

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Introdução do Treinamento

Para uso exclusivo de "Jeniffer Lúcia da Silva - CPF 059.036.659-93"

Objetivo

- Apresentar as melhores práticas, dicas e conceitos para otimizar (tunar) o desempenho de Bancos de Dados Oracle 10G em diante, abordando os seguintes tópicos:
 - Tuning de instância;
 - Tuning de Bancos de Dados;
 - Automatic Workload Repository (AWR);
 - Automatic Database Diagnostics Monitor (ADDM);
 - Estatísticas do Otimizador de queries;
 - Wait Events.

Objetivo

- Capacitar o aluno a identificar, diagnosticar e resolver (**otimizar**) problemas de performance em Bancos de Dados Oracle.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.259-95"

Agenda

- 1ª aula:

- 1. Introdução
- 2. Identificando problemas de performance
- 3. Identificando problemas de performance através dos *Wait Events*
- 4. Utilizando o Statspack, o AWR e o ADDM
- 5. Otimizando uma instância de Banco de Dados (início)

- 2ª aula:

- 5. Otimizando uma instância de Banco de Dados (fim)
- 6. Otimizando o Banco de Dados
- 7. Wait Events X Sugestões de Otimização
- 8. Exercícios Finais

Material

- Totalmente elaborado pelo instrutor do treinamento de acordo com a sua experiência, conhecimentos e pesquisas;
- Parte téorica resumida para **facilitar e agilizar o aprendizado**;
- Composto adicionalmente por **scripts** que aplicam todos os conceitos, técnicas e dicas aprendidas durante o treinamento e que servirão para referências futuras;

Obs.: O material passa por atualizações e correções constantes.

Ambiente recomendado

- Para montar o ambiente de treinamento recomendado para os treinamentos [Fabioprado.net](http://www.fabioprado.net) é necessário:

- Instalar **Oracle VM Virtual Box** versão 4.1.8 ou superior:

Ver artigo <http://www.fabioprado.net/2016/01/instalando-o-oracle-vm-virtual-box.html>

- Importar **Appliance** que contém o ambiente de treinamento;

- Gravar na máquina virtual scripts fornecidos pelo instrutor.

Obs.: *Iniciar a VM neste momento.*

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

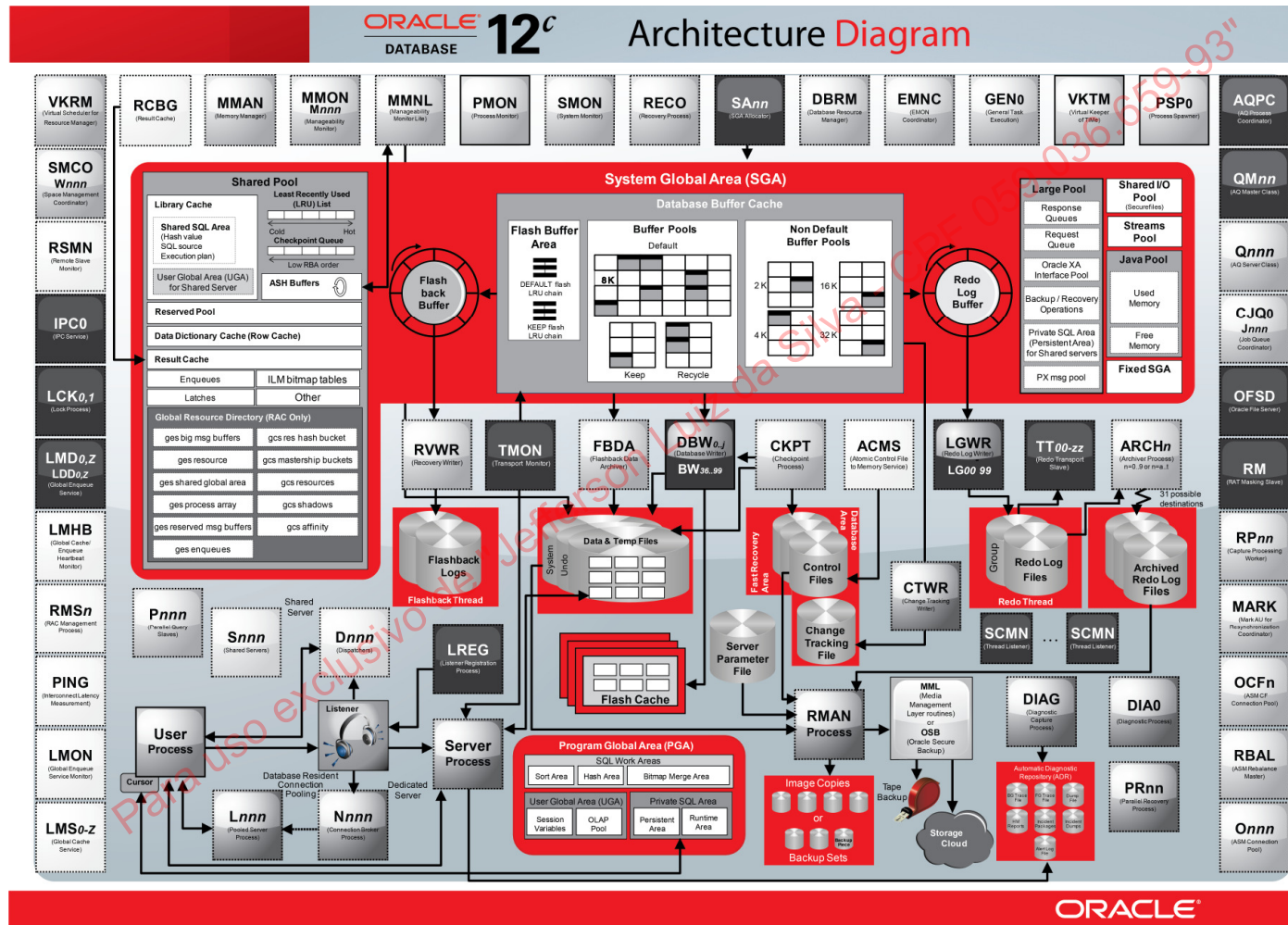
Fábio Prado

Capítulo 1

Introdução Performance Tuning

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Arquitetura do Oracle DB



História e curiosidades

- Em 1979, foi lançado o **Oracle 2**, o primeiro BD relacional (com SQL) comercializado no mundo;
- Em 1983, o **Oracle 3** deixou de fazer bloqueios de leitura e no **Oracle 4** (1984) foi implantada a **leitura consistente**;
- Em 1989, o **Oracle 6** já podia fazer **bloqueios no nível de linhas** e backups quentes;
- Em 1993, no **Oracle 7**, foram criados o **Otimizador Baseado em Custo** (CBO), Triggers, Functions, Stored Procedures, Packages, SQL Dinâmico, Jobs, **paralelismo** (em queries, criações de índices e cargas de dados), **índices bitmap**, **histogramas**, trace etc.;

História e curiosidades

- Em 1997, o **Oracle 8** já suportava tabelas e índices particionados, parallel dml, tabelas IOT, SQL Loader, tabelas temporárias, Statspack etc.;
- Em 2001, o **Oracle 9** já suportava reorganização online de índices e tabelas, o CBO passou a considerar CPU, tabelas externas e a primeira versão de clusters em BD Oracle, na versão 9i;
- Características de BDs **OLTP x OLAP (*)**
 - OLTP: Entre 70% e 80% leitura + 20% ou 30% escrita
 - OLAP: 100% leitura (no período em que o usuário está utilizando o BD)

O que é tuning?

- **Tuning** se escreve com *somente* 2 Ns! **Tunning** com 3 Ns está errado!
 - Ver artigo [O que é Tuning?](#)
- Em TI, refere-se basicamente ao conceito de propor e aplicar mudanças para **otimizar o desempenho** na recuperação ou atualização de dados;

O que é tuning?

- É um termo que desperta um interesse cada vez maior nos profissionais de TI, devido ao aumento:
 - Do legado de sistemas corporativos (ERPs, GEDs etc.) e sistemas web;
 - Da quantidade de usuários de BDs;
 - Da **quantidade de dados**.

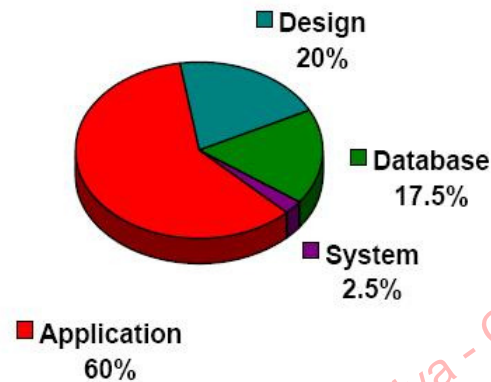
Atividades de tuning

- As 3 atividades de tuning que podem ser realizadas em um BD Oracle são:
 - Planejamento de performance:
 - Definição do ambiente em que o BD será instalado: hardware, software, sistema operacional, infraestrutura de rede etc.
 - Tuning de instância e BD:
 - Ajuste de parâmetros e configurações do BD (inclusive configurações do SO) para melhorar performance.
 - SQL Tuning:
 - Otimização de instruções SQL, incluindo procedimentos relacionados, que podem ajudar nesta tarefa (Ex.: Criação de índices).

Objetivos do tuning

- Minimizar tempo de resposta e recuperação de dados;
- Otimizar a taxa de transferência de dados;
- Minimizar concorrência de acesso aos dados;
- Otimizar a capacidade de carga do Banco de Dados.

Onde tunar?



Estatísticas de problemas de performance em BD Oracle - Fonte: DBAGroup

- 60% dos problemas de performance são decorrentes de **SQL ruim**;
- 20% dos problemas de performance são decorrentes do projeto do BD;
- 17,5% dos problemas de performance são decorrentes de **configurações do BD ou instância de um BD.**

Por que tunar o BD?


Ver artigo [Tuning - Banco de Dados](http://www.taskblog.com.br/01/tuning-banco-de-dados) de Sandro Lopes, publicado em 09/01/12: <http://www.taskblog.com.br/01/tuning-banco-de-dados>

“Um amigo estava me contando que a empresa onde trabalha **trocou um dos seus servidores por uma máquina mais potente**, com vários gigabytes de memória e muitos processadores. O objetivo era fazer com que um determinado programa, considerado crítico pela empresa, fosse executado com mais performance, podendo assim atender melhor a demanda dos vários departamentos que o acessavam.


Mas infelizmente, esta atualização tecnológica (e o investimento financeiro que não foi modesto), não atingiram os objetivos esperados. Houve sim uma melhora na performance do programa, mas ainda estava muito abaixo do esperado.”

...

Por que tunar o BD?

 Oracle Senior DBA Group

Debates Integrantes Promoções Empregos Pesquisar Mais...


Seguir Hassaan

A query related with Oracle performance

Hopefully you guys can help me with your experience . We were planning to clone a database that is standalone database in our current server . This database was exported and imported to a new better server . We also copied the stats and made sure every parameter in Oracle config is same . Infact the way the two servers hard drives are mounted that is also same and the linux parameters in both the machines are also same . Having said we expected the performance of the cloned database to be better as it has been migrated to a better server which is almost double in everything (CPU,IO) etc . Now we were shocked to see that the new server database is behaving very slow .

(We are running Oracle 10G)

Current Server : 16 CPU , 750 GB , 8 GB RAM

New server : 32 CPU , 750 GB and 16 RAM

Any ideas ? . According to my knowledge shouldn't oracle behave the same or better way if everything between two servers are identical . Any suggestions . Thanks in advance.

7 dias atrás

Quando tunar?

- Não dá para “fazer tuning” sem saber qual é o problema do Banco de Dados, portanto, identifique primeiro o problema dele;
- Desafios da metodologia tradicional de performance tuning:
 - Coletar dados estatísticos;
 - Analisar dados coletados;
 - Definir e implementar solução.

Importante

- SQL ruim é a causa nº 1 de problemas de performance;
- Aplicações mal projetadas **executam mal até mesmo em um BD bem tunado** (com bom hardware e bem configurado);
- **Todo trabalho tem que ter ROI**, por isso, investir tempo demais em soluções demoradas ou complexas, muitas vezes não é a melhor escolha;

Importante

- Após alterar, teste e analise:
 - Não acredite na sorte, lendas, **mitos** ou boatos;
 - Verifique se foi obtido o ganho de performance;
 - Se necessário volte atrás.
- Não existe uma **fórmula mágica** para tunar o desempenho de todos os BDs:
 - Se existisse, ela já teria sido aplicada na instalação do BD!
- Primeiro o diagnóstico e depois o tuning.

O que é mais rápido, COUNT(1) ou COUNT(*)?

```
SELECT COUNT(1) FROM OE.ORDER_ITEMS;
```

plan FOR bem-sucedido.
PLAN_TABLE_OUTPUT

Plan hash value: 3419397814

Id	Operation	Name	Rows	Cost	(%CPU)	Time
0	SELECT STATEMENT		1	3	(0)	00:00:01
1	SORT AGGREGATE		1			
2	TABLE ACCESS FULL	ORDER_ITEMS	665	3	(0)	00:00:01

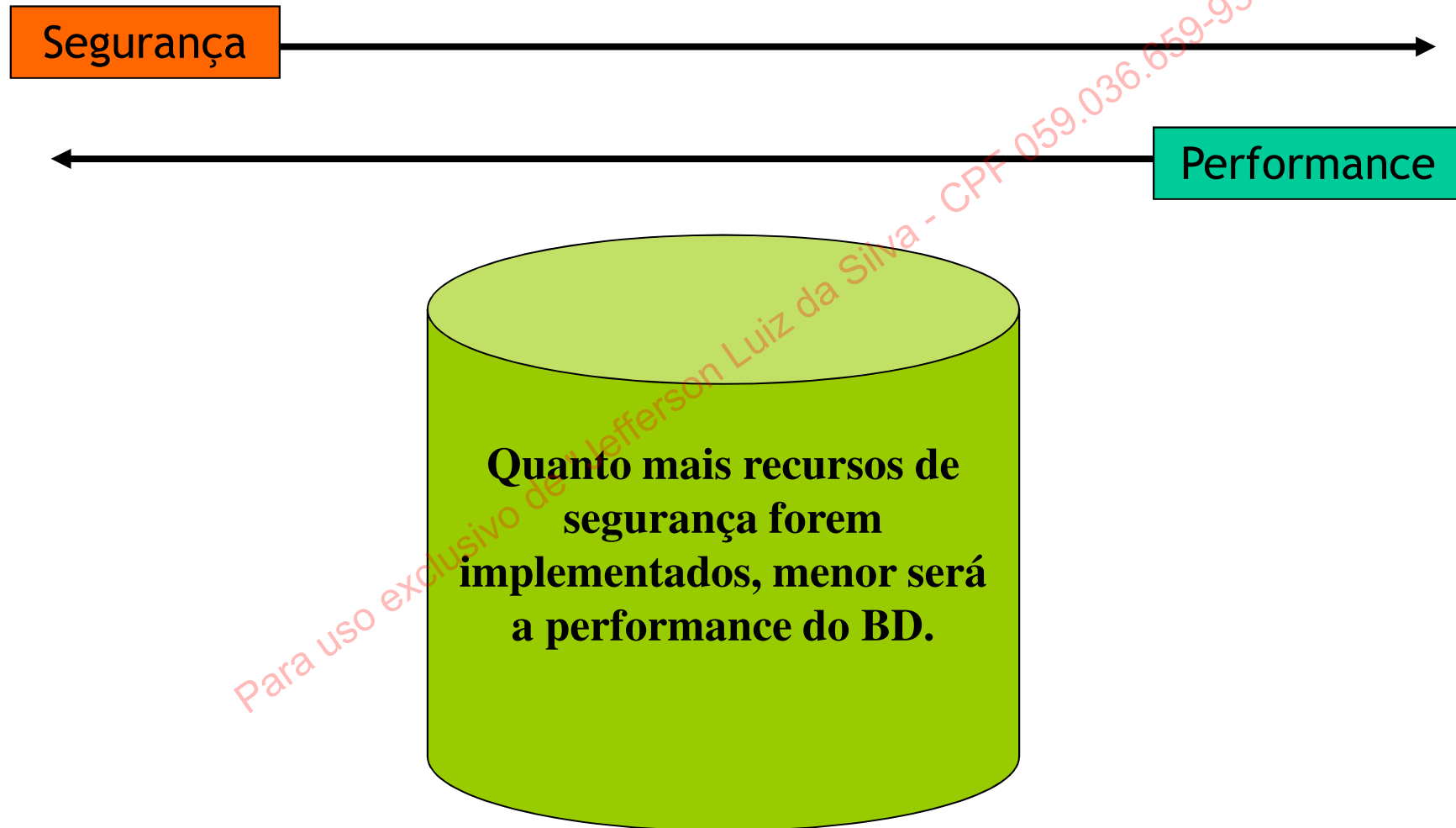

```
SELECT COUNT(*) FROM OE.ORDER_ITEMS;
```

PLAN_TABLE_OUTPUT

Plan hash value: 3419397814

Id	Operation	Name	Rows	Cost	(%CPU)	Time
0	SELECT STATEMENT		1	3	(0)	00:00:01
1	SORT AGGREGATE		1			
2	TABLE ACCESS FULL	ORDER_ITEMS	665	3	(0)	00:00:01

Segurança X Performance



Segurança X Performance

- **TDE** degrada performance de instruções SQL;
- **LOGGING** aumenta tempo de gravação;
- **CONSTRAINTS** aumentam tempo de gravação;
- **LOCKS** ocasionam espera e aumentam tempo de gravação;
- **COMMITs** frequentes aumentam quantidade de gravações;
- Auditoria aumenta consumo de CPU e I/O.

