



Advanced Querying and Query Tuning in SQL Server 2000 and 2005

Itzik Ben-Gan
Solid Quality Learning
itzik@solidqualitylearning.com

© Solid Quality Learning LLC
www.SolidQualityLearning.com

Overview

- Query Tuning Methodology
- Indexes
- Tools to Analyze Query Performance
- Advanced Querying Techniques
- Procedural Solutions
- Row Versioning and Concurrency

© Solid Quality Learning LLC
www.SolidQualityLearning.com

2

Factors Affecting Query Performance

- Remember the main factors that affect query performance:
 - **Data access involves mainly I/O!**
 - Compilations, computations, parallelism involve CPU
 - Hashing, merge, sorting involve memory
 - Some activities must be spooled (tempdb)
 - More RAM = less physical I/O, more room for data and execution plan buffers

© Solid Quality Learning LLC
www.SolidQualityLearning.com

3

Query Tuning Methodology

- Bad performance = waits
- Top-down approach:
 - Identify waits (bottleneck) at the server level
 - I/O latches? Write log? Blocking? Other?
 - Correlate waits with queues
 - Drill down to the database/file level
 - Drill down to the process level
 - Tune problematic queries

© Solid Quality Learning LLC
www.SolidQualityLearning.com

4

Identify Waits (Bottlenecks) at the Server

- SQL Server 2000:
 - DBCC SQLPERF(WAITSTATS, [clear])
 - Other SQLPERF options: UMSSTATS | IOSTATS | RASTATS | THREADS
 - sysprocesses
- SQL Server 2005:
 - Dynamic management views and functions—e.g., sys.dm_os_wait_stats, to clear: DBCC SQLPERF('sys.dm_os_wait_stats', CLEAR)
- Info on waits is provided in the resources slide

© Solid Quality Learning LLC
www.SolidQualityLearning.com

5

Correlate Waits with Queues

- Perfmon counters:
 - Resource utilization, not at full capacity < 80%-90%
 - Queues (disk, CPU), < 2 per item (spindle/CPU)
 - Cache hit ratio, as high as possible → 90%
 - Page Life Expectancy, as high as possible (300s-1000s)
- To query performance counters:
 - SQL Server 2000: master.dbo.sysperfinfo
 - SQL Server 2005: sys.dm_os_performance_counters

© Solid Quality Learning LLC
www.SolidQualityLearning.com

6

Drill Down to the Database/File Level

- Identify stressed database files
- SQL Server 2000
 - fn_virtualfilestats
- SQL Server 2005
 - sys.dm_io_virtual_file_stats

© Solid Quality Learning LLC
www.SolidQualityLearning.com

7

Drill Down to the Process Level

- Use traces smartly...
- A filter on duration/IO will only isolate individual processes (batch/proc/query)
- More important to calculate aggregates on query "pattern"
- When using stored procedures, easy to identify "pattern"
- When not using stored procedures:
 - Quick and dirty: LEFT(query string, n)
 - Use a parser to identify query pattern

© Solid Quality Learning LLC
www.SolidQualityLearning.com

8

Overview

- Query Tuning Methodology
- **Indexes**
- Tools to Analyze Query Performance
- Advanced Querying Techniques
- Procedural Solutions
- Row Versioning and Concurrency

© Solid Quality Learning LLC
www.SolidQualityLearning.com

9

Indexes

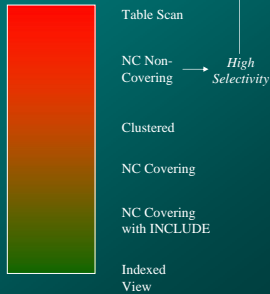
- Effect on Performance
- Index Optimization Scale
- Fragmentation
- Dynamic Management Objects
- Partitioned Tables and Indexes

Effect on Performance

- One of the major factors influencing query performance
 - Effect on: filtering, joins, sorting, group by, blocking and dead-lock avoidance, and more...
 - Effect on modifications: positive effect on locating the rows; negative effect of cost of modification in index
- Understand indexes and their internals
 - Clustered, nonclustered, indexed views and indexes on computed columns, covering scenarios, intersection

