Servicio Andaluz de Salud
CONSEJERÍA DE SALUD

BEST PRACTICES PARA LA IMPLEMENTACIÓN DE RESOURCE MANAGER EN BASES DE DATOS ORACLE RDBMS

Referencia documento:
InfV5_JASAS_BestPractices_ResourceManager_V920.docx
Fecha: 13 de noviembre de 2018
Versión: 9.2.0

Registro de Cambios

Fecha	Autor	Versión	Notas
16 de Junio de 2015	Isidro Granados	6.2	Version Inicial
16 de Junio de 2016	Isidro Granados	7.1.0	Revisión de Junio de 2016, contrato 2014-2016
14 de octubre de 2016	Isidro Granados	7.2.0	Revisión de Noviembre de 2016, contrato 2014-2016
14 de junio de 2017	Isidro Granados	8.1.0	Revisión de Junio de 2017, contrato 2016-2018
16 de noviembre de 2017	Isidro Granados	8.2.0	Revisión de Noviembre de 2017, contrato 2016-2018.
16 de junio de 2018	Isidro Granados	9.1.0	Revisión de junio de 2018, contrato 2016-2018
13 de noviembre de 2018	Isidro Granados	9.2.0	Revisión de noviembre de 2018, contrato 2016-2018

Revisiones

Nombre	Role
Oracle ACS Engineers	ACS Service Engineer
Jose Maria Gomez	ACS Service TAM

Distribución

Copia	Nombre	Empresa
1	Subdirección de Tecnologías de la Información	Consejería de Salud, Junta de Andalucía
2	Servicio de Coordinación de Informática de la Consejería de Hacienda y Administración Pública	Consejería de Hacienda y Administración Pública, Junta de Andalucía

Índice de Contenidos

Control de cambios

Cambio	Descripción	Página
1	No se realizan cambios en esta versión del documento.	N/A

Introducción

La disponibilidad de un sistema de base de datos comprende tanto la funcionalidad como el rendimiento. Si una base de datos está disponible, pero los usuarios no obtienen el nivel de rendimiento que ellos necesitan, tanto los objetivos de la disponibilidad como los de nivel de servicio no se cumplirán. El rendimiento de una aplicación suele estar influido en la manera en que los recursos disponibles son distribuidos entre todas las aplicaciones accediendo a una base de datos.

Tradicionalmente, todos los usuarios y aplicaciones se les han permitido tener similar acceso a los recursos de la base de datos. Aunque esta forma de trabajar es adecuada en muchos casos, no funcionará si se necesita potenciar una necesidad fundamental del negocio, por ejemplo que algunas actividades en la empresa son más importantes que otras.

El objetivo de este documento es presentar las funcionalidades que provee Oracle para gestionar y limitar los recursos de usuarios dentro de la base de datos Oracle.

Oracle puede limitar el uso de recursos mediante la asignación de Perfiles ("Profiles") a usuarios y mediante el Gestor de Recursos, conocido como Oracle Database Resource Manager.

Oracle recomienda el uso de Oracle Database Resource Manager en lugar de usar "Profiles" para establecer límites de recursos ya que ofrece una manera más flexible y amplia de controlar y monitorizar el uso de recursos. Los "Profiles" son recomendados para configurar las preferencias en el mantenimiento de contraseñas de los usuarios.

El informe se centrará en Oracle Database Resource Manager detallando aspectos técnicos relativos a su configuración y administración, así como referencias y notas técnicas de relevancia.

El informe mostrará un ejemplo práctico de cómo diferentes estrategias de rendimiento pueden ser implementadas con Oracle Database Resource Manager. Se incluirán recomendaciones y consideraciones importantes a tener en cuenta cuando se quiera implementar un plan de recursos de Oracle Database Resource Manager.

El informe está basado en versión Oracle RDBMS Release 12cR2 (12.2.0.1) aunque muchos conceptos aplicarán a versiones anteriores.

Oracle Database Resource Manager

Oracle Database Resource Manager es un mecanismo de manejo de recursos dentro de la base de datos que permite controlar cómo los recursos son asignados en la base de datos, gestionando cómo los recursos disponibles en el sistema son repartidos entre todas las sesiones concurrentes en la base de datos.

La funcionalidad de Oracle Database Resource Manager viene incluida sin coste extra en la Oracle Database Enterprise Edition.

Oracle Database Resource Manager fue introducido en Oracle8i, ofreciendo un mecanismo de control de la cantidad de CPU asignada a cada grupo de recursos definidos y de limitar el máximo grado de paralelismo usado. En cada nueva versión de Oracle ha ido mejorando y ampliando la funcionalidad con más opciones.

Oracle Database Resource Manager permite a los administradores de la base de datos tener más control sobre las decisiones en la gestión de recursos y poder implementar políticas que garanticen los niveles de servicio adecuados en la base de datos a aplicaciones y usuarios. El DBA puede controlar como los recursos son asignados a cada grupo de usuarios, de forma que los procesos importantes reciban más recursos de los que no son tan importantes.

Funcionalidades

Con Oracle Database Resource Manager un administrador de base de datos puede:

- Especificar una distribución de los recursos de procesamiento disponibles asignando porcentajes de "CPU time" a diferentes usuarios y aplicaciones con diferentes niveles de prioridad. Con este mecanismo se puede garantizar a ciertos usuarios una cantidad mínima de recursos de procesamiento independientemente de la carga del sistema y del número de usuarios, y consultas ejecutadas por un cierto grupo de usuarios pueden obtener mayor prioridad de ejecución sobre otras operaciones en la base de datos.
- Particionar y dedicar recursos de CPU a una instancia de base de datos en un entorno consolidado.
- Limitar el grado de paralelismo de una operación llevada a cabo por miembros de un grupo de usuarios. Esto asegura que los servidores de ejecución paralela no son asignados a un solo grupo de usuarios.
- Crear un pool de sesiones activas, especificando el máximo número de sesiones de usuario activas permitidas concurrentemente para un grupo de usuarios. De esta forma se podría controlar el máximo número de llamadas concurrentes a la base de datos dentro de un grupo de usuarios. Las llamadas adicionales serían encoladas.
- Limitar la cantidad de memoria PGA usada por cada sesión que pertenece a un grupo de usuarios.

- Limitar las operaciones fuera de control activando un límite absoluto de CPU, consumo de IO físico, consumo de IO lógico o tiempo transcurrido "elapsed time" que una sesión puede consumir. Detectará cuando una sesión alcanza el límite establecido y automáticamente actuará sobre ella, bien finalizando la operación o la sesión o moviendo la sesión a un grupo con menos prioridad en la asignación de recursos. Por tanto puede ser usado para evitar que sesiones con operaciones muy largas impacten de forma negativa en el rendimiento global del sistema. También puede ser usado para guardar información de las sesiones y sentencias sql que sobrepasan más del límite establecido de cpu, IO físico, IO lógico o elapsed time.
- Impedir la ejecución de operaciones largas que el optimizador estima que puede tardar más que un determinando límite.
- Crear un pool de undo, especificando la cantidad de espacio de undo que puede ser consumida por un grupo de usuarios.
- Limitar el tiempo que una sesión puede estar inactiva. Además se puede restringir a aquellas que además de inactivas están bloqueando a otras sesiones.
- Configurar la instancia para usar un método en la asignación de recursos dinámicamente, sin necesidad de reinicio. Se puede definir un plan en horario de oficina para dar prioridad al ONLINE, y otro plan en horario nocturno para dar prioridad a los procesos BATCH, y que el cambio sea dinámico sin tener que reiniciar la instancia.
- Desde versión Oracle 11g, Oracle Database Resource Manager viene configurado por defecto para que las tareas de mantenimiento se ejecuten dentro de un plan de recursos pre configurado "default_maintenance_plan" en todas las ediciones de Oracle Database incluyendo Oracle Database Standard Edition. Sin embargo para poder personalizar planes de mantenimiento es requerido Oracle Database Enterprise Edition.
- En entornos Exadata se añade una funcionalidad extra llamada Exadata I/O Resource Manager (IORM). IORM es usada para controlar el uso del disco entre cargas de trabajo y base de datos. IORM es una herramienta muy valiosa para consolidación de base de datos. Esta funcionalidad está fuera del alcance de este documento. Para más información sobre esta características se recomienda la guía oficial de Oracle Exatada y la nota de MyOracleSupport *Information about this feature Configuring Exadata I/O Resource Manager for Common Scenarios (Doc ID 1363188.1).*
- A partir de versión 10g, Oracle Database Resource Manager es capaz de identificar el trabajo usando Servicios. Cuando una petición de trabajo se conecta a la base de datos usando un Servicio, Oracle Database Resource Manager se puede configurar para que el grupo de consumidores de recursos se le asigne transparentemente en función del Servicio usado. Esto permite manejar las peticiones de trabajo por servicios por orden de importancia.

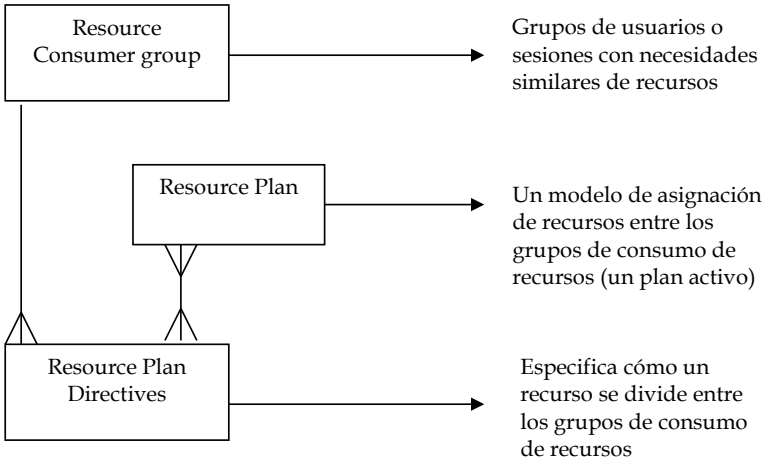
Conceptos

Los recursos se asignarán a los usuarios según un plan de recursos ("Resource Plan") especificado por el administrador, definido con los requisitos funcionales del sistema.

Los siguientes términos son usados en la especificación de un plan:

- **Plan de Recursos (Resource Plan)**: Es un conjunto de directivas o reglas (Oracle Database Resource Manager directives) que especifican cómo los recursos son asignados a los grupos de consumo de recursos ("resource consumer groups"). Los administradores especifican cómo se distribuyen los recursos activando un determinado plan de recursos ("Resource Plan"). Sólo un plan de recursos puede estar activo por instancia de base de datos.
- **Grupo de consumo de recursos (Resource consumer group):** Es un conjunto de sesiones de base de datos que son agrupadas en base a los requerimientos de los recursos. Oracle Database Resource Manager asigna recursos a grupos de consumo de recursos ("Resource consumer groups") no a sesiones individuales. Se puede asignar un usuario a varios grupos de consumo de recursos ("Resource consumer groups"), pero sólo un grupo podrá estar activo para una sesión. Tanto el usuario, como el DBA, pueden cambiar el grupo de consumo de recursos durante una sesión (el usuario tiene que tener privilegios específicos).
- **Directivas de planes de recursos (Resource plan directives):** Determinan como asignar recursos a los grupos de consumo en un plan de recursos especificando parámetros para cada método de asignación de recursos. Las directivas son un medio para:
  - Asignar grupos de consumidores o subplanes a planes de recursos.
  - Asignar recursos entre grupos de consumidores del plan, especificando parámetros para cada método de asignación de recursos.
- **Métodos de asignación de recursos (Resource allocation methods):** Determina que política usar en la asignación por parte de Database Resource Manager de un recurso particular
- **Subplanes (Subplans):** Planes dentro de cada plan de recursos, permitiendo la subdivisión diferentes usuarios para diferentes aplicaciones.
- **Niveles (Levels):** Proporciona un mecanismo de distribución de recursos no usados entre los usuarios disponibles, pudiendo especificar hasta 8 niveles en la asignación de recursos.

NOTA: A partir de ahora en el documento usará los nombres en inglés cuando se quiera referir a Plan de Recursos ("Resource Plan) y a grupos de consumidores de recursos ("Resource Consumer groups).



Esto es, Oracle Database Resource Manager distribuye los recursos entre los usuarios o aplicaciones basándose en un **Resource plan** especificado por el administrador de la base de datos. En el resource plan se especifica los **consumer groups** que pertenecen al plan y contiene las **resource plan directives** especificando cómo los recursos deberían ser distribuidos entre los **resource consumer groups**.

Sobre Resource Consumer Groups

Un resource consumer group (consumer group) es una colección de sesiones de usuario que son agrupadas juntas basándose en las necesidades de procesamiento. Cuando una sesión es iniciada, automáticamente es mapeada a un consumer group basándose en una serie de reglas de mapeo que podemos configurar. Como DBAs, podemos manualmente cambiar una sesión a un consumer group diferente. Igualmente, una aplicación puede ejecutar un procedimiento PL/SQL que cambie su sesión a un consumer group particular. En ambos casos para poder ser cambiada la sesión a otro consumer group, el usuario tendrá que tener privilegios dentro de este grupo.

Ya que Oracle Database Resource Manager distribuye recursos (como CPUs) solo a consumer group, cuando una sesión se convierte en miembro de un consumer group, la asignación de recursos estará determinada por la configuración de estos recursos en el consumer group dentro del plan activo.

Existen predefinidos una serie de consumer groups en el diccionario de datos que serán listados en el siguiente apartado. De los predefinidos hay varios especiales que no pueden ser modificados ni borrados estos son:

- **DEFAULT_CONSUMER_GROUP**, este es el consumer group inicial para todos los usuarios/sesiones a los que explícitamente no se le ha asignado un consumer group inicial. Esto es, es el que se le asigna por defecto cuando a los usuarios que no se le asigna uno en concreto. DEFAULT_CONSUMER_GROUP tiene privilegios asignados a PUBLIC, por tanto todos los usuarios están privilegiados automáticamente para uso de este consumer group.
- **OTHER_GROUPS**: Este consumer group no puede explícitamente asignarse a un usuario. Este grupo hay que definirlo siempre cuando se activen las directivas de un plan. Cualquier usuario que pertenezca a un grupo de recursos para el cual no haya directivas definidas en el plan activo, adquirirá las directivas que tenga este grupo, incluyendo el DEFAULT_CONSUMER_GROUP.
- **SYS_GROUP**: Es el consumer group inicial asignado a todas las sesiones creadas con las cuentas de usuario SYS o SYSTEM. Esta asignación inicial de consumer group puede ser sobrescrita según las reglas de mapeo entre sesiones y consumer group.

Un usuario sólo puede estar en un momento en un consumer group, aunque podría cambiarse (switch). Si no está asignado explícitamente a alguno, aparece como que está en el DEFAULT_CONSUMER_GROUP, si el DEFAULT_CONSUMER_GROUP no tuviera directivas en el plan activo, este tomará las directivas que haya definidas en OTHER_GROUPS, para ese plan. Por tal razón, siempre es necesario incluir estas directivas para OTHER_GROUPS.

Listado de consumer groups predefinidos en Oracle Database 12c Release 2 (12.2.0.1) y su descripción:

Resource Consumer Group	Description
BATCH_GROUP	Consumer group for batch operations. Referenced by the example plan MIXED_WORKLOAD_PLAN.
DSS_CRITICAL_GROUP	Consumer group for critical DSS queries. Referenced by the example plans DSS_PLAN and ETL_CRITICAL_PLAN.
DSS_GROUP	Consumer group for non-critical DSS queries. Referenced by the example plans DSS_PLAN and ETL_CRITICAL_PLAN.
ETL_GROUP	Consumer group for ETL jobs. Referenced by the example plans DSS_PLAN and ETL_CRITICAL_PLAN.
INTERACTIVE_GROUP	Consumer group for interactive, OLTP operations. Referenced by the example plan MIXED_WORKLOAD_PLAN.
LOW_GROUP	Consumer group for low-priority sessions.
ORA$AUTOTASK	Consumer group for maintenance tasks.
OTHER_GROUPS	Default consumer group for all sessions that do not have an explicit initial consumer group, are not mapped to a consumer group with session-to–consumer group mapping rules, or are mapped to a consumer group that is not in the currently active resource plan. OTHER_GROUPS must have a resource plan directive specified in every plan.

	It cannot be assigned explicitly to sessions through mapping rules.
SYS_GROUP	Consumer group for system administrators. It is the initial consumer group for all sessions created by user accounts SYS or SYSTEM. This initial consumer group can be overridden by session-to–consumer group mapping rules.

Resource plans predefinidos:

Oracle Database incluye una serie de resource plans predefinidos.

Listado y descripción de Resource Plans predefinidos en Oracle Database 12c Release 2 (12.2.0.1):

Resource Plan	Description
DEFAULT_MAINTENANCE_PLAN	Default plan for maintenance windows. See "About Resource Allocations for Automated Maintenance Tasks" for details of this plan. Because maintenance windows are regular Oracle Scheduler windows, you can change the resource plan associated with them, if desired. If you do change a maintenance window resource plan, ensure that you include the subplan ORA$AUTOTASK in the new plan.
DEFAULT_PLAN	Basic default plan that prioritizes SYS_GROUP operations and allocates minimal resources for automated maintenance and diagnostics operations.
DSS_PLAN	Example plan for a data warehouse that prioritizes critical DSS queries over non-critical DSS queries and ETL operations.
ETL_CRITICAL_PLAN	Example plan for a data warehouse that prioritizes ETL operations over DSS queries.
INTERNAL_PLAN	For disabling the resource manager. For internal use only.
INTERNAL_QUIESCE	For quiescing the database. This plan cannot be activated directly. To activate, use the QUIESCE command.
MIXED_WORKLOAD_PLAN	Example plan for a mixed workload that prioritizes interactive operations over batch operations. See "An Oracle-Supplied Mixed Workload Plan" for details.

Field Code Changed

El resource plan MIXED_WORKLOAD_PLAN es un ejemplo de configuración donde se priorizan las operaciones tipo OLTP sobre las operaciones tipo BATCH, e incluye los subplanes y consumer groups recomendados por Oracle.

El DEFAULT_PLAN es otro ejemplo de resource plan predefinido en la base de datos. Es un plan con configuración básica cuyo principal objetivo es mantener un estado correcto de la base de datos con un cambio mínimo de configuración, haciendo que los procesos de SYSTEM y SYS se programen con máxima prioridad y que las tareas automáticas de mantenimiento, como la recolección de estadísticas se programen con la más baja prioridad, haciendo que estas tareas no compitan con otras sesiones por CPU. Sin embargo si la carga de trabajo en la base de datos es mínima estas tareas de mantenimiento consumirán cualquier recurso de cpu disponible en la máquina.

Asignando sesiones a Consumer Groups

Antes de habilitar Resource Manager, se deberá configurar cómo las sesiones de usuarios son asignadas a los resource consumer groups. Haremos esto creando reglas de mapeo (mapping rules) que permiten a Resource Manager automáticamente asignar una sesión a un consumer group cuando la sesión arranque basándose en algún atributo de la sesión (Type Login). Después de que la sesión se le haya asignado un consumer group inicial y esté corriendo, podremos llamar a un procedure para manualmente cambiar la sesión a otro consumer group diferente. (Type Run-Time)

Los mapeos que podemos realizar pueden usar cualquiera de estos atributos:

Attibute	Type	Description
ORACLE_USER	Login	The Oracle Database user name
SERVICE_NAME	Login	The database service name used by the client to establish a connection
CLIENT_OS_USER	Login	The operating system user name of the client that is logging in
CLIENT_PROGRAM	Login	The name of the client program used to log in to the server
CLIENT_MACHINE	Login	The name of the computer from which the client is making the connection
CLIENT_ID	Login	The client identifier for the session The client identifier session attribute is set by the DBMS_SESSION.SET_IDENTIFIER procedure.
MODULE_NAME	Run-time	The module name in the currently running application as set by the DBMS_APPLICATION_INFO.SET_MODULE procedure or the equivalent OCI attribute setting
MODULE_NAME_ACTION	Run-time	A combination of the current module and the action being performed as set by either of the following procedures or their equivalent OCI attribute setting: • DBMS_APPLICATION_INFO.SET_MODULE • DBMS_APPLICATION_INFO.SET_ACTION The attribute is specified as the module name followed by a period (.), followed by the action name (module_name.action_name).
SERVICE_MODULE	Run-time	A combination of service and module names in this form: service_name.module_name
SERVICE_MODULE_ACTION	Run-time	A combination of service name, module name, and action name, in this form: service_name.module_name.action_name
ORACLE_FUNCTION	Run-time	An RMAN or Data Pump operation. Valid values are DATALOAD, BACKUP, and COPY. There are predefined mappings for each of these values. If your session is performing any of these functions, it is automatically mapped to a predefined consumer group.

Por ejemplo, el siguiente PL/SQL indica que el usuario SCOTT se mapee al consumer group DEV_GROUP cada vez que conecte a la base de datos:

```
BEGIN
  DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
     (DBMS_RESOURCE_MANAGER.ORACLE_USER, 'SCOTT', 'DEV_GROUP');
END;
/
```

Se pueden usar "wildcards" para estos atributos en el parámetro value en el procedure SET_CONSUMER_GROUP_MAPPING:

```
CLIENT_OS_USER
CLIENT_PROGRAM
CLIENT_MACHINE
MODULE_NAME
MODULE_NAME_ACTION
SERVICE_MODULE
SERVICE_MODULE_ACTION
```

También es posible establecer prioridades en las reglas de mapeos. Esto se usará para resolver los conflictos en las reglas de mapeo. Podemos establecer una prioridad ordenando los atributos de la sesión de mayor a menor importancia. Usaremos el procedure SET_CONSUMER_GROUP_MAPPING_PRI para configurar la prioridad de cada atributo a un valor desde 1 (más importante) a 10 (menos importante.

Uso de Servicios de base de datos con Oracle Database Resource Manager

Una regla de mapeo de las indicadas en el apartado anterior a destacar por su utilidad es el que Database Resource Manager puede automáticamente asignar cada sesión a un consumer group cuando inicia la sesión basándose en Servicios de base de datos.

Cuando un cliente conecta usando un Servicio, el consumer group se le asigna de forma transparente en el momento de establecer la conexión. Asignando servicios directamente a consumer groups nos facilita el poder gestionar la prioridad de los servicios por orden de importancia dentro de la instancia. Por ejemplo, podemos definir los servicios APP y BATCH en la misma instancia, y asignar AP al consumer group con alta prioridad y BATCH a un consumer group con baja prioridad. Las sesiones que conecten a la base de datos con el servicio AP especificándolo en su descriptor de conexión TNS obtendrán más prioridad que aquellos que conecten con el servicio BATCH.

