INDEX

Note: Italicized page locators refer to figures/tables.

A
Active customers, 5
Active load, 4, 5, 6
Affinity routing, 87-90, 96
and HTTP session caching, 221
IP-based versus user-based, 317
load balancer, 88
proxy server impacting even load distribution with, 89
and uneven cluster loading, 316
Apache
  ApacheBench, 187, 191, 195, 381
  Flood, 187
  Jmeter, 381
  Log4J, 50
  Web Server, 91, 209
Application database transfers, and performance tests, 219
Application footprint, 109, 111
Application serialization, diagnosing and resolving, 301
Application server requirements, projecting, 340-342
Application servers, 36, 96-97, 106, 141
  DMZ configuration for, 224
  and plug-ins, 96
  and security, 97
  transfer of data between application database on network and, 220-221
  and Tuning Settings worksheet, 406
  vertical scaling of, 101
Application server tuning tips, 56-57
Application tuning, and CPU saturation, 311
Asynchronous messaging, 140
Atomic scripts, 168, 181, 197
Authentication, 97, 127
Authorization, 97, 127
Average load, 7, 160

B
Back-end systems
diagnosing, 306
outages, 45, 372
resolving, 307
serialization, 302
Bandwidth
and connection software, 229
for e-Commerce web sites, 219
and proxy servers, 82
and router selection, 80
and test environment, 215
bean.instantiate() method, 54, 370
Bean-managed Entity Beans, 66
Bean-managed Persistence, 66
BMP. See Bean-managed Persistence
Bottleneck Removal checklist, 403, 407
Bottlenecks, 1, 12, 34, 98, 141, 282, 343, 345, 351
  and application implementation, 271
  cluster, 273
  and CPU utilization, 285
  identifying, 208, 263, 284
  and network isolation, 217
  removing, 269-270, 272, 284
  and routers, 80
  and scaling, 34
  and software logs, 296
  and thread traces, 292, 293
  and user ramp-up, 267, 268
Bottleneck symptoms, 299-317
bursty utilization, 304-309
high CPU utilization, 309-314
underutilization, 299-304
uneven cluster loading, 314-316
Browser caching, 99, 155
Browser simulation, and caching, 192-193
B2B (business-to-business) web sites, 129-131
asynchronous interactions, 130
caching potential, 130
performance testing of, 130-131
production simulation requirements for, 189-190
user pressure, 130
"Buckle zone," 13, 21, 409
Buffered writing, 49
Buffer pools, 225
Buffers, 230
Burstiness, 304, 325
and client synchronization, 305
and garbage collection, 307
in TriMont performance test, 326-328
and timeout issues, 308
Bursty behavior, example of, 305
Bursty utilization, 299, 304-305, 317
and Bottleneck Removal checklist, 408
diagnosing, 304
resolving, 305

