10g

ORACLE®

SOFTWARE POWERS THE INTERNET™

10g

Services for a DBA

May Your Workloads RIP
(Run In Peace)

David Austin

Manager

Technical Writing for RAC and Grid

Oracle USA, Inc.

david.austin@oracle.com

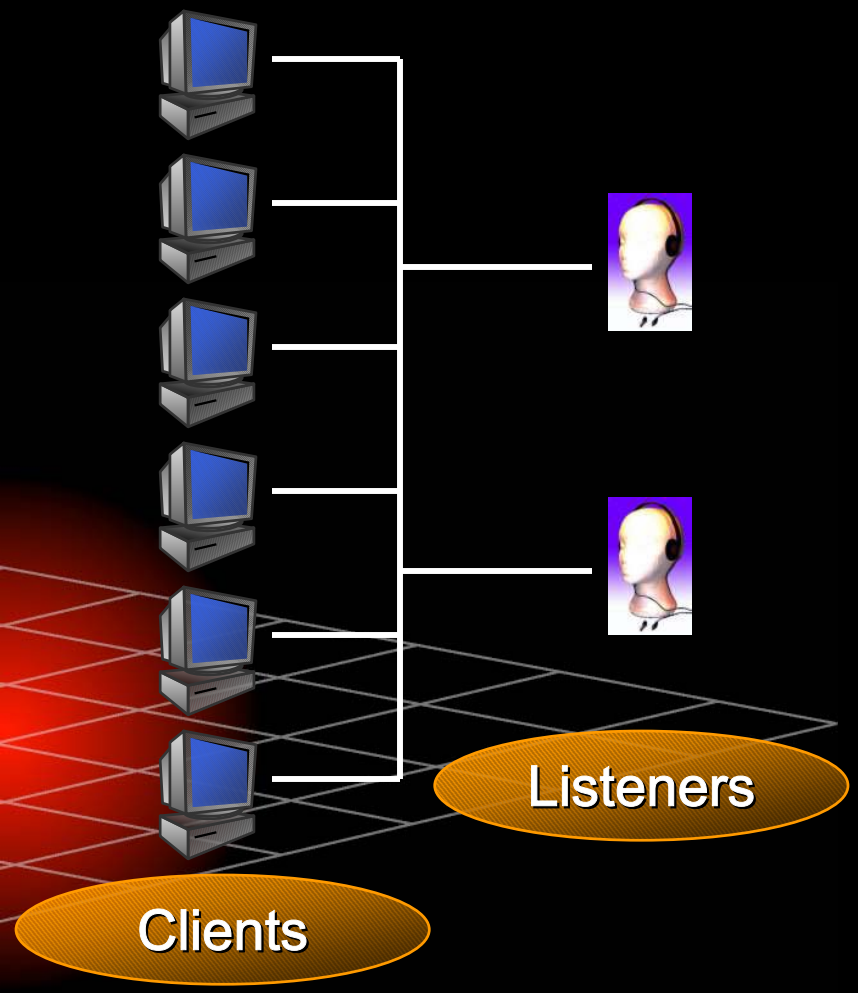
What, When, Why, Where, How?

- Connection load balancing
- Client-Side load balancing
- Server-Side load balancing
- Runtime connection load balancing
- Listener load balancing
- Session load balancing
- Client-Side connect time failover
- Transparent application failover
- Automatic Workload Management

Connection Load Balancing

- Each new connection is routed to the “best” server to maintain balance across workloads
- Two types of connection load balancing
  - Client-Side connection load balancing
  - Server-Side connection load balancing
- Can use both in a Real Application Clusters (RAC) environment

