

Oracle Resource Manager

Author	박민정
Creation Date	2019-03-10
Last Updated	2019-12-20
Version	1.0
Copyright© 2019 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2018-03-19	박민정	문서 최초 작성
2			
3			

Contents

Contents	2
1. Resource Manager 란?	3
1.1. 관리 가능한 포인트	3
2. Resource manager 의 구성 요소	4
2.1. Resource consumer group.....	5
2.2. Resource plan directive	6
2.3. Resource plan	6
3. Resource manager 관리자 권한	8
4. 제어 가능 요소	9
4.1. CPU	9
4.1.1. Management attribute.....	9
4.1.2. Utilization Limit	10
4.2. Parallel execution Server	10
4.2.1. Degree of parallelism limit	10
4.2.2. Parallel server limit.....	10
4.2.3. Parallel Query timeout.....	11
4.3. PGA (Program Global Area)	12
4.4. 약성 Query	12
4.4.1. Automatic Consumer Group Switching.....	12
4.4.2. Canceling SQL and Terminating Sessions.....	12
4.4.3. Execution Time Limit.....	12
4.5. Active session pool with Queuing	12
4.6. Undo pool	12
4.7. Idle time limit	13
5. Plan 작성	14
5.1. Resource plan 생성	14
5.2. 시나리오	21
6. Reference	24

1. Resource Manager란?

Oracle Database Resource Manager (이하 Resource Manager) 는 system resource 및 dbms resource 를 포함한 다중 부하(multiple workload)를 관리할 수 있는 기능입니다.

1.1. 관리 가능한 포인트

- 과도한 overhead
- 비효율적인 스케줄링
- 부적절한 자원배분
- parallel executuin server 및 active session 같은 특정 리소스의 관리 실패

Resource manager 는 위와 같은 자원배분 문제가 발생했을 때, 여러가지 방법으로 자원을 관리 할 수 있습니다.

- 시스템 부하와 상관 없이 특정 session 에 최소 CPU 사용을 보장.
- CPU time 을 다른 user 나 Application 의 특성에 맞게 분배 가능.
- 특정 사용자그룹에 대한 Parallel dgree 를 제한.
- parallel statement queue 에서 그룹의 우선순위를 관리.
- 그룹의 parallel execution server 수를 제한하여 자원 분배.
- Active session pool 을 통한 세션 관리.
- Resource 의 모니터링
- 특정 사용자 그룹에 속한 세션에 대한 PGA memory 제한
- runaway session 에 대한 관리
 - CPU, physical I/O, logical I/O 혹은 elapsed time 를 과도하게 사용하는 세션 혹은 호출을 자동으로 terminated 하거나 소비 제한이 낮은 resource group 으로 switching.
 - 실시간 SQL 모니터링을 통해 지정한 CPU, physical I/O, logical I/O, elapsed time 을 초과하는 SQL 문에 대한 상세 정보를 기록하거나 단순 로깅을 수행.
 - AWR 을 사용하여 지정한 resource 사용 수준을 초과한 SQL 문에 대한 분석을 함.
- 쿼리의 수행시간을 예상하여 장시간 수행되는 것을 제한 (수행시간 상한선 설정)
- session 의 총 유휴(idle)가능시간을 제한.(blocking session 만 지정 가능)
- instance 를 재시작 하지 않고 resource plan 을 동적으로 변경. (Oracle Scheduler 를 이용한 예약 변경 가능)

2. Resource manager의 구성 요소

Resource manager 는 아래 3 가지 요소로 구성되어 있습니다.

구성요소	설명
Resource consumer group	Resource requirement 에 따라 분류되는 사용자 그룹
Resource plan	Resource consumer group 에 적용되는 세부 플랜 모음.
Resource plan directive	Resource consumer group 에 적용되는 세부 플랜.

DBMS_RESOURCE_MANAGER package 의 subprogram 은 아래와 같습니다.

Subprogram (procedure)	Description
BEGIN_SQL_BLOCK	Indicates the start of a block of SQL statements to be treated as a group by resource manager
CALIBRATE_IO	Calibrates the I/O capabilities of storage
CLEAR_PENDING_AREA	Clears the work area for the resource manager
CREATE_CATEGORY	Creates a new resource consumer group category
CREATE_CONSUMER_GROUP	Creates entries which define resource consumer groups
CREATE_PENDING_AREA	Creates a work area for changes to resource manager objects
CREATE_PLAN	Creates entries which define resource plans
CREATE_PLAN_DIRECTIVE	Creates resource plan directives
CREATE_SIMPLE_PLAN	Creates a single-level resource plan containing up to eight consumer groups in one step
DELETE_CATEGORY	Deletes an existing resource consumer group category
DELETE_CONSUMER_GROUP	Deletes entries which define resource consumer groups
DELETE_PLAN	Deletes the specified plan as well as all the plan directives it refers to
DELETE_PLAN_CASCADE	Deletes the specified plan as well as all its descendants (plan directives, subplans, consumer groups)
DELETE_PLAN_DIRECTIVE	Deletes resource plan directives
DEQUEUE_PARALLEL_STATEMENT	Dequeues a parallel statement from the parallel statement queue
END_SQL_BLOCK	Indicates the end of a block of SQL statements that should be treated as a group by resource manager
SET_CONSUMER_GROUP_MAPPING	Adds, deletes, or modifies entries for the login and run-time attribute mappings
SET_CONSUMER_GROUP_MAPPING_PRI	Creates the session attribute mapping priority list
SET_INITIAL_CONSUMER_GROUP	Assigns the initial resource consumer group for a user (Caution: Deprecated Subprogram)
SUBMIT_PENDING_AREA	Submits pending changes for the resource manager

