

Service Quality of Cloud-Based Applications

Eric Bauer
Randee Adams

 **IEEE**
IEEE PRESS

WILEY

SERVICE QUALITY OF CLOUD-BASED APPLICATIONS



IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

IEEE Press Editorial Board 2013
John Anderson, *Editor in Chief*

Linda Shafer	Saeid Nahavandi	George Zobrist
George W. Arnold	Om P. Malik	Tariq Samad
Ekram Hossain	Mary Lanzerotti	Dmitry Goldgof

Kenneth Moore, *Director of IEEE Book and Information Services (BIS)*

Technical Reviewers

Kim W. Tracy, *Northeastern Illinois University*
Rocky Heckman, *CISSP, Architect Advisor, Microsoft*

SERVICE QUALITY OF CLOUD-BASED APPLICATIONS

Eric Bauer
Randee Adams



IEEE PRESS

WILEY

Copyright © 2014 by The Institute of Electrical and Electronics Engineers, Inc.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey. All rights reserved

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Bauer, Eric.

Service quality of cloud-based applications / Eric Bauer, Randee Adams.

pages cm

ISBN 978-1-118-76329-2 (cloth)

1. Cloud computing. 2. Application software—Reliability. 3. Quality of service (Computer networks) I. Adams, Randee. II. Title.

QA76.585.B3944 2013

004.67'82—dc23

2013026569

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

CONTENTS

Figures	xv
Tables and Equations	xxi
1 INTRODUCTION	1
1.1 Approach	1
1.2 Target Audience	3
1.3 Organization	3
I CONTEXT	7
2 APPLICATION SERVICE QUALITY	9
2.1 Simple Application Model	9
2.2 Service Boundaries	11
2.3 Key Quality and Performance Indicators	12
2.4 Key Application Characteristics	15
2.4.1 Service Criticality	15
2.4.2 Application Interactivity	16
2.4.3 Tolerance to Network Traffic Impairments	17
2.5 Application Service Quality Metrics	17
2.5.1 Service Availability	18
2.5.2 Service Latency	19
2.5.3 Service Reliability	24
2.5.4 Service Accessibility	25
2.5.5 Service Retainability	25

2.5.6	Service Throughput	25
2.5.7	Service Timestamp Accuracy	26
2.5.8	Application-Specific Service Quality Measurements	26
2.6	Technical Service versus Support Service	27
2.6.1	Technical Service Quality	27
2.6.2	Support Service Quality	27
2.7	Security Considerations	28
3	CLOUD MODEL	29
3.1	Roles in Cloud Computing	30
3.2	Cloud Service Models	30
3.3	Cloud Essential Characteristics	31
3.3.1	On-Demand Self-Service	31
3.3.2	Broad Network Access	31
3.3.3	Resource Pooling	32
3.3.4	Rapid Elasticity	32
3.3.5	Measured Service	33
3.4	Simplified Cloud Architecture	33
3.4.1	Application Software	34
3.4.2	Virtual Machine Servers	35
3.4.3	Virtual Machine Server Controllers	35
3.4.4	Cloud Operations Support Systems	36
3.4.5	Cloud Technology Components Offered “as-a-Service”	36
3.5	Elasticity Measurements	36
3.5.1	Density	37
3.5.2	Provisioning Interval	37
3.5.3	Release Interval	39
3.5.4	Scaling In and Out	40
3.5.5	Scaling Up and Down	41
3.5.6	Agility	42
3.5.7	Slew Rate and Linearity	43
3.5.8	Elasticity Speedup	44
3.6	Regions and Zones	44
3.7	Cloud Awareness	45
4	VIRTUALIZED INFRASTRUCTURE IMPAIRMENTS	49
4.1	Service Latency, Virtualization, and the Cloud	50
4.1.1	Virtualization and Cloud Causes of Latency Variation	51
4.1.2	Virtualization Overhead	52
4.1.3	Increased Variability of Infrastructure Performance	53
4.2	VM Failure	54
4.3	Nondelivery of Configured VM Capacity	54

