Performance Ponderings

1 Guerrilla Techniques for Robust Performance Engineering (2025)

Data alone is not information. Guerrilla techniques emphasizes the viewpoint that performance models are a vital complement to performance measurements, e.g.,application monitoring, in order to fully understand what information those data aretrying to convey, e.g., long-term scalability.

The Guerrilla approach also provides validation that the measurements are correct, as well as defining a quantification framework to evaluate ROI, e.g., reducing cloud chargeback. Examples of applying this Guerrilla approach to modern computer systems, such as, cloud-based applications and GenAI pre-training LLMs, are presented.

Published in ICPE '25 Companion, of the 16th ACM/SPEC International Conference on Performance Engineering, p. 87–88.

2 P = FS: Parallel is Just Fast Serial (2020)

We prove that parallel processing with homogeneous processors is logically equivalent to fast serial processing. The reverse proposition can also be used to identify obscure opportunities for applying parallelism. To our knowledge, this theorem has not been previously reported in the queueing theory literature. A plausible explanation is offered for why this might be. The basic homogeneous theorem is also extended to optimizing the latency of heterogenous parallel arrays.

Available as an arXiv e-print.

3 Erlang Redux: An Ansatz Method for Solving the M/M/m Queue (2020)

Presents a novel approach to solving the response time (i.e., residence time) of an M/M/m queue. The key idea is the introduction of an ansatz transformation, defined in terms of the Erlang B function, that avoids the more opaque derivation based on applied probability theory. This approach supersedes our earlier

https://perfdynamics.blogspot.com/2017/03/morphing-mmm-new-view-of-old-queue.html approximate morphing transformation

Available as an arXiv e-print.

4 How to Optimize Page Load Times: A Different Slant on the Frustration Index (2019)

My response to a preceding article entitled, by Tim Vereecke, posted on Day 17 of the 2019 Web Performance Calendar.

Web Performance Calendar Day 31, 2019.

5 Tomcat-Applikationsperformance in der Amazon-Cloud unter Linux modelliert (2019)

Die Cloud hat mit ihrem Bezahlmodell nach Verbrauch das gute alte Performancemanagement zurück in den Alltag der Sysadmins gebracht. Dieser Artikel erläutert, wie man mit Hilfe von Modellen das kostengünstigste Deployment in einer Amazon-Cloud findet. Neil Gunther und Mohit Chawla

1. Linux-Magazin, February 2, 2019 (in German)
2. English preprint (PDF)

6 Time: The Zeroth Performance Metric (2018)

Ruminations of the vital importance of timestamps and why they are often missing in web publications.

Web Performance Calendar Day 1, 2018

7 Exposing the Cost of Performance Hidden in the Cloud (2018)

Joint with M. Chawla. CMG cloudXchange Online Conference.

Abstract: Whilst offering lift-and-shift migration and versatile elastic capacity, the cloud also reintroduces an old mainframe concept — **chargeback** — which thereby rejuvenates the need for traditional performance and capacity management in the new cloud context. Combining production JMX data with an appropriate performance model, we show how to assess fee-based **Amazon AWS** configurations for a mobile-user application running on a Linux-hosted Tomcat cluster. The performance model also facilitates ongoing cost-benefit analysis of various **EC2 Auto Scaling** policies.

1. Video presentation
2. Enhanced slides

8 WTF is Modeling, Anyway? (2017)

A video conversation with capacity management veteran Boris Zibitsker on the importance of modeling techniques. The example discussed shows how multiple millions of dollars were saved with a **one-line** performance model (video @ 21:50 minutes) that was accurate to within 5% error (which is better than a typical queueing model).

9 Morphing M/M/m: A New View of an Old Queue (2017)

Presentation at IFORS 21st Conference of the International Federation of Operations Research Societies, July 17–21 in Quebec City, Canada.

Abstract: This year is the centenary of A. K. Erlang's on the determination of waiting times in an M/D/m queue with m telephone lines. Today, M/M/m queues are used to model such systems as, call centers, multicore computers and the Internet. Unfortunately, those who should be using M/M/m models often do not have sufficient background in applied probability theory. Our remedy defines a *morphing approximation* to the exact M/M/m queue that is accurate to within 10% for typical applications. The morphing formula for the residence-time, $R(m, \rho)$, is both simpler and more intuitive

than the exact solution involving the Erlang-C function. We have also developed an . An outstanding challenge, however, has been to elucidate the nature of the corrections that transform the approximate morphing solutions into the exact Erlang solutions. In this presentation, we show:

- The morphing solutions correspond to the m-roots of unity in the complex z-plane.
- The exact solutions can be expressed as a rational function, $R(m, z)$.
- The poles of $R(m, z)$ lie inside the unit disk, $|z| < 1$, and converge around the Szegó curve as m is increased.
- The correction factor for the morphing model is defined by the deflated polynomial belonging to $R(m, z)$.
- The pattern of poles in the z-plane provides a convenient visualization of how the morphing solutions differ from the exact solutions.

