Oracle Net Services 12c

Best Practices for Database Performance and Scalability

Kant C Patel
Director
Oracle Net

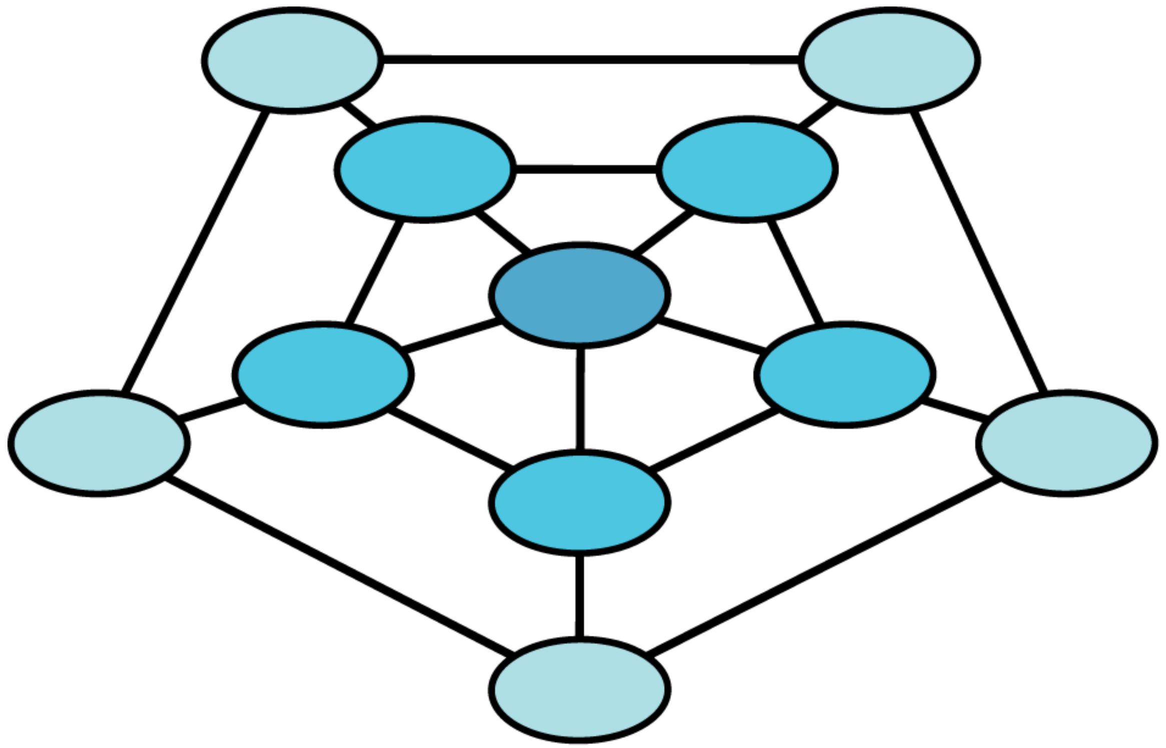
ORACLE OPEN WORLD

October 1–5, 2017
SAN FRANCISCO, CA

ORACLE

Program Agenda

- Overview of Oracle Net
  - Why Optimize Oracle Net?
- Best Practices
  - Database Client
  - Listener and Connection Manager
  - Database Server
- Q/A



Oracle Net Overview

- Primary Communication Foundation for Oracle DB
- Also known as SQL*Net
- Oracle’s Family of Networking Features:
  - Oracle Net
  - Oracle Net Listener
  - Connection Manager
  - Configuration Tools

Why Optimize Oracle Net?

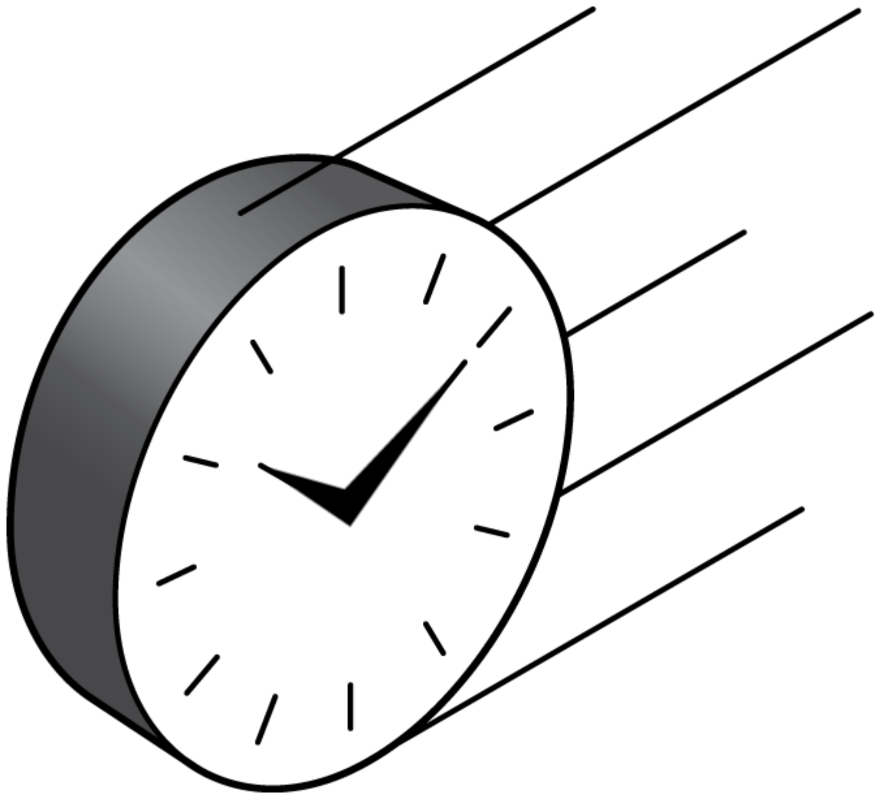
- High Availability
  - Better respond to database/host/network failures
- Network Scalability and Performance
  - Scale better with more client connections
  - Load-balance to improve application experience
  - Increase Network bandwidth utilization
  - Lower database CPU utilization
- Network Security
  - Protect and recover from Denial of Service attacks

Net Configuration Files

- sqlnet.ora
  - Main Oracle Net configuration file
  - On both Client and Server
- listener.ora
  - Configuration for the Net Listener
  - On Server only
- tnsnames.ora
  - Contains Connect Name to Descriptor mappings
  - Used by the TNSNames Naming adapter
  - On both Client and Server
- ldap.ora
  - Contains LDAP configuration information
  - Used the LDAP Naming adapter
  - On both Client and Server

