

Introducing...

# **RESOURCE GOVERNOR IN SQL SERVER 2008**

---

**Presenter – Jason Hall – SQL Sentry Inc.  
Initial Research – Aaron Bertrand, SQL Server MVP**

# 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

# 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

# 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



# 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 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
```

# WHAT ARE THE BASIC STEPS?

---

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

# 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

# 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)

# DEMO

---



# WHAT ARE THE LIMITATIONS?

---

- ✘ CPU / Memory only (no I/O yet)
- ✘ 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!

# OTHER RESOURCES

---

- ✘ Usual suspects : BOL, MSDN, Blogs
- ✘ Official Web Site

<http://www.microsoft.com/sqlServer/>

- ✘ 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/sqllos/archive/2007/12/14/part-1-anatomy-of-sql-server-2008-resource-governor-cpu-demo.aspx>

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

# QUESTIONS?

---