Microsoft

CSS Academy

SQL Server 2008 Performance New Features

Ying Fan

Support Engineer

Customer Service & Support

Asia Pacific & Greater China Region

Welcome!

Microsoft

CSS Academy

Agenda

- New Relational Engine Features
- New Performance Diagnostics and Tools
- Techniques for Controlling Query Performance
- Introduction to Resource Governor

Microsoft

CSS Academy

New Relational Engine Features

Declare with Initialize

- The DECLARE statement has been extended to allow specifying an initial value along with the declaration.
- Example:
  - declare @var varchar(30) = 'Hello'

Compound Assignment Operators

- Perform arithmetic operation and assignment

+=	Add the two values and assign
-=	Subtract the value and assign
*=	Multiply the values and assign
/=	Divide by the value and assign
%=	Modulo and assign
&=	Bitwise AND and assign
^=	Bitwise XOR and assign
|=	Bitwise OR and assign

Row Constructors

- **Support specify/manipulate multiple rows**
- **INSERTs only**

```
CREATE TABLE t (c1 int, c2 int)
go
INSERT t (c1, c2) VALUES
(1, 100),
(2, 200),
(3, 300)
```

MERGE Statement

- New relational operator that can do INSERT, UPDATE and/or DELETEs against a target table depending on whether a corresponding match is found in the source
- You must specify at least one action (INSERT, UPDATE or DELETE) but you can specify any combination of the three

MERGE

- Join phase to qualify matches/non-matches
  - INNER, OUTER or CROSS JOIN depending on clauses present
- Computes action based on clauses
  - Final action is insert/update/delete
  - Normal I/U/D triggers, cascading constraint actions apply

Microsoft

CSS Academy

MERGE - Example

```
MERGE dbo.FactBuyingHabits h
    USING dbo.Purchases p on h.CustomerID = p.CustomerID
and h.ProductID = p.ProductID
WHEN MATCHED
    THEN UPDATE SET h.LastPurchaseDate = p.PurchaseDate
WHEN NOT MATCHED BY TARGET
    THEN INSERT (ProductID, CustomerID, LastPurchaseDate)
VALUES (p.ProductID, p.CustomerID, p.PurchaseDate)
OUTPUT $action, inserted.*, deleted.*;
```

Filtered Index

- Filter condition specified on CREATE INDEX to index specific subset of rows (nonclustered index only)
  - Reduced storage/index maintenance costs
- Example:

```
USE AdventureWorks;
GO
CREATE NONCLUSTERED INDEX FIBillOfMaterialsWithEndDate
ON Production.BillOfMaterials (ComponentID, StartDate)
WHERE EndDate IS NOT NULL;
```

Filtered Statistics

- Statistics that apply to a filtered index or created by CREATE STATISTICS with WHERE clause
- Never auto-created except as a result of a filtered index creation
- Example:

```
USE AdventureWorks;
GO
CREATE STATISTICS ContactPromotion1
   ON Person.Contact (ContactID, EmailAddress, EmailPromotion)
WHERE EmailPromotion = 2
WITH FULLSCAN
```