4.4	Delivery of Degraded VM Capacity	57
4.5	Tail Latency	59
4.6	Clock Event Jitter	60
4.7	Clock Drift	61
4.8	Failed or Slow Allocation and Startup of VM Instance	62
4.9	Outlook for Virtualized Infrastructure Impairments	63
II	ANALYSIS	65
5	APPLICATION REDUNDANCY AND CLOUD COMPUTING	67
5.1	Failures, Availability, and Simplex Architectures	68
5.2	Improving Software Repair Times via Virtualization	70
5.3	Improving Infrastructure Repair Times via Virtualization	72
5.3.1	Understanding Hardware Repair	72
5.3.2	VM Repair-as-a-Service	72
5.3.3	Discussion	74
5.4	Redundancy and Recoverability	75
5.4.1	Improving Recovery Times via Virtualization	79
5.5	Sequential Redundancy and Concurrent Redundancy	80
5.5.1	Hybrid Concurrent Strategy	83
5.6	Application Service Impact of Virtualization Impairments	84
5.6.1	Service Impact for Simplex Architectures	85
5.6.2	Service Impact for Sequential Redundancy Architectures	85
5.6.3	Service Impact for Concurrent Redundancy Architectures	87
5.6.4	Service Impact for Hybrid Concurrent Architectures	88
5.7	Data Redundancy	90
5.7.1	Data Storage Strategies	90
5.7.2	Data Consistency Strategies	91
5.7.3	Data Architecture Considerations	92
5.8	Discussion	92
5.8.1	Service Quality Impact	93
5.8.2	Concurrency Control	93
5.8.3	Resource Usage	94
5.8.4	Simplicity	94
5.8.5	Other Considerations	95
6	LOAD DISTRIBUTION AND BALANCING	97
6.1	Load Distribution Mechanisms	97
6.2	Load Distribution Strategies	99

6.3	Proxy Load Balancers	99
6.4	Nonproxy Load Distribution	101
6.5	Hierarchy of Load Distribution	102
6.6	Cloud-Based Load Balancing Challenges	103
6.7	The Role of Load Balancing in Support of Redundancy	103
6.8	Load Balancing and Availability Zones	104
6.9	Workload Service Measurements	104
6.10	Operational Considerations	105
6.10.1	Load Balancing and Elasticity	105
6.10.2	Load Balancing and Overload	106
6.10.3	Load Balancing and Release Management	107
6.11	Load Balancing and Application Service Quality	107
6.11.1	Service Availability	107
6.11.2	Service Latency	108
6.11.3	Service Reliability	108
6.11.4	Service Accessibility	109
6.11.5	Service Retainability	109
6.11.6	Service Throughput	109
6.11.7	Service Timestamp Accuracy	109
7	FAILURE CONTAINMENT	111
7.1	Failure Containment	111
7.1.1	Failure Cascades	112
7.1.2	Failure Containment and Recovery	112
7.1.3	Failure Containment and Virtualization	114
7.2	Points of Failure	116
7.2.1	Single Points of Failure	116
7.2.2	Single Points of Failure and Virtualization	117
7.2.3	Affinity and Anti-affinity Considerations	119
7.2.4	No SPOF Assurance in Cloud Computing	120
7.2.5	No SPOF and Application Data	121
7.3	Extreme Solution Coresidency	122
7.3.1	Extreme Solution Coresidency Risks	123
7.4	Multitenancy and Solution Containers	124
8	CAPACITY MANAGEMENT	127
8.1	Workload Variations	128
8.2	Traditional Capacity Management	129
8.3	Traditional Overload Control	129
8.4	Capacity Management and Virtualization	131
8.5	Capacity Management in Cloud	133

8.6	Storage Elasticity Considerations	135
8.7	Elasticity and Overload	136
8.8	Operational Considerations	137
8.9	Workload Whipsaw	138
8.10	General Elasticity Risks	140
8.11	Elasticity Failure Scenarios	141
8.11.1	Elastic Growth Failure Scenarios	141
8.11.2	Elastic Capacity Degrowth Failure Scenarios	143
9	RELEASE MANAGEMENT	145
9.1	Terminology	145
9.2	Traditional Software Upgrade Strategies	146
9.2.1	Software Upgrade Requirements	146
9.2.2	Maintenance Windows	148
9.2.3	Client Considerations for Application Upgrade	149
9.2.4	Traditional Offline Software Upgrade	150
9.2.5	Traditional Online Software Upgrade	151
9.2.6	Discussion	153
9.3	Cloud-Enabled Software Upgrade Strategies	153
9.3.1	Type I Cloud-Enabled Upgrade Strategy: Block Party	154
9.3.2	Type II Cloud-Enabled Upgrade Strategy: One Driver per Bus	156
9.3.3	Discussion	157
9.4	Data Management	158
9.5	Role of Service Orchestration in Software Upgrade	159
9.5.1	Solution-Level Software Upgrade	160
9.6	Conclusion	161
10	END-TO-END CONSIDERATIONS	163
10.1	End-to-End Service Context	163
10.2	Three-Layer End-to-End Service Model	169
10.2.1	Estimating Service Impairments via the Three-Layer Model	171
10.2.2	End-to-End Service Availability	172
10.2.3	End-to-End Service Latency	173
10.2.4	End-to-End Service Reliability	174
10.2.5	End-to-End Service Accessibility	175
10.2.6	End-to-End Service Retainability	176
10.2.7	End-to-End Service Throughput	176
10.2.8	End-to-End Service Timestamp Accuracy	177
10.2.9	Reality Check	177

