2. Database Architecture

Date: 14.10.2009

Instructor: Sl. Dr. Ing. Ciprian Dobre

ciprian.dobre@cs.pub.ro

Outline

- Logical and Physical Storage Structure
- Application Architecture
- Memory Architecture
- Process Architecture

Logical and Physical Storage Structures

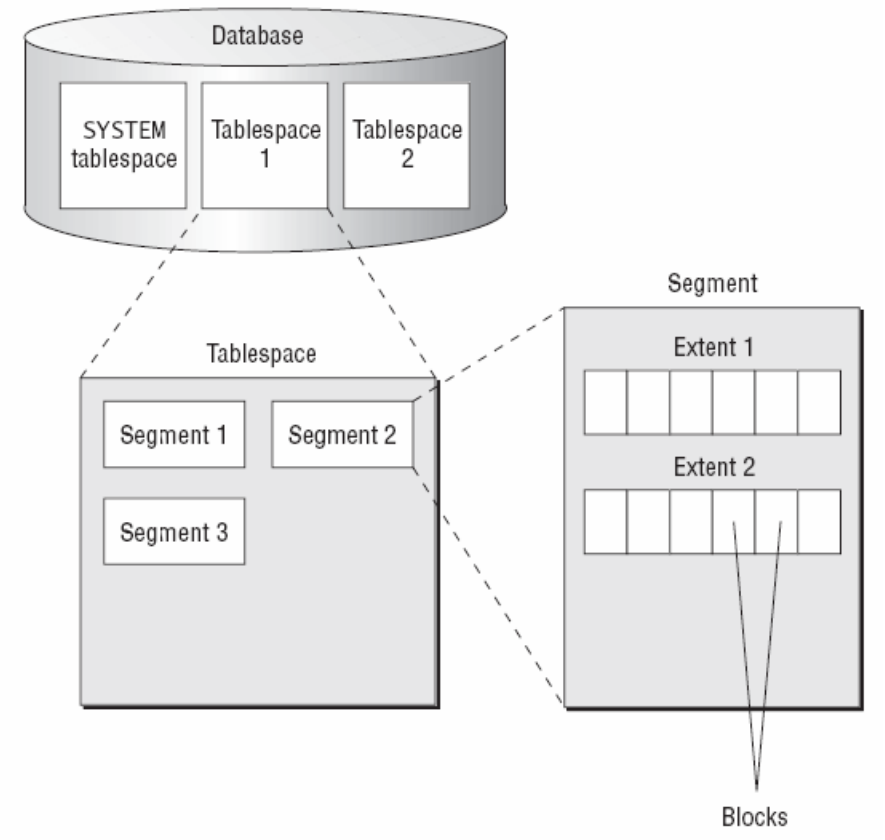
Logical and Physical Storage Structures

- A database system consists of two major components — the ***database*** and the ***instance***
- The **database** = the physical files that store data
- The **instance** = the memory structures and background processes used to access data (from the physical database files)

Logical and Physical Storage Structures

- A database consists of **logical structures** and **physical structures**.
- **Logical structures** = the components seen in the database (such as tables, indexes, and so on)
- **Physical structures** = the method of storage used internally (the physical files).

Logical Storage Structures in Oracle



Logical Storage Structures in Oracle

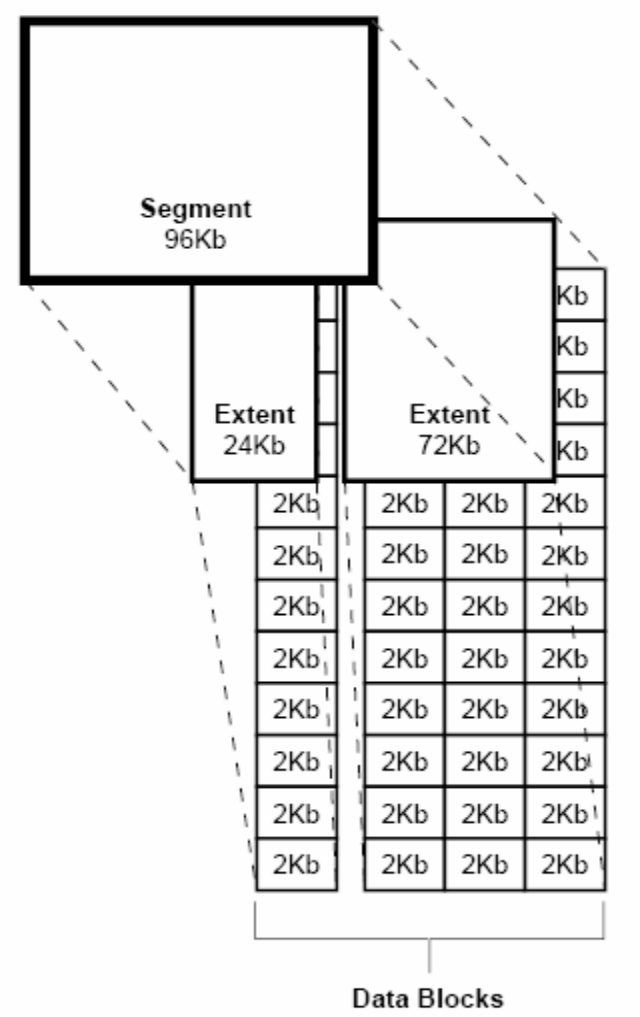
- Tablespace
  - Commonly groups related logical structures together
  - Helps to better administer data related to a particular application for example
  - Minimum – SYSTEM tablespace
- Segment
  - Set of extents allocated to one particular logical structure
  - Associated to only one tablespace
- Extent
  - Grouping of contiguous blocks, allocated in one chunk
- Block
  - Usually a multiple of the OS block size
  - Specified as DB_BLOCK_SIZE

Tablespaces

- Why having tablespaces?
  - Controlling applications
  - Placing logical units on different I/O devices.
  - Controlling and configuring different parameters to each logical unit (file sizes and auto extension, extent sizes etc.)

- System tablespace
  - Holds all system tables. Do not use for other segments
- Temporary tablespaces
  - Used for temporary tasks: Index creation, sorts
- Undo tablespaces
  - Used for placing rollback data

