

Neil J. Gunther

Guerrilla Capacity Planning

A Tactical Approach to Planning for Highly
Scalable Applications and Services

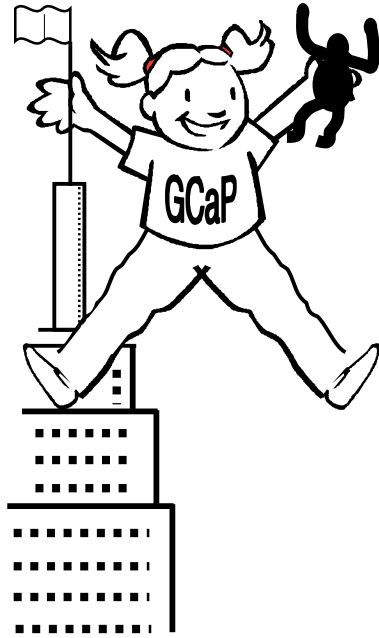
October 14, 2006

Springer

Berlin Heidelberg New York

Hong Kong London

Milan Paris Tokyo



Go forth and Kong-ka!

Preface

This book is based largely on the material used in the professional training course of the same name. Currently, the *Guerrilla Capacity Planning* (GCaP) classes are usually conducted every calendar quarter by Performance Dynamics Educational Services (www.perfdynamics.com) in Pleasanton, California. The same course has also been taught privately at such international organizations as: Amdahl Corporation, AT&T Wireless, Boeing Companies, Federal Express, Peter Harding and Associates (Australia), Sun Microsystems (USA and France), System Administrators Guild of Australia, and Thales Naval (Holland).

Some of the material originates from 1997, when I began teaching a similar class under the title *Practical Performance Methods*, at the Stanford University *Western Institute for Computer Science* summer extension program. My class replaced a similar one that had been taught jointly for many years by Ed Lazowska, Ken Sevcik, and John Zahorjan. Their course and the accompanying book (Lazowska et al. 1984) (now out of print) has provided much inspiration for all my books. Sadly, while writing this book, I learned that Ken Sevcik had passed away.

A major motivation for writing this book was to provide GCaP graduates with a more detailed version of the lecture notes for their later review and reference. Another motivation is to make the same material available to a wider audience who, although they may not be able to attend the GCaP classes, could nonetheless benefit from applying GCaP methods. In these days of ever-shortening planning horizons and contracting times to market, traditional approaches to capacity planning are often seen as inflationary for production schedules. Rather than giving up in the face of this kind of relentless economic pressure to get things done more quickly, GCaP tries to facilitate it through opportunistic and rapid forecasting of capacity requirements. A more detailed list of the advantages of GCaP and guidelines for applying it can be found in the *Guerrilla Manual* located in Appendix F.

Book Structure

This book is not broken into conventional sections, rather there are several themes running concurrently throughout all the chapters. These themes can be grouped as follows:

Guerrilla Tactics

This is the dominant theme that provides the rationale for the title of the book. Put bluntly, the planning horizon has now been reduced to about three months (i.e., a fiscal quarter), thanks to the influence of Wall Street on project cycle times, and only GCaP-style *tactical planning* is crazy enough to be compatible with that kind of insanity.

The Guerrilla theme explains and motivates the GCaP approach to capacity management by identifying opportunistic circumstances in team meetings where capacity planning issues can be brought up, whether they were part of the meeting agenda or not; key concepts, such as the performance homunculus and the universal law of computational scalability; and the use of lightweight tools, such as spreadsheets and operational formulae. The core of this theme is introduced in Chap. 1.

Chapter 2 continues the introduction to the GCaP theme by demonstrating how it is also a natural fit with other standardized IT management frameworks such as ITIL (Information Technology Infrastructure Library). Chapter 11 presents an entirely independent report written by contributing author James Yable, who took the GCaP theme and tailored it to meet the immediate capacity management needs of his data center.

Guerrilla Scalability

Another major theme is the concept of *scalability* assessment in the context of capacity management, and in particular, the scalability of *application software*. Many people use the word “scalability” without defining it clearly. Most people cannot define it *quantitatively*, and if you cannot quantify it, you cannot guarantee it!

