Implementing Connection Load Balancing and TAF

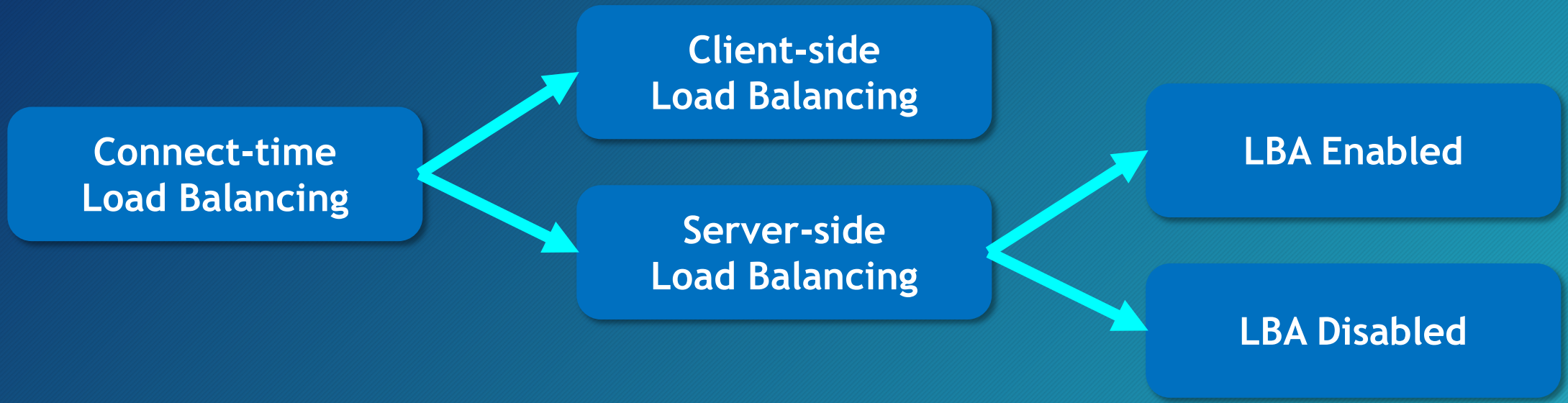
By Ahmed Baraka

Objectives

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

- Set up the following configurations:
  - Client-side connect-time load balancing
  - Server-side connect-time load balancing with and without having the Load Balancing Advisory (LBA) enabled
  - Transparent Application Failover (TAF) on client side
  - Basic TAF on server side
  - Preconnect TAF on the client side and on the server side.
- Describe Fast Connection Failover (FCF)
- Describe and enable Fast Application Notification (FAN)

