PASS SUMMIT 2012

Query Performance Tuning: Start to Finish

Grant Fritchey

AD-399-P

November 6-9, Seattle, WA

PASS

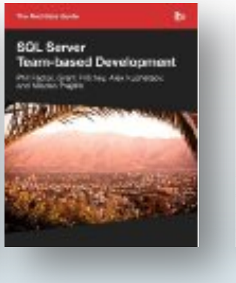
Grant Fritchey

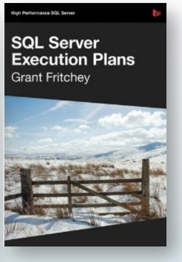
Product Evangelist for Red Gate Software

Email: grant@scarydba.com

Twitter: @gfritchey

Blogs: scarydba.com
simple-talk.com/community/blogs/scary





Why Tune Queries?

Most volatile aspect of a database system

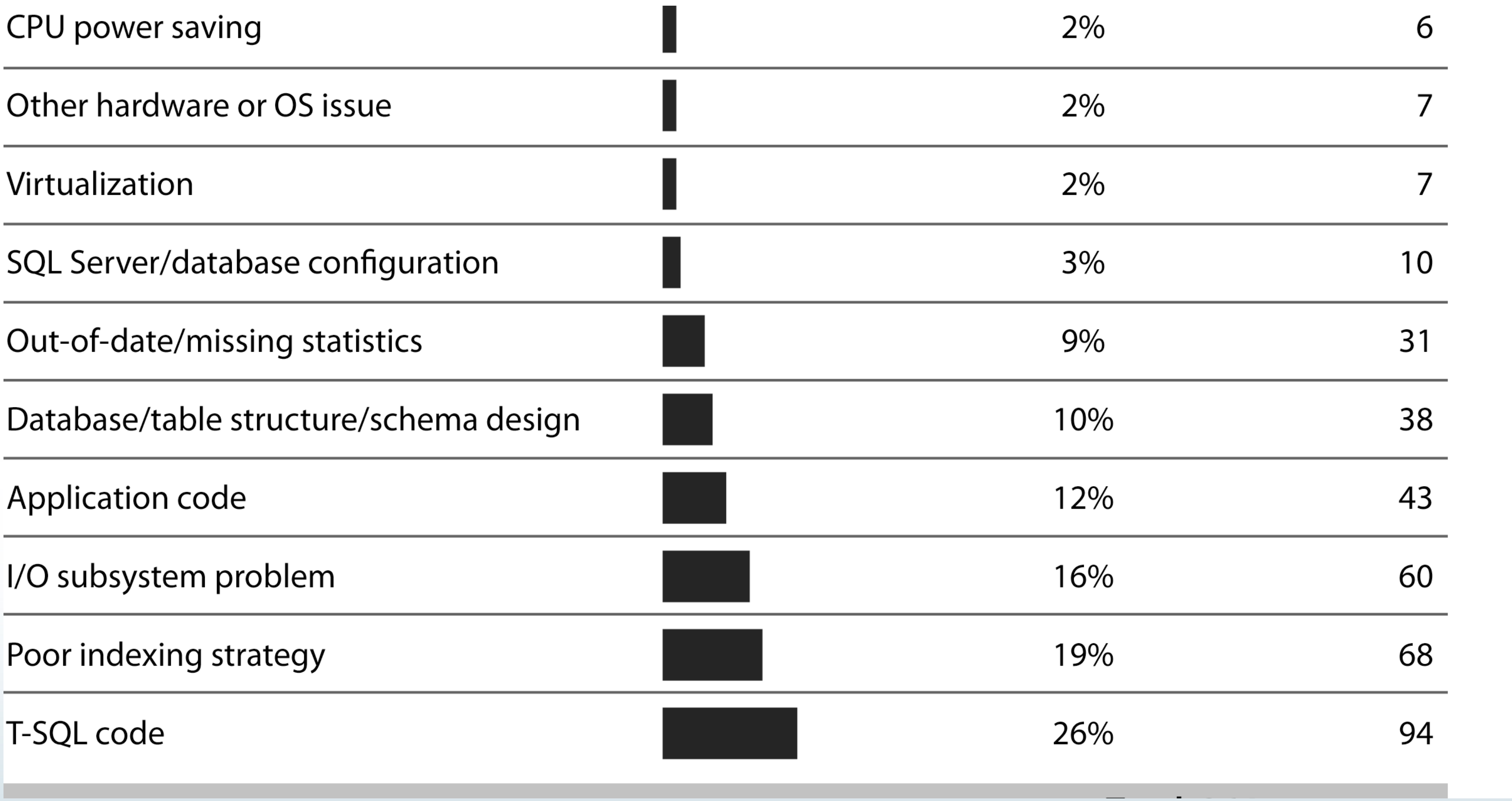
Subject to changes in data

Affected by changes in structure

Impacted by poor coding choices

Victim of resource contention

Why Tune Queries?



CPU power saving	2%	6
Other hardware or OS issue	2%	7
Virtualization	2%	7
SQL Server/database configuration	3%	10
Out-of-date/missing statistics	9%	31
Database/table structure/schema design	10%	38
Application code	12%	43
I/O subsystem problem	16%	60
Poor indexing strategy	19%	68
T-SQL code	26%	94

http://sqlskills.com/blogs/paul/post/survey-results-common-causes-of-performance-problems.aspx

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

Agenda

Gathering Metrics

Optimizer, Statistics, Indexes, Constraints

Reading Execution Plans

Common Problems

Advanced Solutions

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

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 (or better) 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

Where Do Problems Occur?

Memory

Disk I/O

Processor

Network

DO NOT SIMPLY TRUST THESE NUMBERS

Perfmon Memory

Object	Counter	Values
Memory	Pages/sec	Peaks < 50
	Page Faults/sec	Compare with baseline value for trend analysis
SQL Server: Buffer Manager	Page Life Expectancy	Average value > 300, but really compare to baseline
	Checkpoint Pages/Sec	Peak < 30, but must compare to baseline