Chapters 4, 5, and 6 address that need by presenting *the universal law of computational scaling*. The notion of ideal parallelism, as it relates to hardware scalability, is used as a springboard to go beyond such well-known models as Amdahl’s law and multiuser concurrency, to arrive at the universal law of scalability for hardware. A queue-theoretic argument, based on Theorem 6.2 in Chap. 6, which states:

Amdahl’s law for parallel speedup is equivalent to the synchronous throughput bound of the repairman queueing model of a multiprocessor

is invoked to extend the universal law to the prediction of application software scalability. Many examples, based on difficult to obtain hardware and

software scalability measurements, are discussed in these three core chapters. The universal scaling law is also applied, in GCaP style, to the analysis of capacity planning data in later chapters. Although many of these ideas have been developed and applied by the author since about 1991, this book represents the first time they have been brought together and demonstrated in one place.

An important advantage of this universal scaling law is that it provides the underpinnings for a *virtual load testing* environment. It allows the reader to take a sparse set of load measurements (e.g., 4–6 data points) and determine how an application will scale under larger user loads than can be generated in the physical test lab. Moreover, and in keeping with the GCaP theme, much of this scalability analysis can be done in a spreadsheet like Excel.

Guerrilla Victories

The remaining chapters comprise detailed examples of successful applications of the other two themes. Chapter 8 presents the author’s success in applying GCaP to a large-scale Web site capacity planning in the Silicon Valley. In particular, it demonstrates how some more traditional capacity planning techniques that originated on mainframe computer systems can be adapted to modern servers. Chapter 9 presents GCaP techniques for planning the capacity of gargantuan computing environments such as peer-to-peer systems.

Chapter 10 provides an overview of the peculiar impact of certain Internet packet behavior on buffer sizing for routers and servers. The reason this is potentially very important for capacity planning is due the veritable “paper mill” of academic papers written on so-called *self-similar* Internet traffic. This self-similarity refers to long-term clustering of Internet packets first observed in the famous Bellcore measurements circa 1990. Many of these papers, however, are mathematically very sophisticated and impenetrable to the typical network capacity planner. This chapter attempts to provide a simpler mathematical treatment than is generally available, but without any loss in accuracy. The chapter concludes with some recent measurements and analysis done in the U.K. that indicates the severity of these self-similar packet-clustering effects may have been overplayed.

Intended Audience

Each of the three themes just described can be used to advantage by a broad diversity of IT professionals. At the executive level there are chief information officers (CIOs), chief technology officers (CTOs), and vice presidents (VPs). Mid-level management that could benefit from understanding and championing GCaP concepts include directors, senior management, and project managers. GCaP methodologies are useful for mainframe capacity planners in the

process of broadening their skills, performance engineers, and software engineers, system architects, software developers, system analysts and system administrators.

One suggested grouping of themes with professional expertise is shown in the following table.

Theme	Audience	Chapters
Guerrilla tactics	CIOs, CTOs, directors, VPs, senior managers, project managers	Chaps. 1–2
Guerrilla scalability	Mainframe capacity planners, performance and software engineers, QA and test engineers, system architects, software developers, system administrators and analysts	Chaps. 4–6
Guerrilla victories	CTOs, project managers, mainframe capacity planners, performance and Software engineers, software developers, software and performance engineers, system administrators and analysts	Chaps. 1–11, Chaps. 4–10

Acknowledgments

This book has benefited from the insight and assistance of several people, and they deserve my explicit thanks for their contributions.

Steve Jenkin inspired me to put together *The Guerrilla Manual* in Appendix F by pointing out that employees in the trenches often find themselves in the position where being able to point to an authoritative list of methods and aphorisms can make the difference between getting their point across or not. He also suggested the organization of workload types in Table 6.9 based on the range of values for the contention (σ) and coherency (κ) parameters of the universal scalability law for software in Chap. 6.

Ken Christensen performed the event-based simulations that provided empirical support for Theorem 6.2. He also corroborated the findings of Field et al. (2004) regarding self-similar packetization using his own IEEE-validated Ethernet simulation model.