Load Balancing Options



Client-Side Connect-Time Load Balancing

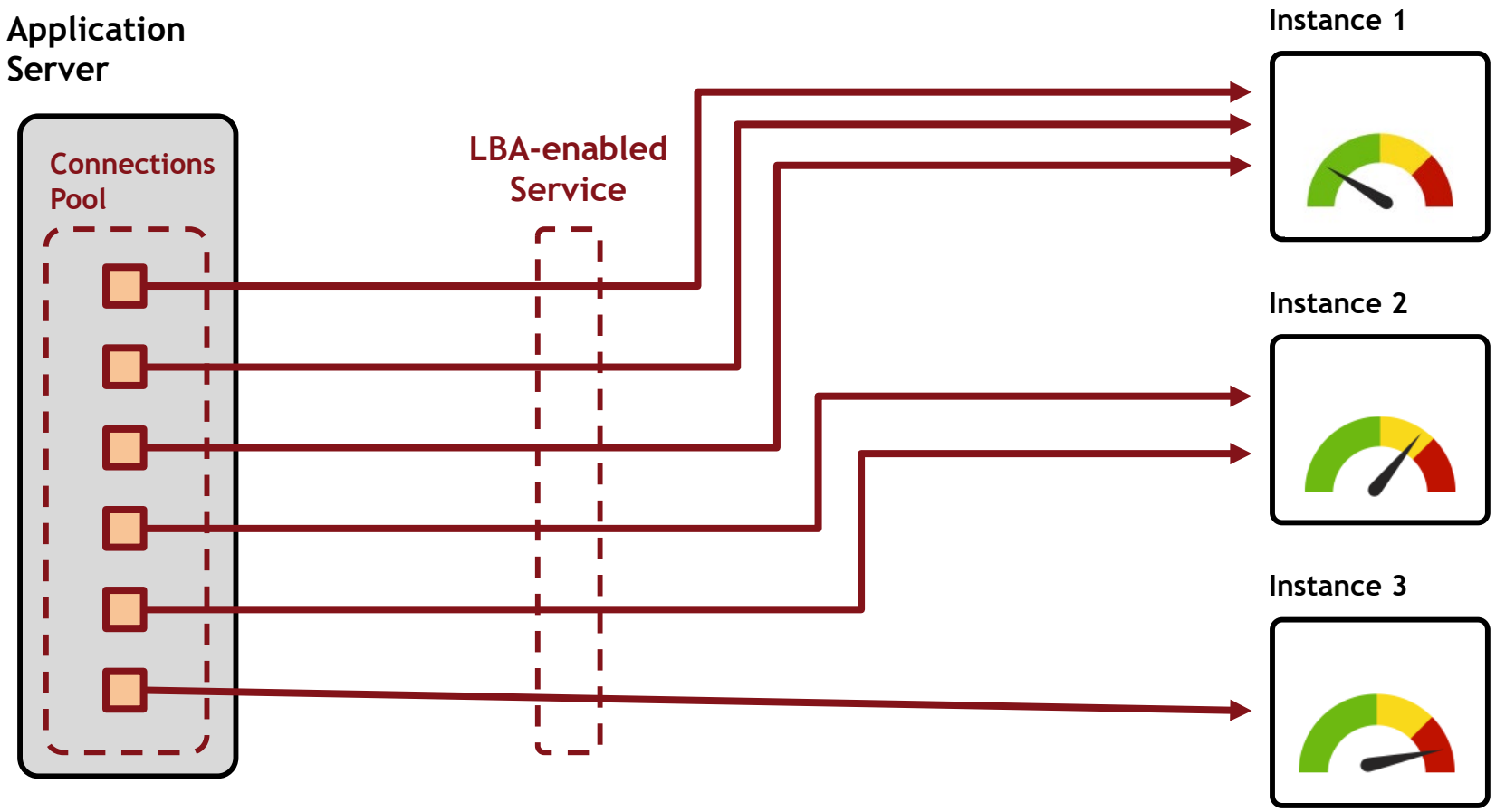
- Defined by setting the parameter **LOAD_BALANCE=ON** in **tnsnames.ora**

```
HRSRV=
 (DESCRIPTION =
  (ADDRESS_LIST =
    (LOAD_BALANCE=ON)
    (ADDRESS=(PROTOCOL=TCP)(HOST=rac1vip)(PORT=1521))
    (ADDRESS=(PROTOCOL=TCP)(HOST=rac2vip)(PORT=1521))
   )
  (CONNECT_DATA=(SERVICE_NAME=HRSRV))
  )
```

Client-Side Connect-Time Load Balancing (cont)

- Defined by setting the parameter **`LOAD_BALANCE=ON`** in **`tnsnames.ora`**
- Typically, you'd use the VIP names in the list
- Client randomly selects an address in the address list and connects to that node's listener.
- Using SCAN IP is irrelevant
- Regardless of the load balancing option in the server side
- Regardless on the load on the nodes

LBA-enabled Service



Overview of the Load Balancing Advisory

- Sends incoming work to the instances based on quality of service
- Recognizes the machine power differences
- Supports the following client technologies:
  - JDBC universal connection pool, OCI session pool, Oracle WebLogic Server Active, GridLink for Oracle RAC, and the ODP.NET Connection Pools.
  - Third party applications: JDBC and Oracle RAC FAN API or callbacks with OCI.
- If LBA is not enabled, connections are distributed equally to the instances.

Enabling LBA in a Service

- To enable LBA in a service, set service-level goals for run-time connection load balancing (**`rlbgoal`**):
  - **`SERVICE_TIME`**: work requests distributed based on response time

```
srvctl modify service -db rac -service orders
-rlbgoal SERVICE_TIME -clbgoal SHORT
```