Since there were no Proceedings for this conference, no paper was required.

10 How to Emulate Web Traffic Using Standard Load Testing Tools (2016)

Joint with J. Brady. Proceedings of CMG 2016, La Jolla, California.

Abstract: Conventional load-testing tools are based on a fifty-year old time-share computer paradigm where a finite number of users submit requests and respond in a synchronized fashion. Conversely, modern web traffic is essentially asynchronous and driven by an unknown number of users. This difference presents a conundrum for testing the performance of modern web applications. Even when the difference is recognized, performance engineers often introduce modifications to their test scripts based on folklore or hearsay published in various Internet fora, much of which can lead to wrong results. We present a coherent methodology, based on two fundamental principles, for emulating web traffic using a standard load-test environment.

Available as an arXiv e-print.

11 Hadoop Superlinear Scalability (2015)

Joint with P. Puglia and K. Tomasette

Superlinearity (i.e., speedup $> 100\%$) is a bona fide observable phenomenon that has been discussed qualitatively, but not understood in a rigorous way. We provide that insight.

Superlinear speedup can be expected to be observed more often as new applications move to distributed systems. Therefore, we believe it's important to understand what superlinear data actually represent and how to address the phenomenon when configuring distributed systems for performance and scalability.

Some of the points covered include:

- Like 'perpetuum mobile,' superlinear performance appears to come for free.
- But our data shows there is a hidden cost that causes degraded scalability.
- The cost arises from latent subsystem constraints.
- Gotchas that Hadoop developers, software engineers, and sysadmins are likely to run into.

Published as Communications of the ACM, Vol. 58 No. 4, Pages 46–55. Video presentation provides more background. An **unabridged** edition of this article is freely available online at ACM Queue, Vol. 13, issue 5, June 4, 2015 and includes application of the USL to Varnish, Memcached, and Zookeeper in an Appendix.

12 A Note on Disk Drag Dynamics (2012)

Abstract: The electrical power consumed by typical magnetic hard disk drives (HDD) not only increases linearly with the number of spindles but, more significantly, it increases as very fast power-laws of speed (RPM) and diameter. Since the theoretical basis for this relationship is neither well-known nor readily accessible in the literature, we show how these exponents arise from aerodynamic disk drag and discuss their import for green storage capacity planning.

Available on arXiv.

13 A Methodology for Optimizing Multithreaded System Scalability on Multicores (2011)

Abstract: We show how to quantify scalability with the Universal Scalability Law (USL) by applying it to performance measurements of memcached, J2EE, and Weblogic on multi-core platforms. Since commercial multicores are essentially black-boxes, the accessible performance gains are primarily available at the application level. We also demonstrate how our methodology can identify the most significant performance tuning opportunities to optimize application scalability, as well as providing an easy means for exploring other aspects of the multi-core system design space.

Available on arXiv.

14 A Note on Parallel Algorithmic Speedup Bounds (2011)

Abstract: A parallel program can be represented as a directed acyclic graph. An important performance bound is the time to execute the critical path through the graph. We show how this performance metric is related to Amdahl speedup and the degree of average parallelism. These bounds formally exclude superlinear performance.

Available on arXiv.

15 The Apdex Index Revealed: Triangles Tell The Tale (2009)

Abstract: The purpose of this talk is not to improve upon or redesign the Apdex Index in any way. Rather, it is to explain the significance of the underlying construction of the Apdex Index, as it is currently defined. Then, I want to consider other ways the Apdex metrics might be applied; especially from a performance visualization standpoint.

Available as PDF slides.

16 Mind Your Knees and Queues. Responding to Hyperbole with Hyperbolæ(2009)

Abstract: How do you determine where the response-time "knee" occurs? Calculating where the response time suddenly begins to climb dramatically is considered by many to be an important determinant for such things as load testing, scalability analysis, and setting application service targets. This question arose in a CMG MeasureIT article. I responded to it in an unconventional, but rigorous way, in this CMG MeasureIT article.

