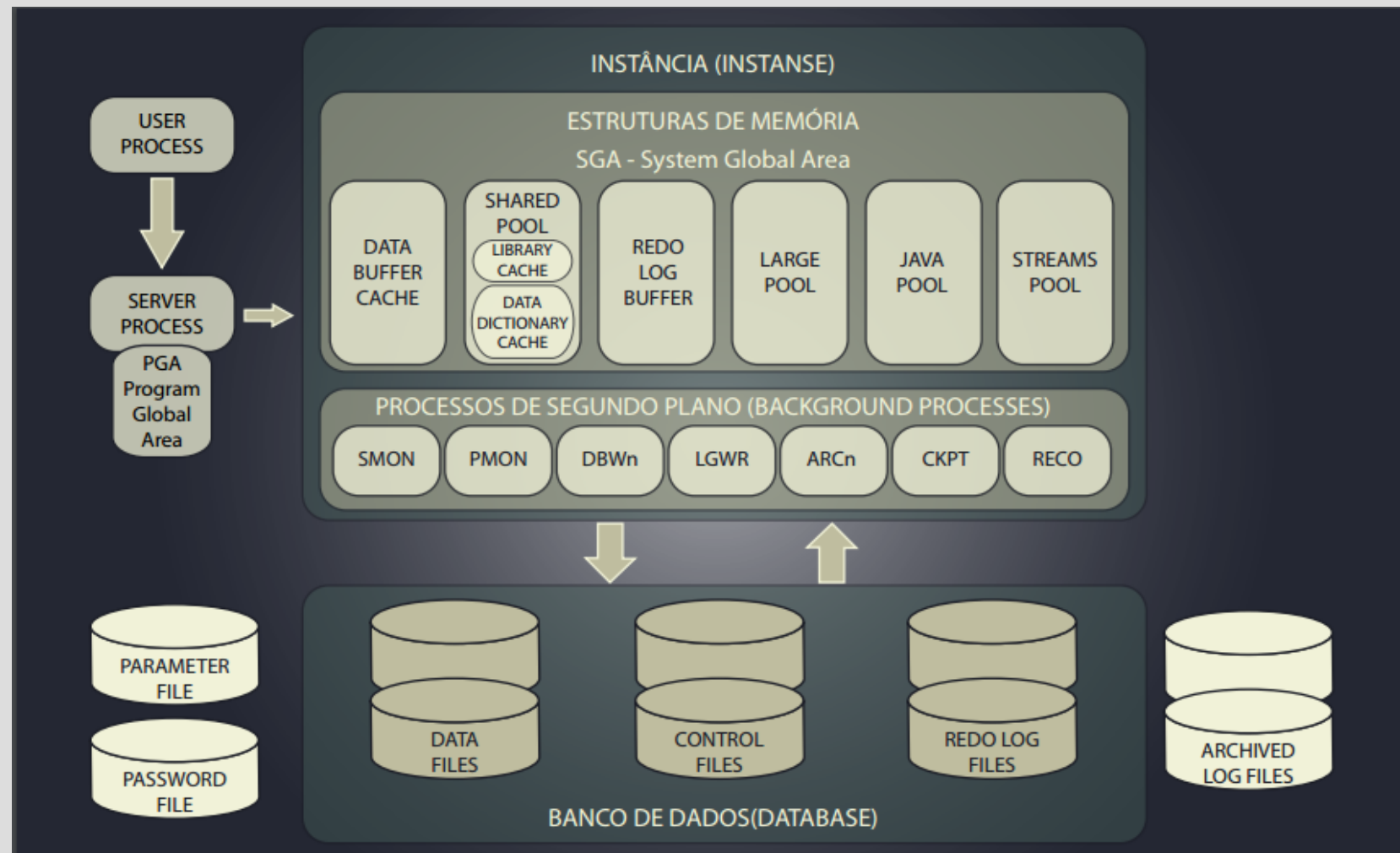


Gerenciamento em Banco de Dados

leonardo.mantovani@uni9.pro.br

Gerenciamento em Banco de Dados

Arquitetura Física – Instância (Instance)



Gerenciamento em Banco de Dados

PGA - Program Global Area

- Área de memória: SGA (que pertence a instância) e a PGA – Program Global Area – Área global de programa
- Área da memória dedicada exclusivamente para cada processo Oracle.
- Memória não compartilhada
- A memória total usada por todas as PGAs individuais é conhecida como instância PGA.
- Os parâmetros de inicialização do banco de dados definem o tamanho da instância PGA e não as PGAs individuais.
- Região privativa na qual são armazenados dados temporários, tais como: variáveis ou endereços de rotinas.