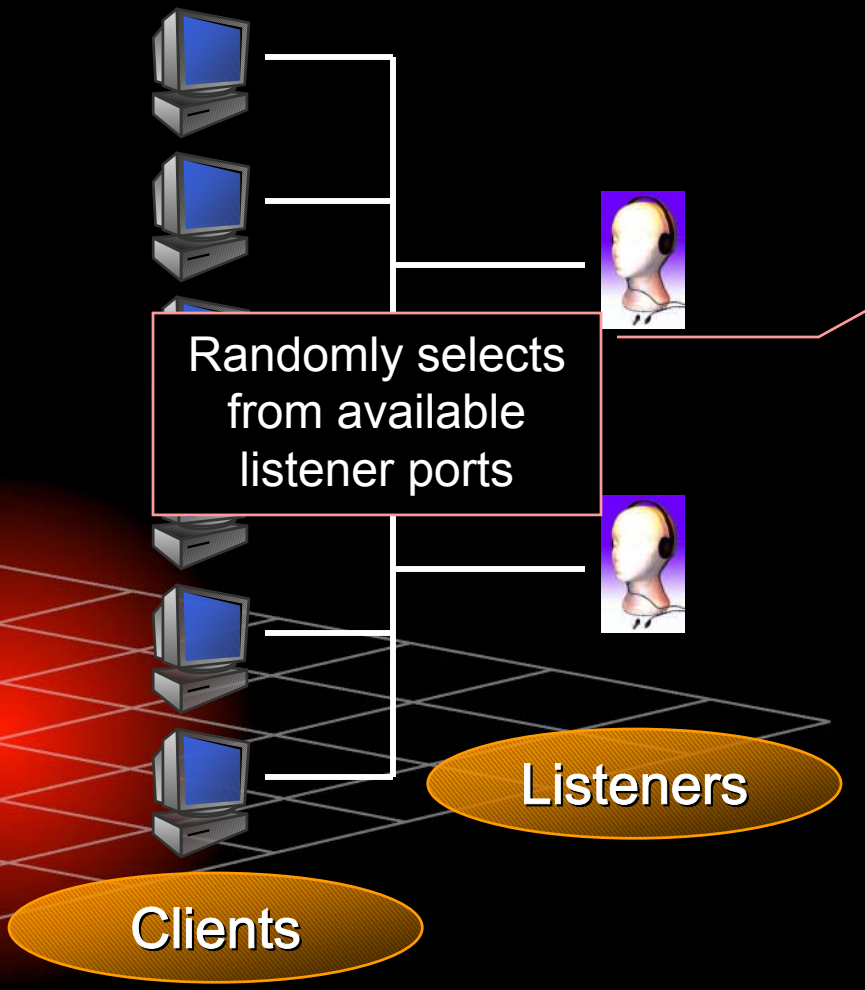
Client-Side Connection Load Balancing




```
sales.us.acme.com=
  (DESCRIPTION=
   (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=app1)
    (PORT=1521))
   (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=app2)
    (PORT=1521)))
   (CONNECT_DATA=
    (SERVICE_NAME=HR)))
```