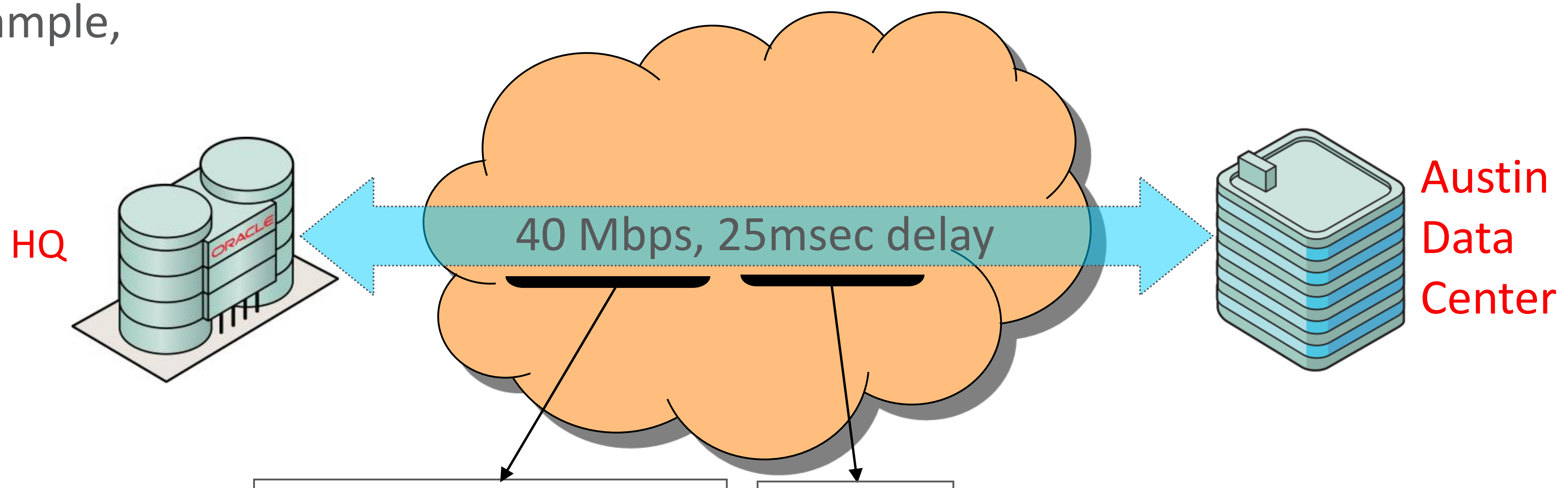
Program Agenda

- Overview of Oracle Net
  - Why Optimize Oracle Net?
- **Best Practices**
  - **Database Client**
  - Listener and Connection Manager
  - Database Server
- Q/A



Tuning the Socket Buffers
What is BDP?

- Bandwidth x Delay Product (BDP)
  - Amount of data on the "wire" at any given point in time
  - Default Operating System buffers do not hold enough data to fill the wire
  - For example,



BDP = $\boxed{(40 \times 1000 \div 8 \text{ Kbytes/sec})} \times \boxed{(0.025 \text{ sec})}$ ~ 128 Kbytes

Tuning the Socket Buffers

- Set send and receive socket buffer sizes using:
  - SEND_BUF_SIZE – OS send buffer size
  - RECV_BUF_SIZE – OS receive buffer size
- Set this size to accommodate the BDP (2x)
- Set on both the server and the client
- Large buffer sizes help
  - Application queue more data to the OS
  - Have more data on the wire
  - Better utilize available bandwidth
  - In WAN deployments

Where to configure?

Client : sqlnet.ora and/or
tnsnames.ora or LDAP

Server: sqlnet.ora and
listener.ora

Tuning the Session Data Unit

- Controls SQL*Net packet size
  - Default: 8K
  - Max: 2MB (12c), 64K (11.2), 32K (pre-11.2)
- Set in
  - sqlnet.ora: DEFAULT_SDU_SIZE
  - tnsnames.ora: SDU in address
- Larger SDU gives
  - Better Network throughput
  - Fewer system calls to send and receive data
  - Less CPU usage – system and user
- Side-effect of larger SDU: Network buffers take up more memory

Where to configure?

Client : sqlnet.ora and/or
tnsnames.ora or LDAP

Server: sqlnet.ora

SDU Recommendations

- Optimal SDU varies with application
- Increase SDU on both client and server
  - SDU for a connection negotiated down to the lower of the two peers
- Increase SDU to 8k (for pre-11g clients)
  - Good default value for most users
- For bulk data transfer scenarios, increase to 64k
  - Large array fetch
  - LOB transfers
  - XML DB
- Do not set to MTU value
  - SDU and MTU are independent

Connect-time Failover for Applications
Oracle Net Timeouts

- Connection establishment timeouts
  - Timeout for TCP connection establishment
    - TCP.CONNECT_TIMEOUT
    - Enabled by default to 60 seconds since 11gR2
  - Timeout for connection to a DB server process
    - SQLNET.OUTBOUND_CONNECT_TIMEOUT
    - Set if session establishment takes a long time
- Configurable at connect string level
- Can be used individually or at the same time
  - Outbound Connect Timeout must be greater than TCP Timeout
- Option to enable connection retries and retry delay

Where to configure?

Client: sqlnet.ora and/or
tnsnames.ora or LDAP

Single Client Access Name (SCAN)

Easy Connect + Address List

- Available since 11gR2
- Single name for clients to access an Oracle Database in a cluster
- Configured during the installation of Grid Infrastructure
- Typically resolves to three IP addresses in the cluster, each associated with a SCAN Listener

```
For example, if DNS resolves
sales-scan to
{10.1.1.1, 10.1.1.2, 10.1.1.3}
```

```
sales-scan:10240/sales
```

is equivalent to

```
(DESCRIPTION=
  (ADDRESS_LIST=
     (ADDRESS=(PROTOCOL=tcp)
         (HOST=10.1.1.1)(PORT=10240))
    (ADDRESS=(PROTOCOL=tcp)
         (HOST=10.1.1.2)(PORT=10240))
    (ADDRESS=(PROTOCOL=tcp)
         (HOST=10.1.1.3)(PORT=10240)))
   (CONNECT_DATA=
      (SERVICE_NAME=sales)))
```