17 A General Theory of Computational Scalability Based on Rational Functions (2008)

Abstract: The universal scalability law of computational capacity is a rational function Cp = P(p)/Q(p) with P(p) a linear polynomial and Q(p) a second-degree polynomial in the number of physical processors p, that has been long used for statistical modeling and prediction of computer system performance. We prove that Cp is equivalent to the synchronous throughput bound for a machine-repairman with state-dependent service rate. Simpler rational functions, such as Amdahl's law and Gustafson speedup, are corollaries of this queue-theoretic bound. Cp is further shown to be both necessary and sufficient for modeling all practical characteristics of computational scalability.

Available on arXiv.

18 Getting in the Zone for Successful Scalability (2008)

Joint with J. Holtman. Accepted for CMG 2008, Las Vegas, Nevada.

Abstract: The (USL) is an analytic function used to quantify application scaling. It is universal because it subsumes Amdahl's law (AL) and linear scaling (LS) as special cases. Using simulation, we show (1) that USL is equivalent to synchronous queueing in a load-dependent machine repairman model, and (2) how LS, AL and USL can be regarded as boundaries defining three performance zones. Typical throughput measurements lie in all three zones. Simulation scenarios provide insight into which application features should be tuned to get into the optimal performance zone.

Available on arXiv.

19 Object, Measure Thyself: Performance Monitoring and Data Collection (2008)

Matthew O'Keefe, Michael Ducy, Greg Opaczewski, and Stephen Mullins, (Orbitz Worldwide). Presented at CMG 2008, Las Vegas, Nevada.

These guys were involved with the development and deployment of ERMA (Extremely Reusable Monitoring API), and : the popular open source time-series plotting tool. (See Section 3.2 of the paper) Although I didn't write any code, I did help them write up the instrumentation work done at Orbitz Worldwide in this paper and have them present it at the . Oh! And I came up with the title, which harks back to a previous attempt of mine at a to encourage IDE vendors to do something similar in their C^{++} and Java libraries but, due a lack of obvious ROI, that never went anywhere.

1. CMG 2008 paper
2. CMG 2008 slides

20 Multidimensional Visualization of Oracle Performance Using Barry007 (2008)

Joint with T. Põder. Presented at CMG 2008, Las Vegas, Nevada.

Abstract: Most generic performance tools display only system-level performance data using 2-dimensional plot or diagram, and this limits the informational detail that can be displayed. A modern relational database system like Oracle, however, can concurrently serve thousands of client processes with different workload characteristics and generic data displays inevitably hide important information. Drawing

on our previous work, this paper demonstrates the application of Barry007 (See paper 23) multidimensional visualization for analyzing Oracle end-user session-level performance showing both collective trends and individual performance anomalies.

Available on arXiv

21 Eine Chance für Linux (2008)

Appears in the May 18 volume of . (in German)

Abstract (Translation): Linux could be in a position to expand its presence in the server market by looking to mainframe computer performance management as a role model and adapting its instrumentation accordingly.

1. German version (PDF)
2. English version (PDF)

22 Better Performance Management Through Better Visualization Tools (2008)

Invited presentation given at the annual Hotsos Symposium in Dallas, Texas, March 2–6 2008.

23 Seeing It All at Once with Barry (2007)

Joint with M. F. Jauvin. Presented at CMG 2007, San Diego, California.

Abstract: Improving data visualization paradigms for performance management is an orphaned area of tool development. Tool vendors avoid investing in development if they see no demand, while capacity planners and performance analysts do not demand what they have not conceived. We attempt to cut this Gordian knot with "Barry"' — a 3D performance visualization suite based on barycentric coordinates. Potentially thousands of active processors, servers, network segments or applications can be viewed as a moving cloud of points that produces easily comprehended visual patterns due to correlations in the workload dynamics. Barry provides an optimal impedance match between the measured computer system and the cognitive computer system (your brain).

1. Paper (PDF)
2. Slides (PDF)
3. Animations (HTML)
4. Tools (HTML)

24 Linux Load Average (2007)

Linux-Magazin p. 84–91, July 2007.