SWITCH_CONSUMER_GROUP_FOR_SESS	Changes the resource consumer group of a specific session
SWITCH_CONSUMER_GROUP_FOR_USER	Changes the resource consumer group for all sessions with a given user name
SWITCH_PLAN	Sets the current resource manager plan
UPDATE_CATEGORY	Updates an existing resource consumer group category
UPDATE_CONSUMER_GROUP	Updates entries which define resource consumer groups
UPDATE_PLAN	Updates entries which define resource plans
UPDATE_PLAN_DIRECTIVE	Updates resource plan directives
VALIDATE_PENDING_AREA	Validates pending changes for the resource manager

2.1. Resource consumer group

Resource consumer group 은 resource requirement 에 따라 분류되는 사용자 그룹이며, resource plan 을 할당할 수 있는 가장 작은 단위입니다. (개별 세션에 resource plan 을 할당하는 것이 아닙니다)

Session 이 생성되면 설정한 규칙(Mapping rule)에 따라 자동으로 그룹화 됩니다. 물론 관리자가 수동으로 전환하거나 수정할 수 있습니다. Data dictionary 에 지정되어있는, 수정/삭제가 불가능한 既생성 그룹(initial group)은 sys_group 과 other_group 등이 있고, 수동으로 28 개까지 생성 가능합니다.

Resource consumer group	Description
BATCH_GROUP	배치 작업을 위한 consumer group.
DSS_CRITICAL_GROUP	Critical DSS query 를 위한 consumer group.
DSS_GROUP	Non-critical DSS query 를 위한 consumer group.
ETL_GROUP	ETL job 을 위한 consumer group.
INTERACTIVE_GROUP	Interactive, OLTP 작업을 위한 consumer group.
LOW_GROUP	우선순위가 낮은 session 을 위한 consumer group.
ORA\$AUTOTASK	Maintenance task 를 위한 consumer group.
OTHER_GROUPS	모든 session 에 대한 Default consumer group. 반드시 모든 resource plan 에 세부 plan (directive plan)으로 포함되어야 함. Session-to-consumer group mapping rule 에 의한 덮어쓰기가 불가능.
SYS_GROUPS	System 관리자(sys/system)를 위한 consumer group. Session-to-consumer group mapping rule 에 의해 덮어쓰기가 가능.

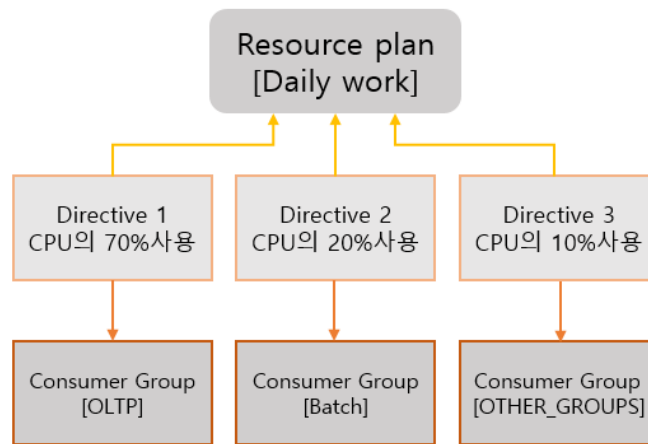
2.2. Resource plan directive

Resource Manager 가 consumer group 에 resource 를 할당하는 것은, 현재 active 되어 있는 resource plan 에 속한 resource plan directive(지침, 이하 directive)을 따릅니다.

Resource plan 와 Resource plan directive 는 부모자식 관계이며, directive 는 하나의 consumer group 을 가리킵니다. 당연히 여러 개의 directive 는 같은 group 에 동시에 적용될 수 없습니다.

Directive 에게는 resource 를 제한하는 몇 가지 방법이 있습니다. 예를들어 특정 group 에 전체 CPU 에 대한 일정 %를 할당할 수 있고, 이 group 에서 활동 가능한 세션 수도 제어할 수 있습니다. .

2.3. Resource plan



resource plan 은 resource consumer group 에 적용되는, '어떻게 resource 를 사용할지'에 대한 계획입니다. 미리 정의된 plan 외에 원하는 만큼 수동 작성이 가능하지만, group 당 하나의 plan 만 활성화(active)가 가능합니다. (이렇게 현재 활성화 된 플랜을 top resource plan 이라고 부릅니다) resource plan 이 활성화되면 아래에 달려있는 Directive resource plan 이 resource consumer group 에 대한 resource 할당을 제어합니다. 또한 모든 plan 은 반드시 OTHER_GROUPS 이라는 이름의 resource consumer group 이 묶여 있어야 합니다.

상기 그림에서 [Daily work]라는 이름의 Resource Plan 을 보면 OLTP Consumer group 으로 들어오는 session 들은 directive plan (Directive 1)에 의해 총 75%의 CPU 사용률을 보장 (guarantee)받습니다.

[Subplan]

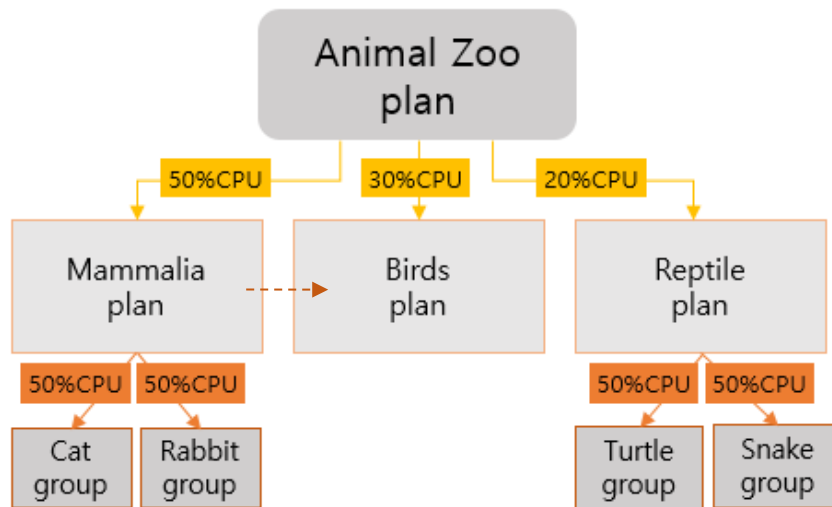
Directive plan 은 다른 resource plan 을 참조할 수 있는데, 이 경우 참조한 계획을 Subplan 이라고 합니다. Subplan 은 directive 를 이용하여 consumer group 과, 다른 subplan 에 자원을 배분합니다.