The relationship Among Segments, Extents, and Data Blocks



Oracle Segments

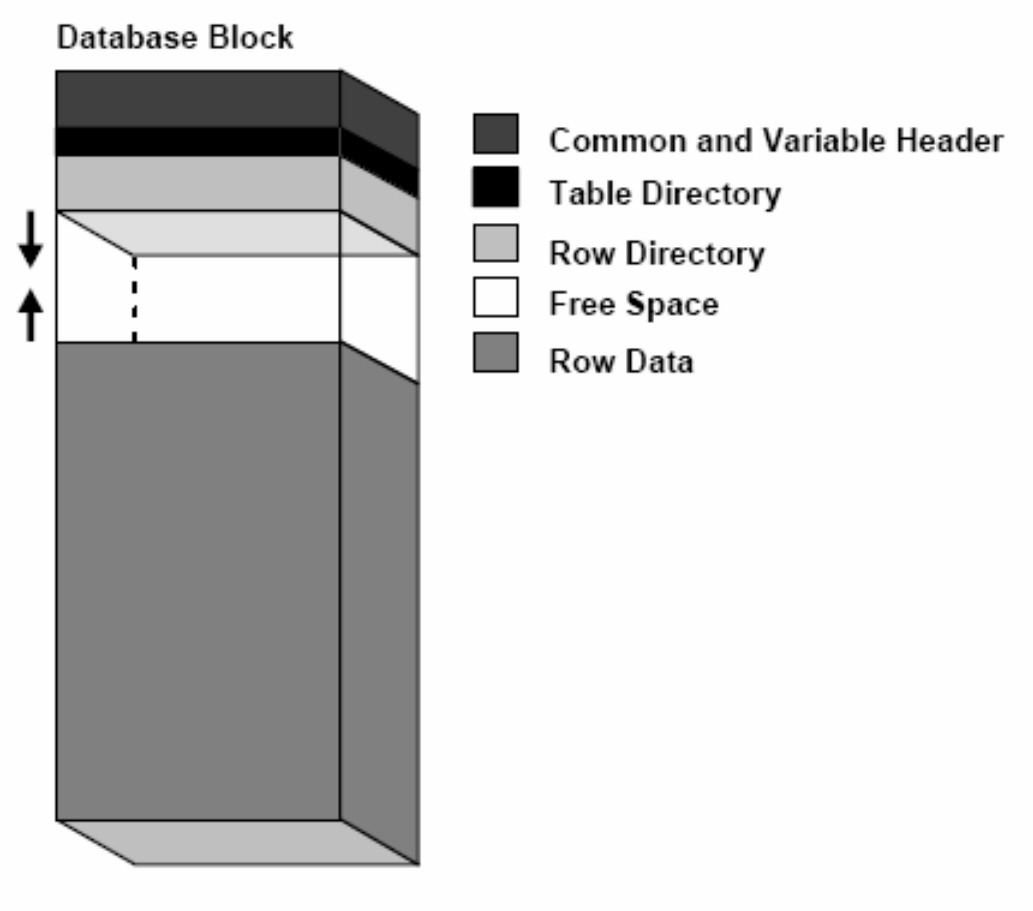
- Any logical entity that requires physical disk space for storing data is referred to as segment
- The main Oracle segments are:
  - Table – Raw data, held in blocks
  - Index – Sorted Data Structures that point to the actual data in the table
  - Partition – Sub-table
  - Rollback Segment

Oracle Extents

- Extent – a physical continuous space inside a single data file
- Extents are allocated according to table storage specifications then tablespace, then datafile

- High water mark – Marks the highest point that data has reached
- Truncate vs. delete – delete deallocates extents until the high water mark, while truncate deallocates all extents and places the HWM on the initial extents

Block Structure



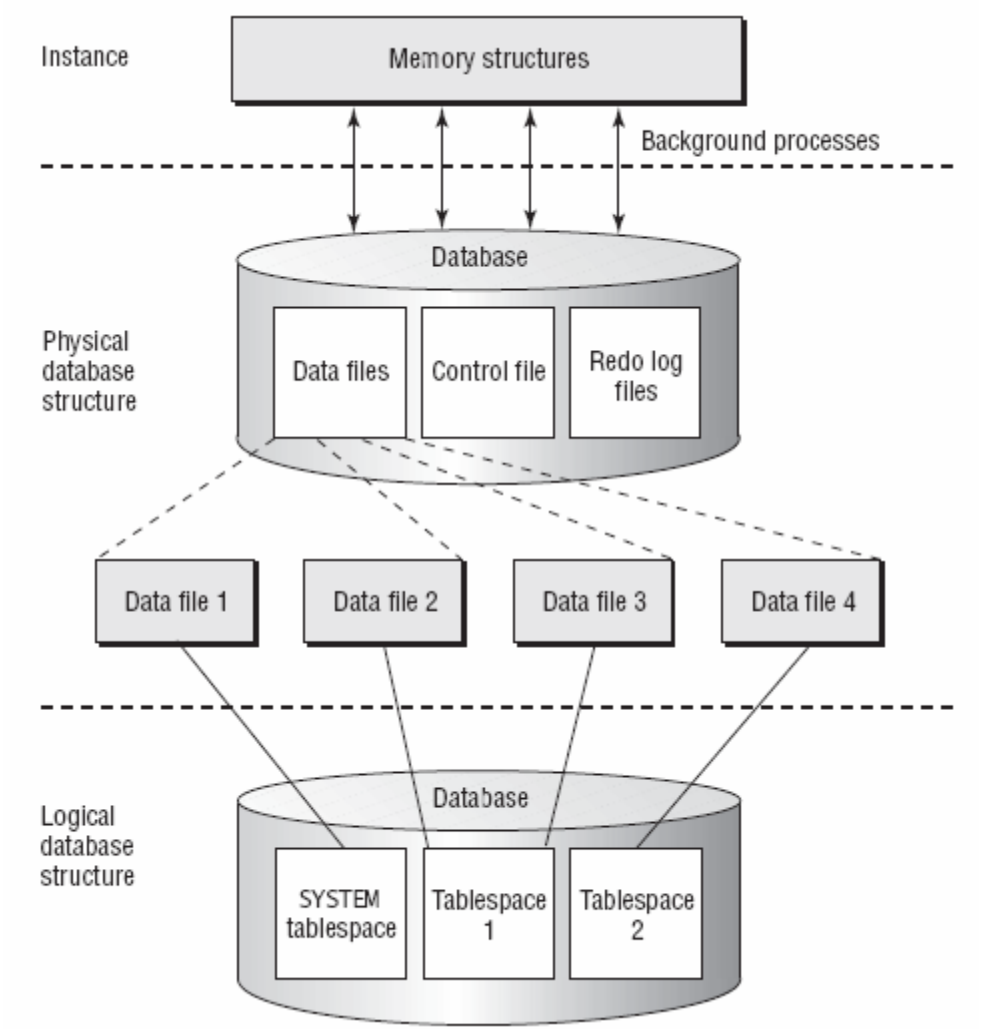
Blocks

- Block header – block address, table and row directory, transaction slots.
- PCTFREE – Space reserved for updates. For high update rate this value should be high to avoid migration and limit the number of records in the block.
  - (Average row size – Initial row size) * 100 / Average row size
- PCTUSED – The block is considered free if usage falls below this percent. For high insert rate this value should be high to decrease the chance a block in the free list will have no space to accommodate a new row
  - 100 – pctfree – (Average row size * 100/ available data space)

Blocks

- Chaining - Happens when a record can not be contained in any block
- Migration – Happens when a record can not be contained in one block

Physical Storage Structures in Oracle



Redo log files

- Cyclic files that store all changes to the database
  - The log writer writes the changes kept in the redo log buffer to the current (active) redo log
  - Must be consistent in order for the database to start
  - May become a performance bottleneck
  - Divided into groups
  - Each group may have one or more members (for backup)
  - At each point in time there is one active group that the log writer is writing sequentially into
  - If archive mode enables the redo log file is being copied to an archive file upon log switch

Note: Don't delete Redo logs

Control files

- Control files
  - Binary files
  - Holds Database info:
    - File layout
    - Data file pointers
    - Last checkpoint
- Usually have few copies as a protection against corruption

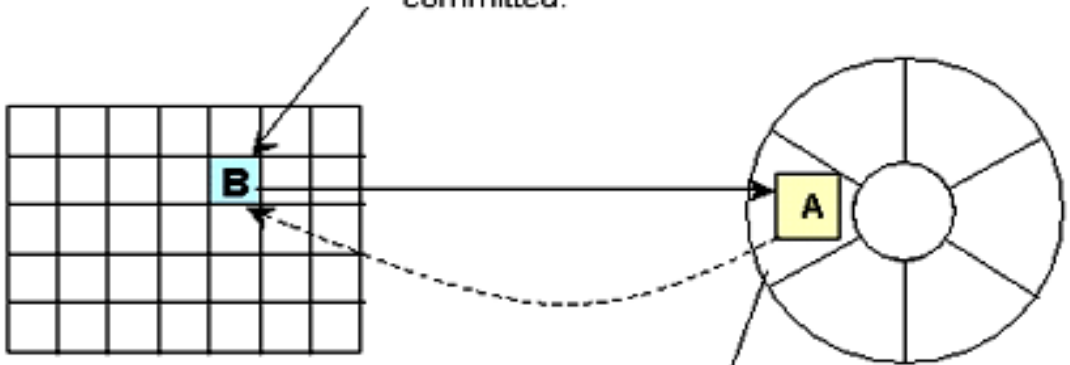
Undo tablespace and rollback segments

- Undo tablespace contain rollback segments
- When a transaction begins, a rollback segment is assigned to it
- Oracle assigns the rollback segment that has the least number of active transactions
- Each transaction write the data to the rollback segment in a circular manner
- Once the all extents were used Oracle will try to use the first extent, if it not yet active
- If it is active new extents will be allocated, if possible

Rollback Segments

Rollback Segment

When data is updated from A to B, B is written to data block and A that is a value before update is written to rollback segment. Data will not be overwritten and will be stored until transaction is committed.



