SQL Server 2008 Query Performance Tuning Distilled

Copyright © 2009 by Grant Fritchey and Sajal Dam

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1902-6

ISBN-13 (electronic): 978-1-4302-1903-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jonathan Gennick
Development Editor: Douglas Pundick
Technical Reviewer: Joseph Sack
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Richard Dal Porto
Copy Editor: Kim Wimpsett
Associate Production Director: Kari Brooks-Copony
Production Editor: Kelly Winquist
Compositor: Patrick Cunningham
Proofreader: April Eddy
Indexer: John Collin
Artist: April Milne
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at `http://www.apress.com`.

Contents

About the Author xix
About the Technical Reviewer xxi
Acknowledgments xxiii
Introduction xxv

CHAPTER 1 **SQL Query Performance Tuning** 1

The Performance-Tuning Process 2
The Core Process 2
Iterating the Process 4
Performance vs. Price 7
Performance Targets 7
“Good Enough” Tuning 7
Performance Baseline 8
Where to Focus Efforts 9
SQL Server Performance Killers 10
Poor Indexing 11
Inaccurate Statistics 11
Excessive Blocking and Deadlocks 11
Non-Set-Based Operations 12
Poor Query Design 12
Poor Database Design 12
Excessive Fragmentation 13
Nonreusable Execution Plans 13
Poor Execution Plans 13
Frequent Recompilation of Execution Plans 14
Improper Use of Cursors 14
Improper Configuration of the Database Log 14
Excessive Use or Improper Configuration of tempdb 14
Summary 15

CHAPTER 2 System Performance Analysis 17

Performance Monitor Tool 17
Dynamic Management Views 19
Hardware Resource Bottlenecks. 20
Identifying Bottlenecks 20
Bottleneck Resolution 21
Memory Bottleneck Analysis 21
SQL Server Memory Management 22
Available Bytes 25
Pages/sec and Page Faults/sec Counters 25
Buffer Cache Hit Ratio. 26
Page Life Expectancy 26
Checkpoint Pages/sec. 27
Lazy writes/sec. 27
Memory Grants Pending 27
Target Server Memory (KB) and Total Server Memory (KB) 27
Memory Bottleneck Resolutions 27
Optimizing Application Workload. 29
Allocating More Memory to SQL Server 29
Increasing System Memory 29
Changing from a 32-bit to a 64-bit Processor 29
Enabling 3GB of Process Space. 30
Using Memory Beyond 4GB Within SQL Server 30
Disk Bottleneck Analysis 32
Disk Counters 32
% Disk Time 33
Current Disk Queue Length 33
Disk Transfers/sec. 34
Disk Bytes/sec 34
Avg. Disk Sec/Read and Avg. Disk Sec/Write 34
Disk Bottleneck Resolutions 35
Optimizing Application Workload. 35
Using a Faster Disk Drive 35
Using a RAID Array. 35
Using a SAN System 37
Aligning Disks Properly 38
Using a Battery-Backed Controller Cache 38
Adding System Memory 38

Creating Multiple Files and Filegroups 39
Placing the Table and Index on Separate Disks 42
Saving Log Files to a Separate Physical Disk 42
Partitioning Tables 43
Processor Bottleneck Analysis. 43
% Processor Time 44
% Privileged Time 44
Processor Queue Length. 44
Context Switches/sec 44
Batch Requests/sec 45
SQL Compilations/sec 45
SQL Recompilations/sec 45
Processor Bottleneck Resolutions. 45
Optimizing Application Workload. 45
Eliminating Excessive Compiles/Recompiles 46
Using More or Faster Processors 46
Using a Large L2/L3 Cache 46
Running More Efficient Controllers/Drivers 47
Not Running Unnecessary Software 47
Network Bottleneck Analysis 47
Bytes Total/sec. 48
% Net Utilization. 48
Network Bottleneck Resolutions 48
Optimizing Application Workload. 48
Adding Network Adapters. 49
Moderating and Avoiding Interruptions 49
SQL Server Overall Performance. 49
Missing Indexes 50
Database Blocking 51
Nonreusable Execution Plans. 52
General Behavior 52
Creating a Baseline 53
Creating a Reusable List of Performance Counters 53
Creating a Counter Log Using the List of Performance Counters. 56
Minimizing Performance Monitor Overhead 57
System Behavior Analysis Against Baseline 59
Summary. 60