SQL Server:Memory Manager	Memory Grants Pending	Peaks = 0

Queries Memory

Sys.dm_os_performance_counters

Sys.dm_os_ring_buffers

Sys.dm_os_memory_brokers

DBCC MEMORYSTATUS

Sys.dm_os_wait_stats

Sys.dm_os_memory_clerks

Perfmon Disk I/O

Object	Counter	Values
PhysicalDisk	% Disk Time	Average value < 85%
	Avg. Disk Queue Length*	Peaks < 2 per disk
	Disk Transfers/sec	Compare to baseline & manufacture specs
	Avg. Disk sec/Read	Compare to baseline
	Avg. Disk sec/Write	Compare to baseline

* Meaningless on a SAN

Queries Disk I/O

Sys.dm_io_virtual_file_stats

Sys.dm_os_wait_stats

Perfmon Processor

Object	Counter	Value
Processor	% Processor Time	Peaks < 80%
System	Context Switches/sec	Peaks < 2,000
	Processor Queue Length	Peaks < 2
SQL Statistics	Batch Requests/sec	Compare to Baseline
	SQL Compilations/sec	Compare to Baseline
	SQL Recompilations/sec	

Queries Processor

Sys.dm_os_performance_counters

Sys.dm_os_wait_stats

Sys.dm_os_workers

Sys.dm_os_schedulers

Perfmon Network

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

Query Metrics

This is where you live

Too much information

Save the data, just not in its original form

DO NOT USE PROFILER GUI ON PRODUCTION SYSTEMS

Query Dynamic Management Objects

These are dependent on cache

No run-time information

Uses T-SQL

Mix & Match

DMOs

- Sys.dm_exec_requests
- Sys.dm_exec_query_stats
- Sys.dm_exec_procedure_stats

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

NOTE: RML Utilities are currently not directly supported in SQL Server 2012

Extended Events

Lightweight and low cost

XML Output

Can be left on the server

Work through GUI or T-SQL

Can output to various locations

Metrics Resources

“SQL Server 2012 Query Performance Tuning”