Tipo de recursos manejados por Oracle Database Resource Manager

Con las "Resource plan directives" especificaremos cómo serán los recursos asignados a los resource consumer groups o subplans. Los siguientes apartados resumen los métodos de asignación de recursos disponibles:

Asignación de CPU

El método de asignación de CPU permite a los administradores indicar cómo los recursos de CPU serán asignados entre los consumer groups o subplans cuando los recursos de CPU estén saturados en el sistema. Los administradores también pueden

establecer un límite en la cantidad de recursos de CPU que pueden ser asignados a un consumer group particular independientemente de la carga que tenga el sistema.

Resource Manager dispone de los siguientes atributos para controlar la asignación de recursos de CPU:

- Asignación de CPU a consumer groups

  Con la asignación de una cantidad mínima de CPU a sesiones en cada consumer group, Oracle Database Resource Manager puede bajar el consumo de CPU a sesiones con baja prioridad, y asignar más recursos a sesiones con mayor prioridad.

  Por defecto, el método de asignación de CPU usa porcentajes (llamado EMPHASIS) y puede ser usado con múltiples niveles o con un sólo nivel. Este método soporta la asignación de CPU hasta ocho diferentes niveles. El uso de múltiples niveles permite priorizar la CPU dentro de un plan. Los consumer groups y subplanes en el nivel 2 obtendrán los recursos que no hayan sido asignados a nivel 1 y aquellos que aunque hayan sido asignados al nivel 1 no están siendo consumidos completamente por consumer groups o subplans del nivel 1. Igualmente, al nivel 3 los recursos serán asignados cuando quede algo no asignado en niveles 1 y 2. Las mismas reglas aplican a los niveles 4 a 8. El uso de múltiples niveles no solo permite una forma de priorizar, también una forma de explícitamente especificar como los recursos primarios y sobrantes tienen que ser usados.

  También existe un método alternativo (llamado RATIO) que en lugar de porcentajes usará ratios en la asignación de CPU a consumer groups y subplanes, en este caso sólo para planes en único nivel. Este será el método que configura EM Database Express 12c.

- Límite máximo de utilización

  Usaremos el atributo UTILIZATION_LIMIT para imponer un límite absoluto superior de la utilización de CPU asignada a un consumer group. Este límite absoluto sobrescribe cualquier redistribución de CPU dentro del plan.

  La configuración de UTILIZATION_LIMIT es opcional. Si se omite este atributo para un consumer group no habrá límite en la cantidad de CPU que el consumer group puede usar. Por tanto, si todas las otras aplicaciones están inactivas u ociosas, un consumer group que no tiene configurado el UTILIZATION_LIMIT puede asignarse el 100% de los recursos de CPU del sistema.

Active Session Pool

Oracle Database Resource Manager permite controlar el número máximo de llamadas concurrentes a la base de datos en cualquier momento dentro de un consumer group. Este máximo es lo que llama "active session pool". Cuando una nueva llamada no puede iniciarse porque el pool está lleno, la sesión se pondrá en una cola. Cuando una de las llamadas activas se complete, la primera sesión en la cola iniciará la ejecución de la llamada solicitada.

También es posible indicar un periodo de timeout que una sesión puede estar esperando en la cola, causando un error en la llamada.

Es recomendado usar el Active Session Pool con cuidado, ya que todas las operaciones de base de datos están sujetas a este límite:

- Parallel queries
- Transactions
- "connect scott/tiger"
- "select sysdate from dual"
- "alter session"

Active Session Pool no debería usarse en cargas de trabajo OLTP junto con operaciones largas, operaciones muy rápidas podrían encolarse hasta que terminaran las largas. Tampoco para implementar un "connection pooling" o para una carga de trabajo con parallel queries, en este caso habría que usar en lugar de esto un Parallel Statement Queuing.

Parallel Execution Servers

Los administradores pueden controlar el uso de los servidores de ejecución paralela (parallel execution servers) de una base de datos de estas formas:

- **Limitando el grado de paralelismo.**

  Se puede limitar el número máximo de servidores de ejecución paralela (parallel servers) asociados con cualquier operación dentro de un consumer group especificando el grado máximo de paralelismo. Usaremos la directiva `PARALLEL_DEGREE_LIMIT_P1` para especificar el límite del grado de paralelismo para un consumer group.

  Este método puede ser muy útil en entornos data warehouse ya que ayuda a impedir que una única operación paralela consuma todos los recursos de procesamiento paralelo del sistema.

  Este límite aplicará a una operación dentro del consumer group, no será un límite del total de grado de paralelismo conjunto de todas las operaciones dentro del consumer group. Sin embargo se puede combinar las directivas `PARALLEL_DEGREE_LIMIT_P1` y `PARALLEL_SERVER_LIMIT` para obtener este control.

  Los siguientes directivas requieren tener habilitado el PARALLEL_DEGREE_POLICY=AUTO/ADAPTIVE (automatic degree of parallelism o Auto DOP).

- **Parallel Server limit**

  Se puede usar la directiva PARALLEL_SERVER_LIMIT para especificar el porcentaje máximo del pool de parallel execution servers que un consumer group puede usar. De esta forma se puede evitar que un único consumer group pueda lanzar tantas sentencias paralelas que usen todos los parallel execution servers.

El número de parallel execution servers usados por un consumer group concreto es contado como la suma de todos los parallel execution servers activas usadas por todas las sesiones en ese consumer group. Si un consumer group está usando un número de parallel execution servers superior al límite establecido por el PARALLEL_SERVER_LIMIT, sus sentencias paralelas serán encoladas.

Por ejemplo, suponiendo que el número de parallel execution servers es 32 (configurado con el parámetro de inicialización PARALLEL_SERVERS_TARGET) y la directiva PARALLEL_SERVER_LIMIT para el consumer group MY_GROUP es del 50%. Esto indicará que el grupo puede usar un máximo del 50% de 32, o 16 parallel execution servers.

En un entorno RAC el límite que establece la directiva PARALLEL_SERVER_LIMIT a un consumer group aplicará agrupando todas las sesiones del grupo de todas las instancias del RAC. El límite donde empezará a encolar será la suma de (PARALLEL_SERVER_LIMIT * PARALLEL_SERVERS_TARGET / 100) de todas las instancias del RAC.

Por ejemplo, en un entorno Oracle RAC, el parámetro de inicialización PARALLEL_SERVERS_TARGET se configura a 32 en los 2 nodos del RAC, lo que hace un total de 32 x 2 = 64 parallel servers que pueden usarse antes de empezar a encolar. Podemos configurar que el consumer group PQ_LOW use el 50% de los parallel servers disponibles (parallel_server_limit = 50) y que las sentencias con baja prioridad vayan al consumer group PQ_LOW. En este escenario, cualquier sentencia paralela del consumer group está limitada a 64 x 50% = 32 parallel servers, incluso aunque haya inactivos otros parallel servers. En este caso, si sentencias del consumer group están usando 32 parallel servers, cualquier nueva sentencia del grupo será encolada.

La base de datos mantendrá una cola de sentencias paralelas para cada consumer group. Cada cola será manejada como una cola First In First Out (FIFO), esto es, según van entrando las sentencias a la cola irán saliendo. Para un entorno RAC la cola será común para todas las sentencias de las sesiones del consumer group en todas las instancias.

Si el total de parallel executions servers resultantes de la configuración del PARALLEL_SERVER_LIMIT de todos los consumer group exceden el 100%, en un escenario en el que el número de parallel executions servers entre todas los consumer group alcancen el valor del parámetro PARALLEL_SERVERS_TARGET, habrá una cola separada de sentencias paralelas que también será manejada como una cola First In First Out (FIFO). Se pueden usar atributos (MGMT_P1, MGMT_P2, etc) para gestionar la prioridad de cómo se van desencolando de esta cola las peticiones paralelas a los distintos grupos de recursos, por ejemplo dando más prioridad a que las sentencias paralelas de ciertos consumer groups salgan antes de la cola.

También en tiempo de ejecución se puede influenciar la gestión de la cola de varias sentencias paralelas si se incluyen el conjunto de sentencias dentro de un bloque gestionado por DBMS_RESOURCE_MANAGER y los procedures BEGIN_SQL_BLOCK END_SQL_BLOCK. De esta forma se pueden agrupar sentencias para que el tiempo en la cola de espera se minimice en el conjunto de sentencias, haciendo que las sentencias utilicen como tiempo de entrada en la cola el de la primera sentencia paralela que se encole dentro del conjunto de SQLs.

En resumen una sentencia paralela será encolada si se cumplen estas condiciones:

- El PARALLEL_DEGREE_POLICY está AUTO o ADAPTIVE.
- El número de parallel execution servers activos a lo largo de todos los consumer groups excede la configuración del PARALLEL_SERVERS_TARGET. Esto será independiente de tener o no el PARALLEL_SERVER_LIMIT, esto es, si no se especifica el PARALLEL_SERVER_LIMIT cogerá el 100% para el consumer group.
- La suma de todos los parallel execution servers activos para el consumer group y el grado de paralelismo de la sentencia excede el límite resultante de la configuración de PARALLEL_SERVER_LIMIT. En single-instance será indicado por el PARALLEL_SERVER_LIMIT/100 * PARALLEL_SERVERS_TARGET y para RAC será la suma de (PARALLEL_SERVER_LIMIT * PARALLEL_SERVERS_TARGET / 100) por cada instancia del RAC.

Aún con el uso de PARALLEL_DEGREE_POLICY a AUTO se podrá evitar que una sentencia paralela sea encolada:

- Configurando la directiva PARALLEL_STMT_CRITICAL a BYPASS_QUEUE para los consumer group con más prioridad haciendo que las sentencias paralelas del consumer group se salten la cola de sentencias paralelas.
- Usando del HINT `NO_STATEMENT_QUEUING` evita que una sentencia entre en la cola de sentencias paralelas.

  NOTA: En ambos casos, potencialmente la sentencia puede provocar que se exceda el valor máximo de parallel execution servers especificado con el valor del parámetro PARALLEL_SERVERS_TARGET. En cualquier caso no se garantiza que la sentencia reciba el número de parallel executions servers solicitados ya que sólo el número de parallel execution servers disponibles en el sistema, cuyo límite lo marca el parámetro PARALLEL_MAX_SERVERS, pueden ser asignados.

Es posible que una base de datos tenga sentencias con el parallelism degree policy configurado a MANUAL y otras a AUTO. En este escenario, sólo las sentencias con el parallelism degree policy a AUTO serán encoladas. Sin embargo, los parallel servers usados por sesiones con un parallelism degree policy a MANUAL se computan en el total de parallel servers que usa un consumer group.

- **Parallel Queue Timeout**

Con la directiva PARALLEL_QUEUE_TIMEOUT se puede especificar un tiempo máximo en segundos para que una sentencia pueda esperar en la cola de sentencias paralelas. Cuando una sentencia paralela alcanza un timeout esta fallará con el siguiente mensaje:

ORA-07454: queue timeout, n second(s), exceeded

Program Global Area (PGA)

Desde versión 12.2.0.1 es posible usar Resource Manager para limitar el uso de PGA a los usuarios de un consumer group usando la directiva `session_pga_limit`.

El valor de este parámetro indicará en Megabytes el tamaño máximo de memoria PGA permitida para cada sesión del consumer group. En el caso de exceder el límite la sesión recibirá un error ORA-10260. Este límite incluye a los procesos paralelos asociados y a los procesos de la job queue.

Este límite afectará a las sesiones del consumer group y es independiente del valor indicado en el parámetro de inicialización PGA_AGGREGATE_LIMIT que afectará de forma global a toda instancia. En este caso cuando la PGA de forma global en la instancia llega a este límite, las sesiones con mayor PGA asignada en primer lugar abortarán sus llamadas, y si esto no libera la memoria, las sesiones son finalizadas.

Umbrales de ejecución:

Automatic Consumer Group Switching

Canceling SQL and Terminating Sessions

Execution Time Limit.

Los administradores pueden controlar los recursos especificando que las sesiones automáticamente puedan ser movidas de un consumer group a otro, normalmente con menor prioridad en base a ciertos criterios. Igualmente se pueden especificar directivas para cancelar sentencias muy largas o incluso terminar sesiones basándose en la cantidad de recursos consumidos en el sistema.

Se puede especificar límites de consumo de CPU, consumo de I/O físico y/o lógico y tiempo total (elapsed time) para las sesiones de un consumer group, así como especificar la acción que Resource Manager tiene que llevar a cabo en caso de que una operación exceda alguno de los límites establecidos. Las posibles acciones son:

- La sesión se cambia dinámicamente a otro consumer group. (Automatic Consumer Group Switching)

  Normalmente el nuevo consumer group es uno con menor asignación de recursos.

  NOTA: Desde versión 12c el usuario adquiere privilegios de ser cambiado al consumer group de forma implícita al ser cambiado por medio de un Automatic Consumer Group Switching, no es requerido dar explícitamente los privilegios como en versiones anteriores.
- La sesión es finalizada (killed).
- La sentencia SQL actual de la sesión es abortada.
- Se genera información de log de la sesión pero no se realiza ninguna acción.

Para establecer que la sesión cambie de forma automática usaremos las siguientes directivas:

- SWITCH_GROUP: Indica el consumer group al que será cambiado en caso de que el criterio de cambio se cumpla. Si el group name es 'CANCEL_SQL', la llamada actual es cancelada en caso de que el criterio se cumpla. Si el the group name is 'KILL_SESSION', la sesión será finalizada. Por defecto el valor es NULL. Si el group name es LOG_ONLY, se registrará información real-time SQL monitoring sobre la sesión, pero no se realizará ninguna acción sobre la sesión.

  Si el group name es 'CANCEL_SQL', el parámetro SWITCH_FOR_CALL será siempre configurado a TRUE, sobrescribiendo cualquier otro valor que se configure.

- SWITCH_TIME: Especifica el tiempo (Segundos de CPU) que una llamada puede ser ejecutada antes de que se ejecute la acción configurada en SWITCH_GROUP. Por defecto es ilimitada (UNLIMITED).

- SWITCH_ESTIMATE:

  Si se configura a TRUE, la base de datos estima el tiempo de ejecución de cada llamada, si el tiempo estimado excede el valor de SWITCH_TIME, la sesión es movida al SWITCH_GROUP antes de que la llamada comience. Por defecto es FALSE.

  La estimación del tiempo de ejecución es obtenida desde el optimizador. La precisión de la estimación depende de diversos factores, especialmente de la calidad de las estadísticas del optimizador.

- SWITCH_IO_MEGABYTES: Indica el número de megabytes de I/O que una sesión puede realizar (lectura y escritura) en una SQL antes de que se ejecute la acción configurada en SWITCH_GROUP. Por defecto es UNLIMITED. Opción disponible desde versión 11g.

- SWITCH_IO_REQS: Indica el número de peticiones de I/O antes de que se ejecute la acción configurada en SWITCH_GROUP. Por defecto es UNLIMITED. Opción disponible desde versión 11g.

- SWITCH_IO_LOGICAL: Indica el número de peticiones de I/O lógicas antes de que se ejecute la acción configurada en SWITCH_GROUP. Por defecto es UNLIMITED. Opción disponible desde versión 12c.

- SWITCH_ELASED_TIME: Indica el número de segundos antes de que se ejecute la acción configurada en SWITCH_GROUP. Por defecto es UNLIMITED. Opción disponible desde versión 12c.

- SWITCH_FOR_CALL: Si es configurada a TRUE, una sesión que haya sido automáticamente movida a otro consumer group (de acuerdo con SWITCH_TIME, SWITCH_IO_MEGABYTES, SWITCH_IO_REQS, SWITCH_IO_LOGICAL o SWITCH_ELASED_TIME) vuelve a su consumer group original cuando el nivel superior de la llamada sea completado. El nivel superior de una llamada en PL/SQL es el bloque completo de PL/SQL que es tratado como una llamada. El nivel superior en SQL es una sentencia SQL individual. En caso de configuración a FALSE, una sesión continuará en el nuevo consumer group hasta que se considere "idle", en ese momento volverá a ser cambiada a su consumer group original. Resource Manager considera que una sesión es "idle" si pasa una cantidad de tiempo entre llamadas. Este intervalo de tiempo son unos pocos segundos (5 segundos) y no es configurable.

En general, es recomendado el uso de SWITCH_FOR_CALL a TRUE, especialmente para aplicaciones en tres capas cuando los servidores middle tier servers usan "session pooling".

Una sesión que ha sido movida a un nuevo consumer group continuará ejecutándose aunque haya un "active session pool" en el nuevo grupo y este pool esté lleno. Bajo estas condiciones, un consumer group podría tener más sesiones activas ejecutándose que las especificadas en su "active session pool".

Tiempo máximo estimado de ejecución

Oracle Database Resource Manager nos permite especificar el tiempo máximo de ejecución estimado permitido para una operación en base de datos. Si la base de datos estima que la ejecución de la operación podría ser más larga que el tiempo máximo, la operación no arranca. Esto puede ser usado para prevenir operaciones largas no permitidas que podrían consumir muchos recursos antes incluso de que se inicien.

Este tiempo máximo se configura con la directiva MAX_EST_EXEC_TIME que especifica el tiempo máximo de ejecución (en segundos de CPU) permitidos para una sesión. Si el optimizador estima que la operación será más larga que lo indicado en el MAX_EST_EXEC_TIME, la operación no arrancará y generará un ORA-07455.
Si el optimizador no genera ninguna estimación, la directiva no tiene efecto. Por defecto es UNLIMITED.

Undo Pool

Los administradores pueden especificar un pool de undo para cada consumer group y así poder control el tamaño de rollback que puede ser generado por las sesiones de un consumer group. Cuando el total de undo generado excede su límite, la correspondiente sentencia SQL es finalizada. Ningún otro miembro del consumer group puede realizar más transacciones hasta que se libere espacio de undo en el pool. Este método es usado para prevenir transacciones fuera de control que consumen excesivo espacio de undo.

Para el manejo de undo se utiliza la directive "undo_pool" que limitará en KBs el "active undo", no el "unexpired undo" (usado para flashback y para lecturas consistentes si el UNDO_RETENTION es alto y hay consultas SQL largas activas).

Límite en el tiempo de inactividad.

Con Oracle Database Resource Manager el administrador puede especificar la cantidad de tiempo que una sesión puede estar inactiva ("idle"), después del cual la sesión será finalizada. Es posible restringir esta finalización de sesiones a sólo aquellas que bloqueen a otras sesiones.

Etapas para configurar Oracle Database Resource Manager

A continuación se listan las típicas etapas que se necesitan para configurar Oracle Database Resource Manager algunas son opcionales:

- Otorgar privilegios para administrar Oracle Database Resource Manager. (SYS o SYSTEM lo tienen).
- Decidir entre crear un Resource Plan simple o complejo.
- Crear un "resource plan"
- Asignar las sesiones de base de datos a los "resource consumer group".
- Otorgar "switch privilege" para los "resource consumer groups"
- Habilitar el resource plan

Otorgar privilegios para Oracle Database Resource Manager

Un administrador con la opción ADMIN puede otorgar el privilegio del sistema ADMINISTER_RESOURCE_MANAGER a otros usuarios o roles para permitirles poder administrar Oracle Database Resource Manager.

```
SQL> exec DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE (
    GRANTEE_NAME => 'SCOTT',
    PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER',
    ADMIN_OPTION => FALSE);
```

En este ejemplo, el privilegio es otorgado a SCOTT. Ya que no se le indica la opción ADMIN, SCOTT no podría otorgar el privilegio a otros usuarios. Este privilegio puede ser revocado usando el procedure REVOKE_SYSTEM_PRIVILEGE de forma similar.

Decidir entre un Resource Plan simple o complejo

Tal y como se indicó más arriba, hay una serie de resource plans predefinidos en la base de datos que podrían ser usados. Si ninguno de estos es suficiente para cumplir con los requerimientos se puede decidir entre crear un resource plan simple o complejo.

- **Resource Plan simple:**

  Es un plan en el que sólo se especifica la distribución de asignación de CPU. Permite crear de forma muy rápida y sencilla un resource plan con el procedure CREATE_SIMPLE_PLAN. Este procedure nos permite crear consumer groups y asignarles recursos con una única llamada. Se permite hasta 8 consumer groups en el procedimiento.

  Ejemplo:

```
SQL> exec DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN
(SIMPLE_PLAN => 'SALES_SIMPLE_PLAN',
```

```
CONSUMER_GROUP1 => 'SLSGRP1', GROUP1_PERCENT => 25,
CONSUMER_GROUP2 => 'SLSGRP2', GROUP2_PERCENT => 30,
CONSUMER_GROUP3 => 'SLSGRP3', GROUP3_PERCENT => 40,
CONSUMER_GROUP4 => 'SLSGRP4', GROUP4_PERCENT => 5);
```

- **Resource Plan complejo:**

Cuando queremos utilizar todas las características de Resource Manager necesitaremos crear un resource plan complejo. Un resource plan se dirá complejo a cualquier resource plan que no haya sido creado con el procedure DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN. Para esto se necesitará crear el plan, con sus directivas y sus consumer groups en una área intermedia llamada "pending area", que permite validar el plan antes de ser guardado en el diccionario de datos de la base de datos.