Top resource plan(active resource plan)은 자원을 ① consumer group 에 대한 자원 할당 ② subplan 에 대한 자원 할당, 이렇게 두가지 체계로 나눌 수 있습니다. 여기서 자원을 나눠받은 각 subplan 들은 또 일부의 자원을 다른 subplan 에 할당합니다. 이런식으로 계층적인 자원나눔 계획이 완성되는 것입니다.

resource subplan 은 resource plan 과 동일한 방법으로 작성합니다. 만약 subplan 으로만 사용계획을 작성하려면 패키지 프로시저 DBMS_RESOURCE_MANAGER.CREATE_PLAN 에서 SUB_PLAN 인수를 사용합니다.

어떤 top level plan 이든 간에, 한번에 오로지 하나의 subplan 만 참조할 수 있습니다. Subplan 은 OTHER_GROUPS 에 대한 directive 는 필요하지 않으며, resource plan 으로 설정할 수 없습니다.

Subplan 을 포함한 resource plan 은 아래 예시처럼 구성됩니다.



이 Animal zoo plan 을 살펴보면 우선, 포유류 플랜과 조류 플랜, 파충류 플랜에 CPU 를 각각 50%, 30%, 20% 할당합니다.

포유류 플랜에 할당되어 있는 50%의 CPU 를 또 반씩 나누어서 고양이 그룹과 토끼 그룹에 할당합니다. (결과적으로는 각 25%씩 할당 받음) 또한 파충류 플랜에 할당되어 있는 20%의 CPU 를 반씩 나누어서 거북이 그룹과 뱀 그룹에 할당합니다. Plan 은 loop 가 불가능 하여 다른 plan 의 group 에 간섭할 수 없습니다.

3. Resource manager 관리자 권한

Resource Manager 를 사용하기 위해서는 반드시 ADMINISTER_RESOURCE_MANAGER 권한이 필요합니다. Resource Manager 는 DBMS_RESOURCE_MANAGER PL/SQL package 안의 모든 procedure 의 실행권한을 가지고 있어야 합니다.

ADMIN 옵션을 가지고 있는 관리자는 DBMS_RESOURCE_MANAGER_PRIVS PL/SQL package 를 이용하여, 다른 user role 에게 관리자 권한을 줄 수 있습니다. 연관된 package procedure 는 아래와 같습니다.

Procedure	Description
GRANT_SYSTEM_PRIVILEGE	사용자나 role 에게 ADMINISTER_RESOURCE_MANAGER 시스템 권한을 부여
REVOKE_SYSTEM_PRIVILEGE	사용자나 role 에게 ADMINISTER_RESOURCE_MANAGER 시스템 권한을 박탈

아래와 같이 특정 user(여기서는 PINNTO user)에게 ㉠ ADMIN 옵션 없이 ㉡ administrative privilege 를 주면, PINNTO 는 DBMS_RESOURCE_MANAGER package 의 모든 procedure 를 실행할 수 있게 됩니다.

```
BEGIN
  DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE(
    GRANTEE_NAME => 'PINNTO',
    PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER',    -- ㉡
    ADMIN_OPTION  => FALSE);                             -- ㉠
END;
/
```

4. 제어 가능 요소

4.1. CPU

Resource manager 는 consumer group 간에 CPU 를 제한할당하거나, 사용하지 않는 여유분을 재분배하는 방법으로 CPU resource 를 관리합니다.

4.1.1. Management attribute

Management attribute 를 사용하면 consumer group 및 subplan 간에 CPU 를 어떻게 할당할 것인지 지정할 수 있습니다.

최대 8 단계의 cpu resource 할당 우선순위를 지정할 수도 있습니다. Level 2 의 consumer group 과 subplan 은 level 1 에서 할당되지 않았거나 사용하지 않은 여분의 resource 를 할당받습니다. 마찬가지로 level 3 의 resource consumer 는 일부 할당이 level 1 과 2 의 나머지 resource 만 할당 받는 식입니다. 이렇게 우선순위를 지정하는 것 외에, 모든 resource 가 어떻게 분배되고 사용되는지를 명시적으로 규정하는 방법을 제공하는 것이라고 할 수 있습니다.

이러한 우선순위는 MGMT_Pn attribute 를 사용하여(n은 1~8 사이의 정수) 지정할 수 있습니다. 예를 들어 MGMT_P1 directive attribute 로 level 1 의 resource 할당을, MGMT_P2 로 level 2 의 resource 할당을 지정하는 식입니다.

Degree of Parallelism Limit 및 Parallel Server Limit 같은 parallel statement directive attributes 를 사용하여 parallel statement queuing 를 제어하는 것도 가능합니다.