10.3	Distributed and Centralized Cloud Data Centers	177
10.3.1	Centralized Cloud Data Centers	178
10.3.2	Distributed Cloud Data Centers	178
10.3.3	Service Availability Considerations	179
10.3.4	Service Latency Considerations	181
10.3.5	Service Reliability Considerations	182
10.3.6	Service Accessibility Considerations	182
10.3.7	Service Retainability Considerations	182
10.3.8	Resource Distribution Considerations	182
10.4	Multitiered Solution Architectures	183
10.5	Disaster Recovery and Geographic Redundancy	184
10.5.1	Disaster Recovery Objectives	184
10.5.2	Georedundant Architectures	185
10.5.3	Service Quality Considerations	186
10.5.4	Recovery Point Considerations	187
10.5.5	Mitigating Impact of Disasters with Georedundancy and Availability Zones	189

III RECOMMENDATIONS 191

11 ACCOUNTABILITIES FOR SERVICE QUALITY 193

11.1	Traditional Accountability	193
11.2	The Cloud Service Delivery Path	194
11.3	Cloud Accountability	197
11.4	Accountability Case Studies	200
11.4.1	Accountability and Technology Components	201
11.4.2	Accountability and Elasticity	203
11.5	Service Quality Gap Model	205
11.5.1	Application's Resource Facing Service Gap Analysis	206
11.5.2	Application's Customer Facing Service Gap Analysis	208
11.6	Service Level Agreements	210

12 SERVICE AVAILABILITY MEASUREMENT 213

12.1	Parsimonious Service Measurements	214
12.2	Traditional Service Availability Measurement	215
12.3	Evolving Service Availability Measurements	217
12.3.1	Analyzing Application Evolution	218
12.3.2	Technology Components	223
12.3.3	Leveraging Storage-as-a-Service	224

12.4	Evolving Hardware Reliability Measurement	226
12.4.1	Virtual Machine Failure Lifecycle	226
12.5	Evolving Elasticity Service Availability Measurements	228
12.6	Evolving Release Management Service Availability Measurement	229
12.7	Service Measurement Outlook	231
13	APPLICATION SERVICE QUALITY REQUIREMENTS	233
13.1	Service Availability Requirements	234
13.2	Service Latency Requirements	237
13.3	Service Reliability Requirements	237
13.4	Service Accessibility Requirements	238
13.5	Service Retainability Requirements	239
13.6	Service Throughput Requirements	239
13.7	Timestamp Accuracy Requirements	240
13.8	Elasticity Requirements	240
13.9	Release Management Requirements	241
13.10	Disaster Recovery Requirements	241
14	VIRTUALIZED INFRASTRUCTURE MEASUREMENT AND MANAGEMENT	243
14.1	Business Context for Infrastructure Service Quality Measurements	244
14.2	Cloud Consumer Measurement Options	245
14.3	Impairment Measurement Strategies	247
14.3.1	Measurement of VM Failure	247
14.3.2	Measurement of Nondelivery of Configured VM Capacity	249
14.3.3	Measurement of Delivery of Degraded VM Capacity	249
14.3.4	Measurement of Tail Latency	249
14.3.5	Measurement of Clock Event Jitter	250
14.3.6	Measurement of Clock Drift	250
14.3.7	Measurement of Failed or Slow Allocation and Startup of VM Instance	250
14.3.8	Measurements Summary	251
14.4	Managing Virtualized Infrastructure Impairments	252
14.4.1	Minimize Application's Sensitivity to Infrastructure Impairments	252
14.4.2	VM-Level Congestion Detection and Control	252
14.4.3	Allocate More Virtual Resource Capacity	253

14.4.4	Terminate Poorly Performing VM Instances	253
14.4.5	Accept Degraded Performance	253
14.4.6	Proactive Supplier Management	254
14.4.7	Reset End Users' Service Quality Expectations	254
14.4.8	SLA Considerations	254
14.4.9	Changing Cloud Service Providers	254
15	ANALYSIS OF CLOUD-BASED APPLICATIONS	255
15.1	Reliability Block Diagrams and Side-by-Side Analysis	256
15.2	IaaS Impairment Effects Analysis	257
15.3	PaaS Failure Effects Analysis	259
15.4	Workload Distribution Analysis	260
15.4.1	Service Quality Analysis	261
15.4.2	Overload Control Analysis	261
15.5	Anti-Affinity Analysis	262
15.6	Elasticity Analysis	263
15.6.1	Service Capacity Growth Scenarios	264
15.6.2	Service Capacity Growth Action Analysis	264
15.6.3	Service Capacity Degrowth Action Analysis	265
15.6.4	Storage Capacity Growth Scenarios	265
15.6.5	Online Storage Capacity Growth Action Analysis	266
15.6.6	Online Storage Capacity Degrowth Action Analysis	266
15.7	Release Management Impact Effects Analysis	267
15.7.1	Service Availability Impact	267
15.7.2	Server Reliability Impact	267
15.7.3	Service Accessibility Impact	267
15.7.4	Service Retainability Impact	267
15.7.5	Service Throughput Impact	267
15.8	Recovery Point Objective Analysis	268
15.9	Recovery Time Objective Analysis	270
16	TESTING CONSIDERATIONS	273
16.1	Context for Testing	273
16.2	Test Strategy	274
16.2.1	Cloud Test Bed	275
16.2.2	Application Capacity under Test	275
16.2.3	Statistical Confidence	276
16.2.4	Service Disruption Time	276
16.3	Simulating Infrastructure Impairments	277
16.4	Test Planning	278
16.4.1	Service Reliability and Latency Testing	279
16.4.2	Impaired Infrastructure Testing	280