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

Performance Tuning with SQL Server Dynamic Management Views, by Louis Davidson and Tim Ford

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

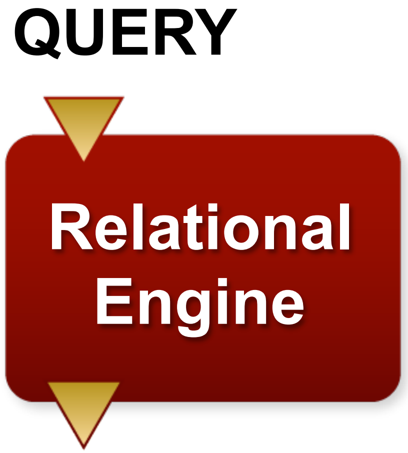
Plan on Helping the Optimizer

Relational Engine

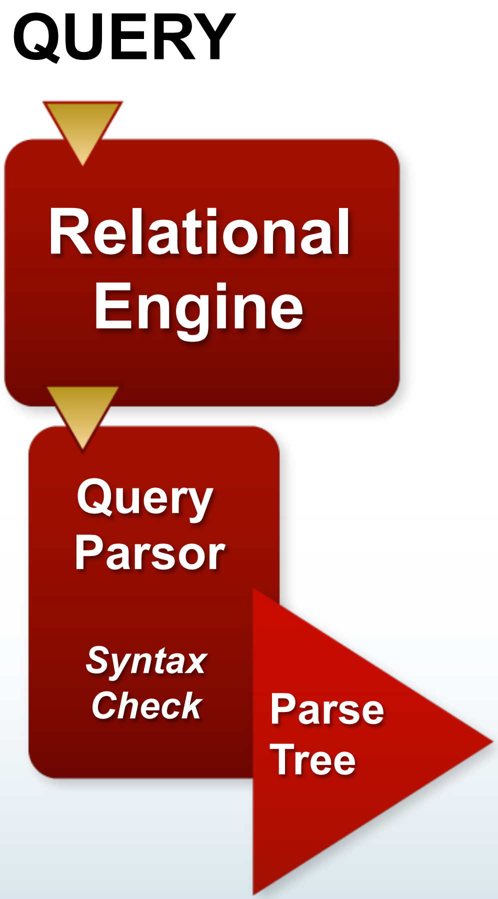
