

# Performance Modeling and Design of Computer Systems

Mor Harchol-Balter

Please direct errors and suggestions to [harchol@cs.cmu.edu](mailto:harchol@cs.cmu.edu)

© 2011 by Mor Harchol-Balter.

**Do not distribute without permission from the  
copyright holder.**



*I have always been interested in finding better designs for computer systems, designs that improve performance without the purchase of additional resources. When I look back at the problems that I have solved, and I look ahead to the problems I hope to solve, I realize that the problem formulations keep getting simpler and simpler, and my footing less secure. Every wisdom that I once believed, I have now come to question: If a scheduling policy helps one set of jobs, does it necessarily hurt some other jobs, or are these “conservation laws” being misinterpreted? Do greedy routing strategies make sense in server farms, or is what’s good for the individual actually disastrous for the system as a whole? When comparing a single fast machine with  $n$  slow machines, each of  $1/n^{\text{th}}$  the speed, the single fast machine is typically much more expensive; but does that mean that it is necessarily better? Should distributed systems really aim to balance load, or is this a convenient myth? Cycle stealing, where machines can help each other when they are idle, sounds like a great idea, but can we quantify the actual benefit? How much is the performance of scheduling policies affected by variability in the arrival rate and service rate and by fluctuations in the load, and what can we do to combat variability? Inherent in these questions is the impact of real user behaviors and real-world workloads with heavy-tailed, highly variable service demands, as well as correlated arrival processes. Also intertwined in my work are the tensions between theoretical analysis and the realities of implementation, each motivating the other. In my search to discover new research techniques which allow me to answer these and other questions, I find that I am converging towards the fundamental core that defines all these problems, and that makes the counter-intuitive more believable.*

# Contents

<b>I</b>	<b>Introduction to Queueing</b>	<b>1</b>
<b>1</b>	<b>Motivating Examples</b>	<b>5</b>
1.1	What is queueing theory? . . . . .	5
1.2	Examples of the power of queueing theory . . . . .	7
1.3	Combining theory and systems . . . . .	16
<b>2</b>	<b>Queueing Theory Terminology</b>	<b>19</b>
2.1	Where we're heading . . . . .	19
2.2	The single server network . . . . .	20
2.3	Classification of queueing networks . . . . .	23
2.4	Open networks . . . . .	23
2.5	More metrics: throughput and utilization . . . . .	25
2.6	Closed networks . . . . .	30
2.6.1	Interactive (terminal-driven) system . . . . .	30
2.6.2	Batch systems . . . . .	33
2.6.3	Throughput in a closed system . . . . .	34
2.7	Differences between closed and open networks . . . . .	35
2.7.1	A question on modeling . . . . .	36
2.8	Related readings . . . . .	36
2.9	Exercises . . . . .	37
<b>II</b>	<b>Necessary Probability Background</b>	<b>39</b>
<b>3</b>	<b>Probability Review</b>	<b>43</b>
3.1	Sample space and events . . . . .	43
3.2	Probability defined on events . . . . .	44
3.3	Conditional probabilities and events . . . . .	46
3.4	Law of total probability . . . . .	48
3.5	Discrete versus continuous random variables . . . . .	50
3.6	Probabilities and densities . . . . .	52
3.6.1	Discrete: probability mass function . . . . .	52