When transaction rollback is committed, data is written back from here.

Rollback segment

If other transactions perform a scan before transaction is committed, a value before update is read from rollback segment.
If committed, rollback segment is released and then overwrite (reuse) will be available.

Temporary segments

- Used for:
  - sorts operation
  - index and table creation
- Each user should be associated with a temporary tablespace
- Data files are called tempfiles and grow on demand

Data files

- Data files holds the actual data
- Data files attributes
  - Size – The initial size of the file
  - Next extent – The next extent that will be allocated
  - Autoextend on/off – Enable/disable extending the datafile
  - Max size – maximal file size
  - Reuse – Reuse existing file
- Why have more than one data file?
  - Performance degradation due to large files
  - Distribute I/O load on several I/O devices

The Checkpoint

- A Checkpoint is a database event which synchronizes the modified data blocks in memory with the data files on disk
- A checkpoint performs the following three operations:
  - Every dirty block in the buffer cache is written to the data files
  - The latest SCN is written (updated) into the data file header
  - The latest SCN is also written to the control files
- Before the checkpoint the redo log buffer is flushed to the disk
- The checkpoint writes all the dirty buffers, including uncommitted data

The Checkpoint

- Checkpoint occurs every:
  - Redo log file switch
  - On checkpoint interval
  - On checkpoint timeout

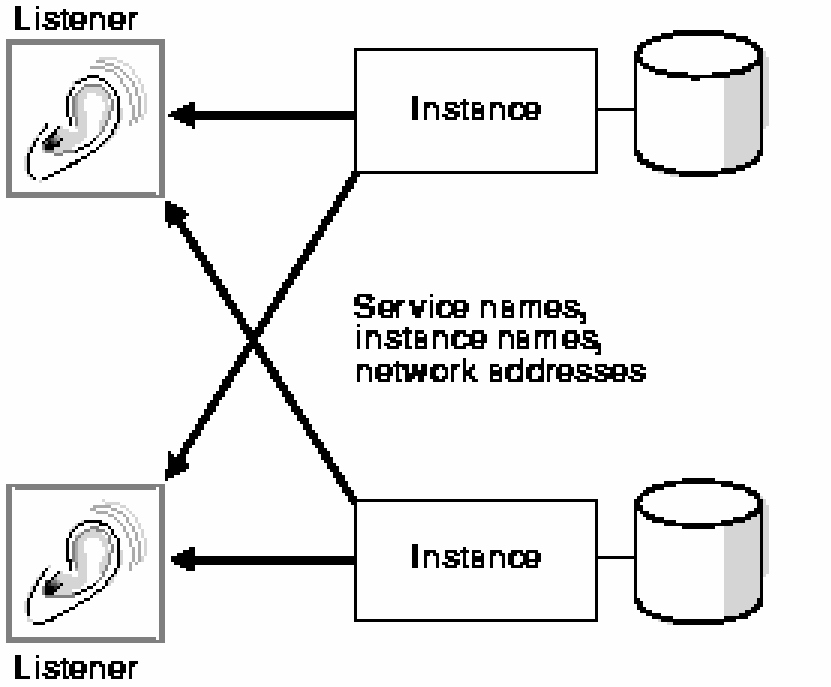
Question: why do we need checkpoints?

The System Change Number

- The SCN is an ever-increasing number
- The SCN is incremented whenever a transaction commits
- Each transaction records the current SCN in the modified block
- At checkpoint the current SCN is recorded in the data files headers and in the control files
- The SCN can be used to determine if a DB is consistent or if a block is consistent

Question: How Oracle uses the SCN to determine consistency?

Opening a DB connection



Opening a DB connection

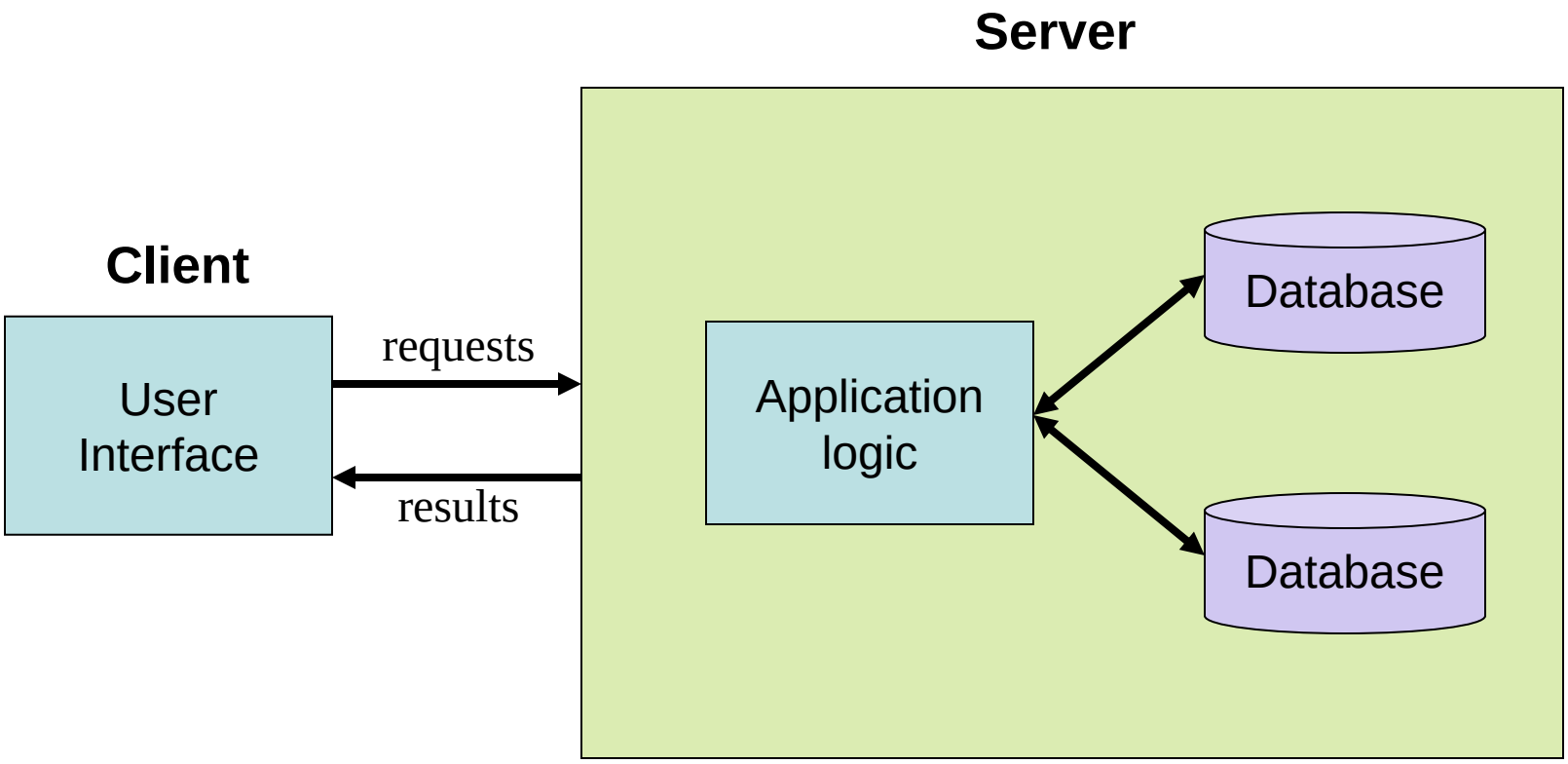
- The client send a session request to the listener
- The listener delegates the request to the relevant instance
- The instance delegate the request to a free server process
- The server process serves all the client's requests