QUERY

▼

Relational Engine



Relational Engine



QUERY

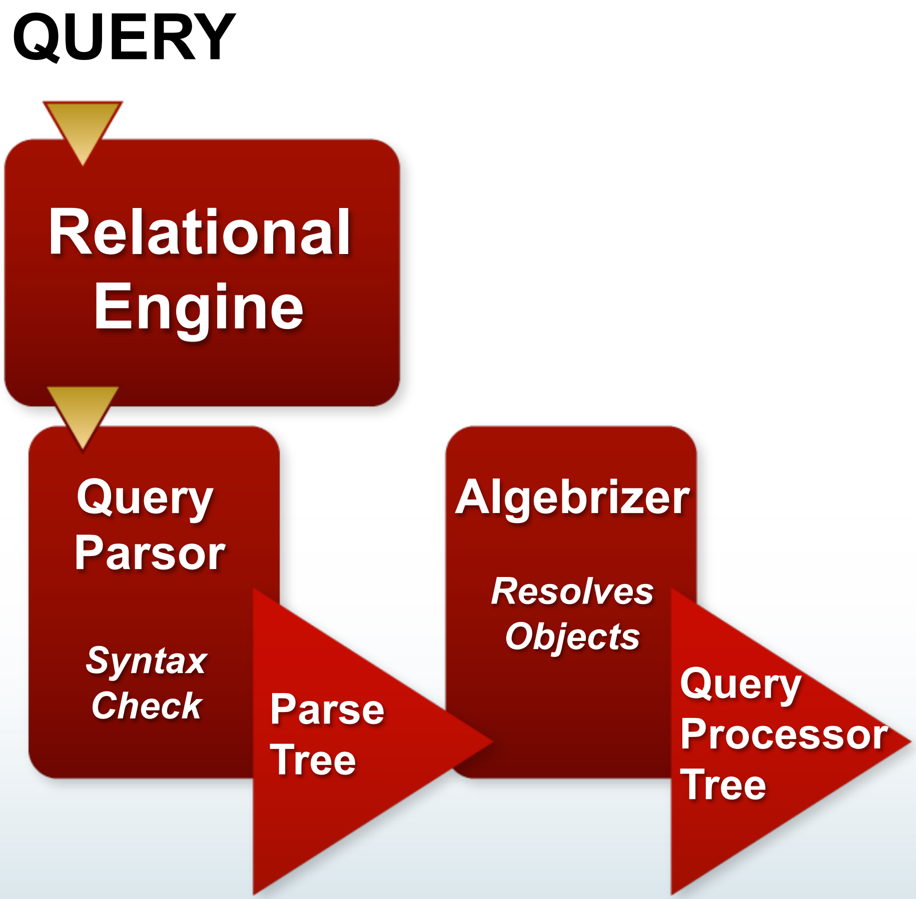
Relational Engine

Query Parsor

Syntax Check

Parse Tree

Relational Engine



QUERY

Relational Engine

Query Parsor

Syntax Check

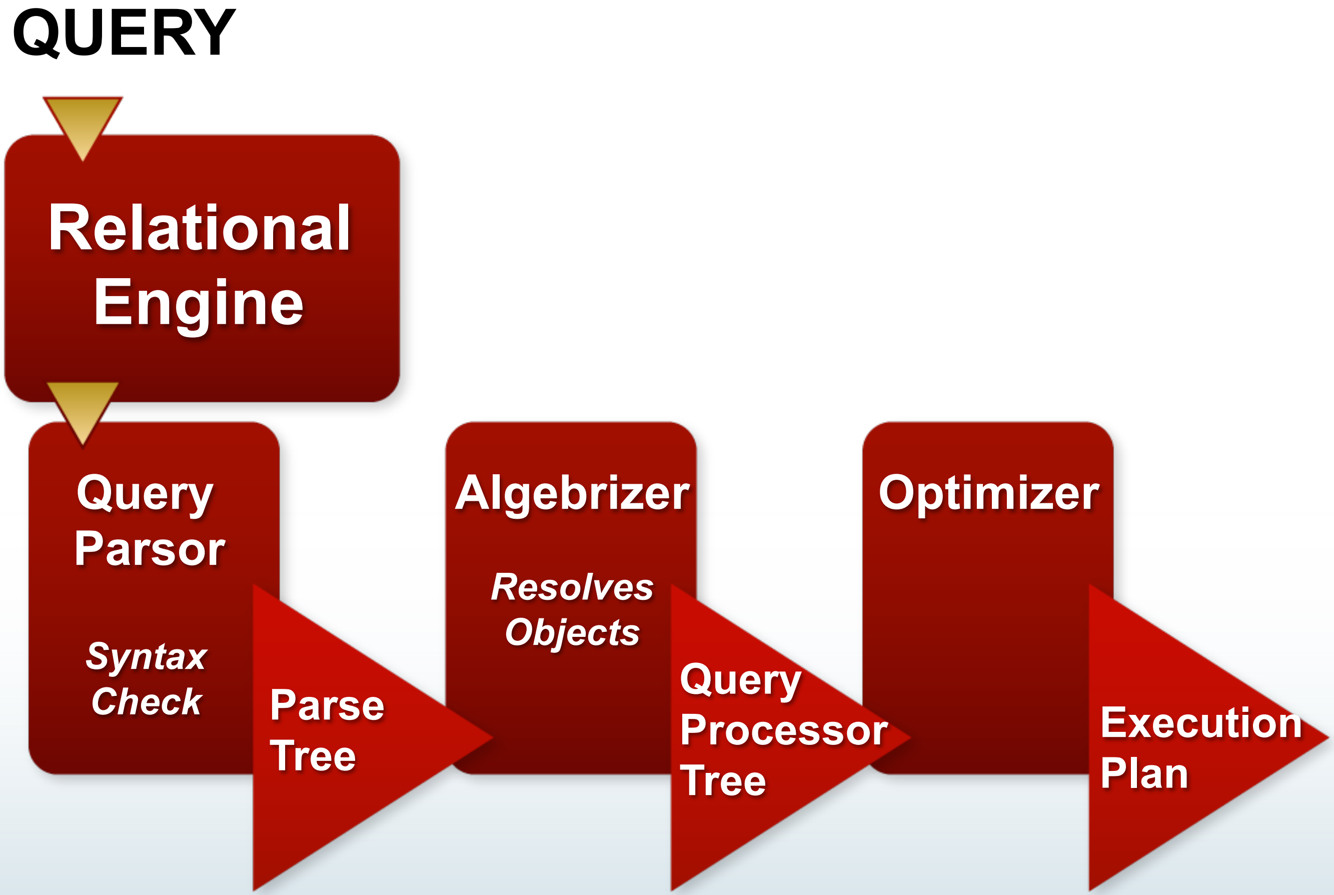
Parse Tree

Algebrizer