16.4.3	Robustness Testing	280
16.4.4	Endurance/Stability Testing	282
16.4.5	Application Elasticity Testing	284
16.4.6	Upgrade Testing	285
16.4.7	Disaster Recovery Testing	285
16.4.8	Extreme Coresidency Testing	286
16.4.9	PaaS Technology Component Testing	286
16.4.10	Automated Regression Testing	286
16.4.11	Canary Release Testing	286
17	CONNECTING THE DOTS	287
17.1	The Application Service Quality Challenge	287
17.2	Redundancy and Robustness	289
17.3	Design for Scalability	292
17.4	Design for Extensibility	292
17.5	Design for Failure	293
17.6	Planning Considerations	294
17.7	Evolving Traditional Applications	296
17.7.1	Phase 0: Traditional Application	298
17.7.2	Phase I: High Service Quality on Virtualized Infrastructure	298
17.7.3	Phase II: Manual Application Elasticity	299
17.7.4	Phase III: Automated Release Management	299
17.7.5	Phase IV: Automated Application Elasticity	300
17.7.6	Phase V: VM Migration	300
17.8	Concluding Remarks	301
	Abbreviations	303
	References	307
	About the Authors	311
	Index	313

FIGURES

Figure 1.1.	Sample Cloud-Based Application.	2
Figure 2.0.	Organization of Part I: Context.	8
Figure 2.1.	Simple Cloud-Based Application.	10
Figure 2.2.	Simple Virtual Machine Service Model.	10
Figure 2.3.	Application Service Boundaries.	11
Figure 2.4.	KQIs and KPIs.	12
Figure 2.5.	Application Consumer and Resource Facing Service Indicators.	14
Figure 2.6.	Application Robustness.	14
Figure 2.7.	Sample Application Robustness Scenario.	15
Figure 2.8.	Interactivity Timeline.	16
Figure 2.9.	Service Latency.	19
Figure 2.10.	Small Sample Service Latency Distribution.	22
Figure 2.11.	Sample Typical Latency Variation by Workload Density.	22
Figure 2.12.	Sample Tail Latency Variation by Workload Density.	23
Figure 2.13.	Understanding Complimentary Cumulative Distribution Plots.	23
Figure 2.14.	Service Latency Optimization Options.	24
Figure 3.1.	Cloud Roles for Simple Application.	30
Figure 3.2.	Elastic Growth Strategies.	32
Figure 3.3.	Simple Model of Cloud Infrastructure.	34
Figure 3.4.	Abstract Virtual Machine Server.	35
Figure 3.5.	Provisioning Interval (T_{Grow}).	38
Figure 3.6.	Release Interval T_{Shrink} .	39
Figure 3.7.	VM Scale In and Scale Out.	40
Figure 3.8.	Horizontal Elasticity.	40