Greg Dawe, Jamie Rybicki, and Andrew Sliwkowski at RSA Security performed the painstaking application measurements which enabled me to develop the PDQ performance models used in Chap. 7. In typical eclectic fashion, Andrew Sliwkowski also drew my attention to the software variant of Amdahl’s law, which provided the bridge to open Chap. 6.

Finally, it is my pleasure to thank Giordano Beretta, Ken Christensen, Mark Friedman, Kathy Hagedon, Jim Holtman, J. Scott Johnson, Scott John-

son, Robert Lane, Pedro Vazquez and Lloyd Williams for providing feedback on early drafts of various chapters, and otherwise improving the overall content of this book. Any remaining shortcomings are mine alone.

Warranty Disclaimer

No warranties are made, express or implied, that the information in this book and the associated computer programs are error free, or are consistent with any particular standard of merchantability, or that they will meet your requirements for any particular application. They should not be relied upon for solving a problem the incorrect solution of which could result in injury to a person or loss of property. The author disclaims all liability for direct or consequential damages resulting from the use of this book.

In Sect. 5.6.2 some precision problems with the values computed by Excel are noted. A more careful analysis is provided in Appendix B. Because of its potential precision limitations, as noted by Microsoft (support.microsoft.com/kb/78113/), you are advised to validate any numerical predictions made by Excel against those calculated by other high-precision tools, such as *Mathematica*, R, S-PLUS or Minitab.

Palomares Hills, California
October 14, 2006

N.J.G.

Contents

Preface	VII
1 What Is Guerrilla Capacity Planning?	1
1.1 Introduction	1
1.2 Why Management Resists Capacity Planning	1
1.2.1 Risk Management vs. Risk Perception	2
1.2.2 Instrumentation Just Causes Bugs	3
1.2.3 As Long as It Fails on Time	4
1.2.4 Capacity Management as a Homunculus	5
1.3 Guerrilla vs. Gorilla	6
1.3.1 No Compass Required	7
1.3.2 Modeling Is Not Like a Model Railway	8
1.3.3 More Like a Map Than the Metro	8
1.4 Tactical Planning as a Weapon	9
1.4.1 Scalability by Spreadsheet	10
1.4.2 A Lot From Little	11
1.4.3 Forecasting on the Fly	13
1.4.4 Guerrilla Guidelines	14
1.5 Summary	16
2 ITIL for Guerrillas	17
2.1 Introduction	17
2.2 ITIL Background	17
2.2.1 Business Perspective	19
2.2.2 Capacity Management	21
2.3 The Wheel of Capacity Management	21
2.3.1 Traditional Capacity Planning	21
2.3.2 Running on the Rim	23
2.3.3 Guerrilla Racing Wheel	24
2.4 Summary	25

3	Damaging Digits in Capacity Calculations	27
3.1	Introduction	27
3.2	Significant Digits	28
3.2.1	Accuracy	28
3.2.2	Precision	29
3.3	Sifting for SigDigs	30
3.3.1	Count by Zeros	30
3.3.2	Significance and Scale	32
3.4	Rounding Rules	32
3.4.1	Golden Rule	34
3.4.2	Sum Rule	34
3.4.3	Product Rule	34
3.5	Planning With Dollars and Sense	35
3.5.1	Cost Metric	35
3.5.2	Significant Digits	36
3.6	Expressing Errors	37
3.6.1	Absolute Error	37
3.6.2	Relative Error	37
3.6.3	Standard Deviation	37
3.6.4	Standard Error	38
3.6.5	Error Bars	38
3.6.6	Instrumentation Error	39
3.7	Interval Arithmetic	39
3.8	Summary	40
4	Scalability—A Quantitative Approach	41
4.1	Introduction	41
4.2	Fundamental Concepts of Scaling	41
4.2.1	Geometric Scaling	42
4.2.2	Allometric Scaling	43
4.2.3	Critical Size	44
4.2.4	Sizing Examples	45
4.3	Hardware Scalability	47
4.3.1	Ideal Parallelism	48
4.3.2	Amdahl's Law	49
4.3.3	Multiuser Scaleup	52
4.3.4	Serial-Parallel Duality	55
4.3.5	Scaled Speedup	56
4.4	Universal Scalability Model	56
4.4.1	The Role of Coherency	58
4.5	Other Scalability Models	63
4.5.1	Geometric Model	63
4.5.2	Quadratic Model	63
4.5.3	Exponential Model	64
4.6	Multicores and Clusters	66