  - **`THROUGHPUT`**: work requests distributed based on throughput

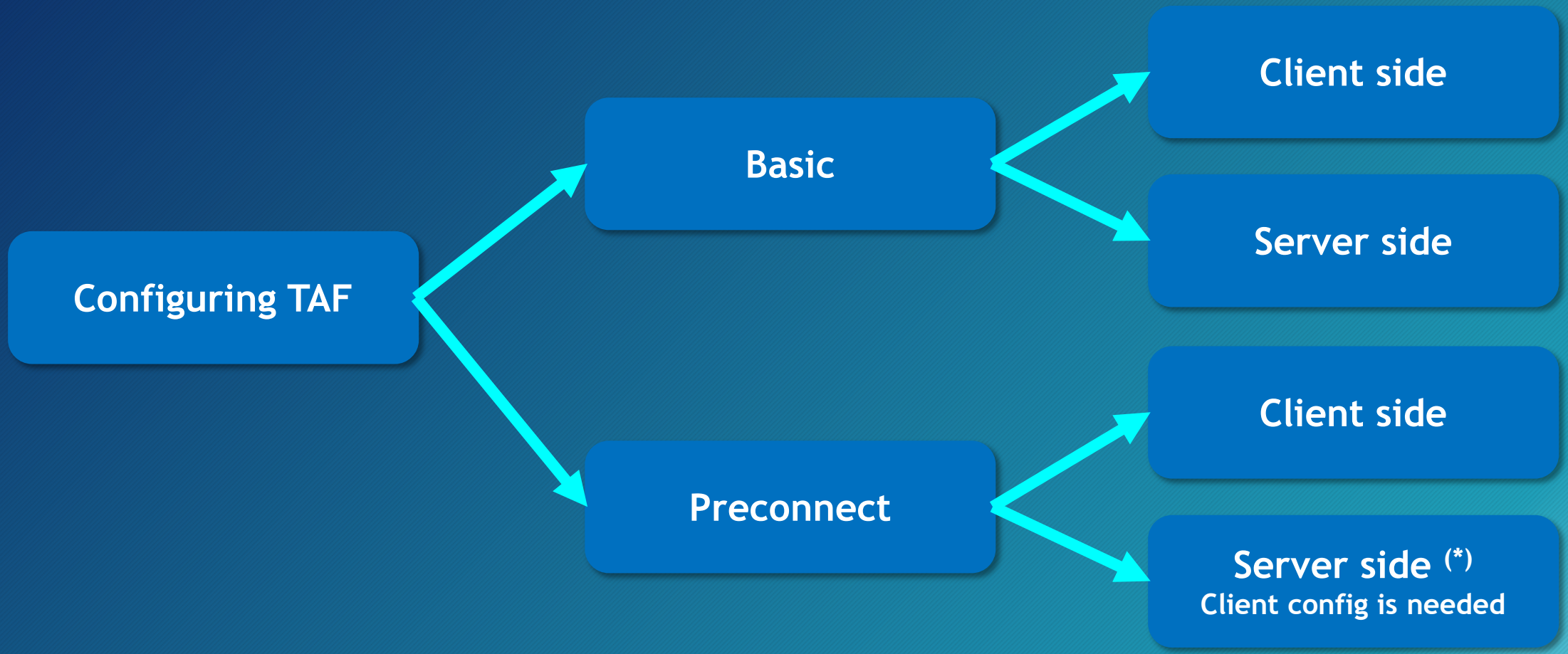
```
srvctl modify service -db rac -service batchj
-rlbgoal THROUGHPUT -clbgoal LONG
```

- If `rlbgoal` is not configured (or set to **`NONE`**), the LBA is disabled

Transparent Application Failover (TAF) Overview

- OCI driver feature that can be used with Thick JDBC clients
- A failed connection is failed over to surviving instance
  - Opened transactions are rolled back
  - It can resume the **`SELECT`** statement
- Two methods supported:
  - **`BASIC`**: application establishes a new connection to a surviving instance
  - **`PRECONNECT`**: a shadow connection is initially made
- Can be configured in the client-side as well as in the server-side

Configuring TAF Options



TAF BASIC Configuration on Client-side

- In `tnsnames.ora` file configure:
  - **FAILOVER**: set it to **ON**
  - **FAILOVER_MODE**
    - **TYPE**: specifies the type of failover. It accepts: `SESSION`, `SELECT`, or `NONE`
    - **METHOD**: speed of the failover. It accepts: `BASIC`, or `PRECONNECT`
    - **RETRIES**: how many times Oracle Net should try to reconnect
    - **DELAY**: how many seconds Oracle Net should wait before trying to connect again

TAF BASIC Configuration on Client-side: Example

```
ctaf =
 (DESCRIPTION =(FAILOVER=ON) (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=TCP)(HOST=rac1-vip)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=rac2-vip)(PORT=1521))
      (CONNECT_DATA =
            (SERVICE_NAME = hrsrv)
            (FAILOVER_MODE = (TYPE=SELECT)
                             (METHOD=BASIC)
                             (RETRIES=10)
                             (DELAY=10)))
  )
```

TAF BASIC Configuration on Server-side

- Use the following options in the **`add service`** command:

```
srvctl add service -d db_unique_name -service service_name
...
[-failovertype {NONE|SESSION|SELECT|TRANSACTION}]
[-failovermethod {NONE | BASIC}]
[-failoverretry failover_retries]
[-failoverdelay failover_delay]
```

```
srvctl add service -db racdb -service hrsrv -preferred rac1
-available rac2,rac3 -failovermethod BASIC
-failovertype SELECT -failoverretry 10 -failoverdelay 5
```

