Introducing…

# RESOURCE GOVERNOR IN SQL SERVER 2008

Steve Wright
Director of Product Support
SQL Sentry Inc. - www.SQLSentry.net

Thanks to Aaron Bertrand –
SQL Server MVP

# WHY A RESOURCE GOVERNOR?

Every DBA wants more control over resource usage on their SQL Server instances. Pressure comes from:

- IT : reducing data center footprint, hosting more instances on less hardware
- Other departments : sales, marketing, finance who want better performance
- Customers : my web report timed out

# WHAT IS THE RESOURCE GOVERNOR?

- A technology that enables you to manage SQL Server workload and resources by specifying limits on resource consumption

*… or …*

- The new way to prevent your peons and pointy-haired bosses from bringing down your server

# BEFORE RESOURCE GOVERNOR…

- What we did before to control resource usage was inflexible and largely reactive:
  - Kill SPIDs with “runaway” queries
  - utilize SET QUERY_GOVERNOR_COST_LIMIT per query (or per instance with sp_configure)
  - use separate instances with affinity to separate workloads
  - juggle scheduled jobs to avoid peak activity times
  - schedule creation / destruction of procedures or even changing passwords to block certain groups during peak load times

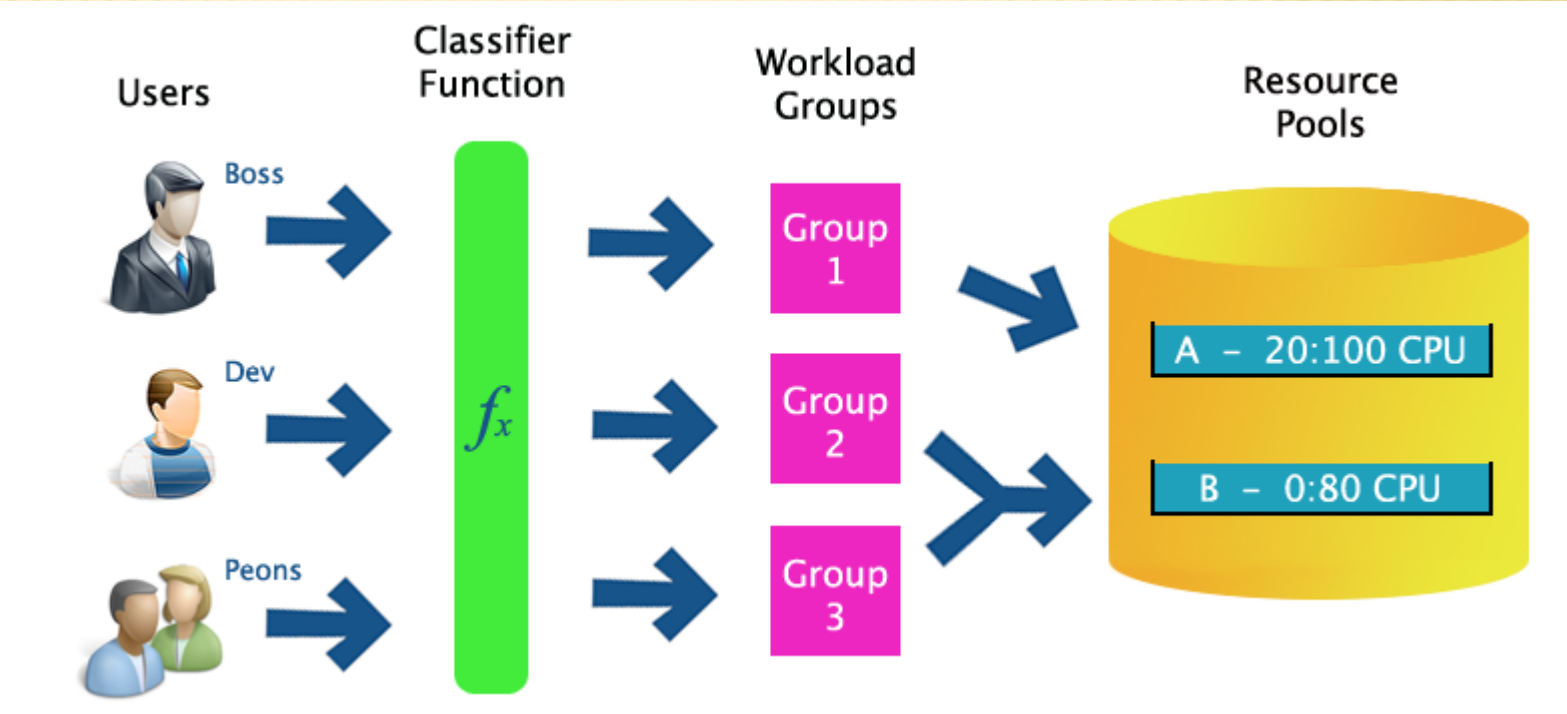
# WHAT ARE THE GOALS?