Microsoft

CSS Academy

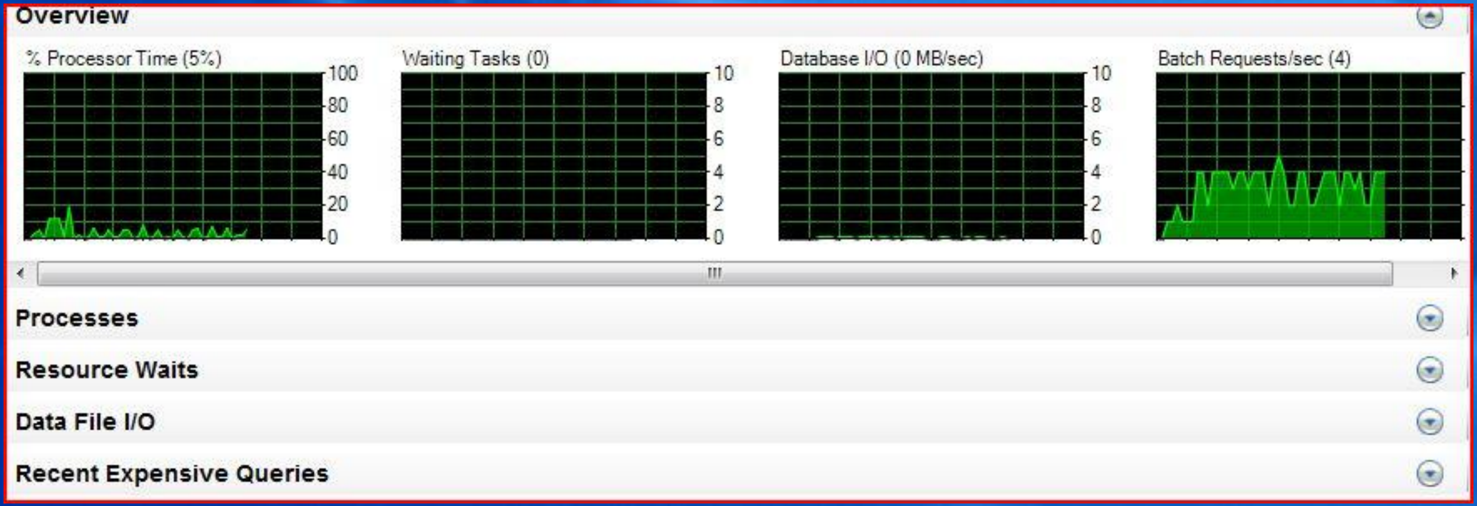
Performance Diagnostics and Tools

Microsoft

Activity Monitor

CSS Academy

- Provides insight into *current/recent* activity by taking delta of two snapshots
- Five different sections available
  - Only issues queries for expanded section(s)
  - User configurable refresh interval
- Can replace Performance Dashboard for initial look at symptom

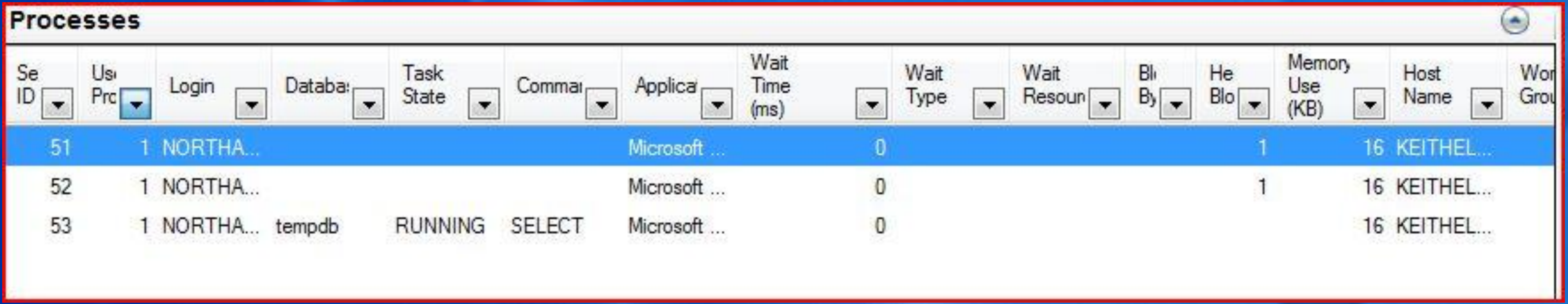


Overview

- %Processor Time for this SQL instance
  - Performance Counter queried via WMI
  - Requires Windows permission to view performance counter
- Percentage of Waiting Tasks
- Database IO Rate
  - Physical IO rate based on sys.dm_io_virtual_file_stats
- Batch Requests/sec
  - From sys.dm_os_performance_counters

Processes

- Basic information from sys.dm_exec_sessions and sys.dm_exec_requests
- Configurable filtering/sorting