Abstract: No doubt you have often seen and even made use of those three little numbers called "oad averages" that appear in shell commands like `procinfo` and `uptime`. But how well do you understand them? Why are there always **three** numbers? How are they calculated? And what is their origin? As well as answering these questions in this article, the concept of stretch factor is introduced as a way to enhance load average data for improved performance management of symmetric multiprocessor and multicore servers.

1. German version Load Average Enträtselt und Erweitert: Leistungsdiagnostik (PDF)
2. English version Understanding the load-average metrics in a Linux operating system (PDF)

25 Berechenbare Performance (Predicting Performance) (2007)

Invited paper published in the German monograph Linux Technical Review 02 - Monitoring.

Abstract: Performance management can be broken into three sequen- tial processes: performance monitoring, performance analy- sis, and performance modeling. Monitoring is the theme of this issue, analysis refers to the capability of look- ing for patterns in monitored data that reside in a database, while modeling attempts to use monitored data to predict future events, such as resource bottlenecks. PDQ (Pretty Damn Quick) is a queueing analysis tool aimed at expediting the prediction process.

1. German version (PDF)
2. English version (PDF)

26 Moore's Law: More or Less? (2007)

Published in the May issue of the CMG MeasureIT e-zine.

27 Visualizing Virtualization (2007)

Guest editorial for the March issue of the CMG e-zine called MeasureIT.

28 Guerrilla Scalability: How to Do Virtual Load Testing (2007)

Invited presentation at the Hotsos Symposium 2007, March, Dallas, Texas.

29 The Virtualization Spectrum from Hyperthreads to GRIDs (2006)

This CMG paper, presented at the 2006 conference in Reno, Nevada, is based on the following observations:

- Disparate types of virtual machines lie on a **discrete spectrum** bounded by hyperthreading at one extreme and hyperservices at the other,
- **Poll-based scheduling** is the common architectural element in most virtual machine implementations.

The associated polling frequency (from GHz to μHz) positions each virtual machine implementation into a region of the VM-spectrum. Several case studies are analyzed to illustrate how this insight can make virtual machines more visible to performance management techniques.

30 Reconstructing the Future: Capacity Planning with Data That's Gone Troppo (2006)

Joint with S. Jenkin.

Abstract: Two of our coworkers, Larry and Katrina, were recently fired for being discovered in a compromising situation at the office. In an act of revenge before leaving, they deleted several directories—some of which contained our validated forecasting data and capacity planning models. Fortunately, our company had the foresight to implement disaster recovery data centers: one in Innisfail, Far North Queensland and the other in New Orleans. We've since joined gamblers anonymous. But, this still left us with a significant problem. What do you do with capacity planning models that worked in the past but now, some of the historical data is no longer accessible? This paper presents techniques for reconstructing your capacity planning models when significant chunks of data are missing. In our case, we were even able to improve our previous forecasting with a more accurate power-law model, which was developed during the model reconstruction phase.

This paper was presented at CMG-A in Sydney, Australia.

31 Benchmarking Blunders and Things That Go Bump in the Night (2006)

Published as Part I and Part II in the CMG *MeasureIT* online magazine. See also paper 35.

32 Unification of Amdahl's Law, LogP and Other Performance Models for Message-Passing Architectures (2005)

This paper generalizes the theorem in paper 41 (below) and was presented at PDCS 2005 *Parallel and Distributed Computing Systems*, IASTED XVII International Conference.

33 (Numerical) Investigations into Physical Power-law Models of Internet Traffic Using the Renormalization Group (2005)

This paper was presented at the Triennial Conference of the International Federation of Operations Research Societies, Honolulu, Hawaii, July 11–15, 2005.

Applies the real-space variant of the renormalization group to exclude certain models that have appeared in the literature to account for so-called *self-similar* Internet traffic and further suggests that the claimed ramifications for Internet capacity planning may have been over-emphasized. Chapter 10 of Guerrilla Capacity Planning presents these conclusions in a less mathematical form.

34 Millennium Performance Problem 1: Performance Visualization (2005)

Published in the CMG MeasureIT online magazine.

35 Benchmarking Blunders and Things That Go Bump in the Night (2004)

Abstract: Benchmarking; by which I mean any computer system that is driven by a controlled workload, is the ultimate in performance testing and simulation. Aside from being a form of institutionalized cheating, it also offer countless opportunities for systematic mistakes in the way the workloads are applied and the resulting measurements interpreted. Right test, wrong conclusion is a ubiquitous mistake that happens because test engineers tend to treat data as divine. Such reverence is not only misplaced, it's also a sure ticket to production hell when the application finally goes live. I demonstrate how such mistakes can be avoided by means of two war stories that are real WOPRs (see what I did there?):

1. How to resolve benchmark flaws over the psychic hotline

2. How benchmarks can go flat with too much Java juice.

In each case I present simple performance models and show how they can be applied to correctly assess benchmark data.