- To classify and prioritize workloads
- To make resource usage more balanced and predictable
- To help prevent, or at least to minimize, the “run away query”
- To monitor and adapt the above tactics to further smooth resource usage

# THREE COMPONENTS OF RESOURCE GOV.

- **Resource pools.** Two resource pools (internal and default) are created when SQL Server 2008 is installed.
- **Workload groups.** Two workload groups (internal and default) are created and mapped to their corresponding resource pools when SQL Server 2008 is installed.
- **Classifier.** There are internal rules that classify incoming requests and route them to a workload group.

# HOW DO THEY WORK TOGETHER?



# WHAT ARE THE BASIC STEPS?

- Create resource pools
- Create workload groups
- Create classifier function
- Enable resource governor
- Monitor and adapt

# WHAT IS A RESOURCE POOL?

- Provides a “slice” of a SQL Server instance’s resources (min/max CPU, memory, or both)
- Pools can overlap or be isolated
- % of resources based on amount “left over” – not being used by internal processes
- Allows for aggregate monitoring of all requests utilizing the pool

# RESOURCE POOL SYNTAX

```
CREATE RESOURCE POOL pool_name
[ WITH
( [
        MIN_CPU_PERCENT = value     ][[,]
        MAX_CPU_PERCENT = value     ][[,]
        MIN_MEMORY_PERCENT = value ][[,]
        MAX_MEMORY_PERCENT = value ]
)];
```

# WHAT IS A WORKLOAD GROUP?

- This acts as a bucket for requests of a similar type (as defined by the “classifier function”) and to place constraints on those requests
- Allows for aggregate monitoring of all requests from all the members of the group

# WORKLOAD GROUP SYNTAX

```
CREATE WORKLOAD GROUP group_name
[ WITH
( [
  IMPORTANCE = { LOW|MEDIUM|HIGH }                    ][[,]
  REQUEST_MAX_MEMORY_GRANT_PERCENT = value ][[,]
  REQUEST_MAX_CPU_TIME_SEC = value                    ][[,]
  REQUEST_MEMORY_GRANT_TIMEOUT_SEC = value ][[,]
  MAX_DOP = value                                     ][[,]
  GROUP_MAX_REQUESTS = value ]
)][
  USING { pool_name | "default" }
];
```

# WHAT IS A CLASSIFIER FUNCTION?

- User-defined scalar function that allows you to customize how incoming requests are routed
- Function returns a workload group name, which tells Resource Governor which pool to associate the request with
- Needs to be very efficient

# CLASSIFICATION PROCESS

In the context of Resource Governor, the login process for a session consists of the following steps:

- Login authentication
- LOGON trigger execution
- Classification

# WHAT ARE SOME CLASSIFICATION EXAMPLES?

- You can segregate incoming requests using a whole slew of criteria:
  - LOGINPROPERTY (DefaultLanguage, DefaultDatabase)
  - ORIGINAL_DB_NAME()
  - HOST_NAME(), APP_NAME() *
  - CONNECTIONPROPERTY() – IP address, protocol, etc.
  - [S]USER_[S]NAME()
  - IS_SRVROLEMEMBER(), IS_MEMBER()
  - Also intrinsic functions, DATEPART, GETDATE(), etc.
- Examples…

# CLASSIFIER FUNCTION EXAMPLE #1

- Give sa high priority, and non-sa low priority

```
CREATE FUNCTION dbo.Classifier()
RETURNS SYSNAME
WITH SCHEMABINDING
AS
BEGIN
    RETURN (SELECT CASE SUSER_SNAME()
        WHEN 'sa' THEN 'HighPriorityGroup'
        ELSE 'LowPriorityGroup'
    END
    );
END
GO
```

# CLASSIFIER FUNCTION EXAMPLE #2

- Give ad hoc Management Studio queries low priority during business hours, and high priority otherwise

```
CREATE FUNCTION dbo.Classifier()
RETURNS SYSNAME
WITH SCHEMABINDING
AS
BEGIN
   RETURN (SELECT CASE
             WHEN APP_NAME() LIKE '%Management Studio%'
             AND DATEPART(HOUR, GETDATE()) BETWEEN 9 AND 17
                 THEN 'LowPriorityGroup'
                 ELSE 'HighPriorityGroup'
             END
    );
END
GO
```