Para crear este tipo de plan se requiere la ejecución de múltiples tareas, como se indica a continuación:

Etapa 1. Creación de una "pending area"

Es una area de trabajo temporal llamada "pending area" que deberá ser creada para alojar la configuración de Resource Manager cuando se crea, modifica o se borra un resource plan.

```
SQL> exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

Los cambios realizados en la "pending area" no son visibles hasta que los cambios son validados y confirmados. Al ser confirmados, todos los cambios pendientes son aplicados en el diccionario de datos y la "pending area" es borrada y desactivada. Sólo se puede tener una "pending area" por base de datos activada en un momento dado.

Los cambios realizados en la "pending area" pueden ser abandonados en cualquier momento limpiándola:

```
SQL> exec DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
```

Etapa 2. Crear, modificar o borrar consumer groups.

Si los resource consumer groups predefinidos en la base de datos no son suficientes para los requerimientos, se pueden crear nuevos resource consumer group usando el procedure CREATE_CONSUMER_GROUP.

Etapa 3. Crear el resource plan.

Si los resource plans predefinidos no son suficientes para los requerimientos, se creará un nuevo resource plan con el procedure CREATE_PLAN.

Etapa 4. Creación de directivas del resource plan.

Usaremos el procedure CREATE_PLAN_DIRECTIVE para crear las directivas del resource plan. Cada directiva pertenece a un plan o subplan y asigna recursos a un consumer group o a un subplan.

Se pueden especificar los siguientes parámetros:

Parameter	Description
PLAN	Name of the resource plan to which the directive belongs.
GROUP_OR_SUBPLAN	Name of the consumer group or subplan to which to allocate resources.
COMMENT	Any comment.
CPU_P1	Deprecated. Use MGMT_P1.
CPU_P2	Deprecated. Use MGMT_P2.
CPU_P3	Deprecated. Use MGMT_P3.
CPU_P4	Deprecated. Use MGMT_P4.
CPU_P5	Deprecated. Use MGMT_P5.
CPU_P6	Deprecated. Use MGMT_P6.
CPU_P7	Deprecated. Use MGMT_P7.
CPU_P8	Deprecated. Use MGMT_P8.
ACTIVE_SESS_POOL_P1	Specifies the maximum number of concurrently active sessions for a consumer group. Other sessions await execution in an inactive session queue. Default is UNLIMITED.
QUEUEING_P1	Specifies time (in seconds) after which a session in an inactive session queue (waiting for execution) times out and the call is aborted. Default is UNLIMITED.
PARALLEL_DEGREE_LIMIT_P1	Specifies a limit on the degree of parallelism for any operation. Default is UNLIMITED.
SWITCH_GROUP	Specifies the consumer group to which a session is switched if switch criteria are met. If the group name is CANCEL_SQL, then the current call is canceled when switch criteria are met. If the group name is CANCEL_SQL, then the SWITCH_FOR_CALL parameter is always set to TRUE, overriding the user-specified setting. If the group name is KILL_SESSION, then the session is killed when switch criteria are met. If the group name is LOG_ONLY, then information about the session is recorded in real-time SQL monitoring, but no specific action is taken for the session. If NULL, then the session is not switched and no additional logging is performed. The default is NULL. An error is returned if this parameter is set to NULL and any other switch parameter is set to non-NULL. Note: The following consumer group names are reserved: CANCEL_SQL, KILL_SESSION, and LOG_ONLY. An error results if you attempt to create a consumer group with one of these names.
SWITCH_TIME	Specifies the time (in CPU seconds) that a call can execute before an action is taken. Default is UNLIMITED. The action is specified by SWITCH_GROUP.
SWITCH_ESTIMATE	If TRUE, the database estimates the execution time of each call, and if estimated execution time exceeds SWITCH_TIME, the session is switched to the SWITCH_GROUP before

	beginning the call. Default is FALSE. The execution time estimate is obtained from the optimizer. The accuracy of the estimate is dependent on many factors, especially the quality of the optimizer statistics. In general, you should expect statistics to be no more accurate than ± 10 minutes.
MAX_EST_EXEC_TIME	Specifies the maximum execution time (in CPU seconds) allowed for a call. If the optimizer estimates that a call will take longer than MAX_EST_EXEC_TIME, the call is not allowed to proceed and ORA-07455 is issued. If the optimizer does not provide an estimate, this directive has no effect. Default is UNLIMITED. The accuracy of the estimate is dependent on many factors, especially the quality of the optimizer statistics.
UNDO_POOL	Sets a maximum in kilobytes (K) on the total amount of undo for uncommitted transactions that can be generated by a consumer group. Default is UNLIMITED.
MAX_IDLE_TIME	Indicates the maximum session idle time, in seconds. Default is NULL, which implies unlimited.
MAX_IDLE_BLOCKER_TIME	Indicates the maximum session idle time of a blocking session, in seconds. Default is NULL, which implies unlimited.
SWITCH_TIME_IN_CALL	Deprecated. Use SWITCH_FOR_CALL.
MGMT_P1	For a plan with the MGMT_MTH parameter set to EMPHASIS, specifies the CPU percentage to allocate at the first level. For MGMT_MTH set to RATIO, specifies the weight of CPU usage. Default is NULL for all MGMT_Pn parameters.
MGMT_P2	For EMPHASIS, specifies CPU percentage to allocate at the second level. Not applicable for RATIO.
MGMT_P3	For EMPHASIS, specifies CPU percentage to allocate at the third level. Not applicable for RATIO.
MGMT_P4	For EMPHASIS, specifies CPU percentage to allocate at the fourth level. Not applicable for RATIO.
MGMT_P5	For EMPHASIS, specifies CPU percentage to allocate at the fifth level. Not applicable for RATIO.
MGMT_P6	For EMPHASIS, specifies CPU percentage to allocate at the sixth level. Not applicable for RATIO.
MGMT_P7	For EMPHASIS, specifies CPU percentage to allocate at the seventh level. Not applicable for RATIO.
MGMT_P8	For EMPHASIS, specifies CPU percentage to allocate at the eighth level. Not applicable for RATIO.
SWITCH_IO_MEGABYTES	Specifies the number of megabytes of I/O that a session can transfer (read and write) before an action is taken. Default is UNLIMITED. The action is specified by SWITCH_GROUP.
SWITCH_IO_REQS	Specifies the number of I/O requests that a session can execute before an action is taken. Default is UNLIMITED. The action is specified by SWITCH_GROUP.
SWITCH_FOR_CALL	If TRUE, a session that was automatically switched to another consumer group (according to SWITCH_TIME, SWITCH_IO_MEGABYTES, or SWITCH_IO_REQS) is returned to its original consumer group when the top level call completes. Default is NULL.
PARALLEL_QUEUE_TIMEOUT	Specifies the maximum time, in seconds, that a parallel statement can wait in the parallel statement queue before it is timed out.

PARALLEL_SERVER_LIMIT	Specifies the maximum percentage of the parallel execution server pool that a particular consumer group can use. The number of parallel execution servers used by a particular consumer group is counted as the sum of the parallel execution servers used by all sessions in that consumer group.
UTILIZATION_LIMIT	Specifies the maximum CPU utilization percentage permitted for the consumer group. This value overrides any level allocations for CPU (MGMT_P1 through MGMT_P8), and also imposes a limit on total CPU utilization when unused allocations are redistributed. You can specify this attribute and leave MGMT_P1 through MGMT_P8 NULL.
SWITCH_IO_LOGICAL	Number of logical I/O requests that will trigger the action specified by SWITCH_GROUP. As with other switch directives, if SWITCH_FOR_CALL is TRUE, then the number of logical I/O requests is accumulated from the start of a call. Otherwise, the number of logical I/O requests is accumulated for the length of the session.
SWITCH_ELAPSED_TIME	Elapsed time, in seconds, that will trigger the action specified by SWITCH_GROUP. As with other switch directives, if SWITCH_FOR_CALL is TRUE, then the elapsed time is accumulated from the start of a call. Otherwise, the elapsed time is accumulated for the length of the session.
SHARES	Allocates resources among pluggable databases (PDBs) in a multitenant container database (CDB). Also allocates resources among consumer groups in a non-CDB or in a PDB.
PARALLEL_STMT_CRITICAL	Specifies whether parallel statements from the consumer group are critical. When BYPASS_QUEUE is specified, parallel statements from the consumer group are critical. These statements bypass the parallel queue and are executed immediately. When FALSE or NULL (the default) is specified, parallel statements from the consumer group are not critical. These statements are added to the parallel queue when necessary.
SESSION_PGA_LIMIT	Specifies the maximum amount of PGA memory, in megabytes, that can be allocated to each session in a particular consumer group. If a session exceeds the limit, then its process is terminated with an ORA-10260 error.

Etapa 5. Asignar las sesiones a los resource consumer groups.

Antes de habilitar el plan, hay que indicar cómo las sesiones de usuarios son asignadas a los resource consumer groups. Esto se hace creando reglas de mapeo para que automáticamente cada sesión se le asigne un consumer group cuando se inicie. Por ejemplo, la siguiente regla mapea el usuario SCOTT al consume group DEV_GROUP cada vez que se valida en la base de datos:

```
SQL> exec DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (
       ATTRIBUTE => DBMS_RESOURCE_MANAGER.ORACLE_USER,
       VALUE => 'SCOTT',
       CONSUMER_GROUP => 'DEV_GROUP');
```

Etapa 6. Validar la "pending area"

Cuando se ha realizado algún cambio en la pending area, se puede usar en cualquier momento el procedure VALIDATE_PENDING_AREA para asegurar que lo configurado hasta el momento en la pending area es válido. Esta etapa es opcional pero útil ya que facilita el hacer una depuración de errores, especialmente cuando se realizan un gran número de cambios de planes.

En esta fase se chequea que las siguientes reglas son cumplidas:

http://docs.oracle.com/database/122/ADMIN/managing-resources-with-oracle-database-resource-manager.htm#GUID-A5C6447B-BFF5-4CCE-8114-8AD230375082

Etapa 7. Confirmar la "pending area".

Después de que los cambios son validados, llamaremos al procedure SUBMIT_PENDING_AREA para hacer que los cambios se activen. El proceso de confirmación realiza la validación y los cambios son solamente activados si todos los cambios en la "pending area" pasan la validación. Confirmando la "pending area" no activa ningún nuevo plan creado. Si el plan que está actualmente activo es modificado, el plan es reactivado con la definición.

```
SQL> exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

El procedimiento anterior limpia y desactiva la "pending area" después de validar y confirmar los validar los cambios.

Otorgar "Switch Privilege" a los consumer groups

En versiones anteriores a 12c, para poder ser cambiado a un consumer group, un usuario o role tienen que tener permiso explícito sobre el consumer group.

A partir de versión 12c, no se requiere otogar "switch privilege" de forma explícita cuando:

- Hay una especificación de mapeo al consumer group especificado con el procedure SET_CONSUMER_GROUP_MAPPING y la sesión es asignada al consumer group debido a que hay una regla en el mapping que lo envía a ese consumer group.
- La sesión es asignada a un consumer group debido a que se cumple una condición de switch_group en las especificadas en las directivas del resource plan.

Sin embargo, si se quiere permitir que un usuario pueda cambiar de consumer group de otra forma, por ejemplo usando DBMS_SESSION.SWITCH_CURRENT_CONSUMER_GROUP el usuario sí requiere que el administrador previamente le otorgue privilegios.

En el siguiente ejemplo al usuario SCOTT se le otorga el privilegio de cambiar al consumer group OLTP. Además se le permite que el propio usuario pueda otorgar el mismo privilegio a otros usuarios:

```
SQL> exec DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
(
GRANTEE_NAME => 'SCOTT',
CONSUMER_GROUP => 'OLTP',
GRANT_OPTION => TRUE);
```

Utilizando "PUBLIC" podemos hacer que cualquier usuario pueda ser cambiado al consumer group indicado, en este ejemplo consumer group BATCH_GROUP.

```
SQL> exec DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
(
GRANTEE_NAME => 'PUBLIC',
CONSUMER_GROUP => 'BATCH_GROUP',
GRANT_OPTION => FALSE);
```

De esta forma el usuario SCOTT se le permite cambiar independientemente de la configuración de los mappings realizada, por ejemplo desde una sesión del usuario SCOTT podrá ejecutar:

```
DECLARE
OLD_CONSUMER_GROUP VARCHAR2(50);
BEGIN
DBMS_SESSION.SWITCH_CURRENT_CONSUMER_GROUP('BATCH_GROUP',old_cons
umer_group, initial_group_on_error => TRUE);
end;
/
```

Habilitar el Resource Plan

Para habilitar Oracle Database Resource Manager se utilizará el parámetro de inicialización RESOURCE_MANAGER_PLAN.

Este parámetro especifica el plan que será usado en la instancia actual. Si no se especifica ningún plan, Resource manager no estará habilitado.

En versión 12cR2 por defecto Resource Manager no está habilitado, excepto durante las ventanas de mantenimiento pre configuradas.

El siguiendo comando de SQL cambia el plan de Resource Manager a mydb_plan, y lo activa si no estuviera ya activo:

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'mydb_plan';
```

Para deshabilitar Resource Manager, habría que:

1. Lanzar este comando SQL:

   ```
   ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
   ```

2. Desconfigurar el Resource Manager en todas las ventanas del Oracle Scheduler. Ver más adelante el apartado Oracle Database Resource Manager y ventanas de Oracle Scheduler.

Oracle Database Resource Manager y ventanas de Oracle Scheduler.

Oracle Database Resource Manager se activa automáticamente si una ventana de Oracle Scheduler que especifica un resource plan se abre. Cuando la ventana cierra, el resource plan asociado con la ventana se deshabilita, y el resource plan que estaba corriendo antes de que la ventana abriera se vuelve a activar. En caso de que ningún plan hubiera estado habilitado cuando la ventana abrió, cuando esta cierre deshabilitará Resource Manager. En un entorno de Oracle Real Application Cluster una ventana del Scheduler aplica a todas las instancias, por tanto el resource plan de la ventana se habilitará en cada instancia.

Por defecto en versión 12cR2 un conjunto de tareas automáticas de mantenimiento ejecutadas durante las ventanas de mantenimiento, las cuales son ventanas predefinidas en el Scheduler miembros del grupo de ventanas MAINTENANCE_WINDOW_GROUP y que por defecto especifican el DEFAULT_MAINTENANCE_PLAN resource plan. Por tanto, el Resource Manager es activado por defecto durante las ventanas de mantenimiento.

El grupo de ventanas predefinido MAINTENANCE_WINDOW_GROUP consiste en siete ventanas:

Window Name	Description
MONDAY_WINDOW	Starts at 10 p.m. on Monday and ends at 2 a.m.
TUESDAY_WINDOW	Starts at 10 p.m. on Tuesday and ends at 2 a.m.
WEDNESDAY_WINDOW	Starts at 10 p.m. on Wednesday and ends at 2 a.m.
THURSDAY_WINDOW	Starts at 10 p.m. on Thursday and ends at 2 a.m.
FRIDAY_WINDOW	Starts at 10 p.m. on Friday and ends at 2 a.m.
SATURDAY_WINDOW	Starts at 6 a.m. on Saturday and is 20 hours long.
SUNDAY_WINDOW	Starts at 6 a.m. on Sunday and is 20 hours long.

El DEFAULT_MAINTENANCE_PLAN está definido con esta asignación de recursos:

Consumer Group/subplan	Level 1	Maximum Utilization Limit
ORA$AUTOTASK	5%	90
OTHER_GROUPS	20%	
SYS_GROUP	75%	

En este plan, cualquier sesión en el consumer group SYS_GROUP (Sesiones de los usuarios SYS y SYSTEM) tendrá más prioridad. Cualquier recurso que no sea usado por sesiones del grupo SYS_GROUP serán compartidas por las sesiones que pertenezcan a otros los consumer groups y subplanes indicados en el plan. De esta asignación, el 5% va a tareas de mantenimiento y el 20% irán a sesiones de usuario.

El límite de utilización máxima para el subplan ORA$AUTOTASK 90. Por tanto, aunque la CPU esté "idle", este grupo/plan no se le asignará más del 90% de los recursos de CPU. Para reducir o aumentar la asignación a las tareas de mantenimiento, podemos ajustar la configuración del plan DEFAULT_MAINTENANCE_PLAN.

Podemos modificar estas ventanas de mantenimiento para que se use un resource plan distinto. Por ejemplo si queremos podemos poner nuestro propio resource plan en la ventana usando el procedure dbms_scheduler.set_attribute:

```
dbms_scheduler.set_attribute('MONDAY_WINDOW',     'RESOURCE_PLAN',
'myplan');
```

NOTA: En caso de que se cambie el plan en las ventanas de mantenimiento asegurar que este plan incluye el subplan `ORA$AUTOTASK`.

También podemos deshabilitar resource manager plan DEFAULT_MAINTENANCE_PLAN durante las ventanas de mantenimiento establecidas en el Oracle Scheduler para siempre con estos comandos:

```
execute
dbms_scheduler.set_attribute('SATURDAY_WINDOW','RESOURCE_PLAN',''
);

execute
dbms_scheduler.set_attribute('SUNDAY_WINDOW','RESOURCE_PLAN','');

execute
dbms_scheduler.set_attribute('MONDAY_WINDOW','RESOURCE_PLAN','');

execute
dbms_scheduler.set_attribute('TUESDAY_WINDOW','RESOURCE_PLAN','')
;

execute
dbms_scheduler.set_attribute('WEDNESDAY_WINDOW','RESOURCE_PLAN','
');

execute
dbms_scheduler.set_attribute('THURSDAY_WINDOW','RESOURCE_PLAN',''
);

execute
dbms_scheduler.set_attribute('FRIDAY_WINDOW','RESOURCE_PLAN','');
```

Deshabilitando los cambios de planes realizados en las ventanas del Oracle Scheduler

En algunos casos, el cambio automático de planes de Resource Manager realizadas por las ventanas del Scheduler podría ser no deseado. Por ejemplo, si existe una tarea muy importante que tiene que finalizar y se activa un plan de Resource Manager para favorecer la prioridad de esta tarea, no querríamos que el plan cambiara. Sin embargo, ya que una ventana del Scheduler podría activarse tras haber activado el plan deseado, el plan podría ser cambiado mientras la tarea crítica está todavía corriendo.

Para evitar esta situación, podemos configurar el parámetro RESOURCE_MANAGER_PLAN con el nombre del plan que queremos activar precedido con el valor "FORCE:" en el nombre:

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'FORCE:mydb_plan';
```

Usando este prefijo FORCE: indicaremos que el plan actual sólo será cambiado cuando el administrador cambie el valor del parámetro de inicialización RESOURCE_MANAGER_PLAN. Esta restricción puede saltarse si volvemos a lanzar el comando quitando "FORCE:".

El procedure DBMS_RESOURCE_MANAGER.SWITCH_PLAN ofrece la misma funcionalidad.

Oracle Resource Manager con RAC

En un entorno Oracle Real Application Cluster, Oracle Database Resource Manager es manejado de forma independiente por cada instancia.

Cada instancia miembro del cluster puede asignar diferentes resource plans para adaptarse a la posibilidad de diferentes carga de trabajo en cada instancia. Por ejemplo, en un cluster de 2 nodos, una instancia que asigne la mayoría de los recursos a usuarios del online, mientras que otra instancia pueda usar un plan diferente asignando sus recursos a operaciones batch. Esto permite tener diferentes aplicaciones o usuarios en la misma base datos sin que haya impacto entre ellos.

Sin embargo, en la mayoría de los despliegues, la recomendación es configurar el mismo plan en todas las instancias de la base de datos RAC.

Gestión de múltiples bases de datos en un único servidor. Instance Caging

Para utilizar de forma eficiente los recursos hardware disponibles, es habitual consolidar múltiples instancias de bases de datos Oracle en un único servidor multi-CPU, multicore. Sin embargo, cuando se ejecutan múltiples instancias en el mismo servidor, las instancias competirán por los recursos de CPU del sistema. Una instancia de base datos que haga un consumo muy intensivo de CPU podría degradar de forma muy significativa el rendimiento en otras instancias. Por ejemplo, un servidor con múltiples cpus con dos instancias de base de datos corriendo, en la que una instancia en un periodo de carga mensual obtiene la mayoría de las CPUs, provocando una degradación del rendimiento en la otra instancia.

Para poder gestionar estas situaciones en las que hay posibilidad de escasez de recursos de CPU se introdujo en versión Oracle Database 11g Release 2 (11.2.0.1) una funcionalidad llamada "Instance Caging".

"Instance caging" proporciona una forma simple de limitar el consumo de CPU para toda la instancia de base de datos. "Instance Caging" utiliza un parámetro de inicialización para limitar el número de CPUs que una instancia puede usar simultáneamente. En un servidor con 16-cpus con dos instancias de bases de datos corriendo, se puede usar la funcionalidad "instance caging" para limitar por ejemplo el número de CPUs a 8 para cada una de las 2 instancias, haciendo que la probabilidad de que una instancia interfiera con la otra se mínima. Así, el uso de "Instance caging" y Oracle Resource Manager juntos ofrecen una forma simple y efectiva de poder gestionar múltiples instancias en un único servidor.

La funcionalidad de "Instance Caging" se basa en tener activado el Oracle Database Resource Manager y en el parámetro de inicialización CPU_COUNT para indicar el límite de cantidad de CPU que la instancia puede consumir.

Hay dos típicos enfoques de configuración:

- Exceso de aprovisionamiento (Overprovisioning): En este enfoque, la suma de los límites de CPU para cada instancia excede el número real de CPUs del sistema. Por ejemplo, en un servidor con 16 cpus, un administrador podría limitar cada instancia a 10 cpus. Cuando un servidor está configurado de esta forma, las instancias pueden impactar entre ellas, pero con Instance Caging el impacto del rendimiento se puede limitar. Por otro lado, si una de las instancias sufre una carga más alta, tiene CPUs disponibles para usarlas. Este acercamiento es razonable para sistemas productivos no críticos con baja carga o sistemas no críticos como desarrollo o pruebas, ya que una o más instancias podrían estar poco cargadas o inactivas.
- Separación (Partitioning): En este enfoque, la suma de todas las asignaciones es igual al número de CPUs en el servidor. Por ejemplo en un servidor con 16 CPUs con 2 instancias, un administrador podría limitar cada instancia a ocho cpus, o podría permitir a una instancia usar 10 cpus dejando 6 cpus para la otra instancia. Dedicando recursos de CPU a cada instancia de base de datos, aseguramos que la carga de una instancia no tenga impacto en la otra y que cada instancia tenga un rendimiento predecible. Este enfoque es ideal para sistemas de producción críticos, sin embargo, cuando la base de datos está inactiva o con baja carga, los recursos de cpu no utilizados no pueden ser usados por otras instancias, incluso cuando otras instancias tienen una falta de recursos de CPU.

Habilitando Instance Caging

Para habilitar "Instance caging", hay que realizar lo siguiente en la instancia de base de datos:

1. Habilitar Oracle Database Resource Manager asignando a la instancia cualquier resource plan (incluido el default_plan), y asegurar que el plan activado tiene las directivas de CPU activadas, tales como MGMT_P1 a MGMT_P8 o el UTILIZATION_LIMIT.
2. Configurar el parámetro dinámico CPU_COUNT al número requerido de CPUs

Por ejemplo:

```
% uname -X | grep CPU
NumCPU = 32
% sqlplus / as sysdba
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'RANDOM_PLAN';
SQL> ALTER SYSTEM SET CPU_COUNT = 8 SCOPE=BOTH;
```

Chequeo de que "Instance caging" está habilitado

Para chequear si "Instance caging" está habilitado, ejecutaremos las siguientes consultas:

```
SQL> SHOW PARAMETER CPU_COUNT

SQL> SHOW PARAMETER RESOURCE_MANAGER_PLAN

SQL> SELECT NAME FROM V$RSRC_PLAN WHERE IS_TOP_PLAN='TRUE'
AND CPU_MANAGED='ON';
```

Si la última consulta devuelve una fila, será una indicación de que el "instance caging" está en uso.