Presented at WOPR2 Workshop hosted by Sun Microsystems in Palo Alto, California on April 15–17, 2004. Available on arXiv.

36 How to Get Unbelievable Load Test Results (2004)

Featured at TeamQuest Corporation as an online capacity planning column.

37 Performance Evaluation of Packet-to-Cell Segmentation Schemes in Input Buffered Packet Switches (2003)

Joint with K. J. Christensen, K. Yoshigoe and A. Roginsky.
Presented at *High-Speed Networks Symposium* of the IEEE International Conference on Communications (ICC 2004). Available on arXiv.

38 Unix Load Average Metric (2003-2004)

Originally published as a series of online performance columns for TeamQuest Corporation.

- Part 1: How It Works
- Part 2: Not Your Average Average
- Part 3: Addendum on Hz vs. HZ

NOTE: The hyperlinked version of the Linux kernel is release 2.6.xx . A more detailed discussion appears in **Chapter 4** of my Perl::PDQ book.

39 Guerrilla Capacity Planning Series (2003)

These two articles:

- **PART 1:** Hit-and-Run Tactics for Website Scalability
- **PART II:** Weapons of Mass Instruction

were published in the CMG *MeasureIT* online magazine.

40 Characterization of the Burst Stabilization Protocol for the RR/CICQ Switch (2003)

Joint with K. J. Christensen and K. Yoshigoe
Presented at the *IEEE Conference on Local Computer Networks*.

Abstract: Input buffered switches with Virtual Output Queueing (VOQ) can be unstable when presented with unbalanced loads. Existing scheduling algorithms, including iSLIP for Input Queued (IQ) switches and Round Robin (RR) for Combined Input and Crossbar Queued (CICQ) switches, exhibit instability for some schedulable loads. We investigate the use of a queue length threshold and bursting mechanism to achieve stability without requiring internal speed-up. An analytical model is developed to prove that the burst stabilization protocol achieves stability and to predict the minimum burst value needed as a function of offered load. The analytical model is shown to have very good agreement with simulation results. These results show the advantage of the RR/RR CICQ switch as a contender for the next generation of high-speed switches.

Available on arXiv.

41 A New Interpretation of Amdahl's Law and Geometric Scalability (2002)

Among other things, this paper presents the following theorem:

> *Amdahl's law for parallel speedup is equivalent to the synchronous queueing bound on throughput in the repairman model of a symmetric multiprocessor.*

Available on arXiv.

42 Hit-and-Run Tactics Enable Guerrilla Capacity Planning (2002)

Abstract: A leaner version of traditional capacity planning can help you hone system performance without inflating schedules or delaying time to market.

Published in IEEE IT Professional, pp. 40-46, July-August, 2002.

43 Hypernets: Good (G)news for Gnutella! (2002)

Online article responding to an earlier analysis on Gnutella written by Jordan Ritter in 2001.
Measurements of both Napster and Gnutella are also discussed in this 2003 paper.
I point out that hypernets, like a 20-degree virtual hypertorus or hypercube, are much more efficient than a Cayley tree. In fact, it looks like *BitTorrent* does something like this.
This online article was slashdotted in Feb, 2002.

44 Of Buses and Bunching: Strangeness in the Queue (2001)

TeamQuest online column.

45 Quantifying Application and Server Scalability (2001)

The following series of three articles:

- Commercial Clusters and Scalability
- How to Measure an Elephant
- Evaluating Scalability Parameters: A Fitting End

was published as a set of TeamQuest online columns.

46 How to Write Application Probes (Updated 2001)

TeamQuest performance column online.

47 Scalability Models for a Hypergrowth e-Commerce Website (2000)

Published in *Performance Engineering: State of the Art and Current Trends*, Springer Lecture Notes in Computer Science series.

48 BIRDS-I: A Benchmark for Image Retrieval on the Internet (2000)

Published as HP Labs Tech Report, HPL-2000-162.