SCAN in Connect Descriptors

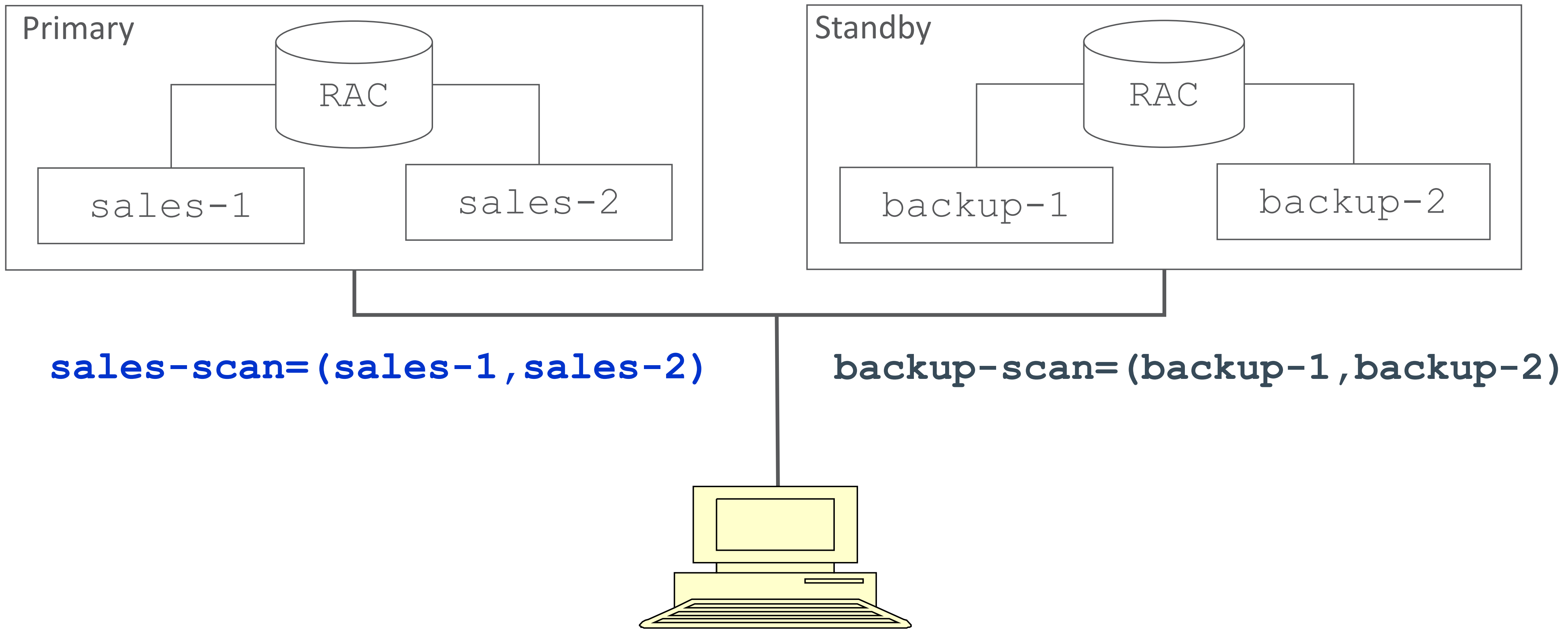
```
sales=
   (DESCRIPTION=
      (ADDRESS_LIST=
         (LOAD_BALANCE=on)
         (ADDRESS=
               (PROTOCOL=tcp)
               (HOST=sales-scan)
               (PORT=10240)))
      (CONNECT_DATA=
          (SERVICE_NAME=sales)))
```

sales-scan:

```
      10.1.1.1, 10.1.1.2, 10.1.1.3
```

```
sales=
   (DESCRIPTION=
      (ADDRESS_LIST=
         (LOAD_BALANCE=on)
         (ADDRESS=
            (PROTOCOL=tcp)
            (HOST=10.1.1.1)
            (PORT=10240))
        (ADDRESS=
            (PROTOCOL=tcp)
            (HOST=10.1.1.2)
            (PORT=10240))
        (ADDRESS=
            (PROTOCOL=tcp)
            (HOST=10.1.1.3)
            (PORT=10240)))
     (CONNECT_DATA=
        (SERVICE_NAME=sales)))
```

Client-side Failover using Address and Description Lists
RAC + Data Guard Example



Connect Descriptor

```
(DESCRIPTION_LIST =
      (LOAD_BALANCE=off)(FAILOVER=on)
      (DESCRIPTION =
          (LOAD_BALANCE=on)
          (ADDRESS=(PROTOCOL=tcp)(HOST=sales-scan)(PORT=1521))
          (CONNECT_DATA=(SERVICE_NAME=sales.example.com)))
      (DESCRIPTION =
          (LOAD_BALANCE=on)
          (ADDRESS=(PROTOCOL=tcp)(HOST=backup-scan)(PORT=1521))
          (CONNECT_DATA=(SERVICE_NAME=sales.example.com))))
```

The Connect Descriptor internally expands to

```
(DESCRIPTION_LIST =
     (LOAD_BALANCE=off)(FAILOVER=on)
     (DESCRIPTION =
         (ADDRESS_LIST=
             (LOAD_BALANCE=on)
             (ADDRESS=(PROTOCOL=tcp)(HOST=sales-1)(PORT=1521))
             (ADDRESS=(PROTOCOL=tcp)(HOST=sales-2)(PORT=1521)))
         (CONNECT_DATA=(SERVICE_NAME=sales.example.com)))
     (DESCRIPTION =
         (ADDRESS_LIST=
             (LOAD_BALANCE=on)
             (ADDRESS=(PROTOCOL=tcp)(HOST=backup-1)(PORT=1521))
             (ADDRESS=(PROTOCOL=tcp)(HOST=backup-2)(PORT=1521)))
         (CONNECT_DATA=(SERVICE_NAME=sales.example.com))))
```

Fail-over for Connected Sessions

- Established client connections could hang when
  - Database host crashes
  - Remote Networks fail
- Detection of such failures could take a while
  - TCP behavior - timeouts in minutes
  - Depends on what the client does
- To catch such failures
  - Set a Receive Timeout
    - If your application is active and does not use long-running queries
  - Use Fast Application Notification (FAN)

CON6711: Best Practices for Application High Availability
Wed. 10/04/2017 at 3:30pm Moscone West - Room 3012

Configuration Example – JDBC App

- Connect Timeout set through property

  `oracle.net.CONNECT_TIMEOUT`
- Read Timeout set through

  `oracle.jdbc.ReadTimeout`
  - Note: Do not use as a query-timeout.
- For Query Timeout, use

