Configuring Databases for Shared Server

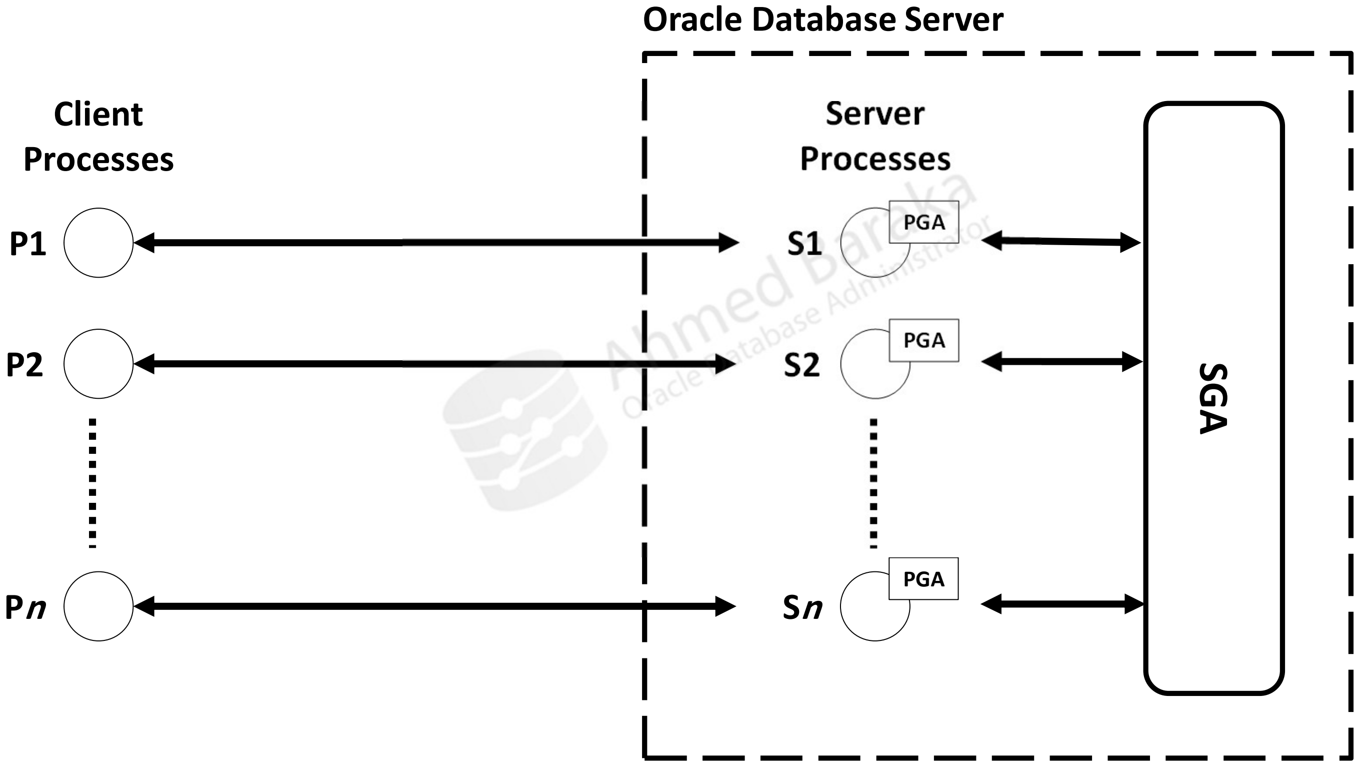
By Ahmed Baraka

Objectives

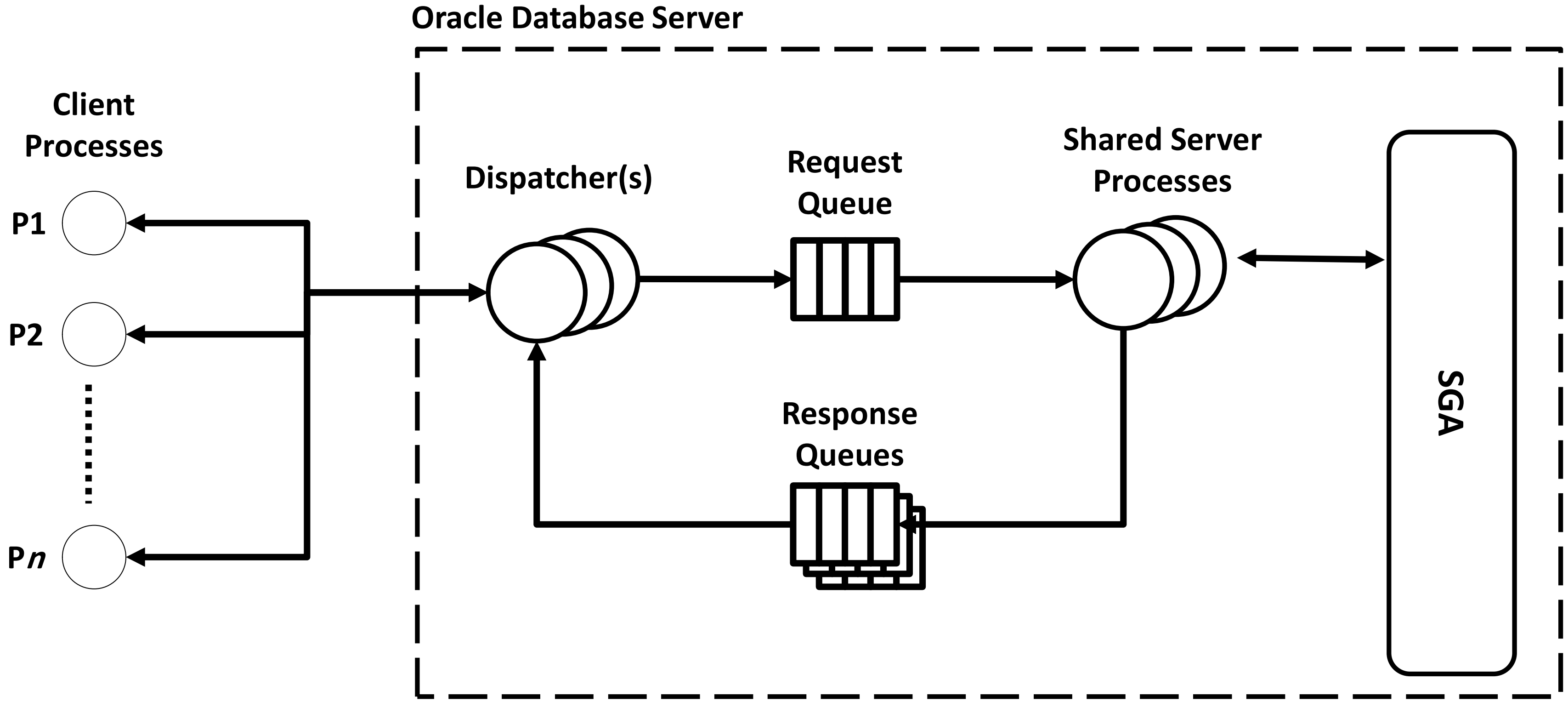
In this lecture, you will learn how to perform the following:

- Describe the shared server architecture and its benefits
- Configure the shared server
- Setup the dispatcher configurations
- Configure Clients for shared and dedicated servers