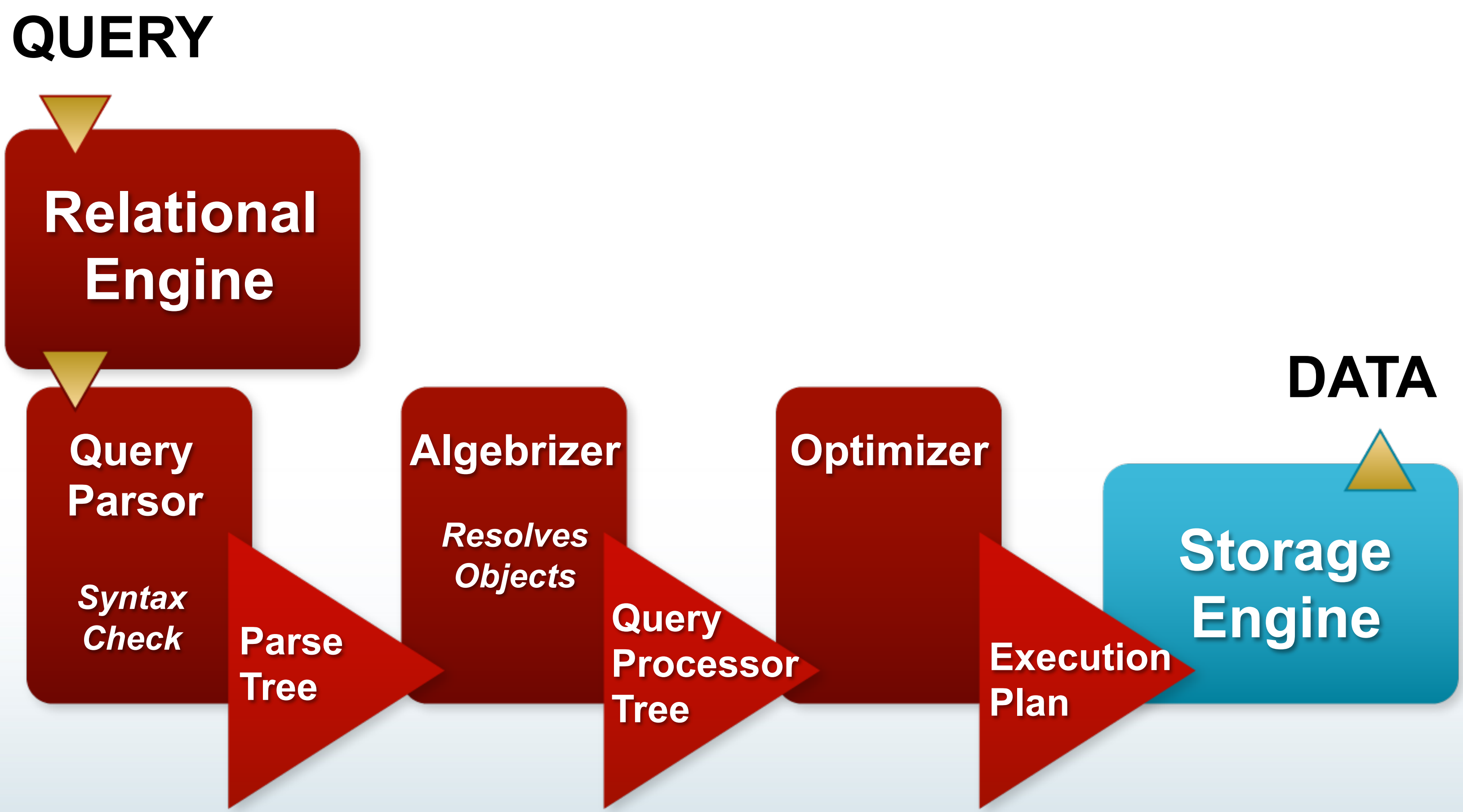
Resolves Objects

Query Processor Tree

Relational Engine



Relational Engine



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

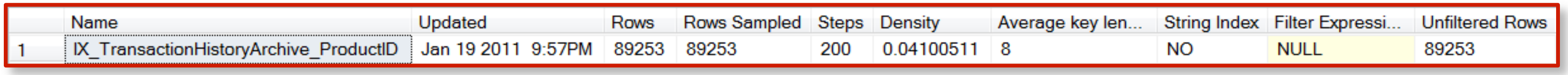
By Default, Maintained Automatically

Automatic Maintenance Is Not Enough

Investigating Statistics

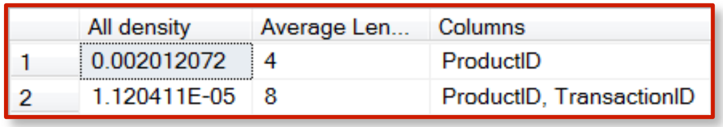