SQL ruim é a causa nº 1 de problemas de performance

Oracle Essentials - Oracle Database 11g

Para uso exclusivo de Jefferson Luiz da Silva - CPF 059.036.659-93"

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 2

Identificando problemas de performance

Para uso exclusivo de "Jerson Luiz da Silva" - CPF 059.036.659-93"

Introdução

- Principais conceitos apresentados neste capítulo:
 - Problemas mais comuns;
 - Métodos tradicionais ou de *verificação de sanidade* para identificar problemas de performance;
 - Prioridades na resolução de problemas;
 - Como descobrir/resolver problemas de performance.

Problemas mais comuns

- Gerenciamento de conexão ruim:

Aplicações conectam e se desconectam rapidamente no BD.

- ➔ • Mau uso de cursores e da shared pool:

Aplicações não usam cursores compartilhados ou variáveis bind..

- SQL ruim:

As instruções SQL consomem muitos recursos do sistema.

- ➔ • Parâmetros de inicialização c/ valores <> padrão:

Parâmetros de inicialização c/ valores configurados inapropriadamente.

- ➔ • I/O ruim:

O armazenamento não está bem distribuído ou largura de banda I/O insuficiente.

Problemas mais comuns

- ➔ **Configuração de Redo Logs:**
Redo logs pequenos ou poucos grupos de redo logs.
- ➔ **Serialização de blocos de dados:**
Ocorre quando faltam listas livres, grupos de listas livres, slots de transação ou segmentos de rollback.
- ➔ **Full Table Scans longos:**
FTS em operações online interativas ou de alto volume, normalmente consomem muito I/O lógico e físico.
- ➔ **Grande quantidade de SQL recursivo (SYS):**
Indicam atividades muito altas de gerenciamento de espaço de armazenamento, tais como alocação de extents.
- **Erros de migração e implantação:**
Aplicações usam muitos recursos do sistema porque o schema contendo as tabelas não foi migrado com sucesso, por exemplo, de um ambiente de desenvolvimento para o ambiente de produção.

Identificando problemas de performance

- Métodos proativos:

- Verifique uso de memória, I/O e CPU;
- Verifique instruções SQL que consomem muitos recursos;
- Verifique configurações inapropriadas de parâmetros de instância;
- Verifique configurações do Banco de Dados.

- Métodos reativos:

- Acrescente na lista anterior:
 - Ouvir *feedback* de usuários;
 - Verificar erros em *alert logs* e *traces*.

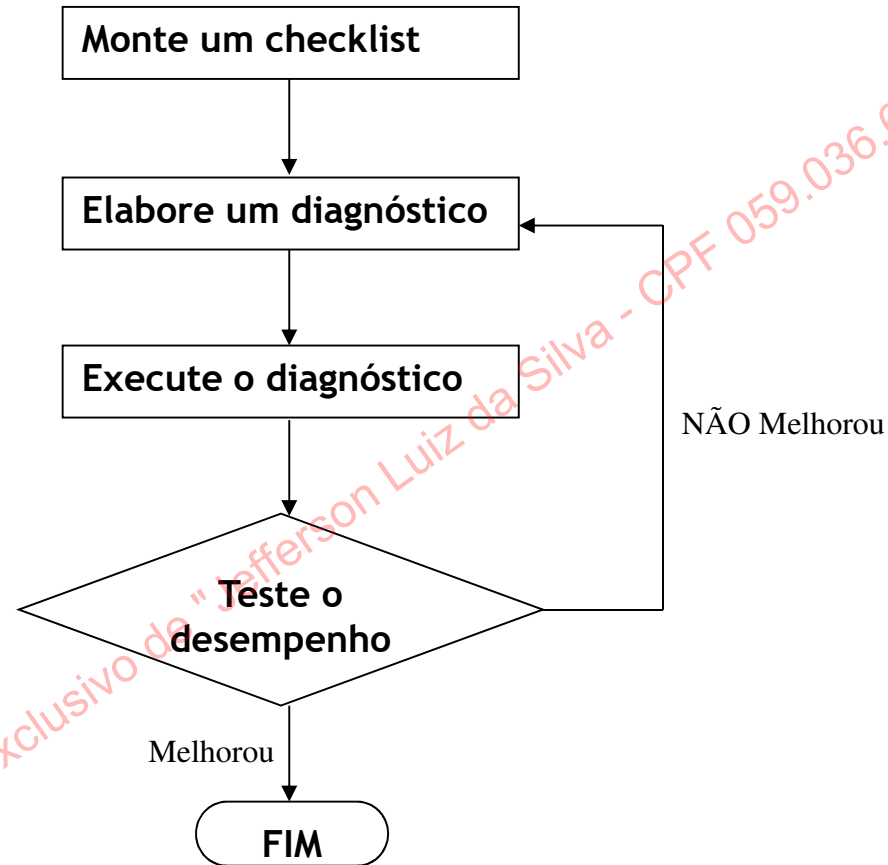
Metodologia de tuning

- **Metodologia de tuning proposta pela Oracle:**
 - Otimize uma aplicação, de cima para baixo:
 - primeiro o **projeto** da aplicação (modelagem);
 - depois o **código** da aplicação (SQL);
 - e por fim, a **instância**.
 - Otimize os itens que apresentam maior potencial de benefício:
 - SQL consumindo muitos recursos;
 - Maiores tempos de espera;
 - Maiores tempos de serviço.
 - Pare quando o objetivo for alcançado.

Prioridades

- Resolva primeiro os problemas que causam maior impacto:
 - Analise **tempo de execução**, **tempo de CPU** e **tempo de espera**;
 - Determine qual componente (memória, rede, cpu, I/O etc) consome a maior quantidade de tempo;
 - Eventos com alto tempo de espera normalmente são aqueles que conseguimos obter maiores ganhos no trabalho de tuning, portanto, eles devem ser analisados primeiro.

Como descobrir/resolver o problema?



Diagnóstico: nome dado a um processo analítico que é utilizado para chegar a alguma conclusão e à conclusão, por si própria

Como descobrir/resolver o problema?

1- Monte um checklist para identificar o(s) problema(s), incluindo uma verificação de sanidade (sanity check):

- **Verificação de uso de recursos computacionais:** memória, disco, rede e cpu;
- **Verificação de erros ou mudanças no BD através do alert.log;**
- Verificação da SGA e PGA;
- Verificação de coleta de estatísticas;
- Verificação de parâmetros da instância;
- Verificação de locks;
- Verificação de instruções SQL que consomem mais recursos;
- Verificação de cache hit ratio.

Como descobrir/resolver o problema?

- 2- **Elabore um diagnóstico**, baseado no que foi analisado no item anterior;
- 3- **Execute o diagnóstico**;
- 4- **Teste e verifique se o desempenho melhorou**. Se melhorou, ótimo, trabalho encerrado! Caso contrário, volte para o item 2 e repita os passos.

Exercícios

1. Execute o Swingbench por 15 minutos com 30 usuários, e **durante** a sua execução, rode os scripts deste capítulo;
2. Identifique e explique no mínimo 3 problemas encontrados.

**“Um problema bem definido
é um problema já resolvido
pela metade”**

John Dewey (Expert Oracle Practices)

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 3

Identificando problemas de performance através de **WAIT EVENTS**

Para uso exclusivo de "Jefferson Lúcia Silva - CPF 059.036.659-93"

Introdução

- Principais conceitos apresentados neste capítulo:
 - O que são Wait Events?;
 - Wait Events por versão do Oracle;
 - Classes de Wait Events;
 - Wait Events e sua relação com o tuning de performance;
 - Ferramentas para analisar Wait Events;
 - Wait Events mais comuns;
 - Tempos de espera aceitáveis.

O que são Oracle Wait Events?

- São uma coleção de eventos que fornecem informações sobre quais recursos computacionais uma sessão ou processo do BD Oracle, está **em estado de espera**;
- Recursos computacionais que podem ser disputados:
 - Físicos:
 - Armazenamento, CPU, memória, rede etc.
 - Lógicos:
 - Latches, locks etc.

Características gerais dos WE

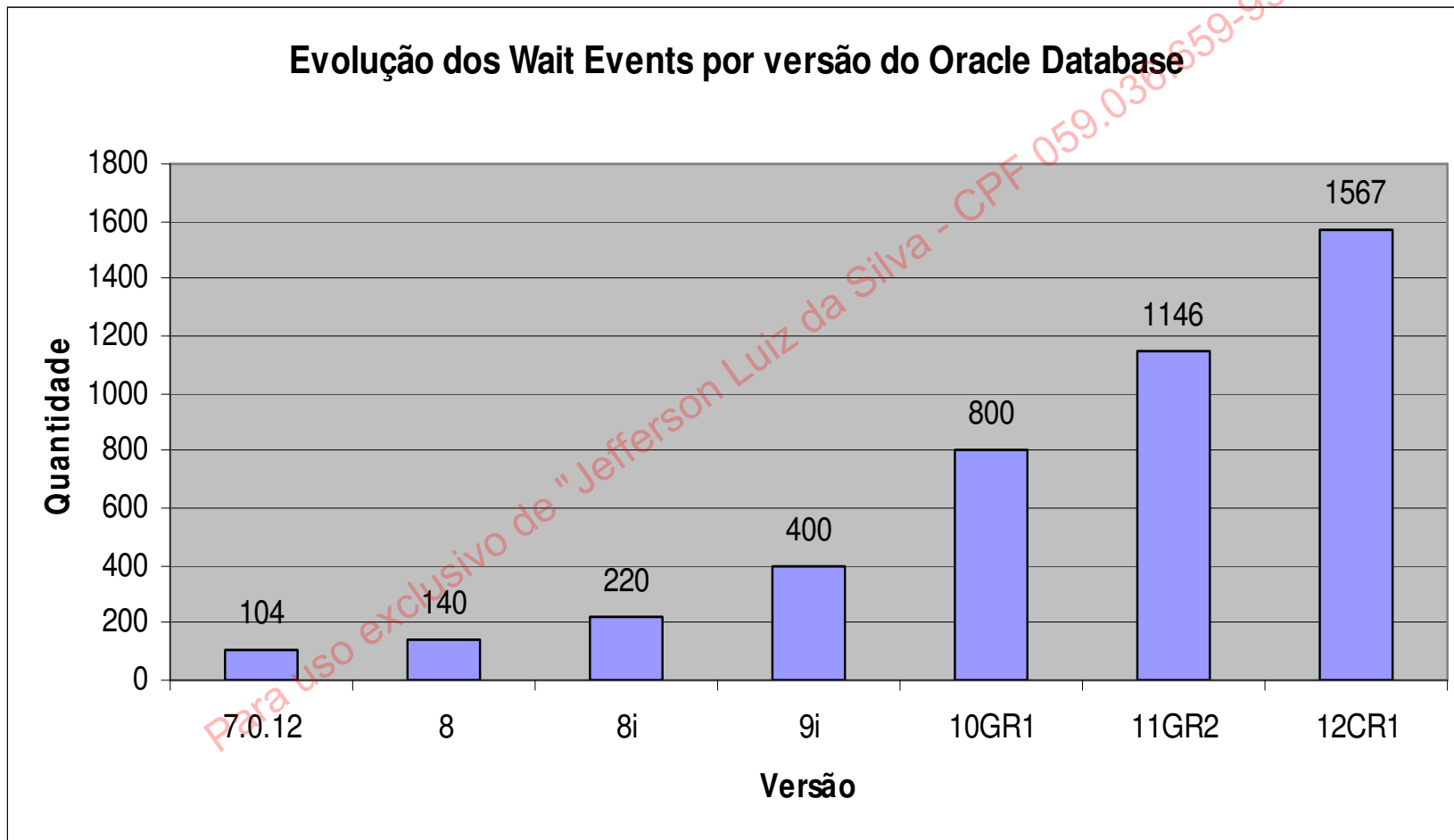
- Normalmente revelam **sintomas** de problemas de performance no BD (ex.: *contenção de buffer* ou *contenção de I/O*);
- Não revelam a **causa** do problema;
- São implementados na *Oracle Wait Interface* (OWI), que nasceu na versão 7.0.12 do BD Oracle;
- Podem ser consultados na visão **V\$EVENT_NAME**.

Evolução dos Oracle Wait Events

Versão SGBD	Quantidade Wait Events
Oracle 7.0.12	104
Oracle 8	140
Oracle 8i	220
Oracle 9i	400
Oracle 10GR1	+ de 800
Oracle 11GR2	1146
Oracle 12cR2	1650

Qtde. Oracle Wait Events X Versão SGBD

Evolução dos Oracle WE



Tuning X Wait Events

- Performance pode ser analisada através do tempo de resposta ou taxa de transferência de dados (*throughput*);
- No BD, o melhor indicador é o tempo de resposta. Quanto menor o tempo de resposta, melhor a performance. O *throughput* normalmente é calculado através de ferramentas executadas no SO (Ex.: iostat);
- Tempo de resposta é igual ao *tempo de serviço* + *tempo de espera*:
$$\text{TEMPO DE RESPOSTA} = \text{TEMPO DE SERVIÇO} + \text{TEMPO DE ESPERA}$$
- Tempos de espera muito altos normalmente são sintomas de problemas de performance e dão pistas para nos ajudar a montar um bom **DIAGNÓSTICO** do problema.