C
Caching/Caches, 98-99, 106, 141
and browser simulation, 192-193
custom-finder, 69
and Java memory leaks, 115
and request ratios, 146
reverse proxy servers, 83
Caching potential
B2B sites, 130
e-Commerce sites, 131-132
financial sites, 126-127
information sites, 136
pervasive client device support, 137
portal sites, 134
Caching proxy servers, 50, 245, 319, 351
Capacity planning, 335
adjusting, 355
estimates, 156
and horizontal scaling, 102
ongoing, 352-354
and production monitoring, 385
and reviewing performance test results, 337-340
and review plan requirements, 335-337
and scaling assumptions, 345-346
Capacity planning case study, 346-351
Capacity Planning Worksheet, 339, 357, 363-367
Capacity Sizing Worksheet, 240
Case studies
capacity planning, 346-351
preparing to test, 237-254
during the test, 319-333
Centralized control and test tool selection, 198-200
Certificates
client-side, 97
server, 94
Charting and test tool selection
automatic, 205, 212
real-time, 206
Classpath configuration issues, and single-user
testing, 266-267
Cleanup intervals, and EJB containers, 70
Client synchronization, 305-306
Clones, 63, 77-78, 96, 101
and load balancers, 90
performance tips, 78
in vertical and horizontal scaling, 77
Cluster loading, uneven, 299, 314-316, 317
Cluster performance and analysis, 281-283
Clusters/Clustering, 33, 79, 96, 102, 279
Cluster scalability, 33, 347, 348, 365
CMP. See Container-managed Persistence
Code optimization, 35
Code paths
optimization, 25, 27
short, 40, 41
web application, 25
Code-profiling tools, 61, 116, 271
Code reviews, 49, 60
Component checklist, 372-375
Concurrent load, 4, 28
Concurrent users, 5
estimating, 357
and headroom, 336
and TriMont performance test, 239, 324
Configurable limits, and object pools, 76
Connection pools, 43, 73, 119
Container-managed Entity Beans, 66
Container-managed Persistence, 66
Content  
dynamic, 36, 51  
static, 36, 51, 83, 91, 105  
Content-based routing, 86  
Cookies, 58, 155, 196  
hard-coded, 183  
and HTTP sessions, 59  
and pervasive client devices, 137  
and plug-ins, 96  
in test script, 165, 166  
support and user simulation, 193  
CPU utilization, 285-289, 355  
and bottleneck removal, 269-270  
data collection on, 352  
and hardware capacity projections, 342, 343  
high, 309-314  
idle CPUs, 286  
monitoring, 354  
monitoring CPU on test tool, 289  
monitoring CPU on UNIX systems, 286-288  
monitoring CPU on Windows systems, 288  
and performance tests, 232  
and tuning, 271  
and vertical scaling, 275  
CPU utilization estimates, 366-367  
Custom finders, caching bean contents in, 69  
Custom pools, 119  
Custom tags, and JavaBeans, 54-55  

D  
Database connection  
management tips, 72, 73  
and Web Application checklist, 371  
Database performance  
analysis, 384  
logs, 227  
monitoring summary, 384  
Database server, 106, 209  
And CPU utilization, 285  
and Component checklist, 374  
CPU utilization projections, 367  
high wait CPU on, 313  
vertically scaling, 345  
Database server problems, 224-227  
Data collection, 285-298  
DataSource class, 72-73  
Day, 146-147, 159  
Disk  
high I/O wait CPU, 314  
I/O management, 225-227  
and scaling, 34  
DMZ, 75, 81, 82, 84, 100, 105  
DNS, 219, 266, 296  
Dynamic content, 36, 51, 83  
Dynamic decisions, 172-174  
Dynamic page requests, and threads, 91  
Dynamic pages, 39, 105  
Dynamic scripts, 170-177  
Dynamic web pages, parsing, 196-197  

E  
e-Commerce web sites, 83, 131-134  
and building test scenarios, 179-180  
caching potential, 131-132  
network capacity planning, 219-221  
performance testing, 133-134  
production simulation requirements for, 190  
profile for Trimont Mountain Outfitters Web site, 239-240  
security, 132  
traffic patterns, 132  
workload mix for, 197  
ECperf, 161  
EJB containers, 65, 105, 277  
EJBs. See Enterprise JavaBeans  
Embedded elements, 13, 14, 218  
Empirix, 188, 200, 381  
Encryption, 97, 273, 274  
and financial site data, 127  
keys, 87, 94  
Enterprise JavaBeans, 37, 65-72, 105  
cache homes for, 69-70  
marking methods as read-only, 69  
passivation, 69  
performance tips for, 66-70  
wrappers for, 54  
Enterprise JavaBeans containers  
cleanup interval, 70-71  
local and remote interfaces, 71  
pool size, 70  
transaction isolation level, 71  
tuning, 70-72  
Entity Beans, 66, 67, 371  
Error paths, 46
INDEX

Ethernet networks, 219, 220
Exceptions in Java, 76, 206
Extensible Markup Language and JavaServer Pages, 55
Extensible Style Language and JavaServer Pages, 55

F
Façade pattern, 67-68, 371
Failover, 63-65, 96, 102-104
and HTTP session data, 64, 273
hot, 87
File descriptors, 93
File-serving servlets, 50-51
Financial web sites
auditing, 127-128
caching potential, 126-127
performance profiles of, 125-129
production simulation requirements for, 189
and security, 127
traffic patterns of, 127
Firewalls, 81, 100, 105, 106, 234
and burstiness, 309
and load balancing, 277
and uneven cluster loading, 315
Footprint
application, 109, 111
HTTP session, 252
memory, 111, 115, 124
user, 115, 150
Front controllers pattern, 39-40
Frozen web sites, 44, 46, 76, 92, 113, 130, 372
"Funnel effect," threading, 43