Consumer Group	Level 1 CPU Allocation	Level 2 CPU Allocation	Level 3 CPU Allocation
HIGH_GROUP	80%		
LOW_GROUP		50%	
MAINT_SUBPLAN		50%	
OTHER_GROUPS			100%

이렇게 지정한 경우, HIGH_GROUP 에 속해있는 application 들은 level1 의 우선순위를 가지므로 CPU 의 80%를 사용할 수 있습니다. 나머지 20%의 CPU 는 LOW_GROUP 과 MAINT_SUBPLAN 이 각각 level2 의 우선순위로 반씩 나누어 가집니다. (전체 CPU 의 50%씩 쓰는 것이 아닙니다) 그리고 level1 과 2 에서 사용하지 않은 여분의 CPU 를 OTHER_GROUPS 에서 사용할 수 있습니다.

CPU 할당비율은 반드시 고정된 값이 아닙니다. 충분한 부하가 없어 사용하지 않는 여분의 CPU 가 있을 경우, 나머지 consumer group 이나 subplan 들에게 할당해 줄 수 있습니다. 가장 마지막 level 에서도 사용하지 않는 여분의 CPU 는, 지정된 할당량에 비례하여 다른 모든 level 에 재배포될 수 있습니다.

4.1.2. Utilization Limit

UTILIZATION LIMIT attribute 를 사용하여 consumer group 의 CPU 사용율 상한을 지정할 수 있습니다. (이 limit 이 적용된 경우 plan 안에서의 재분배는 무시됩니다)

LOW_GROUP 이 CPU 의 90%를 사용한다고 가정했을 때, 사용자는 이 LOW_GROUP 이 속한 application 의 중요도가 낮다고 생각하여 최대 허용치를 지정하고 싶을 수 있습니다.

UTILIZATION LIMIT attribute 설정은 선택사항이며, 지정하지 않을 경우 default 는 no limit 입니다.

만약 Management attribute 50% 인 level 2 의 Consumer group A 가 있다고 가정해봅시다.

level 1 의 consumer group 들이 CPU 를 사용하지 않는 상황에, 같은 level 2 의 다른 group 들도 cpu 를 사용하지 않는 경우라면 이 A 는 제한 없이 resource 를 사용할 수 있습니다. 그러나 UTILIZATION LIMIT 가 30%로 지정되어있다면 아무리 유휴 CPU 가 있어도 전체의 75%까지만 사용할 수 없습니다. 우선순위가 낮은 group 이 모든 CPU 를 사용하지 않도록 제한하는 것입니다.

4.2. Parallel execution Server

Resource Manager 는 database 에 대한 Parallel execution servers 사용도 제한할 수 있습니다.

4.2.1. Degree of parallelism limit

사용자는 PARALLEL_DEGREE_LIMIT_P1 directive 를 이용해 consumer group 의 최대 parallel degree 를 지정할 수 있습니다. 이 제한은 consumer group 내의 한가지 작업 (one operation) 에 적용되며, consumer group 내의 모든 operation 의 parallel degree 를 제한하는 것은 아닙니다.

4.2.2. Parallel server limit

PARALLEL_SERVER_LIMIT directive 를 사용하여 특정 consumer gorup 이 사용할 수 있는 parallel execution server pool 의 최대 비율을 지정할 수 있습니다. 특정 consumer group 이 사용하는 parallel execution server 수는 해당 group 의 모든 세션이 사용하는 parallel execution server 의 합입니다. 우선순위가 높은 consumer group 의 PARALLEL_STMT_CRITICAL directive 를 BYPASS_QUEUE 로 설정하는 방법으로, parallel statement queue 를 우회하도록 할 수도 있습니다.

예를 들어, PARALLEL_SERVERS_TARGET 의 초기 파라미터에 의해 설정된 parallel execution server 의 수가 32 이고, MY_GROUP 라는 comsumer group 에 대해 PARALLEL_SERVER_LIMIT directive 가 50%로 설정되어 있다고 가정하면, MY_GROUP 은 16 개의 parallel execution server 를 사용할 수 있습니다.

만약 관리 속성(MGMT_P1, MGMT_P2 등등)을 가진 resource plan 이 있다면, 각 관리 속성마다의 FIFO queue 가 있다고 볼 수 있습니다. Resource plan 에 이런 관리속성이 존재하지 않는다

면 당연히 하나의 parallel statement queue(FIFO Queue)로 관리될 것입니다.

RAC 환경에서는, 'PARALLEL_SERVER_LIMIT * PARALLEL_SERVERS_TARGET / 100' 의 합이 parallel execution server 의 수가 됩니다. 이 이상 사용자가 사용하게 될 경우, parallel statement 는 queue 에 쌓입니다. RAC 에서 parallel 구문을 사용하지 않는 consumer group 이 있다면, 첫 parallel 구문이 PARALLEL_SERVER_LIMIT 에 의해 지정된 한도를 초과할 수 있습니다.

[Managing Parallel Statement Queuing Using Parallel Server Limit]

PARALLEL_SERVER_LIMIT 속성을 사용하면, consumer group 의 parallel 구문 queue 가 진입하는 시기를 특정할 수 있습니다. Oracle 은 각 consumer group 을 위한 parallel statement queue 를 관리할때, 아래와 같은 조건이 충족될 경우에만 parallel 구문이 queue 에 진입하도록 하고 있습니다.