DBCC SHOW_STATISTICS(*table*, *target*)

- Header



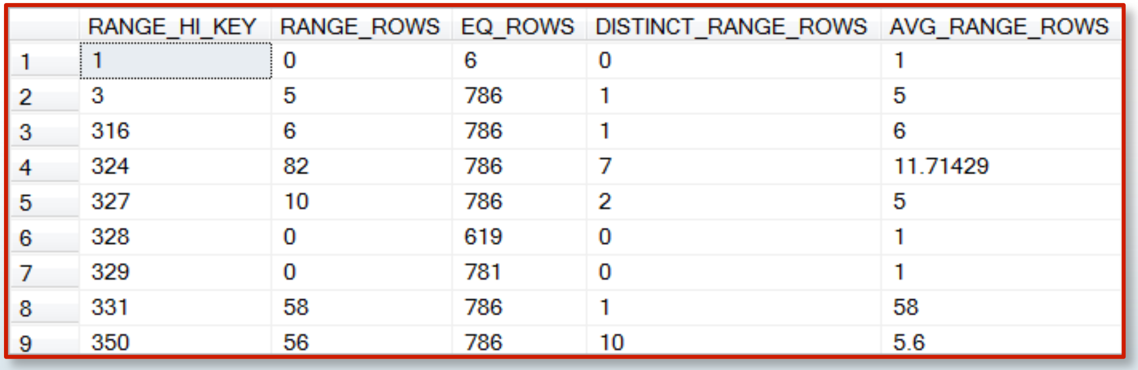
	Name	Updated	Rows	Rows Sampled	Steps	Density	Average key len...	String Index	Filter Expressi...	Unfiltered Rows
1	IX_TransactionHistoryArchive_ProductID	Jan 19 2011 9:57PM	89253	89253	200	0.04100511	8	NO	NULL	89253

