



May 11-13 | Orlando, Florida

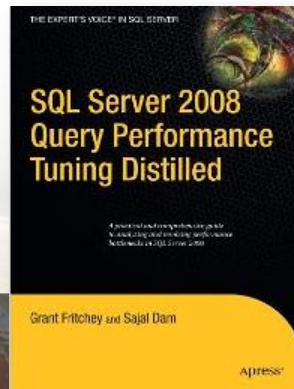
2011

# Query Performance Tuning: Start to Finish

Grant Fritchey

# Who?

- Product Evangelist for Red Gate Software
- Microsoft SQL Server MVP
- PASS Chapter President
- Author:
  - SQL Server Execution Plans
  - SQL Server 2008 Query Performance Tuning Distilled



# Why Tune Queries?

- Most volatile aspect of a database system
- Subject to changes in data
- Affected by changes in structure
- Impacted by poor coding choices
- Victim of resource contention

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Query Performance Tuning – Start to Finish

# GATHERING METRICS

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

# Where to Start Tuning?

- Random
  - Pick a query?
  - Ask a user?
  - Alphabetically?
- Knowledge based
  - Baseline
  - Metrics
  - Records

# Server Metrics

- Start Query Tuning at the Server
  - Hardware
  - Operating System
  - SQL Server
- Establish a Baseline
  - Now is a good time
  - Save the data



# Tools for the Baseline

- Performance Monitor
- Dynamic Management Objects (DMO)
- SQL Data Collection
- Third Party Software

# Where Do Problems Occur?

- Memory
  - Disk I/O
  - Processor
  - Network
- 
- DO NOT SIMPLY TRUST THESE NUMBERS

# Memory

Object	Counter	Values
<b>Memory</b>	Pages/sec	Peaks < 50
	Page Faults/sec	Compare with baseline value for trend analysis
<b>SQL Server: Buffer Manager</b>	Buffer cache hit ratio	Average value $\geq 90$
	Page Life Expectancy	Average value > 300
	Checkpoint Pages/Sec	Peak < 30
<b>SQL Server:Memory Manager</b>	Memory Grants Pending	Peaks = 0

# Disk I/O

Object	Counter	Values
PhysicalDisk	% Disk Time	Peaks < 85%
	Avg. Disk Queue Length*	Peaks < 3 per disk
	Disk Transfers/sec	Maximum value < 100 per disk
	Avg. Disk sec/Read	Compare to baseline
	Avg. Disk sec/Write	Compare to baseline

\* Meaningless on a SAN

# Processor

Object	Counter	Value
Processor	% Processor Time	Peaks < 80%
System	Context Switches/sec	Peaks < 1,000
	Processor Queue Length	Peaks < 3
SQL Statistics	Batch Requests/sec	Compare to Baseline
	SQL Compilations/sec	Peaks > 100
	SQL Recompilations/sec	

# Network

Object	Counter	Value
Network Interface	Bytes Total/sec	Peaks < 50% of NIC capacity
Network Segment	% Net Utilization	Peaks <80% of network bandwidth

# Performance Monitor

- Positives:
  - Absolute Best “Do It Yourself” Option
  - Powerful
  - Accurate
  - Easy to Use
  - Ubiquitous
- Negatives
  - A lot of work to set up, maintain & clean data
  - Very little direction

# Dynamic Management Objects

- Positives
  - Easy to use
  - In a language you know
  - Data is ready to query
  - Ubiquitous
- Negatives
  - Incomplete
  - No direction



# SQL Data Collection

- Positives
  - Immediate results
  - Pre-generated reports
  - Easy to implement
- Negatives
  - Enterprise Only
  - 2008 Only
  - Limited Data Set

# Third Party Tools

- Positives
  - Lots of Direction
  - Ready to consume reports
  - Immediate returns
- Negatives
  - Costly
  - May not collect what you need or want

# Wait Stats & Queues

- `sys.dm_os_wait_stats`
- `sys.dm_exec_requests`
- `sys.dm_waiting_tasks`

# Query Metrics

- This is where you live
- Too much information
- Save the data, just not in its original form
  