Classificação e Wait Events mais comuns

Para uso exclusivo de "Jefferson Luiz de Silva - CPF 059.036.659-93"

Classes de Wait Events

- Os Wait Events são agrupados em classes. A partir do 11GR2 existem 13 classes (consultar [V\\$EVENT_NAME](#)):
 - *Administrative*:
 - ✓ *switch logfile command, control file backup creation*
 - *Application*;
 - ✓ *enq: TX - row lock contention, enq: TM - contention*
 - *Cluster*;
 - ✓ *gc current request, gc buffer busy acquire*
 - *Commit*;
 - ✓ *log file sync, enq: BB - 2PC across RAC instances*
 - *Concurrency*;
 - ✓ *cursor: pin S, cursor: pin S wait on X*
 - *Configuration*;
 - ✓ *log buffer space, latch: redo writing*

Classes de Wait Events

- Idle:
 - ✓ *SQL*Net message from client, PX Deq Credit: send blkd*
- Network:
 - ✓ *SQL*Net more data to client, SQL*Net more data to dblink*
- Other:
 - ✓ *AWR Metric Capture, Sync ASM rebalance*
- Queueing:
 - ✓ *Streams: flow control, LogMiner reader: buffer*
- Scheduler:
 - ✓ *resmgr:cpu quantum, resmgr:pq queued*
- System I/O:
 - ✓ *log file sequential read, control file sequential read*
- User I/O
 - ✓ *db file sequential read, db file scattered read*

Wait Events mais comuns

- Um estudo da Confio Software, realizado em mais de 100 clientes, revela os seguintes números:
 - 2/3 de todos os wait events são relacionados à **I/O**;
 - Wait events relacionados a requisição de **cache** são o 2º maior problema geral de performance em ambientes clusterizados;
 - Os top 15 wait events que serão apresentados a seguir são responsáveis por **90% de todos** os wait events coletados.

Wait Events mais comuns

1- db file sequential read (User I/O):

— Explicação:

- Espera de I/O, durante a leitura de 1 bloco ou leitura sequencial de blocos (normalmente índices).

— Possível(is) causa(s):

- Buffer Cache insuficiente;
- Lentidão de I/O;
- Leitura desnecessárias;
- Index range scan em muitos blocos ou blocos distantes.

— Recomendação:

- Identificar os tablespaces, datafiles e índices que estão consumindo muitos recursos de I/O.

Wait Events mais comuns

2- db file scattered read (User I/O):

— Explicação:

- Espera de I/O durante operações de FTS, Fast full index scan (múltiplos blocos).

— Possível(is) causa(s):

- Buffer Cache insuficiente;
- Lentidão de I/O;
- FTS ou FFIS desnecessários.

— Recomendação:

- Identificar os tablespaces, datafiles e tabelas que estão consumindo muitos recursos de I/O.

Wait Events mais comuns

3- direct path read / direct path read temp (User I/O):

— Explicação:

- Sessões de usuário estão lendo muitos dados diretamente da PGA.

— Possível(is) causa(s):

- PGA insuficiente ou sorts muito grandes;
- Paralelismo excessivo ou SQLs pesados;
- I/O assíncrono habilitado aguardando retorno completo dos dados;
- A partir do 11G, FTS pode fazer leitura direta em tabelas grandes (considerando tamanho da tabela X tamanho da buffer cache e várias outras estatísticas).

— Recomendação:

- Identificar os tablespaces, datafiles e objetos que estão consumindo muitos recursos de I/O.

Wait Events mais comuns

4- global cache cr request (Cluster):

— Explicação:

- Tempo de espera para acessar dados de um cache remoto (global) em instâncias de um cluster (Oracle RAC).

— Possível(is) causa(s):

- Contenção de blocos;
- **Interconnect insuficiente;**
- Buffer cache pequena;
- Queries estão carregando muitos dados em memória;

No 11GR2, o novo algoritmo "bypass reader" pode diminuir a qtd de ocorrências deste wait event.

Wait Events mais comuns

5- buffer busy waits / read by other session

(Concurrency):

— Explicação:

- Mais de uma sessão está simultaneamente tentando acessar os mesmos blocos de dados na Buffer Cache;

— Possível(is) causa(s):

- Hot blocks;
- SQL ruim;
- Queries carregando muitos dados em memória.

Obs.: read by other session existe no Oracle v. 10.1 ou superior

Wait Events mais comuns

6- SQL*Net more data from dblink (User I/O):

— Explicação:

- Espera durante comunicação via rede através de um Database Link (dblink).

— Possível(is) causa(s):

- Muitos dados trafegando pela rede via dblink;
- Largura de banda de rede insuficiente;
- Alta latência na rede.

Wait Events mais comuns

7- log file sync (Commit):

— Explicação:

- Espera de I/O pelo processo LGWR para escrever dados nos Redo Log Files.

— Possível(is) causa(s):

- Muitos commits ocorrendo;
- Lentidão de I/O;
- Log Buffer muito grande.

Wait Events mais comuns

8- direct path write / direct path write temp

(User I/O):

— Explicação:

- Sessões de usuário estão escrevendo (fisicamente) muitos dados.

— Possível(is) causa(s):

- Operações de carga direta;
- Lentidão de I/O;
- Sorts muito grandes ou paralelismo excessivo podem estar ocorrendo;
- Muitas escritas em LOBs não carregados na buffer cache (padrão);
- PGA pode estar pequena.

Wait Events mais comuns

9- library cache lock (Concurrency):

— Explicação:

- Concorrência entre sessões acessando objetos na library cache para adquirir um bloqueio. Pode ocorrer ao localizar o objeto.

— Possível(is) causa(s):

- Parses ou recompilações longas de instruções SQL ou PL/SQL sendo executados;
- Redimensionamento frequente da SGA quando ASMM ou AMM está habilitado;
- Instruções SQL que não utilizam variáveis bind.

Wait Events mais comuns

10- SQL*Net more data to client (Network):

— Explicação:

- Espera durante comunicação via rede, para enviar dados ao(s) cliente(s).

— Possível(is) causa(s):

- Muitos dados trafegando pela rede;
- Largura de banda de rede insuficiente;
- Alta latência na rede.

Wait Events mais comuns

11- db file parallel read / db file parallel write

(User I/O):

— Explicação:

- Espera de I/O durante a leitura/escrita de blocos, durante a execução de instruções paralelizadas.

— Possível(is) causa(s):

- Buffer Cache insuficiente;
- Lentidão de I/O;
- Leituras desnecessárias.

Wait Events mais comuns

12- library cache pin ou Cursor Pin S wait on X

(Concurrency):

— Explicação:

- Espera por objeto bloqueado na Library Cache (modos de bloqueios incompatíveis).

— Possível(is) causa(s):

- Parses ou recompilações de longas de instruções SQL ou PL/SQL sendo executadas;
- Shared Pool pequena.

- Obs.: “Cursor Pin S wait on X” existe no Oracle v. 10.2 ou superior

Wait Events mais comuns

13- log buffer space (Configuration):

— Explicação:

- Espera para gravar dados na log buffer.

— Possível(is) causa(s):

- Log Buffer insuficiente;
- Lentidão de I/O ao gravar Redo Log Files.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Wait Events mais comuns

14- enq: TX - row lock contention (Application):

— Explicação:

- Muitas transações tentando atualizar mesma(s) linha(s).

— Possível(is) causa(s):

- Transações longas;
- Concorrência muito alta;
- Lentidão de I/O.

Wait Events mais comuns

15- cache buffers chains (Concurrency):

— Explicação:

- Múltiplos processos estão tentando acessar os mesmos blocos na buffer cache;
- Excessiva substituição de blocos.

— Possível(is) causa(s):

- SQLs repetidos acessando constantemente mesmos blocos, SQLs ineficientes;
- Buffer cache pequena.

Ferramentas para analisar problemas e Wait Events

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Ferramentas

- Ferramentas para analisar *Wait Events*, e *problemas de performance*, em geral:

- Visões de Performance Dinâmicas:

- Não necessitam de licença adicional;
- Não possuem dados históricos;
- Dão bastante trabalho para analisar O TODO.

- Extended SQL Trace (Event 10046):

- Não necessita de licença adicional;
- Pode afetar desempenho do BD;
- Mostram wait events a partir da versão 9iR1, mas possuem suporte oficial somente a partir da versão 10GR2;
- Alta granularidade (maior detalhamento).

Ferramentas

— Statspack:

- Não necessita de licença adicional;
- Gerenciamento manual;
- Baixa granularidade;
- Possui apenas dados históricos.

— AWR:

- **Necessita de licença adicional (*Diagnostic Pack*);**
- Gerenciamento automático;
- Baixa granularidade;
- Possui apenas dados históricos.

Visões de performance dinâmicas

- Apresentam dados de estatísticas cumulativas da *instância*, *sessões* ou *instruções SQL individuais* e até mesmo de segmentos e serviços;
- São **resetadas em cada startup do BD** e são subdivididas em:
 - **Wait Event Statistics:**
 - ✓ Estatísticas de sessão catalogadas na visão `V$SESSION_EVENT`;
 - ✓ `V$SYSTEM_EVENT` apresenta total de espera por eventos de **TODAS** as sessões.
 - **Systemwide Statistics:**
 - ✓ Estatísticas globais de sistema catalogadas na visão `V$STATNAME`;
 - ✓ `V$SYSSTAT` apresenta total de eventos de sistema.

Versão/release	Qtde. visões perf. dinâmicas
10GR2	398
11GR1	484
11GR2	536
12CR1	645

Evolução das Visões de Performance Dinâmicas por versão do Oracle Database

Visões de performance dinâmicas

— V\$SESSION:

- Apresenta dados das sessões atuais, incluindo o atual ou último evento de espera;

— V\$SESSION_WAIT:

- Apresenta o atual ou último evento de espera das sessões ativas;
- Principal visão da OWI.

— V\$SESSION_EVENT:

- Apresenta de forma acumulada os eventos de espera;
- Versão mais completa e detalhada da V\$SESSION_WAIT.

— V\$SYSTEM_EVENT:

- Apresenta de forma acumulada as estatísticas de todos os eventos de espera de uma instância (desde o seu startup).

Trace estendido

- Também é conhecido como **Event 10046 Trace**;
- Possui vários níveis de informações:
 - 1 = Standard Trace
 - 4 = Level 1 + Bind Variables
 - 8 = Level 1 + Wait Events
 - 12 = Levels 1 + 4 + 8
- É possível ativá-lo em nível de sessão, instância ou sessões de outros usuários;
- Informações importantes do trace:
 - **cr** Consistent reads.
 - **pr** Physical reads.
 - **pw** Physical writes.
 - **time** Elapsed time taken by the operation (in microseconds).
 - **cost** Optimizer cost of the operation.
 - **size** Estimated size (in bytes).
 - **card** Estimated Optimizer Cardinality.

Considerações finais

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Importante

- Muitas vezes, *wait events* não representam problemas:
 - Um valor alto de *virtual circuit wait*, por exemplo, pode indicar sessões compartilhadas ociosas (em espera);
 - *db file sequential read*, *db file parallel read* e *db file scattered read* estarão presentes mesmo em sistemas saudáveis (ambientes c/ BD's bem tunados).

Importante

- Fique preocupado quando no período de 1 hora os *wait events* (em geral aqueles relacionados a leituras/escritas lógicas e físicas) apresentarem:
 - Tempo de espera médio muito maior que 20 ms por evento;
 - Quantidade de ocorrências do evento for maior que 1000.

Tempos de espera típicos aceitáveis

Wait Event	R/W	Synchronous / Asynchronous	Singleblock/ Multiblock	Elapsed Time (with 1000+ waits per hour)
control file parallel write	Write	Asynchronous	Multi	< 15ms
control file sequential read	Read	Synchronous	Single	< 20 ms
db file parallel read	Read	Asynchronous	Multi	< 20 ms
db file scattered read	Read	Synchronous	Multi	< 20 ms
db file sequential read	Read	Synchronous	Single	< 20 ms
direct path read	Read	Asynchronous	Multi	< 20 ms
direct path read temp	Read	Asynchronous	Multi	< 20 ms
direct path write	Write	Asynchronous	Multi	< 15 ms
direct path write temp	Write	Asynchronous	Multi	< 15 ms
log file parallel write	Write	Asynchronous	Multi	< 15 ms

Fonte: “MOS: [How to Tell if the IO of the Database is Slow \[ID 1275596.1\]](#)”

Exercícios

Faça os exercícios do capítulo (ver slides do instrutor).

BOA SORTE!

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-00"

**80% das consequências
advêm de 20% das causas**

Conceito da Lei de Pareto
Vilfredo Pareto (1848 - 1923)

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 4

Utilizando o Statspack, AWR e ADDM

Para uso exclusivo de "Jefferson Luiz da Silva" - CPF 059.036.659-93"

Introdução

- Principais conceitos apresentados neste capítulo:
 - O que é o Statspack;
 - O que é o AWR, como configurá-lo e utilizá-lo;
 - Diferenças entre Statspack e AWR;
 - O que é o ADDM e como utilizá-lo;
 - O que são Advisors e sua relação com o ADDM.

Statspack

- É um utilitário composto de um conjunto de scripts SQL, PL/SQL e SQL Plus, que permitem coletar, armazenar e ver dados de performance do BD;
- Requer o uso dos seguintes scripts:
 - `$ORACLE_HOME/rdbms/admin/spcreate.sql`:
 - Instalar o Statspack.
 - `$ORACLE_HOME/rdbms/admin/spauto.sql`:
 - Criar job que executa coleta de snapshots de hora em hora.
 - `$ORACLE_HOME/rdbms/admin/spreport.sql`:
 - Gerar relatório com dados coletados.
- Os dados ou estatísticas de performance de cada coleta, também conhecidos como **snapshots**, ficam armazenados em objetos de um schema chamado **PERFSTAT**;
- Para gerar snapshots manualmente, execute no SQL Plus:
`EXECUTE STATSPACK.SNAP;`

Statspack

- Cada snapshot possui um número. Para analisar a performance de um BD devemos fornecer como entrada o valor de 2 snapshots;
- **Não necessita de licença adicional;**
- Não possui gerenciamento automático. É necessário criar rotina para gerenciar (reter ou eliminar) dados do schema PERFSTAT;
- Possui apenas dados históricos;
- Sua última versão contém estatísticas de *Oracle Streams*;
- **Não contém métricas ou seções de estatísticas que foram introduzidas a partir da versão 10G;**

Statspack

- Não possui acesso através do *Enterprise Manager*;
- Para mais informações leia o documento `$ORACLE_HOME/rdbms/admin/spdoc.txt`.