Application Architecture

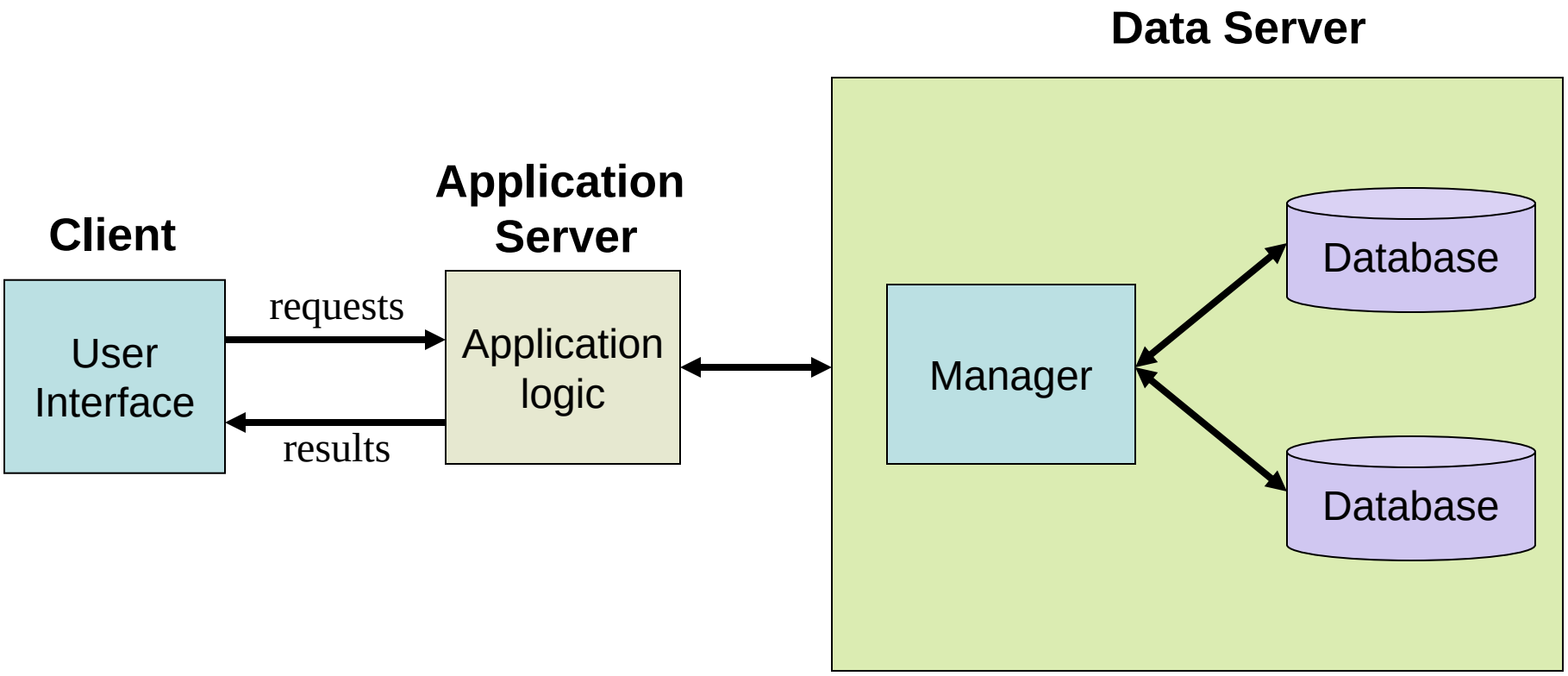
Application Architecture

- Def: The ***software architecture*** of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships between them
- Typical architectures in database design:
  - Two-tier architecture (or client-server architecture)
  - Multi-tier architecture (or n-tier architecture)