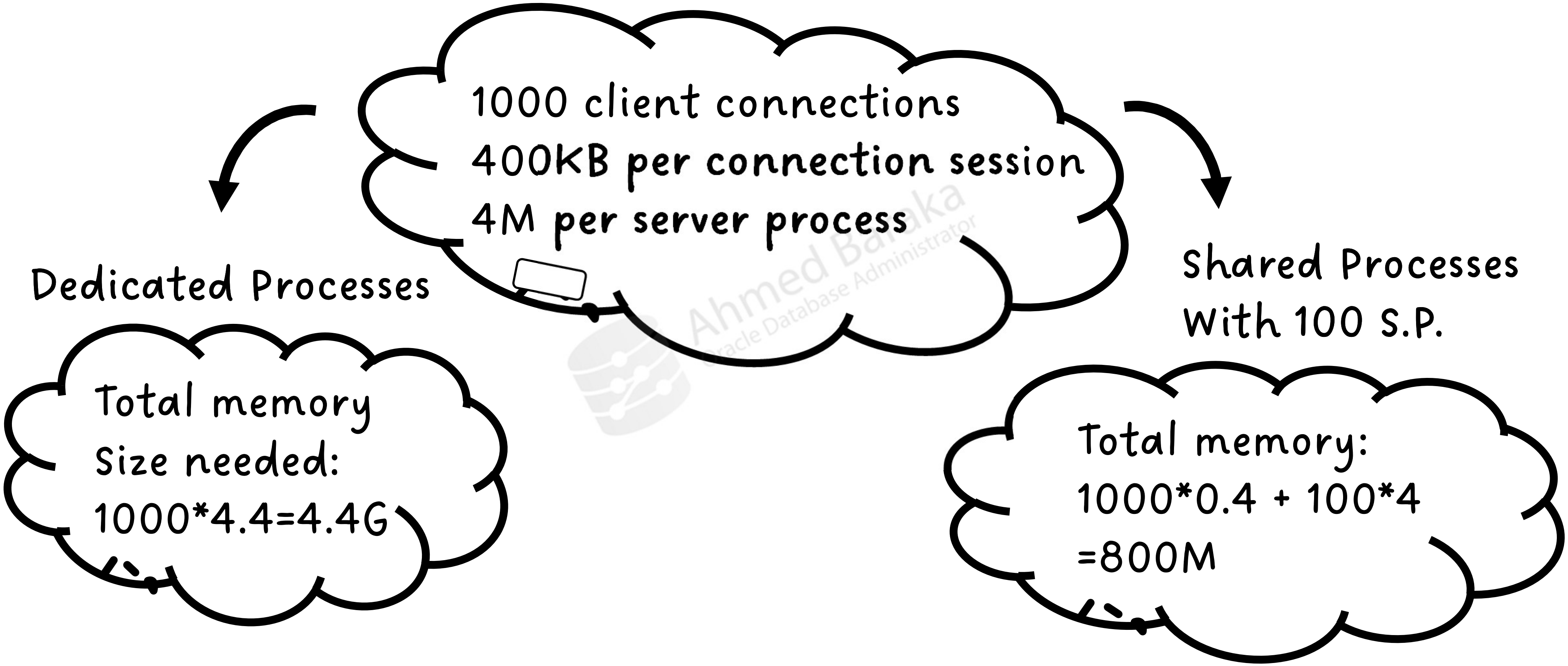
Oracle Database Dedicated Server Processes



Oracle Database Shared Server Processes



Comparison Example



1000 client connections
400KB per connection session
4M per server process

Dedicated Processes

Total memory
Size needed:
1000*4.4=4.4G

Shared Processes
With 100 S.P.

Total memory:
1000*0.4 + 100*4
=800M

About Shared Server Process Configuration

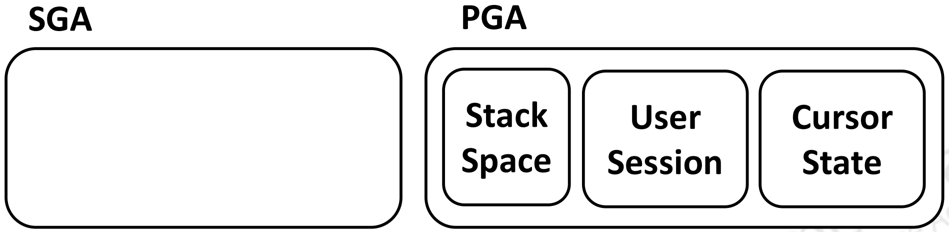
- A shared server process can service multiple user processes.
- A small pool of server processes serve large number of clients
- How it works:
  1. A client process connects to a **dispatcher**. A dispatcher can serve multiple client processes.
  2. A piece of shared memory (called **circuits**) is dedicated to the client process and save the send and receive requests in it
  3. The dispatcher places a virtual circuit on a common **queue** when a request arrives.
  4. An idle **shared server process** picks up the virtual circuit from the common queue, services the request, and relinquishes the virtual circuit.

When to use Shared Server Configuration?