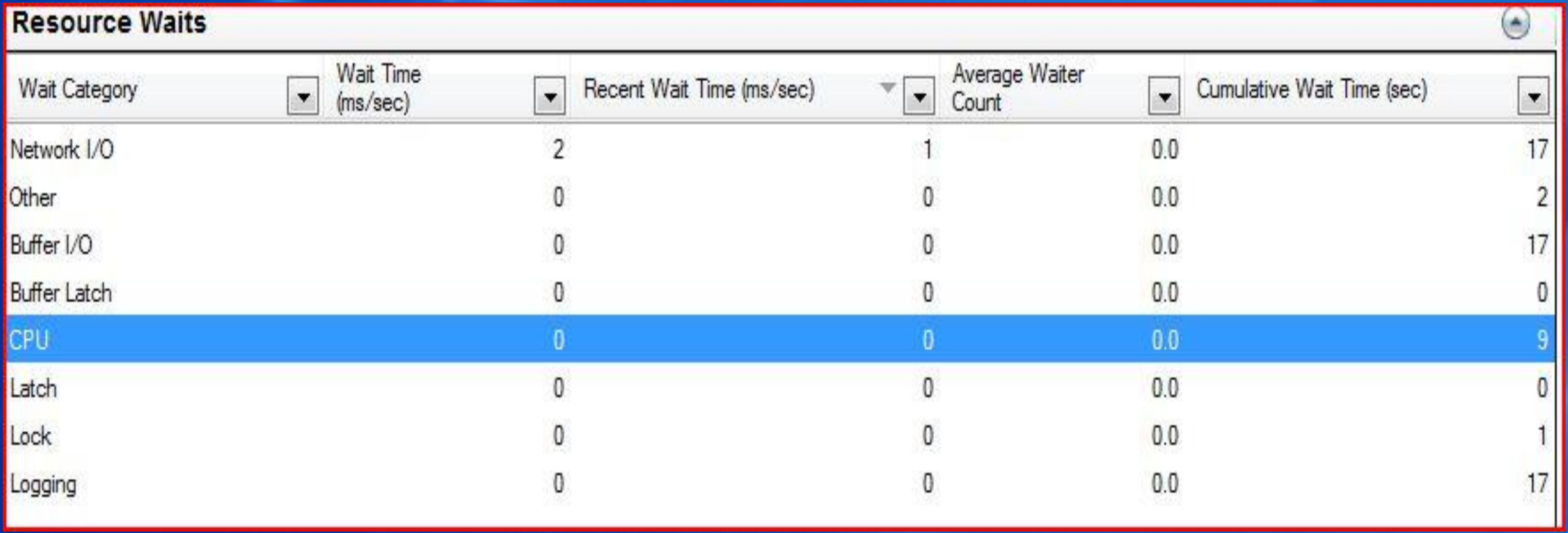


Processes

Se ID	Us Prc	Login	Databa	Task State	Commai	Applica	Wait Time (ms)	Wait Type	Wait Resoun	Bl By	He Blo	Memory Use (KB)	Host Name	Wor Grou
51	1	NORTHA...				Microsoft ...	0				1	16	KEITHEL...	
52	1	NORTHA...				Microsoft ...	0				1	16	KEITHEL...	
53	1	NORTHA...	tempdb	RUNNING	SELECT	Microsoft ...	0					16	KEITHEL...	

Resource Waits

- Categorized wait statistics
- Wait Time is delta for last time period only
- Recent Wait Time is time-weighted value of recent waits (slowly decays to zero over many snapshots)
- Cumulative Wait Time should match sys.dm_os_waits value