  `Statement.cancel` or

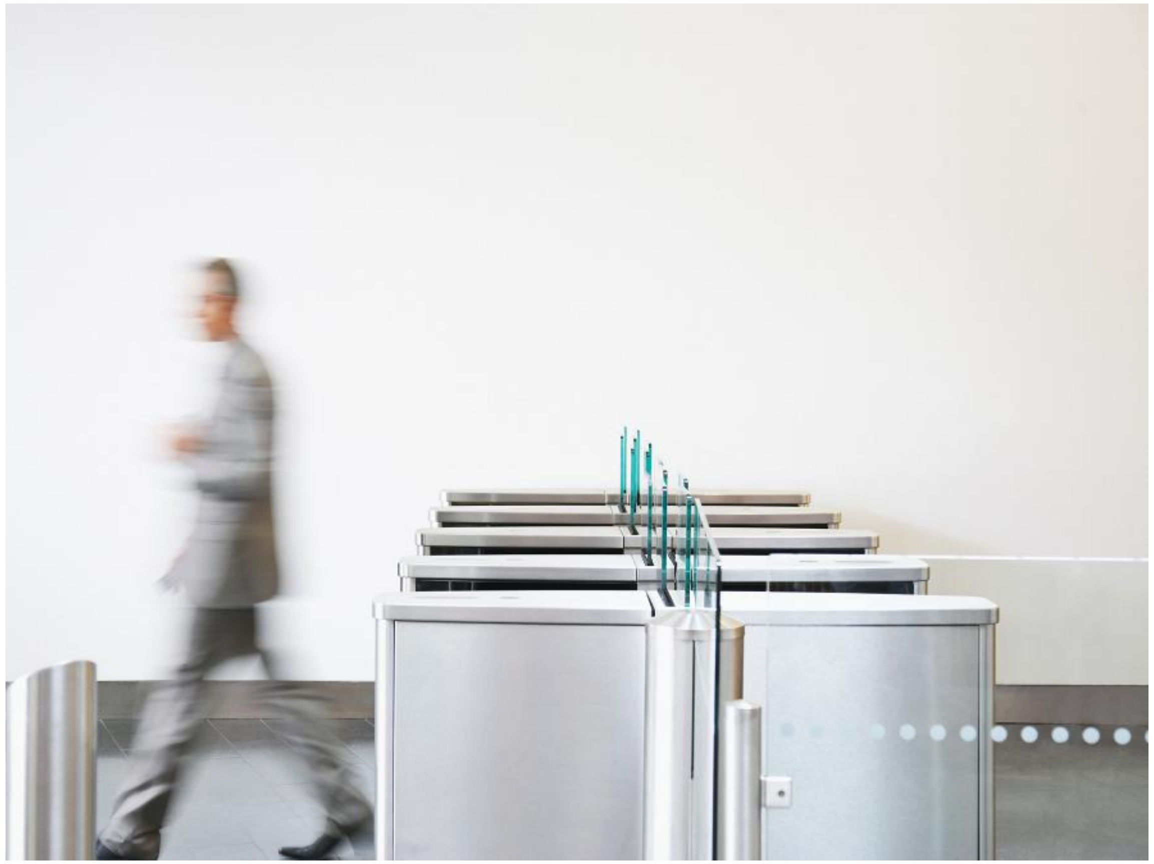
  `Statement.setQueryTimeout`
- How to set properties?

  `specify in code`

```
Code Example:
Properties prop = new Properties();
prop.setProperty("user","scott");
prop.setProperty("password","tiger");
prop.setProperty("oracle.net.CONNECT_TIMEOUT","3000");
prop.setProperty("oracle.jdbc.ReadTimeout","3000");
Conn=(new oracle.jdbc.OracleDriver()).connect(url,prop);
```

Program Agenda

- Overview of Oracle Net
  - Why Optimize Oracle Net?
- **Best Practices**
  - Database Client
  - **Listener and Connection Manager**
  - Database Server
- Q/A



Database Service Availability and Load Balancing

What is the Net Listener?

- First process that clients talk to
- Brokers client requests, handing them off to service handlers
  - Dispatchers
  - Dedicated servers
  - Connection Broker – DRCP
- Receives load updates from the database
- Does server side load-balancing across instances in RAC
- Does server side failover across nodes in RAC
- Can listen on multiple end-points or protocol addresses
- Also supports other presentations – HTTP, FTP

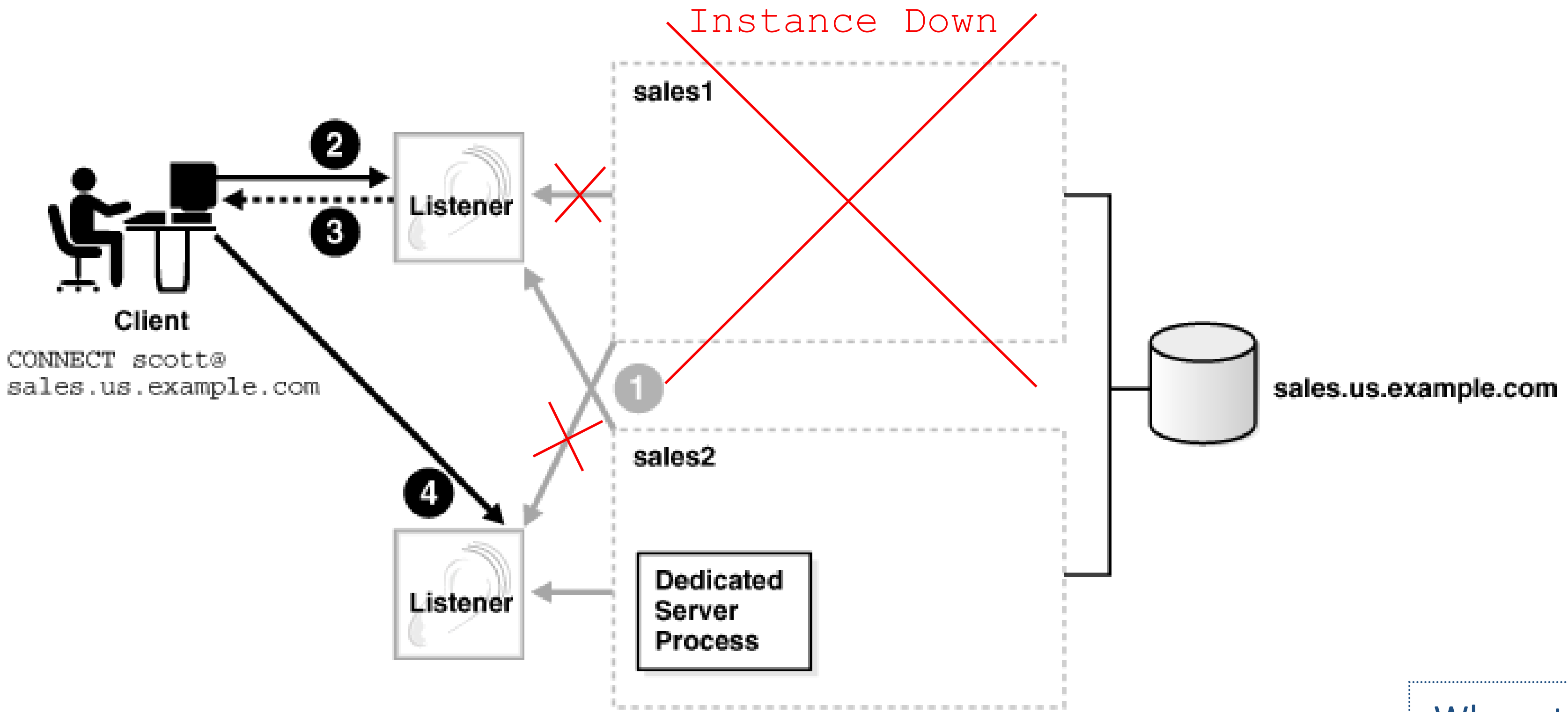
Database Service Availability and Load Balancing
Database Registration with Listener

- Use Dynamic Registration
  - LREG (or PMON in pre-12c) updates the listener about
    - Offered services and available service handlers
    - Load statistics – frequently updated
  - To configure, set in init.ora
    - LOCAL_LISTENER: Address of listeners on local host
    - REMOTE_LISTENER: Address of listeners on remote hosts
  - By default
    - LREG connects to listener on port 1521
    - Automatically setup with RAC
- Remove static SID_LIST configuration in listener.ora
  - Keep only if you want to remotely start the database

Where to configure?