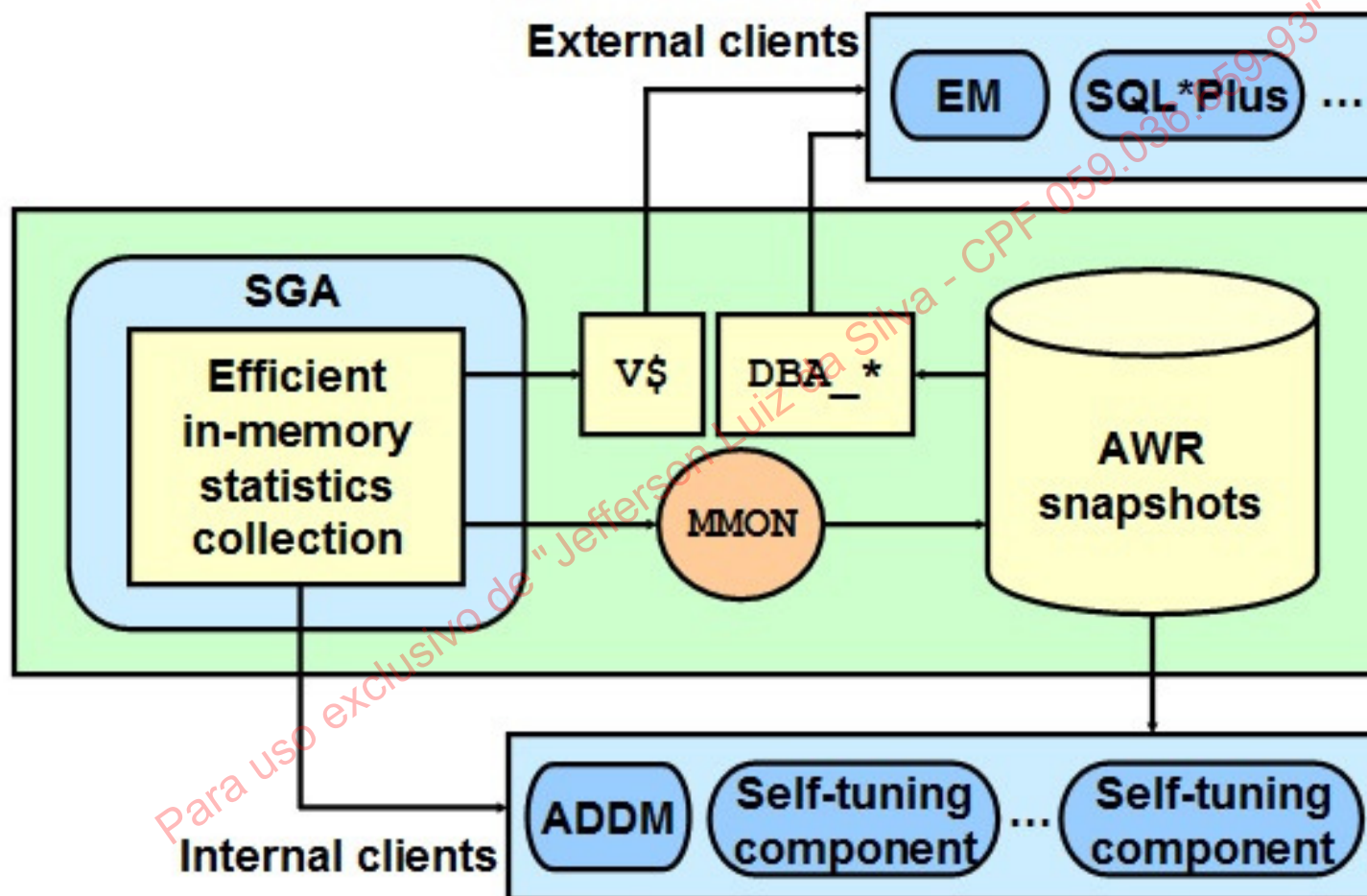
Visão geral AWR

- O Automatic Workload Repository (AWR) é um repositório que contém os seguintes tipos de informações:
 - Estatísticas de acesso e uso de objetos;
 - Estatísticas Time model (qtde. tempo de BD associados a processos);
 - Estatísticas de sistema;
 - Estatísticas de sessão (via Active Session History);
 - Instruções SQL.
- As informações de estatísticas são de propriedade do **SYS** e são armazenados no tablespace **SYSAUX**;
- Existe no Oracle Database a partir da versão 10G e já vem instalado e habilitado por padrão;
- Para usar o AWR é necessário:
 - Ter a licença da Option **Oracle Diagnostics Pack**;
 - Configurar parâmetro da instância **CONTROL_MANAGEMENT_PACK_ACCESS** com valor **DIAGNOSTIC+TUNING** (default) ou somente **DIAGNOSTIC**.

Visão geral AWR

- É uma evolução do Statspack, portanto, possui os mesmos conceitos de utilização;
- Utiliza snapshots e os relatórios são gerados sempre a partir de 2 snapshots, que podem envolver um pequeno ou grande período sem interrupções (shutdown);
- Fornece informações para o ADDM gerar diagnósticos automáticos;
- No 12c, o AWR, ADDM e ASH Reports podem conter todos os PDBs existentes em um CDB e o consumo de recursos é identificado por PDB.

Arquitetura do AWR



Fonte: Oracle Corporation

Diferenças Statspack X AWR

STATSPACK	AWR
Free	Precisa de licença (<i>Diagnostics Pack</i>)
Única opção até Oracle 9	Existe a partir do 10G
Precisa ser instalado	Vem instalado por padrão
Gerenciamento manual	Gerenciamento automático
Não contém estatísticas de WE adicionados no 10G ou superior	Não contém estatísticas para <i>Oracle Streams</i>
Não possui componente para gerar estatísticas de sessão em tempo real	Possui componente para gerar estatísticas de sessão em tempo real (<i>ASH</i>)
Não possui componente para gerar diagnóstico	Possui componente para gerar diagnóstico (<i>ADDM</i>)

Configurando o AWR

- Snapshots são coletados a cada **60 minutos**, mas é possível mudar o intervalo de coleta através do EM ou da procedure:
`DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS`
- Para coletar snapshots manualmente, execute a procedure:
`DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT`
- A retenção dos snapshots é configurada por padrão com o valor de **8 dias**. Snapshots mais velhos que o período de retenção são expurgados automaticamente;
- É possível mudar o intervalo dos snapshots e o período de retenção deles, através do EM ou da procedure:
`DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS`

Analizando o AWR

- Os relatórios de performance, conhecidos como AWR Reports, podem ser gerados via EM ou script `$ORACLE_HOME/rdbms/admin/awrrpt.sql`, no formato *texto* ou *html*;
- É possível comparar 2 relatórios de performance executando o script `$ORACLE_HOME/rdbms/admin/awrddrpt.sql`;
- É possível também, ver estatísticas de uma única instrução SQL, através do script `$ORACLE_HOME/rdbms/admin/awrsqrpt.sql`;

Analizando o AWR

- Para ambientes com RAC existem os seguintes scripts:
 - `$ORACLE_HOME/rdbms/admin/awrgrpt.sql`
 - `$ORACLE_HOME/rdbms/admin/awrgdrpt.sql`
- Para facilitar a análise de relatórios e identificar o que está fora do padrão, crie **baselines fixos** ou **flutuantes** que representam o processamento normal do BD.

Analizando o AWR

- Sugestões de baseline:

- Atividade normal;
- Banco de Dados ocioso;
- Horário de pico;
- Processamento batch;
- Processamento do ciclo de negócios ou final do mês.

Para uso exclusivo de "Jeferson Luiz da Silva - CPF 059.036.659-93"

AWR Baselines

- **Baselines fixos:**

- Podem ser criados a partir de snapshots existentes ou com base em um período de retenção;
- Pode-se criar automaticamente baselines fixos para um intervalo repetido futuro, através de **baseline templates**.

- **Baselines flutuantes:**

- São criados com base em períodos flutuantes, ou seja, períodos que se movem e que são renovados;
- O período flutuante não pode ser maior que o período de retenção dos snapshots;
- Um baseline de período flutuante já vem habilitado no BD com o nome **SYSTEM_MOVING_BASELINE**.

AWR Reports

- Uma das tarefas mais importantes e complexas ao usar o AWR é analisar o **AWR Report**;
- As informações coletadas são agrupadas e formatadas pelas seguintes categorias:
 - Instance efficiency
 - **Top 5 timed events**
 - Memory and CPU statistics
 - Wait information
 - **SQL statement information**
 - Miscellaneous operating system and database statistics
 - Database file and tablespace usage information

AWR Reports

☆ How to Use AWR Reports to Diagnose Database Performance Issues (Doc ID 1359094.1)

Modified: 28-Mar-2014 Type: HOWTO

In this Document

[Goal](#)

[Best Practices](#)

[Pro-Active Problem Avoidance and Diagnostic Collection](#)

[Ask Questions, Get Help, And Share Your Experiences With This Article](#)

• Instance Efficiency

Again, instance efficiency stats are more use for general tuning as opposed to addressing specific issues (unless waits point at these).

```
Instance Efficiency Percentages (Target 100%)
=====
      Buffer Nowait %:    99.91      Redo NoWait %:    100.00
      Buffer Hit   %:    98.14      In-memory Sort %:    99.98
      Library Hit  %:    99.91      Soft Parse %:       94.48
      Execute to Parse %: 99.45      Latch Hit %:        99.97
      Parse CPU to Parse Elapsed %: 71.23      % Non-Parse CPU:    99.00
```

The most important Statistic presented here from the point of view of our example is the '% Non-Parse CPU' because this indicates that almost all SQL may help to improve this.

If we were tuning, then 94.48% soft parse rate would show a small proportion of hard parsing which is desirable. The high execute to parse may not be relevant dependent on the application. For example, in a data warehouse environment, hard parsing may be higher due to usage important.

AWR Reports

- Os 3 primeiros lugares que devemos olhar no AWR Report são:

1- DB Time:

- Se $DB\ Time = Elapsed\ Time * CPU$ então BD sobrecarregado;
- Carga do BD = $DB\ Time / Elapsed\ Time$ -> média de sessões ativas

	Snap Id	Snap Time	Sessions	Curs/Sess
Begin Snap:	3753	23-Out-12 14:00:33	383	11.5
End Snap:	3754	23-Out-12 15:00:38	407	11.8
Elapsed:		60.08 (mins)		
DB Time:		849.87 (mins)		

AWR Reports

2- Instance Efficiency:

- Em geral, os percentuais devem ser maiores que 90% ou serem o mais próximo de 100% possível;
- Parse CPU to Parse Elapsed deve ter o menor valor possível (bom se for menor que 5%), principalmente se % Non-Parse CPU estiver baixo;
- In-Memory Sort baixo é um indício de que deve-se aumentar a PGA.

```
Instance Efficiency Percentages (Target 100%)
~~~~~
      Buffer Nowait %: 100.00      Redo NoWait %: 100.00
      Buffer Hit   %: 99.98      In-memory Sort %: 100.00
      Library Hit  %: 96.41      Soft Parse %: 77.71
      Execute to Parse %: 91.58      Latch Hit %: 97.17
      Parse CPU to Parse Elapsed %: 8.11      % Non-Parse CPU: 99.64
```

AWR Reports

3- Top 5 Timed Foreground Events:

- Ver maiores Wait Events e investigar a causa de seus valores serem os mais altos (o que nem sempre indica problemas de performance);
- Em um BD performático, normalmente os eventos relacionados ao uso de CPU e I/O estarão no topo.

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
virtual circuit wait	163,744	35,243	215	69.1	Network
DB CPU		12,820		25.1	
SQL*Net message from dblink	17,282	1,470	85	2.9	Network
db file sequential read	104,829	351	3	.7	User I/O
eng: TX - row lock contention	55	99	1805	.2	Applicatio

AWR Reports

- Outras seções importantes do AWR Report:

- Shared Pool Statistics:

- Se o valor de **Memory Usage** estiver próximo de 100%, é um indício de que a *Shared Pool* está pequena;
 - O valor **% SQL with executions>1** deve ser alto, pois indica que poucos *hard parses* estão ocorrendo.

Shared Pool Statistics	Begin	End
	-----	-----
Memory Usage %:	66.04	96.68
% SQL with executions>1:	80.71	50.04
% Memory for SQL w/exec>1:	78.39	47.44

AWR Reports

—Load Profile:

- Valores de **Physical reads**, **Physical writes** e **Hard parses** devem ser baixos, ao compará-los com **Logical reads** e **Parses**;
- **Physical reads** e **Physical writes**, em sistemas OLTP, deve ter valor proporcional entre 10:1 e 5:1;
- Compare estes valores com o de um *Baseline* para identificar se algo incomum aconteceu no BD.

AWR Reports

- **Top SQL Statements (SQL ordered by Elapsed Time):**
 - Ver instruções SQL que tem maior tempo execução e descobrir, comparando com um baseline, se o tempo mudou (aumentou) e porque mudou.

```
SQL ordered by Elapsed Time                                DB/Inst: PRD/prd  Snaps: 3753-3754
-> Resources reported for PL/SQL code includes the resources used by all SQL
    statements called by the code.
-> % Total DB Time is the Elapsed Time of the SQL statement divided
    into the Total Database Time multiplied by 100
-> %Total - Elapsed Time as a percentage of Total DB time
-> %CPU    - CPU Time as a percentage of Elapsed Time
-> %IO     - User I/O Time as a percentage of Elapsed Time
-> Captured SQL account for 25.0% of Total DB Time (s):      50,992
-> Captured PL/SQL account for 24.3% of Total DB Time (s):    50,992
```

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id
10,682.7	140,807	0.08	20.9	99.9	.0	8wut0yg6ab455

```
SELECT 1 FROM SCHEMA.TABELA WHERE COLUNA = 123
```

AWR Reports

- Time Model Statistics:
 - Descobrir quem está consumindo maior tempo de BD: instruções SQL, hard/soft parse etc.;

```
. Time Model Statistics DB/Inst: ORCL/orcl Snaps: 330-336
. -> Total time in database user-calls (DB Time): 4588.7s
. -> Statistics including the word "background" measure background process
. time, and so do not contribute to the DB time statistic
. -> Ordered by % or DB time desc, Statistic name
.
. Statistic Name Time (s) % of DB Time
. -----
. sql execute elapsed time 4,358.6 95.0
. DB CPU 608.4 13.3
. parse time elapsed 18.8 .4
. hard parse elapsed time 15.7 .3
. PL/SQL execution elapsed time 13.6 .3
. connection management call elapsed time 8.3 .2
. hard parse (sharing criteria) elapsed time 2.5 .1
. hard parse (bind mismatch) elapsed time 2.1 .0
. PL/SQL compilation elapsed time 1.7 .0
. repeated bind elapsed time 0.0 .0
. sequence load elapsed time 0.0 .0
. DB time 4,588.7
. background elapsed time 629.5
. background cpu time 6.8
. -----
```

4588.7/60=76 minutes

95% of it for SQL execution

AWR Reports

- Operating System Statistics:

- Ver consumo de cpu (% do total pela instância).

