# Analysis of Storage System Activity

Nicholas Dingle and Uli Harder [*]

**Abstract**

In this paper we investigate disk activity caused by accesses to large databases stored on modern storage systems. We show that read and write requests do not arrive according to a Poisson process but rather show a power law behaviour. Also, the aggregated rates of requests show signs of long range dependence through convincing $1/f$ noise in the power spectrum of the process. We then present a naïve simulation model of read and write request performance that is parameterised by this empirical data, and compare observed and predicted response times. Finally we speculate on the origins of the observed behaviour and discuss its consequences for large systems.

## 1 Introduction

Power laws crop up with alarming regularity. Within computer science they have been observed in network traffic [1, 2, 3], file size distributions [4], web server requests [5] and printer traffic [6], to name but a few examples. In a wider context they have also been observed in a wide range of human behaviours [7, 8, 9, 10, 11], suggesting that there is something about the way in which our brains work that manifests itself in the systems that we build and use.

This paper investigates the phenomena of power laws in the context of computer storage systems (either individual disks or RAID arrays). It is well known that such systems typically experience heavy-tailed I/O request inter-arrival distributions [12, 13, 14, 15]. In contrast, analytical performance models of these systems (e.g. [16, 17, 18, 19, 20]) typically assume that I/O requests arrive according to a Poisson distribution, for reasons of tractability.

Our investigation is performed using two datasets captured in academic and commercial contexts. We begin by analysing the properties of the I/O request inter-arrival, service and response time distributions to determine if they are consistent with each-other and with the results of previous studies. We also compare these properties with results from similar analysis of other computing domains (e.g. network traffic) to determine if there are any similarities.

We then use these extracted inter-arrival and service time distributions to parameterise a naïve simulation of storage system performance, and compare the resulting predicted response time distributions with those observed in the datasets. Our intention is to demonstrate the extent to which it is possible to model storage system performance without developing a detailed low-level representation of the storage system under consideration, and furthermore to

[*] Department of Computing, Imperial College London Huxley Building, 180 Queens Gate, London SW7 2RH, UK {njd200|uh}@doc.ic.ac.uk

demonstrate the importance of capturing the power law behaviour observed in real-life systems instead of relying on a Poisson approximation.

The remainder of this paper is organised as follows. Section 2 briefly presents the theoretical background to power laws, before Section 3 introduces the two datasets which will be used throughout the paper and presents our analysis of their relevant properties. Section 4 then introduces our naïve simulation model of a storage system, which is parameterised according to the values extracted in the previous section, and compares the predicted response times against those observed in the datasets. Section 5 concludes and considers possible directions for future work.

## 2 Background

A function like a probability density function $p(x)$, $x \in \mathbb{R}$, is said to follow a a power law if

$$p(x) \propto \beta x^{\gamma}$$

as $x \rightarrow \infty$, for $\beta > 0, \gamma \neq 0$. Long range dependence of a time series manifests itself in the auto-correlation function showing power law decay. For a given continuous time series $X(t)$ with zero mean its auto-correlation function $C(\cdot)$ at lag $\tau$ is defined as

$$C(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{T} dt X(t+\tau) X(t).$$

The power spectrum of a time series is the Fourier transform of the auto-correlation function. Due to the Wiener-Khintchine theorem [21, 22] the power spectrum can also be calculated as

$$S(f) = \lim_{T \rightarrow \infty} \frac{1}{4\pi T} \Big| \int_{-T}^{T} dt X(t) e^{-i2\pi f t} \Big|^2.$$

This tends to be a cheaper and more accurate way of computing the power spectrum. If the auto-correlation function shows a power law, so does the power spectrum. If $S(f)$ behaves like $S(f) \propto 1/f^{\alpha}$, then

$$C(\tau) \propto |\tau|^{\alpha - 1} \quad \text{for } 0 < \alpha < 1$$

$\alpha$ close to but smaller than 1 corresponds to long range dependence. We use the statistical package R [23] to compute discrete power spectra. There are many other ways to check whether a time series shows long-range dependence: see for instance [24].