CHAPTER 3 SQL Query Performance Analysis 61
The SQL Profiler Tool 61
Profiler Traces 62
Events 63
Data Columns 66
Filters 67
Trace Templates. 69
Trace Data. 69
Trace Automation 70
Capturing a Trace Using the GUI 70
Capturing a Trace Using Stored Procedures 71
Combining Trace and Performance Monitor Output 72
SQL Profiler Recommendations. 73
Limiting the Number of Events and Data Columns 73
Discarding Start Events for Performance Analysis. 74
Limiting the Trace Output Size 75
Avoiding Online Data Column Sorting. 75
Running Profiler Remotely 75
Limiting the Use of Certain Events 75
Query Performance Metrics Without Profiler. 76
Costly Queries 76
Identifying Costly Queries. 77
Identifying Slow-Running Queries. 82
Execution Plans 83
Analyzing a Query Execution Plan. 84
Identifying the Costly Steps in an Execution Plan. 87
Analyzing Index Effectiveness 87
Analyzing Join Effectiveness 89
Actual vs. Estimated Execution Plans. 93
Plan Cache 95
Query Cost 95
Client Statistics. 96
Execution Time. 97
STATISTICS IO. 98
Summary. 100

CHAPTER 4 Index Analysis. 101
What Is an Index? 101
The Benefit of Indexes. 103
Index Overhead. 105

Index Design Recommendations. 107
Examine the WHERE Clause and Join Criteria Columns 107
Use Narrow Indexes. 109
Examine Column Uniqueness. 111
Examine the Column Data Type 114
Consider Column Order. 114
Consider the Type of Index. 117
Clustered Indexes 117
Heap Tables 118
Relationship with Nonclustered Indexes 118
Clustered Index Recommendations 120
Nonclustered Indexes 126
Nonclustered Index Maintenance 127
Defining the Bookmark Lookup 127
Nonclustered Index Recommendations 127
Clustered vs. Nonclustered Indexes 128
Benefits of a Clustered Index over a Nonclustered Index 129
Benefits of a Nonclustered Index over a Clustered Index 131
Advanced Indexing Techniques. 132
Covering Indexes 132
Index Intersections. 135
Index Joins 137
Filtered Indexes 138
Indexed Views. 141
Index Compression 145
Special Index Types. 147
Full-Text. 147
Spatial 148
XML 148
Additional Characteristics of Indexes 148
Different Column Sort Order. 148
Index on Computed Columns 148
Index on BIT Data Type Columns 149
CREATE INDEX Statement Processed As a Query. 149
Parallel Index Creation 149
Online Index Creation 150
Considering the Database Engine Tuning Advisor 150
Summary. 150

CHAPTER 5 **Database Engine Tuning Advisor** 151

Database Engine Tuning Advisor Mechanisms. 151
Database Engine Tuning Advisor Examples 155
Tuning a Query 155
Tuning a Trace Workload 159
Database Engine Tuning Advisor Limitations 161
Summary. 162

CHAPTER 6 **Bookmark Lookup Analysis** 163

Purpose of Bookmark Lookups 163
Drawbacks of Bookmark Lookups 165
Analyzing the Cause of a Bookmark Lookup. 166
Resolving Bookmark Lookups 169
Using a Clustered Index 169
Using a Covering Index 169
Using an Index Join 173
Summary. 174

CHAPTER 7 **Statistics Analysis** 175

The Role of Statistics in Query Optimization. 175
Statistics on an Indexed Column. 176
Benefits of Updated Statistics 177
Drawbacks of Outdated Statistics. 179
Statistics on a Nonindexed Column 180
Benefits of Statistics on a Nonindexed Column 181
Drawback of Missing Statistics on a Nonindexed Column 185
Analyzing Statistics 187
Density. 190
Statistics on a Multicolumn Index. 191
Statistics on a Filtered Index 192
Statistics Maintenance 193
Automatic Maintenance 193
Manual Maintenance. 195
Statistics Maintenance Status 198
Analyzing the Effectiveness of Statistics for a Query 199
Resolving a Missing Statistics Issue. 199
Resolving an Outdated Statistics Issue 202

Recommendations. 204
Backward Compatibility of Statistics 204
Auto Create Statistics 204
Auto Update Statistics. 205
Automatic Update Statistics Asynchronously 207
Amount of Sampling to Collect Statistics. 208
Summary. 208