Figure 3.9.	Scale Up and Scale Down of a VM Instance.	41
Figure 3.10.	Idealized (Linear) Capacity Agility.	42
Figure 3.11.	Slew Rate of Square Wave Amplification.	43
Figure 3.12.	Elastic Growth Slew Rate and Linearity.	43
Figure 3.13.	Regions and Availability Zones.	45
Figure 4.1.	Virtualized Infrastructure Impairments Experienced by Cloud-Based Applications.	50
Figure 4.2.	Transaction Latency for Riak Benchmark.	52
Figure 4.3.	VM Failure Impairment Example.	55
Figure 4.4.	Simplified Nondelivery of VM Capacity Model.	55
Figure 4.5.	Characterizing Virtual Machine Nondelivery.	56
Figure 4.6.	Nondelivery Impairment Example.	56
Figure 4.7.	Simple Virtual Machine Degraded Delivery Model.	57
Figure 4.8.	Degraded Resource Capacity Model.	58
Figure 4.9.	Degraded Delivery Impairment Example.	58
Figure 4.10.	CCDF for Riak Read Benchmark for Three Different Hosting Configurations.	59
Figure 4.11.	Tail Latency Impairment Example.	60
Figure 4.12.	Sample CCDF for Virtualized Clock Event Jitter.	61
Figure 4.13.	Clock Event Jitter Impairment Example.	61
Figure 4.14.	Clock Drift Impairment Example.	62
Figure 5.1.	Simplex Distributed System.	68
Figure 5.2.	Simplex Service Availability.	68
Figure 5.3.	Sensitivity of Service Availability to MTRS (Log Scale).	70
Figure 5.4.	Traditional versus Virtualized Software Repair Times.	71
Figure 5.5.	Traditional Hardware Repair versus Virtualized Infrastructure Restoration Times.	72
Figure 5.6.	Simplified VM Repair Logic.	73
Figure 5.7.	Sample Automated Virtual Machine Repair-as-a-Service Logic.	74
Figure 5.8.	Simple Redundancy Model.	75
Figure 5.9.	Simplified High Availability Strategy.	76
Figure 5.10.	Failure in a Traditional (Sequential) Redundant Architecture.	76
Figure 5.11.	Sequential Redundancy Model.	77
Figure 5.12.	Sequential Redundant Architecture Timeline with No Failures.	77
Figure 5.13.	Sample Redundant Architecture Timeline with Implicit Failure.	78
Figure 5.14.	Sample Redundant Architecture Timeline with Explicit Failure.	79
Figure 5.15.	Recovery Times for Traditional Redundancy Architectures.	80
Figure 5.16.	Concurrent Redundancy Processing Model.	81
Figure 5.17.	Client Controlled Redundant Compute Strategy.	82
Figure 5.18.	Client Controlled Redundant Operations.	83
Figure 5.19.	Concurrent Redundancy Timeline with Fast but Erroneous Return.	83
Figure 5.20.	Hybrid Concurrent with Slow Response.	84
Figure 5.21.	Application Service Impact for Very Brief Nondelivery Events.	86
Figure 5.22.	Application Service Impact for Brief Nondelivery Events.	86

Figure 5.23.	Nondelivery Impact to Redundant Compute Architectures.	88
Figure 5.24.	Nondelivery Impact to Hybrid Concurrent Architectures.	89
Figure 6.1.	Proxy Load Balancer.	98
Figure 6.2.	Proxy Load Balancing.	100
Figure 6.3.	Load Balancing between Regions and Availability Zones.	104
Figure 7.1.	Reliability Block Diagram of Simplex Sample System (with SPOF).	116
Figure 7.2.	Reliability Block Diagram of Redundant Sample System (without SPOF).	117
Figure 7.3.	No SPOF Distribution of Component Instances across Virtual Servers.	118
Figure 7.4.	Example of No Single Point of Failure with Distributed Component Instances.	118
Figure 7.5.	Example of Single Point of Failure with Poorly Distributed Component Instances.	119
Figure 7.6.	Simplified VM Server Control.	120
Figure 8.1.	Sample Daily Workload Variation (Logarithmic Scale).	128
Figure 8.2.	Traditional Maintenance Window.	129
Figure 8.3.	Traditional Congestion Control.	130
Figure 8.4.	Simplified Elastic Growth of Cloud-Based Applications.	134
Figure 8.5.	Simplified Elastic Degrowth of Cloud-Based Applications.	135
Figure 8.6.	Sample of Erratic Workload Variation (Linear Scale).	138
Figure 8.7.	Typical Elasticity Orchestration Process.	139
Figure 8.8.	Example of Workload Whipsaw.	139
Figure 8.9.	Elastic Growth Failure Scenarios.	141
Figure 9.1.	Traditional Offline Software Upgrade.	150
Figure 9.2.	Traditional Online Software Upgrade.	151
Figure 9.3.	Type I, “Block Party” Upgrade Strategy.	154
Figure 9.4.	Application Elastic Growth and Type I, “Block Party” Upgrade.	155
Figure 9.5.	Type II, “One Driver per Bus” Upgrade Strategy.	156
Figure 10.1.	Simple End-to-End Application Service Context.	164
Figure 10.2.	Service Boundaries in End-to-End Application Service Context.	165
Figure 10.3.	Measurement Points 0–4 for Simple End-to-End Context.	166
Figure 10.4.	End-to-End Measurement Points for Simple Replicated Solution Context.	167
Figure 10.5.	Service Probes across User Service Delivery Path.	168
Figure 10.6.	Three Layer Factorization of Sample End to End Solution.	170
Figure 10.7.	Estimating Service Impairments across the Three-Layer Model.	171
Figure 10.8.	Decomposing a Service Impairment.	172
Figure 10.9.	Centralized Cloud Data Center Scenario.	178
Figure 10.10.	Distributed Cloud Data Center Scenario.	179
Figure 10.11.	Sample Multitier Solution Architecture.	184
Figure 10.12.	Disaster Recovery Time and Point Objectives.	185
Figure 10.13.	Service Impairment Model of Georedundancy.	187