Gerenciamento em Banco de Dados

PGA - Program Global Area

Dois tipos de configuração:

- Servidor dedicado: cada processo de usuário tem conexão exclusiva com o banco de dados (uma PGA para cada processo de usuário)
- Servidor compartilhado: diversos usuários compartilham a mesma conexão. Menor consumo de memória porém maior tempo de resposta. (ao invés de trabalhar com processos de modo paralelo, trabalharia com enfileiramento/paralelismo)
 - ◆ Situações onde o SO tem dificuldades em lidar com muitos processos acontecendo de modo paralelo

Gerenciamento em Banco de Dados

Instância

- Composta por:
- Estruturas de memória (SGA – System Global Area)
- Processos de segundo plano (background process)

Gerenciamento em Banco de Dados

Instância

- Deve ser iniciada para que se tenha acesso aos dados presente no banco de dados
- Toda vez que uma instância é iniciada uma SGA é alocada e os processos de segundo plano são iniciados
- Informações da instância encontram-se presentes na visão V\$INSTANCE
 - ◆ `SELECT INSTANCE_NAME FROM V$INSTANCE;`

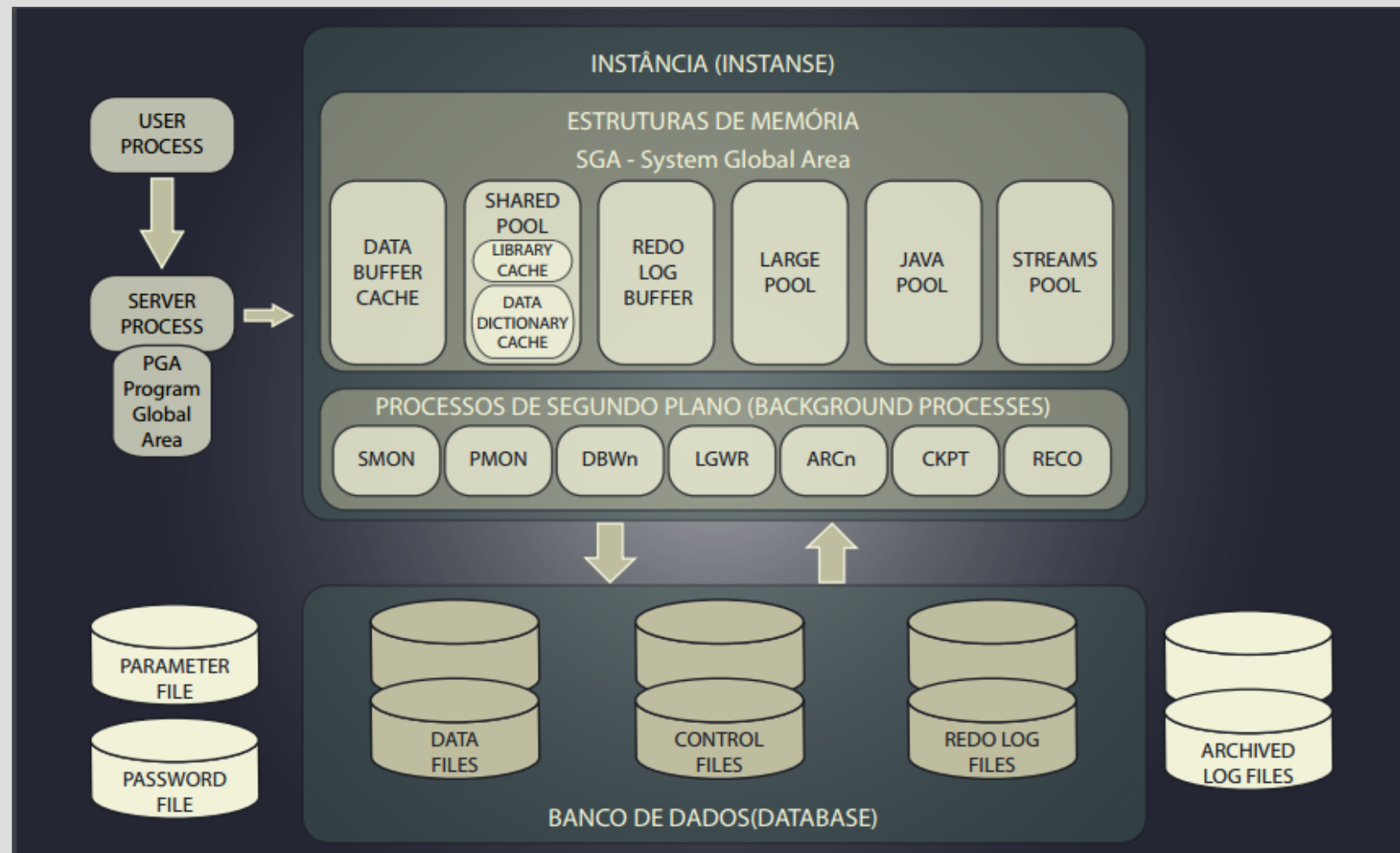
Gerenciamento em Banco de Dados

SGA - System Global Area - Área global do sistema

- Área de memória compartilhada para uma instância Oracle
- Contém dados e controla informações do Oracle
- Quando a instância é inicializada é alocada para a SGA uma quantidade de memória que pode chegar até seu limite expresso pelo parâmetro `SGA_MAX_SIZE`
- Caso `SGA_MAX_SIZE` não esteja especificado, usará o parâmetro `SGA_TARGET`, sendo que `SGA_TARGET` pode ser alterado com a instância em execução
- 11g - `MEMORY_TARGET`: equilíbrio entre SGA e PGA

Gerenciamento em Banco de Dados

SGA



Gerenciamento em Banco de Dados

SGA - System Global Area - Área global do sistema

- Principais sessões que compõem a SGA
- Shared Pool
- Buffer Cache
- Redo Log Buffer
- Large Pool
- Java Pool
- Streams Pool

Gerenciamento em Banco de Dados

Shared Pool (Pool compartilhado)

