# Resource Manager

A qualquer momento, muitos processos podem estar prontos para ser executados. No entanto, o sistema operacional só pode executar um processo por CPU. Todos outros processos esperam em uma fila de execução do sistema operacional. O agendador do sistema operacional permite um processo execute por um curto período de tempo, descifra-lo e selecionar um novo processo da fila de execução para executar em seu lugar. Como o agendador usa algum tipo de algoritmo round-robin para escolher entre todos os processos executáveis, todos eles progridem. A carga de um servidor consiste em todos os processos executáveis ​​- processos que estão sendo executados ou na fila de execução. Sua utilização da CPU é inferior a 100% quando há mais CPUs do que os processos executáveis.

Um servidor está sobrecarregado quando a fila de execução se torna muito grande. Os processos devem aguardar muito tempo para executar e os processos críticos estão assim tão famintos.

O Resource Manager gerencia o uso da CPU controlando a carga do banco de dados em um nível muito preciso. Por padrão, esse nível é definido como o número de CPUs. Isso significa que em um servidor de 4 CPU, o Resource Manager garante que não mais de 4 processos Oracle (mais especificamente, primeiro plano, segundo plano não críticos e intenso uso de CPU) estão sendo executados de cada vez. Ao controlar a carga dobanco de dados, processos segundo plano crítico, podem ser executados em tempo hábil e a carga no servidor é regulamentado. O Gerenciador de recursos funciona muito bem como o agendador do sistema operacional. Resource Manager permite um processo Oracle por CPU para executar em um determinado momento. Todos os outros processos esperam em fila interno oracle para executar, no evento de espera "resmgr: cpu quantum".

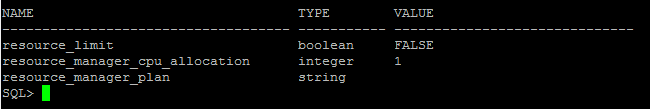
O Resource Manager não vincula processos a CPUs. Seu objetivo não é gerenciar as CPUs do servidor, mas para controlar o uso da CPU de sua instância de banco de dados. Se você quiser restringir uma instância de banco de dados a um conjunto específico de CPUs, você deve usar ferramentas como partições lógicas, conjuntos de processadores ou máquinas virtuais.

O Gerenciador de Recursos não gerencia os processos em segundo plano, a menos que eles não sejam críticos e CPU intensivo. Processos de segundo plano críticos usam a CPU de forma criteriosa e não devem ser a causa de consumo exagerado. A maioria deles precisa de acesso rápido à CPU e, portanto, o desempenho geral do banco de dados sofreria se fossem bloqueados.

# Configuração básica do Resource Manager Oracle

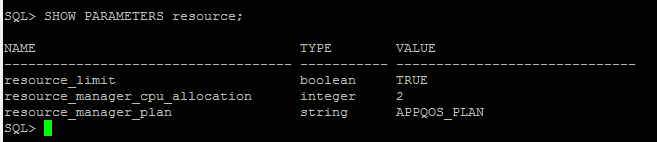
Devemos listar as configurações do spfile da instancia através do comando:

*SHOW PARAMETERS resource;*



## Para a utilização do Resource Manager precisamos modificar alguns parâmetros:

1. Alterar o Resouce\_limit para true;
   1. Commando: *alter system set resource\_limit=true scope=both;*
2. Verificar se o número de cpu do servidor está igual ao valor resouce\_manager\_cpu\_allocation;
   1. Se eu tivesse duas CPU;
   2. O comando*: alter system set resource\_manager\_cpu\_allocation=2 scope=both*;
3. Alterar o resouce\_manager\_plan para nome plano a ser utilizado.
   1. Como ainda não temos o plano criado;
   2. Comando*: alter system set resource\_manager\_plan="APPQOS\_PLAN" scope=both;*
   3. *Se parâmetro estiver vazio significar não existe nenhum plano em execução no momento. Utilizando o parâmetro “scope=both” podemos aplicar a modificação sem reiniciar a instancia.*
4. Resultado depois das modificações



## Descrição dos itens necessário para criação de um plano Resource Manager:

* Criação do grupo de consumidores (consumer group)
  + É uma coleção de sessões que são gerenciadas como uma unidade. Você pode definir grupo de consumidores para cada tipo de aplicação no seu banco dados. Ou você pode definir um grupo para cada tipo de carga de trabalho, OLTP, relatório, manutenção e etc... As sessões podem ser automaticamente mapeadas para um grupo de consumidores, definida pelo grupo consumidor.
* Criação de um plano de recursos
  + Que especifica como a CPU deve ser compartilhada entre os grupos de consumidores. Isto aloca porcentagens da CPU para grupos de consumidores e especifica como a CPU não utilizada é redistribuído. Um plano de recursos contém uma diretiva de plano de recursos para cada grupo de consumidores que especifica sua alocação de CPU. Obrigatoriamente todos os planos precisam do grupo “OTHER\_GROUPS”.
* Definido o usuário a um grupo de consumidores
  + Todo usuário deverá estar configurado em grupo de consumidores do plano atual. Quando estiver sem grupo definido ele será atribuído automaticamente ao grupo “OTHER\_GROUPS”.