Hay que tener en cuenta que el parámetro CPU_COUNT especifica el número de CPUs disponibles para Oracle Database. Por defecto Oracle Database automáticamente inicializa el valor al número de CPUs reportado por el sistema operativo. La base de datos usa el parámetro para calcular los valores por defecto para otros parámetros de inicialización incluyendo el valor mínimo de SGA, el máximo número de parallel servers y el grado de paralelismo. En servidores con un número muy grande de CPUs virtuales, como servidores Oracle Sun SPARC Enterprise T-series, ciertos parámetros son configurados valores muy altos en el arranque de la instancia. Como resultado, la base de datos podría tener un comportamiento ocasionalmente no adecuado al ejecutar ciertas operaciones. Una forma de evitar este comportamiento es manualmente configurar el valor del parámetro CPU_COUNT al número de cores disponibles para la base de datos o a otro valor basándose en datos experimentales. Cuando "Instance caging" no esté habilitado, el parámetro de inicialización CPU_COUNT a un valor distinto al de por defecto no evitará que la instancia de base de datos pueda usar todas las CPUs disponibles en el servidor. La configuración del parámetro simplemente es usado para reducir los valores por defecto de otros parámetros a valores más realistas en el momento de arranque de la instancia.

Monitorización de Oracle Database Resource Manager.

Podemos usar Oracle Enterprise Manager Cloud Control y para seleccionar y activar planes, así como mostrar estadísticas y gráficas que muestren el estado actual del resource plan activo. También en Oracle Enterprise Manager Database Control En Server -> Resource Manager -> Statistics.

Igualmente podemos usar las siguientes vistas dinámicas de rendimiento que nos ayudarán a monitorizar el resultado de la configuración activada de Oracle Database Resource Manager:

V$RSRC_PLAN

V$RSRC_CONSUMER_GROUP

V$RSRC_SESSION_INFO

V$RSRC_PLAN_HISTORY

V$RSRC_CONS_GROUP_HISTORY

V$RSRCMGRMETRIC

V$RSRCMGRMETRIC_HISTORY

Estas vistas nos permiten:

- Conocer información del estado actual de Resource Manager
- Historia de las activaciones de resource plans
- Estadísticas actuales e históricas de los consumos y las esperas por cpu que genera Resource Manager para cada sesión y para cada consumer group.

La vista V$RSRC_CONSUMER_GROUP proporciona medidas muy útiles para medir el rendimiento como la cantidad acumulada de tiempo de CPU consumido, la cantidad acumulada de tiempo esperando por CPU debido a la gestión de Resource Manager, el número acumulado de esperas por CPU para todas las sesiones en cada consumer group, número de peticiones de lectura y escritura, etc. No se incluyen en estas medidas información de esperas por otros conceptos como contención por latches, bloqueos, esperas de IO, etc.

También, la vista `V$SQL_MONITOR` donde Oracle monitoriza por defecto de forma automática las sentencias SQL que se ejecutan en paralelo o que consumen al menos 5 segundos de CPU o de tiempo I/O, ofrece información referente a la última acción realizada por Resource Manager para la sesión en las siguientes columnas: RM_CONSUMER_GROUP, RM_LAST_ACTION, RM_LAST_ACTION_REASON, y RM_LAST_ACTION_TIME.

Podemos monitorizar las estadísticas de ejecución de las sentencias SQL usando las vistas V$SQL_MONITOR, V$SQL_PLAN_MONITOR, y V$SQL_MONITOR_SESSTAT. Igualmente podemos ver esta información accediendo a la página SQL Monitor en EM Cloud Control y EM Database Express 12c.

Chequeo de que Oracle Database Resource Manager está activado

Para determinar si Resource Manager está activado:

```
SQL> select name, cpu_managed from v$rsrc_plan where is_top_plan
= 'TRUE';
```

Esta consulta devolverá el nombre del resource plan actual. Si no devuelve filas, Resource Manager está deshabilitado. La columna "cpu_managed" especifica si Resource Manager está gestionando CPU.

Revisión histórico de activación de Resource Manager

Para determinar la historia del uso de Resource Manager:

```
SQL> select name,
to_char(start_time, 'MON DD HH24:MI') start_time,
to_char(end_time, 'MON DD HH24:MI') end_time,
window_name
from v$rsrc_plan_history order by start_time;
```

Esta sentencia devuelve el nombre del plan, la hora a la que fue habilitado y deshabilitado, y la ventana del scheduler que fue usada para habilitarlo en caso de que hubiera usado una ventana. Cada fila corresponde con una ocurrencia de habilitar o deshabilitar Resource Manager. En el histórico se mantienen las últimas 16 ocurrencias.

Revisión de cómo Resource Manager fue habilitado

En la consulta anterior, la columna window_name indica la ventana del scheduler window que automáticamente habilitó el resource plan. Si window_name es NULL, entonces el resource plan fue habilitado usando el parámetro "resource_manager_plan".

Para determinar la ventana del scheduler que tiene asociado un resource plan:

```
SQL> select window_name, resource_plan, active
from dba_scheduler_windows
where resource_plan is not null and enabled = 'TRUE';
```

Si en "active" marca TRUE, indica que la ventana está actualmente abierta y el resource plan está habilitado.

Para determinar si hay trabajos lanzados por el scheduler:

```
SQL> select owner, job_name, SESSION_ID, RESOURCE_CONSUMER_GROUP
from dba_scheduler_running_jobs;
```

En la columna RESOURCE_CONSUMER_GROUP de la consulta anterior indicará el consumer group asociado al trabajo lanzado por el scheduler.

Revisión del Consumer group usado por las sesiones

La siguiente sentencia muestra el consumer group actual para todas las sesiones en base de datos:

```
SQL> select sid, resource_consumer_group from v$session;
```

La siguiente sentencia muestra las sesiones con la información del consumer group actual y su consumo para las sentencias largas actualmente en ejecución:

```
SQL>
set lines 200 pages 200
col sql_text for a50
col username for a12
col service_name for a12
col RM_CONSUMER_GROUP for a12
col sid for 9999
```

```
col key for 99999999999999
select key, sid, username, SERVICE_NAME, RM_CONSUMER_GROUP,
sql_id, sql_plan_hash_value plan_hash, elapsed_time, cpu_time,
buffer_gets, disk_reads, substr(sql_text,1,50) sql_text
from v$sql_monitor
where status = 'EXECUTING';
```

Causas por las que una sesión no usa un consumer group

Cuando se detecta que una sesión no está en el consumer group esperado podría deberse a varias razones:

- Consumer Group no configurado en el resource plan.

Si una sesión está mapeada o ha sido manualmente movida a un consumer group que no es parte del resource plan en uso, la sesión será cambiada al consumer group, "OTHER_GROUPS". Si las sesiones están siendo asignadas a los consumer groups usando reglas de mapeo, la siguiente consulta puede ser usada para determinar el consumer group que las reglas de mapeo seleccionan, el atributo de mapeo que fue usado, y el consumer group donde inicialmente la sesión arrancó. Si el consumer group mapeado difiere del consumer group inicial, significará que el consumer group mapeado no era parte del resource plan.

```
SQL> select r.sid,
   r.mapped_consumer_group,
   r.mapping_attribute,
   c.consumer_group original_consumer_group
   from v$rsrc_session_info r, dba_rsrc_consumer_groups c
   where r.orig_consumer_group_id = c.consumer_group_id;
```

- Consumer Group Switch

Si el consumer group de una sesión no coincide con su consumer group inicial, indicará que la sesión ha sido movida entre consumer groups. La siguiente consulta detecta las sesiones que automáticamente han sido movidas:

```
SQL> select r.sid,
c1.consumer_group original_consumer_group,
c2.consumer_group current_consumer_group
from v$rsrc_session_info r, dba_rsrc_consumer_groups c1,
dba_rsrc_consumer_groups c2
where r.orig_consumer_group_id = c1.consumer_group_id
and r.current_consumer_group_id = c2.consumer_group_id
and              r.orig_consumer_group_id              !=
r.current_consumer_group_id;
```

Una sesión puede ser movida entre consumer groups por varias razones:

1. El consumer group inicial no es parte del resource plan actual. En este caso, el consumer group será "OTHER_GROUPS".

2. La sesión ha podido ser cambiada automáticamente por el resource plan porque se haya alcanzado algún límite indicado en alguna directiva configurada. La siguiente sentencia indica si el resource plan tiene alguna directiva de este estilo:

```
SQL> select plan, group_or_subplan, switch_time,
switch_io_megabytes, switch_io_reqs
from dba_rsrc_plan_directives
where plan in (select name from v$rsrc_plan)
and (switch_time is not null
or switch_io_megabytes is not null
or switch_io_reqs is not null);
```

3. Cambio manual de Consumer Group:

   Se puede manualmente mover una sesión a un consumer group usando los siguientes procedures:

   dbms_session.switch_consumer_group()

   dbms_resource_manager.switch_consumer_group_for_user()

   dbms_resource_manager.switch_consumer_group_for_sess()

   El cambio no se produce inmediatamente, ocurrirá cuando la sesión ejecute la siguiente llamada.

- Falta de privilegios

En versiones anteriores a 12c esto era habitual porque aunque la sesión estuviera mapeada para usar un consumer group, si no tenía permiso explicito para usarlo la regla de mapeo se ignoraba, en tal caso la sesión era enviada al consumer group OTHERS_GROUP. En 12c sólo aplica la necesidad de explícitamente dar privilegios cuando el cambio de consumer group venga realizado con:

dbms_session.switch_consumer_group()

dbms_resource_manager.switch_consumer_group_for_user()

dbms_resource_manager.switch_consumer_group_for_sess()

En tal caso la siguiente consulta mostrará los permisos para todos los consumer groups. Para verificar los permisos para un usuario concreto, se puede especificar con una clausula `where grantee ="nombreusuario"`.

```
SQL> select grantee, granted_group from
DBA_RSRC_CONSUMER_GROUP_PRIVS order by granted_group;
```

En la sección anterior viene indicado el procedimiento para dar permisos.

Monitorización de la gestión de CPU. Evento de espera "resmgr:cpu quantum".

Podemos monitorizar el efecto de Resource Manager en la instancia de base de datos de forma global. También podemos monitorizar el efecto de Resource Manager en cada consumer group.

Determinando cuánto está actuando Resource Manager en la gestión de cpu de la instancia

Oracle permite varias formas de monitorizar la gestión de los recursos de CPU realizada por Resource Manager:

- Midiendo la cantidad de eventos de espera

  El evento de espera "resmgr: cpu quantum" ocurre cuando un proceso está esperando en base de datos por ser asignado por Resource Manager a una cpu. El tiempo de espera es el tiempo que la sesión esperó para coger cpu.

  Para determinar cuánto está actuando Resource Manager en la gestión de la cpu se puede monitorizar el evento de espera "resmgr:cpu quantum".

  Estas esperas pueden observarse generando un AWR report y examinando el apartado "Foreground Wait Events".

  Esta espera es del tipo de wait class "scheduler". Por tanto Resource Manager también puede ser monitorizado cuantificando las esperas de la clase "scheduler". Esta clase también agrupa las esperas producidas por Resource Manager debido sobrepasar límites de concurrencia.

  La siguiente consulta monitoriza las esperas del tipo wait class "scheduler". "dbtime_in_wait" muestra el porcentaje de tiempo de base de datos que se va en esperas de Resource Manager. "time_waited" muestra el valor actual del tiempo esperado en microsegundos.

```
SQL> select to_char(h.begin_time, 'HH:MI') time,
h.average_waiter_count, h.dbtime_in_wait, h.time_waited
from v$waitclassmetric_history h, v$system_wait_class c
where h.wait_class_id = c.wait_class_id and c.wait_class =
'Scheduler'
order by h.begin_time;
```

  Cuando una instancia de base de datos recibe una carga de trabajo que provoca un consumo que excede la capacidad de CPU del servidor, se verá el evento "resmgr:cpu quantum" en el top de los eventos de espera. Aunque Resource Manager esté provocando estas esperas, es importante resaltar que el rendimiento general de la instancia será igual que cuando Resource Manager está deshabilitado. Sin Resource Manager, el tiempo gastado en esperas por "resmgr:cpu quantum" será en su lugar gastado en esperas en la run queue del sistema operativo. La ventaja de Resource Manager es que en base a las directivas de cpu que hayamos definido podremos gestionar qué sesiones son las que se verán afectadas por las esperas, dándoles más prioridad a unas que a otras. Sin Resource Manager las esperas en la run queue del sistema operativo serán repartidas entre todas las sesiones.

- Usando las vistas de Resource Manager

  Para determinar cuánto está actuando Resource Manager en la gestión de la cpu se pueden monitorizar las vistas de Resource Manager. La siguiente sentencia nos dará una comparación minuto a minuto de:

  - total: la cantidad de CPU time disponible en el servidor
  - db_total: la cantidad de CPU time disponible para la instancia de base de datos (será menor si Instance Caging está habilitado)
  - consumed: la cantidad de CPU time consumido
  - throttled: la cantidad de tiempo que los procesos esperaron por la actuación de Resource Manager.

  En todos los casos, el tiempo está expresado en segundos:

```
SQL> select to_char(begin_time, 'HH:MI') time,
60 * (select value from v$osstat where stat_name =
'NUM_CPUS') total,
60 * (select value from v$parameter where name =
'cpu_count') db_total,
sum(cpu_consumed_time) / 1000 consumed,
sum(cpu_wait_time) / 1000 throttled
from gv$rsrcmgrmetric_history
group by begin_time order by begin_time;
```

  Si el tiempo de CPU consumido es relativamente pequeño respecto del total de CPU, indicará que el servidor no está siendo altamente utilizado por la instancia.

- Revisando la media de sesiones afectadas por minuto

  Para determinar cuánto está actuando Resource Manager en la gestión de la cpu también se puede cuantificar el número medio de sesiones que estaban esperando debido a la actuación de Resource Manager. La siguiente consulta muestra minuto a minuto una comparativa de:

  - num_cpus: el número de CPUs
  - num_db_cpus: el número de CPUs disponibles para esta instancia de base de datos (será menor si Instance Caging está habilitado)
  - avg_running: media de sesiones Oracle ejecutándose
  - avg_throttled: media de sesiones Oracle esperando debido a la actuación de Resource Manager.

```
SQL> select to_char(begin_time, 'HH:MI') time,
(select value from v$osstat where stat_name = 'NUM_CPUS')
num_cpus,
(select value from v$parameter where name = 'cpu_count')
num_db_cpus,
sum(cpu_consumed_time) / 60000 avg_running,
sum(cpu_wait_time) / 60000 avg_throttled
from v$rsrcmgrmetric_history
```

```
group by begin_time order by begin_time;
```

Si la media de sesiones afectadas es igual al número de CPUs, entonces de forma media cada CPU tuvo una sesión esperando para ejecutarse. Esto sería un ejemplo de una instancia de base de datos y servidor sobrecargado.

En caso de que un sistema esté en estas condiciones, si el rendimiento de la aplicación no es adecuado, se debería ajustar el resource plan para darle más prioridad de asignación de CPU al consumer group asociado a la aplicación. Este ajuste se verá más claro en la siguiente sección. Si el consumer group ya tiene la mayoría de la CPU asignada, la única opción es físicamente añadir más CPUs al sistema, asumiendo que la aplicación y la base de datos ya no pueden ser optimizadas más a nivel de optimización de base de datos (tuning de SQL, ajuste de parámetros de la instancia, ajuste de SGA, etc).

Determinando cuánto está actuando Resource Manager en la gestión de cpu de la instancia a nivel de Consumer Groups

Una vez que un resource plan ha sido habilitado, el rendimiento de los consumer groups debería ser monitorizado y las asignaciones de CPU deberían ser ajustadas si fuera necesario.

La siguiente sentencia mostrará minuto a minuto una comparación agrupando por consumer group de:

- total: la cantidad de CPU time disponible en el sistema.
- db_total: la cantidad de CPU time disponible para la instancia de base de datos (será menor si "Instance Caging" está habilitado)
- consumed: la cantidad de CPU time consumida por un consumer group
- cpu_utilization: el porcentaje de CPU disponible de esta base de datos consumida por el consumer group
- throttled: la cantidad de tiempo que el consumer group esperará por la actuación de Resource Manager.

En todos los casos el tiempo está expresado en segundos.

```
SQL> select to_char(begin_time, 'HH:MI') time,
consumer_group_name,
60 * (select value from v$osstat where stat_name =
'NUM_CPUS') total,
60 * (select value from v$parameter where name =
'cpu_count') db_total,
cpu_consumed_time / 1000 consumed,
cpu_consumed_time / (select value from v$parameter where
name =
'cpu_count') / 600 cpu_utilization,
cpu_wait_time / 1000 throttled
from v$rsrcmgrmetric_history
order by begin_time;
```

Chequeos de consumos actuales de las sesiones

Con la siguiente consulta obtendremos información del actual del consumo por sesión de todas las sesiones en base de datos exceptuando los procesos background:

```
set lines 150 pages 500

SELECT se.sid sess_id, co.name consumer_group,
 se.state,  se.consumed_cpu_time,  se.current_active_time,  se.cpu_wait_time,
se.queued_time
 FROM v$rsrc_session_info se, v$rsrc_consumer_group co
 WHERE  se.current_consumer_group_id  =  co.id  and  co.name  not  like
'_ORACLE_BACKGROUND_GROUP_';
```

Script con chequeo sobre la configuración de Resource Manager y estadísticas

```
PROMPT RESOURCE MANAGER CONFIGURATION AND STATS
PROMPT ========================================

SET LINESIZE 250
set pagesize 100

BREAK ON PLAN SKIP 1 NODUP ON STATUS NODUP ON TYPE NODUP ON REPORT

COL PLAN                    FOR A18
COL STATUS                  FOR A7
COL GROUP_OR_SUBPLAN        FOR A26    HEADING 'RCG/subPlan'
COL TYPE                    FOR A14
COL CPU_P1                  FOR 999    HEADING 'CPU|N 1'
COL CPU_P2 LIKE CPU_P1                 HEADING 'CPU|N 2'
COL CPU_P3 LIKE CPU_P1                 HEADING 'CPU|N 3'
COL CPU_P4 LIKE CPU_P1                 HEADING 'CPU|N 4'
COL CPU_P5 LIKE CPU_P1                 HEADING 'CPU|N 5'
COL CPU_P6 LIKE CPU_P1                 HEADING 'CPU|N 6'
COL CPU_P7 LIKE CPU_P1                 HEADING 'CPU|N 7'
COL CPU_P8 LIKE CPU_P1                 HEADING 'CPU|N 8'
COL ACTIVE_SESS_POOL_P1     FOR 9990   HEADING 'Sess|Pool'
COL PARALLEL_DEGREE_LIMIT_P1 FOR   90  HEADING 'DOP|Lím'
COL MAXEXECTIME             FOR A9     HEADING 'Máx Exec|Time HMS'
COL MAXIDLETIME             FOR A9     HEADING 'Máx idle|Time HMS'
COL MAXIDLETIMEBLOCK       FOR A9     HEADING 'Blocking idle|Time HMS'
COL  SWITCH_DIRECTIVES_CPU            FOR A20      HEADING 'Switch secs cpu|2
Group[Estim]'
COL  SWITCH_DIRECTIVES_IO_LOGICAL            FOR A20      HEADING 'Switch IO
Logical|2 Group'
COL SWITCH_DIRECTIVES_IO_MEGABYTES         FOR A20    HEADING 'Switch IO MB|2
Group'
COL SWITCH_DIRECTIVES_ELAPSED        FOR A20    HEADING 'Switch secs elapsed|2
Group'
COL UNDO_POOL_MB            FOR 99G990 HEADING 'UNDO|MB'
COL SESSION_PGA_LIMIT       FOR 99G990 HEADING 'PGA|MB'

COMPUTE  SUM  LABEL  'TOTAL  CPU/PLAN'  OF  CPU_P1  CPU_P2  CPU_P3  CPU_P4  CPU_P5
CPU_P6 CPU_P7 CPU_P8 ON PLAN

SELECT    DECODE(PLAN,NAME,'<<'||PLAN||'>>',PLAN)    PLAN,    STATUS,    TYPE,
GROUP_OR_SUBPLAN,
       CPU_P1,
       CPU_P2,
       CPU_P3,
-- SWITCH_FOR_CALL,
```

```
       ACTIVE_SESS_POOL_P1, PARALLEL_DEGREE_LIMIT_P1,
       DECODE(MAX_EST_EXEC_TIME,NULL,NULL,MAX_EST_EXEC_TIME) MAXEXECTIME,
           DECODE(MAX_IDLE_TIME,NULL,NULL,MAX_IDLE_TIME) MAXIDLETIME,
           DECODE(MAX_IDLE_BLOCKER_TIME,NULL,NULL,MAX_IDLE_BLOCKER_TIME)
MAXIDLETIMEBLOCK,

DECODE(SWITCH_TIME,NULL,NULL,SWITCH_TIME||'>'||SWITCH_GROUP||DECODE(SWITCH_EST
IMATE,'TRUE','*',NULL)) SWITCH_DIRECTIVES_CPU,

DECODE(SWITCH_ELAPSED_TIME,NULL,NULL,SWITCH_ELAPSED_TIME||'>'||SWITCH_GROUP)
SWITCH_DIRECTIVES_ELAPSED,

DECODE(SWITCH_IO_LOGICAL,NULL,NULL,SWITCH_IO_LOGICAL||'>'||SWITCH_GROUP)
SWITCH_DIRECTIVES_IO_LOGICAL,

DECODE(SWITCH_IO_MEGABYTES,NULL,NULL,SWITCH_IO_MEGABYTES||'>'||SWITCH_GROUP)
SWITCH_DIRECTIVES_IO_MEGABYTES,
       ROUND(UNDO_POOL/1024) UNDO_POOL_MB,
       ROUND(SESSION_PGA_LIMIT) SESSION_PGA_LIMIT
  FROM DBA_RSRC_PLAN_DIRECTIVES LEFT OUTER JOIN V$RSRC_PLAN ON (PLAN=NAME)
 WHERE PLAN LIKE 'RSCR_PLAN_TEST' -- Solo nuestros planes de recursos
 ORDER BY PLAN, CPU_P1 DESC, CPU_P2 DESC, CPU_P3 DESC, CPU_P4 DESC, CPU_P5
DESC, CPU_P6 DESC, CPU_P7 DESC, CPU_P8 DESC
/

COL NAME                     LIKE GROUP_OR_SUBPLAN
COL ACTIVE_SESSIONS          FOR    999G990   HEADING 'Current|#Sess'
COL EXECUTION_WAITERS        FOR    999G990   HEADING 'Current|Wait|4 CPU'
COL QUEUE_LENGTH             FOR    999G990   HEADING 'Current|Queued|Sess'
COL  REQUESTS                                    FOR  99G999G990      HEADING
'Cumulative|Executed|Requests'
COL   CPU_WAIT_TIME                               FOR   99G999G990D0   HEADING
'Cumulative|Seconds|Waited'
COL CPU_WAITS                FOR 99G999G990   HEADING 'Cumulative|Waits'
COL  CONSUMED_CPU_TIME             FOR  99G999G990D0  HEADING  'Cumulative|Used
CPU|seconds'
COL YIELDS                   FOR 99G999G990   HEADING 'Cumulative|#Yields'
COL CURRENT_UNDO_CONSUMPTION FOR 99G999G990   HEADING 'Current|KB UNDO used'

COMPUTE SUM LABEL 'Total' OF ACTIVE_SESSIONS EXECUTIONS_WAITERS QUEUE_LENGTH
REQUESTS      CPU_WAIT_TIME      CPU_WAITS      CONSUMED_CPU_TIME      YIELDS
CURRENT_UNDO_CONSUMPTION ON REPORT

SELECT  NAME,  ACTIVE_SESSIONS,  EXECUTION_WAITERS,  QUEUE_LENGTH,  REQUESTS,
ROUND(CPU_WAIT_TIME/1000,1) CPU_WAIT_TIME,
       CPU_WAITS,  ROUND(CONSUMED_CPU_TIME/1000,1)  CONSUMED_CPU_TIME,  YIELDS,
CURRENT_UNDO_CONSUMPTION
  FROM V$RSRC_CONSUMER_GROUP
/


COL NAME                     LIKE GROUP_OR_SUBPLAN
COL ACTIVE_SESSIONS          FOR    999G990   HEADING 'Current|#Sess'
COL EXECUTION_WAITERS        FOR    999G990   HEADING 'Current|Wait|4 CPU'
COL QUEUE_LENGTH             FOR    999G990   HEADING 'Current|Queued|Sess'
COL  REQUESTS                                    FOR  99G999G990      HEADING
'Cumulative|Executed|Requests'
COL   CPU_WAIT_TIME                               FOR   99G999G990D0   HEADING
'Cumulative|Seconds|Waited'
COL CPU_WAITS                FOR 99G999G990   HEADING 'Cumulative|Waits'
COL  CONSUMED_CPU_TIME             FOR  99G999G990D0  HEADING  'Cumulative|Used
CPU|seconds'
COL YIELDS                   FOR 99G999G990   HEADING 'Cumulative|#Yields'
COL CURRENT_UNDO_CONSUMPTION FOR 99G999G990   HEADING 'Current|KB UNDO used'

COMPUTE SUM LABEL 'Total' OF ACTIVE_SESSIONS EXECUTIONS_WAITERS QUEUE_LENGTH
REQUESTS      CPU_WAIT_TIME      CPU_WAITS      CONSUMED_CPU_TIME      YIELDS
CURRENT_UNDO_CONSUMPTION ON REPORT

SELECT  INST_ID,  NAME,  ACTIVE_SESSIONS,  EXECUTION_WAITERS,  QUEUE_LENGTH,
REQUESTS, ROUND(CPU_WAIT_TIME/1000,1) CPU_WAIT_TIME,
```

```
      CPU_WAITS,  ROUND(CONSUMED_CPU_TIME/1000,1)  CONSUMED_CPU_TIME,  YIELDS,
CURRENT_UNDO_CONSUMPTION
  FROM GV$RSRC_CONSUMER_GROUP ORDER BY NAME
/
```