Resource Waits

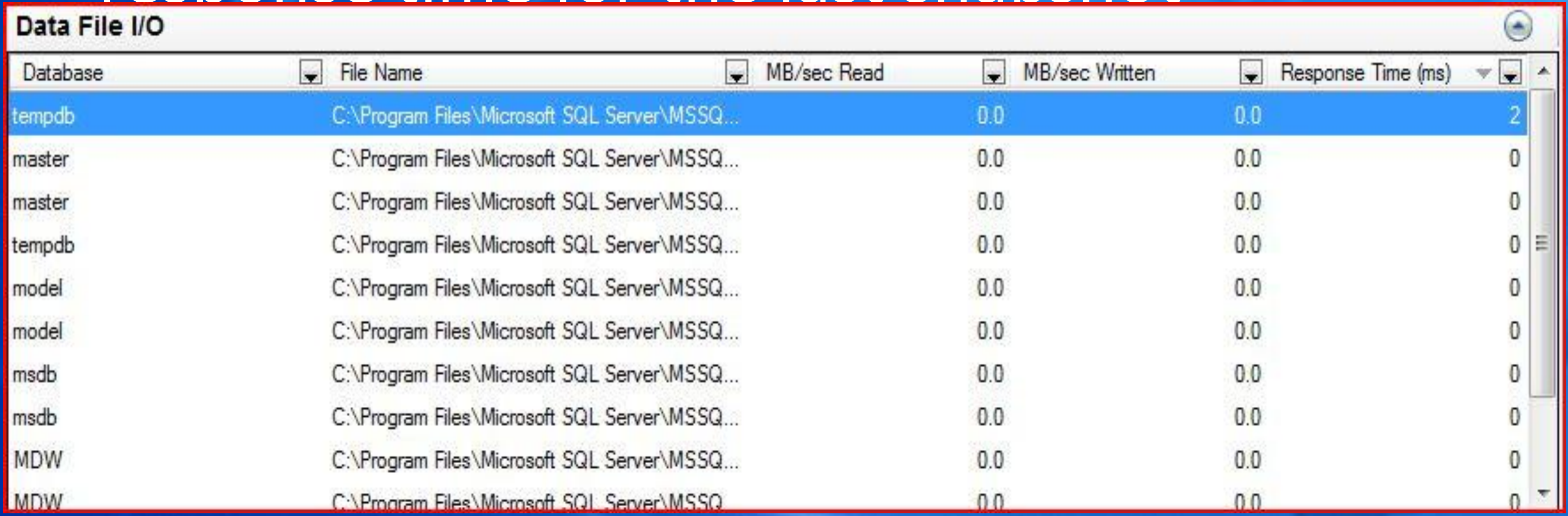
Wait Category	Wait Time (ms/sec)	Recent Wait Time (ms/sec)	Average Waiter Count	Cumulative Wait Time (sec)
Network I/O	2	1	0.0	17
Other	0	0	0.0	2
Buffer I/O	0	0	0.0	17
Buffer Latch	0	0	0.0	0
CPU	0	0	0.0	9
Latch	0	0	0.0	0
Lock	0	0	0.0	1
Logging	0	0	0.0	17