G
Garbage collection, 108, 124, 307-308
diagnosing/resolving, 308
and Java Virtual Machine, 107, 111-115
minimizing, 115
reducing cycles of, 47
verbose, 289-291
gifs, 91
and e-Commerce sites, 240
embedded in catalog pages, 83
in HTML pages, 13
and insufficient network capacity, 301
and page rate, 14
and performance tests, 218
Gigabit networks, 221, 222, 249, 376
Graphics
and information sites, 136
and single-user testing, 266

H
"Handshaking," 249, 273, 274
Hardware
and bottlenecks, 28, 29
poor configuration of, 224
projecting capacity of, 324-345
selection by web site teams, 103-104
and test planning, 233-235
for TriMont team, 319-320
and user simulation, 194
Hardware Sizing worksheet, 320-321, 357, 362
Headroom, 153, 336, 345
Heap, 108
and corresponding settings in JVM, 110
JVM and management of, 108-111
settings, 124
tuning settings, 109, 111
Heap size, 60
and garbage collector, 112, 113
monitoring, 289
High utilization, 299, 317
Hits, 13-14, 159
Hits per day estimates, 145-149
determining peak throughput from hits per day,
147-149
Horizontal scaling, 29, 32, 102-103, 106, 272,
345, 346. See also Vertical scaling
clones, 77
cluster performance and analysis, 281-283
example, 21
individual server tests, 279
and load distribution, 89
quantifying load balancing overhead, 280-281
and test environment configuration, 277-283
HTTP. See HyperText Transfer Protocol
HTTP requests, 42
and inbound traffic, 220
one HTML assembled from multiple, 145
and performance tests, 218
HTTP servers, 37, 79, 82, 90-96, 100, 104, 105,
106, 209
and affinity routing, 87
built-in, 75
and high system CPU, 312
keep-alive, 93
logs, 92, 164
and network testing, 234
operating system settings, 93
optimum settings for, 91
plug-ins, 95-96
and processor utilization projections, 342
SSL/HTTPS, 93-95
threads or processes (listeners), 91-92
timeouts, 92
HTTP session data
estimating size of, 221
and performance tests, 219
serialization of, 64-65
HTTP session objects, and Java memory leaks, 115
HTTP sessions, 4, 57-65
and bloat, 60
and cookies, 59
for JavaServer Pages, 53
management, 35
memory leaks, 116
memory usage, 58-63, 252, 361
serializing to streams, 61
sharing, 63-65, 273
size, 59, 63-64
and think time, 191
timeout, 61-62

I
IBM
DB2, 227, 384
Resource Access Control Facility, 97
WebSphere Application Developer
  Workbench, 380
WebSphere Application Server, 48, 53, 59, 61,
  293, 294, 302, 316, 385
WebSphere Application Server Resource
  Analyzer tool, 289, 297
WebSphere MQSeries, 71, 130
WebSphere Performance Benchmark Sample
  (Trade2), 196
Implicit object creation, 124
Inbound traffic, 220, 247
Indexing, 230
and poor database software configuration,
  224-225
Information sites, 135-137
init() method, 41, 42, 46, 48, 57
IP address, 86, 88, 90, 193
IP affinity routing, 282, 316
IP spraying, 85, 86, 232
Isolated networks, 216-217, 235
Iterative test and tuning process, 257