Errores comunes

A continuación se indican algunos comunes malentendidos sobre Resource Manager:

- Resource Manager no distribuye correctamente la cpu asignada a los consumer groups.

  Esta afirmación es debida a que cuando se mide el porcentaje de utilización de cpu de cada consumer group este no coincide con el porcentaje indicado en el Resource Plan.

  Cuando se monitoriza Resource Manager, es una práctica común comparar la utilización real de CPU por consumer group con la asignación indicada en la configuración del plan (p.e. mgmt_p1, mgmt_p2, mgmt_p3, etc.). Si todos los consumer groups tienen suficiente carga para utilizar de forma complete todas las asignaciones del plan, entonces la utilización real de CPU debería ser muy similar a las asignaciones del plan. Sin embargo, esto no es lo más habitual, debido a varias razones:

  - La utilización de un consumer group podría ser mayor que la asignación indicada en el plan ya que hay otros consumer groups que no están utilizando su asignación.
  - La utilización de cpu de un consumer group podría ser menor que su asignación en el plan debido a que no requiere el consumo de todo lo asignado. En este caso, no deberían haber (o muy pocas) esperas producidas por la acción de Resource Manager en este consumer group.
  - El consumer group tiene configurada la directiva utilization_limit a un valor menor que la asignación del plan.

- Binding Processes

  Resource Manager no asigna procesos a CPUs. Su objetivo no es gestionar las CPUs del sistema si no controlar el uso de CPU de la instancia de base de datos. Si lo que se quiere es restringir una instancia de base de datos a un conjunto específico de CPUs, debería utilizarse otras técnicas a más bajo nivel (particiones lógicas, processor sets, virtual machines, etc).

- Procesos de Backgroup

  Resource Manager no gestiona procesos background a menos que estos no sean críticos y de consumo de cpu intensivo. Los procesos background críticos usan la CPU de forma razonable y no deberían ser la causa de un consumo excesivo o fuera de control de CPU. La mayoría de ellos requieren

un acceso rápido a la CPU y si actuara Resource Manager sobre ellos el rendimiento global de la base de datos sería afectado.

- Threaded CPUs

  Muchos sistemas actualmente usan procesadores hyper-threaded o CMT processors. Para estos sistemas, una única CPU core puede contener múltiples threads, o CPUs. Por ejemplo, un procesador Xeon contiene dos threads, cada uno de los cuales es contado como una CPU, un UltraSparc T2 core contiene 8 threads, cada uno de los cuales es contado como una CPU.

  Por defecto, Resource Manager gestiona la carga de la base de datos basándose en el total de threads en el sistema. Este número es el equivalente al número de CPUs que reporta el sistema operativo. Este comportamiento permite a la instancia de base de datos utilizar de forma completa el sistema. Si se quiere bajar el nivel límite de utilización de la instancia podremos habilitar el "Instance Caging".

Referencias

Los siguientes enlaces muestran información muy útil sobre Oracle Database Resource Manager:

- Oracle Documentation on Oracle Database Resource Manager:

12cR2 - Oracle® Database Administrator's Guide 12cR2 Release 2 (12.2.0.1)

Managing Resources with Oracle Database Resource Manager

http://docs.oracle.com/database/122/ADMIN/managing-resources-with-oracle-database-resource-manager.htm#ADMIN027

Field Code Changed

- MyOracleSupport

  Hay mucha información y ejemplos relativos a Oracle Database Resource Manager en MyOracleSupport.

  - Master Note: Overview of Oracle Resource Manager and DBMS_RESOURCE_MANAGER (Doc ID 1484302.1)

    Desde esta nota hay enlaces a notas que proporcionan más ejemplos de implantación de Oracle Database Resource Manager en diferentes escenarios.

  - Oracle Database Resource Manager Webcast Recording:

    Note 1119407.1 Resource Manager Training (11.2 features included) [Video]

    Note 1456176.1 Oracle Database Resource Manager - Webcast

- Oracle White Papers on Database Resource Manager

Effective Resource Management Using Oracle Database Resource Manager: http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-056-oracledb-rm-419380.pdf

Field Code Changed

Using Oracle Database Resource Manager: http://www.oracle.com/technetwork/database/performance/resource-manager-twp-133705.pdf

Field Code Changed

Database Instance Caging: A Simple Approach to Server Consolidation: http://www.oracle.com/technetwork/database/performance/instance-caging-wp-166854.pdf

Field Code Changed

Implementando Oracle Resource Manager en un entorno de pruebas. Recomendaciones

Antes de implementar una estrategia de Resource Manager, como una primera etapa tendremos que definir cuáles son los objetivos que queremos conseguir. Sin una clara definición de los objetivos de rendimiento será muy complicado poder evaluar y ajustar el rendimiento y el consumo de recursos. Normalmente, estos objetivos vienen derivados de los requerimientos del negocio. Cuando estos se definan deberían ser lo más realistas, razonables y medibles posible.

En esta sección se mostrará con un ejemplo práctico diferentes estrategias de rendimiento implementando Oracle Database Resource Manager. Se incluirán en cada sección recomendaciones y consideraciones a tener en cuenta cuando se implemente un plan de Resource Manager.

Entorno de prueba para la configuración de Resource Manager

El ejemplo consistirá en una implementación de Resource Manager para controlar cómo los recursos son asignados a diferentes aplicaciones. Esto podría ser un ejemplo de cómo Resource Manager puede ser usado para consolidar aplicaciones o bases de datos en una única base de datos en único servidor con múltiples procesadores.

En el ejemplo suponemos que se han consolidado distintas aplicaciones en la misma base de datos y cada una de ellas tiene unos requerimientos de rendimiento distinto. Por ejemplo, una aplicación será crítica y tendrá más prioridad que otras pequeñas aplicaciones con una necesidad de nivel de servicio más baja.

- **Objetivos a alcanzar con Oracle Database Resource Manager (es sólo un ejemplo):**
    - Restringir el consumo total de CPU de la instancia de base de datos. Independientemente de la carga de la base datos, la carga de trabajo en el sistema generada por la base de datos nunca excederá el 80% de CPU, dejando el 20% de CPU para otras aplicaciones compartiendo el servidor.
    - Garantizar un porcentaje de CPU para aplicaciones críticas. Algunas aplicaciones tendrán más prioridad que otras cuando escaseen los recursos de cpu.
    - Establecer límites para evitar sentencias fuera de control. Algunas aplicaciones necesitarán que operaciones muy largas sean abortadas, otras que simplemente se muevan a una zona de cuarentena donde reciban menos recursos.
    - Limitar la CPU para algunas aplicaciones.
    - Limitar el tamaño de PGA para las sesiones de algunas aplicaciones.

- Algunas aplicaciones tendrán un número máximo de sesiones con llamadas concurrentes a la base de datos.
- Algunas aplicaciones cancelarán sesiones que están inactivas durante un largo tiempo y que bloquean otras sesiones.
- Algunas aplicaciones requieren limitar el tamaño de undo generado.

- **Resource Plan:**
  - RSCR_PLAN_TEST

- **Resource Consumer Groups:**
  - CG_Applications_GOLD. Usado por las aplicaciones con la mayor prioridad.
  - CG_Applications_SILVER. Usado por aplicaciones con menos prioridad y que necesitan ser controladas y limitadas.
  - QUARENTINE. Usado como zona de cuarentena para procesos fuera de control ejecutados en otros consumer groups cuando consuman una gran cantidad de recursos inesperada.

- **Mapeo de sesiones con Consumer Groups:**

  En este caso, se usarán servicios de base de datos para mapear conexiones a Resource Groups:
  - Aplicaciones que usen el servicio application1 irán al consumer group CG_Applications_GOLD.
  - Aplicaciones que usen los servicios application2 y application3 irán al consumer group CG_Applications_SILVER.
  - Aplicaciones usando otros tipos de conexión sin estos servicios irán a OTHER_GROUPS.
  - Un usuario concreto llamado OSS será asignado directamente al Resource Group CG_Applications_GOLD independientemente del Servicio usado.

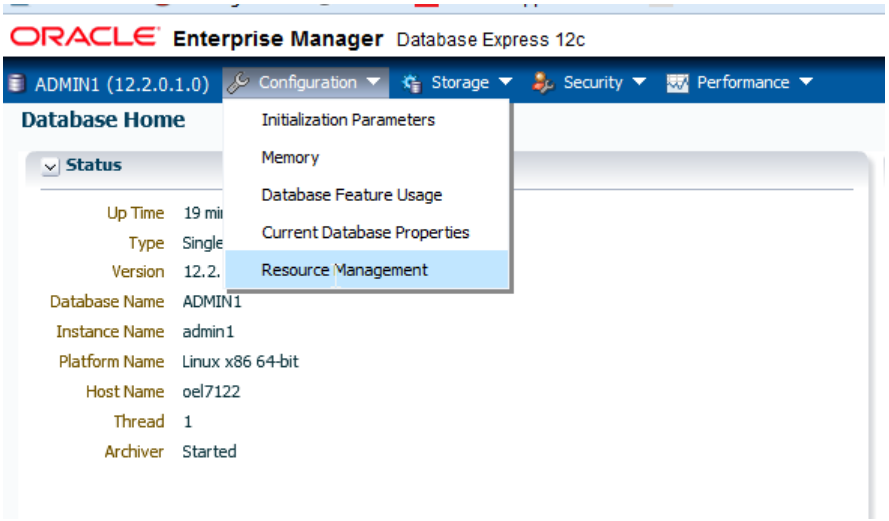
Creación del Resource Plan RSCR_PLAN_TEST

Para crear y configurar el entorno de Resource Manager en versión 12.2 podremos usar Oracle Enterprise Manager Database Express 12c, Oracle Enterprise Manager Cloud Control 12c/13c, SQL Developer o la API PLSQL DBMS_RESOURCE_MANAGER.

En nuestro ejemplo realizaremos la configuración del plan de recursos con Oracle Enterprise Manager Database Express 12c, aunque también se indicará el uso de la API PLSQL DBMS_RESOURCE_MANAGER en aquellas directivas que no cubre EM Database Express 12c .

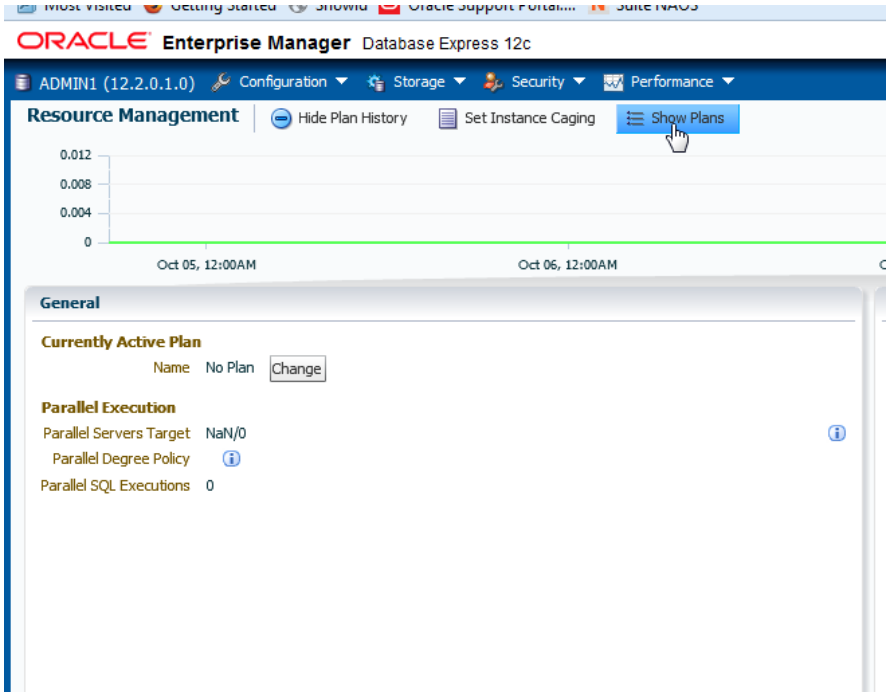
A continuación se irán mostrando en cada paso de la configuración las pantallas de la herramienta.

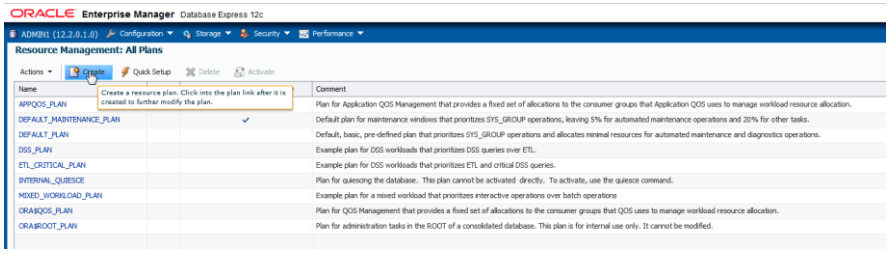
La administración de Resource Manager en Oracle Enterprise Manager Database Express 12c está en Configuration -> Resource Management

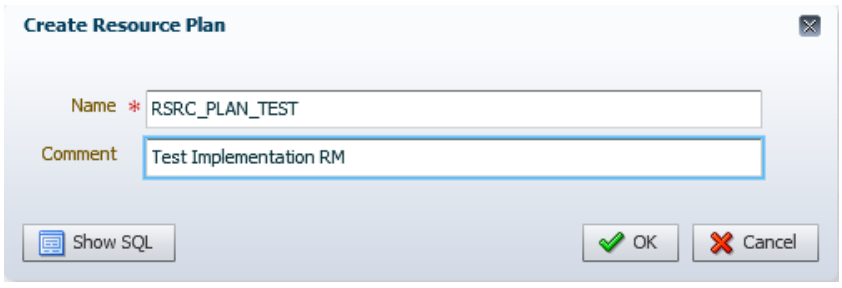


1. Creación Resource Plan

Desde la pantalla de Resource Management se le dará a CREATE para crear el Resource Plan RSCR_PLAN_TEST



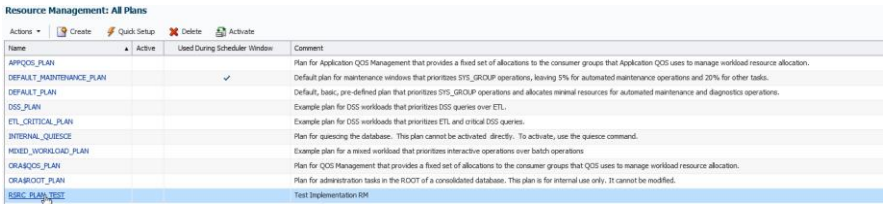


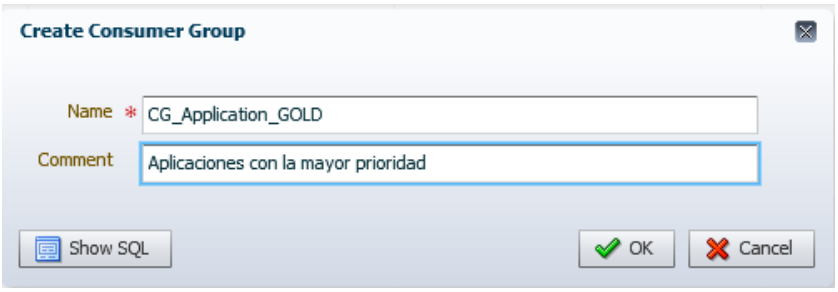


2. Creación Consumer Groups

Desde el plan creado crearemos 3 nuevos Resource Consumer Groups que no existen previamente: CG_Applications_GOLD, CG_Applications_SILVER, QUARENTINE.

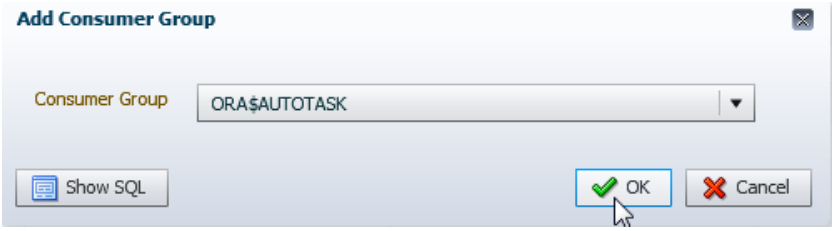
La pantalla muestra la creación del resource group CG_Applications_GOLD:

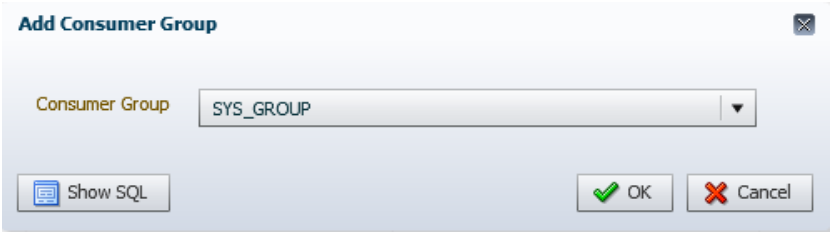




De forma análoga crearemos los consumer groups CG_Applications_SILVER y QUARENTINE.

A continuación se añaden al plan los Consumer Group ORA$AUTOTASK y SYSGROUP





3. Configuración de directivas

3.1. Porcentajes de CPU

Dependiendo de los requerimientos de rendimiento de cada aplicación, se podrá adaptar una configuración de las múltiples combinaciones disponibles para la distribución de la CPU. En este caso, se mostrarán 3 opciones de implementación para mostrar con un ejemplo las diferencias entre unas distribuciones y otras:

Option 1: (Single level Resource Plan)

Asignaciones de Recursos CPU

	Level 1
SYS_GROUP	50
CG_Applications_GOLD	32
CG_Applications_SILVER	12
OTHER_GROUPS	2
ORA$AUTOTASK	2
QUARENTINE	2

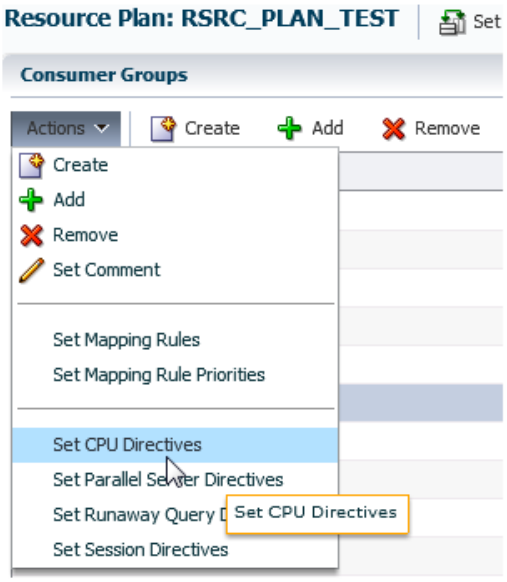
Descripción:

Esta es la configuración más simple y la recomendada. Los recursos de CPU son asignados a los consumer group basándose en los porcentajes o ratios dados a cada grupo. Cualquier recurso que no sea asignado o consumido por un consumer group será asignado a los otros grupos en base a los mismos porcentajes.