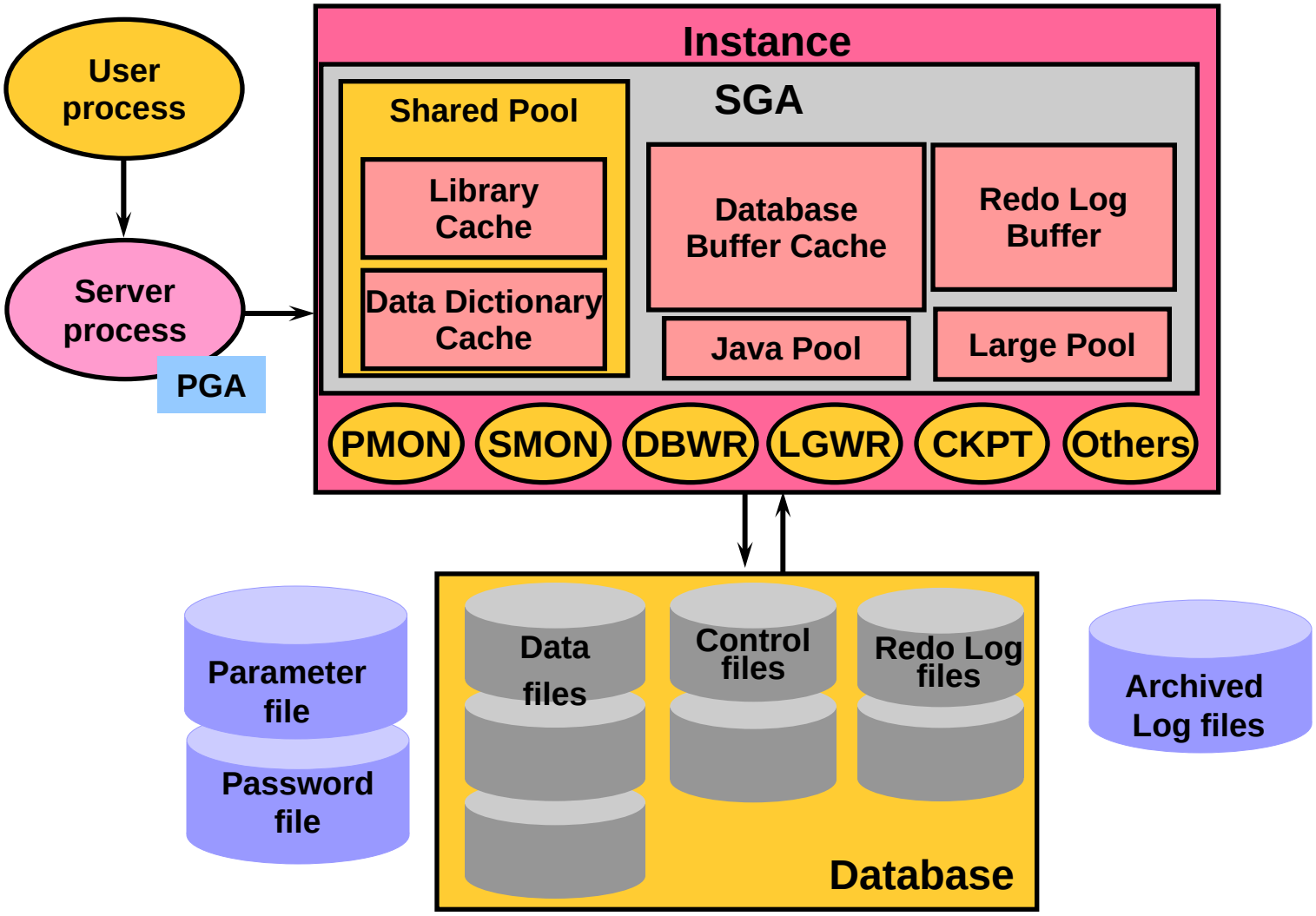
Client-server architecture



Three-tier architecture



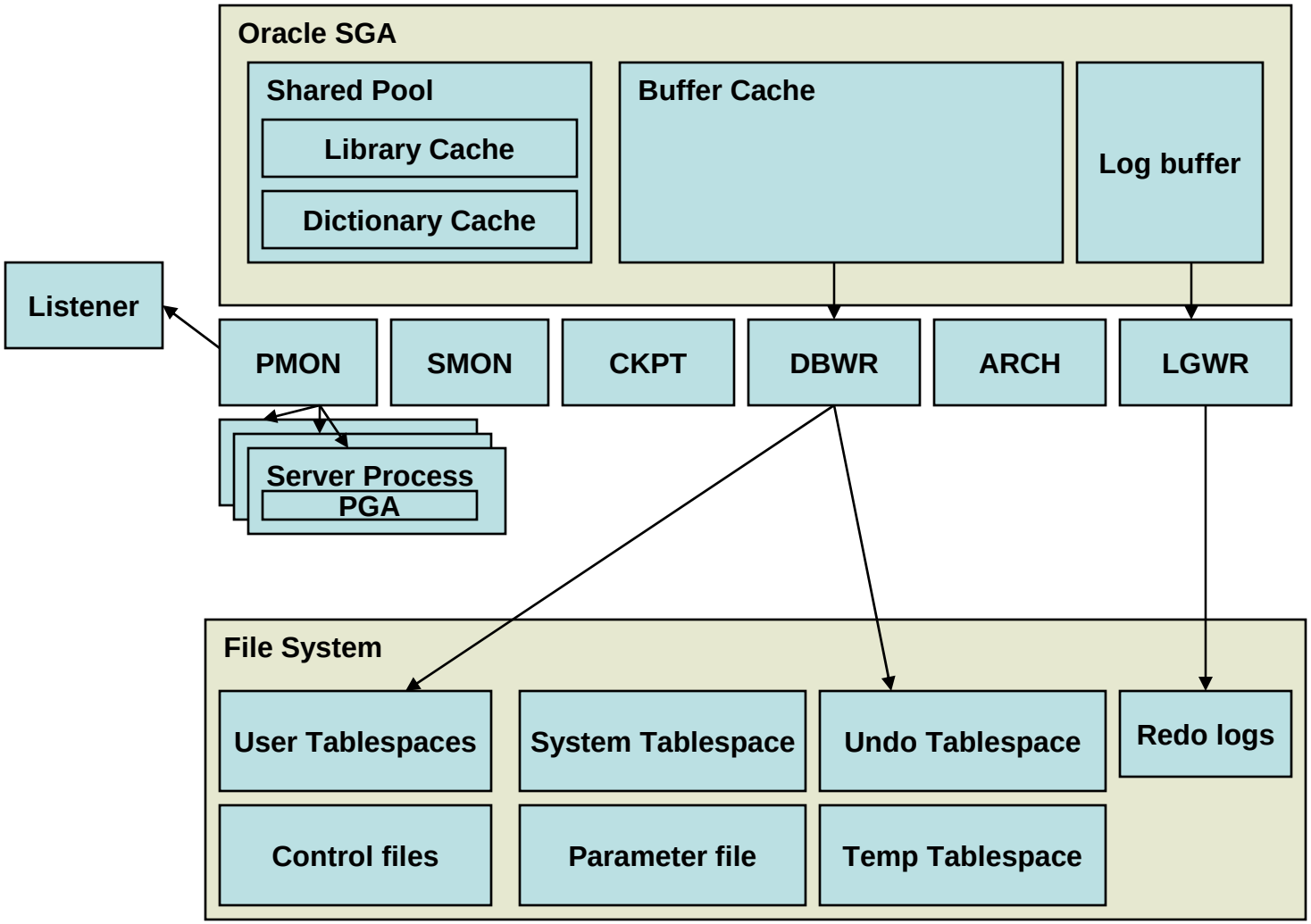
Ex: Oracle Architecture



Video lesson…

Memory Architecture

The Oracle Instance Architecture



System Global Area

System Global Area

- SGA – group of shared memory structures that contain data and control information for one Oracle instance

System Global Area

- Database buffer cache
- Redo log buffer
- Shared pool
- Java pool
- Large pool (optional)
- Data dictionary cache
- Other miscellaneous information

System Global Area

- Fixed SGA:
  - General information about the state of the database and the instance
- Dynamic SGA:
  - Oracle allows dynamically changing the size of:
    - buffer cache,
    - shared pool,
    - large pool,
    - process-private memory

Without shutting down the instance

But only up to SGA_MAX_SIZE

Tips&Tricks

1. For optimal performance the entire SGA should fit in real memory.

- If virtual memory is used to store parts of it, then overall database system performance can decrease dramatically
  - portions of the SGA are paged (written to and read from disk) by the operating system.

2. The amount of memory dedicated to all shared areas in the SGA also has performance impact.

SGA parameters

- The following parameters affect the size of SGA:

Parameter	Description
DB_CACHE_SIZE	The size of the cache of standard blocks.
LOG_BUFFER	The number of bytes allocated for the redo log buffer.
SHARED_POOL_SIZE	The size in bytes of the area devoted to shared SQL and PL/SQL statements.
LARGE_POOL_SIZE	The size of the large pool; default is 0.

Dynamic SGA Granules

- With dynamic SGA, the unit of allocation is called a granule.
- Oracle tracks SGA memory use in integral numbers of granules, for each SGA component.
  - If you specify a size for a component that is not a multiple of granule size, then Oracle rounds the specified size up to the nearest multiple.
  - For example, if the granule size is 4 MB and you specify DB_CACHE_SIZE as 10 MB, you will actually be allocated 12 MB.

Dynamic SGA Granules

- The size of a granule is determined by the total SGA size.
  - On most platforms, the size of a granule is 4 MB if the total SGA size is less than 128 MB, and it is 16 MB for larger SGAs.
  - Some platform-dependency: on 32-bit Windows NT, the granule size is 8 MB for SGAs larger than 128 MB.
- The granule size that is currently being used for SGA can be viewed in the view V$SGA_DYNAMIC_COMPONENTS.
- The same granule size is used for all dynamic components in the SGA.

Dynamic SGA Granules

- Oracle keeps information about the components and their granules in a *scoreboard*.
  - For each component that owns granules, the scoreboard contains:
    - the number of granules allocated to the component,
    - any pending operations against this component,
    - the target size in granules,
    - the progress made toward the target size.
- Oracle also maintains the initial number of granules and the maximum number of granules for each component.
- The information about a granule is stored in a corresponding *granule entry*.