Database parameter file

Database Service Availability and Load Balancing (Example)



- Change behavior by setting Connection Load Balancing Goal
  - Long – for applications with long-lived connections (default)
  - Short – for applications with short-lived connections

Where to configure?

On server, using srvctl utility, or DBMS_SERVICE package

Listener Logon Storm Handler

- Logon storm
  - Sudden spike in incoming connection rate
    - Normal – middle-tier reboot
    - Abnormal – DoS attack
  - Storms cause CPU starvation for existing sessions
- Enable the Connection Rate Limiter feature
  - Provides end-point level control of throttling

```
LISTENER=(ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales)(PORT=1521)(RATE_LIMIT=3))
    (ADDRESS=(PROTOCOL=tcp)(HOST=lmgmt)(PORT=1522)(RATE_LIMIT=no)))
```

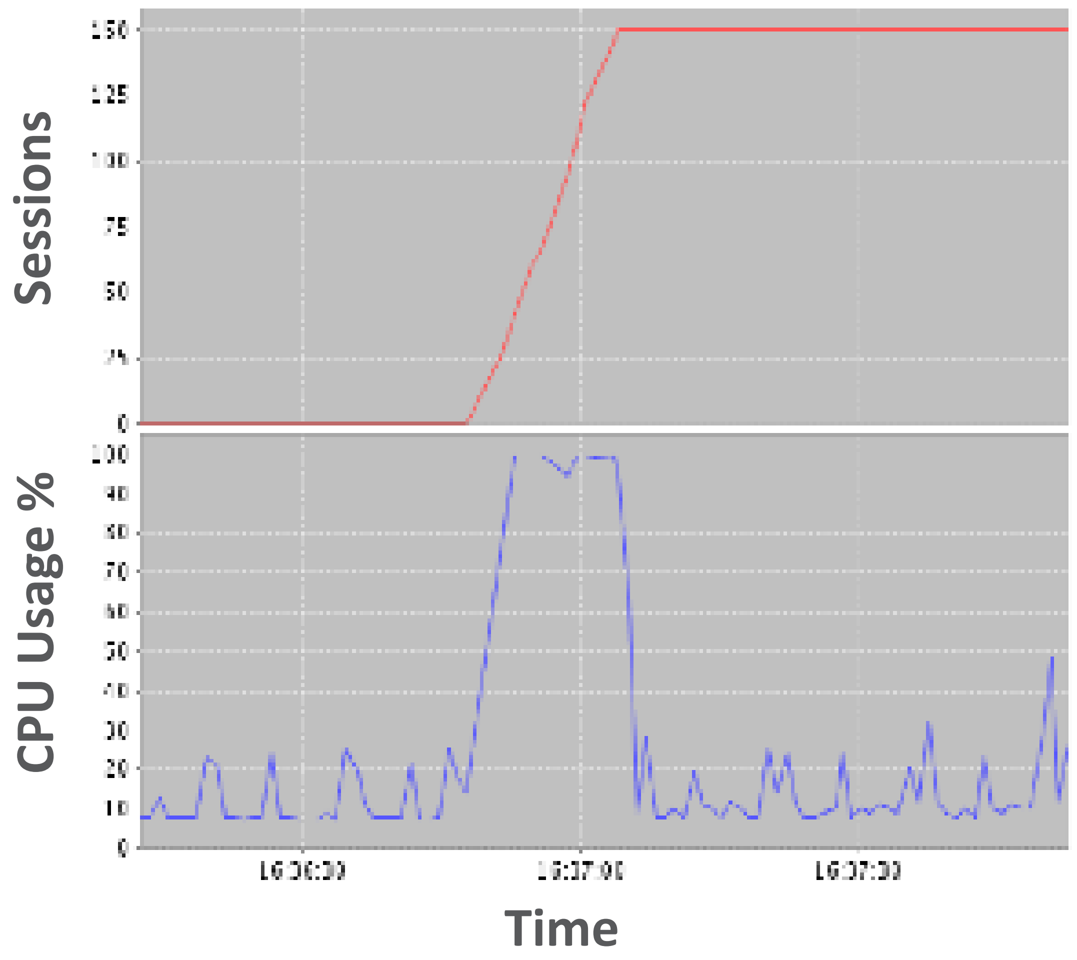
  - Can also be set globally at the listener level
  - Set the Rate Limit to a value that matches your machine capabilities

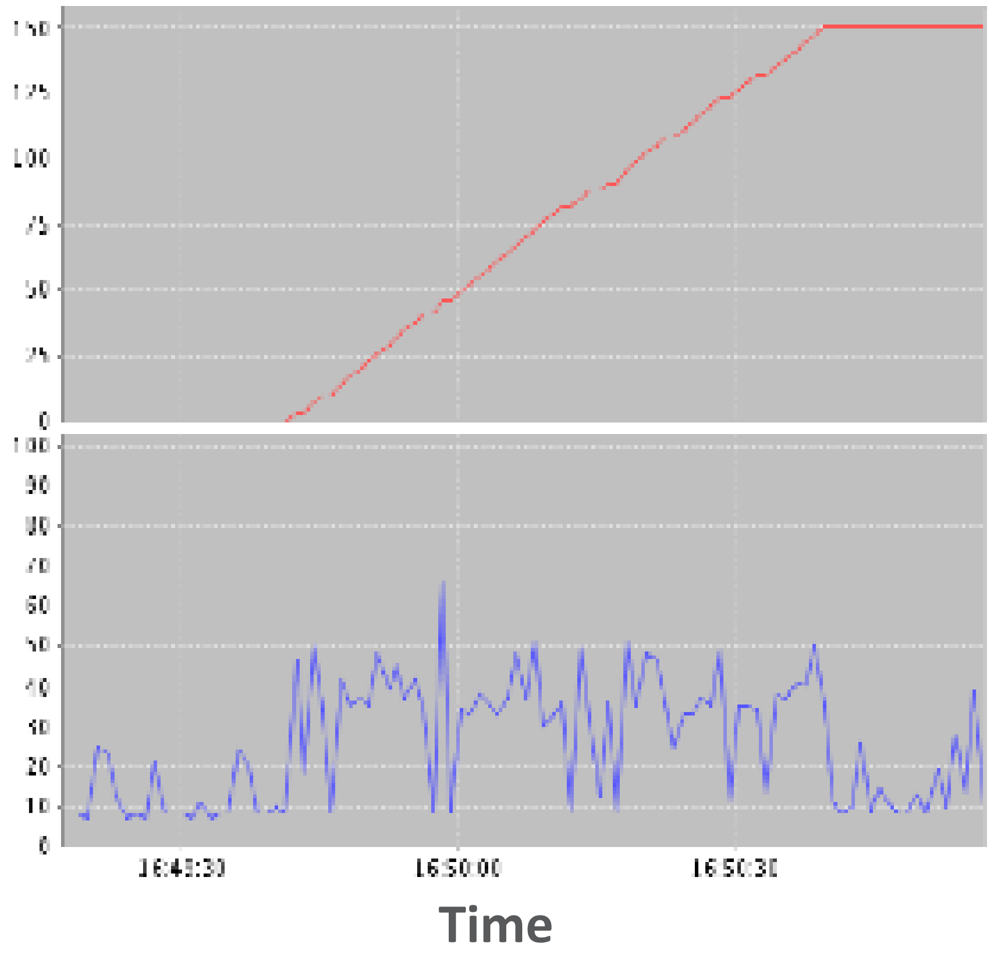
Where to configure?