- DO NOT USE PROFILER ON PRODUCTION SYSTEMS

# The Server Side Trace

- Profiler to Generate the Script
- Files work best
- Clean and Store the Data
- Profiler GUI can be used to Browse Data
  - Works with Perfmon Data
- Schedule the Start and Stop

# RML Utilities

- Free
- Huge Time Savings
- Excellent Resource
- Still Need Long-Term Storage & Reporting

# Query DMOs

- `sys.dm_exec_requests`
- `sys.dm_exec_query_stats`
- `sys.dm_exec_procedure_stats`

# Metrics Resources

- “SQL Server 2008 Query Performance Tuning Distilled”
- Microsoft White Paper: Performance Tuning Waits and Queues.doc  
<http://technet.microsoft.com/en-us/library/cc966413.aspx>
- Microsoft White Paper: Troubleshooting Performance Problems in SQL Server 2008  
<http://msdn.microsoft.com/en-us/library/dd672789.aspx>
- Brad McGehee
- Louis Davidson &



# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

Query Performance Tuning: Start to Finish

# **OPTIMIZER, STATISTICS, INDEXES & CONSTRAINTS**

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

# Optimizer

- Simply an Amazing piece of Software
- Cost-based
- Not Perfect
- Plan on Helping the Optimizer

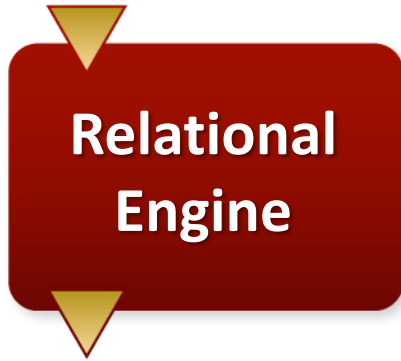
# Relational Engine

QUERY



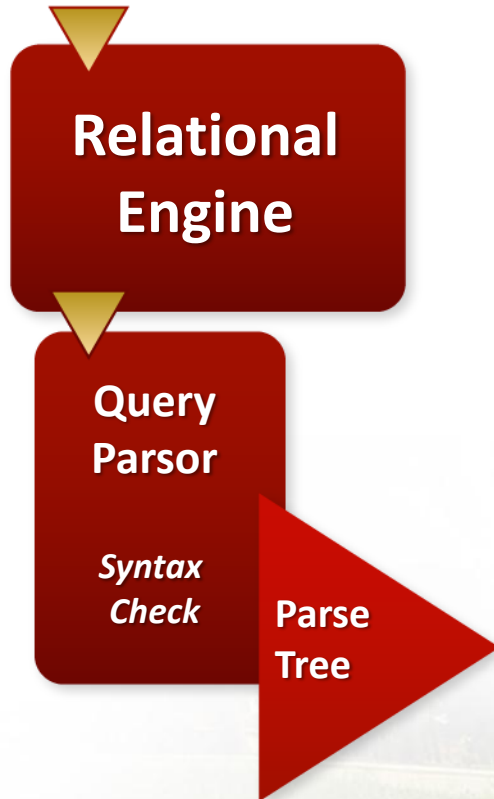
# Relational Engine

QUERY



# Relational Engine

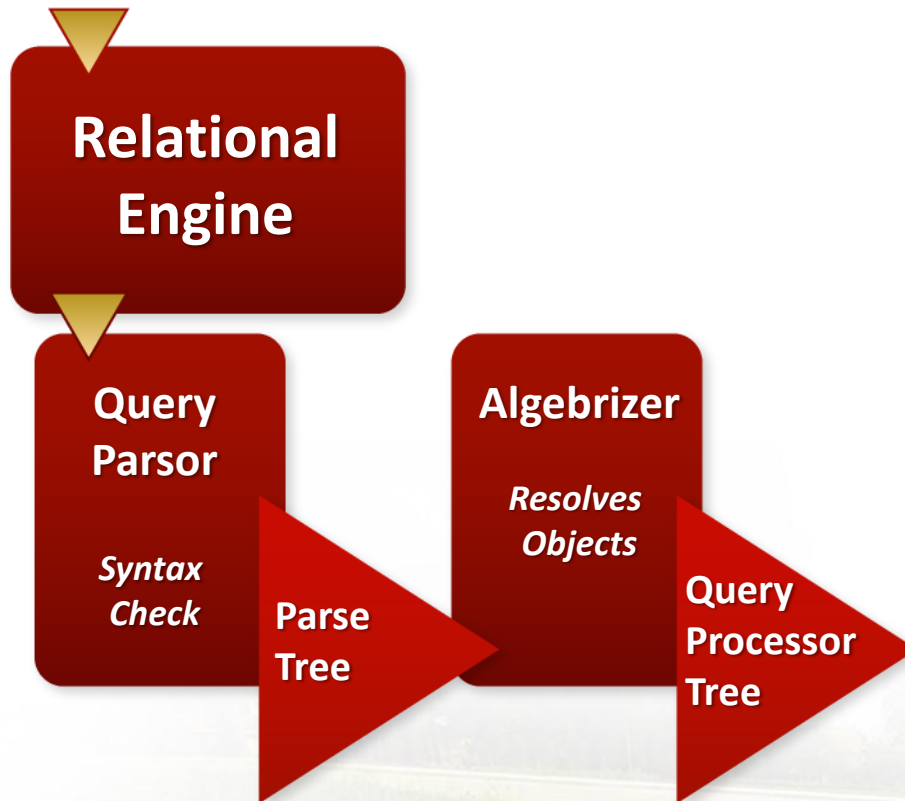
QUERY





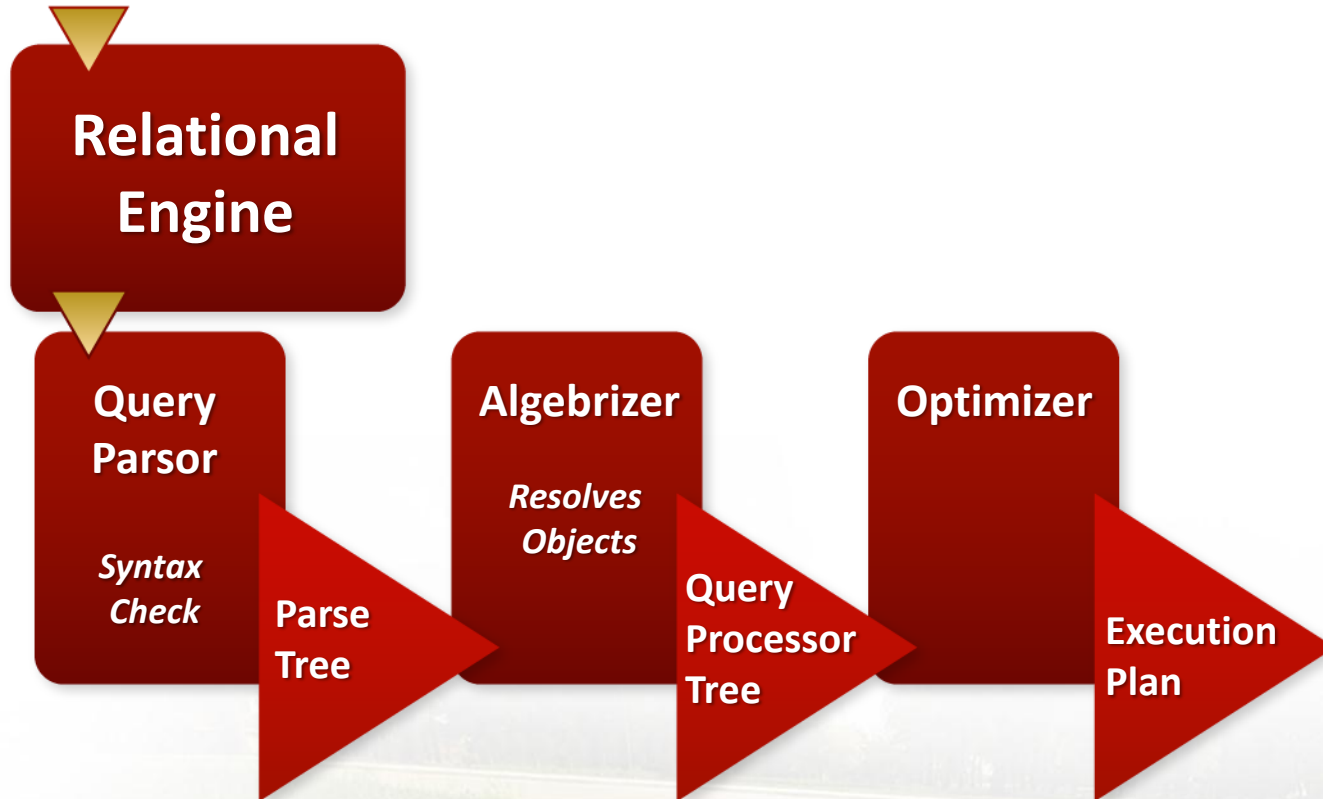
# Relational Engine

QUERY



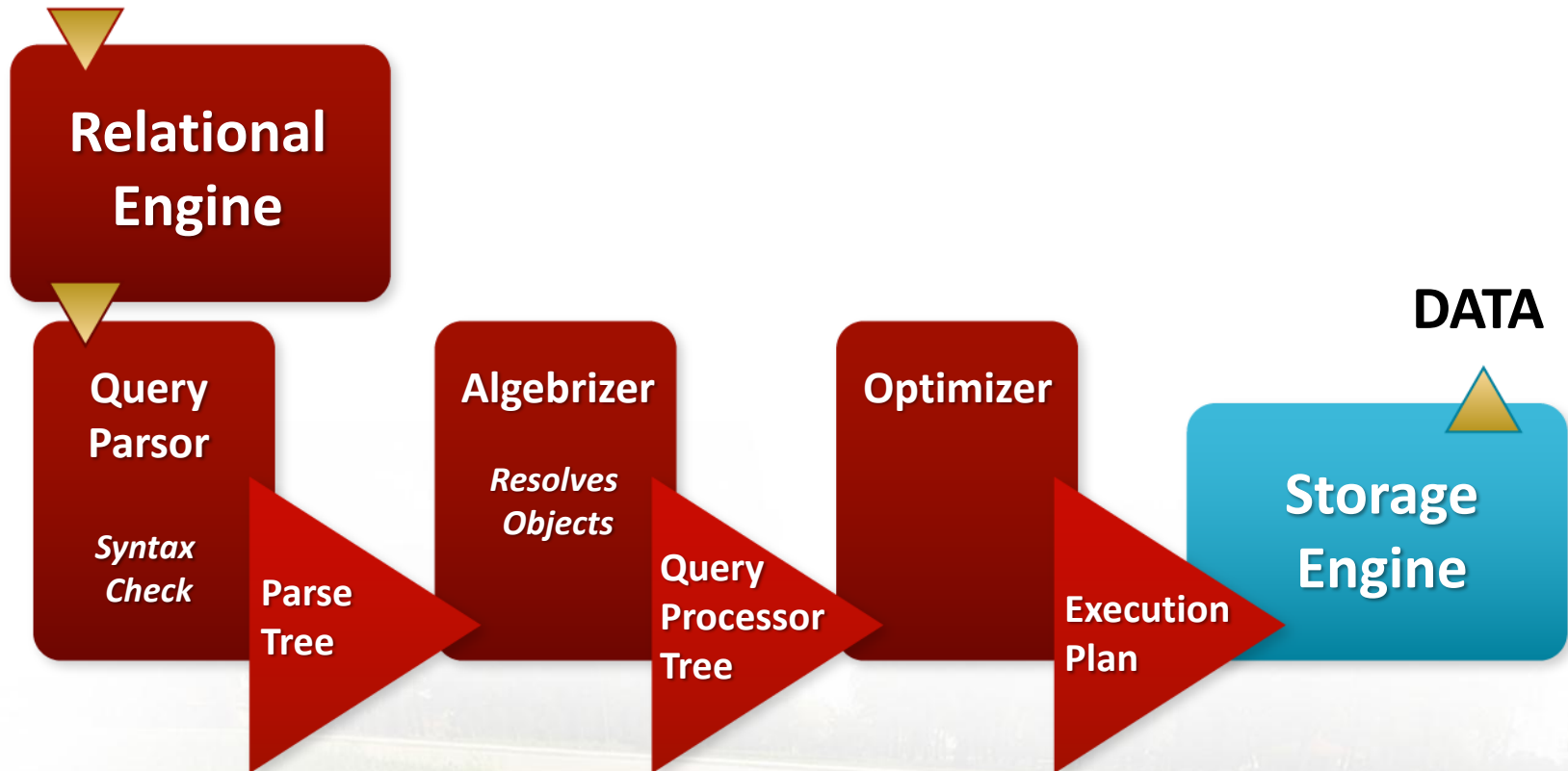
# Relational Engine

QUERY



# Relational Engine

QUERY



# Observing the Optimizer

- `sys.dm_exec_query_optimizer_info`
- Execution Plans

# Statistics

- Information about the Distribution of the Data
  - Created on Index Keys
  - Created on columns
  - Created manually