```
. Instance CPU
. ~~~~~
.          % of total CPU for Instance:      8.2
.          % of busy CPU for Instance:      55.1
.          %DB time waiting for CPU - Resource Mgr:    0.0
.
. Operating System Statistics                      DB/Inst: ORCL/orcl  Snaps: 330-336
.
. Statistic                                     Value      End Value
. -----
. ..
. BUSY_TIME                                   111,572
. IDLE_TIME                                  640,189
. SYS_TIME                                   26,197
. USER_TIME                                  85,376
. ..
. NUM_CPUS                                   8
. NUM_CPU_CORES                              4
. NUM_CPU_SOCKETS                            1
. -----
```

AWR Reports

- **File IO Stats:**

- Descobrir onde I/O está lento. “Av Reads/s” > 20 ms indica I/O lento;
- Muitas escritas/leituras no TEMP podem indicar necessidade de criar mais tablespaces temporários.

File IO Stats DB/Inst: ORCL/orcl Snaps: 330-336
-> ordered by Tablespace, File

Tablespace	Filename	Av Reads	Av Reads/s	Av Rd(ms)	Av Blks/Rd	Av Writes	Av Writes/s	Av Buffer Waits	Av Buf Wt(ms)
SOE	C:\ORA\ORADATA\ORCL\SOE01.DBF	14,679	16	54.5	1.0	3,434	4	48	1.3
SOE	D:\ORA\ORADATA\ORCL\SOE02.DBF	13,689	15	184.6	1.0	2,010	2	31	0.0
SYSAUX	C:\ORA\ORADATA\ORCL\SYSAUX01.DBF	506	1	17.5	1.1	959	1	0	0.0
SYSTEM	C:\ORA\ORADATA\ORCL\SYSTEM01.DBF	812	1	10.0	1.0	21	0	8	0.0
TEMP	C:\ORA\ORADATA\ORCL\TEMP01.DBF	4	0	15.0	1.0	10	0	0	N/A
UNDOTBS1	C:\ORA\ORADATA\ORCL\UNDOTBS01.DBF	2	0	145.0	1.0	215	0	35	0.0
USERS	C:\ORA\ORADATA\ORCL\USERS01.DBF	2,145	2	69.8	1.0	3,452	4	0	0.0

AWR Reports

- **Commits:**

- Qtde. média de chamadas por usuário (user calls) / commits (user commits) e rollbacks (user rollbacks), por transação, deve ser preferencialmente maior que 30.

✓ Exemplo: $53.5 / (1 + 0) = 53.5 \rightarrow$ muito bom

Statistic	Total	per Second	per Trans
transaction tables consistent re	28	0.0	0.0
transaction tables consistent re	3,395	0.0	0.0
undo change vector size	1,306,969,712	2,173.9	1,894.5
user calls	36,904,812	61.4	53.5
user commits	689,127	1.2	1.0
user I/O wait time	4,848,107	8.1	7.0
user rollbacks	733	0.0	0.0

AWR Reports

- **Buffer Pool Advisory:**

- Verificar melhor tamanho possível para reduzir **Estimated Phys Reads**.

- **PGA Memory Advisory:**

- Verificar quando **Estd Extra Written to Disk** não diminui mais e **Estd PGA Overalloc Count** igual a 0 ou mais próximo de zero possível;
- Ver MOS doc ID 786554.1: How to Read PGA Memory Advisory Section in AWR and Statspack Reports to Tune PGA_AGGREGATE_TARGET.

- **Shared Memory Advisory:**

- Verificar melhor tamanho possível para reduzir **Load Time (s)**.

- **SGA Target Advisory:**

- Verificar melhor tamanho possível para reduzir **Est Physical Reads**.

AWR Reports

- Se top timed events são relacionados a muito I/O físico, examine:
 - SQL ordenados por leitura;
 - SQL ordered by Elapsed Time;
 - Tablespace I/O Stats;
 - File I/O Stats.
- Se top timed events são relacionados ao uso de CPU, examine:
 - Load Profile;
 - Time Model Statistics;
 - SQL ordered by CPU Time;
 - SQL ordered by Gets.

ASH Reports

- É possível também gerar relatórios que coletam informações do ASH (Active Session History), para ver o que está acontecendo nas sessões do BD, em tempo real, com precisão de até 1s;
- Para gerar ASH Reports, execute o script:
 - `$ORACLE_HOME/rdbms/admin/ashrpt.sql`
- Para ver rapidamente quais são os dados mais antigos do ASH, consulte a visão `DBA_HIST_ACTIVE_SESS_HISTORY`.

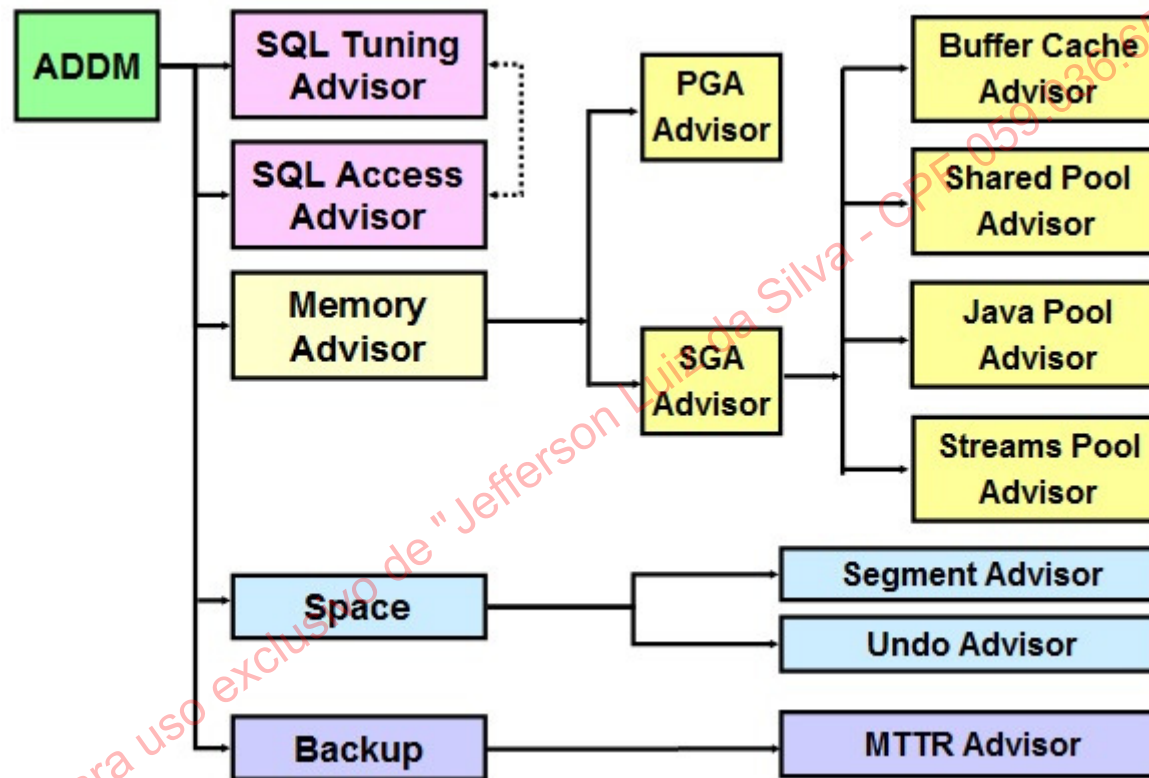
Visão Geral do ADDM

- O *Automatic Database Diagnostic Monitor* (ADDM) é um componente do BD que analisa informações do AWR e fornece **recomendações** de melhoria de performance;
- Sua função principal é monitorar o que está acontecendo no BD, identificar gargalos e sugerir recomendações para eliminá-los;
- A análise do ADDM é executada automaticamente depois da geração de cada snapshot do AWR;
- O diagnóstico do ADDM pode ser executado através dos seguintes caminhos:
 - Script `$ORACLE_HOME/rdbms/admin/addmrpt.sql`;
 - Package `DBMS_ADDM` (a partir do 11GR2);
 - Enterprise Manager.

Visão Geral do ADDM

- Suas informações também são armazenadas no repositório do AWR;
- O ADDM utiliza **Advisors** para realizar as suas atividades;
- Os *Advisors* são ferramentas poderosas para nos ajudar a identificar problemas no BD e dar sugestões (limitadas) de otimização.

ADDM X Advisors



Fonte: Oracle Corporation

Exercícios

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Exercícios

- 1- Gere um snapshot, execute o Swingbench por 5 minutos com 50 usuários e gere outro snapshot ao término desta execução;
- 2- Tire um relatório AWR considerando os snapshots gerados no exercício anterior;
- 3- Analise o relatório gerado no exercício anterior e anote as respostas das questões abaixo:
 - a) Em qual horário havia a maior sobrecarga no BD?
 - b) A SGA e PGA estão configuradas apropriadamente?
 - c) Existe algum problema de I/O nos tablespaces das aplicações?
 - d) A quantidade média de commits de usuário por transação está boa?
 - e) Qual os problemas principais do BD (aponte no mínimo 3)?

“O ótimo é inimigo do bom”

Voltaire (1694 - 1778)

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 5

Otimizando a instância do Banco de Dados

Para uso exclusivo de "Jeferson Luiz da Silva - CPF 059.036.659-93"

Introdução

- Principais conceitos apresentados neste capítulo:
 - Configuração da **SGA** e **PGA**;
 - Dicas diversas para tunar *Redo Log Buffer*, *Buffer Cache*, *Shared Pool*, instruções SQL e **I/O**, em geral.

Avalie cuidadosamente as dicas e técnicas apresentadas ao longo deste capítulo e preste atenção nos comentários do instrutor, pois dependendo das características do seu ambiente e aplicações, nem sempre você terá condições de utilizá-las ou benefícios ao utilizá-las.

Principais parâmetros

```
Verificacoes_instancia_parametros.sql

- shared_pool_reserved_size
  Deve ser configurado para armazenar dados de grandes requisicoes de armazenamento na shared pool. Usa-lo reduz fragmentacao
  na Shared Pool.
  Valor default Eh de 5% da shared pool.

- large_pool_size
  Permite configurar tamanho da large pool com gerenciamento manual de memoria ou tamanho minimo dela se estiver usando AMM ou
  ASMM.

- result_cache_max_result, result_cache_max_size, result_cache_mode
  Permite configurar cache de resultados para otimizar leituras. Dados ficam em uma area dentro da shared pool, ao invEhs de
  ficarem na buffer cache.
  Se configurados, devem ser muito bem gerenciados.
  Valores possiveis para result_cache_mode: force e manual (padrao)

- log_buffer
  Parametro que possui algoritmo interno para determinar tamanho do buffer de log. Se tamanho default for pequeno Eh possivel
  aumenta-lo.