CHAPTER 8 **Fragmentation Analysis** 209

Causes of Fragmentation 209
Page Split by an UPDATE Statement 212
Page Split by an INSERT Statement 215
Fragmentation Overhead 217
Analyzing the Amount of Fragmentation 220
Analyzing the Fragmentation of a Small Table 222
Fragmentation Resolutions 224
Dropping and Re-creating the Index. 225
Re-creating the Index with the DROP_EXISTING Clause. 225
Executing the ALTER INDEX REBUILD Statement. 226
Executing the ALTER INDEX REORGANIZE Statement 228
Significance of the Fill Factor 230
Automatic Maintenance 233
Summary. 239

CHAPTER 9 **Execution Plan Cache Analysis** 241

Execution Plan Generation 241
Parser. 243
Algebrizer 243
Optimization 244
Execution Plan Caching 251
Components of the Execution Plan. 251
Query Plan. 251
Execution Context 251
Aging of the Execution Plan. 252
Analyzing the Execution Plan Cache. 252
Execution Plan Reuse 253
Ad Hoc Workload 254
Prepared Workload 255
Plan Reusability of an Ad Hoc Workload 256
Plan Reusability of a Prepared Workload. 264

Query Plan Hash and Query Hash 274
Execution Plan Cache Recommendations 278
Explicitly Parameterize Variable Parts of a Query. 278
Create Stored Procedures to Implement Business Functionality 278
Code with sp_executesql to Avoid Stored Procedure Maintenance. 278
Implement the Prepare/Execute Model to Avoid Resending a Query String. 279
Avoid Ad Hoc Queries 279
Prefer sp_executesql over EXECUTE for Dynamic Queries. 279
Parameterize Variable Parts of Queries with Care 280
Do Not Allow Implicit Resolution of Objects in Queries 280
Summary. 281

■CHAPTER 10 **Stored Procedure Recompilation**. 283

Benefits and Drawbacks of Recompilation 283
Identifying the Statement Causing Recompilation 286
Analyzing Causes of Recompilation 288
Schema or Bindings Changes 289
Statistics Changes 289
Deferred Object Resolution. 292
SET Options Changes 294
Execution Plan Aging. 295
Explicit Call to sp_recompile 295
Explicit Use of the RECOMPILE Clause 296
Avoiding Recompilations 298
Do Not Interleave DDL and DML Statements 298
Avoiding Recompilations Caused by Statistics Change. 300
Using Table Variables 302
Avoiding Changing SET Options Within a Stored Procedure. 304
Using OPTIMIZE FOR Query Hint 305
Using Plan Guides 307
Summary. 311

CHAPTER 11 **Query Design Analysis** 313

Query Design Recommendations 313
Operating on Small Result Sets. 314
Limit the Number of Columns in select_list 314
Use Highly Selective WHERE Clauses. 315
Using Indexes Effectively 316
Avoid Nonsargable Search Conditions 316
Avoid Arithmetic Operators on the WHERE Clause Column 320
Avoid Functions on the WHERE Clause Column 322
Avoiding Optimizer Hints 324
JOIN Hint 325
INDEX Hints. 327
Using Domain and Referential Integrity 329
NOT NULL Constraint. 329
Declarative Referential Integrity 332
Avoiding Resource-Intensive Queries. 334
Avoid Data Type Conversion. 335
Use EXISTS over COUNT(*) to Verify Data Existence 337
Use UNION ALL Instead of UNION 338
Use Indexes for Aggregate and Sort Conditions 339
Avoid Local Variables in a Batch Query 339
Be Careful Naming Stored Procedures. 344
Reducing the Number of Network Round-Trips 345
Execute Multiple Queries Together. 346
Use SET NOCOUNT 346
Reducing the Transaction Cost 346
Reduce Logging Overhead 347
Reduce Lock Overhead. 348
Summary. 350

CHAPTER 12 **Blocking Analysis** 351

Blocking Fundamentals 351
Understanding Blocking 352
Atomicity 353
Consistency. 355
Isolation 356
Durability 356

Database Locks 357
Lock Granularity 357
Lock Escalation 362
Lock Modes. 362
Lock Compatibility 369
Isolation Levels. 369
Read Uncommitted 370
Read Committed. 371
Repeatable Read 372
Serializable 375
Snapshot 381
Effect of Indexes on Locking 381
Effect of a Nonclustered Index. 382
Effect of a Clustered Index 384
Effect of Indexes on the Serializable Isolation Level 385
Capturing Blocking Information. 385
Capturing Blocking Information with SQL 386
Profiler Trace and the Blocked Process Report Event 388
Blocking Resolutions. 390
Optimize the Queries. 390
Decrease the Isolation Level 391
Partition the Contended Data. 392
Covering Index on Contended Data. 392
Recommendations to Reduce Blocking 393
Automation to Detect and Collect Blocking Information 394
Summary. 399