3.6.2	Continuous: probability density function . . . . .	55
3.7	Expectation & variance . . . . .	59
3.8	Normal distribution . . . . .	64
3.8.1	Linear transformation property . . . . .	66
3.8.2	Central limit theorem . . . . .	69
3.9	Joint probabilities and independence . . . . .	72
3.10	Conditional probabilities and expectations . . . . .	74
3.11	Probabilities and expectations via conditioning . . . . .	78
3.12	Linearity of expectation . . . . .	81
3.13	Sum of a random number of random variables . . . . .	84
3.14	Exercises . . . . .	86
<b>4</b>	<b>Generating Random Variables</b>	<b>93</b>
4.1	Inverse transform method . . . . .	93
4.1.1	The continuous case . . . . .	94
4.1.2	The discrete case . . . . .	95
4.2	Accept/reject method . . . . .	97
4.2.1	Discrete case . . . . .	97
4.2.2	Continuous case . . . . .	100
4.2.3	Some hard problems . . . . .	104
4.3	Readings . . . . .	104
4.4	Exercises . . . . .	105
<b>5</b>	<b>Sample Paths, Convergence, and Averages</b>	<b>107</b>
5.1	Convergence . . . . .	108
5.2	Strong and weak laws of large numbers . . . . .	113
5.3	Time average versus ensemble average . . . . .	115
5.3.1	Motivation . . . . .	115
5.3.2	Definition . . . . .	116
5.3.3	Interpretation . . . . .	116
5.3.4	Equivalence . . . . .	119
5.3.5	Simulation . . . . .	121
5.3.6	Average time in system . . . . .	121
5.4	Related readings . . . . .	122
5.5	Exercises . . . . .	122
<b>III</b>	<b>The Predictive Power of Simple Operational Laws: “What-If” Questions and Answers</b>	<b>125</b>
<b>6</b>	<b>Operational Laws</b>	<b>129</b>
6.1	Little’s law for open systems . . . . .	129

6.2	Intuitions . . . . .	130
6.3	Little's law for closed systems . . . . .	130
6.4	Open systems . . . . .	131
6.4.1	Corollaries . . . . .	136
6.5	Closed systems . . . . .	138
6.5.1	Statement . . . . .	138
6.5.2	Proof . . . . .	138
6.6	Generalized Little's law . . . . .	140
6.7	Examples applying Little's law . . . . .	140
6.8	More operational laws: The forced flow law . . . . .	144
6.9	Combining operational laws . . . . .	147
6.10	Device demands . . . . .	150
6.11	Readings and further topics related to Little's law . . . . .	152
6.12	Exercises . . . . .	152
<b>7</b>	<b>Modification Analysis</b>	<b>155</b>
7.1	Review . . . . .	156
7.2	Asymptotic bounds for closed systems . . . . .	157
7.3	Modification analysis for closed systems . . . . .	161
7.4	More modification analysis examples . . . . .	163
7.5	Comparison of closed and open networks . . . . .	167
7.6	Readings . . . . .	168
7.7	Exercises . . . . .	168
<b>IV</b>	<b>From Markov Chains to Simple Queues</b>	<b>173</b>
<b>8</b>	<b>Discrete-time Markov Chains</b>	<b>177</b>
8.1	Discrete time versus continuous time Markov chains . . . . .	179
8.2	Definition of DTMC . . . . .	179
8.3	Examples of finite state DTMCs . . . . .	180
8.3.1	Repair facility problem . . . . .	180
8.3.2	Umbrella problem . . . . .	181
8.3.3	Program analysis problem . . . . .	182
8.4	Powers of $P$ : n-step transition probabilities . . . . .	183
8.5	Stationary equations . . . . .	186
8.6	The stationary distribution equals the limiting distribution . . . . .	187
8.7	Examples of solving stationary equations . . . . .	190
8.7.1	Repair facility problem with cost . . . . .	190
8.7.2	Umbrella problem . . . . .	191
8.8	Infinite state DTMCs . . . . .	192