Index Optimization Scale

- Cost of access methods:
 - Physical/Logical
 - Random/Sequential
 - 8K/64K



Fragmentation

- Fragmentation: has a significant effect on query performance
 - Logical fragmentation: percent out-of-order pages
 - Page density: page population
- Use DBCC SHOWCONTIG to get fragmentation statistics and examine LogicalFragmentation and Avg. Page Density (full)
- In SQL Server 2005, use the sys.dm_db_index_physical_stats function, and examine AvgFragmentation
- Rebuild indexes to handle fragmentation; can be online in SQL Server 2005 (WITH (ONLINE = ON))

© Solid Quality Learning LLC
www.SolidQualityLearning.com

13

Dynamic Management Objects

- Fragmentation:
 - sys.dm_db_index_physical_stats
- Low-level I/O, locking, latching, and access method activity:
 - sys.dm_db_index_operational_stats
- Index usage / Counts of operations:
 - sys.dm_db_index_usage_stats

© Solid Quality Learning LLC
www.SolidQualityLearning.com

14

Partitioned Tables and Indexes

- Partitioning is virtually unavoidable in large environments
- Improves maintenance:
 - Dealing with fragmentation, backup, etc.
- Used for tuning:
 - Purging/archiving data, loading data
 - More efficient plans, partition elimination, parallel processing, collocated joins
- SQL Server 2000: Partitioned Views
- SQL Server 2005: Native Partitioned Tables, Indexes, and Indexed Views

© Solid Quality Learning LLC
www.SolidQualityLearning.com

15

Overview

- Query Tuning Methodology
- Indexes
- Tools to Analyze Query Performance
- Advanced Querying Techniques
- Procedural Solutions
- Row Versioning and Concurrency

© Solid Quality Learning LLC
www.SolidQualityLearning.com

16

Tools to Analyze Query Performance

- Graphical Execution Plan
- STATISTICS IO: scan count, logical reads, physical reads, read ahead reads
- STATISTICS TIME: duration and net CPU time
- SHOWPLAN_TEXT: estimated plan
- SHOWPLAN_ALL: detailed estimated plan
- STATISTICS PROFILE: detailed actual plan
- SET STATISTICS XML: detailed actual perf info in XML format (2005 only)
- SET SHOWPLAN_XML: detailed estimated perf info in XML format (2005 only)
- Dynamic Management Objects

© Solid Quality Learning LLC
www.SolidQualityLearning.com

17

Query Hints

- Not really hints—rather, forcing SQL Server/optimizer to use a specific option
- Distinguish between performance hints and logical hints
- Performance:
 - Mainly useful for debugging/troubleshooting
 - Refrain from using hints in production

© Solid Quality Learning LLC
www.SolidQualityLearning.com

18

Types of Hints

- Performance
 - Index, join, group, union, parallelism, fast n
- Locking
 - Type, granularity, isolation, readpast
- New in 2005:
 - Use plan, optimize for, recompile query hint, readpast for modifications

Overview

- Query Tuning Methodology
- Indexes
- Tools to Analyze Query Performance
- **Advanced Querying Techniques**
- Procedural Solutions
- Row Versioning and Concurrency

Advanced Querying Techniques

- Set-Based Solutions
- Set-Based Thinking, Tuning Example
- Row Numbers
- Common Table Expressions (CTE)
- Pivoting/Unpivoting
- Auxiliary Tables
- Custom Aggregates

Set-Based Solutions

- Avoid using cursors and loops as first choice
 - Require row-at-a-time processing
- Advantages of using a set-based approach:
 - Allows the optimizer to choose from a wide variety of optimization techniques
 - Usually performs better
 - Requires shorter code, which is easier to manage, maintain, and debug

© Solid Quality Learning LLC
www.SolidQualityLearning.com

22

Set-Based Thinking, Tuning Example

- Given an Employees table with 9 employees and an Orders table with 1,660,000 orders, return active employees (those who made orders) who did not make any orders after April 1998