- Possui dois subcaches principais: Library Cache e Data Dictionary Cache
- Library Cache: área responsável por armazenar o código executado recentemente na sua forma analisada por parse para que ele possa ser reutilizado sem a necessidade de ser reanalisado. Isto melhora muito o desempenho quando são realizadas consultas ao banco de dados.
- Data Dictionary Cache: coleção de tabelas do banco de dados que contêm os metadados (estruturas das tabelas, privilégios, papéis etc)
- Tamanho: parâmetro SHARED_POOL_SIZE (32 MB / 84 MB)

Gerenciamento em Banco de Dados

Shared Pool (Pool compartilhado)

- Tamanho: parâmetro SHARED_POOL_SIZE
- Se igual a 0 então
 - ◆ Se SGA_TARGET setado, então considera tal valor para todo SGA
 - ◆ Se SGA_TARGET não setado
 - ◆ 32 MB para 32 bits
 - ◆ 84 MB para 64 bits
- Alterar tamanho:
 - ◆ alter system set SHARED_POOL_SIZE = 300M scope=spfile;
 - ◆ Restart oracle

Gerenciamento em Banco de Dados

Buffer Cache (Data Buffer Cache)

- Contém blocos de dados que foram lidos recentemente em uma consulta (SELECT) ou que foram alterados ou adicionados através de uma instrução DML (UPDATE, INSERT, etc.)
- A partir do Oracle 10g, o redimensionamento pode ser realizado manual ou automaticamente (conforme a carga de trabalho).
- Mantém os blocos mais lidos em memória, sempre que possível, sempre que houver blocos disponíveis – seguem a lista LRU (LAST RECENTLY USED)

Gerenciamento em Banco de Dados

Buffer Cache (Data Buffer Cache)

Cada bloco pode estar em quatro estados:

- **LIVRE:** Ainda não foi preenchido com algum dado proveniente de disco. No instante imediatamente posterior à abertura do banco, só há blocos livres.
- **OCUPADO:** Já foi preenchido. A ocupação também acontece segundo um algoritmo LRU.
- **SUJO:** Já foi preenchido e alterado, portanto deve ser gravado em disco brevemente.
- **ROLLBACK:** Possui um dado que poderá ser reaproveitado caso a transação que o alterou seja encerrada sem gravação.