- PARALLEL_DEGREE_POLICY 를 AUTO 로 세팅하는 경우
- 모든 consumer group 의 (Active 상태인) parallel execution server 수가 PARALLEL_SERVERS_TARGET 에 설정된 값을 초과하는 경우. (PARALLEL_SERVER_LIMIT 설정여부와는 관계없이 적용되며, 설정하지 않았을 경우에는 default 값이 100%)
- consumer group 에 대한 active 상태인 parallel execution server 수와 parallel 구문의 parallel degree 의 합이 active parallel execution servers 의 target 수를 초과하는 경우. (target 수는 PARALLEL_SERVER_LIMIT/100 * PARALLEL_SERVERS_TARGET 로 계산)

4.2.3. Parallel Query timeout

PARALLEL_QUE_TIMEOUT directive 의 attribute 를 설정하여 parallel 구문이 parallel statement queue 에서 대기할 수 있는 최대 시간(초)를 지정할 수 있습니다.

parallel statement queue 를 사용할 때 DB 에 parallel 구문을 사용할 만한 리소스가 충분하지 않으면, 리소스를 할당받을 수 있을 때까지 대기하게 됩니다. 하지만 리소스를 받을 수 있는 시간이 너무 오래 소요되어 queue 에서 대기해야 하는 경우가 생길 수 있으므로, 이런 경우 지정하여 time out 으로 빠져나가도록 할 수 있습니다.

각 consumer group 별로 PARALLEL_QUEUE_TIMEOUT 를 지정할 수 있고, resource plan 에 다른 속성(MGMT_P1, MGMT_P2, 등등..)을 지정하지 않는 경우에도 적용이 가능합니다.

Parallel 구문이 time-out 초과가 되면, 구문실행 종료와 함께 ORA-07454 가 발생합니다.

모든 세션에 대하여 parallel execution server 의 사용이 모니터링 되기는 하지만, parallel execution server directive attribute 는 PARALLEL_DEGREE_POLICY=AUTO 로 설정된 경우에

한하여 ACTIVE 상태의 세션에만 영향을 미칩니다. 만약 MANUAL 로 설정된 경우, 이 세션의 parallel 구문은 queue 에 넣지 않지만 PARALLEL_SERVER_LIMIT 의 한도를 결정하는 카운트에는 포함이 되고, 이 limit 을 초과하더라도 queue 에 포함되는 것은 아닙니다.

4.3. PGA (Program Global Area)

PGA resources 를 관리하기 위해서, Resource Manager 는 특정 consumer group 에 속한 각 세션들에게 할당할 pga 메모리의 양을 제한할 수 있습니다. 바로 DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE procedure package 의 session_pga_limit 파라미터를 설정하는 것입니다. 이 값은 허용가능한 최대 pga (MB) 이며, 세션이 이 이상 초과하여 pga 를 사용할 경우 ORA-10260 이 발생합니다. 이 제한은 Parallel query slaves 와 job queue processes 에도 적용됩니다.

4.4. 악성 Query

Resource Manager 는 성능에 영향을 주는 session 과 call 을 관리하기 위해서, 지정된 양의 CPU 와 물리적 혹은 논리적 I/O, elapsed time 등에 대해 핸들링 할 수 있습니다. cpu 사용이 적은 consumer group 으로 session 및 call 을 switch 시켜 allocate 하거나, 아예 terminate 해버리는 방법을 사용합니다.

4.4.1. Automatic Consumer Group Switching

조건이 충족되는 session 이 지정된 consumer group 으로 switching 될 수 있도록 하여 resource 할당을 제어할 수 있다. 보통은 우선순위가 높은 그룹에서 낮은 그룹으로 switching 을 시킵니다.

4.4.2. Canceling SQL and Terminating Sessions

Resource 사용에 대한 특정 directive(지시사항)을 지정해서, 오래 수행되는 sql query 를 취소하거나, 장기간 붙어있는 session 을 정리할 수 있습니다.

4.4.3. Execution Time Limit

Operation 이 수행되는 최대 수행시간을 지정해서, 오래 실행되는 operation 을 terminate 시킬 수 있습니다.

4.5. Active session pool with Queuing

최대치의 active session 수를 active session pool 정의로 제한할 수 있습니다.

4.6. Undo pool

각 consumer group 에 undo pool 을 할당할 수 있다. Consumer group 에 의해 생성될 수 있는

un-commit 트랜잭션의 총량을 제어한다고 볼 수 있습니다. 만약 이 한도를 초과하여 수행되는 DML 은 terminate 됩니다. Undo pool 에서 undo space 가 분리되기 전까지는 이 consumer group 에 속한 다른 구성원은 추가적인 data 조작을 할 수 없습니다.

4.7. Idle time limit

Session 이 idle 상태일 수 있는 시간을 지정할 수 있으며, time out 되면 terminate 시킵니다.

5. Plan 작성

5.1. Resource plan 생성

5.1.1. Simple plan

CREATE_SIMPLE_PLAN 을 이용하면 간편하게 resource plan 을 작성할 수 있습니다.

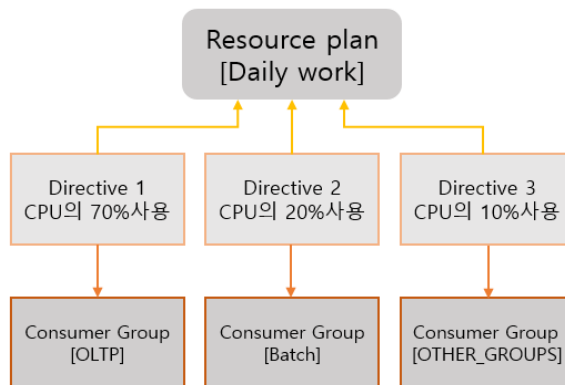
한번의 procedure 호출로 consumer group 의 생성과 resource 할당을 수행할 수 있고, pending area, 각 consumer group 의 생성 및 지시사항 지정 등은 여기서 설정하지 않아도 됩니다.

CREATE_SIMPLE_PLAN procedure 의 argument 는

SIMPLE_PLAN(플랜명), CONSUMER_GROUP n (그룹명), GROUP n _PERCENT(CPU 사용율)입니다.

만약 2.3 에서 예시로 나온 그림을 기준으로 한다면 아래와 같이 작성합니다.