A Pareto distribution is defined as

$$p(x) = \alpha k^{\alpha} x^{-\alpha - 1},$$

where $\alpha, k > 0$ and $x \geq k$. This is one of the simplest power law distributions and is closely related to the Zipf law [25]. In a double logarithmic plot, $\log(x)$ against $\log(p(x))$, the Pareto distribution is a straight line with gradient $-\alpha - 1$. Another function exhibiting a power law is the symmetric Cauchy distribution with a pdf given by

$$p_c(x) = \frac{1}{\pi} \frac{s}{s^2 + x^2}$$

where $s > 0$. We can turn this into the asymmetric Cauchy distribution by multiplying this pdf by two and restricting $x \geq 0$. For large $x$ this behaves like $1/x^2$

$$\tilde{p}(x) = \frac{2}{\pi} \frac{s}{s^2 + x^2} \text{ where } s > 0 \text{ and } x \geq 0$$

In a double logarithmic plot the Cauchy distribution will tend asymptotically to a straight line with gradient $-2$. In this paper we use the truncated Cauchy distribution whose pdf is given by:

$$p(x) = \begin{cases} \tilde{p}(x)/C & 0 \leq x_{\min} \leq x \leq x_{\max} \\ 0 & \text{else} \end{cases}$$

where $C$ is a normalisation constant

$$C = \int_{x_{\min}}^{x_{\max}} \tilde{p}(x) dx.$$

The truncated Cauchy distribution has finite moments, but does not obey a *power law* as it is a finite distribution.

# 3 Analysis

In this section we investigate the properties of two I/O request datasets. The first captured I/O behaviour resulting from requests to the research database of the Department of Computing (DoC) at Imperial College London over the period of one week [26]. The second was obtained from the Storage Networking Industry Association (SNIA) website and is an I/O trace from a Microsoft Buildserver for a single day [27].

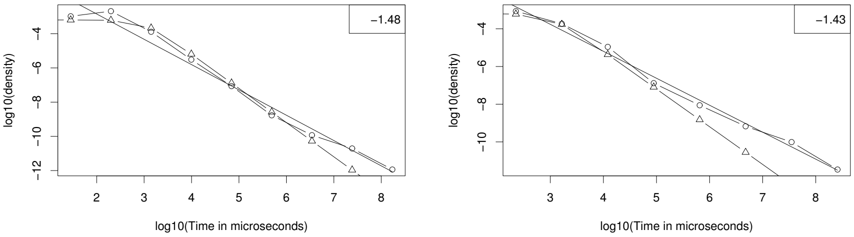

Figure 1: The inter-arrival times between read (left) and write (right) requests to the DoC database. The circles are the estimated pdf and the triangles a fitted Cauchy distribution. The number in the legend is the gradient of the straight line fitted through the tail of the pdf.

Figures 1 and 2 show the inter-arrival times for requests to the DoC database and Buildserver respectively in a double logarithmic plot. In both cases it can be seen that the arrival patterns are definitely non-Poisson as they show a power law in the tail of the arrival patterns and there is definitely a huge variation in inter-arrival times (e.g. for the DoC database servers inter arrival times range from $10\mu s$ to about four minutes). Interestingly, there appears to

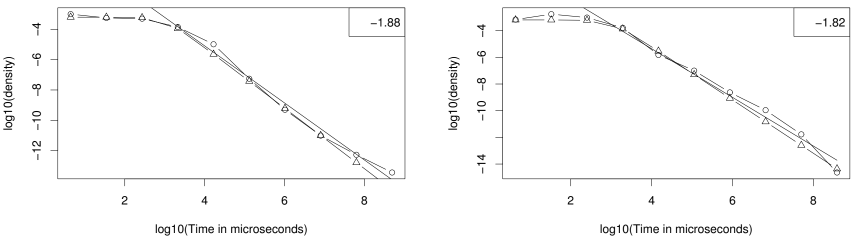

Figure 2: The inter-arrival times between read (left) and write (right) requests to the Buildserver. The circles are the estimated pdf and the triangles a fitted Cauchy distribution. The number in the legend is the gradient of the straight line fitted through the tail of the pdf.