© Solid Quality Learning LLC
www.SolidQualityLearning.com

23

Cursor-Based Solution (2 minutes)

```
DECLARE @EmployeeID AS INT, @OrderDate AS DATETIME,
        @PrevEmployeeID AS INT, @PrevOrderDate AS DATETIME

DECLARE EmpOrdersCursor CURSOR FAST_FORWARD
FOR SELECT EmployeeID, OrderDate FROM Orders ORDER BY EmployeeID, OrderDate

OPEN EmpOrdersCursor
FETCH NEXT FROM EmpOrdersCursor INTO @EmployeeID, @OrderDate

SELECT @PrevEmployeeID = @EmployeeID, @PrevOrderDate = @OrderDate

WHILE @@fetch_status = 0
BEGIN
    IF @PrevEmployeeID <> @EmployeeID AND @PrevOrderDate < '19980501'
        PRINT @PrevEmployeeID

    SELECT @PrevEmployeeID = @EmployeeID, @PrevOrderDate = @OrderDate

    FETCH NEXT FROM EmpOrdersCursor
    INTO @EmployeeID, @OrderDate
END

IF @PrevOrderDate < '19980501' PRINT @PrevEmployeeID

CLOSE EmpOrdersCursor
DEALLOCATE EmpOrdersCursor
```

www.SolidQualityLearning.com

24

Set-Based Solution (3 seconds)

```
SELECT EmployeeID
FROM Orders
GROUP BY EmployeeID
HAVING MAX(OrderDate) < '19980501'
```

© Solid Quality Learning LLC
www.SolidQualityLearning.com

25

Writing Several Different Solutions

- Always try several different solutions
- Examine execution plans, I/O, and CPU stats
- Good for improving performance and also for practicing different T-SQL techniques
- Result in a better understanding of how the optimizer works, which techniques it utilizes, and where it needs some help

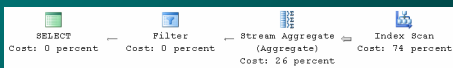
© Solid Quality Learning LLC
www.SolidQualityLearning.com

26

Set-Based Solution 1

```
SELECT EmployeeID
FROM Orders
GROUP BY EmployeeID
HAVING MAX(OrderDate) < '19980501'
```

- 2 seconds
- 4529 logical reads



© Solid Quality Learning LLC
www.SolidQualityLearning.com

27

Row Numbers

- Sequential integers assigned to rows of a resultset
 - Based on desired order (e.g., OrderDate, OrderID)
 - Optionally within partitions (e.g., CustomerID)

RowNum	OrderID	OrderDate	CustomerID	EmployeeID
1	10248	1996-07-04 00:00:00.000	VINET	5
2	10249	1996-07-05 00:00:00.000	TOMSP	6
3	10250	1996-07-08 00:00:00.000	HANAR	4
4	10251	1996-07-08 00:00:00.000	VICTE	3
5	10252	1996-07-09 00:00:00.000	SUPRD	4
6	10253	1996-07-10 00:00:00.000	HANAR	3
7	10254	1996-07-11 00:00:00.000	CHOPS	5
8	10255	1996-07-12 00:00:00.000	RICSU	9
9	10256	1996-07-15 00:00:00.000	WELLI	3
10	10257	1996-07-16 00:00:00.000	HILAA	4

© Solid Quality Learning LLC
www.SolidQualityLearning.com

31

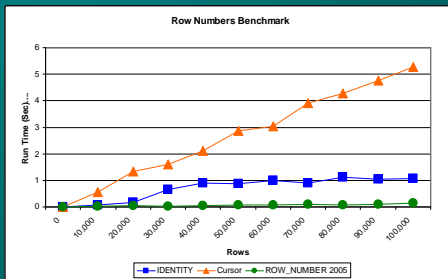
Calculating Row Numbers

- For client manipulation, calculate at client
- For server manipulation:
 - SQL Server 2000:
 - ☹️ • Oops technique: SELECT INTO ... ORDER BY
 - Set-based query
 - Cursor
 - CREATE TABLE with IDENTITY, INSERT SELECT ... ORDER BY
 - SQL Server 2005:
 - Far-out technique: ROW_NUMBER function

© Solid Quality Learning LLC
www.SolidQualityLearning.com

32

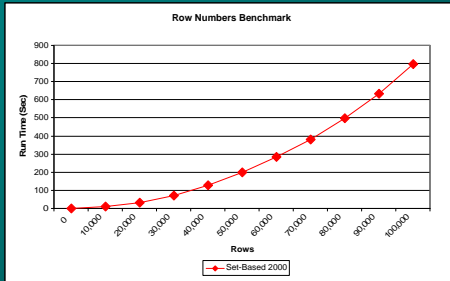
Benchmark



© Solid Quality Learning LLC
www.SolidQualityLearning.com

33

Benchmark



© Solid Quality Learning LLC
www.SolidQualityLearning.com

34

Practical Applications of Row Numbers

- Paging
- Top n for each group
- Matching Current and Adjacent occurrences
- Islands
- Median
- Deleting Duplicates
- And many others...