Microsoft

Database IO

CSS Academy

- Shows each file, IO rate (MB/sec) and response time for the last snapshot



Data File I/O

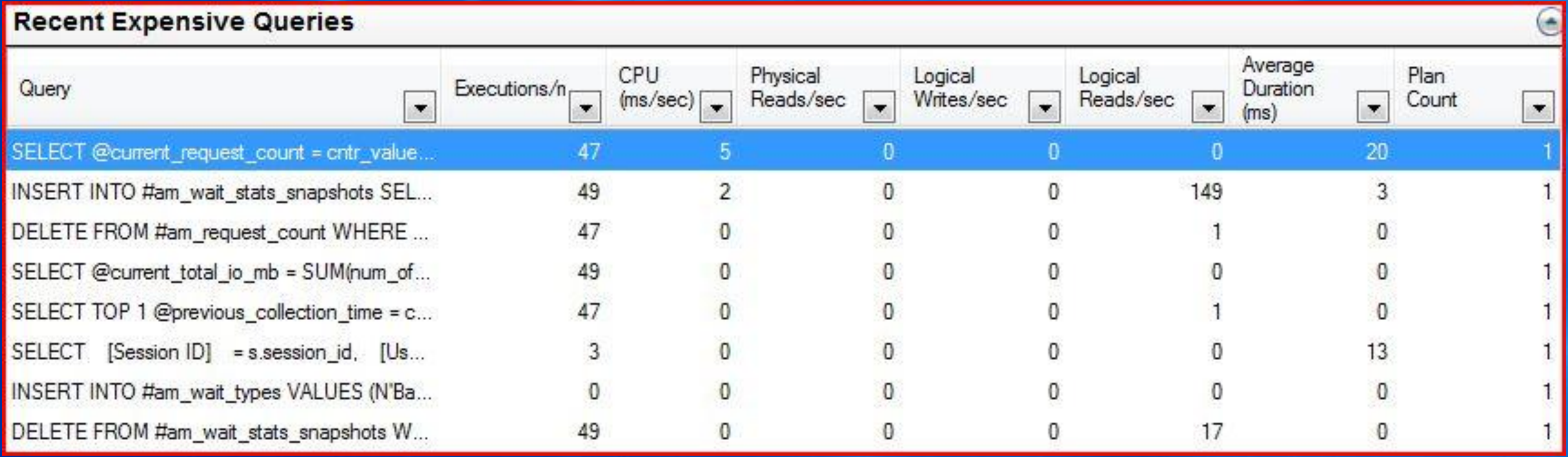
Database	File Name	MB/sec Read	MB/sec Written	Response Time (ms)
tempdb	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	2
master	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
master	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
tempdb	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
model	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
model	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
msdb	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
msdb	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
MDW	C:\Program Files\Microsoft SQL Server\MSSQ...	0.0	0.0	0
MDW	C:\Program Files\Microsoft SQL Server\MSSQ	0.0	0.0	0

Microsoft

CSS Academy

Expensive Queries

- Query must have run within last 4 hours
- Thereafter delta in query_stats/request values taken to show rates
- CPU may be double-counted for short, looping queries
- Data aggregated with GROUP BY on query_plan_hash to show stats per-same plan and not per-query



Recent Expensive Queries

Query	Executions/n	CPU (ms/sec)	Physical Reads/sec	Logical Writes/sec	Logical Reads/sec	Average Duration (ms)	Plan Count
SELECT @current_request_count = cntr_value...	47	5	0	0	0	20	1
INSERT INTO #am_wait_stats_snapshots SEL...	49	2	0	0	149	3	1
DELETE FROM #am_request_count WHERE ...	47	0	0	0	1	0	1
SELECT @current_total_io_mb = SUM(num_of...	49	0	0	0	0	0	1
SELECT TOP 1 @previous_collection_time = c...	47	0	0	0	1	0	1
SELECT [Session ID] = s.session_id, [Us...	3	0	0	0	0	13	1
INSERT INTO #am_wait_types VALUES (N'Ba...	0	0	0	0	0	0	1
DELETE FROM #am_wait_stats_snapshots W...	49	0	0	0	17	0	1

Microsoft

CSS Academy

Management Data Warehouse

- Infrastructure to capture & report SQL Server performance information

- Terminology
  - Target Server – the server being monitored
  - MDW Host – server holding the MDW database
  - Collection Type – type of data being collected (Query results, Perfmon, Trace events, …)
  - Collection Item – defines specific data to capture (e.g., script to run, perfmon event to capture, etc) and collection schedule
  - Collection Set – group of collection items, along with upload schedule

Microsoft

CSS Academy

Techniques for Controlling Query Performance

Microsoft

CSS Academy

Controlling Query Performance

New Query Hints

- **FORCESEEK**

  E.g. SELECT * FROM t WITH (**FORCESEEK**) WHERE vc LIKE 'Test%‘

- **OPTIMIZE FOR UNKNOWN**

  E.g. Select * from t where col > @p1 or col2 > @p2 order by col1
  option (OPTIMIZE FOR (@p1 **UNKNOWN**, @p2 **UNKNOWN**))

- **TABLE HINT**

  E.g. select * from t with (index(i)) where c1 = 0 option (**table hint** (t))

Controlling Query Performance

- **Plan Freezing**

The plan freezing framework in SQL Server 2008 builds on plan guides and is intended to be used to “lock down” plans for all repeated queries against a database or the entire system as opposed to just a handful of queries.