- 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

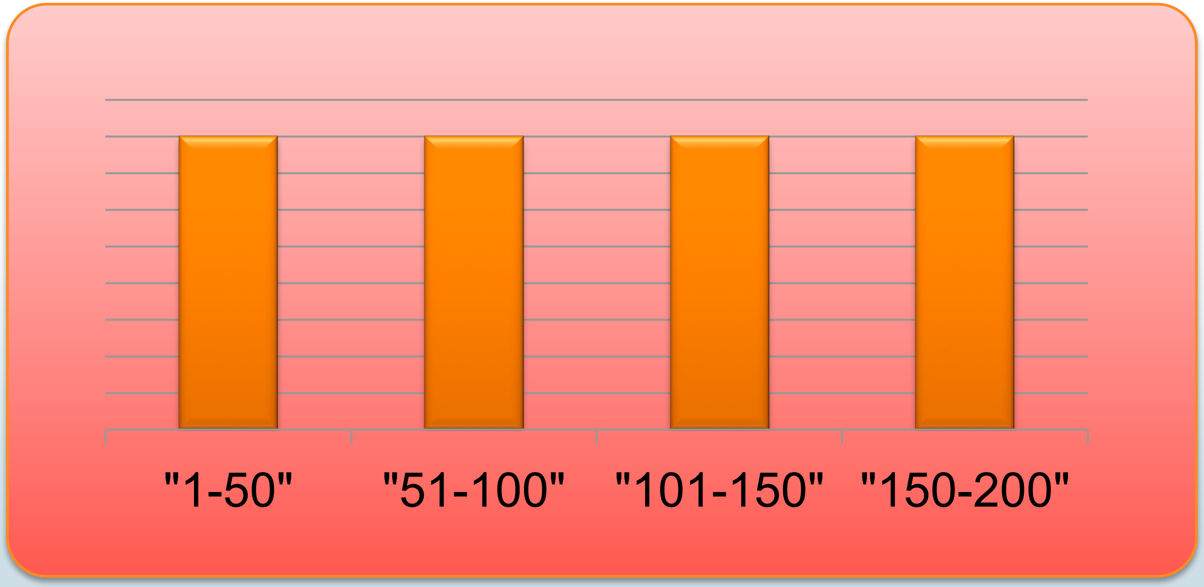
Histogram

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

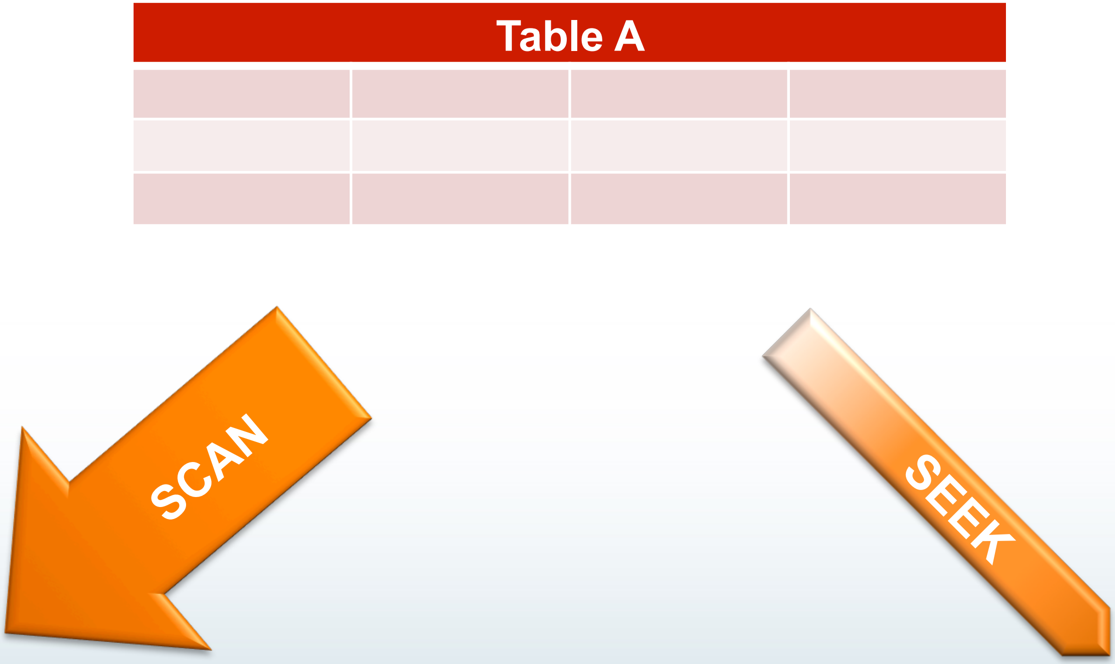
- No indexes are created with this constraint
- Be sure you use WITH CHECK