Database Buffer Cache

- The portion of the SGA that holds copies of data blocks read from data files.
- All user processes concurrently connected to the instance share access to the database buffer cache.

Organization of the Database Buffer Cache

- The buffers in the cache are organized in two lists:
  - the write list
  - the least recently used (LRU) list.
- The **write list** holds dirty buffers*:*
  - Contain data that has been modified but has not yet been written to disk.
- The **LRU list** holds:
  - **Free buffers -** do not contain any useful data and are available for use.
  - **Pinned buffers -** are currently being accessed.
  - **Dirty buffers** - have not yet been moved to the write list.

Operations on the Database Buffer Cache

- The first time an Oracle user process requires a particular piece of data, it searches for the data in the database buffer cache.
- If the process finds the data already in the cache (a **cache hit**), it can read the data directly from memory.
- If the process cannot find the data in the cache (a **cache miss**), it copy the data block from a data-file on disk into a buffer in the cache before accessing the data.

- Accessing data through a cache hit is faster than data access through a cache miss.

Operations on the Database Buffer Cache

- Before reading a data block into the cache, a process first find a free buffer.
  - The process searches the LRU list, starting at the least recently used end of the list.
  - The process searches either until it finds a free buffer or until it has searched the threshold limit of buffers.
- If the user process finds a dirty buffer as it searches the LRU list, it moves that buffer to the write list and continues to search.
- When the process finds a free buffer, it reads the data block from disk into the buffer and moves the buffer to the MRU end of the LRU list.
- If an Oracle user process searches the threshold limit of buffers without finding a free buffer, the process stops searching the LRU list and signals the DBW0 background process to write some of the dirty buffers to disk.

The data dictionary

- Oracle holds the structure of the database in a set of internal tables and views
- When we run an SQL statement against the DB Oracle uses the data dictionary to validate our statement

The Data Dictionary

- Data dictionary views
  - all, user, dba
  - dba_tables – All tables
  - dba_indexes – All indexes,
  - dba_tab_columns – All columns for each table
  - dba_ind_columns – All columns for each index
  - dba_tab_partitions – All partitions for each table
  - dba_ind_partitions – All partitions for each index
  - dba_users – All users,
  - dba_data_files – All data files for each tablespace
  - dba_tablespaces – All tablespaces
  - dba_rollback_segs – All rollback segments

Redo Log Buffer

- Circular buffer in the SGA that holds information about changes made to the database.
  - This information is stored in **redo entries**.
- Redo entries contain the information necessary to reconstruct, or redo, changes made to the database by INSERT, UPDATE, DELETE, CREATE, ALTER, or DROP operations.
- Redo entries are used for database recovery, if necessary.

Shared Pool

- The shared pool portion of the SGA contains three major areas:
  - library cache,
  - dictionary cache,
  - buffers for parallel execution messages and control structures.

- The total size of the shared pool is determined by the initialization parameter SHARED_POOL_SIZE.
  - The default value of this parameter is 8MB on 32-bit platforms and 64MB on 64-bit platforms.

Library cache

- The library cache includes:
  - Shared SQL areas
  - Private SQL areas (in case of a multiple transaction server)
  - PL/SQL procedures and packages
  - Control structures such as locks and library cache handles.
- Shared SQL areas are accessible to all users, so the library cache is contained in the shared pool within the SGA.

PL/SQL Program Units and the Shared Pool

- To process PL/SQL program units (procedures, functions, packages, anonymous blocks, and database triggers) Oracle allocates a shared area to hold the parsed, compiled form of a program unit.
  - It allocates a private area to hold values specific to the session that runs the program unit, including local, global, and package variables (also known as package instantiation) and buffers for executing SQL.

Large Pool

- The DBA can configure an optional memory area called the **large pool** to provide large memory allocations for:
  - Session memory for the shared server and the Oracle XA interface (used where transactions interact with more than one database)
  - I/O server processes
  - Oracle backup and restore operations
  - Parallel execution message buffers, if the initialization parameter PARALLEL_AUTOMATIC_TUNING is set to true (otherwise, these buffers are allocated to the shared pool)

Program Global Areas (PGA)

Program Global Area

- A **program global area (PGA)** is a memory region which contains data and control information for a server process.
- A non-shared memory created by Oracle when a server process is started.
  - Access to it is exclusive to that server process and is read and written only by Oracle code acting on behalf of it.
- The total PGA memory allocated by each server process attached to an Oracle instance is also referred to as the **aggregated PGA memory** allocated by the instance.

Program Global Area

- The content of the PGA memory varies, depending on whether the instance is running the shared server option or not.
- Generally speaking, the PGA memory can be classified as follows:
  - Private SQL Area
  - Cursors and SQL Areas
  - Session memory

Oracle processes

Introduction to Processes

- All connected Oracle users must run two modules of code to access an Oracle database instance:
  - Application or Oracle tool: A database user runs a database application (such as Oracle Forms) or an Oracle tool (such as SQL*Plus), which issues SQL statements to an Oracle database.
  - Oracle server code: Each user has some Oracle server code executing on his or her behalf, which interprets and processes the application's SQL statements.