CHAPTER 13 Deadlock Analysis 401

Deadlock Fundamentals. 401
Choosing the Deadlock Victim. 402
Using Error Handling to Catch a Deadlock 403
Deadlock Analysis 403
Collecting Deadlock Information 404
Analyzing the Deadlock. 405
Avoiding Deadlocks. 410
Accessing Resources in the Same Chronological Order 410
Decreasing the Number of Resources Accessed 411
Minimizing Lock Contention. 411
Summary. 413

CHAPTER 14 **Cursor Cost Analysis** 415

Cursor Fundamentals 415
    Cursor Location 417
    Cursor Concurrency 418
    Cursor Types 419
Cursor Cost Comparison 422
    Cost Comparison on Cursor Location 422
    Cost Comparison on Cursor Concurrency 424
    Cost Comparison on Cursor Type 426
Default Result Set 428
    Benefits 429
    Drawbacks 429
Analyzing SQL Server Overhead with Cursors 432
    Analyzing SQL Server Overhead with T-SQL Cursors 432
Cursor Recommendations 436
Summary 437

CHAPTER 15 **Database Workload Optimization** 439

Workload Optimization Fundamentals 439
Workload Optimization Steps 440
    Sample Workload 441
Capturing the Workload 444
Analyzing the Workload 446
Identifying the Costliest Query 447
Determining the Baseline Resource Use of the Costliest Query 449
    Overall Resource Use 449
    Detailed Resource Use 449
Analyzing and Optimizing External Factors 452
    Analyzing the Batch-Level Options Used by the Application 452
    Analyzing the Effectiveness of Statistics 453
    Analyzing the Need for Defragmentation 454
Analyzing the Internal Behavior of the Costliest Query 458
    Analyzing the Query Execution Plan 458
    Identifying the Costly Steps in the Execution Plan 459
    Analyzing the Effectiveness of the Processing Strategy 460
Optimizing the Costliest Query 460
    Modifying an Existing Index 461
    Analyzing the Application of a Join Hint 463
    Avoiding the Clustered Index Scan Operation 466
    Modifying the Procedure 466

Analyzing the Effect on Database Workload 469
Iterating Through Optimization Phases 471
Summary. 473

CHAPTER 16 SQL Server Optimization Checklist. 475

Database Design 475
Balancing Under- and Overnormalization 476
Benefiting from Entity-Integrity Constraints 477
Benefiting from Domain and Referential Integrity Constraints. 479
Adopting Index-Design Best Practices 480
Avoiding the Use of the sp_ Prefix for Stored Procedure Names 482
Minimizing the Use of Triggers 482
Query Design 482
Use the Command SET NOCOUNT ON 483
Explicitly Define the Owner of an Object 483
Avoid Nonsargable Search Conditions 483
Avoid Arithmetic Operators on the WHERE Clause Column 484
Avoid Optimizer Hints 485
Stay Away from Nesting Views 485
Ensure No Implicit Data Type Conversions 485
Minimize Logging Overhead. 485
Adopt Best Practices for Reusing Execution Plans. 486
Adopt Best Practices for Database Transactions 487
Eliminate or Reduce the Overhead of Database Cursors 488
Configuration Settings. 488
Affinity Mask. 488
Memory Configuration Options 488
Cost Threshold for Parallelism. 489
Max Degree of Parallelism 489
Optimize for Ad Hoc Workloads. 489
Query Governor Cost Limit 490
Fill Factor (%) 490
Blocked Process Threshold 490
Database File Layout. 490
Database Compression. 491

Database Administration 491

Keep the Statistics Up-to-Date 491

Maintain a Minimum Amount of Index Defragmentation. 492

Cycle the SQL Error Log File. 492

Avoid Automatic Database Functions Such As AUTO_CLOSE or AUTO_SHRINK. 492

Minimize the Overhead of SQL Tracing 493

Database Backup. 493

Incremental and Transaction Log Backup Frequency 493

Backup Distribution 494

Backup Compression. 495

Summary. 495

INDEX 497