8.9	Infinite-state stationarity result . . . . .	192
8.10	Solving stationary equations in infinite-state DTMC . . . . .	196
8.11	Exercises . . . . .	200
<b>9</b>	<b>Ergodicity Theory</b>	<b>203</b>
9.1	Ergodicity questions . . . . .	203
9.2	Finite-state DTMCs . . . . .	205
9.2.1	Existence of limiting distribution . . . . .	205
9.2.2	Mean time between visits to a state . . . . .	210
9.2.3	Time averages . . . . .	212
9.3	Infinite-state Markov chains . . . . .	213
9.3.1	Recurrent versus transient . . . . .	214
9.3.2	Positive recurrent versus null recurrent . . . . .	222
9.3.3	Ergodic theorem of Markov chains . . . . .	224
9.3.4	Time averages . . . . .	226
9.4	Limiting probabilities interpreted as rates . . . . .	230
9.5	Time-reversibility theorem . . . . .	232
9.6	Conclusion . . . . .	234
9.7	Proof of Ergodic Thm of Markov Chains* . . . . .	235
9.8	Exercises . . . . .	242
<b>10</b>	<b>Real-World Examples: Google, Aloha</b>	<b>247</b>
10.1	Google's page rank algorithm . . . . .	247
10.1.1	Google's DTMC algorithm . . . . .	247
10.1.2	Problems with real web graphs . . . . .	251
10.1.3	Google's solution to dead ends and spider traps . . . . .	254
10.1.4	Evaluation of Google algorithm . . . . .	255
10.1.5	Practical implementation considerations . . . . .	255
10.2	Aloha protocol analysis . . . . .	256
10.2.1	The slotted Aloha protocol . . . . .	257
10.2.2	The Markov chain . . . . .	257
10.2.3	Properties of the Markov chain . . . . .	259
10.2.4	Improving the protocol . . . . .	261
10.3	Summary and readings . . . . .	262
10.4	Exercises . . . . .	262
<b>11</b>	<b>Generating Functions for Markov Chains</b>	<b>265</b>
11.1	The z-transform . . . . .	266
11.2	Solving for $f_n$ . . . . .	267
11.3	Example: Simple random walk . . . . .	270
11.4	Exercises . . . . .	273

<b>12 Exponential Distribution &amp; Poisson Process</b>	<b>275</b>
12.1 Definition of the exponential distribution . . . . .	275
12.2 Memoryless property of the exponential . . . . .	277
12.3 Relating exponential to geometric . . . . .	280
12.4 More properties of the exponential . . . . .	281
12.5 The celebrated Poisson process . . . . .	285
12.6 Merging independent Poisson processes . . . . .	290
12.7 Poisson splitting . . . . .	291
12.7.1 Uniformity . . . . .	294
12.8 Exercises . . . . .	295
<b>13 Transition to Continuous-Time Markov Chains</b>	<b>299</b>
13.1 Defining CTMC's . . . . .	299
13.2 Solving CTMCs . . . . .	305
13.3 Generalization and interpretation . . . . .	310
13.3.1 Interpreting the balance equations for CTMC . . . . .	310
13.4 Reading remarks . . . . .	312
13.5 Exercises . . . . .	313
<b>14 M/M/1 and PASTA</b>	<b>315</b>
14.1 The M/M/1 queue . . . . .	315
14.2 Examples on M/M/1 . . . . .	320
14.3 PASTA . . . . .	323
14.4 Further reading . . . . .	328
14.5 Exercises . . . . .	328
<b>V Server Farms and Networks: Multi-server, Multi-queue systems</b>	<b>333</b>
<b>15 Server Farms: M/M/k &amp; M/M/k/k</b>	<b>337</b>
15.1 Time-reversibility for CTMCs . . . . .	338
15.2 M/M/k/k loss system . . . . .	340
15.3 M/M/k . . . . .	344
15.4 Comparison of 3 server organizations . . . . .	351
15.5 Readings . . . . .	353
15.6 Exercises . . . . .	353
<b>16 Capacity Provisioning for Server Farms</b>	<b>361</b>
16.1 What does load really mean in an M/M/k? . . . . .	362
16.2 The M/M/ $\infty$ . . . . .	364
16.2.1 Analysis of the M/M/ $\infty$ . . . . .	364