## Criação do grupo de consumidores

BEGIN

DBMS\_RESOURCE\_MANAGER.CLEAR\_PENDING\_AREA;

--Eliminar alguma tentativa de configuração anterior.

DBMS\_RESOURCE\_MANAGER.CREATE\_PENDING\_AREA;

--Criar área para criação do plano. Obrigatório.

DBMS\_RESOURCE\_MANAGER.CREATE\_CONSUMER\_GROUP(

consumer\_group => 'GR\_APP',

--Nome do grupo

comment => 'GR\_APP'

);

DBMS\_RESOURCE\_MANAGER.CREATE\_CONSUMER\_GROUP(

consumer\_group => 'GR\_ADM',

comment => 'GR\_ADM'

);

DBMS\_RESOURCE\_MANAGER.VALIDATE\_PENDING\_AREA();

--Validação do commandos executados

DBMS\_RESOURCE\_MANAGER.SUBMIT\_PENDING\_AREA();

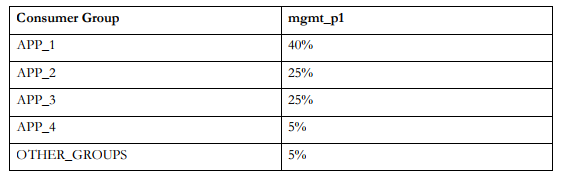
--Criação do grupo

END;

/

## Criação do plano de recursos

Uma diretiva de plano de recursos especifica quanto de CPU deve ser alocado para o grupo de consumidores. Você pode criar uma nova diretiva com o comando dbms\_resource\_manager.create\_plan\_directive (). Você pode atualizar ou excluir uma diretiva chamando update\_plan\_directive() ou delete\_plan\_directive (). O parâmetro, mgmt\_p1, especifica a porcentagem da CPU para alocar para o grupo consumidor. Por exemplo, se você tiver 4 aplicativos, você pode querer alocar a CPU da seguinte maneira. Observe que você deve fornecer uma diretiva para OTHER\_GROUPS, o grupo consumidor padrão.



Comando:

BEGIN

DBMS\_RESOURCE\_MANAGER.CLEAR\_PENDING\_AREA;

--Eliminar alguma tentativa de configuração anterior.

DBMS\_RESOURCE\_MANAGER.CREATE\_PENDING\_AREA;

--Criar área para criação do plano. Obrigatório.

--plan

DBMS\_RESOURCE\_MANAGER.CREATE\_PLAN(

PLAN => 'COPASA\_DIA',

--Nome do plano

COMMENT => 'Resource'

);

--Definido configurações do plano.

DBMS\_RESOURCE\_MANAGER.CREATE\_PLAN\_DIRECTIVE(

PLAN => 'COPASA\_DIA',

--Plano da configuração

GROUP\_OR\_SUBPLAN => 'APP1',

-- Grupo que obedecera a configuração

COMMENT => 'APP\_1',

MGMT\_P1 => 40

-- Definido a Prioridade da CPU em primeiro nível.

);

DBMS\_RESOURCE\_MANAGER.CREATE\_PLAN\_DIRECTIVE(

PLAN => 'COPASA\_DIA',

--Plano da configuração

GROUP\_OR\_SUBPLAN => 'APP\_2',

-- Grupo que obedecera a configuração

COMMENT => 'APP\_3',

MGMT\_P1 => 25

-- Definido a Prioridade da CPU em primeiro nível.

);

DBMS\_RESOURCE\_MANAGER.CREATE\_PLAN\_DIRECTIVE(

PLAN => 'COPASA\_DIA',

--Plano da configuração

GROUP\_OR\_SUBPLAN => 'APP\_3',

-- Grupo que obedecera a configuração

COMMENT => 'APP\_3',

MGMT\_P1 => 25

-- Definido a Prioridade da CPU em primeiro nível.