© Solid Quality Learning LLC
www.SolidQualityLearning.com

35

Common Table Expressions (CTE)

- Support recursion
- Useful for graph/tree/hierarchy manipulation

```
WITH SubsCTE
AS
(
  -- Anchor member returns root node
  SELECT empid, mgrid, empname, salary
  FROM dbo.Employees
  WHERE empid = @root

  UNION ALL

  -- Recursive member returns next level of children
  SELECT C.empid, C.mgrid, C.empname, C.salary
  FROM SubsCTE AS P
  JOIN dbo.Employees AS C
  ON C.mgrid = P.empid
)
SELECT * FROM SubsCTE;
```

www.SolidQualityLearning.com

36

Pivoting

- Rotating data from rows to columns
- SQL Server 2000 techniques:
 - Using series of agg_func(CASE ...) expressions
 - Use matrix table to shorten query string
- SQL Server 2005
 - New PIVOT (and UNPIVOT operators)

© Solid Quality Learning LLC
www.SolidQualityLearning.com

37

Auxiliary Tables

- Helper tables:
 - Auxiliary table of numbers (Nums)
 - Auxiliary table of dates (Calendar)

© Solid Quality Learning LLC
www.SolidQualityLearning.com

38

Custom Aggregates

- Four classes:
 - Cursor
 - Pivoting
 - CLR-Based User Defined Aggregate (UDA)
 - Specialized

© Solid Quality Learning LLC
www.SolidQualityLearning.com

39

Other T-SQL Enhancements

- TOP
 - Supports input expressions and allowed with modifications
- APPLY
 - Can apply right table expression to every row of a left table expressions
- OUTPUT (DML with Results)
 - Modification statements can now return output to: caller, table variable/temp table/real table

© Solid Quality Learning LLC
www.SolidQualityLearning.com

40

Overview

- Query Tuning Methodology
- Indexes
- Tools to Analyze Query Performance
- Advanced Querying Techniques
- **Procedural Solutions**
- Row Versioning and Concurrency

© Solid Quality Learning LLC
www.SolidQualityLearning.com

41

Procedural Solutions

- Following constructs are considered “evil” but have their time and place when used wisely:
 - Cursors
 - Temp Tables
 - Dynamic Execution
- Programmable objects
 - Stored Procedures
 - Triggers
 - Views
 - User Defined Functions

© Solid Quality Learning LLC
www.SolidQualityLearning.com

42

Cursors

- You need to be able to identify scenarios where cursors are suitable/better:
 - Procedural logic / must access row by row
 - Ordered calculations

© Solid Quality Learning LLC
www.SolidQualityLearning.com

43

Temporary Tables

- Useful when:
 - You have intermediate result sets that need to be accessed several times
 - You need a temporary storage area for data while running procedural code
- Use temp tables when:
 - Typically large volumes of data, where plan efficiency is important and non-trivial
- Use table variables when:
 - Typically small volumes of data, where plan efficiency is not as significant as recompilations, or when plans are trivial

© Solid Quality Learning LLC
www.SolidQualityLearning.com

44

Comparing Temp Tables, Table Variables, and Derived Tables

Functionality	Temp Table	Table Variable	Derived Table/CTE
Can refer to multiple times in a query	Yes	Yes	No/Yes
Good for small/large data volumes	Large	Small	Any
Requires tempdb storage	Yes	Yes	No
Recompilation issues	Yes	No	No
Scope	Current and inner levels	Batch	Query
Locking/logging	Yes	Minimal	#NA
Affected by rollback	Yes	No	#NA

© Solid Quality Learning LLC
www.SolidQualityLearning.com

45

Dynamic Execution

- Disadvantages:
 - Ugly code; hard to maintain
 - Requires direct permissions (in 2000)
 - Security risk of SQL Injection
- Use smartly:
 - Dynamic filters and sorting to get good plans
 - Dynamic pivoting/unpivoting
 - And more...