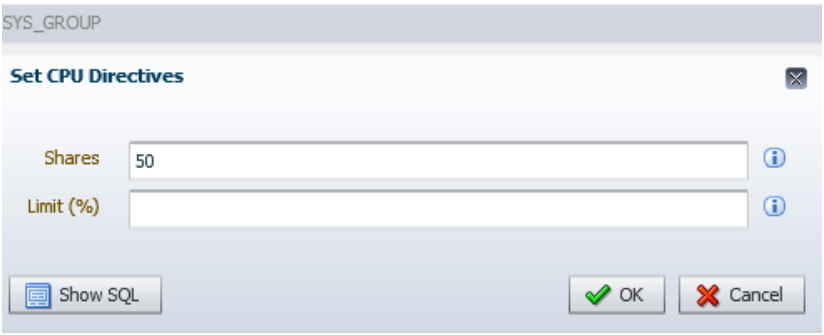
Suponiendo que no hay sesiones de SYS_GROUP consumiendo recursos de CPU, , CG_Applications_GOLD tiene garantizado obtener al menos el (32*100)/50=64% de los recursos de cpu, CG_Applications_SILVER tiene garantizado al menos (12*100)/50=24%, y los otros grupos el 4% cada uno de ellos.

Configuración:

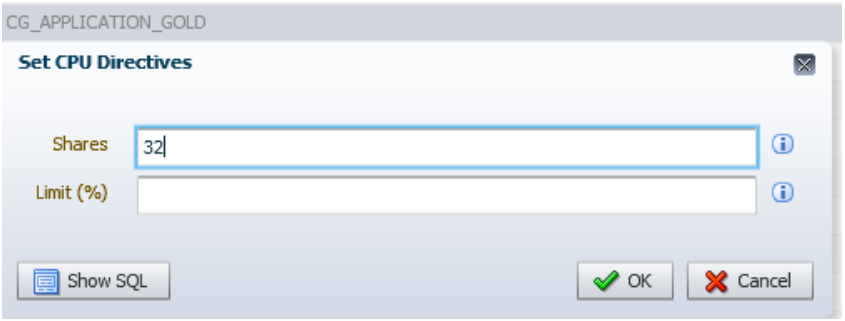
Desde EM Database Express 12c en cada consumer group añadido al plan que se está configurando se selecciona “Set CPU Directives”:



Se van configurando las directivas de cpu para cada consumer group añadido, por ejemplo para SYSGROUP



Para CG_APPLICATION_GOLD:



Así para cada resource group que queremos definir.

En este caso, la suma de los shares (ratios) configurados entre todos los resource group la hacemos a 100, con lo que equivaldrá el uso de ratios a porcentajes.

Se revisa configuración con la query indicada en el apartado Script con chequeo sobre la configuración de Resource Manager y estadísticas:

```
RESOURCE MANAGER CONFIGURATION
=======================================

                                                          CPU  CPU  CPU  CPU
PLAN                TYPE           RCG/subPlan           N 1  N 2  N 3  N 4
------------------- -------------- ------------------------ ---- ---- ---- ----
<<RSCR_PLAN_TEST>> CONSUMER_GROUP SYS_GROUP                 50    0    0    0
                                  CG_APPLICATION_GOLD       32    0    0    0
                                  CG_APPLICATIONS_SILVER    12    0    0    0
                                  QUARENTINE                 2    0    0    0
                                  ORA$AUTOTASK               2    0    0    0
                                  OTHER_GROUPS               2    0    0    0
******************* ************** ------------------------ ---- ---- ---- ----
TOTAL CPU/PLAN                                            100    0    0    0
```

Option 2: (Multi-level Resource Plan)

Asignaciones de Recursos CPU:

	Level 1	Level 2	Level 3	Level 4
SYS_GROUP	50			
CG_Applications_GOLD		70		
CG_Applications_SILVER			70	
OTHER_GROUPS			10	

```
ORA$AUTOTASK                          20
QUARENTINE                                    100
```

Descripción:

En el nivel 1, SYS_GROUP tendrá garantizado al menos el 50%, esto es, al menos el 50% de los recursos de CPU del sistema estarán disponibles para operaciones de SYS o SYSTEM. Cualquier recurso de cpu que esté disponible y no consumido por el nivel 1 estará disponible a los consumer group de nivel 2. Esto es, nivel 2, se garantiza obtener al menos el 50%, más cualquier recurso de CPU no usado por SYS_GROUP.

Suponiendo que no hay sesiones de SYS_GROUP consumiendo recursos de CPU, al asignar solo el 70% de CPU a CG_Applications_GOLD en nivel 2, las sesiones en nivel 3 tienen garantizado obtener al menos el 30%, más cualquier recurso de CPU no usado por CG_Applications_GOLD.

QUARENTINE, sin embargo, no tiene garantizado ningún recurso de CPU. Obtendrá CPU sólo si CG_Applications_SILVER, OTHER_GROUPS, ORA$AUTOTASK no consumen toda su asignación.

Configuración:

Desde EM Database Express no se pueden configurar multi-level plans. Se tiene que usar la API PLSQL DBMS_RESOURCE_MANAGER:

NOTA:

Previo si se quieren usar multilevel plans a la hora de crear el plan hay que indicar que el método de asignación de recursos de CPU (MGMT_MTH) del plan sea del tipo EMPHASIS (valor por defecto). De esta forma se usarán porcentajes de cómo la CPU es distribuida entre consumers groups y niveles. Sin embargo, por defecto EM Database Express configura el método RATIO en la creación del plan, que indica el ratio de cómo será distribuida la CPU en un único nivel.

Si se quiere utilizar multilevel hay que cambiar en el plan creado con EM Databas Express el método a EMPHASIS, para esto:

```
BEGIN
    DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.UPDATE_PLAN(
        PLAN => 'RSCR_PLAN_TEST',
        NEW_COMMENT => 'Test Implementation RM',
        NEW_MGMT_MTH => 'EMPHASIS');
    DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

A continuación se indica la configuración:

```
BEGIN
    DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'SYS_GROUP',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 50,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'CG_APPLICATION_GOLD',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 70,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'CG_APPLICATIONS_SILVER',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 70,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 10,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'ORA$AUTOTASK',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 20,
        NEW_MGMT_P4 => 0,
```

```
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'QUARENTINE',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 100,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);
    DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

Se revisa configuración con la query indicada en el apartado Script con chequeo sobre la configuración de Resource Manager y estadísticas:

```
RESOURCE MANAGER CONFIGURATION
======================================

                                                          CPU  CPU  CPU  CPU
PLAN               TYPE           RCG/subPlan             N 1  N 2  N 3  N 4
------------------ -------------- ------------------------ ---- ---- ---- ----
<<RSCR_PLAN_TEST>> CONSUMER_GROUP SYS_GROUP                 50    0    0    0
                                  CG_APPLICATION_GOLD        0   70    0    0
                                  CG_APPLICATIONS_SILVER     0    0   70    0
                                  ORA$AUTOTASK               0    0   20    0
                                  OTHER_GROUPS               0    0   10    0
                                  QUARENTINE                 0    0    0  100
****************** ************** ------------------------ ---- ---- ---- ----
TOTAL CPU/PLAN                                             50   70  100  100
```

Option 3: (Multi-level Resource Plan)

Asignaciones de Recursos CPU

	Level 1	Level 2	Level 3
SYS_GROUP	50		
CG_Applications_GOLD		70	
CG_Applications_SILVER			20
OTHER_GROUPS		4	
ORA$AUTOTASK		6	
QUARENTINE			100

Descripción:

En el nivel 1, SYS_GROUP tendrá garantizado al menos el 50%, esto es, al menos el 50% de los recursos de CPU del sistema estarán disponibles para operaciones de SYS o SYSTEM. Cualquier recurso de cpu que esté disponible y no consumido por el nivel 1 estará disponible a los consumer group de nivel 2. Esto es, nivel 2, se garantiza obtener al menos el 50%, más cualquier recurso de CPU no usado por SYS_GROUP.

Suponiendo que no hay sesiones de SYS_GROUP consumiendo recursos de CPU, al asignar solo el 70% de CPU a CG_Applications_GOLD en nivel 2, las sesiones en CG_Applications_SILVER tienen garantizado al menos un 20%, pero cualquier CPU no usada por CG_Applications_GOLD al estar en su mismo nivel serán ofrecidas al resource group QUARENTINE, en lugar de a CG_Applications_SILVER, OTHER_GROUPS, ORA$AUTOTASK.

En esta configuración, si no hay suficiente carga de CPU por parte de CG_Applications_GOLD, el consumer group QUARENTINE tendrá más porcentaje de CPU garantizado que CG_Applications_SILVER.

Configuración:

Desde EM Database Express no se pueden configurar multi-level plans. Se tiene que usar la API PLSQL DBMS_RESOURCE_MANAGER:

```
BEGIN
    DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'SYS_GROUP',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 50,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'CG_APPLICATION_GOLD',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 70,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
```

```
        GROUP_OR_SUBPLAN => 'CG_APPLICATIONS_SILVER',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 20,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 4,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'ORA$AUTOTASK',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 6,
        NEW_MGMT_P3 => 0,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);

    DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(
        PLAN => 'RSCR_PLAN_TEST',
        GROUP_OR_SUBPLAN => 'QUARENTINE',
        NEW_COMMENT => '',
        NEW_MGMT_P1 => 0,
        NEW_MGMT_P2 => 0,
        NEW_MGMT_P3 => 100,
        NEW_MGMT_P4 => 0,
        NEW_MGMT_P5 => 0,
        NEW_MGMT_P6 => 0,
        NEW_MGMT_P7 => 0,
        NEW_MGMT_P8 => 0);
    DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

Se revisa configuración con la query indicada en el apartado Script con chequeo sobre la configuración de Resource Manager y estadísticas:

```
RESOURCE MANAGER CONFIGURATION
```

```
========================================
                                                          CPU  CPU  CPU  CPU
PLAN                TYPE            RCG/subPlan           N 1  N 2  N 3  N 4
------------------- --------------- ------------------------ ---- ---- ---- ----
<<RSCR_PLAN_TEST>> CONSUMER_GROUP SYS_GROUP                 50    0    0    0
                                  CG_APPLICATION_GOLD        0   70    0    0
                                  CG_APPLICATIONS_SILVER     0   20    0    0
                                  ORA$AUTOTASK               0    6    0    0
                                  OTHER_GROUPS               0    4    0    0
                                  QUARENTINE                 0    0  100    0
******************* ************** ------------------------ ---- ---- ---- ----
TOTAL CPU/PLAN                                              50  100  100    0
```

En nuestro ejemplo se escoge la opción 1.

Consideraciones y recomendaciones sobre esta directiva

- Oracle recomienda como buena práctica no usar multi-level plans principalmente debido a:
    - La mayoría de los clientes malinterpretan como trabajan estos multi-level plans. Pon tanto, estos multi-level plans no funcionan como ellos esperan.
    - Es una buena práctica tener una configuración lo más simple posible de los planes de Resource Manager.
    - Multi-level plans no están soportados para PDBs o CDBs en 12c
    - Multi-level plans no pueden ser configurados con EM Database Express 12c

  Información oficial sobre esta buena práctica:

  NOTE:1338988.1 - Scripts and Tips for Monitoring CPU Resource Manager

  Note: 1590299.1 Considerations about multi level resource plan

- Para cumplir estos porcentajes indicados, el consumo de cpu en el sistema tiene que ser máximo (idle < 1%). Resource Manager está diseñado para maximizar el rendimiento, por tanto la directiva de distribución de cpu no es un límite. A menos que sesiones concurrentes de otros grupos estén consumiendo su CPU asignada, una sesión puede consumir todos los recursos disponibles, incluso aunque sobrepase el porcentaje límite configurado. Si una sesión no alcanza su límite de CPU, a los resource group restantes se les ofrece esos ciclos de cpu. Por tanto, si dos trabajos concurrentes no alcanzan el límite del porcentaje indicado en sus directivas, resource manager no tendrá ningún efecto.

  Oracle Database Resource Manager no es una compensación para un sistema mal ajustado, pero teniendo en cuenta un sistema sobrecargado, proporciona al DBA la habilidad de poder gestionar y repartir la cantidad total de recursos de CPU entre las sesiones concurrentes basándose en la importancia de unas sobre otras. Por supuesto, en un sistema sobrecargado no se podrá garantizar cierto tiempo de respuesta, pero con esta directiva

de Resource Manager, el DBA puede planear de forma más proactiva y predecible el rendimiento cumpliendo niveles de servicio establecidos.

- Si nuestro plan está activo en la ventana de mantenimiento, Oracle recomienda que el resource plan contenga una directiva para las tareas de mantenimiento. Para hacer esto, tenemos que asegurar que se incluye el como consumer group ORA$AUTOTASK en el nuevo plan. Si no se incluyen estas tareas irán al consumer group OTHER_GROUPS.
- Con esta configuración las operaciones de backup con RMAN irán a OTHER_GROUPS.

  Por defecto, debido al mapeo predefinido de consumer group, las operaciones de backup de RMAN van al resource group BATCH_GROUP y en nuestro caso este consumer group no ha sido añadido como parte del plan definido, por tanto será asignado al default group OTHER_GROUPS.

  En resumen, aunque un RMAN BACKUP sea ejecutado como SYSDBA, no irá al grupo SYS_GROUP, irá a OTHER_GROUPS.

  Hay otras operaciones predefinidas en la base de datos que van a consumer group predefinidos:

  http://docs.oracle.com/database/122/ADMIN/managing-resources-with-oracle-database-resource-manager.htm#ADMIN11915

  > *Table 27-7 summarizes the consumer group mapping rules that are predefined in Oracle Database.*
  >
  > *BATCH_GROUP:The session is running a backup operation with RMAN. The session is automatically switched to BATCH_GROUP when the operation begins.*
  >
  > *COPY: The session is running a copy operation with RMAN. The session is automatically switched to BATCH_GROUP when the operation begins.*
  >
  > *DATALOAD: The session is performing a data load operation with Data Pump. The session is automatically switched to ETL_GROUP when the operation begins.*

Si no queremos este comportamiento, podremos añadir los grupos BATCH_GROUP y ETL_GROUP en el nuevo plan, o se puede usar el DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING procedure para modificar o borrar estas reglas de mapeo.

Podemos verificar estas reglas consultando la vista DBA_RSRC_GROUP_MAPPINGS.

Por ejemplo:

```
select * from DBA_RSRC_GROUP_MAPPINGS;

ATTRIBUTE         |VALUE       |CONSUMER_GROUP
------------------|------------|-----------------
ORACLE_USER       |SYS         |SYS_GROUP
ORACLE_USER       |SYSTEM      |SYS_GROUP
ORACLE_FUNCTION   |BACKUP      |BATCH_GROUP
ORACLE_FUNCTION   |COPY        |BATCH_GROUP
```

```
ORACLE_FUNCTION   |DATALOAD     |ETL_GROUP
ORACLE_FUNCTION   |INMEMORY     |ORA$AUTOTASK
```

Podemos cambiar las reglas de mapeo, por ejemplo, para hacer que las sesiones de backup de RMAN vayan al consumer group SYS_GROUP podemos usar el procedure SET_CONSUMER_GROUP_MAPPING donde el CONSUMER_GROUP es NULL:

```
BEGIN
    dbms_resource_manager.clear_pending_area();
    dbms_resource_manager.create_pending_area();
    dbms_resource_manager.set_consumer_group_mapping(
            attribute => 'ORACLE_FUNCTION',
            value => 'BACKUP'
    );
    dbms_resource_manager.submit_pending_area();
END;
/
```

```
select * from DBA_RSRC_GROUP_MAPPINGS;

ATTRIBUTE         |VALUE       |CONSUMER_GROUP
------------------|------------|-----------------
ORACLE_USER       |SYS         |SYS_GROUP
ORACLE_USER       |SYSTEM      |SYS_GROUP
ORACLE_FUNCTION   |COPY        |BATCH_GROUP
ORACLE_FUNCTION   |DATALOAD    |ETL_GROUP
ORACLE_FUNCTION   |INMEMORY    |ORA$AUTOTASK
```

3.2. Límite máximo de utilización

Descripción:

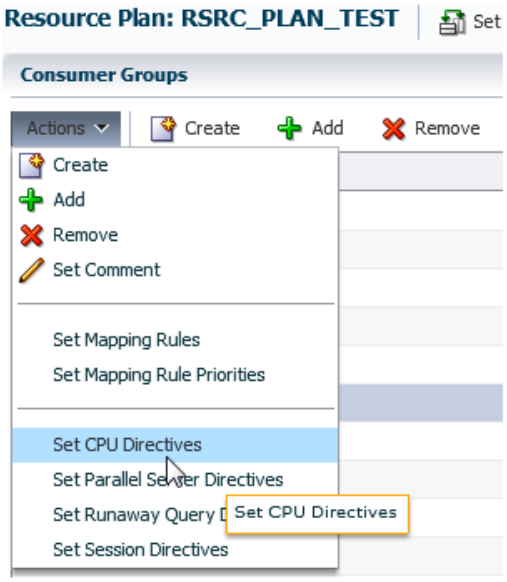
En nuestro ejemplo se plantea como objetivo restringir el uso total de la CPU por parte de la base de datos. Independientemente de la carga de la base de datos, el consumo de la base de datos no deberá exceder el 80% de la CPU del sistema.

También, se pondrán límites más bajos de la cantidad máxima de utilización de CPU para sesiones en dos consumer groups, limitando a 50% para OTHERS_GROUPS y a 20% a QUARENTINE.

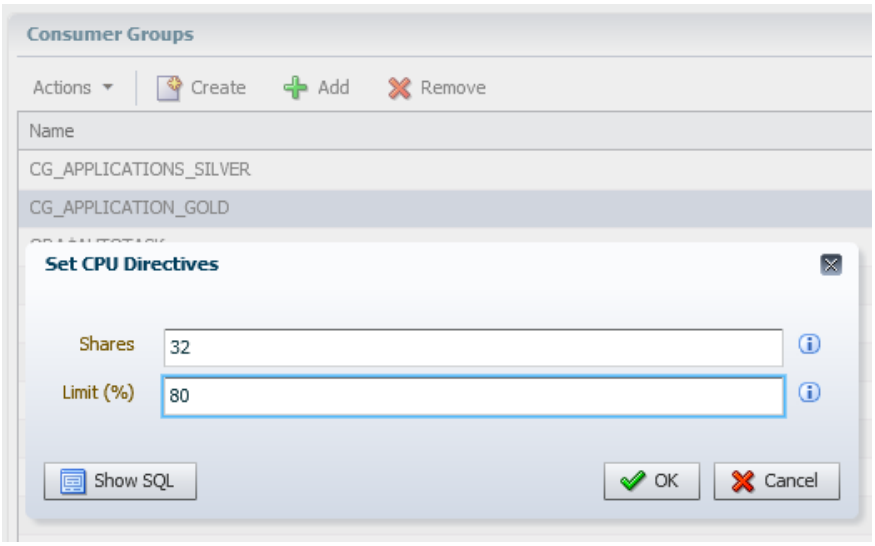
En la directiva de "Consumer Group Switching" se configurará para que las sentencias fuera de control que hayan consumido una cantidad de recursos inesperada sean movidos a este consumer group QUARENTINE, y aquí se asegurará que estas sentencias no pueden obtener en su conjunto más de un 20% de los recursos de cpu del servidor.

Configuración:

Desde EM Database Express 12c en cada consumer group añadido al plan que se está configurando se selecciona "Set CPU Directives":



Se van configurando las directivas de cpu para cada consumer group añadido, por ejemplo para CG_APPLICATION_GOLD



Consideraciones y recomendaciones sobre esta directiva

- Este límite impide que un consumer group pueda usar todos los recursos de CPU aunque los otros consumer group no los estén usado y estos estén disponibles.
- Aunque se podría configurar el límite máximo de utilización a cero para un consumer group y así completamente poner en cuarentena las sentencias fuera de control, no se recomienda hacer esto. Si las sentencias en este estado tienen otros recursos asignados a nivel de base de datos – memoria PGA, bloqueos, etc – y son requeridos por otra sesión, entonces una asignación de cero podría provocar problemas a otras sesiones.
- Si una sesión es automáticamente o manualmente movida a otro consumer group que no es parte del plan actual, esta sesión será movida al grupo "OTHER_GROUPS". En nuestro ejemplo, si las conexiones no usan correctamente los servicios de base de datos, estas sesiones irá a OTHER_GROUPS y tendrán más limitados los recursos de CPU. Esto también aplicará a las operaciones de backup con RMAN como se comentó más arriba.

3.3. Active Session Pool.

Descripción:

Otro de los objetivos en la definición de plan a crear es que algunas aplicaciones tuvieran un número máximo de operaciones concurrentes en la base de datos.

En este caso, especificaremos un número máximo de operaciones concurrentes en el consumer group CG_Applications_SILVER de 50. Cuando una nueva llamada a la base de datos desde una sesión en este consumer group no pueda iniciarse porque está lleno el pool, la sesión se pondrá en una cola. Cuando otra de las llamadas activas finalice, la primera sesión en la cola se pondrá lista para ejecución.

Configuración:

Esta directiva no puede configurarse desde EM Database Express 12c, hay que utilizar la API PLSQL:

```
begin
     sys.dbms_resource_manager.clear_pending_area();
     sys.dbms_resource_manager.create_pending_area();
     sys.dbms_resource_manager.update_plan_directive(
         plan => 'RSCR_PLAN_TEST',
         group_or_subplan => 'CG_APPLICATIONS_SILVER',
              NEW_ACTIVE_SESS_POOL_P1 => 50);

     sys.dbms_resource_manager.validate_pending_area();
     sys.dbms_resource_manager.submit_pending_area();
     sys.dbms_resource_manager.clear_pending_area();
exception
     when others then
     sys.dbms_resource_manager.clear_pending_area();
     raise;
```