);

DBMS\_RESOURCE\_MANAGER.VALIDATE\_PENDING\_AREA();

--Validação do commandos executados

DBMS\_RESOURCE\_MANAGER.SUBMIT\_PENDING\_AREA();

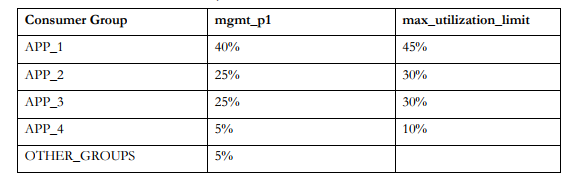
--Criando o plano

END;

/

A qualquer momento, se qualquer CPU não estiver sendo usada por um ou mais grupos de consumidores, então o recurso é redistribui-lo para os grupos de consumidores que precisam disso. No exemplo acima, se APP\_1 for o único grupo de consumidores em execução, o Resource Manager lhe dará 100% da CPU. Portanto, mgmt\_p1 especifica a quantidade de CPU que o grupo de consumidores está garantido para obter. A quantidade máxima de CPU que o grupo consumidor pode consumir é de 100%. Se você está cobrando dos proprietários do aplicativo pela atribuição de CPU e não quer que eles consumam mais do que pagaram, você pode querer limitar a utilização da CPU. Você consegue fazer isso usando a diretiva max\_utilization\_limit.

No exemplo abaixo, cada grupo de consumidores é permitido exceder sua alocação em 5%:



Comando:

BEGIN

SYS.DBMS\_RESOURCE\_MANAGER.clear\_pending\_area();

SYS.DBMS\_RESOURCE\_MANAGER.create\_pending\_area();

SYS.DBMS\_RESOURCE\_MANAGER.update\_plan\_directive(

plan => 'COPASA\_DIA',

group\_or\_subplan => 'APP\_1',

max\_utilization\_limit => 45

);

SYS.DBMS\_RESOURCE\_MANAGER.update\_plan\_directive(

plan => 'COPASA\_DIA',

group\_or\_subplan => 'APP\_1',

max\_utilization\_limit => 45

);

SYS.DBMS\_RESOURCE\_MANAGER.update\_plan\_directive(

plan => 'COPASA\_DIA',

group\_or\_subplan => 'APP\_2',

max\_utilization\_limit => 30

);

SYS.DBMS\_RESOURCE\_MANAGER.update\_plan\_directive(

plan => 'COPASA\_DIA',

group\_or\_subplan => 'APP\_3',

max\_utilization\_limit => 30

);

SYS.DBMS\_RESOURCE\_MANAGER.update\_plan\_directive(

plan => 'COPASA\_DIA',

group\_or\_subplan => 'APP\_4',

max\_utilization\_limit => 10

);

SYS.DBMS\_RESOURCE\_MANAGER.submit\_pending\_area();

END;

Em alguns casos, você pode querer alocar recursos de CPU para um grupo de consumidores somente se grupos de maior prioridade de consumidores não usarem sua alocação. Os níveis do plano permitem que você especifique como a CPU não utilizada é reatada. A CPU não utilizada ou não alocada a partir do nível 1 é usada pelo consumidor grupos no nível 2, usando a diretiva mgmt\_p2. A CPU não utilizada ou não alocada a partir do nível 2 é usada por grupos de consumidores no nível 3, usando a diretiva mgmt\_p3. E assim por diante.

## Definindo usuários ao grupo consumidores e definindo como padrão:

BEGIN

DBMS\_RESOURCE\_MANAGER.CLEAR\_PENDING\_AREA;

--Eliminar alguma tentativa de configuração anterior.

DBMS\_RESOURCE\_MANAGER.CREATE\_PENDING\_AREA;

--Criar área para criação do plano. Obrigatório. --adm

SYS.DBMS\_RESOURCE\_MANAGER\_PRIVS.GRANT\_SWITCH\_CONSUMER\_GROUP (

'MARCOS.NEXA',

--Usuário do banco dados

'GR\_ADM',

--Grupo de recurso

False

--Permissão ADM no grupo

);

--user

SYS.DBMS\_RESOURCE\_MANAGER\_PRIVS.GRANT\_SWITCH\_CONSUMER\_GROUP (

'SDE',

--Usuário do banco dados

'GR\_APP',

--Grupo de recurso

False

--Permissão APP no grupo

);

SYS.DBMS\_RESOURCE\_MANAGER.submit\_pending\_area();

--Usuários configurado no grupo

END;

/

BEGIN

--Definindo grupo padrão do usuário

SYS.DBMS\_RESOURCE\_MANAGER.SET\_INITIAL\_CONSUMER\_GROUP (

'MARCOS.NEXA',

--Usuário do banco dados

'GR\_ADM'

--Grupo de recurso

);

--Definindo grupo padrão do usuário

SYS.DBMS\_RESOURCE\_MANAGER.SET\_INITIAL\_CONSUMER\_GROUP (

'SDE',

--Usuário do banco dados

'GR\_APP'

--Permissão APP no grupo

);

END;

/

Para terminar o procedimento Temos que configurar o plano:

Alterar o resouce\_manager\_plan para nome plano a ser utilizado.

Comando*: alter system set resource\_manager\_plan='COPASA\_DIA' scope=both;*