Unique Constraint

- This constraint is an index

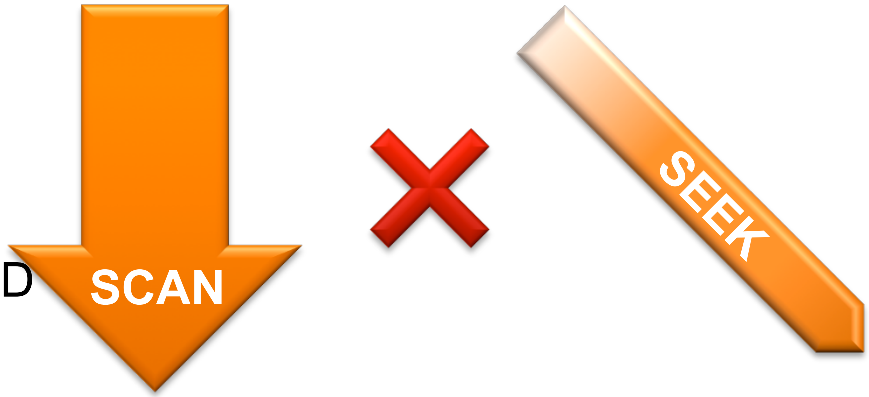
What’s All This Mean?

SELECT ID FROM TableA WHERE ID = 42



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
```



LOOP
HASH
MERGE

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 2012 Internals” Kalen Delaney

“Inside the SQL Server Optimizer” Benjamin Nevarez

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

What To Look For

First Operator

Warnings

Most Costly Operations

Fat Pipes

Extra Operations

Scans

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

What To Look For

First Operator

Warnings

Most Costly Operations

Fat Pipes

Extra Operations

Scans

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.mspx
- Compilation and Caching
  http://www.microsoft.com/technet/prodtechnol/sql/2005/recomp.mspx
- 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

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

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

- Slow running query
- Expensive to run query
- The query the boss notices

Indications

- The query is slow

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

Bad Parameter Sniffing

Description

- In general, parameter sniffing 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
- Seriously don’t go there last resort – turn parameter sniffing off

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

Azure

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

Azure

The same

Except where it's different

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… ?

Grant Fritchey

Product Evangelist for Red Gate Software

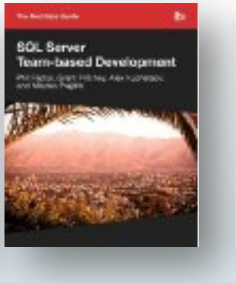
Email: grant@scarydba.com

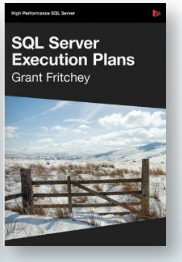
Twitter: @gfritchey

Blogs: scarydba.com

simple-talk.com/community/blogs/

scary





PASS Resources

Free SQL Server and BI training

Free 1-day Training Events

Regional Event

Local and Virtual User Groups

Free Online Technical Training

This is Community

Learning Center

Thank you

for attending this session and
the 2012 PASS Summit in Seattle