Server: listener.ora

Logon Storm Comparison

150 concurrent connections





Other Best Practices

- Increase the maximum concurrent requests per listener end-point
  - QUEUESIZE parameter in listener.ora
  - Max `listen()` backlog for TCP (operating system parameter)
  - Set to your expected Connection Request rate

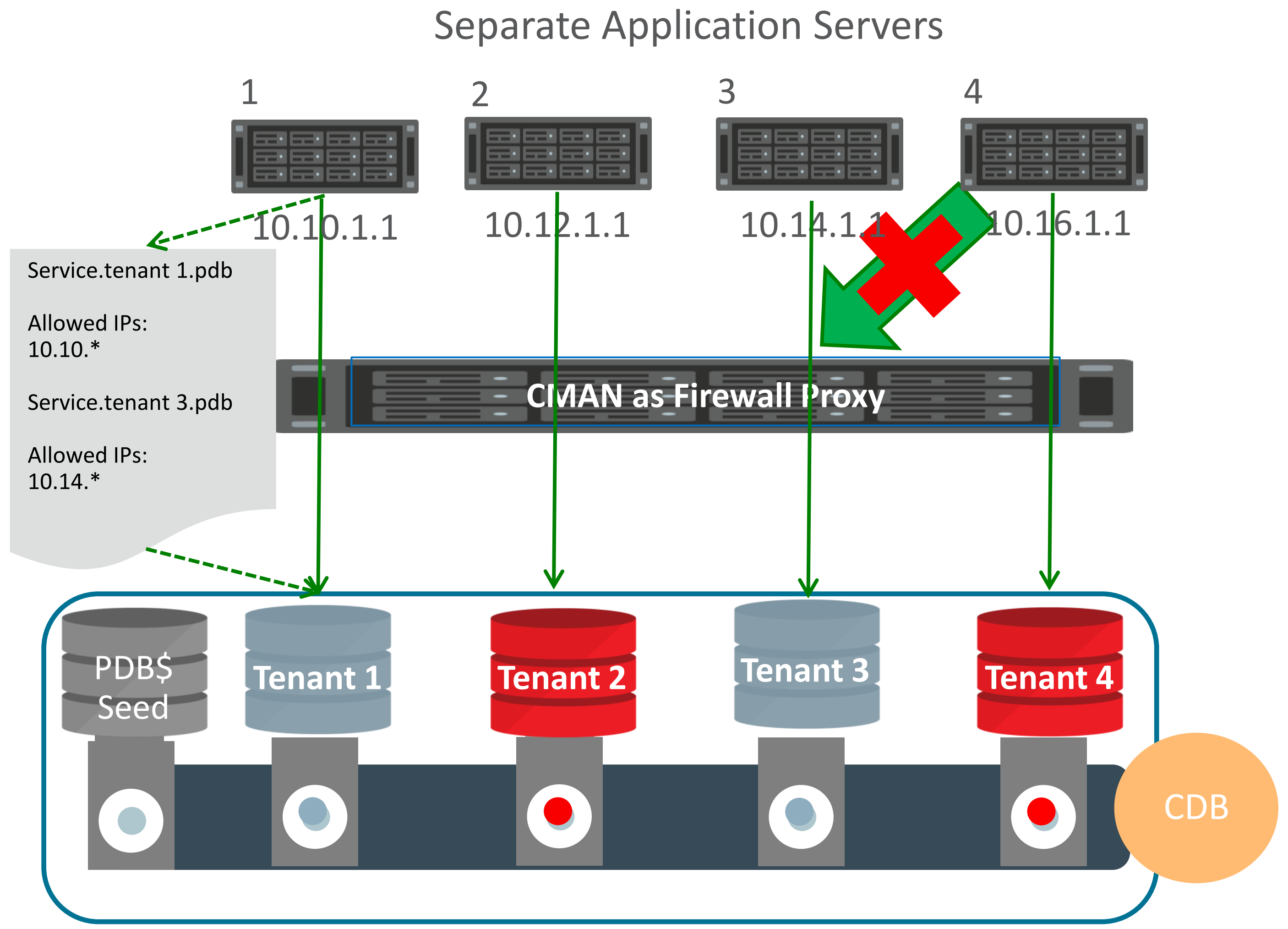
- Optimize Environment variables for the oracle account
  - Longer the PATH, longer it takes to fork off the Oracle process
    - Ensure that PATH is small
    - Does not include any network shares
  - Cut down the number of environment variables

Oracle Connection Manager (CMAN)

- Firewall Proxy aware of Oracle database services
  - Fully Transparent: no application changes required
  - Can be used to hide database subnet, including RAC re-directs, from clients
  - Provides network isolation for multi-tenant environments
- Supports all SQL*Net protocols as well as protocol conversion
  - Can be used as a bridge between IPv4 and IPv6 networks
- Provides access control to services based on configurable rule lists
- Auto-updates when new databases and services are added

Oracle Connection Manager

Access Control



Program Agenda

- Overview of Oracle Net
  - Why Optimize Oracle Net?
- **Best Practices**
  - Database Client
  - Listener and Connection Manager
  - **Database Server**
- Q/A