Figure 11.1.	Traditional Three-Way Accountability Split: Suppliers, Customers, External.	195
Figure 11.2.	Example Cloud Service Delivery Chain.	195
Figure 11.3.	Service Boundaries across Cloud Delivery Chain.	196
Figure 11.4.	Functional Responsibilities for Applications Deployed on IaaS.	198
Figure 11.5.	Sample Application.	201
Figure 11.6.	Service Outage Accountability of Sample Application.	201
Figure 11.7.	Application Elasticity Configuration.	203
Figure 11.8.	Service Gap Model.	205
Figure 11.9.	Service Quality Zone of Tolerance.	206
Figure 11.10.	Application's Resource Facing Service Boundary.	207
Figure 11.11.	Application's Customer Facing Service Boundary.	208
Figure 12.1.	Traditional Service Operation Timeline.	216
Figure 12.2.	Sample Application Deployment on Cloud.	217
Figure 12.3.	"Network Element" Boundary for Sample Application.	218
Figure 12.4.	Logical Measurement Point for Application's Service Availability.	218
Figure 12.5.	Reliability Block Diagram of Sample Application (Traditional Deployment).	219
Figure 12.6.	Evolving Sample Application to Cloud.	220
Figure 12.7.	Reliability Block Diagram of Sample Application on Cloud.	220
Figure 12.8.	Side-by-Side Reliability Block Diagrams.	221
Figure 12.9.	Accountability of Sample Cloud Based Application.	221
Figure 12.10.	Connectivity-as-a-Service as a Nanoscale VPN.	222
Figure 12.11.	Sample Application with Database-as-a-Service.	224
Figure 12.12.	Accountability of Sample Application with Database-as-a-Service.	224
Figure 12.13.	Sample Application with Outboard RAID Storage Array.	225
Figure 12.14.	Sample Application with Storage-as-a-Service.	225
Figure 12.15.	Accountability of Sample Application with Storage-as-a-Service.	226
Figure 12.16.	Virtual Machine Failure Lifecycle.	227
Figure 12.17.	Elastic Capacity Growth Timeline.	229
Figure 12.18.	Outage Normalization for Type I "Block Party" Release Management.	230
Figure 12.19.	Outage Normalization for Type II "One Driver per Bus" Release Management.	231
Figure 13.1.	Maximum Acceptable Service Disruption.	235
Figure 14.1.	Infrastructure impairments and application impairments.	244
Figure 14.2.	Loopback and Service Latency.	246
Figure 14.3.	Simplified Measurement Architecture.	251
Figure 15.1.	Sample Side-by-Side Reliability Block Diagrams.	256
Figure 15.2.	Worst-Case Recovery Point Scenario.	268
Figure 15.3.	Best-Case Recovery Point Scenario.	269
Figure 16.1.	Measuring Service Disruption Latency.	277

Figure 16.2.	Service Disruption Latency for Implicit Failure.	277
Figure 16.3.	Sample Endurance Test Case for Cloud-Based Application.	283
Figure 17.1.	Virtualized Infrastructure Impairments Experienced by Cloud-Based Applications.	288
Figure 17.2.	Application Robustness Challenge.	289
Figure 17.3.	Sequential (Traditional) Redundancy.	290
Figure 17.4.	Concurrent Redundancy.	290
Figure 17.5.	Hybrid Concurrent with Slow Response.	291
Figure 17.6.	Type I, “Block Party” Upgrade Strategy.	293
Figure 17.7.	Sample Phased Evolution of a Traditional Application.	296

TABLES AND EQUATIONS

TABLES

TABLE 2.1.	Mean Opinion Scores [P.800]	26
TABLE 13.1.	Service Availability and Downtime Ratings	236

EQUATIONS

Equation 2.1.	Availability Formula	18
Equation 5.1.	Simplex Availability	68
Equation 5.2.	Traditional Availability	69
Equation 10.1.	Estimating General End-to-End Service Impairments	171
Equation 10.2.	Estimating End-to-End Service Downtime	172
Equation 10.3.	Estimating End-to-End Service Availability	173
Equation 10.4.	Estimating End-to-End Typical Service Latency	173
Equation 10.5.	Estimating End-to-End Service Defect Rate	175
Equation 10.6.	Estimating End-to-End Service Accessibility	175
Equation 10.7.	Estimating End to End Service Retainability (as DPM)	176
Equation 13.1.	DPM via Operations Attempted and Operations Successful	238
Equation 13.2.	DPM via Operations Attempted and Operations Failed	238
Equation 13.3.	DPM via Operations Successful and Operations Failed	238
Equation 14.1.	Computing VM FITs	248
Equation 14.2.	Converting FITs to MTBF	249

INTRODUCTION

Customers expect that applications and services deployed on cloud computing infrastructure will deliver comparable service quality, reliability, availability, and latency as when deployed on traditional, native hardware configurations. Cloud computing infrastructure introduces a new family of service impairment risks based on the virtualized compute, memory, storage, and networking resources that an Infrastructure-as-a-Service (IaaS) provider delivers to hosted application instances. As a result, application developers and cloud consumers must mitigate these impairments to assure that application service delivered to end users is not unacceptably impacted. This book methodically analyzes the impacts of cloud infrastructure impairments on application service delivered to end users, as well as the opportunities for improvement afforded by cloud. The book also recommends architectures, policies, and other techniques to maximize the likelihood of delivering comparable or better service to end users when applications are deployed to cloud.