tnsnames.ora

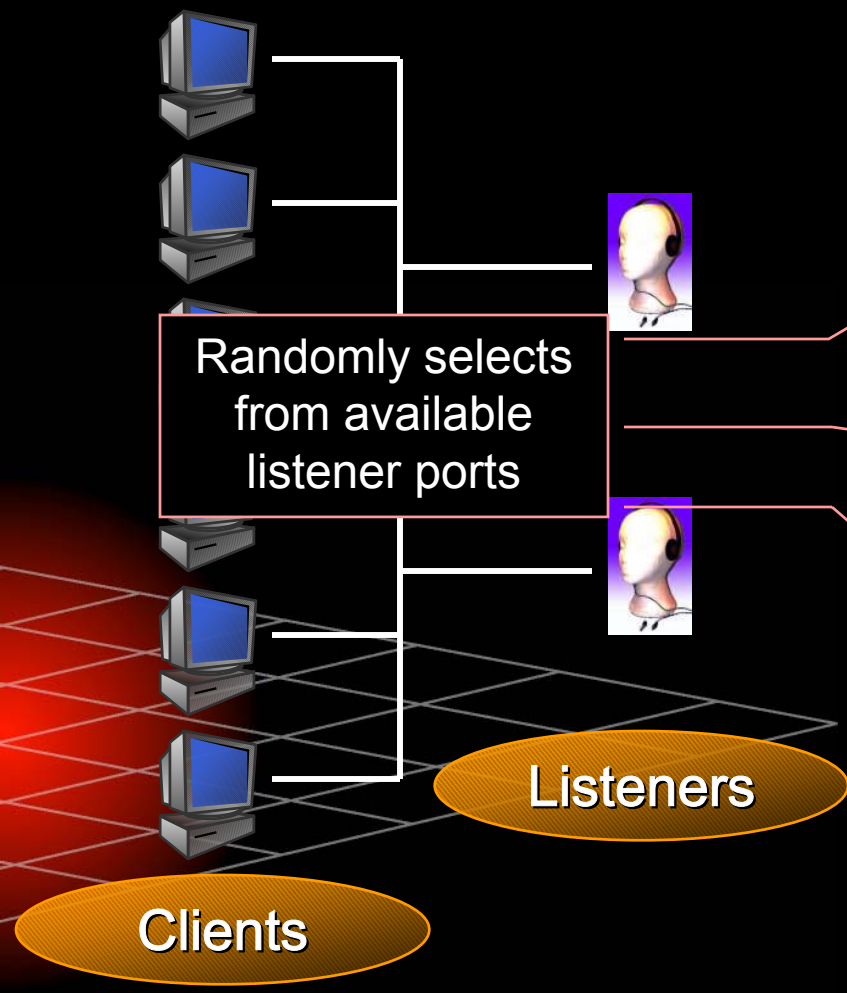
Client-Side Connection Load Balancing




```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
    (LOAD_BALANCE=on)
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=app1)
      (PORT=1521))
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=app2)
      (PORT=1521)))
    (CONNECT_DATA=
      (SERVICE_NAME=HR)))
```

tnsnames.ora

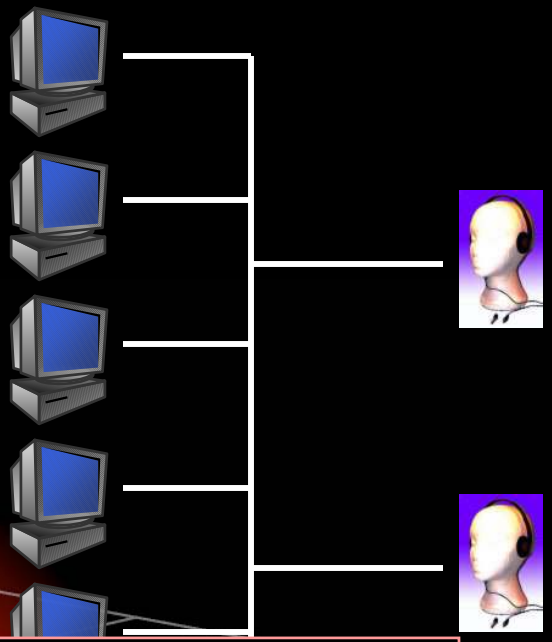
Client-Side Connection Load Balancing




```
sales.us.acme.com=
 (DESCRIPTION=
  (ADDRESS_LIST=
  (LOAD_BALANCE=on)
  (ADDRESS=
   (PROTOCOL=tcp)
   (HOST=app1)
   (PORT=1521))
  (ADDRESS=
   (PROTOCOL=tcp)
   (HOST=app2)
   (PORT=1521)))
  (CONNECT_DATA=
   (SERVICE_NAME=HR)))
```

tnsnames.ora

Server-Side Connection Load Balancing



Listener selects least loaded instance providing the required service

teners

Clients

```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
    (LOAD_BALANCE=on)
    (ADDRESS=
     (PROTOCOL=tcp)
     (HOST=app1)
     (PORT=1521))
    (ADDRESS=
     (PROTOCOL=tcp)
     (HOST=app2)
     (PORT=1521)))
    (CONNECT_DATA=
     (SERVICE_NAME=HR)))
```