- Cardinality
- By Default, Created Automatically
- By Default, Maintained Automatically
- Automatic Maintenance Is Not Enough

# Investigating Statistics

- DBCC SHOW\_STATISTICS(*table*, *target*)

– Header

	Name	Updated	Rows	Rows Sampled	Steps	Density	Average key len...	String Index	Filter Expressi...	Unfiltered Rows
1	IX_TransactionHistoryArchive_ProductID	Jan 19 2011 9:57PM	89253	89253	200	0.04100511	8	NO	NULL	89253

– Density

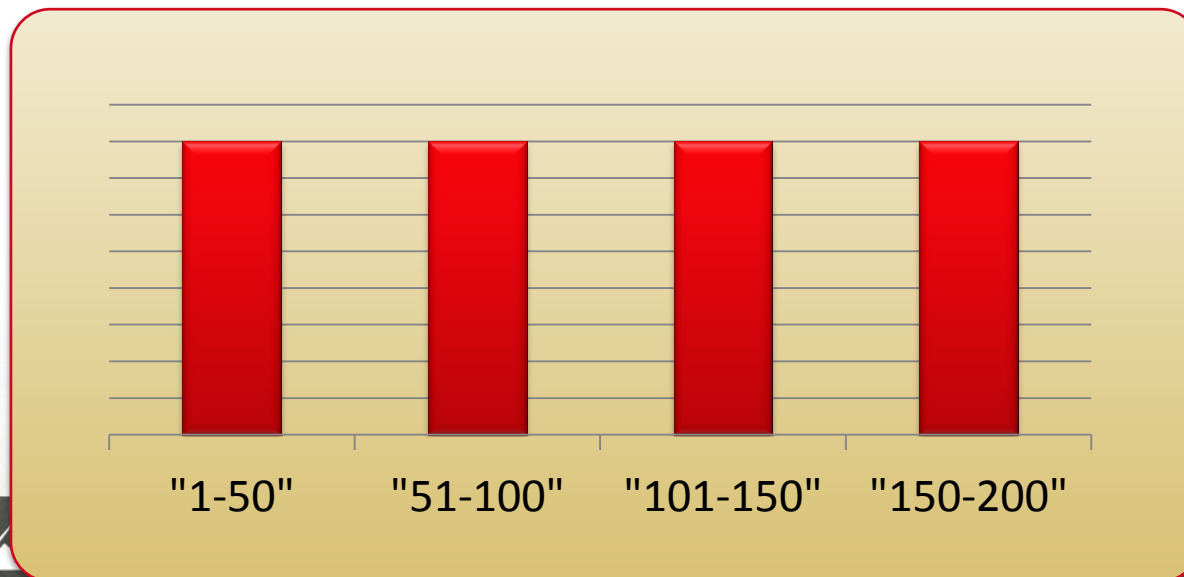
	All density	Average Len...	Columns
1	0.002012072	4	ProductID
2	1.120411E-05	8	ProductID, TransactionID

