multilevel feedback; this is effective when task sizes are unknown. In [22], the authors argue that Web servers are in a unusual position; they can estimate task size upon task arrival because, for static Web pages, the file size is known at request time. As a result, they argue for the use of shortest-remaining-processing-time (SRPT) scheduling within Web servers. One significant drawback of SRPT is that it improves the response time of small tasks at the expense of large tasks; however the authors argue that this is acceptable when tasks follow heavy-tailed distributions such as are encountered in the Web. The reason is that the mass-count disparity means that under SRPT, although large tasks are interrupted by small tasks, the small tasks represent only a minor fraction of total system load. Thus the great majority of tasks have their response time improved, while the relatively few large tasks are not seriously punished. In [10] the authors describe an actual Web server implemented to use this scheduling policy. The paper shows evidence that the new server exhibits mean response times 4-5 times lower than a popularly deployed server (Apache); and that the performance impacts on large tasks are relatively mild.

Routing and Switching in the Internet In Internet traffic management, a number of improved approaches to routing and switching have been proposed, based on the observation that the lengths of bulk data flows in the Internet exhibit heavy tails.

One promising routing technique is to use switching hardware, by creating *shortcuts* (temporary circuits) for long sequences of packets that share a common

source and destination. Shortcuts provide the benefits of fast switch-based routing, at the expense of network and switch overhead for their setup. The authors in [15] argue that Web traffic can be efficiently routed using this technique. Their results rely on the mass-count disparity, showing that the majority of the bytes can be routed by creating shortcuts for only a small fraction of all data flows. They show that in some settings, a setup threshold of 25 packets (the number of same-path packets to observe before creating a switched connection) is sufficient to eliminate 90% of the setup costs while routing more than 50% of the bytes over switched circuits. The choice of threshold implicitly makes use of the expectation paradox: longer thresholds can be used to offset larger setup costs, since longer thresholds identify flows whose expected future length is longer as well.

Another proposed routing technique is *load-sensitive* routing. Load sensitive routing attempts to route traffic around points of congestion in the network; current Internet routing only makes use of link state (up or down). Unfortunately, load-sensitive routing can be expensive and potentially unstable if applied to every routing decision. However, the authors in [42] show that if applied only to the long-lived flows, it can be efficient and considerably more stable. The success of this technique relies on the heavy tailed distribution of Internet flows: the mass-count disparity means that a large fraction of bytes can be routed by rerouting only a small fraction of the flows; and the expectation paradox allows the policy to observe a flow for some period of time to classify it as a long flow.

Acknowledgments

The author is grateful to Mor Harchol-Balter, with whom some of the ideas in this paper were developed and clarified. This work was supported by NSF grants CCR-9501822, CCR-9706685, and by grants from Hewlett-Packard Laboratories.

References

1. Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world wide web. *Nature*, 401:130–131, 1999.
2. Virgílio Almeida, Azer Bestavros, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the WWW. In *Proceedings of 1996 International Conference on Parallel and Distributed Information Systems (PDIS '96)*, pages 92–103, December 1996.
3. Martin F. Arlitt and Carey L. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997.
4. Paul Barford and Mark E. Crovella. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of Performance '98/SIGMETRICS '98*, pages 151–160, July 1998.
5. Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and zipf-like distributions: Evidence and implications. In *Proceedings of INFOCOM '99*, pages 126–134, 1999.

6. Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web: experiments and models. In *Proceedings the Ninth World Wide Web Conference (WWW9)*, 2000.
7. R. Cáceres, P. B. Danzig, S. Jamin, and D. J. Mitzel. Characteristics of wide-area TCP/IP conversations. *Computer Communication Review*, 21, 1991.
8. M. E. Crovella, M. Harchol-Balter, and C. D. Murta. Task assignment in a distributed system: Improving performance by unbalancing load. Technical Report TR-97-018, Boston University Department of Computer Science, October 31 1997.
9. Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.
10. Mark E. Crovella, Robert Frangioso, and Mor Harchol-Balter. Connection scheduling in Web servers. In *1999 USENIX Symposium on Internet Technologies and Systems (USITS '99)*, 1999.
11. Mark E. Crovella, Mor Harchol-Balter, and Cristina Duarte Murta. Task assignment in a distributed system: Improving performance by unbalancing load. In *Proceedings of SIGMETRICS '98 (poster paper)*, July 1998.
12. Mark E. Crovella and Lester Lipsky. Simulations with heavy-tailed workloads. In Kihong Park and Walter Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley / Wiley Interscience, New York, 1999.
13. Carlos A. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.
14. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of SIGCOMM '99*, 1999.
15. Anja Feldmann, Jennifer Rexford, and Ramon Caceres. Efficient policies for carrying web traffic over flow-switched networks. *IEEE/ACM Transactions on Networking*, December 1998.
16. Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In *Proceedings of IEEE INFOCOM'97*, pages 1098–1116, April 1997.
17. Sharad Garg, Lester Lipsky, and Maryann Robbert. The effect of power-tail distributions on the behavior of time sharing computer systems. In *1992 ACM Symposium on Applied Computing*, Kansas City, MO, March 1992.
18. Steven Glassman. A caching relay for the World Wide Web. In *Proceedings of the First International World Wide Web Conference*, pages 69–76, 1994.
19. Charles M. Goldie and Claudia Kluppelberg. Subexponential distributions. In Robert J. Adler, Raisa E. Feldman, and Murad S. Taqqu, editors, *A Practical Guide To Heavy Tails*, pages 435–460. Chapman & Hall, New York, 1998.
20. Michael Greiner, Manfred Jobmann, and Lester Lipsky. The importance of power-tail distributions for telecommunication traffic models. *Operations Research*, 41, 1999.
21. S. D. Gribble, G. S. Manku, D. Roselli, E. A. Brewer, T. J. Gibson, and E. L. Miller. Self-similarity in file systems. In *Proceedings of SIGMETRICS '98*, pages 141–150, 1998.
22. M. Harchol-Balter, M. E. Crovella, and S. Park. The case for SRPT scheduling in Web servers. Technical Report MIT-LCS-TR-767, MIT Lab for Computer Science, October 1998.

23. M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, 1997.
24. Mor Harchol-Balter, Mark E. Crovella, and Cristina D. Murta. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, Special Issue on Software Support for Distributed Computing, September 1999.
25. Gordon Irlam. Unix file size survey - 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.
26. Cheng Jin, Qian Chen, and Sugih Jamin. Inet: internet topology generator. Technical Report CSE-TR-433-00, U. Michigan Computer Science, 2000.
27. Butler W. Lampson. Hints for computer system design. *Proceedings of the Ninth SOSP, in Operating Systems Review*, 17(5):33–48, October 1983.
28. W. E. Leland and T. J. Ott. Load-balancing heuristics and process behavior. In *Proceedings of Performance and ACM Sigmetrics*, pages 54–69, 1986.
29. W. E. Leland and D. V. Wilson. High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In *Proceeedings of IEEE Infocomm '91*, pages 1360–1366, Bal Harbour, FL, 1991.
30. W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.
31. Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York, 1983.
32. Alberto Medina, Ibrahim Matta, and John Byers. BRITE: a flexible generator of internet topologies. Technical Report BU-CS-TR-2000-05, Boston University Computer Science, January 2000.
33. Norifumi Nishikawa, Takafumi Hosokawa, Yasuhide Mori, Kenichi Yoshida, and Hiroshi Tsuji. Memory-based architecture for distributed WWW caching proxy. *Computer Networks and ISDN Systems*, 30:205–214, 1998.
34. I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.
35. Kihong Park, Gi Tae Kim, and Mark E. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the Fourth International Conference on Network Protocols (ICNP'96)*, pages 171–180, October 1996.
36. Vern Paxson. Empirically-derived analytic models of wide-area tcp connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.
37. Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, pages 226–244, June 1995.
38. D. Peterson and R. Grossman. Power laws in large shop DASD I/O activity. In *CMG Proceedings*, pages 822–833, December 1995.
39. David L. Peterson. Data center I/O patterns and power laws. In *CMG Proceedings*, December 1996.
40. David L. Peterson and David B. Adams. Fractal patterns in DASD I/O traffic. In *CMG Proceedings*, December 1996.
41. Matthew Roughan, Darryl Veitch, and Michael Rumsewicz. Computing queue-length distributions for power-law queues. In *Proceedings of INFOCOM '98*, pages 356–363, 1998.
42. Anees Shaikh, Jennifer Rexford, and Kang Shin. Load-sensitive routing of long-lived IP flows. In *Proceedings of ACM SIGCOMM '99*, pages 215–226, September 1999.

43. R. W. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15:406–413, 1978.
44. Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.
45. G. K. Zipf. *Human Behavior and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA, 1949.