4.7	Summary	68
5	Evaluating Scalability Parameters	71
5.1	Introduction	71
5.2	Benchmark Measurements	72
5.2.1	The Workload	72
5.2.2	The Platform	74
5.2.3	The Procedure	75
5.3	Minimal Dataset	75
5.3.1	Interpolating Polynomial	76
5.3.2	Regression Polynomial	76
5.4	Capacity Ratios	77
5.5	Transforming the Scalability Equation	77
5.5.1	Efficiency	78
5.5.2	Deviation From Linearity	78
5.5.3	Transformation of Variables	79
5.5.4	Properties of the Regression Curve	80
5.6	Regression Analysis	82
5.6.1	Quadratic Polynomial	82
5.6.2	Parameter Mapping	83
5.6.3	Interpreting the Scalability Parameters	85
5.6.4	Error Reporting	86
5.7	Less Than a Full Deck	87
5.7.1	Sparse Even Data	88
5.7.2	Sparse Uneven Data	90
5.7.3	Missing $X(1)$ Datum	91
5.8	Summary	94
6	Software Scalability	97
6.1	Introduction	97
6.2	Amdahl's Law for Software	98
6.3	Universal Software Scalability	100
6.4	Concurrent Programming and Coherency	102
6.5	UNIX Multitasking Application	103
6.5.1	The Workload	103
6.5.2	The Platform	104
6.5.3	Regression Analysis	104
6.6	Windows-Based Applications	107
6.6.1	The Workload	107
6.6.2	The Platform	108
6.6.3	Regression Analysis	109
6.7	Multitier Architectures	110
6.7.1	The Workload	111
6.7.2	The Platform	111
6.7.3	Regression Analysis	112

6.7.4	Why It Works	114
6.8	Classification by Workload.....	115
6.9	Summary	116
7	Fundamentals of Virtualization	117
7.1	Introduction	117
7.2	The Spectrum of Virtual Machines	118
7.2.1	VM Spectroscopy	118
7.2.2	Polling Rates and Frequency Scales	119
7.3	Microlevel Virtual Machines: Hyperthreading	119
7.3.1	Micro-VM Polling	122
7.3.2	Thread Execution Analysis	123
7.3.3	Missing MIPS Explained	124
7.3.4	Windows 2000 Production Server.....	126
7.3.5	Guerrilla Capacity Planning	127
7.4	Mesolevel Virtual Machines: Hypervisors	127
7.4.1	Fair-Share Scheduling	129
7.4.2	Meso-VM Polling.....	132
7.4.3	VMWare Share Allocation Analysis	134
7.4.4	J2EE WebLogic Production Application.....	135
7.4.5	VMWare Scalability Analysis	137
7.4.6	Guerrilla Capacity Planning	138
7.5	Macrolevel Virtual Machines: Hypernets.....	138
7.5.1	Macro-VM Polling.....	139
7.5.2	Bandwidth Scalability Analysis	140
7.5.3	Remote Polling Rates	141
7.5.4	Guerrilla Capacity Planning	142
7.6	Summary	142
8	Web Site Planning	143
8.1	Introduction	143
8.2	Analysis of Daily Traffic	144
8.2.1	The Camel and the Dromedary	144
8.2.2	Unimodal but Bicoastal	146
8.3	Effective Demand	148
8.3.1	Modeling Assumptions	149
8.3.2	Statistical Approach	149
8.4	Selecting Statistical Tools	150
8.4.1	Spreadsheet Programming	150
8.4.2	Online Support	150
8.5	Planning for Data Collection	151
8.5.1	Commercial Collectors: Use It or Lose It	151
8.5.2	Brewing in the Background	151
8.6	Short-Term Capacity Planning	152
8.6.1	Multivariate Regression of Daily Data.....	152