TAF PRECONNECT Configuration on Client-side: Example

```
HRSRV =
 (DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=srv1-vip)(PORT=1521))
   (CONNECT_DATA=
        (SERVICE_NAME=hrsrv)
        (FAILOVER_MODE=
              (BACKUP=HRSRV2)
              (TYPE=SELECT)
              (METHOD=PRECONNECT)))
  )
...
```

TAF PRECONNECT Configuration on Client-side: Example (cont)

```
HRSRV2 =
 (DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=srv2-vip)(PORT=1521))
   (CONNECT_DATA=
        (SERVICE_NAME=hrsrv)
        (FAILOVER_MODE=
              (BACKUP=HRSRV)
              (TYPE=SELECT)
              (METHOD=PRECONNECT)))
  )
```

TAF PRECONNECT Configuration on Server-side: Example

- On the server side:

```
srvctl add service -db racdb -service hrsrv -preferred rac1
-available rac2,rac3 -tafpolicy PRECONNECT
```

- An internal **`hrsrv_preconnect`** service will be created
- In this example, TAF is actually not enabled, because **`FAILOVER_TYPE`** is not defined.
- Cannot be configured in policy-managed RAC databases

TAF PRECONNECT Configuration on Server-side: Example (cont)

```
# client side configuration for the server-side TAF PRECONNECT:
PRESRV =
 (DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=srv-scan)(PORT=1521))
   (CONNECT_DATA=
        (SERVICE_NAME=presrv)
        (FAILOVER_MODE=
               (BACKUP=PRESRV2)
               (TYPE=SELECT)
               (METHOD=PRECONNECT)))
  )
```

TAF PRECONNECT Configuration on Server-side: Example (cont)

```
PRESRV2 =
 (DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=srv-scan)(PORT=1521))
   (CONNECT_DATA=
        (SERVICE_NAME=presrv_preconnect)
        (FAILOVER_MODE=
               (BACKUP=HRSRV)
               (TYPE=SELECT)
               (METHOD=PRECONNECT)))
  )
```

TAF Verification

- Current sessions attributes show if failover is enabled:

```
SELECT MACHINE, FAILOVER_METHOD, FAILOVER_TYPE,
FAILED_OVER, SERVICE_NAME, COUNT(*)
FROM GV$SESSION
GROUP BY MACHINE, FAILOVER_METHOD, FAILOVER_TYPE,
FAILED_OVER, SERVICE_NAME;
```

- Service attributes:

```
SELECT NAME, FAILOVER_METHOD, FAILOVER_TYPE
FROM DBA_SERVICES
```

About Fast Application Notification (FAN)

- The mechanism used by Oracle to notify the processes about the service level information and instances status, as UP or DOWN.
- Supported by the following technologies without programmatic changes:
  - Oracle JDBC Universal Connection Pool
  - ODP.NET connection pool
  - OCI session pool
  - Oracle WebLogic Server Active Gridlink
- Can be used by programming in JDBC, ODP .NET, and OCI

About Fast Notification Application (FAN) (cont)

- Used by Fast Connection Failover (FCF)
- Used by Application Continuity or Transaction Guard
- To enable FAN in a service:

```
srvctl modify service -db rac -s hrsrv -notification TRUE
```

About Fast Connection Failover (FCF)

- A failover feature that you can enable in the connection pool
- Supports the following failover features:
  - Planned outages: in-use connections are not interrupted, closed only after the work is done.
  - Unplanned outages: connections made to a failed instance automatically removed from pool
  - Is aware of adding a new node
- Oracle Notification Service (ONS) to propagate database events
- Recommended over TAF configuration

Configuring FCF Overview

- FCF can be configured for the following clients:
  - JDBC OCI and JDBC Thin clients
  - OCI Clients (even if TAF is configured)
- .. in the following connection pooling technologies:
  - Universal Connection Pool or Implicit Connection Cache (*deprecated*)
  - Oracle WebLogic Server Active GridLink
  - OCI connection pool, and OCI session pools.
- Details provided in "*Universal Connection Pool Developer's Guide*" and "*Oracle Call Interface Programmer's Guide*"

Summary

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

- Set up the following configurations:
  - Client-side connect-time load balancing
  - Server-side connect-time load balancing with and without having the Load Balancing Advisory (LBA) enabled
  - Transparent Application Failover (TAF) on client side
  - Basic TAF on server side
  - Preconnect TAF on the client side and on the server side.
- Describe Fast Connection Failover (FCF)
- Describe and enable Fast Application Notification (FAN)