(작성 유저는 SYS 로 하고 실행은 PUBLIC 권한으로 함)



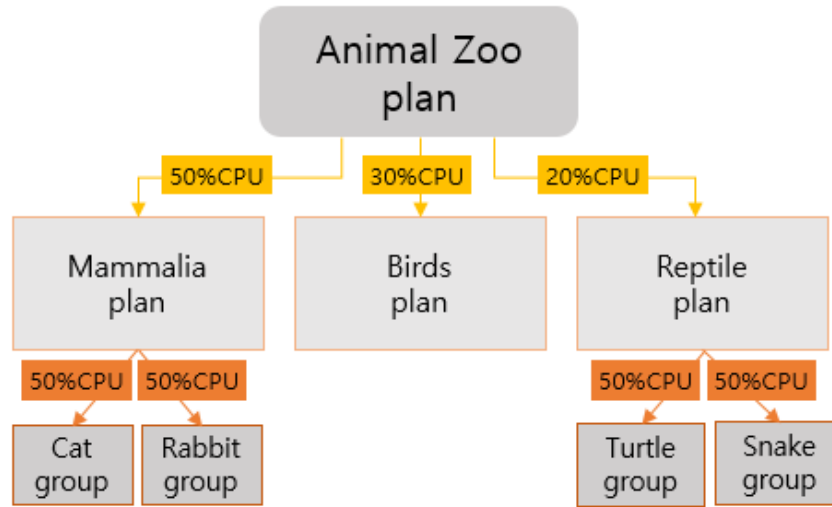
```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN(
SIMPLE_PLAN => 'Daily_work',
CONSUMER_GROUP1 => 'OLTP', GROUP1_PERCENT => 70,
CONSUMER_GROUP2 => 'Batch', GROUP2_PERCENT => 20);
END;
/
```

이 명령어로 지정된 Consumer group 은 SYS_GROUP 을 포함하여 4 개이며, OLTP 와 Batch 는 user-specified consumer groups 라고 부릅니다.

- SYS_GROUP : level 1 로 100% 사용
- OLTP : level 2 로 70% 사용
- Batch : level 2 로 20% 사용
- OTHER_GROUPS : level 3 로 100% 사용

5.1.2. Resource plan with Directive

그럼 이번에는 지시사항(directive)를 포함하여 더 복잡한 Resource plan 을 생성해보겠습니다.



우선 **Pending Area** 라는, 현재 실행 중인 애플리케이션에 영향을 주지 않고 새 resource plan 을 생성 혹은 업데이트, 삭제가 가능한 준비 영역을 만들어야 합니다. Resource plan 뿐 만 아니라 consumer group 에 대한 작업도 이곳에서 이루어집니다. 이곳에 생성한 plan 은 바로 활성화되는 것이 아니라, 단순히 dictionary 에 정보가 갱신되는 것이지만, 현재 활성화된 plan 을 수정하게 되는 경우에는 새로 정의된 plan 으로 바로 재활성화 됩니다.

Step	Description	Commands
0	권한 부여	<p>만약 SYS 유저로 작성하는 것이 아니라면, (편의를 위해 Plan 의 이름이 user 명이라고 가정하여) 아래와 같이 권한을 유저에게 따로 부여할 수도 있습니다.</p> <pre> BEGIN DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE(GRANTEE_NAME => 'MAMMALIA', PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER', ADMIN_OPTION => FALSE); DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE(GRANTEE_NAME => 'BIRDS', PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER', ADMIN_OPTION => FALSE); DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE(GRANTEE_NAME => 'REPTILE', PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER', ADMIN_OPTION => FALSE); END; / </pre>

1	pending area 생성	BEGIN
		DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA (); DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
2	Consumer group 생성	DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (CONSUMER_GROUP => 'CAT', COMMENT => 'OLTP_GROUP1'); DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (CONSUMER_GROUP => 'RABBIT', COMMENT => 'BATCH_GROUP'); DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (CONSUMER_GROUP => 'TURTLE', COMMENT => 'OLTP_GROUP2'); DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (CONSUMER_GROUP => 'SNAKE', COMMENT => 'APP_GROUP1'); DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (CONSUMER_GROUP => 'BIRDS', COMMENT => 'APP_GROUP2');

CAT 과 TURTLE 이 OLTP 업무, RABBIT 가 Batch 업무, SNAKE 가 Application 에서 들어오는 업무 (Batch+OLTP)를 위한 consumer group 이라고 가정합니다. Birds 유저로 들어오는 세션들은 무조건 BIRDS consumer group 으로 분류하도록 합니다.

Session 을 분류하는 기준은 보통 3 가지인데, @user 명, @service 명, @client program 입니다. 우리가 만들 상기 plan 의 그림을 표로 정리하면 아래와 같습니다.

업무명	업무분류	USER 명	Service 명	Program(Module)명
CAT	OLTP	Mammalia	ANIMAL	CAT.exe
RABBIT	Batch			RABBIT.exe
Other	Other	Birds		JDBC Client
TURTLE	OLTP	Reptile		TURTLE.exe
SNAKE	OLTP+Batch		SNAKE	JDBC Client

그리고 Mapping rule 이 베이스로 삼는 session 속성의 두가지 유형은 @Login, @run-time 이다. Login 속성은 resource manager 가 session 의 초기(initial) consumer group 을 결정하는데 이용 되고, run-time 속성은 session 이 login 된 동안 항상 적용되는 속성이며, 다른 consumer group 에 재할당 할 때 사용하는 속성이다.

SET_CONSUMER_GROUP_MAPPING 에 사용하는 속성으로 예를 들면, ORACLE_USER, SERVICE NAME, CLIENT_ID 등은 LOGIN 타입이며, MODULE_NAME, ORACLE_FUNCTION 등은 Run-time 타입이다.