Gerenciamento em Banco de Dados

Buffer Cache (Data Buffer Cache)

- Tamanho do bloco: aconselhável que seja múltiplo do tamanho do bloco do sistema operacional. Parâmetro `DB_BLOCK_SIZE`
- Tamanho do Data Buffer Cache. Parâmetro `DB_BLOCK_BUFFERS` (quantidade de blocos que estarão presentes no data buffer cache). Se igual a zero, controle é feito pelo Oracle segundo necessidade e alocação de memória dedicada a SGA. Para conhecer tamanho em kbytes (Tkb) faz-se necessário efetuar a seguinte conta: $Tkb = DB_BLOCK_SIZE * DB_BLOCK_BUFFERS$

Gerenciamento em Banco de Dados

Redo Log Buffer

- Usado para controlar as alterações efetuadas no banco de dados pelo servidor e pelos processos de segundo plano.
- Registra todas as transações 'comitadas' (finalizadas). (marca especial chamada de commit marker)
- Uma transação confirmada não é considerada completa até que os registros do redo log buffer sejam gravados com sucesso nos arquivos redo log.
- Tamanho do redo log buffer: show parameter LOG_BUFFER

Gerenciamento em Banco de Dados

Large Pool

- É uma área opcional utilizada para transações que interagem com mais de um banco de dados. O Large Pool torna disponível grandes blocos de memória para operações que demandam muita memória.
- O tamanho do Large Pool é controlado através do parâmetro dinâmico `LARGE_POOL_SIZE`.

Gerenciamento em Banco de Dados

Java Pool

- Disponibiliza memória para a JVM (Java Virtual Machine) do Oracle. Este recurso é necessário para processamento dos códigos e dados utilizados pelo Java dentro de uma sessão de usuário.
- O tamanho do Java Pool é controlado através do parâmetro dinâmico `JAVA_POOL_SIZE`

Gerenciamento em Banco de Dados

Streams Pool

- Contém estruturas de dados e de controle necessários para gerenciar o compartilhamento de dados em um ambiente distribuído.
- O dimensionamento desta área é controlado através do parâmetro de inicialização `STREAMS_POOL_SIZE`. Caso este parâmetro esteja configurado como zero, será alocada memória para operações de streams a partir do Shared Pool, podendo consumir até 10% deste recurso.

OBS: discorrer sobre banco de dados distribuído

```
SELECT * FROM V$SGA;
```

Gerenciamento em Banco de Dados

Processos de segundo plano (background processes)

PROCESSOS

Pequenos programas que executam tarefas específicas: integração entre as estruturas de memória e os arquivos em disco, conexão ao servidor, etc.

PROCESSO USUÁRIO

Cada vez que ocorre uma conexão, dispara-se um processo usuário executado na estação cliente. Eles têm a função de encaminhar ao servidor as requisições do cliente.

PROCESSO SERVIDOR

Recebe as requisições de processos usuários e as encaminha ao servidor. Podem ser dedicados (um para cada cliente) ou não. Neste caso, implementa-se o recurso MTS (Multi-threaded Server) que permite compartilhar os processos servidores com vários usuários.