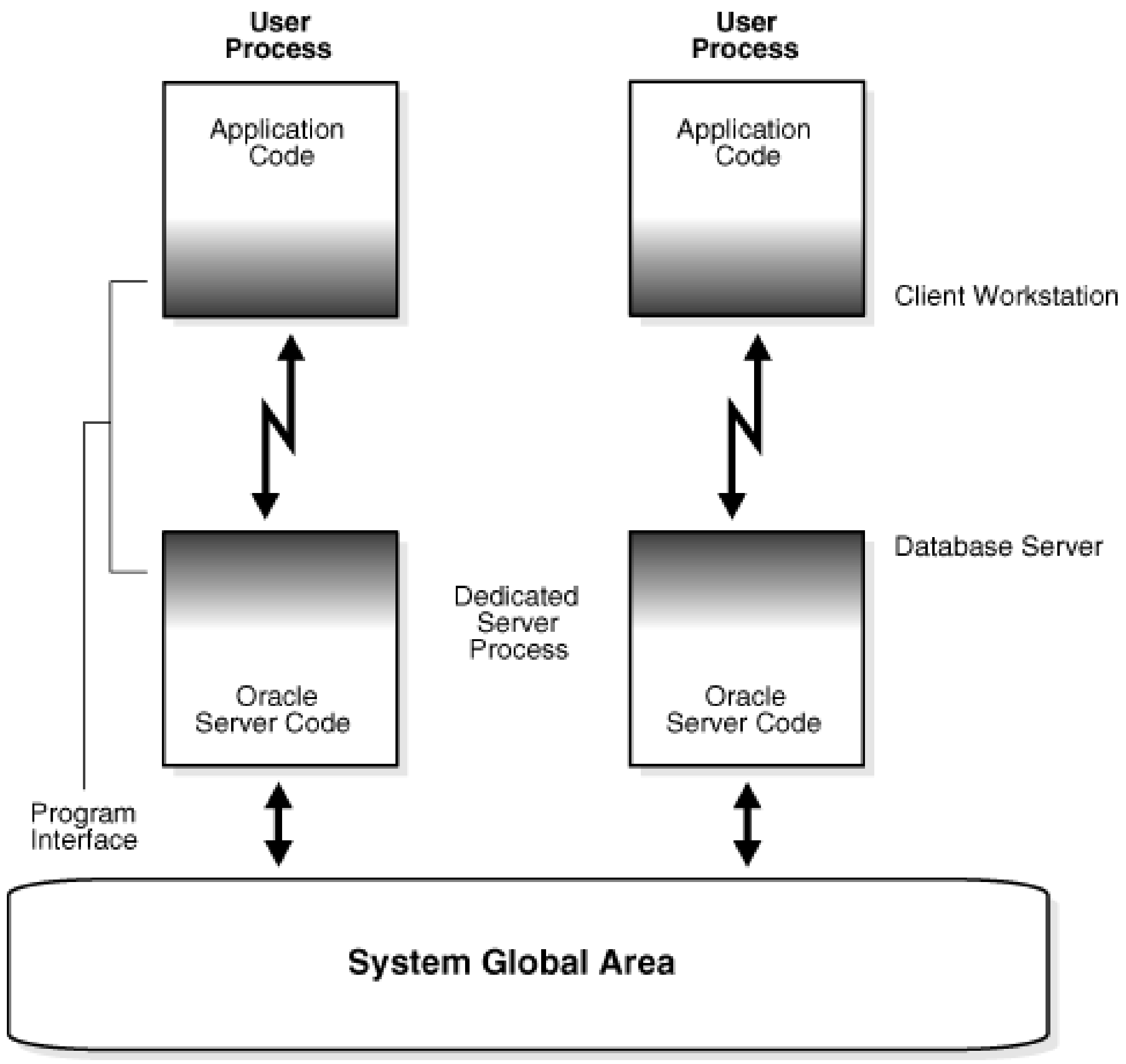
Oracle Server Architecture Overview

- Choosing the right server architecture is critical to meeting scalability requirements

- Oracle Database Server supports three architectures
  - Dedicated Server (default)
  - Shared Server aka MTS
  - Database Resident Connection Pool

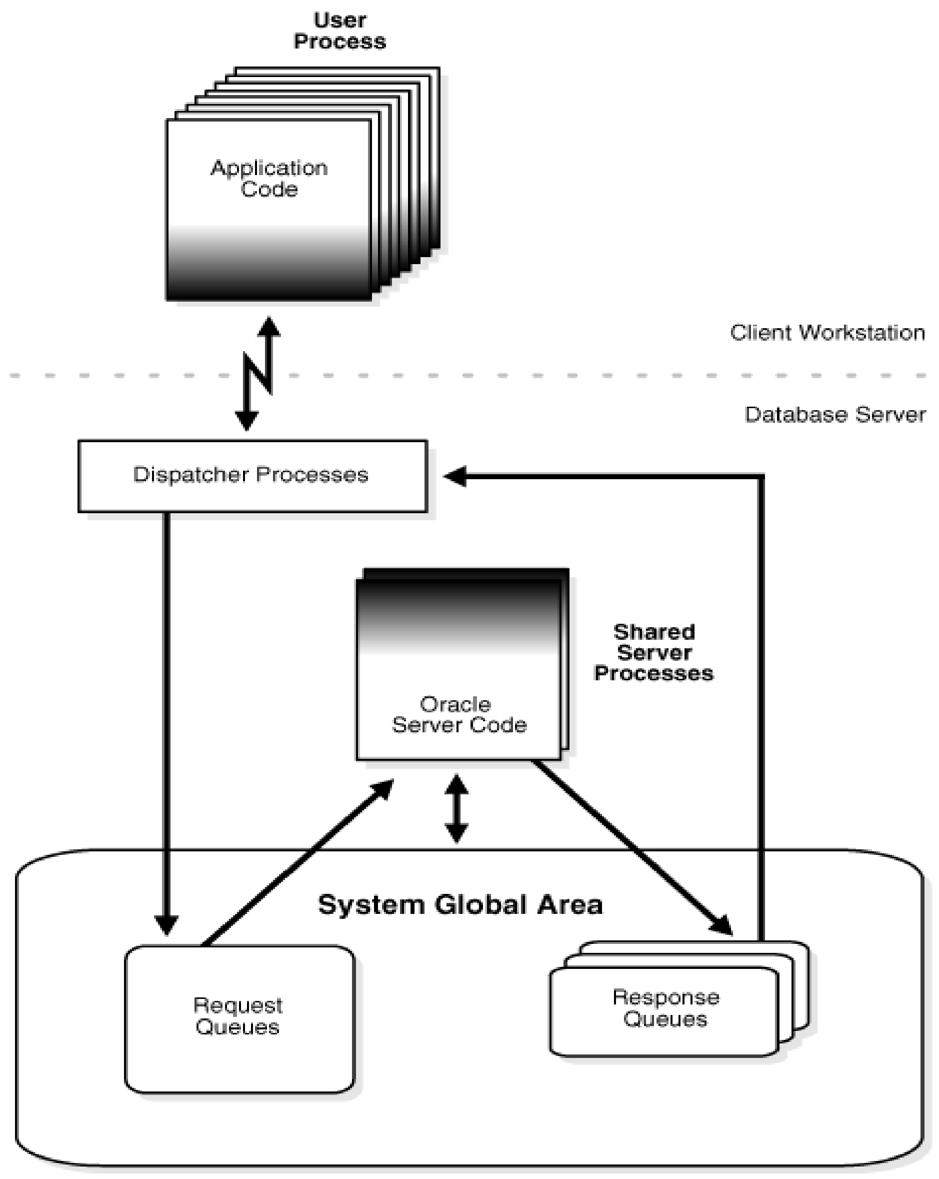
Dedicated Servers

- Each client connection has its own process (thread on Windows)
- Dedicated process ensures lower latencies
- Have to start a new process on connect
- Have to tear down a process on disconnect
- Scalability limits
  - Memory
  - Number of Processes