[속성 분류]

Attribute	Type	Description
ORACLE_USER	Login	Oracle 의 user name
SERVICE_NAME		Database service name
CLIENT_OS_USER		Client 에서 접속한 OS user name
CLIENT_PROGRAM		Client program name
CLIENT_MACHINE		DB 에 접속한 Computer name
CLIENT_ID		DBMS_SESSION.SET_IDENTIFIER 로 설정한 client 구분자
MODULE_NAME	Run-time	DBMS_APPLICATION_INFO.SET_MODULE 혹은 equivalent OCI 속성으로 설정된, 현재 running 중인 application module.
MODULE_NAME_ACTION (module_name.action_name)		현재 module 과, 아래 procedure 를 통해 OCI 환경설정으로 셋팅한 action 을 결합한 속성. - DBMS_APPLICATION_INFO.SET_MODULE - DBMS_APPLICATION_INFO.SET_ACTION
SERVICE_MODULE (service_name.module_name)		Service 와 module 명을 결합한 속성.
SERVICE_MODULE_ACTION (service_name.module_name.action_name)		Service 와 module 명과 action 을 결합한 속성.
ORACLE_FUNCTION (DATALOAD, BACKUP, COPY)		RMAN 혹은 Datapump 작업. (관련 function 을 사용시 자동분류)

Step	Description	Commands
3	Consumer gorup 에 session 을 mapping.	<pre> BEGIN -- user 로 mapping DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER.ORACLE_USER, 'BIRDS', 'BIRDS_GROUP'); -- service 명으로 mapping DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER.SERVICE_NAME, 'SNAKE', 'SNAKE_GROUP'); -- 같은 service 내에서 다른 module 명으로 mapping 1 DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER. SERVICE_MODULE, 'ANIMAL.CAT.exe', 'CAT_GROUP'), -- 같은 service 내에서 다른 module 명으로 mapping 2 </pre>

		<pre> DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER. SERVICE_MODULE, 'ANIMAL.RABBIT.exe', 'RABBIT_GROUP'), -- 같은 service 내에서 다른 module 명으로 mapping 3 DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER. SERVICE_MODULE, 'ANIMAL.Turtle.exe', 'TURTLE_GROUP'); END; / </pre>
--	--	---

Resource manager 를 enable 하기 전에, 반드시 user 의 session 들을 각 consumer group 에 mapping 해야 한다. Session 의 특성에 따라, 생성 즉시 바로 mapping 될 수 있도록 mappimap rule 을 생성할 수도 있고, 수동으로 다른 consumer 그룹에 넣어버릴 수도 있다. 또한 자동으로 (dynamically) Resource manager 가 resource 상황에 따라, 돌고 있는 세션의 group 을 이동 (switching, 물론 낮은 우선순위의 널널한 rule 로 이동된다!)시키도록 할 수도 있다.

Mapping rule 의 우선순위는 rule 간의 충돌을 방지하기 위해 꼭 필요하다.

만약 reptile user 가 통상적으로 [ANIMAL service 를 이용하여 database 에 연결]된다고 가정한다. 그리고 관리자가 Mapping rule A 를 만들어서 [reptile user 는 taurtle group 에서 시작]한다고 하고, Mapping rule B 에서 [ANIMAL service 는 Rabbit group 에서 시작] 한다고 명시하면 Rule 간의 충돌이 생긴다. reptile user 중에는 SNAKE sevice 로 들어오는 session 들이 있기 때문이다. 이러한 상황에서 우선순위 설정이 필요하다.

모든 user 및 session 은 2.1 에서 설명한 사전정의된 consumer group 중 SYS_GROUP 에 기본적으로 mapping 된다. Datapump 를 사용하여 data 를 load 하거나, RMAN 을 사용하여 backup 또는 copy 작업을 수행하는 session 들도 미리 사전정의된 consumer group 중 하나에 자동으로 mapping 된다.

Step	Description	Commands
4	Resource plan 생성	<pre> BEGIN DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN => 'ANYMALZOO', COMMENT => 'ROOT RESOURCE PLAN'); END; / </pre>
5	Subplan or Resource Plan directive (지시사항) 생성	<pre> BEGIN DBMS_RESOURCE_MANAGER.CREATE_PLAN (PPLAN => 'MAMMALIA_PLAN',, MGMT_MTH => 'RATIO', </pre>

		<pre> COMMENT => Allocate CPU to cat and rabbit group'); DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (PLAN => 'MAMMALIA_PLAN', GROUP_OR_SUBPLAN => 'CAT_GROUP_PLAN', COMMENT => 'plan for cat', MGMT_P1 => 50); DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (PLAN => 'MAMMALIA_PLAN', GROUP_OR_SUBPLAN => 'RABBIT_GROUP_PLAN', COMMENT => 'plan for rabbit', MGMT_P1 => 50); DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN => 'BIRDS_PLAN', MGMT_MTH => 'RATIO', COMMENT => Allocate CPU to birds app group"); DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN => 'BIRDS_PLAN', GROUP_OR_SUBPLAN => 'BIRD_GROUP_PLAN', COMMENT => 'plan for bird', MGMT_P1 => 30); DBMS_RESOURCE_MANAGER.CREATE_PLAN (PLAN => 'BIRDS_PLAN', MGMT_MTH => 'RATIO', COMMENT => Allocate CPU to birds app group"); DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (PLAN => 'SERVICE_LEVEL_PLAN', GROUP_OR_SUBPLAN => 'BRONZE_CG', COMMENT => 'Bronze service level customers', MGMT_P1 => 2); DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (PLAN => 'SERVICE_LEVEL_PLAN', GROUP_OR_SUBPLAN => 'OTHER_GROUPS', COMMENT => 'Lowest priority sessions', MGMT_P1 => 1); </pre>
--	--	---

		END;
6	Pending Area 를 활성화 (validate).	BEGIN DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA(); END; /
7	Pending Area 를 submit.	BEGIN DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA(); END; /

5.2. 시나리오

지금까지의 내용을 참조하여 하나의 시나리오를 만들어 보겠습니다.