© Solid Quality Learning LLC
www.SolidQualityLearning.com

46

Stored Procedures

- Many logical advantaged
- Performance advantages:
 - Server side
 - Reuse of execution plans
- Note:
 - Requirements for plan reuse
 - Plan reuse is not always a good thing
 - New in SQL Server 2005:
 - OPTIMIZE FOR, RECOMPILE query hints

© Solid Quality Learning LLC
www.SolidQualityLearning.com

47

Triggers

- Expensive (when rollback, undo as opposed to reject)
- Main performance impact involves accessing inserted and deleted views
 - SQL Server 2000: transaction log
 - SQL Server 2005: row versioning (tempdb)
- Stripe transaction log/tempdb when using triggers
- Try to utilize set-based activities
- Identify # of affected rows and react accordingly
- UPDATE triggers record delete followed by insert in the log

© Solid Quality Learning LLC
www.SolidQualityLearning.com

48

Views

- Virtual – little impact on performance unless many levels
- Use indexed views to materialize view's resultset on disk

© Solid Quality Learning LLC
www.SolidQualityLearning.com

49

User-Defined Functions

- Scalar
 - Try to avoid embedding in queries where columns from outer table are provided as arguments
 - Consider using CLR-based functions instead
- Inline Table Valued
 - Treated internally similar to a view
- Multi-Statement Table Valued
 - Keep in mind that you're populating a table variable, with all that implies

© Solid Quality Learning LLC
www.SolidQualityLearning.com

50

Overview

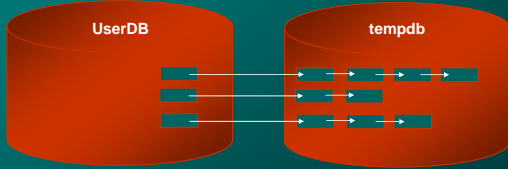
- Query Tuning Methodology
- Indexes
- Tools to Analyze Query Performance
- Advanced Querying Techniques
- Procedural Solutions
- Row Versioning and Concurrency

© Solid Quality Learning LLC
www.SolidQualityLearning.com

51

Row Versioning and Concurrency

- Row Versioning impacts:
 - New Isolation Levels, Triggers, MARS, Online Index Operations
- New Isolation Levels:
 - SNAPSHOT, READ COMMITTED with SNAPSHOT



© Solid Quality Learning LLC
www.SolidQualityLearning.com

52

Resources

- Upcoming Book (Late 06/H1):
 - *Inside SQL Server 2005 – T-SQL Programming*, MSPress
- SQL Server Magazine (www.sqlmag.com)
 - T-SQL Black-Belt Column
 - T-SQL 2005 Web Column
- SQL Server 2005 Transact-SQL Enhancements:
 - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq190/html/sql_05TSQLEnhance.asp

© Solid Quality Learning LLC
www.SolidQualityLearning.com

53

Resources

- Waitstats:
 - Gert Drapers site:
 - <http://www.sqldev.net>
 - http://www.sqldev.net/misc/sp_waitstats.htm
 - http://www.sqldev.net/articles/dbcc_sqlperf.htm
 - <http://www.sqldev.net/articles/WaitTypes.htm>
 - Tom Davidson's article in SQLMag - Opening Microsoft's Performance-Tuning Toolbox:
 - <http://www.windowsitpro.com/Article/ArticleID/40925/40925.html>
 - SQL Server 2005 BOL

© Solid Quality Learning LLC
www.SolidQualityLearning.com

54

Resources

- Troubleshooting Performance Problems in SQL Server 2005
By Sunil Agarwal, Boris Baryshnikov, Tom Davidson, Keith Elmore, Denzil Ribeiro, Juergen Thomas:
 - <http://download.microsoft.com/download/1/3/4/134644fd-05ad-4ee8-8b5a-0aed1c18a31e/TShootPerfProbs.doc>
- Batch Compilation, Recompilation, and Plan Caching Issues in SQL Server 2005 By Arun Marathe:
 - <http://www.microsoft.com/technet/prodtechnol/sql/2005/recomp.mspx>

© Solid Quality Learning LLC
www.SolidQualityLearning.com

55

Advanced Querying and Query Tuning in SQL Server 2000 and 2005

End

Questions?

© Solid Quality Learning LLC
www.SolidQualityLearning.com

56