```
end;
/
```

Consideraciones y recomendaciones sobre esta directiva

- En el "active pool" las sesiones son consideradas activas aunque estén esperando por otros eventos de espera Oracle (Enqueue, IO, library cache lock, etc.). Por ejemplo, si hay bloqueos en una tabla entre usuarios de la base de datos, un valor muy bajo de un active session pool puede provocar que todas las sesiones del mismo resource Group se queden bloqueadas.
- El active session pool no debería usarse en resource group con cargas de trabajo tipo OLTP, ni para implementar un connection pooling.
- En nuestro ejemplo no se activa el Active Queue Timeout. Si fuera configurado, después del periodo indicado de tiempo esperando en la cola la sesión recibirá un timeout y terminará con un error.
- Esta directiva puede ser útil para restringir el número de procesos batchs durante horas de carga máxima.

3.4. Undo Pool.

Descripción:

El undo pool controla la cantidad total de undo que puede ser generado por las sesiones de un consumer group. En nuestro ejemplo se hará este reparto:

CG_Applications_GOLD undo_pool => 2GB

CG_Applications_SILVER undo_pool => 1GB

OTHER_GROUPS undo_pool => 1GB

Configuración:

Esta directiva no puede configurarse desde EM Database Express 12c, hay que utilizar la API PLSQL:

```
begin
     sys.dbms_resource_manager.clear_pending_area();
     sys.dbms_resource_manager.create_pending_area();

     sys.dbms_resource_manager.update_plan_directive(
         plan => 'RSCR_PLAN_TEST',
         group_or_subplan => 'CG_APPLICATION_GOLD',
         NEW_UNDO_POOL => 2097152);

     sys.dbms_resource_manager.update_plan_directive(
         plan => 'RSCR_PLAN_TEST',
         group_or_subplan => 'CG_APPLICATIONS_SILVER',
         NEW_UNDO_POOL => 1048576);

     sys.dbms_resource_manager.update_plan_directive(
         plan => 'RSCR_PLAN_TEST',
         group_or_subplan => 'OTHER_GROUPS',
```

```
          NEW_UNDO_POOL => 1048576);

     sys.dbms_resource_manager.validate_pending_area();
     sys.dbms_resource_manager.submit_pending_area();
     sys.dbms_resource_manager.clear_pending_area();
exception
     when others then
          sys.dbms_resource_manager.clear_pending_area();
          raise;
end;
/
```

Consideraciones y recomendaciones sobre esta directiva

- Cuando el total de undo generado por un resource group excede el undo limit, la sentencia SQL que esté generando undo fallará.
- Ningún miembro del consumer group podrá manipular datos hasta que se libere espacio de undo del pool.
- La gestión de undo en la directiva "undo_pool" limita el tamaño del "active undo", no el "unexpired undo" (usado para flashback y para lecturas consistentes si el UNDO_RETENTION es alto y hay consultas SQL largas activas).
- Este método es usado para prevenir transacciones fuera de control que consumen excesivo espacio de undo.

3.5. Thresholds / Consumer Group Switching

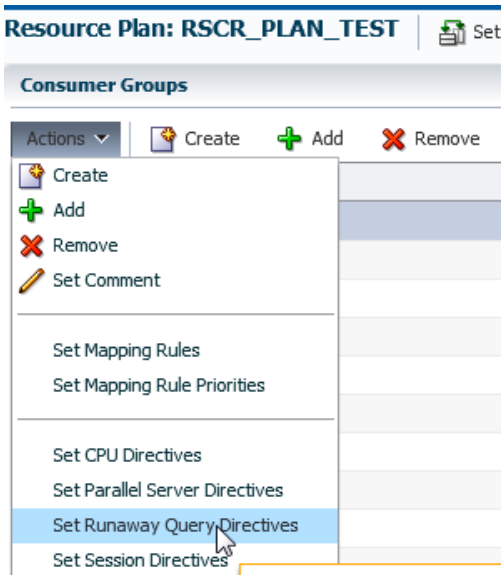
Descripción:

En este ejemplo se establecerán los siguientes límites: Si una sesión en los grupos CG_Applications_GOLD o CG_Applications_SILVER ejecuta una llamada durante más de 1200 o 600 segundos respectivamente, la sesión será automáticamente trasladada al resource group QUARENTINE. Este consumer group ha sido configurado con una utilización máxima de cpu del 20%. Al final de la llamada la sesión vuelve a su resource group original.

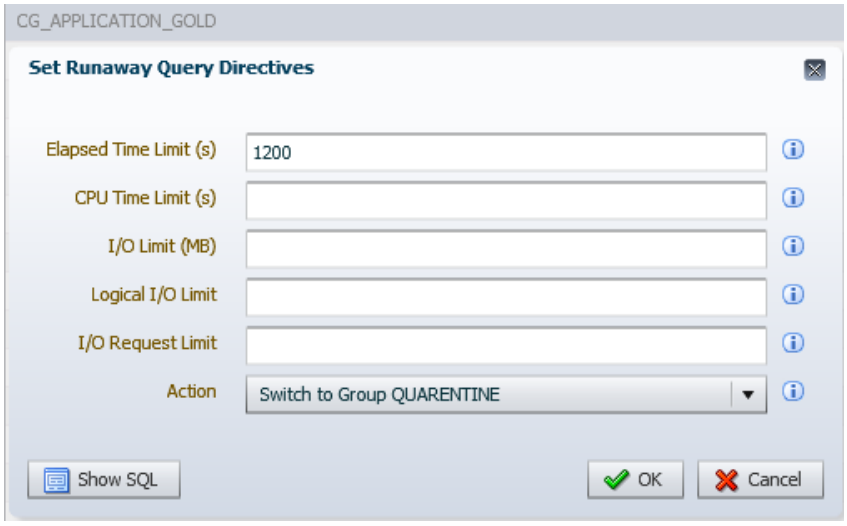
Adicionalmente QUARENTINE group tendrá un límite de tiempo de ejecución de 3600 segundos, después de esto la operación será cancelada.

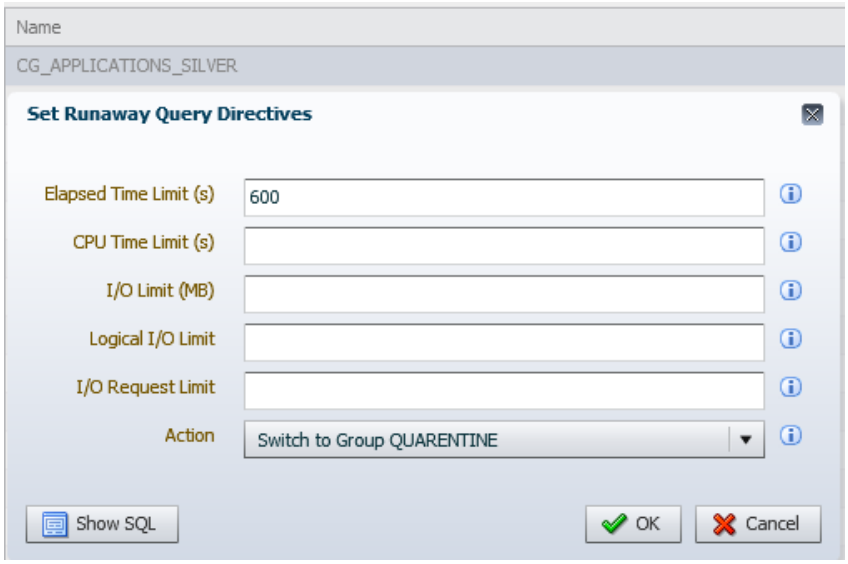
Configuración:

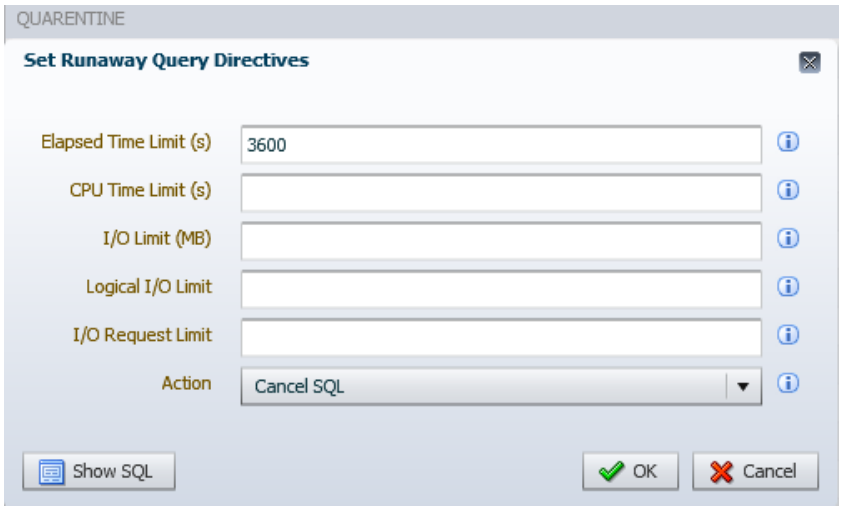
Desde EM Database Express 12c en cada consumer group añadido al plan que se está configurando se selecciona "Set Runaway Query Directives":



Se van configurando las directivas de límites para el cambio de cada consumer group como se indica en la descripción:







Consideraciones y recomendaciones sobre esta directiva

- En este ejemplo, solo el Elapsed Time Limit (SWITCH_ELAPSED_TIME) se establece.

  También se podrían establecer las siguientes directivas (se indican los nombres requeridos en caso de querer utilizar la API PLSQL):

  Límites I/O: SWITCH_IO_REQS, SWITCH_IO_MEGABYTES

Límites I/O Lógicas: SWITCH_IO_LOGICAL

CPU Time Limit (s): SWITCH_TIME

De forma que se pueda limitar el número de llamadas I/O, I/O lógicas o cpu que una sesión puede realizar y una vez que la sesión alcanzara ese límite haría el cambio a otro resource group o se cancelaría.

El Elapsed Time Limit se refiere al tiempo de ejecución en segundos de la operación tanto CPU come eventos de espera. El CPU Time Limit (s): SWITCH_TIME sólo contabiliza CPU pero no esperas.

Esto es sólo un ejemplo, lo más normal sería también limitar los recursos de cpu o I/O para determinar que una sentencia está fuera de control ya que el elapsed time puede estar influenciado puntualmente por eventos de espera que no tienen por qué consumir recursos.

- Por defecto EM Database Express 12c configura la directiva SWITCH_FOR_CALL=TRUE, en versiones anteriores Enterprise Manager sí tenía un checkbox "*Revert after call*" para poder cambiar esta configuración. En general es recomendado tener SWITCH_FOR_CALL=TRUE. En tal caso Resource Manager no esperará hasta que la sesión se marque como idle para volver a su resource consumer group original. La sesión vuelve a su consumer group original cuando el nivel superior de la llamada sea completado. El nivel superior de una llamada en PL/SQL es el bloque completo de PL/SQL que es tratado como una llamada. El nivel superior en SQL es una sentencia SQL individual. En caso de que se configure a FALSE, tras una sesión continuará en el nuevo consumer group hasta que se considere "idle", en ese momento volverá a ser cambiada a su consumer group original y volver a marcar los contadores de IO y CPU a cero. Resource Manager considera que una sesión es "idle" si pasa una cantidad de tiempo entre llamadas. Este intervalo de tiempo son unos pocos segundos (5 segundos) y no es configurable.

  Si se quiere cambiar el SWITCH_FOR_CALL=FALSE habrá que configurarlo con el procedure de la API PLSQL DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE.

  NOTA: La configuración de SWITCH_FOR_CALL=FALSE es incompatible con el uso de la directiva Elapsed Time Limit. Sólo en el caso de no configurar esta y usar por ejemplo el CPU Time Limit, se podría cambiar el SWITCH_FOR_CALL=FALSE.

- Existe otra directiva relacionada no configurable desde EM Express 12c llamada SWITCH_ESTIMATE. EM Express 12c por defecto la configura como FALSE. Con la API PLSQL se podría cambiar SWITCH_ESTIMATE =TRUE y la base de datos estimará el tiempo de ejecución de cada llamada y si la estimación excede el tiempo indicado en SWITCH_TIME, la sesión hará el movimiento al SWITCH_GROUP indicado antes de que empiece la ejecución. El tiempo estimado es obtenido del optimizador. La precisión de la estimación depende de muchos factores especialmente de la calidad de las estadísticas del optimizador y podría no siempre precisa. En general se debe tener un margen de precisión de ± 10 minutos.

- En el ejemplo cuando un proceso largo ejecutándose en la application1 después de 1200 segundos de elapsed time pasará al consumer group QUARENTINE y se le aplicará sus límites y configuración de cpu. Esto es, si continúa, después de consumir otros 3600segundos el proceso será cancelado y recibirá el error *ORA-56735: elapsed time limit exceeded - call aborted*. En caso de usar la directiva CPU Time Limit recibirá el error *ORA-00040 Active Time Limit Exceeded* cuando la sesión consuma en tiempo de cpu lo indicado.

3.6. Tiempo máximo estimado de ejecución

Descripción:

En este ejemplo se establecerá la siguiente restricción: las sesiones que estén en el resource group CG_APPLICATIONS_SILVERS no se les permitirá iniciar operaciones que se estimen que van a tardar más de 6000 segundos.

La directiva MAX_EST_EXEC_TIME especifica el tiempo máximo estimado de ejecución.

Configuración:

Esta directiva no puede configurarse desde EM Database Express 12c, hay que utilizar la API PLSQL:

```
begin
     sys.dbms_resource_manager.clear_pending_area();
     sys.dbms_resource_manager.create_pending_area();

     sys.dbms_resource_manager.update_plan_directive(
         plan => 'RSCR_PLAN_TEST',
         group_or_subplan => 'CG_APPLICATIONS_SILVER',
         NEW_MAX_EST_EXEC_TIME => 6000);

     sys.dbms_resource_manager.validate_pending_area();
     sys.dbms_resource_manager.submit_pending_area();
     sys.dbms_resource_manager.clear_pending_area();
exception
     when others then
          sys.dbms_resource_manager.clear_pending_area();
          raise;
end;
/
```

Consideraciones y recomendaciones sobre esta directiva

- Esta directiva puede ser útil en cualquier base de datos. Si la base de datos estima que la operación va a tardar más de ese tiempo máximo estimado de ejecución la operación no arranca. Los administradores de base de datos pueden usar este método para prevenir a ciertos resource group que se lancen operaciones muy largas que consumirán muchos recursos antes de que comiencen.

- Si el optimizador estima que la operación tardará más que el MAX_EST_EXEC_TIME, la operación no arrancará y se generará un error ORA-07455.

  Ejemplo:

  Un usuario se conecta usando el Servicio application2 tendrá un límite estimado de ejecución de 6000 segundos. Si lanza por error la siguiente sentencia, resource manager previene de su ejecución:

```
SQL> conn user1/user1@application2

select count(*) from dba_objects, dba_objects, dba_objects
                     *
ERROR at line 1:
ORA-07455: estimated execution time (69707477 secs),
exceeds limit (6000 secs)
```

- La estimación la genera el Optimizador basado en coste. Es muy importante tener estadísticas fiables en los objetos referenciados en la consulta.
- Si el optimizador no puede dar una estimación, esta directiva no tendrá efecto. Por ejemplo, si la sesión aplica el hint RULE la directive no tendrá efecto:

```
SQL> conn user1/user1@application2
SQL> select /*+ RULE */ count(*) from dba_objects,
dba_objects, dba_objects;
```

  Esta consulta no da error.

3.7. Idle time

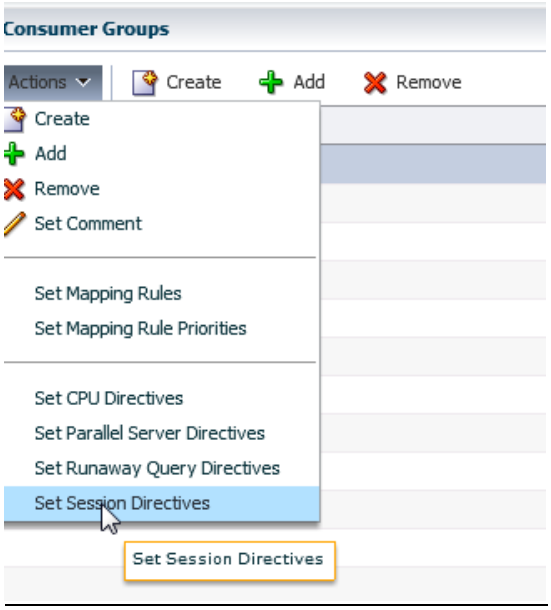
Descripción:

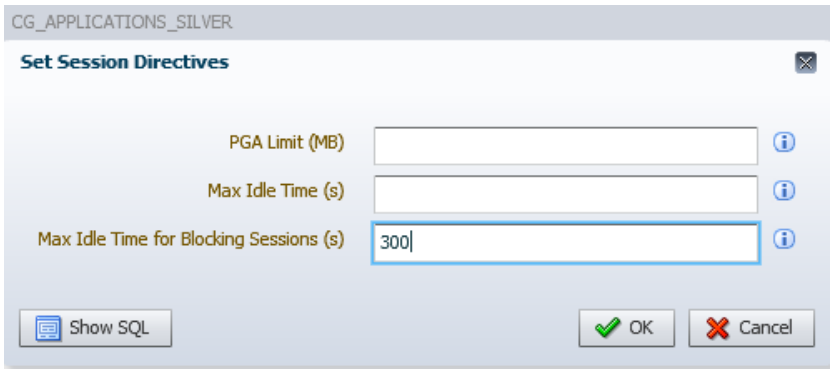
Con esta directiva el administrador puede especificar la cantidad de tiempo que una sesión puede estar inactiva ("idle"), después del cual la sesión será finalizada. Es posible restringir esta finalización de sesiones a sólo aquellas que bloqueen a otras sesiones.

En nuestro ejemplo se configura el tiempo máximo inactivo para las sesiones de CG_Applications_SILVER pero restringiendo a aquellas que bloquean a otras sesiones.

Configuración:

Desde EM Database Express 12c en "Set Session Directives" se configura:





Consideraciones y recomendaciones sobre esta directiva

- El tiempo se refiere al tiempo que lleva inactiva, no al tiempo que está bloqueando a otra sesión.

  Ejemplo:

```
Sesión 1:

SQL> conn user1/user1@application2
SQL> create table test (col1 number);
SQL> insert into test values (1);
SQL> commit
SQL> update test set col1=2;
SQL>
```

```
Sesión 2:
SQL> conn user1/user1@application2
SQL> delete test;

--> Se bloquea... hasta que la sesión 1 llega al límite de
300 segundos inactiva
```

Si han pasado más de 300segundos tras lanzar el comando update en la sesión 1, en cuanto entre un bloqueo producido por la sesión 1, la sesión será terminada y el bloqueo se liberará.

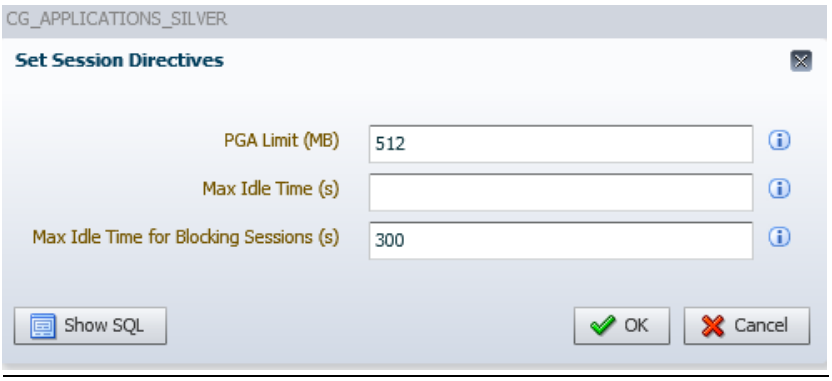
```
Session 1:
SQL> select * from dual;
select * from dual
       *
ERROR at line 1:
ORA-03113: end-of-file on communication channel
Process ID: 8852