Parameters

- For all listeners to recognize all available instances set the LOCAL_LISTENER and REMOTE_LISTENER parameters in the instance parameter file(s). For example:

```
LOCAL_LISTENER=
   (ADDRESS=(PROTOCOL=tcp)(HOST=app1)(PORT=1521))
```

and

```
REMOTE_LISTENER=
   (ADDRESS=(PROTOCOL=tcp)(HOST=app2)(PORT=1521))
```

on the app1 server

- To improve randomization set

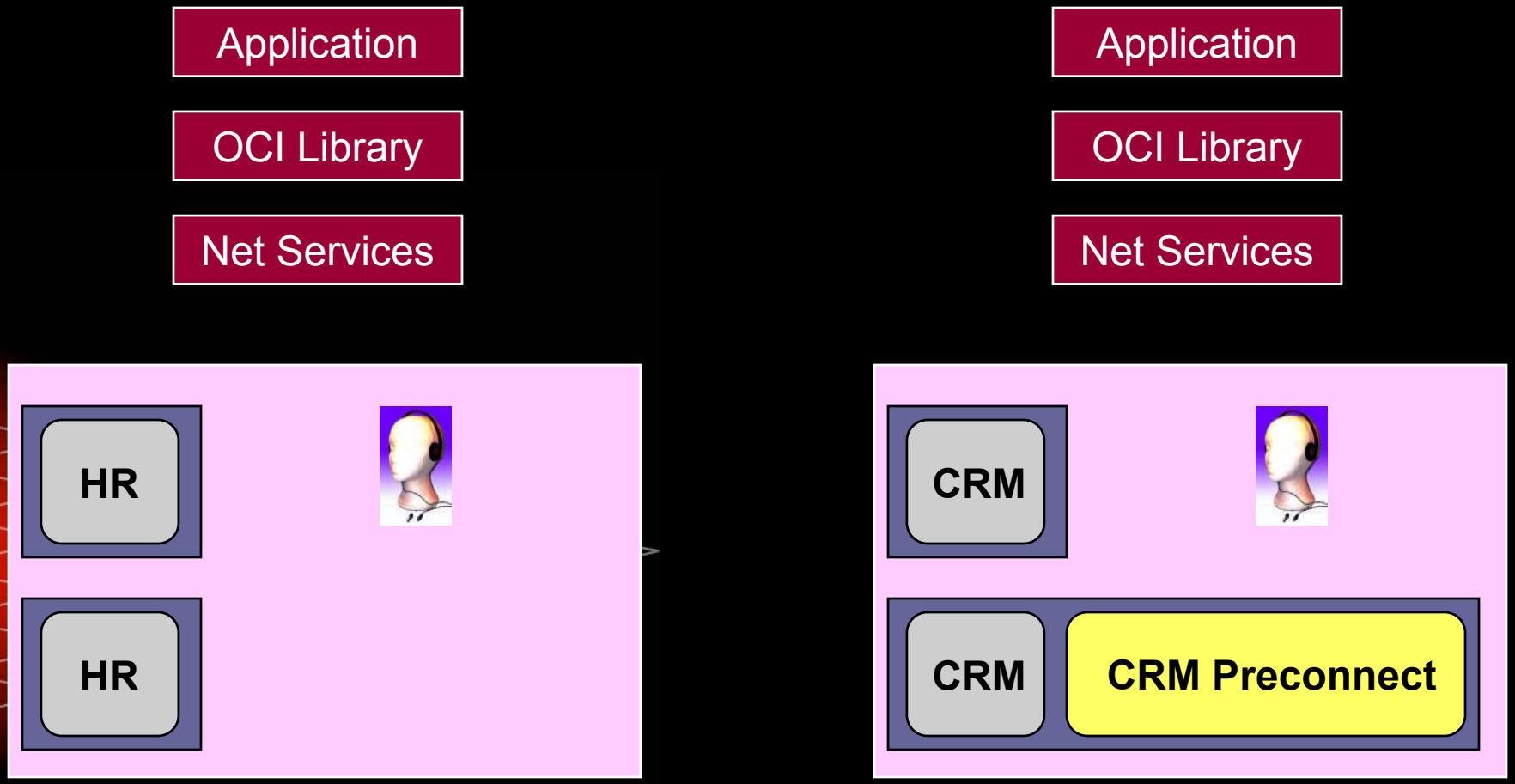
```
PREFER_LEAST_LOADED_NODE_<listener>=OFF
```

(in `sqlnet.ora` file)