- optimizer_mode
  Modo em que o otimizador trabalha para recuperar linhas. Valor ALL_ROWS favorece FTS. Valor FIRST_ROWS favorece IS.
```

Configurando a SGA x PGA

- Gerenciar e otimizar memória era uma das tarefas mais críticas de um DBA, até surgir no 10G o *Automatic Shared Memory Management* (ASMM) e no 11G o *Automatic Memory Management* (AMM);
- **AMM** (padrão no 11G) é bom para ambientes mistos: OLTP + batch jobs;
- **ASMM** isola SGA e PGA e permite usar *Huge Pages* (em SO Linux). Configuração indicada p/ ambientes c/ Data Guard e c/ armazenamento ASM.
- **Cuidado para não alocar muita memória e gerar paginação (Swap);**

Importante: Antes de alterar de um modo de gerenciamento para outro, faça uma cópia do SPFILE em PFILE.

Configure Redo Log Buffer

- Configure apropriadamente a **Redo Log Buffer** para reduzir I/O excessivo que pode ser gerado ao gravar muitos *redo log files* em um curto período de tempo;
- O valor default varia entre **5 MB** e **32 MB**, conforme tamanho da SGA, CPU_COUNT e arquitetura do SO), mas a Oracle notifica que muitos BDs não precisam de mais de 1MB;
- **Cuidado, Redo Log Buffer muito grande pode gerar Log File Sync Waits e aumenta risco de perda de dados** (no caso de um crash do BD);
- Em ambientes com Data Guard a Oracle recomenda no mínimo 8MB;
- O valor de redo log space requests na `v$sysstat` deve ser próx. de 0.

Configure a Keep Buffer Pool

- Configure a **Keep Buffer Pool** (KP) para otimizar o acesso a objetos ou dados que são constantemente acessados;
- Objetos na KP ficam fora da lista LRU e por isso são mantidos *fixos* na *Buffer Cache*;
- Na seção **Instance Activity Stats** do AWR, um valor muito alto na linha `table scans (short tables)` pode indicar que é bom negócio habilitar a KP p/ as tabelas pequenas (menos que 20 blocos e menos que 20% da BC) ;
- Pode-se usar a KP através dos seguintes métodos:
 - Alterando a DDL de um objeto para que ele guarde seus dados na KP;
 - Incluindo um *hint* na instrução SQL que acessa um ou mais objetos.

Configure e use Result Cache

- Configure e use **Result Cache** para otimizar acesso àquelas consultas que demoram p/ processar e que retornam poucos dados. Ótimo para BDs OLAPs;
- **Result Cache** armazena os dados em uma área reservada da *Shared Pool* (e não na Buffer Cache);
- Existem 3 formas de usar result cache:
 - **Query result cache**;
 - PL/SQL function result cache;
 - **Client OCI result cache**.
- Principais restrições de uso de query result cache:
 - Não suporta funções de tempo e hora. Ex.: SYSDATE;
 - Não suporta SEQUENCES;
 - Não suporte tabelas temporárias
 - Não suporta pseudo colunas.

Fixe objetos na Shared Pool

- Configure objetos PL/SQL (SPs, packages, functions, triggers e types) que são constantemente acessados para ficarem “fixos” (pinados) na Shared Pool;
- *Pinar* o objeto mantém ele fora da lista LRU e pode otimizar o uso de objetos que são constantemente utilizados.

Configure CURSOR_SHARING

- Em ambientes OLTP, utilize **variáveis bind** ou compartilhamento de cursores, configurando o parâmetro **CURSOR_SHARING** com o valor **FORCE**;
- O compartilhamento de cursores **reduz tempo de parse** das instruções SQL, pois reutiliza planos de execução pré-existentes. Também ajuda a economizar memória da Shared Pool;
- Em ambientes OLAP mantenha o valor de padrão de CURSOR_SHARING;

Configure CURSOR_SHARING

POC com Cursor Sharing habilitado no nível da sessão

USERNAME	EXECUTIONS	cpu_time (s)	BUFFER_GETS	SORTS	LOADS	ROWS_PROCESSED	SQL_TEXT
FABIO	4	0,002011	10	0	1	2	select * from schema.tabela where cd_objeto = :SYS_B_0"
FABIO	1	0,00137	3	0	1	1	select * from schema.tabela where cd_objeto = 11980
FABIO	1	0,001344	3	0	1	1	select * from schema.tabela where cd_objeto = 11981
FABIO	1	0,007712	3	0	1	1	select * from schema.tabela where cd_objeto = 11982
FABIO	1	0,001527	3	0	1	1	select * from schema.tabela where cd_objeto = 11983

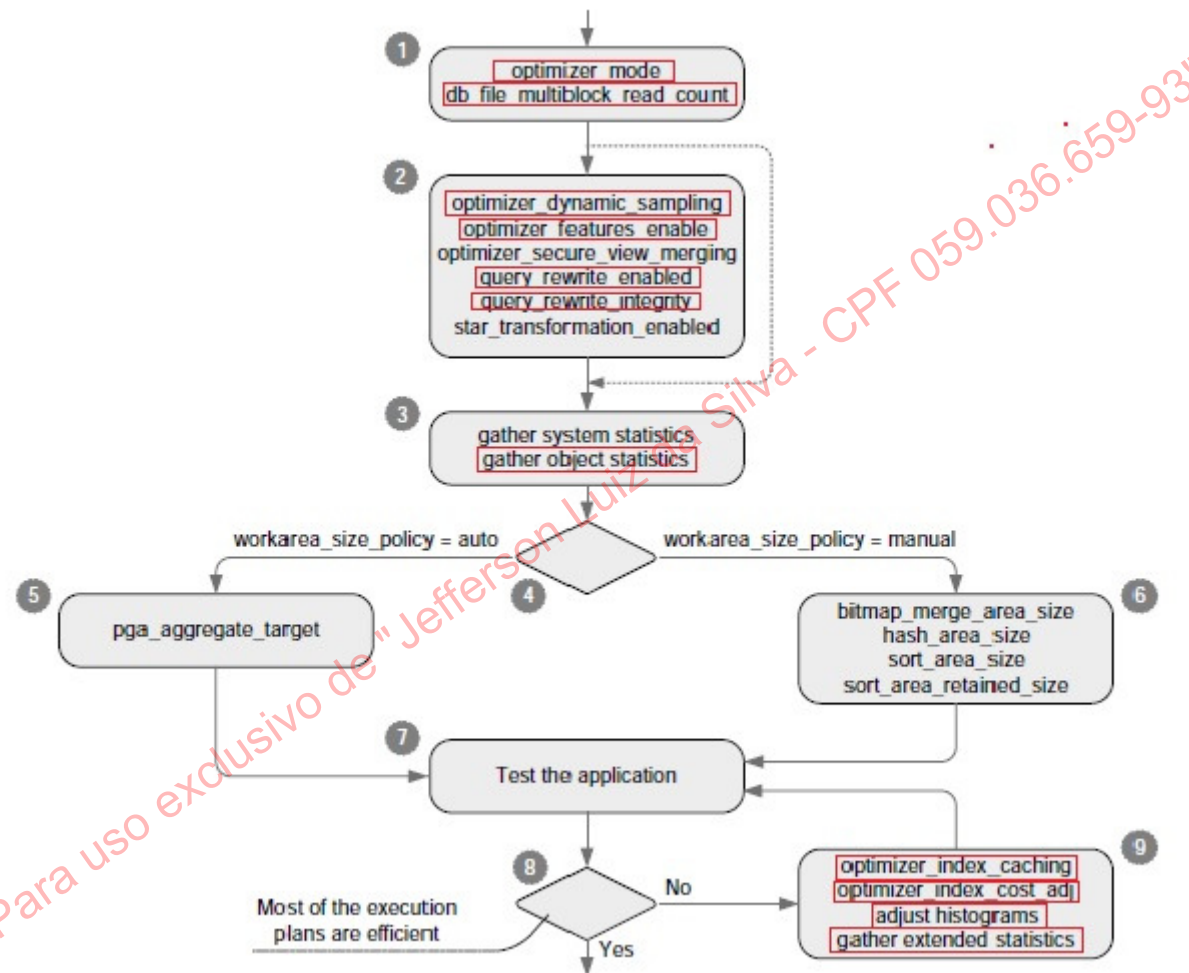
Obs.: Consulta efetuada em uma tabela do sistema de XYZ no BD PRD2 em 21/05/2010 às 18:20h

Tempo médio cpu instruções COM variável bind	0,00050275
Tempo médio cpu instruções SEM variável bind	0,00298825
GANHO DESEMPENHO COM VARIÁVEL BIND	594%

Configure OPTIMIZER_MODE

- Em ambientes OLTP é possível configurar o parâmetro **OPTIMIZER_MODE** com o valor **FIRST_ROWS** para favorecer o uso de índices pelo otimizador. Em OLAP mantenha o valor padrão;
- **ALL_ROWS** = *melhor taxa de transferência de dados*. **FIRST_ROWS** = melhor tempo de resposta.

Parâmetros que influenciam o Otimizador



Configure blocos maiores

- Configure áreas de armazenamento reservadas na *Buffer Cache* para permitir **tamanhos de blocos maiores** e otimizar acesso aos dados;
- Blocos maiores reduz I/O, mas pode gerar maior contenção de blocos (wait event latch: cache buffers chain) em atualizações concorrentes;
- Em OLTP, avalie configurar tamanhos de blocos maiores para índices muito grandes, cujas consultas necessitam de máxima performance;
- Em OLAP considere tamanhos de blocos maiores para índices e dados.

Habilite escrita assíncrona

- Em ambientes com muita escrita (normalmente OLTP), habilite **escrita de dados assíncrona**;
- Nem todos os SOs permitem escrita assíncrona (Ex.: RHEL 3 ou inferior);
- É necessário habilitar na instância a escrita assíncrona para os datafiles.

Use múltiplos DBWriters

- Configure o uso de **múltiplos processos escritores de dados**, se você não pode habilitar escrita assíncrona ou para otimizar operações de escrita do processo DBWn;
- Para configurar a quantidade ideal de processos escritores, considere a quantidade de CPUs e de discos/controladoras de discos;
- Pode ser usado p/ eliminar ou reduzir qtde. de wait events **free buffer busy waits** ou quando você encontrar no alert a msg: **Checkpoint not complete.**

Habilite paralelismo

- Para otimizar longas instruções SQL utilize paralelismo;
- Considere habilitar paralelismo automático no nível da instância ou sessão em ambientes OLAP ou de processamento batch;
- No 11G o paralelismo automático evoluiu bastante e faz **cálculo automático de DOP** conforme tamanho do segmento e carga geral do sistema. Para mais informações leia o artigo [Paralelismo automático no Oracle Database 11G](#).

Configure db_file_multiblock_read_count

- Para reduzir qtde de requisições de I/O (lógico) e consequentemente otimizar acesso aos dados, configure apropriadamente o parâmetro **db_file_multiblock_read_count**;
- Configure valores menores em OLTP e valores maiores em OLAP;
- Valor muito alto diminui o custo de operações de FTS e IFFS;
- A partir do 10GR2 seu valor é configurado automaticamente, considerando estatísticas de I/O e tamanho da buffer cache.

Configure db_file_multiblock_read_count

- Configurar este parâmetro é essencial quando não há estatísticas de sistema coletadas com carga de trabalho.

Testes em uma consulta que retorna poucas linhas utilizando um índice:

query	select * from TABELA where CODIGO in (...)		
db_file_multiblock_read_count	8	16	32
tempo médio (s)	0,05673972	0,028773735	0,037445023
ganho desempenho (%)	-97,19%	0	-30,14%

Habilite commit assíncrono

- Em ambientes com muita escrita (normalmente OLTP), habilite **commit assíncrono**;
- Isso permitirá escrita assíncrona nos *Redo Log Files* e consequentemente otimizará I/O;
- **Cuidado, commit assíncrono pode gerar perda de dados** (no caso de um crash do BD).

**Veja como 4 pequenas
alterações na instância
podem influenciar na
performance do BD**

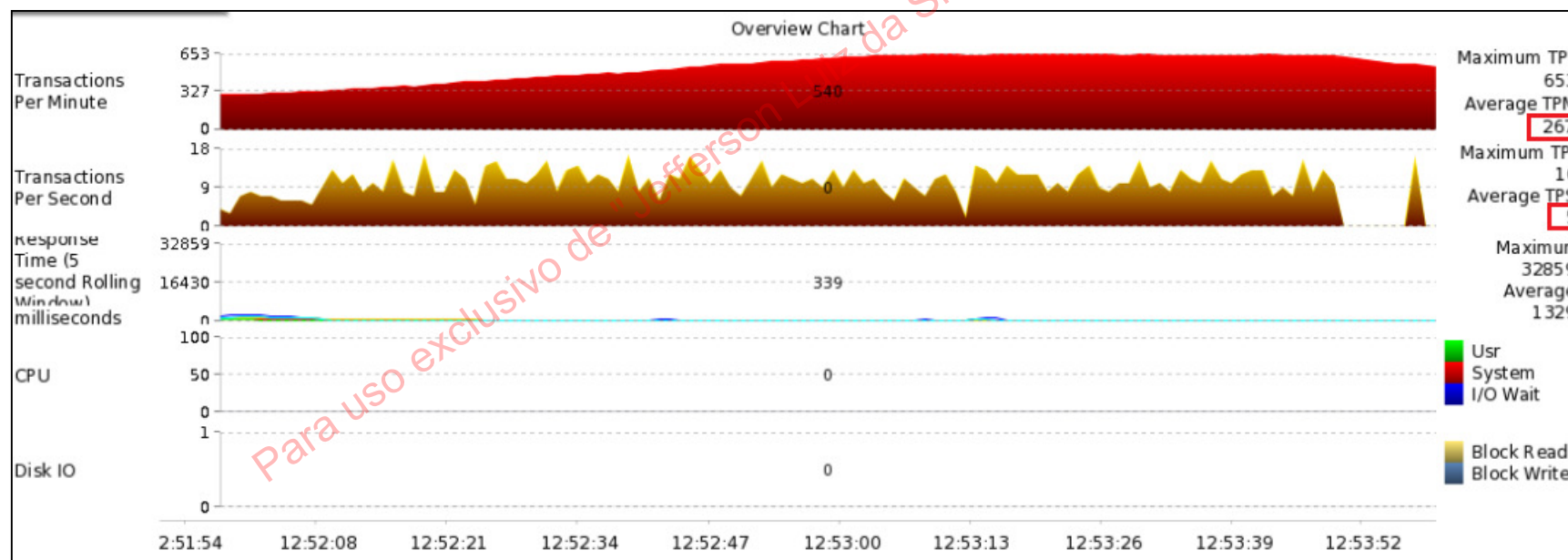
Para uso exclusivo de Jefferson Luiz de Silveira CPF 059.036.659-93"

Otimização a instância

Antes

Swingbench 15 sessões por 5 minutos, 11GR2:

- I/O síncrono: **FILESYSTEMIO_OPTIONS** = **NONE**
- 1 GB de SGA + PGA: **MEMORY_MAX_TARGET**, **MEMORY_TARGET** = **1G**
- Commit síncrono: **COMMIT_WAIT** = **WAIT**
- 1 processo escritor: **DB_WRITER_PROCESSES** = **1**

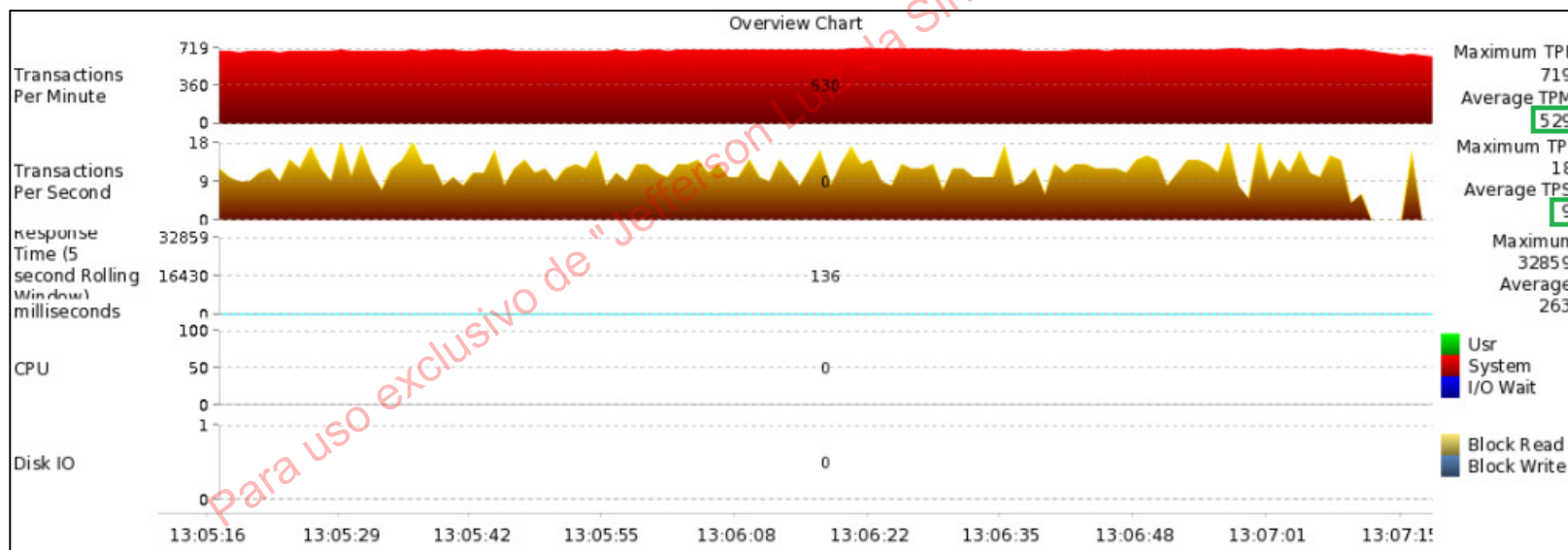


Otimizando a instância

Depois

SwingBench 15 sessões por 5 minutos, 11GR2:

- I/O assíncrono: **FILESYSTEMIO_OPTIONS = SETALL**
- 2 GB de SGA + PGA: **MEMORY_MAX_TARGET, MEMORY_TARGET = 2G**
- Commit assíncrono: **COMMIT_WAIT = NOWAIT**
- 3 processos escritores: **DB_WRITER_PROCESSES = 3**



A média de TPM aumentou de **267** p/ **529** (melhorou 198%)

**Evitar ou diminuir I/O é
essencial para otimizar a
performance de qualquer
sistema ou banco de dados**

Para uso exclusivo de "Jefferson Luiz de Silva" CPF 059.036.659-93"

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 6

Otimizando o Banco de Dados

Para uso exclusivo de "Jefferson Luiz da Silva" - CPF 059.036.659-93"

Introdução

- Principais conceitos apresentados neste capítulo:
 - Dicas diversas para otimizar a configuração e armazenamento do BD;
 - Dicas para habilitar/desabilitar recursos que podem otimizar o BD;
 - Técnicas diversas para reduzir I/O, economizar recursos e memória do BD.
- Obs.: Neste capítulo veremos diversas técnicas e dicas para otimizar um BD Oracle. Algumas podem ser aplicadas somente em ambientes OLTP, outras somente em OLAP e muitas delas nos 2 ambientes ou ambientes mistos.

Coleta de estatísticas

- Para montar um plano de execução, o *Otimizador* (CBO) baseia-se, resumidamente, em **estatísticas de objetos** (quantidade de linhas, cardinalidade, seletividade) e **estatísticas de sistema** (memória, cpu, I/O);
- As **estatísticas de sistema podem ser coletadas 1 única vez** se nenhum item de hardware for alterado na máquina servidora do BD, ou se nenhum upgrade de versão do SGBD ocorrer;
- As **estatísticas de objetos são coletadas automaticamente, em períodos diários**, a partir do Oracle 10G. *É importante ter estatísticas de objetos atualizadas para que o Otimizador monte um bom plano de execução;*
- Faz parte do trabalho do DBA executar/configurar (quando necessário) estas coletas para que o BD tenha estatísticas mais precisas ou garantir que elas ocorram;

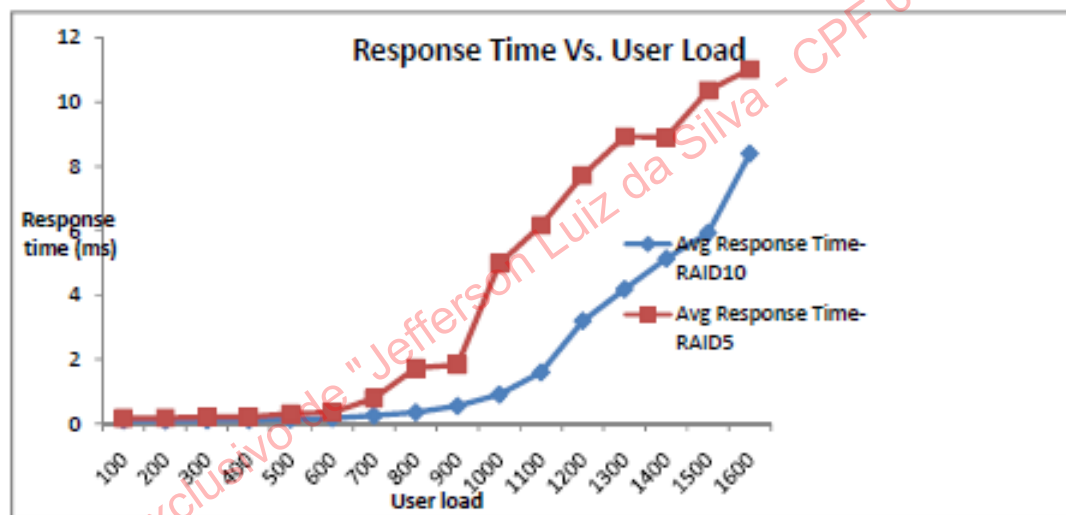
Coleta de estatísticas

- Sempre que for necessário atualizar manualmente as estatísticas dos objetos, utilize a package **DBMS_STATS**;
- Para mais informações, leia o artigo [Coletando estatísticas para o otimizador de queries do Oracle.](#)

Planeje o armazenamento

Ambiente típico OLTP c/ 70% leitura - 30% escrita

Carga de 400 GB de dados



Análise de tempo de resposta de RAID 10 X RAID 5

Fonte: Comprehending the Tradeoffs between Deploying Oracle® Database on RAID 5 and RAID 10 Storage Configurations, A Dell® Technical White Paper

Planeje o armazenamento

- Faça um bom planejamento da estrutura dos grupos de discos (ou discos) e volumes lógicos onde os dados serão armazenados;
- Configurações de RAID devem **otimizar leituras** e degradar o mínimo possível as escritas;
- Itens que podem ser avaliados:
 - Criar um volume lógico para *dados e índices*;
 - RAID sugerido: **RAID 10**;
 - Criar um volume lógico específico (isolado) para os tablespaces **SYSTEM**, **TEMP** e **UNDO**;
 - RAID sugerido: **RAID 10**;
 - Se for possível, criar volume lógicos específicos para armazenar tablespaces de objetos **quentes** (concorrentemente muito acessados);

Planeje o armazenamento

- Criar um volume lógico para os *redo logs* em um volume lógico separado dos *datafiles*;
 - RAID sugerido: RAID 1.
- Criar um volume lógico para os *archive logs* em um volume lógico separado dos *redo logs*;
 - RAID sugerido: RAID 1;
- Para reduzir a complexidade de gerenciamento do armazenamento, com ASM a Oracle geralmente recomenda não mais que 2 diskgroups (ver doc Automatic Storage Management Overview and Technical Best Practices):
 - **Database area:** datafiles control files, redo logs e CTF;
 - **Flash recovery area:** cópia control file, redo logs, archived, backups e flashback logs.

Planeje o armazenamento

Capacidade de cada disco da storage	450 GB			
TIPOS DE BDs DA SP3	QTDE. DISCOS	CAPACIDADE TOTAL		
PRODUÇÃO (, host)	17	6.15 TB		
HOMOLOGAÇÃO (, VM na)	17	6.15 TB		
DISTRIBUIÇÃO DOS DISCOS POR TIPO DE BD				
Finalidade principal	Qtde. discos	Tipo RAID	Onde?	Capacidade útil total
Armazenar Datafiles	8	RAID 10 (1+0)	Storage	1.8 TB
Armazenar Índices	2	RAID 0	Storage	900 GB
Armazenar Redo Logs	4	RAID 1	Storage	900 GB
Backups	3	RAID 5	Storage	900 GB
Armazenar Archive Logs	3	RAID 5	Discos locais	900 GB
DISTRIBUIÇÃO DE LVMs RAID 10				