– Histogram

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1	1	0	6	0	1
2	3	5	786	1	5
3	316	6	786	1	6
4	324	82	786	7	11.71429
5	327	10	786	2	5
6	328	0	619	0	1
7	329	0	781	0	1
8	331	58	786	1	58
9	350	56	786	10	5.6

# Histogram

- 200 steps across the data
- An equal distribution of rows
- Leads to best possible sampling of data
  - But it's not perfect



# Updating Statistics

- sp\_updatestats
  - Can resample
  - Won't run everywhere
- UPDATE STATISTICS X
  - WITH FULLSCAN



# Indexes

- Clustered Index
  - Primary Key Default (but not necessary)
  - Data is stored at the leaf level
  - Data is ordered by the key
- Non-clustered Index
  - Uses cluster key or RID of a heap
  - INCLUDE stored at leaf
- And the rest – outside the scope of this session

# Constraints

- Primary Key
  - Cluster by default, but doesn't have to be
  - Always an index
- Foreign Key
  - No indexes are created with this constraint
- Unique Constraint
  - This constraint is an index

# What's All This Mean?

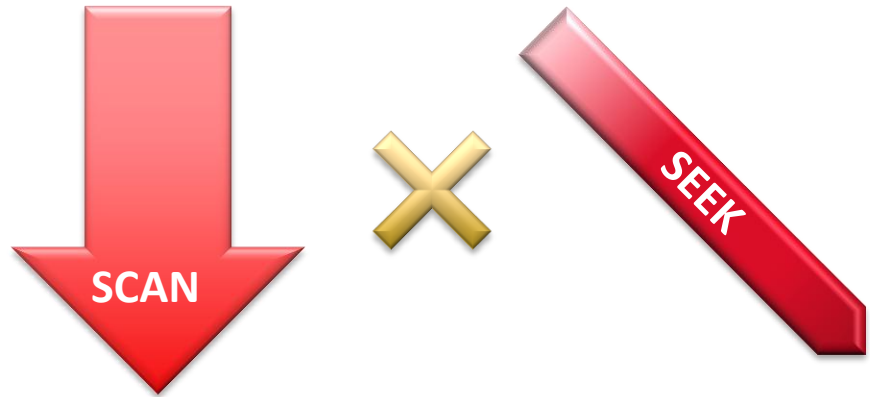
```
SELECT ID FROM TableA WHERE ID = 42
```

Table A			



# What's All This Mean?

```
SELECT a.ID, b.Name, c.Value  
FROM TableA as a  
    JOIN TableB as b  
        On a.ID = B.ID  
    JOIN TableC as c  
        ON b.OtherID = c.OtherID  
WHERE a.ID = 42
```



324 Possible Plans

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

# Optimizer Resources

- Dr. Dewitt's Key Note, PASS Summit 2010  
<http://www.facebook.com/l.php?u=http%3A%2F%2Fwww.slideshare.net%2FGraySystemsLab%2Fpass-summit-2010-keynote-david-dewitt&h=306f5>
- “Inside SQL Server 2008 T-SQL Querying” Itzik Ben-Gan
- “SQL Server 2008 Internals” Kalen Delaney

Query Performance Tuning: Start to Finish

# READING EXECUTION PLANS



# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

# Why Execution Plans

- What will be accessed
- What indexes were used
- What kind of joins were used
- How much did all these operations cost
- Tuning
- Troubleshooting

# Concepts and Architecture

- Relational Engine
  - Estimated Execution Plan
- Storage Engine
  - Actual Execution Plan
- Optimizer
  - Cost-Based
    - Just an estimate
    - Not based on your computer
- Cache
  - Most queries go to cache

# Graphical Plans

- Basic Execution
- Join
- Update
- Delete
- Insert
- Sub-select
- Views

# XML Plans

- Every Graphical Plan is XML
- All cached plans are XML
- Text plans show less information

# Indicators

- Scans (but not always)
- Hash Joins (but not always)
- Work Tables (but not always)
- Extra Operators
- Fat Pipes
- Estimated/Actual Disparity