Considerations

- Available since Oracle9*i*
- Can use for RAC or Data Guard logical standby database connections
- Logon storms can overwhelm the algorithms
  - May overload one instance regardless of settings
  - More likely with a small number of listeners and instances

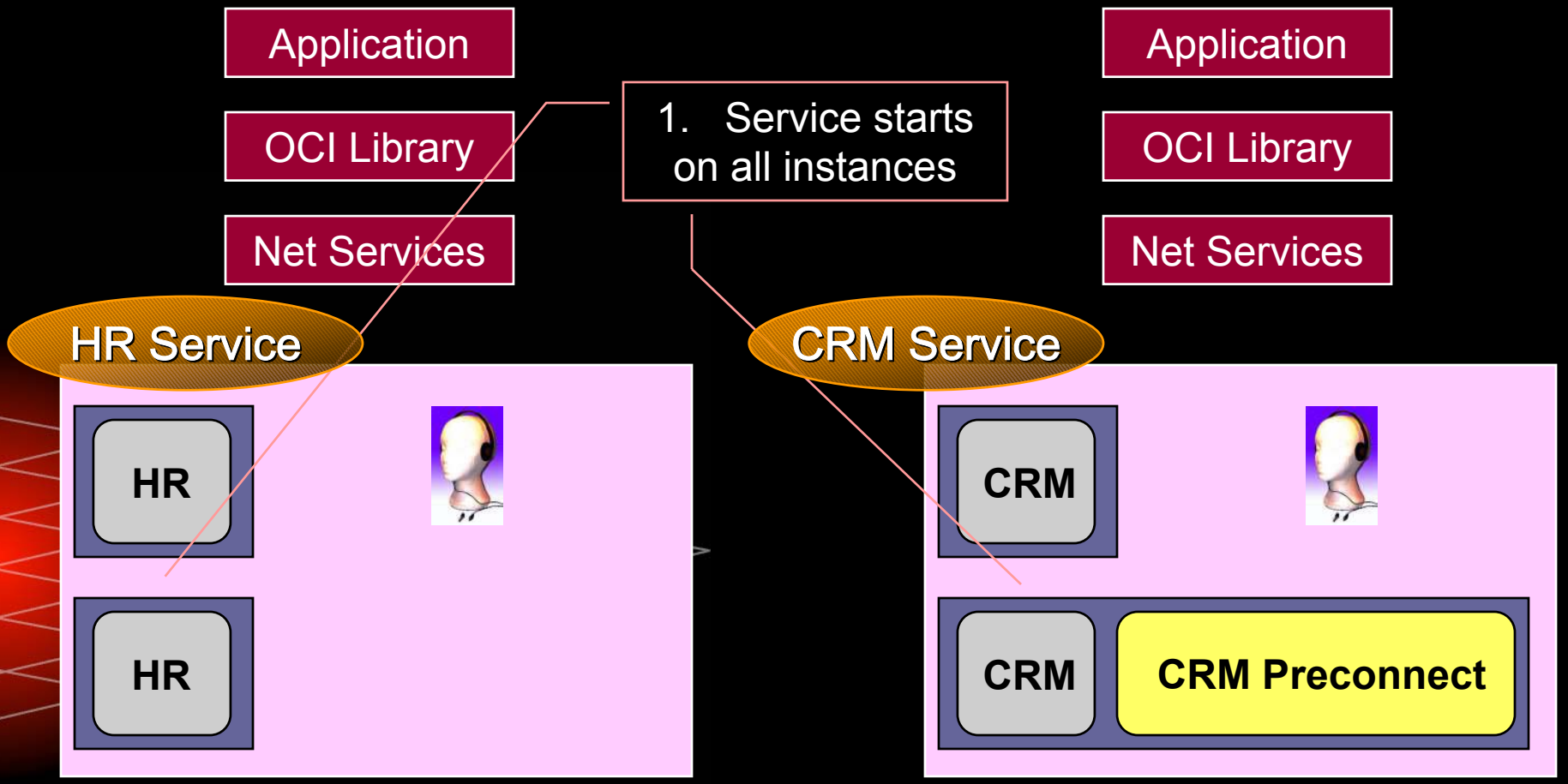
Transparent Application Failover: 10g Overview