Microsoft

CSS Academy

Example - Plan Freezing

- **Get the plan handle**

```
select * from sys.dm_exec_query_stats qs
  cross apply sys.dm_exec_sql_text(qs.sql_handle)
where text like '%SalesOrderHeader%'
```

- Using the plan handle found in the previous step, create a plan guide for the query

```
sp_create_plan_guide_from_handle 'MyPlanGuide',
0x06000600B92F6E114021568F000000000000000000000000
```

Microsoft

CSS Academy

Introduction To Resource Governor

Introduction To Resource Governor

- History and Goals for Resource Governor
- When to Use Resource Governor

What Resource Can Be Governed

- CPU – managed through how often worker thread is scheduled
- Memory – managed through setting specific limits as a percentage of the pool
- Degree of Parallelism for a workload
- Level of importance of a workload to other workloads in the same resource pool
- Number of simultaneous statements running for a given workload

What Resource Cannot Be Governed

- Anything not listed on previous slide, but specifically, the number of reads or writes a statement can make (ie. Disk I/O is not regulated or limited).
- Resource management is limited to the SQL Server Database Engine. Resource Governor can not be used for Analysis Services, Integration Services, and Reporting Services.
- Resource Governor settings are instance specific and do not cross SQL Server instances.

How To Use Resource Governor

- Create additional and/or alter existing resource pools with the appropriate settings
- Create additional and/or alter existing workload group(s) with the appropriate settings and assign each workload group to a specific resource pool.
- Create/alter a user-defined function which returns the name of a workload group based on connection related information
- Register the UDF with Resource Governor
- Start the Resource Governor

Things to Remember

- Resource Governor is configurable and scriptable through a GUI interface in SQL Server Management Studio.
- Once enabled, all connections are governed into workload groups by the classifier function with the exception of the DAC. Use DAC to troubleshoot issues
- When ALTER RESOURCE GOVERNOR DISABLE is executed, all the configuration settings are set to their defaults and the classifier function is unregistered (not deleted)
- The default workload group will NOT prevent a query from running. This is for backwards compatibility with Yukon.

Resource Governor Related DMVs

- sys.dm_resource_governor_workload_groups
- sys.dm_resource_governor_resource_pools
- sys.dm_resource_governor_configuration

Examples - Resource Governor

- Create Resource Pool:

```
CREATE RESOURCE POOL PoolMarketingAdhoc
CREATE RESOURCE POOL PoolVP
```

- Create Workload Group

```
CREATE WORKLOAD GROUP GroupMarketing
USING PoolMarketingAdhoc

CREATE WORKLOAD GROUP GroupVP
USING PoolVP
```

Microsoft

CSS Academy

Examples - Resource Governor

- Create the classifier function

```
CREATE FUNCTION CLASSIFIER_V1 ()
RETURNS SYSNAME WITH SCHEMABINDING
BEGIN
        DECLARE @val varchar(32)
        if  'UserVP' = SUSER_SNAME()
                SET @val = 'GroupVP';
        else if 'UserMarketing' = SUSER_SNAME()
                SET @val = 'GroupMarketing';
        return @val;
END
GO
```

Examples - Resource Governor

- Make function known to the Resource Governor

```
ALTER RESOURCE GOVERNOR
WITH (CLASSIFIER_FUNCTION = dbo.CLASSIFIER_V1)
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```

- Adjust the pool according to your requriments

```
ALTER RESOURCE POOL PoolMarketingAdhoc
WITH (MAX_CPU_PERCENT = 50)
GO
ALTER WORKLOAD GROUP GroupMarketing
WITH (IMPORTANCE = Low)
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
```

Microsoft

CSS Academy

Q & A

Microsoft

CSS Academy

Thank You!

Microsoft

CSS Academy

Microsoft

Your potential. Our passion.™