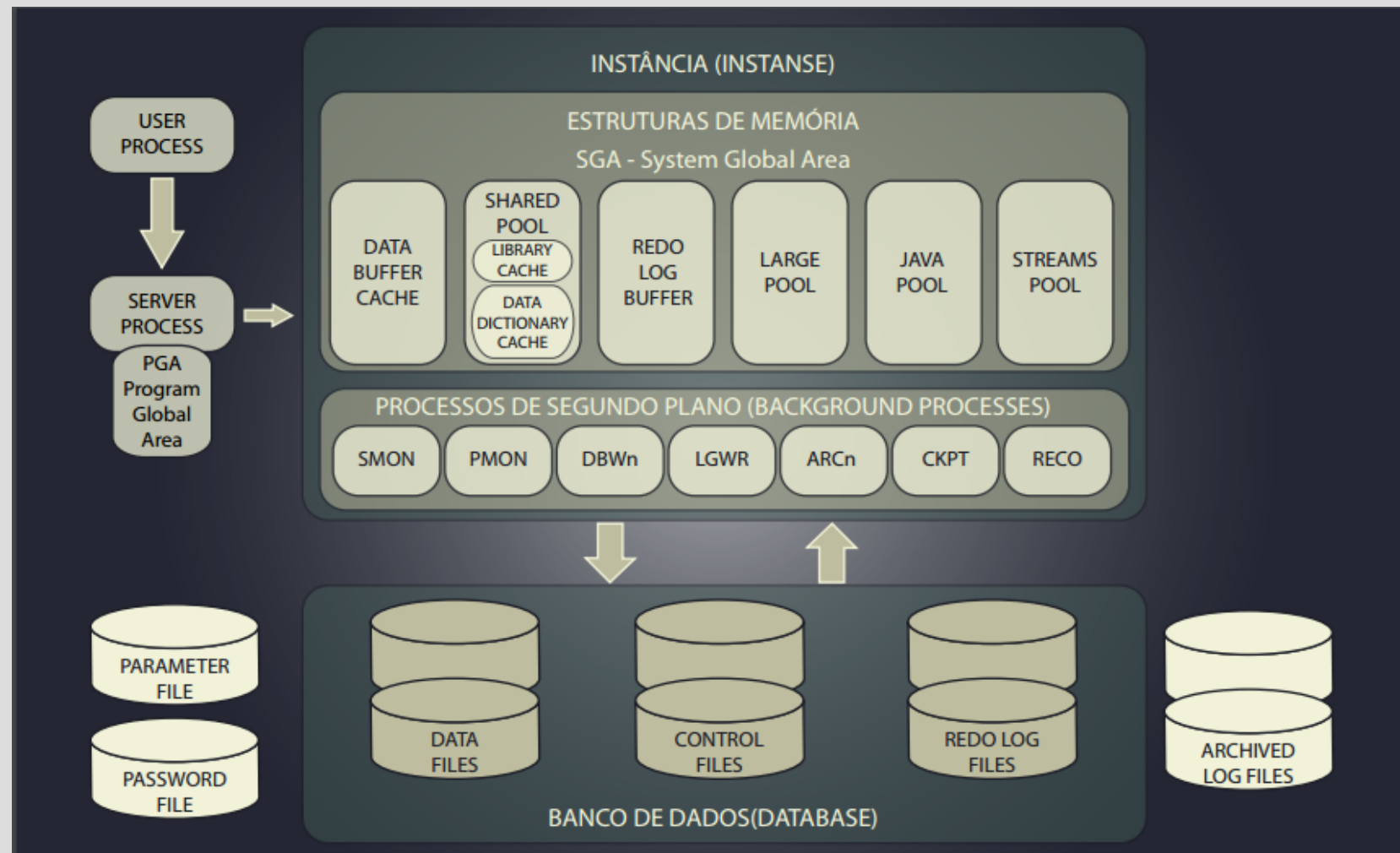
Gerenciamento em Banco de Dados

Processos de segundo plano (background processes)

- Têm como finalidade principal integrar as estruturas de memória aos arquivos em disco.
- Os processos de segundo plano executam tarefas de E/S e monitoram outros processos Oracle, para oferecer maior paralelismo, o que aumenta o desempenho e a confiabilidade.

Gerenciamento em Banco de Dados

SGA



Gerenciamento em Banco de Dados

Processos de segundo plano (background processes)

- Os principais processos de segundo plano são:
- SMON: System Monitor.
- PMON: Process Monitor.
- DBWn: Database Writer.
- LGWR: Log Writer.
- ARCn: Achiver Process.
- CKPT: Checkpoint Process.
- RECO: Recoverer Process.

Gerenciamento em Banco de Dados

SMON: System Monitor

- Verifica a consistência no banco de dados e, se necessário, inicia a recuperação do banco de dados quando ele é aberto (em caso de queda de energia, falha da CPU, por exemplo) a partir dos arquivos de redo log.
- Durante o processo de reinicialização do sistema, são eliminados os segmentos temporários em todos os tablespaces.
- Além disso, como parte de sua rotina, o SMON junta os espaços livres nos tablespaces gerenciados por dicionário

Gerenciamento em Banco de Dados

PMON: Process Monitor

- Ele limpa o data buffer cache e libera outros recursos utilizados pelo usuário quando ocorrer uma falha no processo do usuário em questão
- O Process Monitor também interage com os listeners, fornecendo informações sobre o status da instância para futuras solicitações de conexão

Gerenciamento em Banco de Dados

DBWn: Database Writer

- Grava blocos de dados novos ou alterados (também denominados de blocos "sujos") do Data Buffer Cache para os arquivos de dados (datafiles).
- O parâmetro DB_WRITER_PROCESSES especifica o número de processos DBWn.
- Há quatro circunstâncias que farão o DBWn gravar em disco: excesso de blocos "sujos", ausência de blocos "livres", um tempo limite de três segundos e quando há um checkpoint.