Nome do LVM	Tipos de arquivos	Tamanho	Observações	
/ora01	Datafiles	1.2 TB		
/oratbs	Datafiles system, undo e temp	200 GB	Armazenar este volume em discos diferentes do LVM /ora01	
/ora04	Flashback Archive Logs	200 GB		
DISTRIBUIÇÃO DE LVMs RAID 0				
Nome do LVM	Tipos de arquivos	Tamanho	Observações	
/ora02	Índices	900 GB		
DISTRIBUIÇÃO DE LVMs RAID 1				
Nome do LVM	Tipos de arquivos	Tamanho	Observações	
/ora04	Control Files + Redo Logs	200 GB		
/ora05	Control Files + Redo Logs	200 GB	Armazenar este volume em discos diferentes do LVM /ora04	
/ora00	Software Oracle	100 GB		
/oraexp	Exports	300 GB		
DISTRIBUIÇÃO DE LVMs RAID 5 STORAGE				
Nome do LVM	Tipos de arquivos	Tamanho	Observações	
/orabkp	Backups	900 GB		
DISTRIBUIÇÃO DE LVMs RAID 5 DISCO LOCAL				
Nome do LVM	Tipos de arquivos	Tamanho	Observações	
/oraarc	Archive Logs	200 GB		
/oratmp	Arq. Temporários	100 GB		

Desfragmente os segmentos

- Para otimizar leitura (principalmente FTS), desfragmente os segmentos de tabelas, índices, LOBs e MVs utilizando o **SHRINK**;
- Principais características do **SHRINK**:
 - Elimina blocos vazios (não utilizados) e reduz a qtde. de linhas migradas dos segmentos;
 - É executado em modo **ONLINE** e atualiza índices;
 - **Não necessita de espaço extra e não invalida cursores** ;
 - Executa reorganização interna através de INSERTs e DELETEs internos, mas não dispara triggers;
 - Pode fazer a desfragmentação e movimentação da HWM em processo separados;
 - Possui algumas restrições:
 - Não pode ser usado em segmentos de UNDO, tabelas clusterizadas e tabelas com colunas LONG;
 - A partir do 11G pode compactar tablespaces temporários;
- **Importante:** Ao utilizar o SHRINK em tabelas ou MVs indexadas, utilize a opção **CASCADE**:

Elimine o encadeamento de linhas

- Para otimizar acesso (FTS ou IS) aos dados, reorganize as tabelas, eliminando o encadeamento de linhas;
- Encadeamento é causado por INSERTs e UPDATES:

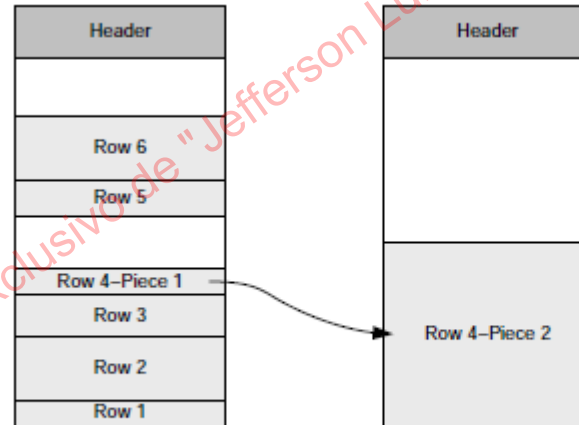


Figure 12-4. Updated rows that can no longer be stored in the original block are migrated to another one.

Fonte: Troubleshooting Oracle Performance, Editora: Apress, Autor: Christian Antognini

Elimine a migração de linhas

- Para otimizar acesso (FTS ou IS) aos dados, reorganize as tabelas, eliminando também a migração de linhas;
- Migração de linhas é causada por UPDATES:

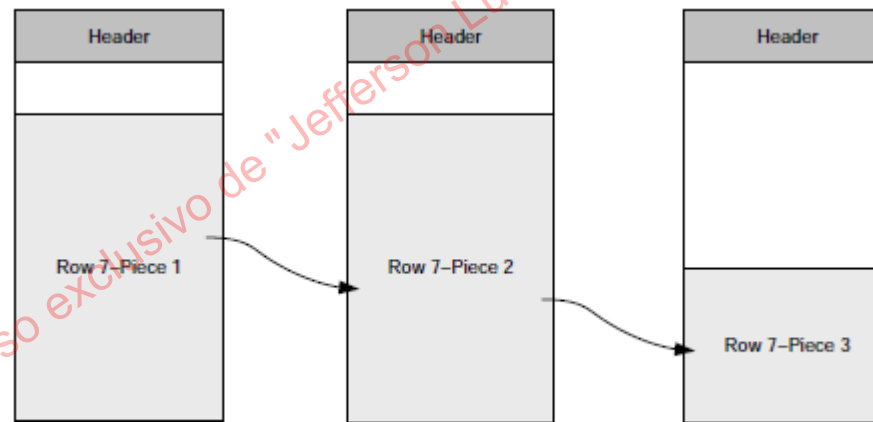


Figure 12-5. A chained row is split into two or more pieces.

Fonte: Troubleshooting Oracle Performance, Editora: Apress, Autor: Christian Antognini

Eliminando encadeamento/migração

- Existem várias formas de executar este processo:
 - Usando a package *DBMS_REDEFINITION*;
 - *CTAS + rename table*;
 - *ALTER TABLE MOVE* ou Mover dados para outro tablespace e voltar dados para tablespace original;
 - Export, TRUNCATE, Import;
 - **Reinserir linhas encadeadas:**
 - 1) *ANALYZE TABLE CHAINED_ROWS* para identificar linhas encadeadas;
 - 2) Salvar linhas encadeadas em tabela temporária;
 - 3) Deletar as linhas encadeadas;
 - 4) Reinserir as linhas encadeadas a partir dos dados da tabela temporária.

Gerencie eficientemente os tablespaces

- Crie **tablespaces gerenciados localmente (TGLs)**:
 - Causam menos contenção e são + rápidos em ambientes c/ alta carga.
- Configure **ASSM** (Automatic Segment Space Management) ao invés de gerenciar **PCTUSED**, **freelists** e **freelists groups**:
 - Otimiza a maior parte das aplicações OLTP e é recomendado p/ RAC;
 - Em casos específicos, com alta concorrência em FTS e alto volume de INSERTs, avalie o uso do MSMM;
- Configure o **auto-incremento** no tablespace calculando o valor ideal para ocorrer a menor quantidade de extensões automáticas possível:
 - O objetivo principal é diminuir o SQL recursivo executado pelo SYS, podendo eliminar ou reduzir wait event *Datafile init write*.
- Para ver mais dicas, leia o artigo [Boas práticas para gerenciar Tablespaces](#).

Crie Bigfile Tablespaces

- Até o 10G você podia ter bons motivos para não criar **Bigfile Tablespaces** (Ex.: Tamanho do backup). A partir do 11G não há mais;
- *Bigfile Tablespaces* oferecem os seguintes benefícios:
 - Simplificam o gerenciamento de datafiles;
 - Permitem criar BDs 1024 X maiores do que com *SmallFile Tablespaces*;
 - Melhoram a performance dos seguintes itens:
 - Abertura do BD;
 - Processos de checkpoint;
 - Processos DBWR.
 - Podem evitar ou reduzir os seguintes *wait events*:
 - Control File single write, Control File parallel write e Control Sequential read.
- Detalhe importante:
 - Bigfile Tablespaces só podem ser criados em tablespaces ASSM.

Crie índices p/ otimizar consultas

- Crie **índices para otimizar consultas**, mas tome **cuidado com as atualizações**;
- Índices são essenciais para as consultas em ambientes OLTP, de todo modo, crie somente os índices necessários (e elimine os índices desnecessários);
- Segundo o *Performance Tuning Guide da Oracle*, cada índice criado em uma tabela degrada em média 3 X a performance das atualizações!

Crie Tablespaces c/ blocos maiores

- Crie tablespaces com **tamanhos de blocos maiores** que o tamanho padrão (8k) para índices em OLTP e para dados e índices em OLAP;
- Tamanho de blocos maiores criam estruturas de árvores binárias menores em *índices b-tree*. Com a árvore menor, fica mais rápido acessar os dados indexados;
- **CUIDADO!** Tamanhos de blocos maiores otimiza a leitura dos dados, porém pode ocasionar contenção de linhas (muitos locks) em tabelas que sofrem muitas atualizações concorrentes. Índice de chave reversa pode amenizar este problema.

Crie Tablespaces c/ blocos maiores

query	select * from TABELA where CODIGO < 13500 and rownum < 100										
tablespace dados	8	8	8	8	8	16	16	32	32	32	32
tablespace indice	8	16	32	16	32	8	8	8	8	32	32
tablespace dados com logging	sim	sim	sim	sim	sim	sim	não	sim	não	não	não
tablespace indice com logging	sim	sim	sim	não	não	sim	sim	sim	sim	sim	não
tempo médio (s)	0,220108258	0,190573238	0,059496245	0,07346511	0,06879801	0,17752898	0,08973762	0,247605813	0,05682986	0,05367079	
ganho desempenho (%)	0	-42,85%	55,40%	44,93%	48,43%	-33,07%	32,73%	-85,60%	57,40%	59,77%	100,00%

TABELA	TABELA
TOTAL LINHAS	2902
TAMANHO	3,3 MB

Configure PCTFREE e INITRANS

- Tabelas que sofrem muitas alterações podem ter configurações diferenciadas:

– PCTFREE:

- Espaço livre reservado em cada bloco para permitir extensões de dados dentro do mesmo bloco e evitar LEMs;
- Valor padrão é 10. Aumente para **evitar ou minimizar** LEMs ou alta concorrência quando existem muitas linhas no mesmo bloco.

– INITRANS:

- Configura quantidade inicial de transações que podem ser executadas no mesmo bloco;
- Valor padrão em tabelas é 1 e em índices é 2. Valor máximo permitido é 255 ou no máximo 50% do tamanho do bloco;
- Cada INITRANS ocupa 24 bytes no cabeçalho do bloco;
- Blocos de 8K permitem no máximo 169 transações concorrentes:
 $169 * 24 = 4056 \text{ bytes}$, que é igual a 48% de 8k
- Configure um valor aproximado ao total de transações concorrentes para **minimizar ou evitar contenção de blocos**.

Gerencie colunas LOB

- Tabelas com colunas LOB requerem cuidados especiais;
- Para otimizar performance (principalmente FTS que não envolvem a coluna LOB) e evitar fragmentação de linhas, **armazene as colunas LOB em um tablespace separado da tabela, preferencialmente com tamanho de bloco maior;**
- O espaço nunca é reutilizado, portanto, crie rotinas de manutenção periódicas para fazer **SHRINK** nas colunas LOB de tabelas que sofrem muitas deleções;
- Dados de colunas do tipo LOB por padrão **não são armazenadas na Buffer Cache**. Verifique se não vale a pena mudar este modo de armazenamento para as colunas que são constantemente acessadas. Usar Flash Cache é uma boa opção!

Gerencie colunas LOB

- Usar **Flash Cache** como cache de LOBs é uma boa opção!
- Para mais informações leia doc **LOB Performance Guideline (Doc ID 268476.1)** no MOS.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 050.036.659-93"

Crie SecureFile Lobs

- A partir do 11G (padrão no 12c), pode-se criar colunas LOB do tipo **SecureFiles**, que permitem adicionar os seguintes recursos aos dados da coluna:
 - **Compressão**
 - **Desduplicação**
 - **Criptografia**
- No 11G **Parallel DML** pode ser executado em **SecureFiles LOB** de tabelas particionadas. No 12C, pode ser executado também em tabelas não-particionadas;
- Compressão e desduplicação dependem de licenciamento da option *Advanced Compression*. Criptografia dependem de licenciamento da option *Advanced Security*.

Crie Tabelas particionadas

- Crie **tabelas particionadas** para otimizar acessos e atualizações de dados em tabelas grandes (considerar tabelas maiores que 2GB);
- São ótimas para **operações FTS** em VLDBs e BDs OLAP (tabelas de histórico);
- O acesso às partições é totalmente transparente para as aplicações;
- Requer licença da **Option Partitioning**. Para mais informações leia o artigo [Criando tabelas particionadas para otimizar consultas em tabelas](#).

Escolha o melhor índice particionado

- Crie índices particionados globais em ambientes OLTP para otimizar acesso às consultas que retornam poucos dados;
- Crie índices particionados locais em ambientes OLAP para facilitar o gerenciamento das partições;
- Índices Bitmap (mais indicados em OLAP) em tabelas particionadas são sempre locais.

Utilize NOLOGGING

- Utilize **NOLOGGING** no nível de instrução, tabela ou tablespace para otimizar operações tais como:
 - CTAS;
 - Direct load;
 - Create ou alter index;
 - Alter Table Move;
- Não funciona em BDs configurados com FORCE LOGGING (configuração recomendada p/ Data Guard) e não afeta instruções DML convencionais;
- **Cuidado!** NOLOGGING reduz geração de informações de redo e por isso pode impossibilitar a recuperação de dados em determinadas operações do BD.

Utilize Paralelismo

- Utilize paralelismo no nível das tabelas ou **instruções SQL** (através de *hints*);
- É muito útil em OLAP ou grandes consultas e grandes atualizações (Ex.: tabelas particionadas), mas em **ambientes OLTP com muitos usuários concorrentes** e pequenas transações ou pequenas consultas, **pode sobrecarregar** (ao invés de otimizar) o BD;
- Oracle XE (Express Edition) e Standard Edition não permitem o uso de paralelismo.

Cuidado com Auditoria

- No Oracle existem 3 formas de auditar o BD:
 - Auditoria Padrão;
 - FGA (Fine Grained Auditing);
 - Auditoria baseada em valor;
- **Evite auditar** objetos ou operações desnecessárias. **Audite somente o que for necessário pelo tempo necessário**, pois ela gera consumo adicional de CPU e I/O;

Cuidado com Auditoria

Testes de auditoria padrão em Oracle 11GR2 gerando aproximadamente **250 registros de auditoria por segundo**, usando 50% de carga da CPU, antes de habilitar a auditoria:

Audit Trail Setting	Additional Throughput Time	Additional CPU Usage
OS	1.39%	1.75%
XML	1.70%	3.51%
XML, Extended	3.70%	5.26%
DB	4.57%	8.77%
DB, Extended	14.08%	15.78%

Fonte: Oracle Corporation

Gerencie a Auditoria

- Avalie gravar os registros de auditoria em arquivos do SO, pois esta é a forma de gravação que gera a menor sobrecarga de desempenho;
- Se for utilizada auditoria com gravação em BD, crie um tablespace exclusivo para as tabelas SYS.AUD\$ e SYS.FGA_LOG\$;
- Para otimizar consultas aos registros de auditoria, crie uma rotina para limpar os registros obsoletos.

Utilize compressão de dados

Compression Method	Compression Ratio	CPU Overhead	CREATE and ALTER TABLE Syntax	Typical Applications
Basic table compression	High	Minimal	COMPRESS [BASIC]	DSS
OLTP table compression	High	Minimal	COMPRESS FOR OLTP	OLTP, DSS

Fonte: Oracle Corporation

Obs.: A Compressão OLTP necessita de licença adicional da Option *Oracle Advanced Compression*.

Utilize compressão de dados

- **Tabelas compactadas** apesar de consumirem mais CPU, economizam espaço em disco e **reduzem a quantidade de I/O**;
- Compressão de dados nas tabelas pode **otimizar as consultas e deleções**, mas normalmente degrada as atualizações e inserções;
- Se a tabela tiver muitos INSERTs, pode causar *WE cell single block physical read*;
- Tipos de compressão de tabelas:
 - **Basic**: Comprime até 10X operações de carga direta;
 - **OLTP**: Antigo *FOR ALL OPERATIONS*. Comprime de 2X à 4X todas as operações DML;

Utilize compressão de dados