# Execution Plan Resources

- SQL Server Execution Plans
- Microsoft Whitepapers and Web Sites
  - Statistics used by the Query Optimizer  
<http://www.microsoft.com/technet/prodtechnol/sql/2005/qrystats.mspix>
  - Compilation and Caching  
<http://www.microsoft.com/technet/prodtechnol/sql/2005/recomp.mspix>
  - Showplan Security  
<http://technet.microsoft.com/en-us/library/ms189602.aspx>
  - Understanding Joins  
<http://technet.microsoft.com/en-us/library/ms191426.aspx>
  - Analyzing a Query  
<http://technet.microsoft.com/en-us/library/ms191227.aspx>
  - Database Engine Developer Info Center  
<http://technet.microsoft.com/en-us/library/ms191267.aspx>
  - Database Engine Architect Info Center  
<http://technet.microsoft.com/en-us/library/ms175560.aspx>
  - Forcing Query Plans  
[http://download.microsoft.com/download/4/7/a/47a548b9-249e-484c-abd7-29f31282b04d/Forcing\\_Query\\_Plans.doc](http://download.microsoft.com/download/4/7/a/47a548b9-249e-484c-abd7-29f31282b04d/Forcing_Query_Plans.doc)
- PASS Top 10 Execution Plan Web Sites

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems



Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

Query Performance Tuning: Start to Finish

# DYNAMIC MANAGEMENT OBJECTS

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

# What is a DMOs

- Internal State Data
- Functions and Views
- Security
  - VIEW SERVER STATE
  - VIEW DATABASE STATE

# Currently Running

- Answer Immediate Questions
- Not Cache Dependent
  - But is dependent on execution
- Options:
  - Sys.dm\_exec\_requests
  - Sys.dm\_tran\_active\_transactions
  - Sys.dm\_os\_waiting\_tasks
  - Sys.dm\_db\_index\_operational\_stats

# Recently Running

- Answer Historical Questions
- Aggregate data, not individual calls
- Completely dependent on the cache
- Options:
  - Sys.dm\_exec\_query\_stats
  - Sys.dm\_exec\_procedure\_stats

# Execution Plans

- Estimated Plans Only
- Parameters are unavailable
- Cache Dependent
- Options:
  - Sys.dm\_exec\_cached\_plans
  - Sys.dm\_exec\_query\_plan
  - Sys.dm\_exec\_text\_query\_plan

# Query Text

- Batch Text
- Statement Text
  - Using Offset
- Cache Dependent
- Options:
  - `sys.dm_exec_sql_text`



# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

Query Performance Tuning: Start to Finish

# COMMON PROBLEMS

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

# Common Problems

- Slow Running Query
- Key Lookup
- Parameter Sniffing
- Index Use
- Table Valued User Defined Functions
- Triggers
- Other Ways to Get Them

# Slow Running Query

- Description
  - Most commonly used purpose for execution plans
  - Great tool for observing behavior
  - Lousy tool for comparing performance
  - Drilling down is the key
- Indications
  - Fat pipes
  - Scans
  - Estimated/Actual
  - Extra operations
- Solutions
  - Fix it

# Key Lookup

- Description
  - AKA Bookmark Lookup
  - Not necessarily a problem
- Indications
  - Key Lookup Operator and a Join
- Solutions
  - Change Query
  - Change the index
  - INCLUDE

# Parameter Sniffing

- Description
  - In general, this is a good thing
  - Depends on the data distribution and parameters used
- Indications
  - Intermittent poor performance
  - Disparity on estimated & actual rows
  - Different execution plans at different times
- Solutions
  - OPTIMIZE FOR query hint
  - Use local parameters
  - Last resort – RECOMPILE query hint
  - Last last resort – Plan Guides



# Index Use

- Descriptions
  - Just because you see the index name, doesn't mean it's getting used properly
  - Scans are not necessarily bad
  - Seeks are not necessarily good
- Indications
  - Table Scan
  - Index Scan
  - Extra operators like table spool or sort
- Solutions
  - Create an index
  - Modify an index
  - Modify the query