J
Java application server performance, 35-78
  beyond the basics, 57-75
  web application basics, 36-57
  web content types, 36
JavaBeans, 37, 39, 41
  and custom tags, 54-55
  instantiate() method, 54, 370
  and Web Application checklist, 370
Java coding techniques, 117-123
  minimizing object creation, 117-119
  multi-threading issues, 119-121
  synchronization, 121-123
Java monitoring, 289-294
Java Pet Store, 161-163, 171-172, 178
Java Pet Store test scripts
  LoadRunner sample, 391
  SilkPerformer sample, 396
Java profilers, 383
tools, 379-380
JavaScript, 14, 36, 197, 246
  and performance tests, 218
JavaServer Pages, 37, 41, 51-56
  and application flow, 47
  basic runtime operation, 51-52
  performance tips for, 52-54
  precompiling, 22, 53-54
  and use of JavaBeans and custom tags, 54-55
  and XML/XSL, 55-56
Java Virtual Machine, 31, 35, 42, 45, 60, 96, 101,
  106, 107-116, 289, 303
  and clones, 77
CPU utilization, 274, 275
garbage collection, 111-115
heap management, 108-111
  and memory leaks, 115-116
paging, 286
sample heap monitor, 290
and threading, 43
vertical with multiple, 276
jpegs, 91
  and e-Commerce sites, 240
  embedded in catalog pages, 83
  in HTML pages, 13
  and insufficient network capacity, 301
  and performance tests, 218
JSPs. See JavaServer Pages
JTrack jLog, 50
JVM. See Java Virtual Machine
K
  Keep-alive, and HTTP servers, 93
L
  LDAP. See Lightweight Directory Access Protocol
Legacy servers
  applications, 157, 227
  and host database, 229-230
Licenses, 191, 200-202, 238
Lightweight Directory Access Protocol, 97, 219
Linear scaling, 33-34, 103, 156, 278, 339, 345
dynamics, 282
Link names, 176-177
Link numbers, 175-176
Listeners, 91
  and high system CPU, 312
  HTTP, 42, 91-92
  for reverse proxy servers, 83
Load, 1, 3, 21, 263
  active, 4, 5
  average, 7
  bottleneck: resource contention under, 26-29
  concurrent, 4
  estimating, 365
  and headroom, 336
  multi-threading and testing of, 120
  peak, 7, 9, 357
  and response time, 19
  on saturated CPUs, 310
  and user ramp-up, 267
Load balancers, 79, 85-87, 221, 281
  affinity routing, 88
  distributing HTTP traffic, 85
  and horizontal scaling, 102
  tuning, 256
  and uneven cluster loading, 315
Load balancing
  architecture options, 278
  algorithms, 33
  and high user CPU, 311
  and horizontal scaling, 33, 277
  quantifying overhead, 280-281
Load Driver checklist, 387-390
LoadRunner (Mercury Interactive), 165, 196, 382
  example of customizing think times with, 184
  parameter properties GUI, 173
  run-time monitor, 208, 210
  scripts, 166, 173, 179, 391-395
  sample summary report, 204
Log analysis, 163-165
Log4J (Apache), 50
Logging
  framework, 50
  and high wait CPU, 314
  and HTTP I/O servers, 92
  strategy, 49
Logging levels, 49
Long-run testing, 265
M
Master/slave configurations, 231-232
Measurement process, and tuning process, 264
Measurement terminology, 1-23
  load: customers using Web site, 1-9
  response time: time to serve the customer, 15-23
  throughput: customers served over time, 9-15
Memory
  and bottlenecks, 28, 29
  and scaling, 34
Memory footprints, 111, 115, 124
Memory leaks
  in Java, 107, 111, 112, 114
  Java profiler tools for finding, 380
  resolving, 116
  sample verbosegc output with, 291
  typical pattern, 116
Mercury Interactive Corporation, 188, 197, 200
  Astra LoadTest, 381
  LoadRunner, 165, 382
Message-driven Beans, 71
Message-Oriented Middleware, 71-72, 98
Microsoft Web Capacity Analysis Tool, 187-188
Model-View-Controller
  design pattern, 36-38
  within Java web application, 38
INDEX 419

MOM. See Message-Oriented Middleware
Monitoring tools, 270, 288, 379-400
Multi-threading, 107
in Java, 119-12
servlets, 42, 120-121
MVC. See Model-View-Controller

N
"Net quasar," 88
netstat command, 295, 301
Network address management, and user
simulation, 193
Network and performance/stress testing, 216-223
capacity, 218-219, 300-301
components, 221-222
isolation, 216-217
network protocol analyzers and network
monitoring, 222-223
Network Capacity Sizing worksheet, 357, 359-361
Network components, 79-85, 222, 266
firewalls, 81
load balancers, 85-87
network interface cards, 84-85
proxy servers, 82-83
routers, 80-81
Network interface cards, 84-85, 106, 295
and Component checklist, 373
and testing, 232
Network issues, 315-316
diagnosing, 309, 315
isolation, 216-217, 235
resolving, 309, 316
Network latency, 266
Network monitoring, 295-296
Network protocol analyzers, 85, 217, 296,
384-385
cautions, 296
diagnosing network issues with, 300, 309
and network monitoring, 222-223, 295-296
resolving uneven cluster loading with, 316
NICs. See Network interface cards