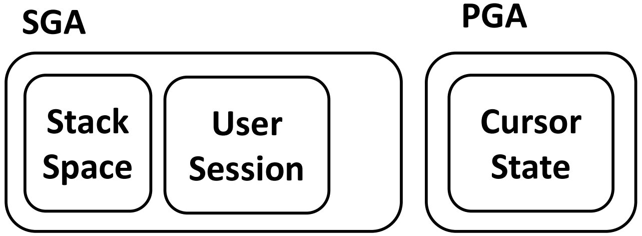
- **Benefits**:
  - Reduces the number of processes on the operating system
  - Reduces instance PGA memory
  - Increases the number of concurrent clients
  - May be faster than dedicated server when the rate of client connections and disconnections is high
- **Disadvantage**: slower response time in some cases and increased complexity for setup and tuning.
- Consider using shared server configuration when the server resources are not sufficient for the required number of concurrent users
- The database can still accept dedicated server connection requests

Memory for User Session Data

- **Dedicated server processes:**



- **Shared server processes:** user session saved in Shared Pool or Large Pool



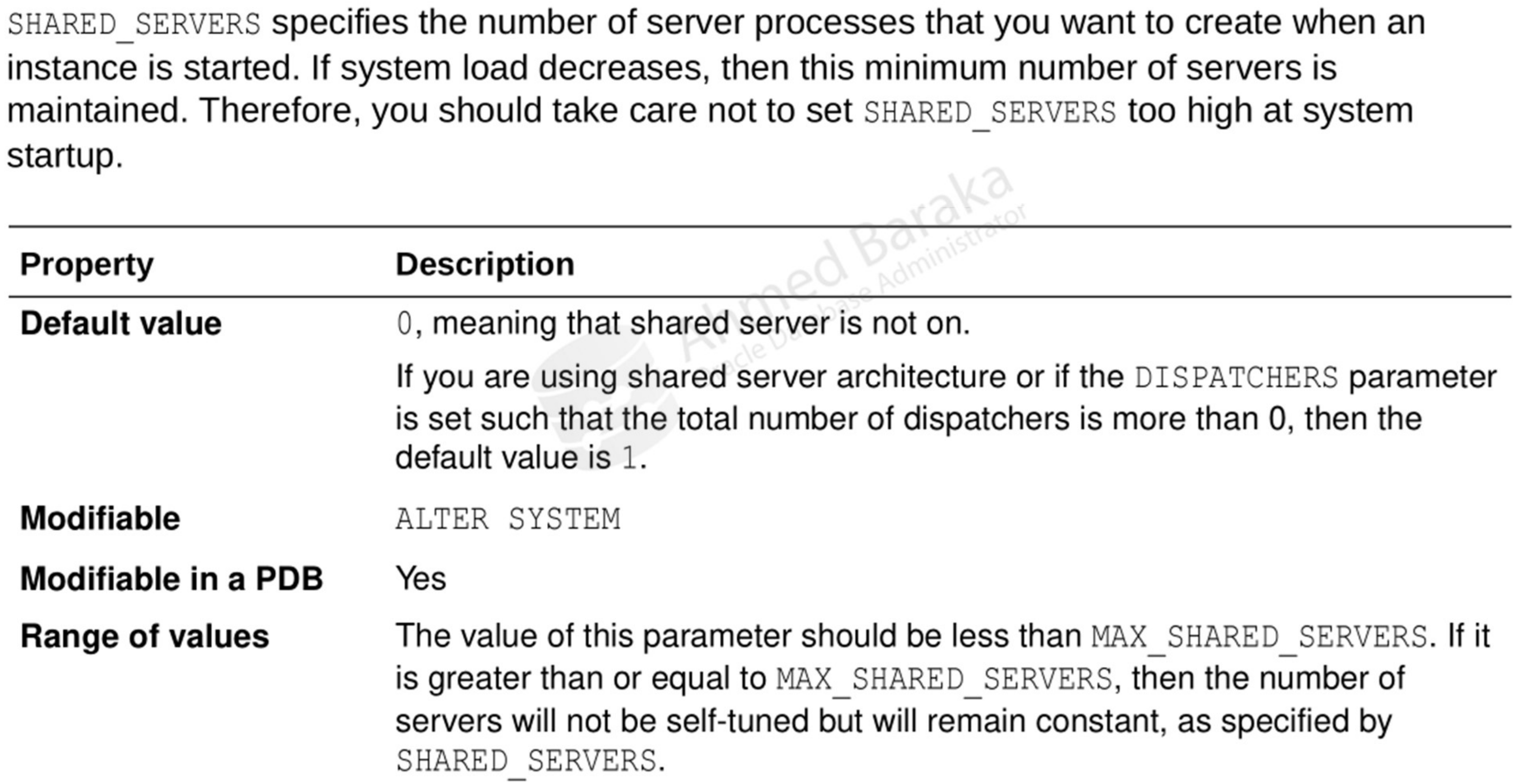
Comparison between Dedicated Server and Shared Server Configurations

