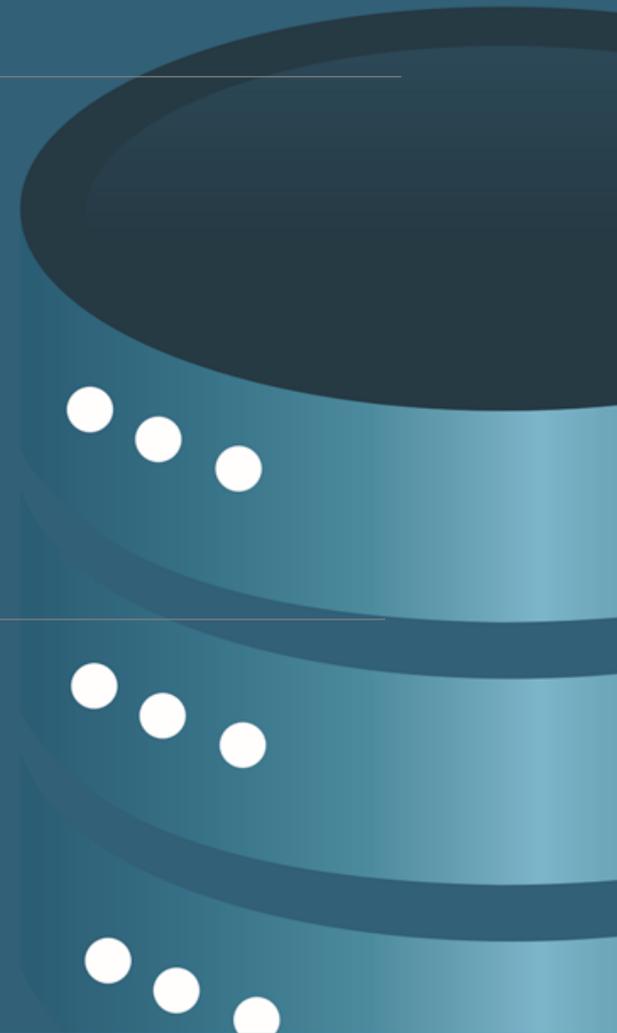

AULA 12

MÓDULO ORACLE RAC

PERFORMANCE ORACLE RAC



Na aula anterior...

Na aula 11 vimos a instalação do RAC One Node e suas vantagens.

Na aula de hoje

Vamos estudar performance de Oracle RAC.

Algumas correções e formas de pró-atividade

Tuning em Ambiente RAC

Sempre que nos deparamos com um problema aparentemente de performance em Oracle RAC, devemos primeiramente analisar a fonte deste problema.

Como toda análise de ambiente Oracle, a primeira coisa a se fazer é uma análise de qual o evento de espera ligado à suposta lentidão.

Para isso utilizamos a famosa query de identificação de Oracle Wait interfaces (OWI) largamente utilizada no curso de Tuning.

Tuning em Ambiente RAC

Query OWI:

```
SELECT i.instance_name, w.event, COUNT (*), SUM (seconds_in_wait)
FROM gv$instance i,gv$session_wait w
WHERE i.inst_id = w.inst_id
and event not like 'Streams AQ%'
and event not like 'queue messages%'
and event not like 'SQL*Net%'
and (w.inst_id, SID) IN (SELECT inst_id, SID FROM gv$session WHERE status = 'ACTIVE' AND username IS NOT
NULL)
GROUP BY i.instance_name,event
ORDER BY 3 DESC;
```

Tuning em Ambiente RAC

Sempre que nos é reportada alguma lentidão em ambiente RAC, devemos levar em consideração se estamos com bom balanceamento da carga das conexões.

Para isto basta rodar o script abaixo:

```
select inst_id, count(*)  
from gv$session  
group by inst_id;
```

Esta query bem simples irá nos mostrar a quantidade de conexões de cada inst_id (cada instância do RAC). E com ela conseguiremos ter uma idéia de como estão divididas as cargas de conexões.

Lembrando que este balanceamento é feito automaticamente pelo Oracle Clusterware do RAC.

Iniciando análise Tuning RAC

Inicialmente começamos a resolução de problemas analisando cada servidor onde estão alocadas as instâncias do RAC.

Com o comando `top -c` por exemplo analisamos como está o load da máquina, a carga de memória, consumo de CPU, etc.

É bem comum em Oracle RAC o problema do conjunto se resolver quando é sanada alguma situação pontual na instância isoladamente.

Então o ponto de iniciar a análise é ponto a ponto do RAC, ou seja, nó a nó a nível de Sistema Operacional mesmo.

Iniciando análise Tuning RAC

Outra dica muito importante é a utilização do AWR para verificar performance no Oracle RAC.

Rodar AWR em ambiente RAC é um pouco diferente do que vemos em ambiente single conforme abaixo.

No caso do de querer investigar através do AWR poderá utilizar a seguinte sequência:

```
cd $ORACLE_HOME/rdbms/admin
```

```
sqlplus "/ as sysdba"
```

```
SQL>@awrrpt.sql;
```

Com o script acima irá gerar AWR por instância

Caso queira rodar um global que traga dados globais de ambas as instâncias utilizar:

```
SQL>@awrgrpt.sql
```

Iniciando análise Tuning RAC

Outra ferramenta da família Enterprise que você poderá utilizar é o relatório ADDM (Automatic Database Diagnostic Monitor), onde serão sugeridas dicas baseadas em estudo dos AWR.