O
Objects
minimizing, 117-119
pooling, 119
Operating system settings, 93

Optimization terminology, 23-34
bottleneck: resource contention under load,
26-29
path length: steps to service a request, 24-26
scaling: adding resources to improve
performance, 29-34
Outages, 238, 296
back-end, 372
planning for, 46
Outbound traffic, 220, 247
Overutilization, 302

P
Parsing
delimiters, 176
dynamic web pages, 196-197
Passivation, 69
Path length, 1, 24-26, 34
Peak load, 7, 9, 142-144, 159, 220, 251, 354
and EJB pool, 70
estimating, 357
and heap settings, 109
and planning, 335
and response time, 16
Peak throughput, 147-149
Peak usage projections
capacity plans based on, 335
for load and throughput, 357
Performance analysis tools, 379-386
Performance language, 1-34
measurement terminology, 1-23
optimization terminology, 23-34
Performance profiles of web sites, 125-140
B2B (business-to-business), 129-131
e-Commerce, 131-134
financial sites, 125-129
information, 135-137
pervasive client device support, 137-139
portal, 134-135
web services, 140
Performance projections, 340-346
application server requirements projections,
340-342
hardware capacity projections, 342-345
scaling assumptions, 345-346
Performance test planning, 141-160
building performance team, 156-159
scalability testing, 156
  test goals, 142-153
  test scope defined, 153-156
Performance Test Results Worksheet, 403-404
Performance test tool resources, and Test
  Simulation and Tooling checklist, 378
Performance test tools, 381-382
  ApacheBench, 381
  Apache Jmeter, 381
  e-Load, 381
  Mercury Interactive Astra LoadTest, 381
  Mercury Interactive LoadRunner, 382
  QALoad, 381
  Quest Benchmark Factory, 382
  RadView Software WebLOAD, 382
  Rational Suite TestStudio, 382
 Segue SilkPerformer, 382
  PerformaSure (Sitraka), 383
Personal digital assistants, 55, 137
  Pervasive client device support, 137-139
Pet Store. See Java Pet Store, 196
Planning worksheets, 357-367
  capacity sizing, 357-359
  hardware sizing, 362
  JVM memory HTTP session sizing, 361
  network capacity sizing, 359-361
Plug-ins, 42
  and HTTP servers, 95-96
  and vertical scaling, 101
Portal sites, 134-135
  caching potential, 134, 136
  performance testing for, 135
  traffic patterns, 134-135
Precise Indepth for J2EE, 383
Pre-test checklists, 369-378
  component checklist, 372-375
  test environment checklist, 376-377
  test simulation and tooling checklist, 377-378
  test team checklist, 375-376
  Web Application checklist, 369-372
Primitive scripts, 168-170
Processor utilization, See also CPU
  and performance test results, 340
  projecting, 342-343
  vertical scaling, 274
Production monitoring, 385-386
Production web sites
  and back-end legacy systems, 229
  and load balancer affinity routing, 88
  reusing test scripts in, 198
Proxy servers, 82-83, 89-90, 104, 106
  caching, 99
Q
  QALoad (Compuware), 381
Quality assurance, 35
  people, 157-158
Quest, 200
  Benchmark Factory, 382
Real-time server machine test monitoring, 207-212
Reload intervals
  increasing, 57
  for JavaServer Pages, 53
Remote interfaces, 71
remove() method, 69
Repeatability, 202, 217
Reporting, 203-206
  key differentiators for, 203
  key results, 203-205
  multiple runs, 206
  real-time, 205-206
  steady-state measurements, 205
Resource Analyzer (IBM), 297, 298, 302
Response time, 1, 15, 16, 21, 22
  data collection on, 352
  and e-Commerce sites, 131
  and high user CPU, 310
  measurements, 141, 152-153
  monitoring, 354
  and production data analysis, 353, 354
  and single-user paths, 266
  and think time, 17-19
Results verification, 207
Results Verification checklist, 403, 405
Reverse proxy servers, 82, 83
Round-robin
  and load balancers, 86
INDEX 421

and plug-ins, 96
Routers, 80-81, 106, 221, 316, 372
Runaway thread creation, 75, 77
Run consistency, 259-261
Run variation, 259, 260-263