Gerenciamento em Banco de Dados

LGWR: Log Writer

- Grava as alterações registradas no Redo Log Buffer nos Arquivos de Redo Log
- Uma transação somente será considerada bem sucedida quando o LGWR gravar com êxito as informações nos arquivos de redo log
- Se o banco de dados tem um redo log multiplexado, o LGWR grava as entradas de redo log para um grupo de redo log files. Caso um dos arquivos de redo log for danificado, o LGWR gravará nos membros restantes do grupo de redo log files e registrará o erro no log de alertas. Se todos os membros do grupo não puderem ser utilizados, o processo LGWR falhará e a instância ficará suspensa até que seja realizada a correção do problema.

Gerenciamento em Banco de Dados

LGWR: Log Writer

- A gravação no Arquivos de Redo Log ocorrem em 5 situações:
- (1) a cada 3 segundos
- (2) quando 1/3 do buffer estiver cheio
- (3) quando o comando commit for emitido
- (4) quando as entradas de redo no buffer atingir 1 MB
- (5) antes do processo de segundo plano DBWn gravar as alterações do cache de banco de dados nos data files.

Gerenciamento em Banco de Dados

ARCn (Archiver Process)

- Somente para casos onde opera no modo archivelog
- Copiará os redo logs para um ou mais dispositivos de armazenamento (nos arquivos de logs arquivados)
- É importante para o desempenho do banco que o processo de arquivamento termine antes que o redo log preenchido seja necessário novamente. Os usuários não poderão concluir suas transações até as entradas sejam gravadas no redo log file e este não estará pronto para aceitar novas entradas, pois ainda estará sendo gravado em um ou mais dos arquivos de logs arquivados.

Gerenciamento em Banco de Dados

CKPT (Checkpoint Process)

- Define um SCN (System Change Number) no segmento de redo do banco de dados.
- SCN - número em formato atômico, mantido pelo Oracle para registrar quais alterações foram feitas no banco de dados
- Toda vez que um commit é realizado, um novo SCN é gerado, marcando os arquivos para que o Oracle saiba onde e quanto de recuperação deverá ser aplicado em caso de falha. Isto é feito pelo processo CKPT.

Gerenciamento em Banco de Dados

CKPT (Checkpoint Process)

- Os arquivos de controle são fundamentais neste processo, pois informam ao Oracle quais os SCNs corretos. Os checkpoints são registrados no arquivo de controle e em cada cabeçalho dos arquivos de dados (datafiles). Portanto, os checkpoints são um elemento essencial na tarefa de recuperação (recovery) do banco de dados.
- O último checkpoint pode ser verificado na visão: V\$DATABASE.
 - ◆ `SELECT NAME, CHECKPOINT_CHANGE# FROM V$DATABASE;`
 - ◆ `ALTER SYSTEM CHECKPOINT;`
 - ◆ `SELECT NAME, CHECKPOINT_CHANGE# FROM V$DATABASE;`

Gerenciamento em Banco de Dados

RECO (Recoverer Process)

- É um processo de background que resolve transações distribuídas que estão pendentes por causa de uma falha de rede ou sistema em um banco de dados distribuído. Ele utiliza as informações na tabela de transações pendentes para finalizar o status de transações em dúvida. Em determinados intervalos de tempo, o recoverer process local tenta se conectar a bancos de dados remotos e completar automaticamente o commit ou rollback da parte local das transações distribuídas pendentes. Todas as transações automaticamente resolvidas pelo Recoverer Process são removidas da tabela de transações pendentes.

Gerenciamento em Banco de Dados

Processos de segundo plano

- Para ver uma lista completa com os nomes de todos os processos de segundo plano disponíveis em seu banco de dados utilize a seguinte consulta:
- `SELECT NAME FROM V$BGPROCESS;`

Gerenciamento em Banco de Dados

Processando uma consulta (SELECT)

- As consultas (SELECT) são distintas dos outros comandos DML, pois retornam resultados (uma ou mais linhas), enquanto os outros comandos retornam apenas se houve êxito ou não.
- Podemos dividir uma consulta em 3 passos

Gerenciamento em Banco de Dados

Processando uma consulta (SELECT)

- 1. Analisando uma instrução SQL
- Procura cópia da instrução SQL no Shared Pool.
- Valida a instrução SQL (sintaxe).
- Efetua pesquisas no dicionário de dados para validar tabelas e campos (semântica).
- Verifica os privilégios do usuário.
- Determina o plano de execução ideal para a instrução (índices).
- Armazena versão parseada (compilada) do comando (Shared Pool – Library Cache).