# Table Valued User Defined Functions

- Description
  - Yes, I see it. It says 0%. It's a lie.
  - “One row is a tragedy; one million is a statistic.”  
Joseph Stalin (sort of)
- Indications
  - Table Scan with a cost of 0%
  - Or Table Valued Function with a cost of 0%
- Solutions
  - When working with more than a few rows... don't use them

# Triggers

- Description
  - Triggers are not immediately visible
  - Estimated plan won't display
  - Slow performance from query that shouldn't be
  - Querying from optimizer...TEST TEST TEST this
- Indications
  - Second plan with the actual plan
  - No hint of it in the estimated plan
- Solutions
  - Be sure the trigger is optimized
  - Avoid where possible

# Individual Statement is Slow

- Large queries or lots of queries
- The exact execution plan you want may be hard to find
- SHOWPLAN\_XML - Estimated
- STATISTICS XML - Actual

# Query is Sometimes Slow

- Intermittent behavior is hard to catch
- Profiler
  - Not the gui
  - Server-side trace
- Even with a server-side trace, capturing execution plans is more expensive (primarily disk space), exercise restraint
  - Data size increase from 2k to 64k for an XML Plan per statement
  - Added overhead for storage and processing

# Query Was Slow Earlier Today

- Knowing that the query is in cache is the key
- Once it's in cache, DMV's are your friend
  - sys.dm\_exec\_cached\_plans
  - sys.dm\_exec\_query\_plan
    - Really large plans won't be stored here
  - sys.dm\_exec\_query\_stats
  - sys.dm\_exec\_plan\_attributes
  - sys.dm\_exec\_sql\_text
  - sys.dm\_exec\_text\_query\_plan
    - Used for really large plans

# Identifying Similar Queries

- Ad hoc systems need hugs/tuning too
  - Identifying similar queries can suggest needed indexes
  - Similar queries could be candidates for procedures
- Multiple stored procedures may have same query
- Query Hash to see similarities in query
- Query Plan Hash to see similarities in query plan

# Working With Large Plans

- Really large plans are hard to read
- Large plans in text
- Large plans in XML
  - In XML, XQuery opens up the plan
- Using XML has other benefits



# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

Query Performance Tuning: Start to Finish

# ADVANCED SOLUTIONS

# Advanced Solutions

- Query Hints
- JOIN Hints
- Table Hints
- Plan Guides
- Plan Forcing

# Hints

- Are you smarter than these guys?
- Have you spent more time working on SQL Server internals than these guys?
- Then why do you think you should take control of the optimizer?

# Query Hints

- Unions
- Joins
- FORCE ORDER
- MAXDOP
- OPTIMIZE FOR
- ROBUST PLAN
- KEEPFIXED PLAN

# Join Hints

- Loop
- Merge
- Hash

# Table Hints

- NOEXPAND
- INDEX()
- FAST N



# Plan Guides

- For Use When You Can't Modify Code
- Three Kinds
  - Object
  - SQL
  - Template
- Applies Hints

# Plan Forcing

- USE PLAN
- As close as you can get to direct control of the Optimizer
- Still can't actually control it
- Absolute Last Ditch Efforts
- Limits:
  - Must be a valid plan
  - No INSERT, UPDATE, DELETE
  - No distributed or full text queries
  - Cursors can only be static or fast\_forward

# Finish Line

- The ability to collect performance metrics on their servers as part of an overall query tuning methodology
- The ability to generate, read, and understand execution plans from multiple sources in support of troubleshooting poorly performing queries
- An understanding of how the optimizer works in support of writing better TSQL code as well as troubleshooting poorly performing queries
- A working knowledge of DMVs that will help them identify and fix performance issues on their servers
- The ability to address common query performance problems

Questions?

How would you...?

What happens when...?

Why does...?

When do I... ?

# Please Complete the Evaluation Form

## Pick up your evaluation form:

- In each presentation room

## Drop off your completed form

- Near the exit of each presentation room
- At the registration area

*Presented by Dell*





May 11-13 | Orlando, Florida

2011

# THANK YOU!

For attending this session and  
PASS SQLRally Orlando, Florida

*Presented by Dell*