# CLASSIFIER FUNCTION EXAMPLE #3

- Get the ~~Dallas~~ Atlanta office back for that April Fool's joke they played on the DBA

```
CREATE FUNCTION dbo.Classifier()
RETURNS SYSNAME
WITH SCHEMABINDING
AS
BEGIN
   RETURN (SELECT CASE
     WHEN CONNECTIONPROPERTY('Local_Net_Address')
             LIKE '192.168.2.%' THEN 'Group_With_Max_CPU_1_Percent'
             ELSE 'HighPriorityGroup'
     END
     );
END
GO
```

# USER-DEFINED FUNCTION CHARACTERISTICS :

- The user-defined function is evaluated for every new session, even when connection pooling is enabled.
- The user-defined function gives workload group context for the session. After group membership is determined, the session is bound to the workload group for the lifetime of the session.
- If the user-defined function returns NULL, default, or the name of non-existent group the session is given the default workload group context. The session is also given the default context if the function fails for any reason.
- Only one user-defined function can be designated as a classifier at a time.
- The classifier user-defined function cannot be dropped or altered unless its classifier status is removed.

# THE DEFAULT WORKLOAD GROUP

Requests are classified into the default group when the following conditions exist:

- There are no criteria to classify a request.
- There is an attempt to classify the request into a non-existent group.
- There is a general classification failure.

# HOW DO I MONITOR?

- New Perfmon objects with lots of counters:
  - SQLServer : Resource Pool Stats
  - SQLServer : Workload Group Stats
- New trace events (e.g. CPU Threshold Exceeded)
- There are also new DMVs:
  - sys.dm_resource_governor_workload_groups
  - sys.dm_resource_governor_resource_pools
  - sys.dm_resource_governor_configuration

# DEMO

# WORKLOAD GROUP SYNTAX

```
CREATE WORKLOAD GROUP group_name
[ WITH
( [
  IMPORTANCE = { LOW|MEDIUM|HIGH }                    ][[,]
  REQUEST_MAX_MEMORY_GRANT_PERCENT = value ][[,]
  REQUEST_MAX_CPU_TIME_SEC = value                    ][[,]
  REQUEST_MEMORY_GRANT_TIMEOUT_SEC = value ][[,]
  MAX_DOP = value                                     ][[,]
  GROUP_MAX_REQUESTS = value ]
)][
  USING { pool_name | "default" }
];
```

# HOW DO I ADAPT?

- Re-schedule contentious processes based on observations from Perfmon, DMVs, trace
- Place different constraints on pools / groups
- Modify classifier function to change routing rules based on properties of request
- Note that Classification changes do not affect existing connections, but pool / group changes do (after RECONFIGURE)

# WHAT IF I MESS EVERYTHING UP?

- Use the DAC
- Start the Server in Single User Mode

# WHAT ARE THE LIMITATIONS?

- CPU / Memory only (no I/O yet)
- 2008 Database Engine only (no SSAS, SSRS, SSIS)
- Single instance only
- Short OLTP operations are immune to constraints
- Lack of contention can also prevent enforcement
- Cannot constrain “internal” processes
- Must disable classification to modify classifier function (you can’t apply these changes to existing sessions)
- Pool / group names are case sensitive
- Enterprise Edition only!

# COMMON MISTAKES

- Expecting importance to mean priority; importance applies weights to resources, does not form a "queue"
- Expecting classifier function to be case insensitive; MiS-CAsEd group names will end up in default group
- Creating classifier function in wrong database; function must be in master
- Cluster / multi-instance failover scenarios; need to be prepared to have instances share resources

# OTHER RESOURCES

- Usual suspects : BOL, MSDN, Blogs
- Some links that go beyond documentation:

http://blogs.msdn.com/psssql/archive/2008/01/10/sql-server-2008-resource-governor-questions.aspx

http://blogs.technet.com/sqlos/archive/2007/12/14/part-1-anatomy-of-sql-server-2008-resource-governor-cpu-demo.aspx

http://blogs.technet.com/sqlos/archive/2008/01/18/part-2-resource-governor-cpu-demo-on-multiple-cpus.aspx

# QUESTIONS?

# THANK YOU!

- www.sqlsentry.net
- Download a free trial version or SQL Sentry Event Manager and Performance Advisor