16.2.2	A first-cut at a capacity provisioning rule . . . . .	366
16.3	Square-root staffing . . . . .	367
16.4	Readings . . . . .	371
16.5	Exercises . . . . .	371
<b>17</b>	<b>Time-Reversibility &amp; Burke's Theorem</b>	<b>373</b>
17.1	More examples of finite-state CTMCs . . . . .	374
17.1.1	Networks with finite buffer space . . . . .	374
17.1.2	Batch System With M/M/2 I/O . . . . .	376
17.2	The reverse chain . . . . .	378
17.3	Burke's theorem . . . . .	382
17.4	An alternative (partial) proof of Burke's theorem . . . . .	384
17.5	Application: tandem servers . . . . .	386
17.6	General acyclic networks with probabilistic routing . . . . .	389
17.7	Readings . . . . .	390
17.8	Exercises . . . . .	390
<b>18</b>	<b>Jackson Network of Queues</b>	<b>391</b>
18.1	Jackson network . . . . .	391
18.1.1	Why the prior solution approaches don't work . . . . .	393
18.1.2	The local balance approach . . . . .	396
18.2	Readings . . . . .	403
18.3	Exercises . . . . .	403
<b>19</b>	<b>Classed Network of Queues</b>	<b>407</b>
19.1	Overview . . . . .	407
19.2	Motivation for classed networks . . . . .	408
19.3	Notation and modeling for classed Jackson networks . . . . .	411
19.4	A single-server classed network . . . . .	414
19.5	Product form theorems . . . . .	417
19.6	Examples using classed networks . . . . .	423
19.7	Readings . . . . .	432
19.8	Exercises . . . . .	433
<b>20</b>	<b>Closed Networks of Queues</b>	<b>435</b>
20.1	Motivation . . . . .	435
20.2	Product-form solution . . . . .	437
20.2.1	Local balance equations for closed network . . . . .	439
20.2.2	Summary: solving closed batch Jackson networks . . . . .	441
20.2.3	Example of deriving limiting probabilities . . . . .	441
20.3	Mean Value Analysis (MVA) . . . . .	443
20.4	Readings . . . . .	452

20.5 Exercises . . . . .	453
<b>VI Real-World Workloads: High-Variability and Heavy Tails</b>	<b>455</b>
<b>21 Tales of Tails: Real-World Workloads</b>	<b>459</b>
21.1 Grad school tales ... process migration . . . . .	459
21.2 UNIX process lifetime measurements . . . . .	461
21.3 Properties of the Pareto distribution . . . . .	464
21.4 The Bounded Pareto distribution . . . . .	465
21.5 Heavy tails . . . . .	466
21.6 To migrate or not to migrate? . . . . .	467
21.7 Pareto distributions are everywhere . . . . .	468
21.8 Exercises . . . . .	469
<b>22 Phase-Type Workloads &amp; Matrix-Analytic</b>	<b>473</b>
22.1 Representing general distributions by Exponentials . . . . .	474
22.2 Markov chain modeling of PH workloads . . . . .	479
22.3 The Matrix-Analytic method . . . . .	483
22.4 Analysis of time-varying load . . . . .	484
22.4.1 High-level ideas . . . . .	484
22.4.2 The generator matrix, $\mathbf{Q}$ . . . . .	485
22.4.3 Solving for $\mathbf{R}$ . . . . .	488
22.4.4 Finding $\vec{\pi}_0$ . . . . .	489
22.4.5 Performance metrics . . . . .	490
22.5 More complex chains . . . . .	491
22.6 Readings and further remarks . . . . .	495
22.7 Exercises . . . . .	496
<b>23 Networks of Time-Sharing (PS) Servers</b>	<b>499</b>
23.1 Review of product-form networks . . . . .	499
23.2 BCMP result . . . . .	500
23.2.1 Networks with FCFS servers . . . . .	500
23.2.2 Networks with PS servers . . . . .	502
23.3 M/M/1/PS . . . . .	505
23.4 M/Cox/1/PS . . . . .	506
23.5 Tandem network of M/G/1/PS servers . . . . .	514
23.6 Network of PS servers with probabilistic routing . . . . .	518
23.7 Readings . . . . .	518
23.8 Exercises . . . . .	519