Gerenciamento em Banco de Dados

Processando uma consulta (SELECT)

- 2. Executando a instrução SQL
- Neste ponto, o servidor Oracle possui os recursos necessários para a execução dos comandos, no caso de instrução SELECT, é preparado o processo de recuperação de dados. Identifica as linhas para extração.
- Leituras lógicas: blocos que já estão na memória (cache hit).
- Leituras físicas: blocos que não estão na memória. (cache miss)
- ◆ DATA BUFFER CACHE

Gerenciamento em Banco de Dados

Processando uma consulta (SELECT)

- 3. Extraíndo as linhas de uma consulta
- Neste estágio as linhas são selecionadas e ordenadas se necessário e passadas pelo servidor ao usuário. Dependendo do número de linhas podem ser necessários um ou mais processo de extração.

Gerenciamento em Banco de Dados

Processando uma instrução DML (INSERT, UPDATE, DELETE)

- 1. Analisando uma instrução DML (Igual ao processo consulta SQL) .Procura cópia da instrução SQL no Shared Pool.
- Valida a instrução SQL (Sintaxe).
- Efetua pesquisas no dicionário de dados para validar tabelas e campos (Data Dictionary Cache).
- Verifica os privilégios do usuário (Data Dictionary Cache).
- Determina o plano de execução ideal para a instrução (índices).

Gerenciamento em Banco de Dados

Processando uma instrução DML (INSERT, UPDATE, DELETE)

- 2. Executando a instrução DML
- Se não houver blocos de rollback e dados no Data Buffer Cache, o processo de servidor fará sua leitura dos Data Files para o Cache de Buffer.
- Processo de servidor bloqueia as linhas que serão modificadas.
- No Redo Log Buffer, são registradas as alterações a serem feitas.

Gerenciamento em Banco de Dados

Processando uma instrução DML (INSERT, UPDATE, DELETE)

- O Processo de servidor registra a imagem original do bloco de rollback e atualiza o bloco de dados. Essas duas alterações são efetuadas no cache de buffer do banco de dados.
- Qualquer bloco alterado no cache de buffer será marcado como buffer sujo, ou seja, os buffers que não são iguais aos blocos correspondentes no disco.
- LGWR grava as informações para o arquivo de redo log
- DBWn grava os dados no arquivo de dados
- ARCn grava os dados para os arquivos de log arquivados

Gerenciamento em Banco de Dados

Processando uma instrução DML (INSERT, UPDATE, DELETE)

- OBS1: a qualquer momento podemos ter a ação do CKPT e RECO
- OBS2: Os blocos de rollback registram os valores dos dados antes de serem modificados, armazenam de forma que as instruções DML possam ser submetidas a rollback se necessário. As alterações dos blocos de dados registram os novos valores de dados

Gerenciamento em Banco de Dados

Estágios para que o banco de dados esteja no ar

Logar com o papel de sysdba no SQL*PLUS

connect system/uninove as sysdba

connect sys/uninove as sysdba

connect sys/manager as sysdba

SHUTDOWN

- *shutdown*
- Instância fechada
- Banco fechado

Gerenciamento em Banco de Dados

Estágios para que o banco de dados esteja no ar

NOMOUNT

- *startup nomount*
- Instância aberta
- Banco fechado

Ativam-se os processos background e aloca-se memória para a SGA. Após essa operação pode-se criar um Banco de Dados ou consertar Control Files.

Gerenciamento em Banco de Dados

MOUNT

- *alter database mount;*
- Instância aberta
- Banco parcialmente aberto

Lêem-se os Control Files e determina-se a localização dos demais arquivos. Recuperação e mudança da forma de arquivamento (processo ARCH) acontece após essa operação.

OPEN

- *alter database open;*
- Instância aberta
- Banco aberto

Abrem-se os Data Files e Redo Log Files. Usuários que não sejam DBA podem se conectar.