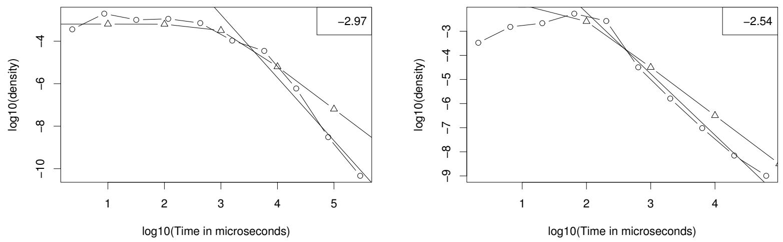

Figure 3: The service time for reads (bottom left) and writes (bottom right) for the Buildserver. The circles are the estimated pdf and the triangles a fitted Cauchy distribution. The number in the legend is the gradient of the straight line fitted through the tail of the pdf.

be little difference between the inter-arrival time distributions for reads and writes, and also little difference between inter-arrival time distributions across the two completely different systems. In both cases the distributions appear to be power laws asymptotically, though with a slightly different gradient between the systems.

Similarly, the service times for the Buildserver shown in Figure 3 are non-Poisson and show power law behaviour in their tails. We theorise that this stems from a heavy-tailed distribution of file sizes; such a distribution has observed to occur in other studies [4, 28] and we assume that service time is in some way proportional to the size of the file being accessed.

The Cauchy distribution appears to be a good fit for both inter-arrival and service time distributions as unlike the Pareto distribution it captures the small inter-arrival and service times in addition to the heavy tail. Figures 1, 2 and 3 all display Cauchy distributions fitted to the distributions extracted from the datasets. The parameters of these distributions along with those of the observed distributions are given in Table 1. We should however emphasise that the truncated Cauchy distributions are fitted by eye. A better method would be to try a non-linear fit or the attempt to match means of the real data with

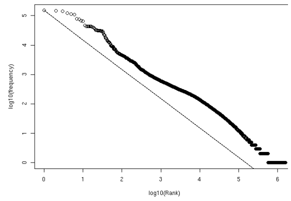

Figure 4: The rank frequency plot of files on the Buildserver. The straight line, inserted for comparison, has a gradient of $-1$.

that of the truncated Cauchy distribution. One should also check the data against a log-normal distribution which has been proposed as model for file size distributions [28].

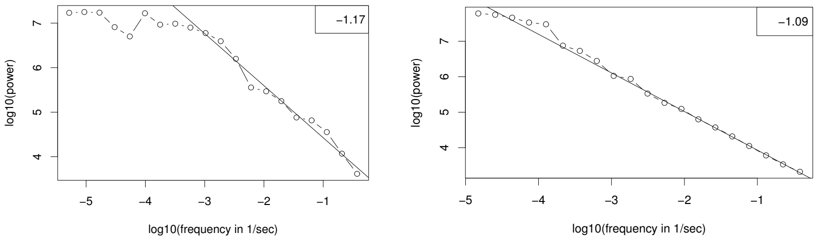

Figure 5: The power spectrum of the read requests per second to the DoC database (left) and the Buildserver (right). The number in the legend is the gradient of the straight line fitted trough the tail of the power spectrum.

We have also plotted the power spectra of read requests per second for both the DoC and Buildserver datasets, as shown in Figure 5. Both time series display $1/f$ noise. This is is similar to what has been observed in network traffic [1, 2, 3], but the fit here is more convincing than in previous studies. This is a sign of long-range dependence in the data as we explained in Section 2 because the power spectrum is closely related to the auto-correlation function, and is in keeping with the results of other studies of storage system inter-arrival times [15]. Given that the gradient of the power spectra, especially that of the Buildserver, is close to $-1$ one could speculate that the system could be in a critical state. Many physical systems show $1/f$ noise and a subset of them are systems that show self-organised criticality (SOC) [29], which means that they stay in the critical region irrespective of external parameters.