Session ID: 14 Serial number: 1207
```

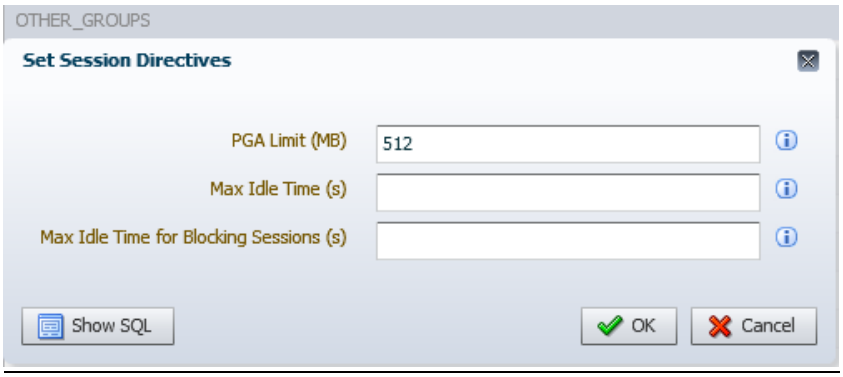
3.8. PGA

Descripción:

En nuestro ejemplo se configura la cantidad máxima de PGA para los Grupos CG_APPLICATIONS_SILVER y QUARENTINE de 512MB.

Configuración:





Consideraciones y recomendaciones sobre esta directiva

- Con esta directiva el administrador puede especificar cuando una sesión es finalizada cuando se alcanza un consumo máximo de PGA.
- Los grupos que no se configura valor implica NULL, que indicará ilimitado.

4. Creación de los mapeos con los Resource Consumer Group

Descripción:

En este ejemplo usaremos servicios de base de datos para mapear conexiones a resource consumer groups.

Para conectar, las aplicaciones usarán configuraciones de SQL*Net usando como SERVICE_NAME estos nombres (application1, application2, application3).

Previamente la base de datos tiene configurados y arrancados los siguientes servicios:

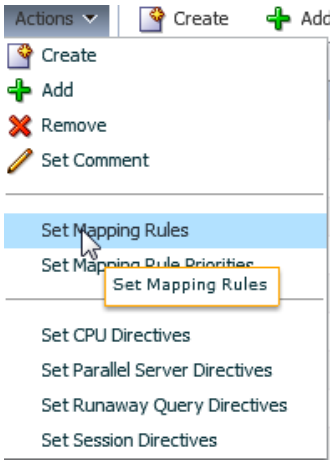
```
INST_ID|NAME             |NETWORK_NAME
--------|-----------------|--------------------
       1|admin1           |admin1
       1|admin1XDB        |admin1XDB
       1|application1     |application1
       1|application2     |application2
       1|application3     |application3
       1|SYS$BACKGROUND   |
       1|SYS$USERS        |
```

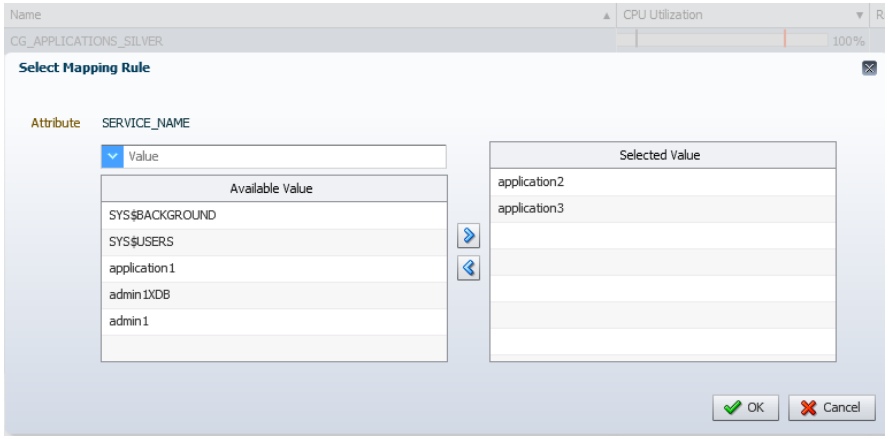
Resource Manager mapeará las conexiones a los consumer groups de esta forma:

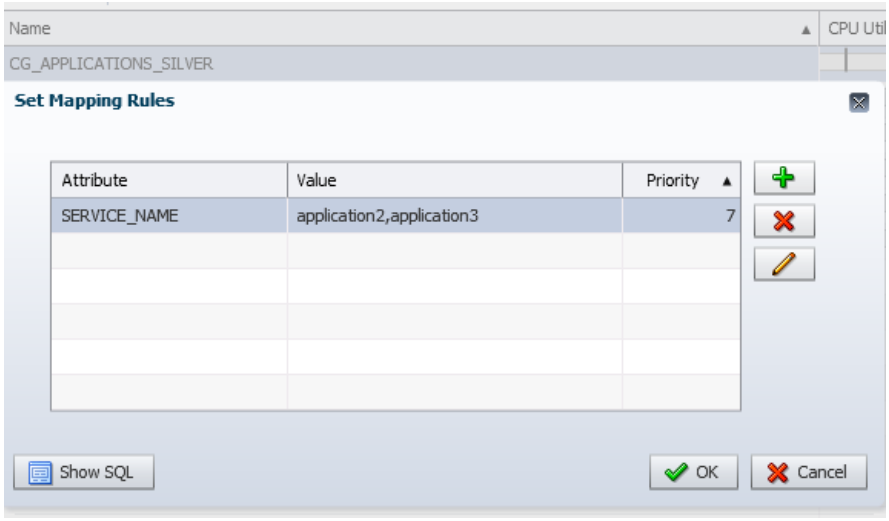
- Aplicaciones que usen el servicio application1 irán al consumer group CG_Applications_GOLD.

- Aplicaciones que usen los servicios application2 y application3 irán al consumer group CG_Applications_SILVER.
- Aplicaciones usando otros tipos de conexión sin estos servicios irán a OTHER_GROUPS.
- Un usuario concreto llamado OSS será asignado directamente al Resource Group CG_Applications_GOLD independientemente del Servicio usado.

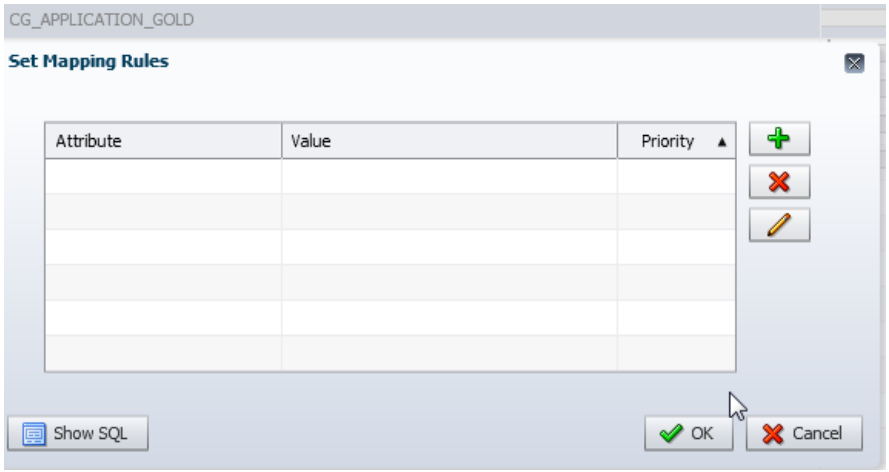
Configuración con EM Database Express 12c:

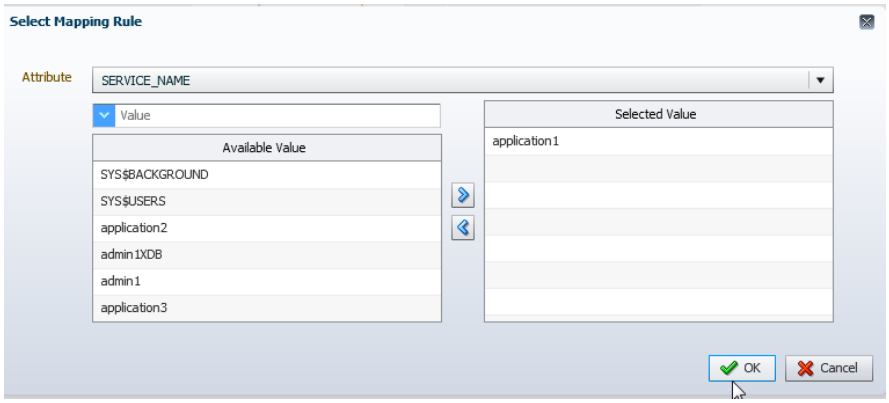


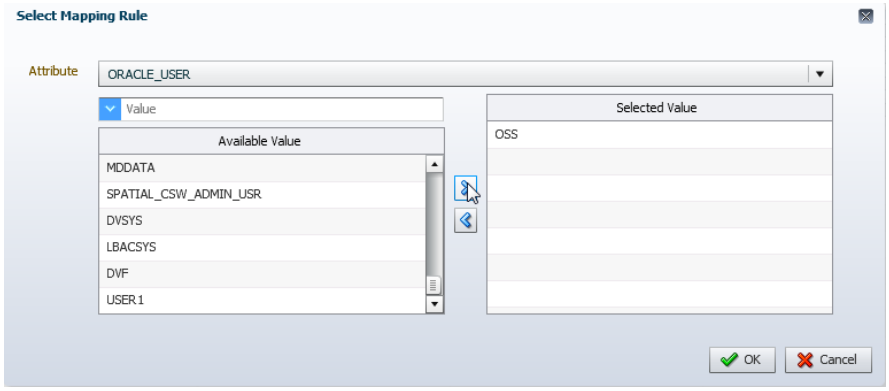


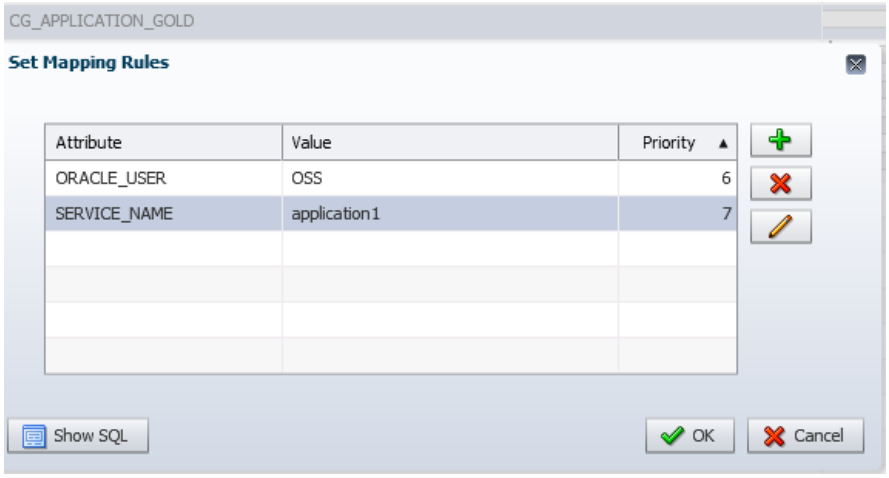


Para CG_APPLICATIONS_GOLD mapeamos tanto el servicio application1 como el usuario OSS:




Select Mapping Rule
Attribute
SERVICE_NAME
Value
Available Value
SYS$BACKGROUND
SYS$USERS
application2
admin1XDB
admin1
application3
Selected Value
application1
OK
Cancel

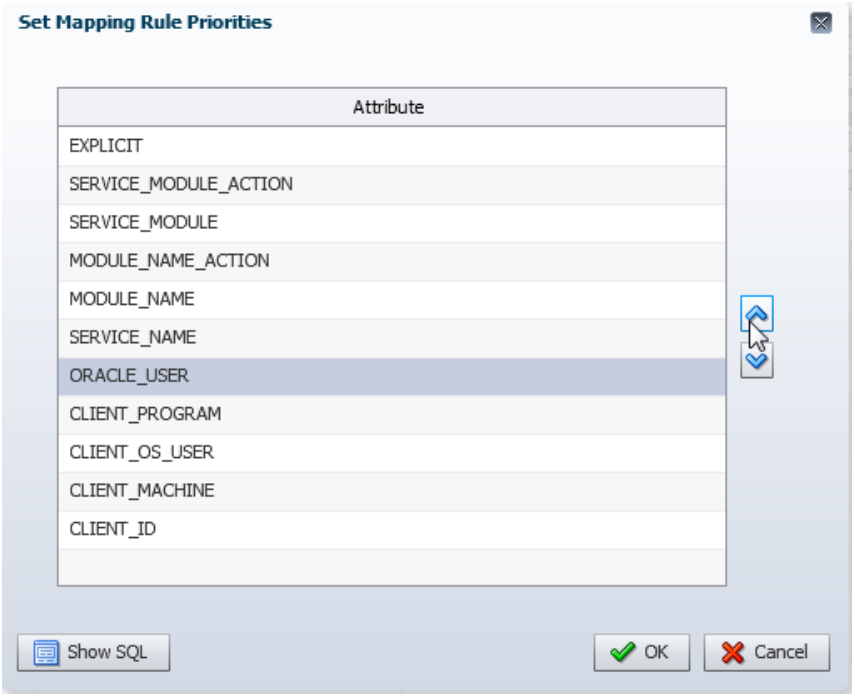

Select Mapping Rule
Attribute
ORACLE_USER
Value
Available Value
MDDATA
SPATIAL_CSW_ADMIN_USR
DVSYS
LBACSYS
DVF
USER1
Selected Value
OSS
OK
Cancel


CG_APPLICATION_GOLD
Set Mapping Rules
Attribute
Value
Priority
ORACLE_USER
OSS
6
SERVICE_NAME
application1
7
Show SQL
OK
Cancel

Cambio de prioridades en las reglas de mapeo:

Por defecto "Service" tiene más prioridad que el "Oracle User". En caso de conflicto, por ejemplo el usuario OSS conectado al Servicio application3, hace que coincidan 2 mapeos (CG_APPLICATIONS_GOLD y CG_APPLICATIONS_SILVER), como el usuario solo puede estar en uno sólo a la vez, se elegirá el que tenga más prioridad. En nuestro ejemplo se quiere que OSS vaya siempre a CG_APPLICATIONS_GOLD por tanto se tiene cambiar la prioridad.

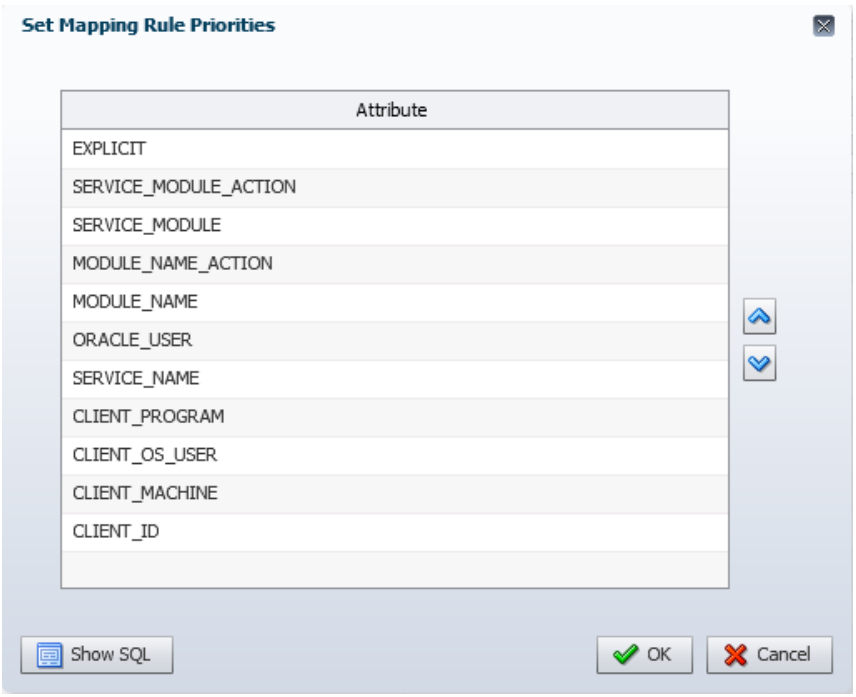
Para el cambio mostramos las opciones con EM Database Express 12c y con la API PLSQL:




```
begin
     sys.dbms_resource_manager.clear_pending_area();
     sys.dbms_resource_manager.create_pending_area();
     sys.dbms_resource_manager.set_consumer_group_mapping_pri(
         explicit => 1,
         oracle_user => 6,
         service_name => 7,
         client_os_user => 9,
         client_program => 8,
         client_machine => 10,
         module_name => 5,
         module_name_action => 4,
```

```
            service_module => 3,
            service_module_action => 2,
            client_id => 11);
      sys.dbms_resource_manager.validate_pending_area();
      sys.dbms_resource_manager.submit_pending_area();
      sys.dbms_resource_manager.clear_pending_area();
exception
      when others then
           sys.dbms_resource_manager.clear_pending_area();
           raise;
end;
/
```

Tras el cambio:



5. Privilegios

En versión 12c, no es requerido que los usuarios tengan que ser otorgados de privilegios para poder usar los consumer Groups definidos ya que estos serán asignados de forma implícita cuando la sesión cumpla alguna de las condiciones del mapping o por el hecho de ser cambiado con un switch por alguna de las directivas del plan.

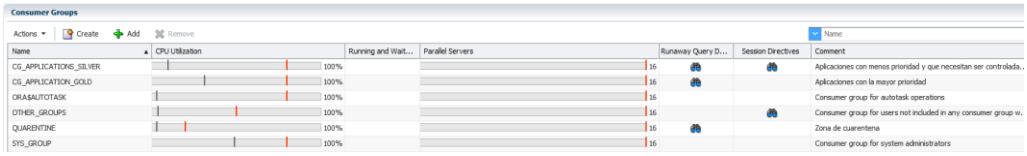
Sólo es requerido que explícitamente se otorguen los privilegios cuando se quiere permitir que el usuario se cambie de consumer plan explícitamente con:

```
DBMS_SESSION.SWITCH_CURRENT_CONSUMER_GROUP
DBMS_RESOURCE_MANAGER.SWITCH_CONSUMER_GROUP_FOR_SESS
DBMS_RESOURCE_MANAGER.SWITCH_CONSUMER_GROUP_FOR_USER
```

En este caso no se permitirá este cambio y por tanto no se dan explícitamente estos privilegios ni a usuarios ni a roles.

Resumen del Resource Plan RSCR_PLAN_TEST

Desde la consola se EM Express 12c puede ver un resumen de la configuración:



También se puede usar la consulta indicada en el apartado Script con chequeo sobre la configuración de Resource Manager y estadísticas para ver todas las directivas configuradas:

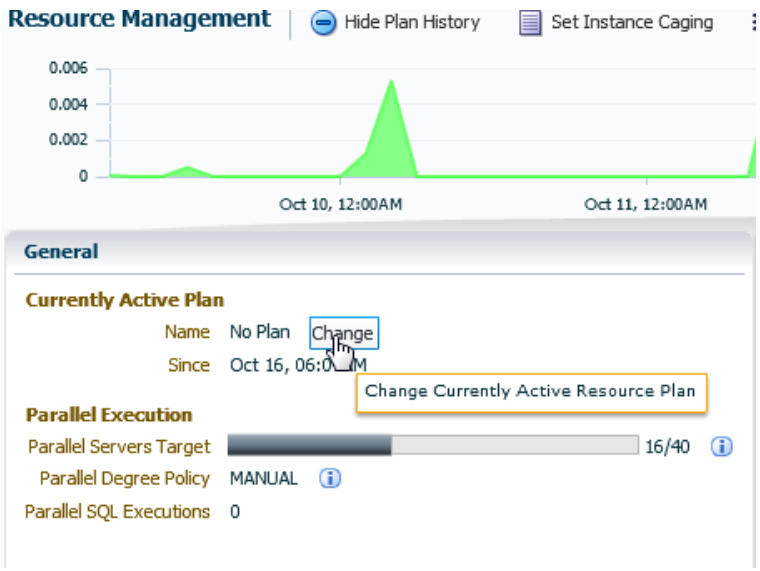
```
RESOURCE MANAGER CONFIGURATION
=======================================

                                                          CPU  CPU  CPU  CPU  Sess  DOP Máx Exec Máx idle
PLAN               STATUS  TYPE           RCG/subPlan     N 1  N 2  N 3  N 4  Pool Lím Time HMS  Time HMS
------------------ ------- -------------- --------------- ---- ---- ---- ---- ----- ---- --------- ---------
<<RSCR_PLAN_TEST>>         CONSUMER_GROUP SYS_GROUP        50    0    0    0
                           CG_APPLICATION_GOLD             32    0    0    0
                           CG_APPLICATIONS_SILVER          12    0    0    0    50      6000
                           OTHER_GROUPS                     2    0    0    0
                           ORA$AUTOTASK                     2    0    0    0
                           QUARENTINE                       2    0    0    0
****************** ******* ************** ---- ---- ---- ----
TOTAL CPU/PLAN                                            100    0    0    0
```

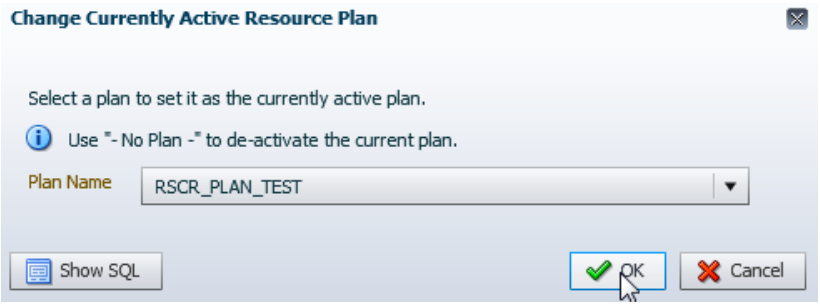
```
Máx idle Blocking  Switch secs cpu      Switch secs elapsed  Switch IO Logical      Switch IO MB            UNDO      PGA
Time HMS   Time HMS  2 Group[Estim]       2 Group              2 Group                2 Group                   MB       MB
--------- --------- ------------------- -------------------- -------------------- -------------------- ------- -------

                                 1200>QUARENTINE                                                        2,048
          30                     600>QUARENTINE                                                         1,024     512
                                                                                                        1,024     512

                                 3600>CANCEL_SQL
```

Activación del Resource Plan RSCR_PLAN_TEST

Podemos activar el Resource Plan en la instancia usando EM, sqlplus o con una ventana del Scheduler.

1. Manualmente desde EM Database Express 12c





2. **Usando SQL:**

   - Para activar el plan en la instancia:

```
SQL> alter system set resource_manager_plan='RSCR_PLAN_TEST';
```

También se podría programar para que se cambiara automáticamente con una tarea en el crontab o desde la propia base de datos con una ventana del Scheduler. Por ejemplo tener el plan activado durante las horas de oficina y crear otro más adecuado a procesos batchs en otro horario.

En nuestro ejemplo caso queremos evitar que el plan cambie por la acción de una ventana del Scheduler. Para esto basta activar el plan con la opción FORCE:

```
SQL> alter system set resource_manager_plan='FORCE:RSCR_PLAN_TEST'
scope=both;
```

   - Para desactivar el plan:

```
SQL> alter system set resource_manager_plan='';
```

Comprobando las conexiones con el resource plan

Después de activar el Resource Plan chequemos que las sesiones establecidas usando servicios son enviadas a su resource consumer group correcto.

En nuestro ejemplo en el tnsnames.ora habrá cuatro entradas de conexión a la base de datos usando servicios diferentes:

```
application1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostname)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = application1)
    )
  )


application2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostname)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = application2)
    )
  )

application3 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostname)(PORT = 1521))
```

```
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = application3)
    )
  )

application4 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = hostname)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = admin1)
    )
  )
```

Comprobamos todas las distintas conexiones establecidas inician su correcto consumer group:

```
SQL> CONN user1/user1@application1
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP         |Service              |Instance
--------------------------------|---------------------|----------
CG_APPLICATIONS_GOLD            |application1         |admin1

SQL> CONN user1/user1@application2
Conectado.
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP         |Service              |Instance
--------------------------------|---------------------|----------
CG_APPLICATIONS_SILVER          |application2         |admin1

SQL> CONN user1/user1@application3
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP         |Service              |Instance
--------------------------------|---------------------|----------
CG_APPLICATIONS_SILVER          |application3         |admin1

SQL> CONN user1/user1@application4
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP         |Service              |Instance
--------------------------------|---------------------|----------
OTHER_GROUPS                    |admin1            |admin1


SQL> CONN oss/oss@application3
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP         |Service              |Instance
--------------------------------|---------------------|----------
CG_APPLICATIONS_GOLD            | application3        |admin1

SQL> CONN system/oracle@application3
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');
```

```
RESOURCE_CONSUMER_GROUP          |Service              |Instance
---------------------------------|---------------------|----------
SYS_GROUP                        | application3        |admin1

SQL> conn / as sysdba
SQL> select resource_consumer_group , SERVICE_NAME,
sys_context('USERENV','INSTANCE_NAME') as INSTANCE_NAME from v$session
where audsid = userenv('SESSIONID');

RESOURCE_CONSUMER_GROUP          |Service              |Instance
---------------------------------|---------------------|----------
SYS_GROUP                        |SYS$USERS            |admin1
```

Mappings:

Type Connection	Consumer Group
Using service: application1	CG_APPLICATIONS_GOLD
Using service: application2	CG_APPLICATIONS_SILVER
Using service: application3	CG_APPLICATIONS_SILVER
Using service: admin1	OTHER_GROUPS
Using username: OSS	CG_APPLICATIONS_GOLD
Using SYSTEM	SYS_GROUP
Using "/ as sysdba"	SYS_GROUP