	Dedicated Server	Shared Server
Concept	a server process dedicated to each client process	a shared process can serve multiple client processes
Resource utilization	less resource utilization	more resource utilization because the resources are shared
Implementation scenarios	- OLTP systems where the resources are sufficient for all - Warehouse databases	- OLTP systems where the resources are less than the total number of concurrent users - High-think applications
Session memory location	PGA	SGA

Configuring Shared Server

- Required parameters
  - **`SHARED_SERVERS`**
  - **`DISPATCHERS`**
- Optional parameters:
  - **`MAX_DISPATCHERS`**
  - **`MAX_SHARED_SERVERS`**
  - **`CIRCUITS`**
  - **`SHARED_SERVER_SESSIONS`**

About SHARED_SERVERS Parameter



SHARED_SERVERS specifies the number of server processes that you want to create when an instance is started. If system load decreases, then this minimum number of servers is maintained. Therefore, you should take care not to set SHARED_SERVERS too high at system startup.

Property	Description
Default value	0, meaning that shared server is not on. If you are using shared server architecture or if the DISPATCHERS parameter is set such that the total number of dispatchers is more than 0, then the default value is 1.
Modifiable	ALTER SYSTEM
Modifiable in a PDB	Yes
Range of values	The value of this parameter should be less than MAX_SHARED_SERVERS. If it is greater than or equal to MAX_SHARED_SERVERS, then the number of servers will not be self-tuned but will remain constant, as specified by SHARED_SERVERS.

Setting `SHARED_SERVERS` Guideline

- Start with allocating 1 shared server for every 10 concurrent connections
- If the request rate is low, the connections-to-servers ratio could be higher and vice versa
- The parameter is dynamic, therefore, consider dynamically setting the parameter based on the workload type along the day

Configuring Dispatchers

- Is configured by setting the parameter **`DISPATCHERS`**
- At least one dispatcher process is required for shared server to work.
- If you set **`SHARED_SERVER`** to a nonzero value, then by default Oracle Database creates one dispatcher for the TCP protocol

```
DISPATCHERS=(PROTOCOL=TCP)
```

- More than one **dispatcher configuration** can be configured
  - Each one could be assigned to specific protocol/port/service name

DISPATCHERS Simplified Syntax

```
DISPATCHERS = '(PROTOCOL = protocol) |
               (ADDRESS = address) |
               (DESCRIPTION = description
   [( DISPATCHERS = integer |
      SESSIONS = integer |
      CONNECTIONS = integer |
      LISTENER = tnsname |
      SERVICE = service |
      INDEX = integer)
   ]'
```

DISPATCHERS Parameter Attributes

Attribute	Description
PROTOCOL	The network protocol of the dispatcher listening endpoint.
ADDRESS	The network protocol address of the dispatcher listening endpoint.
DESCRIPTION	The network description address of the dispatcher listening endpoint.
DISPATCHERS	The initial number of dispatchers to start. The default is 1
SESSIONS	The maximum number of network sessions to allow for each dispatcher.
CONNECTIONS	The maximum number of network connections to allow for each dispatcher.
LISTENER	Specifies the network name of an address or address list of the Oracle Net listeners with which the dispatchers will register.
SERVICE	Specifies one or more names by which clients can connect to the dispatchers.
INDEX	Indicates which dispatcher configuration we want to modify.

Databases Created by dbca

- Databases created by dbca has the following setting by default:

```
SHARED_SERVERS=1
DISPATCHERS=(PROTOCOL=TCP) (SERVICE=oradbXDB)
```

- Can be used only for Oracle XML DB
- Cannot be used for connecting to regular database sessions

Configuring a Dispatcher Configuration Example

- Configuring one dispatcher-configuration:

```
ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=TCP)";

ALTER SYSTEM SET DISPATCHERS="(PROTOCOL=TCP)(DISPATCHERS=2)";

ALTER SYSTEM SET DISPATCHERS = "(PROTOCOL=TCP) (SERVICE=hrdb)";

ALTER SYSTEM SET DISPATCHERS = "(ADDRESS=(PROTOCOL=TCP)(HOST=srv1)
(PORT=26979)) (CONNECTIONS=20) (SERVICE=oradb.localdomain)";
```

- To retrieve information about the existing dispatcher configurations:

```
SELECT CONF_INDX, DISPATCHERS, CONNECTIONS FROM V$DISPATCHER_CONFIG;
```