1.1 APPROACH

Cloud-based application software executes within a set of virtual machine instances, and each individual virtual machine instance relies on virtualized compute, memory,

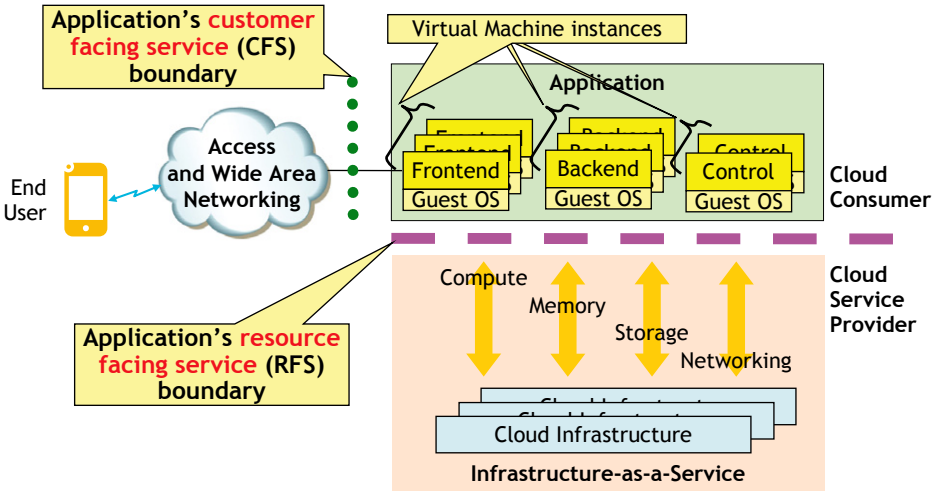


Figure 1.1. Sample Cloud-Based Application.

storage, and networking service delivered by the underlying cloud infrastructure. As shown in Figure 1.1, the application presents **customer facing service** toward end users across the dotted service boundary, and consumes virtualized resources offered by the Infrastructure-as-a-Service provider across the dashed **resource facing service** boundary. The application's service quality experienced by the end users is primarily a function of the application's architecture and software quality, as well as the service quality of the virtualized infrastructure offered by the IaaS across the resource facing service boundary, and the access and wide area networking that connects the end user to the application instance. This book considers both the new impairments and opportunities of virtualized resources offered to applications deployed on cloud and how user service quality experienced by end users can be maximized. By ignoring service impairments of the end user's device, and access and wide area network, one can narrowly consider how application service quality differs when a particular application is hosted on cloud infrastructure compared with when it is natively deployed on traditional hardware.

The key technical difference for application software between native deployment and cloud deployment is that native deployments offer the application's (guest) operating system direct access to the physical compute, memory, storage, and network resources, while cloud deployment inserts a layer of hypervisor or virtual machine management software between the guest operating system and the physical hardware. This layer of hypervisor or virtual machine management software enables sophisticated resource sharing, technical features, and operational policies. However, the hypervisor or virtual machine management layer does not deliver perfect hardware emulation to the guest operating system and application software, and these imperfections can adversely impact application service delivered to end users. While Figure 1.1 illustrates application deployment to a single data center, real world applications are often deployed

to multiple data centers to improve user service quality by shortening transport latency to end users, to support business continuity and disaster recovery, and for other business reasons. Application service quality for deployment across multiple data centers is also considered in this book.

This book considers how application architectures, configurations, validation, and operational policies should evolve so that the acceptable application service quality can be delivered to end users even when application software is deployed on cloud infrastructure. This book approaches application service quality from the end users perspective while considering standards and recommendations from NIST, TM Forum, QuEST Forum, ODCA, ISO, ITIL, and so on.

1.2 TARGET AUDIENCE

This book provides application architects, developers, and testers with guidance on architecting and engineering applications that meet their customers' and end users' service reliability, availability, quality, and latency expectations. Product managers, program managers, and project managers will also gain deeper insights into the service quality risks and mitigations that must be addressed to assure that an application deployed onto cloud infrastructure consistently meets or exceeds customers' expectations for user service quality.

1.3 ORGANIZATION

The work is organized into three parts: context, analysis, and recommendations.