In Figure 4 we show the rank frequency plot of the file requests to the Buildserver, plotting the rank of a file (rank 1 is the most popular file) against the frequency with which it was accessed. This plot show a convincing power law and is extremely well approximated by a Zipf law. In [25] it is explained how the rank-frequency plot is related to cumulative distribution function. The plot shows that a small number files are requested frequently and a large number very rarely. The Zipf law gives rise to the so-called 80-20 (or Pareto) principle. So, when it comes to disk access everything seems to point to power laws.

# 4 Simulation Model

| | $s$ | $x_{\min}$ | $x_{\max}$ | $\mu_c$ | $\sigma_c$ | $\mu$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| $\Delta t_r^d$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $4.0 \times 10^5$ | $1.6 \times 10^6$ |
| $\Delta t_w^d$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $3.0 \times 10^5$ | $4.6 \times 10^6$ |
| $\Delta t_r^b$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $1.4 \times 10^4$ | $1.7 \times 10^6$ |
| $\Delta t_w^b$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $1.5 \times 10^4$ | $7.8 \times 10^5$ |
| $s_r^b$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $1.8 \times 10^3$ | $2.9 \times 10^3$ |
| $s_w^b$ | $5 \times 10^1$ | 1.0 | $10^5$ | $2.4 \times 10^2$ | $1.8 \times 10^3$ | $1.4 \times 10^2$ | $5.1 \times 10^2$ |
| $W_r^b$ | $10^3$ | 1.0 | $10^6$ | $4.4 \times 10^3$ | $2.5 \times 10^4$ | $6.0 \times 10^3$ | $1.8 \times 10^4$ |
| $W_w^b$ | $2 \times 10^2$ | 1.0 | $10^6$ | $1.1 \times 10^3$ | $1.1 \times 10^4$ | $2.2 \times 10^2$ | $3.0 \times 10^3$ |
| $W_r^s$ | $10^6$ | 1.0 | $10^8$ | $2.9 \times 10^6$ | $7.4 \times 10^6$ | $3.5 \times 10^6$ | $2.0 \times 10^7$ |
| $W_w^s$ | $10^2$ | 1.0 | $10^6$ | $5.9 \times 10^2$ | $8.0 \times 10^3$ | $2.8 \times 10^2$ | $2.0 \times 10^3$ |

Table 1: $\Delta t.$ are the read and write inter arrival times $s.$ are the service times and $W.$ are the response times. $s$, $x_{\min}$, $x_{\max}$ are the parameters of the *fitted* Cauchy distribution, $\mu_c$ and $\sigma_c$ are the mean and standard deviation of the Cauchy distribution and $\mu$ and $\sigma$ are the mean and standard deviation of the real or simulated data. All values are in microseconds. The superindices $d, b, s$ correspond to the DoC, Buildserver and simulation data respectively.

We have implemented a naïve simulation model of a storage system using the Java Implementation of a Network-of-Queues Simulation (JINQS) [30] package. It is comprised of three nodes:

- The source node, which generates I/O requests according to the distributions described in Table 1.
- A single G/G/1 queue with service time distribution specified in Table 1, which represents the storage system. The queue has a processor-sharing queueing discipline to model the way in which a storage system may process multiple high-level requests in parallel.
- The sink node, which collects completed I/O requests and stores their overall response time.

To parameterise this model we have used the inter-arrival time and service time values captured in the Microsoft Buildserver trace from SNIA (plotted in Figures 2 and 3 respectively). As in Section 3, we have assumed that these follow a Cauchy distribution with the parameters given in Table 1 as this appears to fit the characteristics of the data well. Unfortunately, the DoC dataset did not include service times and so could not be used to parameterise the model.

The Buildserver dataset captures the response time for each I/O request issued, and we can therefore compare these observed times with those predicted by our simulation. Figure 6 shows the response times for reads and writes as produced by our simulation. They appear to capture well the heavy-tailed nature of the original request response times shown in Figure 7, although as can be seen from Table 1 there is considerable difference between the simulated and true means and standard deviations.