Configuring Multiple Dispatcher Configurations Examples

- Using a name-value syntax:

```
... DISPATCHERS='(PROTOCOL=tcp)', '(PROTOCOL=ipc)'
```

- Using **`INDEX`** option:

```
DISPATCHERS='(PROTOCOL=tcp)(INDEX=0)', '(PROTOCOL=tcps) (INDEX=1)'
```

```
ALTER SYSTEM SET DISPATCHERS =
'(INDEX=0)(PROTOCOL=TCP)(SERVICE=oradbXDB)',
'(INDEX=1)(PROTOCOL=TCP)(DISPATCHERS=10)(SERVICE=oradb.localdomain)';
```

- Adding a new dispatcher or altering an existing one:

```
ALTER SYSTEM SET DISPATCHERS ='(INDEX=1)(PROT=TCP)'
```

  - It creates a new dispatcher configuration or alters the existing one

Notes on Altering Dispatchers

- The **`INDEX`** value can range from 0 to n-1, where n is the current number of dispatcher configurations
- If you do not specify **`INDEX`**, then the **first** dispatcher configuration matching the **`DESCRIPTION`**, **`ADDRESS`**, or **`PROTOCOL`** specified will be modified
- Changes that apply on existing dispatchers: **`LISTENER`**, **`SERVICES`**, and **`SESSIONS`** (if reduced)
- Changes that apply on new dispatchers: **`DESCRIPTION`**, **`ADDRESS`**, **`PROTOCOL`**, **`CONNECTIONS`**, and **`SESSIONS`** (if increased)
  - Changes on the current dispatchers do not take effect until we shut them down

Shutting Down Specific Dispatcher Processes

1. Identify the name of the specific dispatcher process to shut down:

```
SELECT NAME, NETWORK FROM V$DISPATCHER;
```

2. Shutdown the process:

```
ALTER SYSTEM SHUTDOWN [IMMEDIATE] 'D002';
```

Restricting Dispatcher Ports

- By default, a port is automatically assigned to the dispatcher process and could change with every restart:

```
lsnrctl service
...
 "D001" established:170 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: srv1.localdomain, pid: 4282>
         (ADDRESS=(PROTOCOL=tcp)(HOST=srv1)(PORT=17259))
...
```

- To set a specific port number, use **`ADDRESS`** attribute:

```
DISPATCHERS='(ADDRESS=(PROTOCOL=tcp)(HOST=srv1)(PORT=1579))(DIS=2)'
```

- We cannot connect to dispatchers using local authentication

Setting **`MAX_SHARED_SERVERS`**

- The maximum number of shared server processes allowed to be running simultaneously.
- It should be greater than or equal to **`SHARED_SERVERS`** and less than **`PROCESSES`**
- Its value should allow other processes to run effectively

Other Shared Server Related Parameters

Parameter	Description
`MAX_DISPATCHERS`	The maximum number of dispatcher processes allowed to be running simultaneously (can be ignored in 19c)
`CIRCUITS`	The total number of virtual circuits that are available for inbound and outbound network sessions
`SHARED_SERVER_SESSIONS`	Specifies the total number of shared server sessions to allow.

Configuring Clients for Environments with Both Shared and Dedicated Servers

- If it is not specified, the database decides the connection type
- If you want a particular client to always use a dispatcher:

```
hrdb =
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=hr.mydomain.com)
  (SERVER=SHARED)))
```

- If you want a particular client to always use a dedicated process:

```
hrdb =
 (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=hr.mydomain.com)
  (SERVER=DEDICATED)))
```

Monitoring Shared Server Processes

View	Description
V$DISPATCHER_CONFIG	Provides configuration information about the dispatchers configured in the database.
V$DISPATCHER	Provides information about the dispatcher processes
V$DISPATCHER_RATE	Provides rate statistics for the dispatcher processes
V$SHARED_SERVER	Provides information on the shared server processes
V$SHARED_SERVER_MONITOR	Provides rate statistics for the shared server processes
V$QUEUE	Provides information about the shared server message queues
V$SESSION	`SERVER` column could be either `'SHARED`' or `'NONE`'

Summary

In this lecture, you should have learnt how to perform the following:

- Describe the shared server architecture and its benefits
- Configure the shared server
- Setup the dispatcher configurations
- Configure Clients for shared and dedicated servers