Tudo gerado pelo próprio Oracle através da página do Enterprise Cloud Control ou através de geração manual conforme abaixo.

Primeiramente verificamos que existe algum laudo do ADDM:

```
Select a.execution_end, b.type, b.impact, d.rank, d.type,  
'Message          : '||b.message MESSAGE,  
'Command To correct: '||c.command COMMAND,  
'Action Message   : '||c.message ACTION_MESSAGE  
From dba_advisor_tasks a, dba_advisor_findings b,Dbas_advisor_actions c, dba_advisor_recommendations d  
Where a.owner=b.owner and a.task_id=b.task_idAnd b.task_id=d.task_id and b.finding_id=d.finding_id and  
a.task_id=c.task_id and d.rec_id=c.rec_id and a.task_name like 'ADDM%' and a.status='COMPLETED'  
Order by b.impact, d.rank
```

Relatório ADDM

Posteriormente podemos querer montar nossa própria análise de ADDM pelo modelo abaixo.

Suponha que eu queira criar um ADDM do snapshot 970 ao 1032:

```
BEGIN
```

```
-- Cria-se a ADDM task.
```

```
DBMS_ADVISOR.create_task (advisor_name => 'ADDM',  
task_name      => '970_1032_AWR_SNAPSHOT',  
task_desc     => 'Advisor for snapshots 970 to 1032.');
```

Relatório ADDM

-- Seta o snapshot inicial e final.

```
DBMS_ADVISOR.set_task_parameter (  
task_name => '970_1032_AWR_SNAPSHOT',  
parameter => 'START_SNAPSHOT',  
value    => 970);
```

```
DBMS_ADVISOR.set_task_parameter (  
task_name => '970_1032_AWR_SNAPSHOT',  
parameter => 'END_SNAPSHOT',  
value    => 1032);
```

Relatório ADDM

-- Execute a task

```
DBMS_ADVISOR.execute_task(task_name => '970_1032_AWR_SNAPSHOT');  
END;  
/
```

--Verificando relatório

```
SET LONG 100000  
SET PAGESIZE 50000  
SELECT DBMS_ADVISOR.get_task_report('970_1032_AWR_SNAPSHOT') AS report  
FROM dual;
```

Relatório ADDM conclusões

Do relatório do ADDM normalmente surgem informações e dicas muito importantes como dicas de utilização de índices em consultas, dicas de utilização de outros advisors, necessidade de melhora em queries.

Na maioria das vezes uma análise de tuning nos remete ao desenvolvimento das queries nas aplicações e suas melhoras necessárias através de um tuning de SQL e não mais de banco de dados.

Dicas importantes de Tuning Oracle RAC

Dentre diversas dicas que poderíamos citar aqui neste momento estão as mais comuns e que normalmente geram maiores resultados:

- 1- Procure ao máximo reduzir consultas que façam full table scan em ambientes OLTP;
- 2- Use ASSM (Automatic Segment Space Management);
- 3- Quando utilizar sequences, diminua o cache armazenado;
- 4- Utilize sempre que possível Oracle Partitioning , o particionamento Oracle, que diminui muito o volume de dados trafegado (item pago);
- 5- Construa queries que façam o mínimo de parse possível (ver utilização de bind variable);
- 6- Remova índices que não sejam seletivos;
- 7- Garanta uma boa comunicação de interconnect (conforme citado sua relevância nas primeiras aulas.

Balanceamento por serviço

Geralmente o Oracle é balanceado por quantidade simples de conexões, ou seja, o clusterware divide para que tenhamos quantidades proporcionais de conexões divididas pelos nós do RAC.

Porém pode ser que tenhamos a necessidade de fazer o balanceamento por serviço e não por conexões.

Isso acontece quando por exemplo temos servidores de características diferentes umas das outras e queremos que a mais robusta execute serviços que sabemos serem mais "pesados".

Para isso criamos serviços distintos em servidores distintos:

Comando:

```
srvctl add service -db ORCL -service FATURAMENTO -preferred RAC02 -available RAC01;
```

Com isto criamos um serviço chamado FATURAMENTO que rodará preferencialmente no nó RAC02 mas que também poderá rodar no nó RAC01.

Balanceamento por serviço

Assim, basta chamar o serviço no tnsnames e ele encontrará o servidor correto automaticamente e direcionará a instância correta.

Caso algo ocorra com a instância onde o serviço está ativo, ele poder rodar na outra para qual ele está disponível (available).

Alguns comandos interessantes

- Start a named service on all configured instances:

```
$ srvctl start service -db orcl -service AP
```

- Stop a service:

```
$ srvctl stop service -db orcl -service AP -instance orcl4
```

- Disable a service at a named instance:

```
$ srvctl disable service -db orcl -service AP -instance orcl4
```

- Set an available instance as a preferred instance:

```
$ srvctl modify service -db orcl -service AP -instance orcl5  
-preferred
```

- Relocate a service from one instance to another:

```
$ srvctl relocate service -db orcl -service AP -oldinst  
orcl5 -newinst orcl4
```

Na próxima aula...

Na próxima aula teremos algumas dicas e conclusões a respeito de Oracle RAC.

AULA 12

fim