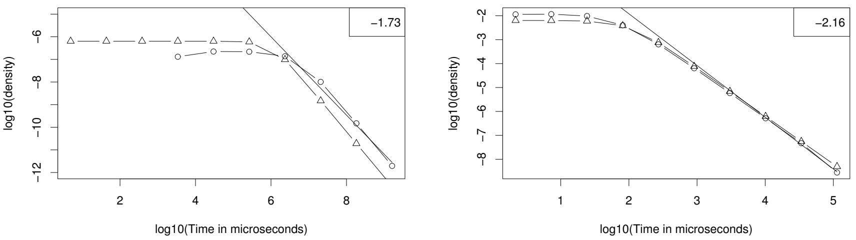

Figure 6: The simulated response time for reads (left) and writes (right) for the Buildserver. The circles are the estimated pdf and the triangles a fitted Cauchy distribution. The number in the legend is the gradient of the straight line fitted trough the tail of the pdf.

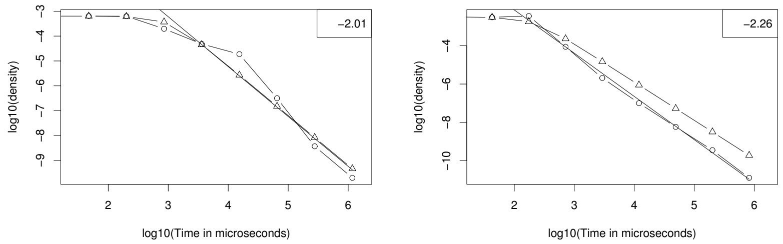

Figure 7: The response times for reads (left) and writes (right)for the Buildserver. The circles are the estimated pdf and the triangles a fitted Cauchy distribution. The number in the legend is the gradient of the straight line fitted trough the tail of the pdf.

# 5 Conclusions and Future Work

We have presented an analysis of storage system activity based on data taken from two real-world datasets. We have established that the inter-arrival time and service time distributions are not Poisson, as has been assumed in a number of prior analytical models of such systems, and suggested that a heavy-tailed Cauchy distribution provides a good fit to the observed data. Furthermore, we have observed convincing $1/f$ noise in the power spectra of read requests per second in both data sets, which is a sign of long-range dependence. The I/O request response times are also heavy-tailed, which suggests that there are no time-outs or restarts, and that requests are not dropped if they take too long.

These observations have major implications for the design of real-life storage systems. Request arrivals will be bursty, and thus queues will sometimes become very long and buffers will not be large enough to hold all requests. Simply increasing the size of these buffers is not the answer, however, as they introduce extra queueing (when perhaps the user would be better to restart the request at a later time when load was less) and are vulnerable to data loss in the event of power failure.

We have implemented a simple simulation parameterised with the inter-arrival and service time distributions observed in the data, and have shown that it reproduces the heavy-tailed response times observed in real life. What this simulator does not provide, however, is any insight into why these phenomena occur in the first place: we can tell that they do occur but cannot conclusively say why. This remains a major focus of future work as we seek to determine whether the power law behaviour comes from human nature, technological aspects (e.g. file-size distributions) or a combination of the two.

We would also like to investigate the queue length behaviour for the Buildserver as this would be of importance for buffer size considerations. Similarly, the data should allow us to compute the utilisation.

## Acknowledgements

We would like to thank the Computer Support Group at DoC for capturing the database trace files and making them available to us.

## References

[1] W. E. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of ethernet traffic (extended version),” *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, February 1994.

[2] A. J. Field, U. Harder, and P. G. Harrison, “Measurement and modelling of self-similar traffic in computer networks,” *IEE Proceedings - Communications*, vol. 151, pp. 355–363, August 2004.

[3] A. J. Field, U. Harder, and P. G. Harrison, “Network traffic behaviour in switched Ethernet systems,” *Performance Evaluation*, vol. 58, pp. 243–260, November 2004.

[4] G. Irlam, “Unix file size survey.” `http://www.base.com/gordoni/ufs93.html`.

[5] M. E. Crovella and A. Bestavros, “Self-similarity in World Wide Web traffic: evidence and possible causes,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[6] U. Harder and M. Paczuski, “Correlated dynamics in human printing behaviour,” *Physica A*, vol. 361, no. 1, pp. 329–336, 2006.