```
SELECT table_name,compression,num_rows FROM my_user_tables;
```

TABLE_NAME	COMPRESSION	NUM_ROWS
EMP_NC	DISABLED	1835008
EMP_DSS	ENABLED	1835008

```
SELECT COUNT(ename) FROM emp_nc
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	2	0.31	1.88	10974	10978	0	1
total	4	0.31	1.88	10974	10978	0	1

```
SELECT COUNT(ename) FROM emp_dss
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	2	0.56	0.71	3057	3060	0	1
total	4	0.56	0.71	3057	3060	0	1

Fonte: *Data Compression in Oracle*, Carl Dudley, University of Wolverhampton - UK

Dicas diversas

- Crie Visões Materializadas;
- Crie Tabelas Clusterizadas;
- Escolha apropriadamente o melhor Character Set para o seu Banco de Dados;
- Utilize Smart Flash Cache (11GR2):
 - *Flash Cache* permite usar SSD para armazenar dados que não caberiam na buffer cache e gerar menos I/O quando a SGA não tem tamanho suficiente para comportar dados com mais eficiência (menos I/O). Para mais informações, leia o artigo [Otimizando Bancos de Dados Oracle com Database Smart Flash Cache](#).

Dicas diversas

- Crie índices em discos SSD:
 - Ver artigo [Otimizando I/O com discos SSD](#).
- Dimensione apropriadamente o tamanho dos Redo Logs.

Para uso exclusivo de "Jefferson Luiz da Silva" CPF 059.036.659-93"

Exercícios

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Exercícios

- 1- Implemente 3 dicas de tuning que você viu nos capítulos 5 e 6 para resolver os problemas que você encontrou nos exercícios do capítulo 4;
- 2- Crie um novo snapshot;
- 3- Gere um novo AWR Report, compare com o anterior e veja se algo melhorou (se o seu Diagnóstico gerou resultados positivos).

Boa performance é a ausência de má performance

Oracle Essentials - Oracle Database 11g

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Fábio Prado

Capítulo 7

Wait Events X Otimização

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Introdução

- Neste capítulo você verá uma lista de *Wait Events* e possíveis soluções para otimizar o BD e eliminar ou reduzir estes Wait Events.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.037.659-94"

db file sequential read

- O que pode ser feito:
 - Otimize instruções SQL;
 - Configure apropriadamente a SGA (aumentando a **buffer cache**);
 - Reduza I/O;
 - Verifique se os índices são eficientes e melhore a indexação (rebuild ou coalesce talvez ajude);
 - Considere o uso de tabelas clusterizadas ou **particionadas**;
 - Utilize *Flash Cache*.

db file scattered read

- O que pode ser feito:

- Crie índices apropriados ou visões materializadas;
- Otimize instruções SQL;
- Use compressão de dados;
- Use tabelas particionadas;
- Reduza I/O;
- Verifique se as estatísticas de objetos estão atualizadas;
- Aumente o tamanho da *Buffer Cache*;
- Desfragmente as tabelas;
- Elimine linhas encadeadas e migradas;
- Utilize *Result Cache* ou *Keep Pool* ou *Smart Flash Cache*;
- Se as tabelas sofrem poucas atualizações, considere armazená-las em tablespaces com tamanhos de blocos maiores.

direct path read / direct path read temp direct path write / direct path write temp

- O que pode ser feito:

- Aumente o tamanho da PGA;
- Otimize o I/O;
- Habilite I/O assíncrono;
- Reduza a quantidade de *sorts*;
- Reduza o uso de paralelismo nas consultas.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

global cache cr request

- O que pode ser feito:
 - Otimize as instruções SQL para acessar/recuperar menos dados;
 - Aumente o tamanho da buffer cache;
 - **Otimize a rede Interconnect.**

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.038.659-93"

buffer busy waits / read by other session

- O que pode ser feito:

- Otimize as queries para carregar menos dados em memória;
- Melhore a indexação;
- Evite manutenção de índices em horário de utilização das respectivas tabelas.
- Elimine FTS desnecessários;
- Se estiver ocorrendo muitos UPDATES e/ou INSERTs, considere aumentar PCTFREE e/ou INITRANS;
- Considere mover objetos que apresentam *Segment Header Contention* para tablespaces ASSM;
- Utilize compressão de dados;
- Aumente a quantidade de *database writers*.

SQL*Net more data from dblink

- O que pode ser feito:
 - Reduza a quantidade de dados a serem transferidos via dblink;
 - Crie visões materializadas com atualizações incrementais temporárias para substituir acesso(s) remoto(s);
 - Otimize a rede entre os BDs local e remoto.

Obs.: Este evento pode não ser um problema!

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

- O que pode ser feito:
 - Otimize as aplicações para fazer menos COMMITs;
 - Aumente o tamanho dos redo log files;
 - Verifique se o parâmetro FAST_START_MTTR_TARGET está configurado com um valor muito baixo (e diferente de zero);
 - Habilite commit assíncrono;
 - Otimize I/O onde os *redo log files* são gravados;
 - Reduza o tamanho da log buffer.

library cache lock

- O que pode ser feito:
 - Elimine a causa dos bloqueios (mate a sessão bloqueadora);
 - Evite ou reduza recompilação de objetos PL/SQL;
 - Evite executar frequentemente instruções SQL dinâmicas individuais;
 - Evite instruções DDL que alteram a estrutura de objetos em horário de produção;
 - Evite redimensionamentos frequentes de memória quando estiver habilitado ASMM ou AMM;
 - **Escreva instruções SQL com variáveis bind.**

SQL*Net more data to client

- O que pode ser feito:

- Reduza a quantidade de dados a serem retornados pelas queries;
- Otimize a rede entre o BD e os clientes;
- Utilize stored procedures.

Obs.: Este evento pode não ser um problema!

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

db file parallel read / db file parallel write

- O que pode ser feito:

- Otimize as instruções SQL;
- Melhore a indexação;
- Otimize I/O;
- Habilite I/O assíncrono;
- Distribua I/O entre discos ou grupos de discos diferentes;
- Aumente o tamanho da buffer cache.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

library cache pin / Cursor Pin S wait on X

- O que pode ser feito:
 - Aumente o tamanho da **Shared Pool**;
 - Execute flush na *Shared Pool* para possibilitar liberação de bloqueios;
 - Evite ou reduza recompilação de objetos PL/SQL;
 - Reduza o tempo o parse das instruções SQL;
 - Evite executar frequentemente instruções SQL dinâmicas;
 - Execute instruções DDL que alteram a estrutura de objetos em janelas de manutenção;
 - Evite redimensionamentos frequentes de memória quando estiver habilitado ASSM ou AMM;
 - Aplicações devem abrir cursores o mais tarde possível e liberá-los o mais cedo possível.

- O que pode ser feito:
 - Aumente o valor do parâmetro LOG_BUFFER;
 - Mova os log files para discos mais rápidos;
 - Utilize NOLOGGING;
 - Altere somente o que for necessário (se você precisa atualizar 2 colunas de uma tabela, construa uma instrução UPDATE que atualiza somente as 2 colunas e não TODAS as colunas).

enq: TX - row lock contention

- O que pode ser feito:
 - Otimize instruções SQL que fazem INSERT, UPDATE, DELETE ou MERGE;
 - Finalize as transações o mais rápido possível;
 - Use Stored Procedures;
 - Use GTTs para carregar dados temporários;
 - Elimine índices bitmaps em tabelas que são constantemente atualizadas;
 - Se estiverem ocorrendo muitas transações concorrentes na mesma tabela, aumente PCTFREE e INITRANS.

cache buffers chains

- O que pode ser feito:

- Otimize as queries;
- Elimine FTS em tabelas pequenas;
- Otimize ou substitua índices ineficientes;
- Utilize particionamento;
- Utilize compressão de dados;
- Aumente o tamanho da Buffer Cache.

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

FIM

Para uso exclusivo de "Jefferson Luiza da Silva - CPF 059.036.659-93"

Performance Tuning

em Bancos de Dados Oracle

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Final

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"

Dicas finais

- Outras dicas e recursos que podem ser avaliados, conforme as configurações e características do ambiente:
 - Habilitar **Shared Servers** (ver artigo [Configurando Shared Servers no Oracle](#));
 - Implementar **Connection Pooling**;
 - Configurar apropriadamente o tamanho dos tablespaces **UNDO** e **TEMP**;
 - Configurar **gerenciamento automático** de **UNDO**;
 - Verificar necessidade de criar mais tablespaces temporários ou grupos de tablespaces temporários. No 12c, crie no mínimo um tablespace temporário por PDB;

Dicas finais

- Criar Tabelas `IOT`, `GTT`;
- Configurar valor do parâmetro `FAST_START_MTTR_TARGET`;
- Configurar `Recycle Pool`;
- Compactar periodicamente o(s) tablespace(s) temporários;
- Configurar e habilitar o `Database Resource Manager`;
- Utilizar `queries paralelizadas` em ambiente RAC;
- Utilizar `SQL Performance Analyzer`;
- Configurar apropriadamente o valor do parâmetro `OPTIMIZER_FEATURES_ENABLE`;

Dicas finais

- Incrementar valor do parâmetro `SESSION_CACHED_CURSORS` para reduzir soft parses;
- Utilizar `Stored Procedures` (ver artigo [Otimizando a performance de aplicações com o uso de stored procedures](#));
- Efetuar `menos COMMITs`;
- Configurar `HugePages` (ver artigo [Otimizando Oracle Database com HugePages](#));
- Para identificar `I/O lento` leia o MOS doc ID 301137.1: OS Watcher User Guide
- Configurações OLAP X OLTP:
 - Ver artigo [Tuning em Bancos de Dados OLAP](#).

Avaliação

- Gostaram do treinamento?
- Por favor, façam **sugestões**, **comentários** ou **críticas** preenchendo a “Ficha de Avaliação”.

Para uso exclusivo de "Jefferson Luiz da Silva, CPF 059.036.659-93"

Blog

- Acompanhe os artigos do blog **FABIOPRADO.NET** em <http://www.fabioprado.net>



Facebook

CURTA a página do blog **FABIOPRADO.NET** no **FACEBOOK** para acompanhar:

- Novidades e notícias do blog;
- Novidades e notícias sobre Bancos de Dados e TI, em geral;
- Oportunidades de emprego.



<http://www.facebook.com/DBAFabioPrado>

SIGA o twitter do DBA FABIOPRADO.NET para acompanhar:

- Novidades e notícias do blog;
- Novidades e notícias sobre Bancos de Dados e TI, em geral;
- Oportunidades de emprego.



<http://twitter.com/dbafabioprado>

LinkedIn

Se você gostou do treinamento, **POR FAVOR**, deixe uma recomendação no perfil do **Fábio Prado** no *LinkedIn*:



The screenshot shows the LinkedIn profile of Fábio Prado. On the left is a profile picture of a man at an Oracle event. To the right, the name 'Fábio Prado' is displayed above his title 'DBA/Instrutor Oracle' and location 'São Paulo e Região, Brasil'. His current company is 'FABIOPRADO.NET, TRE-SP'. Below this, a dropdown menu is open, showing options: 'Sugerir atualização', 'Recomendar' (highlighted in blue), and 'Localizar referências'. To the right of the menu, it says '+ de 500 conexões'. At the bottom right of the profile section is a button for 'Informações de contato'. Below the profile picture is a link to his public profile: 'br.linkedin.com/pub/fabio-prado/37/261/71/'. At the bottom left of the profile section is a button labeled 'Histórico profissional'.

Fábio Prado
DBA/Instrutor Oracle
São Paulo e Região, Brasil | Treinamento e orientação profissional

Atual FABIOPRADO.NET, TRE-SP
Voltar Ka Solution, Veris IBTA, Grupo Pão de Açúcar
Formação acadêmica Veris IBTA

Enviar mensagem ▼

+ de 500 conexões

Sugerir atualização
Recomendar
Localizar referências

Informações de contato

br.linkedin.com/pub/fabio-prado/37/261/71/

Histórico profissional

Referências

A Study of the Top 15 Wait Events that Cause Database Delays:

Confio Software

Best Practices for Gathering Optimizer Statistics:

An Oracle White Paper, April 2016

Comprehending the Tradeoffs between Deploying Oracle Database on RAID 5 and RAID 10 Storage Configurations:

A Dell Technical White Paper, April 2009

Expert Oracle Practices:

Editora: Apress, Autores: Jonathan Lewis, Graham Wood e outros ...

Logging or Nologging, That is the Question:

Autor: Francisco Munoz Alvarez

My Oracle Support:

<https://support.oracle.com>

Referências

Oracle Database Auditing: Performance Guidelines:

An Oracle White Paper, August 2016

Oracle Database Performance Tuning Guide 11G Release 2:

http://docs.oracle.com/cd/E11882_01/server.112/e16638/toc.htm

Oracle Database 11g Performance Tuning Recipes:

Editora: Apress, *Autor:* Sam R. Alapati, Darl Kuhn e Bill Padfield

Oracle Essentials - Oracle Database 11g

Editora: O'Reilly, *Autor:* Rick Greenwald, Robert Stackowiak e Jonathan Stern

Oracle Performance Diagnostics & Tuning

Autor: Nerv Informática

Referências

Particionamento com o Oracle Database 11g Release 2:

Um artigo técnico da Oracle, Setembro de 2016

Tuning When you can't touch the code

http://www.nyoug.org/Presentations/2000/no_touch_s.pdf

Interpreting AWR Report - Straight to the Goal

<http://prezi.com/glqm9zemzhup/interpreting-awr-report-straight-to-the-goal/>

Até a próxima!

Para uso exclusivo de "Jefferson Luiz da Silva - CPF 059.036.659-93"