우리가 구현할 **DWPLAN** resource plan 의 개괄적인 정보는 아래 표와 같습니다.

DWPLAN General information									
업무명	업무분류	Oracle USER 명	Service 명	Program (module)명	요구사항				
					CPU	UNDO pool	PGA	IDLE	Threshold
CAT	OLTP	Mammalia	ANIMAL	CAT.exe		①100M			
RABBIT	Batch			RABBIT.exe					
BIRD	Bird app	Bird		BIRDS.exe				②1min	
TURTLE	OLTP	Reptile		JDBC Client	③30%				④EXEC 5mins => KILL
SNAKE	Snake app		SNAKE	JDBC Client	⑤20%	⑥10M	⑦10M		
Consumer Group Mapping Constants									
Consumer group	name		Attribute		=> Specified Value				
OTHER_GROUP									
MAMALIA_GROUP	oracle_user		'ORACLE_USER'		① Mammalia				
BIRD_GROUP	oracle_user		'ORACLE_USER'		② Bird				
TURTLE_GROUP	service_name. service_module		'SERVICE_NAME','SERVICE_MODULE'		③④ ANIMAL . JDBC Client				
SNAKE_GROUP	service_name		'SERVICE_NAME'		⑤⑥⑦ SNAKE				
Additional requirements									
<ul style="list-style-type: none"> CPU 할당: 우선순위는 TURTLE_GROUP : LEVEL 2 / SNAKE_GROUP : LEVEL 3. / OTHER_GROUPS : LEVEL 4. Turtle group 은 수행시간이 5 분 이상일 경우 session kill 수행. (30 초 대기 후) 									

Step	Description	Commands
1	consumer group 생성	<pre> BEGIN dbms_resource_manager.clear_pending_area(); dbms_resource_manager.create_pending_area(); dbms_resource_manager.create_consumer_group(consumer_group => 'MAMMALIA_GROUP', comment=>'cat oltp and rabbit bat group'); dbms_resource_manager.create_consumer_group(consumer_group=> 'BIRD_GROUP', comment=>'bird app group'); dbms_resource_manager.create_consumer_group(consumer_group=> 'TURTLE_GROUP', comment=>'turtle oltp group'); dbms_resource_manager.create_consumer_group(consumer_group=> 'SNAKE_GROUP', comment=>'snake app group'); </pre>
2	Plan 생성 및 directive 생성	<pre> dbms_resource_manager.create_plan(plan=>'DWPLAN', comment=>'DW resource Plan'); dbms_resource_manager.create_plan_directive(plan => 'DWPLAN', comment => 'DWPLAN for MAMMALIA_GROUP', group_or_subplan => 'MAMMALIA_GROUP', undo_pool => 102400); dbms_resource_manager.create_plan_directive(plan => 'DWPLAN', comment => 'DWPLAN for BIRD_GROUP', group_or_subplan => 'BIRD_GROUP', max_idle_time => 60); dbms_resource_manager.create_plan_directive(plan => 'DWPLAN', comment => 'DWPLAN for TURTLE_GROUP', group_or_subplan => 'TURTLE_GROUP', mgmt_p2 => 30, max_est_exec_time => 300, switch_group => 'kill_session', switch_time => 30); dbms_resource_manager.create_plan_directive(plan => 'DWPLAN', comment => 'DWPLAN for SNAKE_GROUP', group_or_subplan => 'SNAKE_GROUP', mgmt_p3 => 20, </pre>

		<pre>undo_pool => 1024, session_pga_limit => 10);</pre>
3	<p>유저에게 consumer gorup 을 initialize.</p>	<pre>dbms_resource_manager_privs.grant_switch_consumer_group(grantee_name => 'MAMMALIA', consumer_group => 'MAMMALIA_GROUP', grant_option => TRUE); dbms_resource_manager_privs.grant_switch_consumer_group(grantee_name => 'BIRD', consumer_group => 'BIRD_GROUP', grant_option => TRUE); dbms_resource_manager.set_initial_consumer_group('MAMMALIA','MAMMALIA_GROUP'); dbms_resource_manager.set_initial_consumer_group('BIRD','BIRD_GROUP');</pre>
4	<p>consumer group mapping.</p>	<pre>DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(dbms_resource_manager.oracle_user, 'MAMMALIA', 'MAMMALIA_GROUP'); DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(dbms_resource_manager.oracle_user, 'BIRD', 'BIRD_GROUP'); DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(dbms_resource_manager.service_name, 'SNAKE', 'SNAKE_GROUP'); DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(dbms_resource_manager.service_module, 'ANIMALJDBC Thin Client', 'TURTLE_GROUP'); dbms_resource_manager.validate_pending_area(); dbms_resource_manager.submit_pending_area(); END; /</pre>
	<p>Parameter setting</p>	<pre>alter system set resource_manager_plan = 'DWPLAN';</pre>

상기 설정한 조건 중, SNAKE Service 로 접속하는 경우 pga 을 10m 만 사용하도록 하였는데, 실제로 확인해보면 10m 이상으로는 사용이 불가능 한 것을 알 수 있습니다.



6. Reference

- `$ORACLE_HOME/rdbms/admin/dbmsrmand.sql`
- https://docs.oracle.com/en/database/oracle/oracle-database/12.2/arpls/DBMS_RESOURCE_MANAGER.html#GUID-9876C289-99E4-416B-AB6F-D8318642053E

< 끝 >