<b>24 M/G/1 Queue &amp; Inspection Paradox</b>	<b>521</b>
24.1 The inspection paradox . . . . .	522
24.2 The M/G/1 queue and its analysis . . . . .	522
24.3 Renewal-reward theory . . . . .	526
24.4 Deriving expected excess . . . . .	529
24.5 Back to the inspection paradox . . . . .	532
24.6 Back to the $M/G/1$ queue . . . . .	534
24.7 Exercises . . . . .	535
<b>25 Task Assignment for Server Farms</b>	<b>539</b>
25.1 Task assignment for non-preemptive workloads . . . . .	541
25.2 Task assignment for PS server farms . . . . .	554
25.3 Optimal server farm design . . . . .	560
25.4 Readings and further followup . . . . .	565
25.5 Exercises . . . . .	568
<b>26 Transform Analysis</b>	<b>573</b>
26.1 Definitions of transforms and some examples . . . . .	573
26.2 Getting moments from transforms . . . . .	577
26.3 Linearity of transforms . . . . .	581
26.4 Conditioning . . . . .	584
26.4.1 Example: Distribution of response time in an M/M/1 . . . . .	585
26.5 Combining Laplace and z-transforms . . . . .	586
26.6 More results on transforms* . . . . .	588
26.7 Readings . . . . .	590
26.8 Exercises . . . . .	590
<b>27 M/G/1 Transform Analysis</b>	<b>593</b>
27.1 Two-step outline . . . . .	593
27.2 The z-transform of the number in system . . . . .	593
27.3 The Laplace transform of time in system . . . . .	598
27.4 Readings . . . . .	601
27.5 Exercises . . . . .	601
<b>28 Power Optimization Application</b>	<b>603</b>
28.1 The power optimization problem . . . . .	604
28.2 Busy period analysis of M/G/1 . . . . .	606
28.3 M/G/1 with setup cost . . . . .	610
28.4 Comparing ON/IDLE versus ON/OFF . . . . .	614
28.5 Readings . . . . .	616
28.6 Exercises . . . . .	616

<b>VII Smart Scheduling</b>	<b>619</b>
<b>29 Performance Metrics</b>	<b>623</b>
29.1 Traditional metrics . . . . .	623
29.2 Commonly-used metrics for single queues . . . . .	624
29.3 Today's trendy metrics . . . . .	625
29.4 Starvation/fairness metrics . . . . .	626
29.5 Deriving performance metrics . . . . .	627
<b>30 Non-Preemptive, Non-Size-Based Policies</b>	<b>629</b>
30.1 FCFS, LCFS, and RANDOM . . . . .	629
30.2 Readings . . . . .	634
30.3 Exercises . . . . .	635
<b>31 Preemptive, Non-Size-Based Policies</b>	<b>637</b>
31.1 Processor-Sharing (PS) . . . . .	637
31.1.1 Motivation behind PS . . . . .	637
31.1.2 Ages of jobs in M/G/1/PS system . . . . .	639
31.1.3 Response time as a function of job size . . . . .	640
31.1.4 Intuition for PS results . . . . .	643
31.1.5 Implications of PS result on understanding FCFS . . . . .	644
31.2 Preemptive-LCFS . . . . .	646
31.3 FB scheduling . . . . .	649
31.4 Proof of Theorem 69 (Optional Reading)* . . . . .	655
31.4.1 Reverse chain lemma . . . . .	655
31.4.2 Easy example: M/M/1/PS . . . . .	656
31.4.3 M/G/1/PS . . . . .	658
31.5 Readings . . . . .	664
31.6 Exercises . . . . .	664
<b>32 Non-Preemptive, Size-Based Policies</b>	<b>667</b>
32.1 Priority queueing . . . . .	668
32.2 Non-preemptive priority . . . . .	669
32.3 Shortest Job First (SJF) . . . . .	673
<b>33 Preemptive, Size-Based Policies</b>	<b>679</b>
33.1 Motivation . . . . .	679
33.2 Preemptive priority queueing for M/G/1 . . . . .	680
33.3 Preemptive Shortest Job First (PSJF) . . . . .	685
33.4 Transform analysis of PSJF . . . . .	688
33.5 Exercises . . . . .	691
<b>34 Scheduling: SRPT and Fairness</b>	<b>693</b>

34.1	Shortest Remaining Processing Time (SRPT) . . . . .	693
34.2	Precise derivation of SRPT waiting time* . . . . .	697
34.3	Comparisons with other policies . . . . .	700
34.3.1	Comparison with PSJF . . . . .	701
34.3.2	SRPT versus FB . . . . .	701
34.3.3	Comparison of all scheduling policies . . . . .	702
34.4	Fairness of SRPT . . . . .	705
34.5	Readings . . . . .	710