49 Solaris Resource Manager: All I Ever Wanted Was My Unfair Advantage ... and Why You Can't Get It! (1999)

This paper is about **virtualization** implemented in SHARE II (see footnote references 1 and 2 below) sitting on top of the Solaris kernel and rebadged by Sun as their System Resource Manager (SRM). Although it was written in 1999, the by now ubiquitous forms of virtualization, whether it be **hypervisors** (e.g., VMware, XenServer, AWS), or Linux **CFS**, all use some form of **fair-share scheduler** (see footnote reference 3 below). Astoundingly, most people deploying applications, either on local hypervisors or remote cloud services, remain blissfully unaware of this fact, and its potential impact on application performance and observed capacity. See also paper 29, "The Virtualization Spectrum from Hyperthreads to GRIDs".

> Traditional UNIX time-share schedulers attempt to be fair to all users by employing a round-robin style algorithm for allocating CPU time. Unfortunately, a loophole exists whereby the scheduler can be biased in favor of a greedy user running many short CPU-bound processes. This loophole is not a defect but an intrinsic property of the round-robin scheduler that ensures responsiveness to the short CPU demands associated with multiple interactive users. A new generation of UNIX system resource management (SRM) software constrains the scheduler to be equitable to all users regardless of the number of processes each may be running. This *fair-share* scheduling draws on the concept of pro-rating resource "shares" across users and groups and then dynamically adjusting CPU usage to meet those share proportions. The simple notion of statically allocating these shares, however, belies the potential consequences for performance as measured by user response time and service level targets. We demonstrate this point by modeling several simple share allocation scenarios and analyzing the corresponding performance effects. A brief comparison of commercial SRM implementations from Hewlett-Packard, IBM, and Sun Microsystems is also presented.

Presented at the Computer Measurement Group Conference, Reno, NV, Dec. 5–10, 1999.

1. A, Bettison, A, Gollan, C. Maltby, N. Russell, "SHARE II —A User Administration and Resource Control System for UNIX," LISA V, San Diego, CA, Sep. 20-Oct. 3, 1991.
2. N. J. Gunther, "UNIX Resource Managers: Capacity Planning and Resource Issues," SAGE-AU Conference, Gold Coast, Queensland, Australia, July 3-7, 2000.
3. J. Kay, and P. Lauder, "A Fair Share Scheduler," Communications of the ACM, 31(1), pp. 44-55, 1988.

50 Microsoft Windows NT Scalability (1997–1998)

The following articles on Windows NT performance were published as a series in the USENIX login magazine:

1. NT to the Max (NoT!) — Nov (1997)
2. The ABCs of TPCs — Feb (1998)
3. How to Stack Cyberbrix — Jun (1998)

Additionally, **AUUG** (Australian Unix Users Group) apparently picked them up and published articles (1) and (2) in the February 1998 edition of their AUUGN journal, which is still online! (as a scanned PDF). Scroll down to the Table of Contents.

51 The MP Effect: Parallel Processing in Pictures (1996)

This paper presents a purely diagrammatic way of understanding computer system scalability, including a picture-based derivation of Amdahl's law.
It received the Best Paper award at CMG 1996.
These ideas are elaborated on in Chap. 14 of The Practical Performance Analyst book and Chap. 4 of the Guerrilla Capacity Planning book.

52 A Simple Capacity Model of Massively Parallel Transaction Systems (1993)

This is the original paper that presented the Universal Scalability Law (or USL) model under the name: "Super Serial" model, because I viewed it as an extension of the *serial fraction* concept contained in Amdahl's law. This version of the USL model was developed for assessing database scalability while I was at Pyramid Technology.
The paper was given at CMG 1993 in San Diego, California, as well as Jim Gray's 4th HPTS (High Performance Transaction Systems) Workshop in Asilomar, California, 1993.
Later, I realized more generally that contention and coherency could act as independent effects. See Paper 17 and Chaps. 4-6 of Guerrilla Capacity Planning for a more recent technical overview of the USL.

53 Performance Collapse of Networks (1988)

1. Background information. "Performance Collapse in Packet Networks and Computer Systems — small Numbers, BIG Consequences."
2. Information Processing Letters, 1989. "Path Integral Methods for Computer Performance Analysis." (5.1 MB scanned PDF).

   > **Abstract:** The Feynman path integral is presented for analyzing computer performance models that exhibit critical behavior due to the presence of unstable states. We examine previously ignored deterministic "jump" solutions associated with instability transients. Using these solutions, an explicit expression for the lifetime of an induced metastable state is derived for a queueing model of virtual memory.