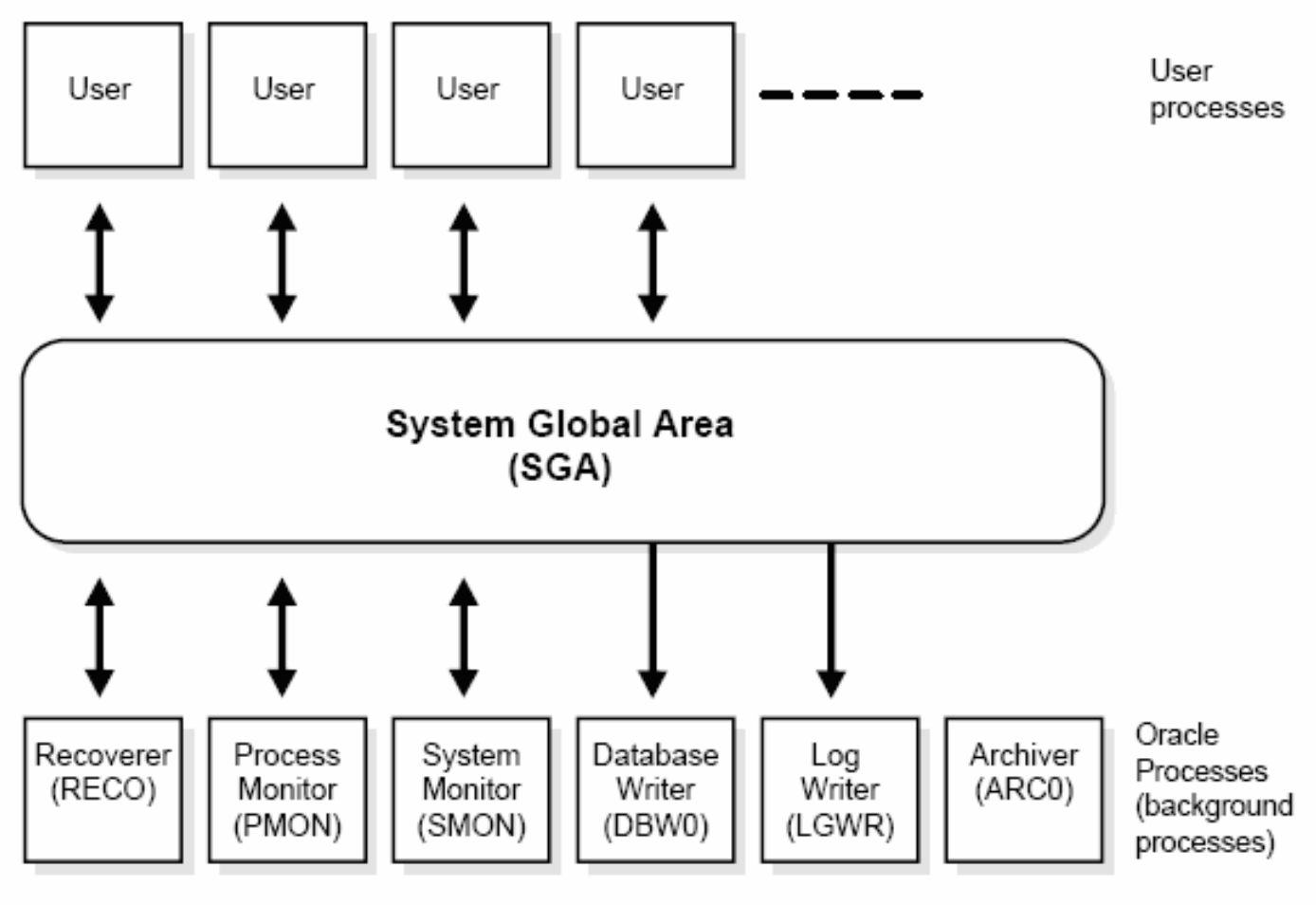
Multiple-Process Oracle Systems

- **Multiple-process Oracle** (also called **multi-user Oracle**) uses several processes to run different parts of the Oracle code and additional processes for the users
  - One process for each connected user and more processes shared by multiple users.

- Most database systems are multi-user, because one of the primary benefits of a database is managing data needed by multiple users at the same time.

- Each process in an Oracle instance performs a specific job.
  - By dividing the work of Oracle and database applications into several processes, multiple users and applications can connect to a single database instance simultaneously while the system maintains excellent performance.

Oracle Instance



Types of Processes

- The processes in an Oracle system can be categorized into:
- User processes run the application or Oracle tool code.
- Oracle processes run the Oracle server code. They include:
  - server processes
  - background processes

User Processes Overview

- When a user runs an application program or an Oracle tool (such as Enterprise Manager or SQL*Plus), Oracle creates a **user process** to run the user's application.

Connections and Sessions

- A **connection** is a communication pathway between a user process and an Oracle instance.
  - A communication pathway is established using available inter-process communication mechanisms (on a computer that runs both the user process and Oracle) or network software (when different computers run the database application and Oracle, and communicate through a network).
- A **session** is a specific connection of a user to an Oracle instance through a user process.
  - For example, when a user starts SQL*Plus, the user must provide a valid username and password, and then a session is established for that user. A session lasts from the time the user connects until the time the user disconnects or exits the database application.

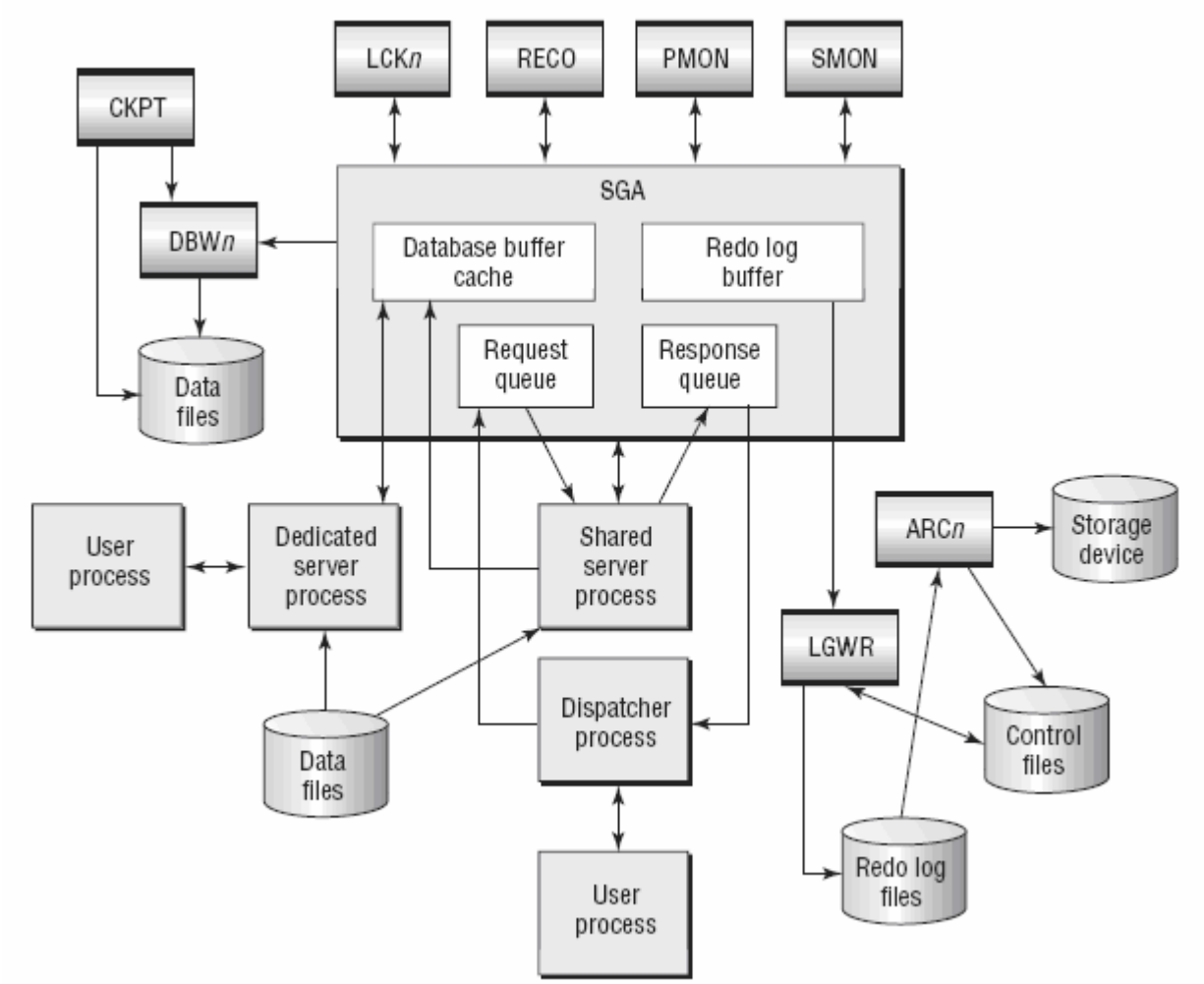
Server Processes

- Oracle creates **server processes** to handle the requests of user processes connected to the instance.
  - In some situations when the application and Oracle operate on the same machine, it is possible to combine the user process and corresponding server process into a single process to reduce system overhead. However, when the application and Oracle operate on different machines, a user process always communicates with Oracle through a separate server process.
- Server processes (or the server portion of combined user/server processes) created on behalf of each user's application can perform one or more of the following:
  - Parse and run SQL statements issued through the application
  - Read necessary data blocks from data files on disk into the shared database buffers of the SGA, if the blocks are not already present in the SGA
  - Return results in such a way that the application can process the information

Background processes

- The background processes in an Oracle instance include the following:
  - DatabaseWriter (DBW0 or DBW*n*)
  - Log Writer (LGWR)
  - Checkpoint (CKPT)
  - System Monitor (SMON)
  - Process Monitor (PMON)
  - Archiver (ARC*n*)
  - Recoverer (RECO)
  - Lock Manager Server (LMS) - Real Application Clusters only
  - Queue Monitor (QMN*n*)
  - Dispatcher (D*nnn*)
  - Server (S*nnn*)