S
Saturation point, 270, 310
and bottleneck removal, 269-270
throughput curve for determining, 268
Scalability, 1
linear, 339
and Stateful Session and Entity Beans, 69
and test environment, 215
testing, 156, 160, 234, 271, 278
Scaling, 29-34, 79
capacity planning, 345-346
horizontal, 31-33, 102-103, 106
issues, 303-304
linear, 33-34
techniques, 24
vertical, 29-31, 100-101, 106
Scenarios
building and weighting, 197
development of, 160
testing, 233
user, 150-151
Script Development Workbench (Compuware), 381
Scripts
atomic, 168, 180-181, 197
and dynamic data support, 195-196
multiple, short, 167
parsing dynamic web pages, 196-197
primitive, 168-170
recording, 189
repetitive, 195
search, 182
and test tool selection, 195-198
Secure Socket Layer (SSL), 87, 93-95, 273
and e-Commerce site, 132
encryption, 95
handshake, 94
TCP/IP, 92
Security, 106
and application servers, 97
for e-Commerce sites, 132
for financial sites, 127, 189
and firewalls, 81
and network protocol analyzers, 223
reverse proxy server used for, 82
and servlet reload, 57
and test environment configurations, 273-274
Segue, 188, 197, 200
SilkPerformer, 165, 174, 382
Serializable interface, 64
Serialization, 65
back-end, 302
for HTTP session data, 64-65
and underutilization, 299
Servers. See also Application servers
application server machines, 223-224
capacity ratio required between, 278
database servers, 224-227
individual testing of, 279, 280
legacy servers, 227-230
saturated, 310
statistics collection from, 289
and vertical scaling, 31
servlet.log() method, 49, 50, 370
Servlets, 37, 39-51
caching JNDI lookups inside (example), 48-49
file-serving, 50-51
and frozen web site danger, 44-46
multi-threaded, 120-121
performance tips for, 46-50
reloads, 57
servlet operation, 41-42
servlet scope, 39-41
threads, queueing and, 42-43
Session Beans, 66
Shared network
impact of, 218
and run-to-run variation, 262
Shareware load drivers, 195
SilkPerformer (Segue), 196, 197, 382
and browser caching, 193
and dynamic selection, 175
scripts, 179
and steady-state capture, 205
SilkPerformer sample script, 396-402
new account script, 400-402
search script, 399-400
sign-in, browse, and purchase script, 396-399
Simple Object Access Protocol (SOAP), 74, 140
Simple-to-complex test strategy, 155-156
Index

Single-threaded models, 42
Single-user paths, 265-267
Single-user testing, 271, 284
Sitraka
   JProbe Suite, 380
   PerformaSure, 383
SMP. See Symmetric multiprocessing
SOAP. See Simple Object Access Protocol
Sockets, 225
Soft exceptions, 76
Solaris, 107, 295
timeout defaults for TCP/IP connections with, 308
Spikes, 111
   and planning estimates, 220
   simulating, 192
SSL. See Secure Socket Layer
Stability tests, 228
Stateful Session Beans, 66, 371
Stateless Session Beans, 66
direct JDBC calls from within, 68
   as façades, 67-68, 371
Static content, 36, 51, 83, 91, 95, 105
caching, 99
   and e-Commerce sites, 131
Static web sites, 188
Steady state, 22-23, 205
   measurement interval, 13, 24
   measuring over long runs and simulating, 259
   on-line store, 22-23
   traditional store, 22
"Sticky routing," 87
StringBuffer class, 118, 372
String class, 26, 117-118, 372
String management, 47, 107
StringTokenizer, 26
String usage, in Java, 117-118
"Striping" database logs, 227
Style sheets, 36
SUT. See System under test
Switched networks, 220, 249
Switches, 221
Symmetric multiprocessing, 31, 269
Synchronization, 107, 120, 121-123, 124, 301
System CPU, 286
System resources, 209, 270
System under test, 194, 289
T
   Table indexing, 224-225
   Tag library, 51, 52
TCP/IP
   address, 92
   and "guaranteed delivery," 220
   packets, 87
Test analysis and tuning process, 256-265
   analyze, 263-264
   test and measure, 257-259
   tune, 264-265
   validate, 259-263
Test clients
capacity, 209
   and CPU utilization, 285
   monitoring, 303
   resource, 303
   and TriMont performance test, 328
Test environment
every, 155
   monitoring components in, 208
   one possible test ordering for large, 235
Test Environment checklist, 376-377
Test environment configurations, 271-283
   adding complexity, 272-273
   core web site features, 273-274
   horizontal scaling, 277-283
   HTTP session sharing, 273
   security, 273-274
   simple, 272
   vertical scaling, 274-277
Test environment construction and tuning,
   215-236
   example test environment, 216
   hardware and test planning, 233-235
   load generators, 230-233
   network, 216-223
   servers, 223-230
Test execution
successful, 255-284
   test analysis and tuning process, 256-265
   test environment configurations, 271-283
   testing overview, 255-256
   test phases, 265-271
Test goals, 142-153
   developing, 141
   peak load, 142-144
   response time measurements, 152-153
INDEX