8.6.2	Automation Using Spreadsheet Macros	153
8.7	Long-Term Capacity Planning	155
8.7.1	Nonlinear Regression of Weekly Data	155
8.7.2	Procurement Curves	156
8.7.3	Estimating Server Scalability	157
8.7.4	Calculating Capacity Gains	158
8.7.5	Estimating the Doubling Period	161
8.8	Summary	162
9	Gargantuan Computing—GRIDs and P2P	165
9.1	Introduction	165
9.2	GRIDs vs. P2P	166
9.3	Analysis of Gnutella	167
9.4	Tree Topologies	168
9.4.1	Binary Tree	169
9.4.2	Rooted Tree	169
9.4.3	Cayley Tree	169
9.5	Hypernet Topologies	169
9.5.1	Hypercube	170
9.5.2	Hypertorus	170
9.6	Capacity Metrics	170
9.6.1	Network Diameter	170
9.6.2	Total Nodes	171
9.6.3	Path Length	171
9.6.4	Internal Path Length	171
9.6.5	Average Hop Distance	171
9.6.6	Network Links	172
9.6.7	Network Demand	172
9.6.8	Peer Demand	172
9.6.9	Bandwidth	173
9.7	Relative Bandwidth	173
9.7.1	Cayley Trees	173
9.7.2	Trees and Cubes	174
9.7.3	Cubes and Tori	175
9.7.4	Ranked Performance	176
9.8	Summary	176
10	Internet Planning	179
10.1	Introduction	179
10.2	Bellcore Traces	180
10.3	Fractals and Self-Similarity	182
10.4	Fractals in Time	186
10.4.1	Short-Range Dependence	186
10.4.2	Long-Range Dependence	188
10.5	Impact on Buffer Sizing	190

XVIII Contents

10.5.1	Conventional Buffer Sizing	190
10.5.2	LRD Buffer Sizing	192
10.6	New Developments	193
10.6.1	Ethernet Packetization	194
10.6.2	LRD and Flicker Noise	196
10.7	Summary	197
11	Going Guerrilla—A Case Study	199
11.1	Introduction	199
11.2	Guerrilla Monitoring Phase	199
11.3	The Basic Solution	201
11.3.1	Implementation Details	202
11.3.2	Orca Output Examples	203
11.3.3	Round-Robin Database	203
11.4	Extending the Basic Solution	206
11.4.1	Mainframe Data Processing	206
11.4.2	Guerrilla Planning Phase	207
11.4.3	Monitoring With ORCAalerts	208
11.5	Future Developments	209
11.6	Summary	210

Appendix

A	Amdahl and the Repairman	213
A.1	Repairman Queueing Model	213
A.2	Amdahl's Law for Parallel Subtasks	214
A.2.1	Single Task	215
A.2.2	Two Subtasks	215
A.2.3	Multiple Subtasks	215
A.3	Amdahl's Law for Concurrent Multitasks	217
A.4	Note On Nelson's Approach	217
B	Mathematica Evaluation of NUMA Parameters	219
B.1	Mathematica Packages	219
B.2	Import the Data	219
B.3	Tabulate the Data	220
B.4	Plot Normalized Data	220
B.5	Nonlinear Regression	221
B.6	ANOVA Report	221
B.7	Maximal CPU Configuration	222
B.8	Plot of Regression Model	222

C Abbreviations and Units 223

 C.1 SI Prefixes 223

 C.2 Time Suffixes 223

 C.3 Capacity Suffixes 224

D Programs for Chapter 3 225

 D.1 Determine SigDigs in VBA 225

 D.2 Determine SigDigs in Mathematica 226

 D.3 Determine SigDigs in Perl 227

E Programs for Chapter 8 229

 E.1 Example Data Extractor in Perl 229

 E.2 VBA Macro for Calculating U_{eff} 231

F The Guerrilla Manual 235

 F.1 Weapons of Mass Instruction 235

 F.2 Capacity Modeling Rules of Thumb 238

 F.3 Scalability on a Stick 240

 F.3.1 Universal Law of Computational Scaling 240

 F.3.2 Areas of Applicability 241

 F.3.3 How to Use It 241

Bibliography 243

Index 249