Oracle background Processes



Oracle background processes

- The background processes in an Oracle instance include the following:
  - DatabaseWriter (DBW0 or DBW*n*)
  - Log Writer (LGWR)
  - Checkpoint (CKPT)
  - System Monitor (SMON)
  - Process Monitor (PMON)
  - Archiver (ARC*n*)
  - Recoverer (RECO)
  - Lock Manager Server (LMS) - Real Application Clusters only
  - Queue Monitor (QMN*n*)
  - Dispatcher (D*nnn*)
  - Server (S*nnn*)

Oracle processes and services

- Listener
  - Listens on a designated port for incoming connection requests
- PMON – Process monitor
  - In charge of releasing process resources
- SMON – System monitor
  - In charge of recovery and database consistency
- RECO - Recoverer process
  - Used with the distributed database configuration that automatically resolves failures involving distributed transactions

Database Writer Process

- The **database writer process** **(DBWn)** - in charge of writing data from the buffer cache to the data files and rollback segments

Database Writer Process

- One database writer process (DBW0) is adequate for most systems
- One can configure additional processes (DBW1 through DBW9) and

  (DBWa through DBWj)

  to improve write performance if the system modifies data heavily

Database Writer Process

- The DBW*n* process writes dirty buffers to disk under the following conditions:
  - When a server process cannot find a clean reusable buffer after scanning a threshold number of buffers, it signals DBW*n* to write. DBW*n* writes dirty buffers to disk asynchronously while performing other processing.
  - DBW*n* periodically writes buffers to advance the **checkpoint**, which is the position in the redo thread (log) from which instance recovery begins. This log position is determined by the oldest dirty buffer in the buffer cache.

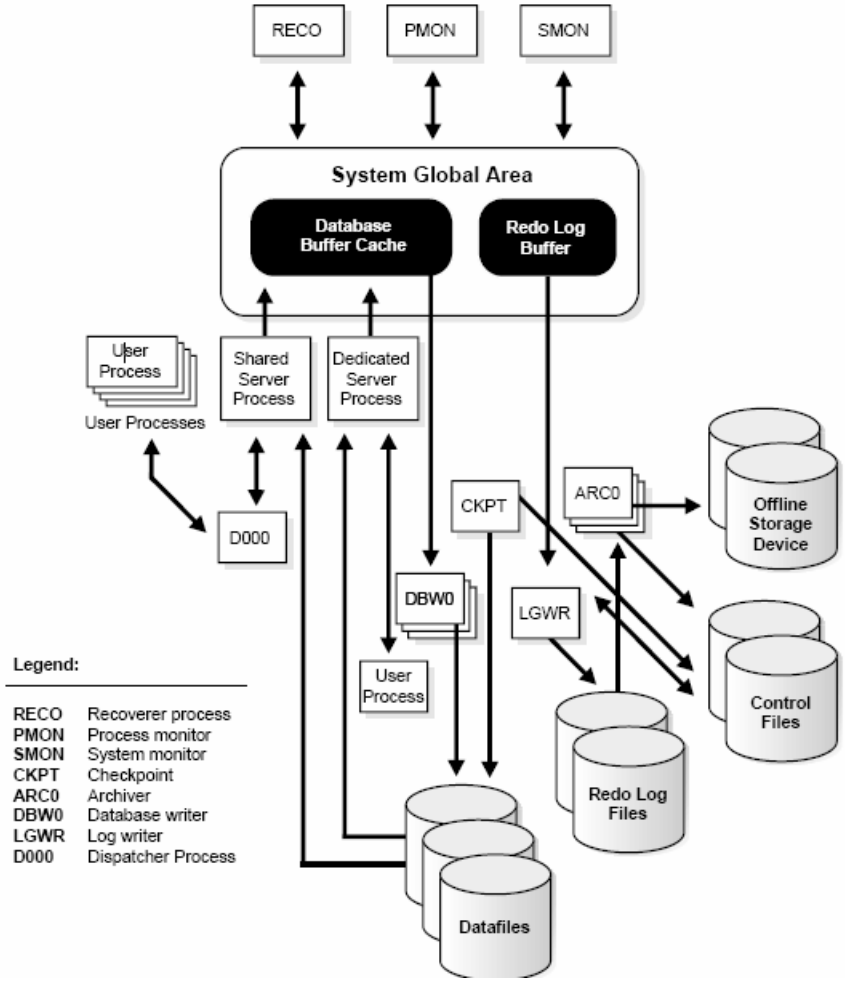
Log Writer Process

- The **log writer process (LGWR)** is responsible for redo log buffer management—writing the redo log buffer to a redo log file on disk.
- LGWR writes all redo entries that have been copied into the buffer since the last time it wrote.

Log Writer Process

- LGWR writes one contiguous portion of the buffer to disk. LGWR writes:
  - A commit record when a user process commits a transaction
  - Redo log buffers
    - Every three seconds
    - When the redo log buffer is one-third full
    - When a DBW*n* process writes modified buffers to disk, if necessary

The Background Processes of a Multiple-Process Oracle Instance



Shared Server Architecture

- **Shared server** architecture eliminates the need for a dedicated server process for each connection.
- A dispatcher directs multiple incoming network session requests to a pool of shared server processes.
- An idle shared server process from a shared pool of server processes picks up a request from a common queue, which means a small number of shared servers can perform the same amount of processing as many dedicated servers.
  - Because the amount of memory required for each user is relatively small, less memory and process management are required, and more users can be supported.

Instance recovery

- Performed when DB had crashed or was aborted
- If the SCN in the header of the data files is larger than current SCN a recovery is required
- Recovery procedure:
  - Find the last checkpoint in the current redo log
  - Redo all committed transactions
  - Undo all uncommitted transaction using the rollback segment

Parameter file

- Holds all Database parameters
  - The first file that is being accessed when the Oracle DB starts up
  - There are few parameters that are mandatory (control files, block size and rollback segments)
  - All the rest of the parameters have default value that can be overridden by placing other value in the parameter file
  - If the file is not syntactically correct the database will not start
  - The file can be textual – init.ora file or binary - spfile

Users and schema

- Each Oracle user has its own DB schema
  - Tables
  - Indexes
  - Views
  - Packages/stores procedures
- Sys as sysdba – used for database administration
- Sys – used for data dictionary creation, system wide changes and granting roles for users (password: change_on_install)
- System – used for administrating users, tablespaces (password: manager)

Performing SQL statements

- How does Oracle executes a select statement?
  - User process sends the query to the server process
  - The server process parse the query – soft parse if the query is in the shared pool and hard parse if not
  - The server validates the query against the data dictionary
  - The optimizer decides on an execution plan
  - Fetch phase – fetching of data is done in ‘windows’ depending on the amount of data being fetched

Performing a DML statement

- How Oracle executes an update statement:
  - The user process sends the query to the server process
  - The server process parses the query – soft parse if the query is in the shared pool and hard parse if not
  - The server validates the query against the data dictionary
  - The optimizer decides on an execution plan
  - The server process reads the data and rollback block from the data files if they are not already in the buffer cache
  - The server process places locks on the data (per record)
  - The server process records the old values (before image) and new values (after image) to the log buffer
  - The server process records the before image to a rollback segment in the buffer cache and the after image to data block in the buffer cache. Both blocks are marked as dirty

Performing commit statement

- How Oracle executes a commit statement
  - The server process places a commit record to the log buffer along with the SCN (system change number)
  - LGWR performs a contiguous write to all redo entries up to and including the commit record
  - This is how Oracle can insure that no data will be lost in case of failure

Note: rolling back a transaction does not flush the log buffer since Oracle always rolls back transaction when it loads

Putting it all together