Part I: Context frames the context of service quality of cloud-based applications via the following:

- “*Application Service Quality*” (Chapter 2). Defines the application service metrics that will be used throughout this work: service availability, service latency, service reliability, service accessibility, service retainability, service throughput, and timestamp accuracy.
- “*Cloud Model*” (Chapter 3). Explains how application deployment on cloud infrastructure differs from traditional application deployment from both a technical and an operational point of view, as well as what new opportunities are presented by rapid elasticity and massive resource pools.
- “*Virtualized Infrastructure Impairments*” (Chapter 4). Explains the infrastructure service impairments that applications running in virtual machines on cloud infrastructure must mitigate to assure acceptable quality of service to end users. The application service impacts of the impairments defined in this chapter will be rigorously considered in Part II: Analysis.

Part II: Analysis methodically considers how application service defined in Chapter 2, “Application Service Quality,” is impacted by the infrastructure impairments

enumerated in Chapter 4, “Virtualized Infrastructure Impairments,” across the following topics:

- “*Application Redundancy and Cloud Computing*” (Chapter 5). Reviews fundamental redundancy architectures (simplex, sequential redundancy, concurrent redundancy, and hybrid concurrent redundancy) and considers their ability to mitigate application service quality impact when confronted with virtualized infrastructure impairments.
- “*Load Distribution and Balancing*” (Chapter 6). Methodically analyzes work load distribution and balancing for applications.
- “*Failure Containment*” (Chapter 7). Considers how virtualization and cloud help shape failure containment strategies for applications.
- “*Capacity Management*” (Chapter 8). Methodically analyzes application service risks related to rapid elasticity and online capacity growth and degrowth.
- “*Release Management*” (Chapter 9). Considers how virtualization and cloud can be leveraged to support release management actions.
- “*End-to-End Considerations*” (Chapter 10). Explains how application service quality impairments accumulate across the end-to-end service delivery path. The chapter also considers service quality implications of deploying applications to smaller cloud data centers that are closer to end users versus deploying to larger, regional cloud data centers that are farther from end users. Disaster recovery and georedundancy are also discussed.

Part III: Recommendations covers the following:

- “*Accountabilities for Service Quality*” (Chapter 11). Explains how cloud deployment profoundly changes traditional accountabilities for service quality and offers guidance for framing accountabilities across the cloud service delivery chain. The chapter also uses the service gap model to review how to connect specification, architecture, implementation, validation, deployment, and monitoring of applications to assure that expectations are met. Service level agreements are also considered.
- “*Service Availability Measurement*” (Chapter 12). Explains how traditional application service availability measurements can be applied to cloud-based application deployments, thereby enabling efficient side-by-side comparisons of service availability performance.
- “*Application Service Quality Requirements*” (Chapter 13). Reviews high level service quality requirements for applications deployed to cloud.
- “*Virtualized Infrastructure Measurement and Management*” (Chapter 14). Reviews strategies for quantitatively measuring virtualized infrastructure impairments on production systems, along with strategies to mitigate the application service quality risks of unacceptable infrastructure performance.

- “*Analysis of Cloud-Based Applications*” (Chapter 15). Presents a suite of analysis techniques to rigorously assess the service quality risks and mitigations of a target application architecture.
- “*Testing Considerations*” (Chapter 16). Considers testing of cloud-based applications to assure that service quality expectations are likely to be met consistently despite inevitable virtualized infrastructure impairments.
- “*Connecting the Dots*” (Chapter 17). Discusses how to apply the recommendations of Part III to both existing and new applications to mitigate the service quality risks introduced in Part I: Basics and analyzed in Part II: Analysis.

As many readers are likely to study sections based on the technical needs of their business and their professional interest rather than strictly following this work’s running order, cross-references are included throughout the work so readers can, say, dive into detailed Part II analysis sections, and follow cross-references back into Part I for basic definitions and follow references forward to Part III for recommendations. A detailed index is included to help readers quickly locate material.

ACKNOWLEDGMENTS

The authors acknowledge the consistent support of Dan Johnson, Annie Lequesne, Sam Samuel, and Lawrence Cowsar that enabled us to complete this work. Expert technical feedback was provided by Mark Clougherty, Roger Maitland, Rich Sohn, John Haller, Dan Eustace, Geeta Chauhan, Karsten Oberle, Kristof Boeynaems, Tony Imperato, and Chuck Salisbury. Data and practical insights were shared by Karen Woest, Srujal Shah, Pete Fales, and many others. Bob Brownlie offered keen insights into service measurements and accountabilities. Expert review and insight on release management for virtualized applications was provided by Bruce Collier. The work benefited greatly from insightful review feedback from Mark Cameron. Iraj Saniee, Katherine Guo, Indra Widjaja, Davide Cherubini, and Karsten Oberle offered keen and substantial insights. The authors gratefully acknowledge the external reviewers who took time to provide through review and thoughtful feedback that materially improved this book: Tim Coote, Steve Woodward, Herbert Ristock, Kim Tracy, and Xuemei Zhang.

The authors welcome feedback on this book; readers may e-mail us at Eric.Bauer@alcatel-lucent.com and Randee.Adams@alcatel-lucent.com.