[7] Y. Chen, M. Ding, and J. A. S. Kelso, “Long memory processes ($1/f^\alpha$ Type) in human coordination,” *Physical Review Letters*, vol. 79, pp. 4501–4504, December 1997.

[8] D. L. Gilden, T. Thornton, and M. W. Mallon, “1/f noise in human cognition,” *Science*, vol. 267, pp. 1837–1839, March 1995.

[9] K. Linkenkaer-Hansen, V. V. Nikouline, J. M. Palva, and R. J. Ilmoniemi, “Long-range temporal correlations and scaling behaviour in human brain oscillations,” *The Journal of Neuroscience*, vol. 21, pp. 1370–1377, February 2001.

[10] R. M. Dünki, E. Keller, P. F. Meier, and B. Ambühl, "Temporal patterns of human behaviour: are there signs of deterministic 1/f scaling?," *Physica A*, vol. 275, pp. 596–609, 2000.

[11] H. E. Stanley, J. L. A. N. Amaral, J. S. Andrade, S. V. Buldyrev, S. Halvin, H. A. Maske, C.-K. Peng, B. Suki, and G. Viswanathan, "Scale-invariant correlations in the biological and soial sciences," *Philosophical Magazine B*, vol. 77, no. 5, pp. 1373–1388, 1998.

[12] M. E. Gomez and V. Santonja, "Characterizing temporal locality in I/O workload," in *Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '02)*, (San Diego, CA), 2002.

[13] N. Mi, Q. Zhang, A. Riska, E. Smirni, and E. Riedel, "Performance impacts of autocorrelated flows in multi-tiered systems," *Perform. Eval.*, vol. 64, no. 9-12, pp. 1082–1101, 2007.

[14] A. Riska and E. Riedel, "Disk drive level workload characterization," in *ATEC '06: Proceedings of the annual conference on USENIX '06 Annual Technical Conference*, (Berkeley, CA, USA), pp. 97–103, USENIX Association, 2006.

[15] A. Riska and E. Riedel, "Long-range dependence at the disk drive level," in *QEST '06: Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems*, (Washington, DC, USA), pp. 41–50, IEEE Computer Society, 2006.

[16] S. Chen and D. Towsley, "A performance evaluation of RAID architectures," *IEEE Transactions on Computers*, vol. 45, no. 10, pp. 1116–1130, 1996.

[17] P. G. Harrison and S. Zertal, "Queueing models of RAID systems with maxima of waiting times," *Performance Evaluation*, vol. 64, pp. 664–689, August 2007.

[18] A. S. Lebrecht, N. J. Dingle, and W. J. Knottenbelt, "Modelling and validation of response times in zoned RAID," in *16th IEEE International Symposium on Modeling, Analysis, and Simulationof Computer and Telecommunication Systems (MASCOTS '08)*, September 2008.

[19] E. Varki, "Response time analysis of parallel computer and storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, pp. 1146–1161, November 2001.

[20] E. Varki, A. Merchant, J. Xu, and X. Qiu, "Issues and challenges in the performance analysis of real disk arrays," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 559–574, June 2004.

[21] N. Wiener, "Generalized harmonic analysis," *Acta Mathematica*, vol. 55, p. 117, 1930.

[22] A. Khintchine, "Korrelationtheorie der stationären Prozesse," *Mathematische Annalen*, vol. 109, p. 604, 1934.

[23] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.

[24] R. G. Clegg, "A practical guide to measuring the Hurst parameter," *ArXiv Mathematics e-prints*, Oct. 2006.

[25] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, p. 323, 2005.

[26] "DoC database." `http://www.doc.ic.ac.uk/~uh/disk-activity/`.

[27] "Storage Networking Industry Association (SNIA) data sets." `http://iotta.snia.org/`.

[28] A. B. Downey, "The structural cause of file size distributions," *SIGMETRICS/Performance*, pp. 328–329, 2001.

[29] P. Bak, *How Nature works: The Science of Self-Organized Criticality*. New York: Copernicus, 1996.

[30] A. Field, "JINQS: An extensible library for simulating multiclass queueing networks, v1.0 user guide," August 2006. `http://www.doc.ic.ac.uk/~ajf/Research/manual.pdf`.