throughput estimates, 144-152
Test ordering, 235
Test plans, 159
  actual site performance compared to, 354, 355
  and hardware, 233-235
Test scenarios, 165
  building, 179-183
  exercising whole web site, 182-183
  putting scripts together, 180-181
  and weighted testing, 181-182
Test scope definition, 153-156
Test scripts, 161-186, 234
  basics, 167-170
  building test scenarios, 179-183
  common pitfalls with, 183-185
  dynamic, 170-179
  example, 166
  example with parameters, 173
  getting started, 161-166
  myopic, 185
  oversimplified, 185
Pet Store overview, 161-162
random simulation in, 263
and resolving insufficient network capacity, 301
reusing in production, 198
and sample data, 230
and single-user tests, 267
typical, 165-166
user behavior determined, 162-165
Test Simulation and Tooling checklist, 377-378
TestStudio (Rational Suite), 382
Test Team checklist, 375-376
Test tools, 165, 180, 258, 379-402
  CPU monitored on, 289
Load Driver checklist, 387-390
performance analysis and test tool sources, 379-386
Test tool selection, 187-213
  buy versus build, 211-212
  most critical factor in, 202
  production simulation requirements, 188-202
  and reporting capabilities, 203-206
  tools requirements for reproducible results, 202-211
Thick client applications, 59-60, 98, 143, 157
Think times, 17-19, 155, 347
  examples of, 10, 184
  simulations of, 191
  in test scripts, 165
  and TriMont performance test, 323, 327, 328
  unsuitable, 183-184
Thread Analyzer tool, 294
Thread dumps, 270, 292, 293
  and insufficient resources, 302
  and system resources, 270
Thread pools, 75
  absolute maximums set for, 77
  size of and web container, 56
Threads
  and HTTP servers, 91-92
  servlets, queuing and, 42-43
  timeout for, 77
Threadsafe code, 122
Thread safety, and multi-threaded access, 229
Thread stalls, 45, 47
Thread traces, 45, 292-294
  and application serialization, 301
  and application synchronization, 304
  and high user CPU, 311
  and high wait CPU, 313
  sample, 292-293
Throughput, 1, 9, 21, 22, 194, 263, 264
  calculating on Capacity Sizing worksheet, 359
  and cluster performance/analysis, 281, 282, 283
  data collection on, 352
  estimating, 357, 365
  and financial web sites, 189
  increasing, 271, 275-276
  and information sites, 135
  and linear scaling, 33
  maximum, 12
  monitoring, 354
  network interface cards and increase in, 84
  and on-line store, 10-11
  peak usage projections for, 357
  plotting on Summary Test Results graph, 409, 410
  for progressive test runs, 267
  and response time, 19
  reviewing, 335
  saturation point, 19, 21
  and scaling ratio calculations, 341
  and Summary Test Results graph, 338
  and think time, 191
  and traditional store, 9, 10, 19
INDEX