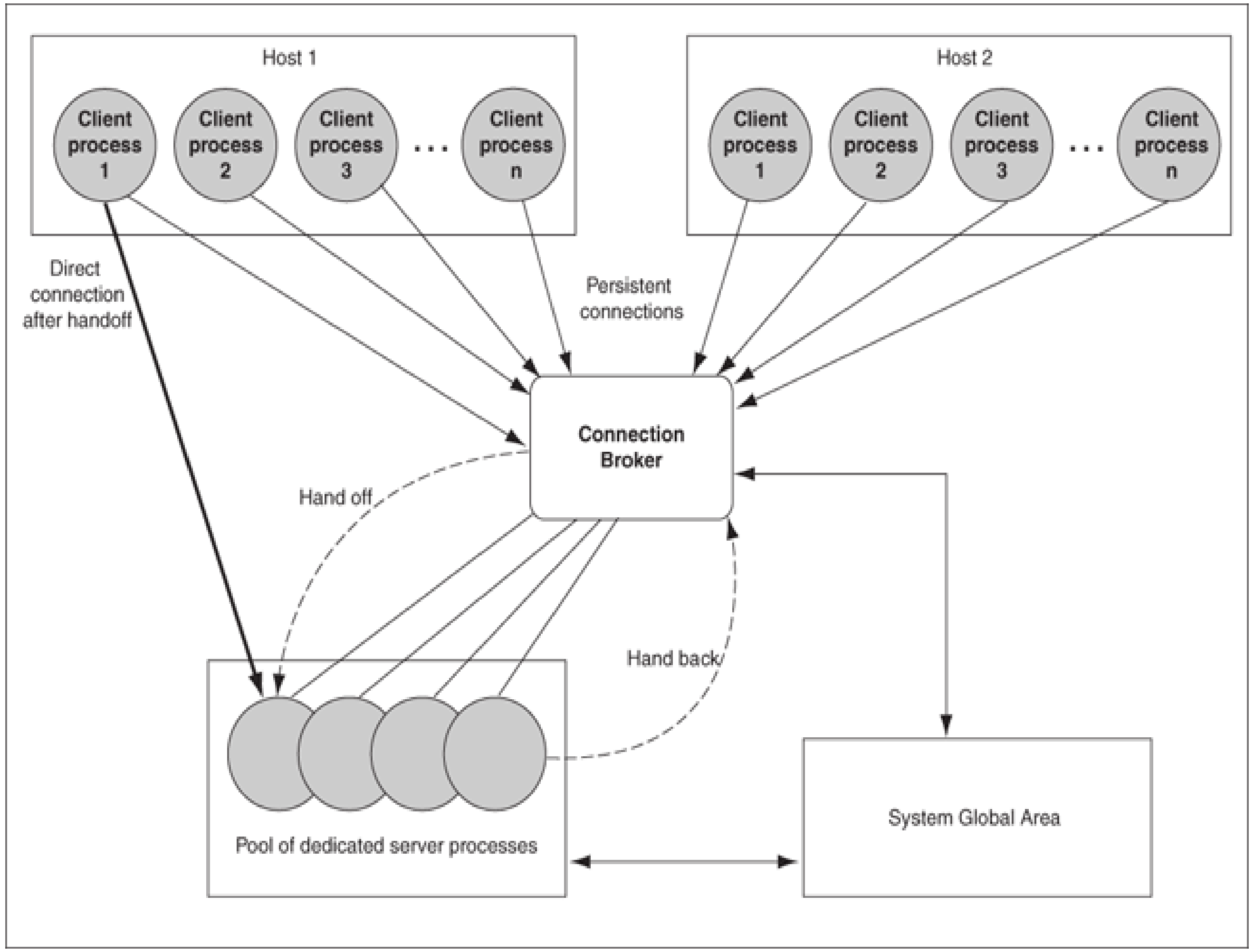
Shared Servers (aka MTS)

- Each server handles multiple clients
- Dispatchers relay requests and responses between clients and servers
- Idle connections will not consume much memory
- Good for large number of connections with many idle
- Latency increase due to man- in-the-middle



Database Resident Connection Pool

- Pooled dedicated servers shared across client systems and processes
- Low connect/disconnect costs
  - Server "locked" on connect
  - Server "released" on disconnect
- Low-latency performance of dedicated servers
- Extreme scalability with a DRCP-capable client driver



Dedicated vs. Shared vs. DRCP

- Use dedicated for:
  - High-performance connections
  - Active, long-running, data transfer intensive operations
- Use shared for:
  - Sessions that may be idle for some time
  - Clients that frequently connect and disconnect
- Use DRCP:
  - When you have thousands of clients which need access to a database server session for a short period of time
  - Applications mostly use same database credentials, and have identical session settings
  - OCI, OCCI, JDBC, PHP (OCI8 extension), Python (cx_Oracle), Perl (DBI)

Using Shared Servers

- Enable shared servers with init.ora parameters
  - Becomes new default
- To force server type, specify server type during connect
  - Dedicated:
    - `sales-server/sales.us.example.com:`**`dedicated`**
  - Shared:
    - `sales-server/sales.us.example.com:`**`shared`**
- Rough guidelines:
  - 20 Shared Servers per 500 sessions, then tune from there
  - 1 dispatcher for every 250 sessions

Using DRCP

- Pooling is enabled by the DBA using

```
EXECUTE DBMS_CONNECTION_POOL.START_POOL
        ('SYS_DEFAULT_CONNECTION_POOL');
```

- Change connect string on client in tnsnames.ora:

```
(DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
     (CONNECT_DATA=(SERVICE_NAME=sales)(SERVER=pooled)))
```

- Can use Easy Connect syntax too

```
sqlplus joeuser@sales-server:1521/sales:pooled
```

- In test environment, we were able to support more than 20,000 connections to a 2 GB Database Server (11g)

ORACLE

Database Server Security
Inbound Connection Timeouts

- Limits the time taken for a client to connect and authenticate
- SQLNET.INBOUND_CONNECT_TIMEOUT
  - Controls timeout for Database server processes
- INBOUND_CONNECT_TIMEOUT_*listener_name*
  - Controls timeout for the listener
- Enabled by default to 60 seconds
- Independent of client-side timeouts

Where to configure?

Server: sqlnet.ora
listener.ora

ORACLE

Dead Connection Detection

- Used by server to detect outage of client nodes
  - Enabled using `sqlnet.expire_time`
- Legacy DCD feature uses Oracle Net probe packets
  - Relies on TCP send failures
  - Slower and has more overhead
- New DCD (12c)
  - Relies on TCP keep-alive
  - Efficient detection of dead clients
  - Avoids probe buffering or out-of-buffer issues
  - Maintains backwards compatibility with older clients

Where to configure?

Server: sqlnet.ora

Database Server Security
TCP Valid Node Checks

- Use TCP Invited Nodes
  - List of IPs or hostnames that are permitted to connect
- Use TCP Excluded Nodes
  - List of IPs or hostnames that are NOT permitted to connect
- Use CIDR notation and wildcard format for ease of configuration whenever possible
- Invited nodes takes precedence over excluded
- To enable, set

```
TCP.VALIDNODE_CHECKING = YES
TCP.INVITED_NODES  = (hostname1, hostname2)
TCP.EXCLUDED_NODES = (hostname3, hostname4)
```

Where to configure?

Server: sqlnet.ora

For More Information

search.oracle.com

Oracle Net Services Product Overview

or

oracle.com/technetwork/database/enterprise-edition/index-098579.html

ORACLE

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.