and TriMont performance test, 329, 331
upper bound, 10, 12, 21, 23
Throughput curve, 11-15, 268
hit, 13-14
on-line store, 12-13
page rate, 14
traditional store, 11-12
transaction rate, 14
user rate, 14-15
Throughput estimates, 144-152
Timeouts, 53
diagnosing/resolving issues with, 308
HTTP, 61-62, 92
HTTP session, 61-62, 251
interval changes, 61
TCP/IP, 93
values, 45, 76
top tool, 288
Traffic
and internal networks, 217
between master and slaves, 232
Traffic patterns
and determining user behavior, 162
e-Commerce sites, 132
financial sites, 127
information sites, 136
pervasive client device support, 138
portal sites, 134-135
two examples of, 110
volume, 147
Transaction isolation level, 71
Transaction rate, 14, 239-240
Translation servers, 139
Tuning, 352
and data analysis, 263
and high user CPU, 311
load balancers, 256
parameters, 259
and resolving insufficient resources problems, 302
Tuning adjustments
and CPU utilization measurements, 285
per test iteration, 264
Tuning process and iterative test, 257
Tuning Settings worksheet, 403, 405-407

U
UDDI. See Universal Description, Discovery and
Integration
Underutilization, 299-304, 317, 325
application serialization, 301
and Bottleneck Removal checklist, 407-408
eexample, 300
insufficient network capacity, 300-301
insufficient resources, 302
insufficient test client resource, 303
problem resolution during TriMont
performance test, 330-332
scalability problem, 303-304
Uneven cluster loading, 299, 314, 315, 317
and Bottleneck Removal checklist, 409
network issues, 315-316
routing issues, 316
Universal Description, Discovery and
Integration, 74
UNIX
CPU monitored on, 286-288
and platform requirements, 194
Usage patterns
future, 144
monitoring, 74-75
User CPU, 286
User load
monitoring, 354
and ramp-up, 192
User per day estimates, 149-152
User pressure, 130
User ramp-up, 267-271
User rate, 14-15
Users, 4, 13, 159
behavior determination, 162-165
concurrent, 5
defined, 149
as hits, 149
idealized, 185
User scenarios, 150-151
User simulations, 133, 190-195
ability to control user load and ramp-up, 192
browser simulation and caching, 192-193
cookie support, 193
hardware and platform support, 194
network address management, 193
think time, 191
V
Variance, 284
Variation
  acceptable, 261
  with downward trend, 261
  high run-to-run, 260
  run-to-run, 259
Verbose garbage collection, 109, 289-291
verbosegc, 109, 289, 308
Vertical scaling, 29-31, 33, 100-101, 106, 272, 345. See also
  Horizontal scaling
  of application server, 101
  and clones, 77, 78
  and heap settings, 111
  and load distribution, 89
  with multiple JVMs, 276
  on-line store, 30-31
  and test environment configuration, 274-277
  traditional store, 29-30
Virtual users
  licenses, 139, 238, 323, 325
  and short-term licenses, 201
  and smaller scripts, 167
  and test costs, 190-191
  and TriMont performance test, 324-325, 329, 330
VMGEAR Optimizeit Suite, 380
vmstat, 286, 287
W
Wait, and CPU monitoring on UNIX systems, 286
Wait time, 19
"Warm-up" period, 192
Web application basics, 36-57
  assorted application server tuning tips, 56-57
  JavaServer Pages, 51-56
  Model-View-Controller design pattern, 36-38
  servlets, 39-51
Web Application checklist, 369-372
Web containers
  and Component checklist, 373
  load balancing between EJB containers and, 277
  and security, 97
  thread pool size, 56
WebLOAD (RadView Software), 382
Web services, 74-75, 140
Web site topologies, 79, 99-105
  best practices, 104-105
  and hardware selection, 103-104
  horizontal scaling, 102-103
  vertical scaling, 100-101
Web site types
  performance characteristics compared for, 140
  performance characteristics of, 189
WebSphere Application Developer Workbench (IBM), 380
WebSphere Application Server (IBM), 293, 294, 302, 316, 385
WebSphere Application Server Resource Analyzer tool (IBM), 289, 297
WebSphere MQSeries (IBM), 130
WebSphere Performance Benchmark Sample (Trade2), 168, 196
Weighting
  scenarios, 197
  test scripts, 181-182, 186, 252
Wily Technologies Introscope, 383-384
Wireless Markup Language (WML), 55
"Write once, run anywhere" capability, 107
X
XML. See Extensible Markup Language
XSL. See Extensible Style Language