TAF Basic

TAF Preconnect

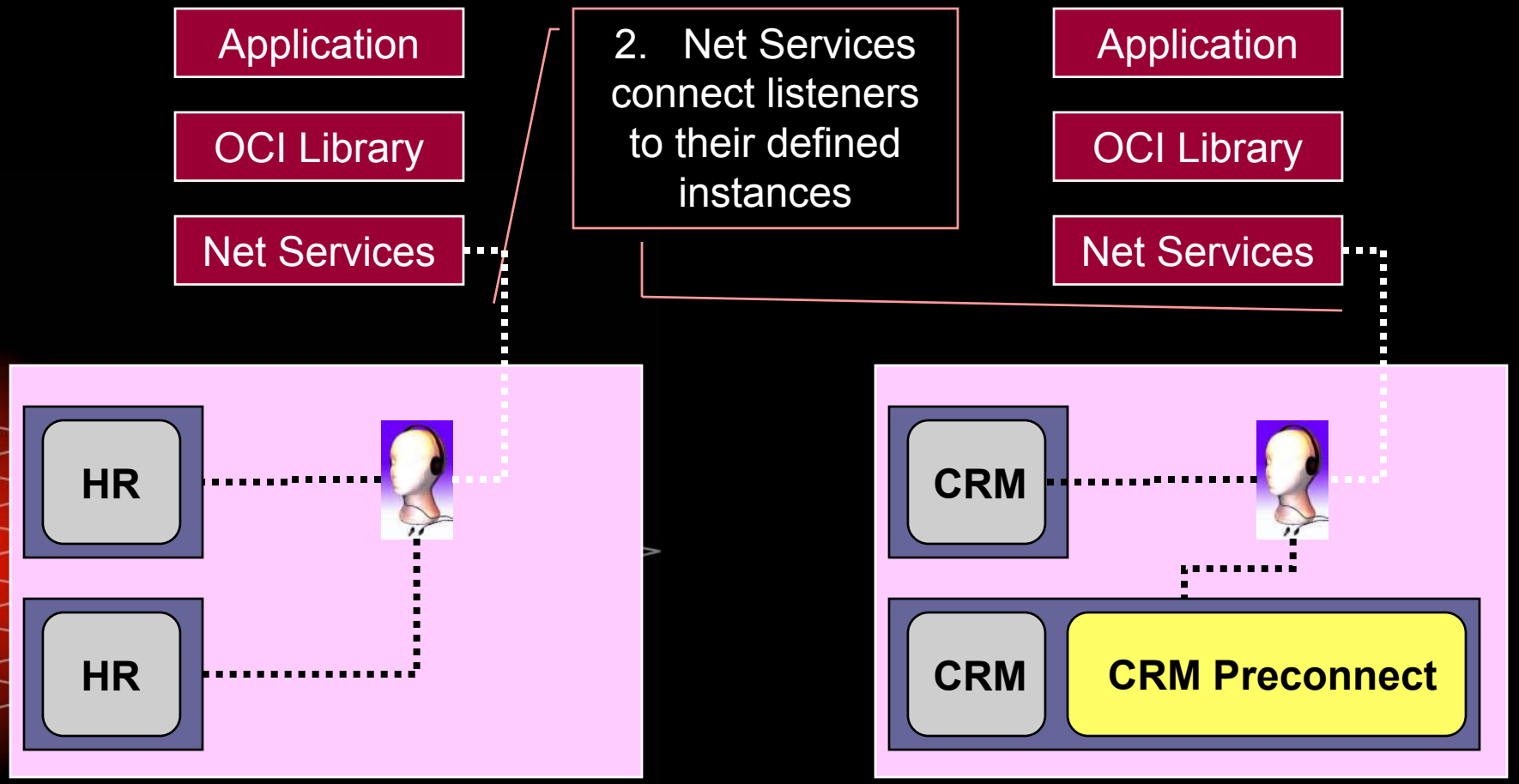
Transparent Application Failover: Overview



TAF Basic

TAF Preconnect

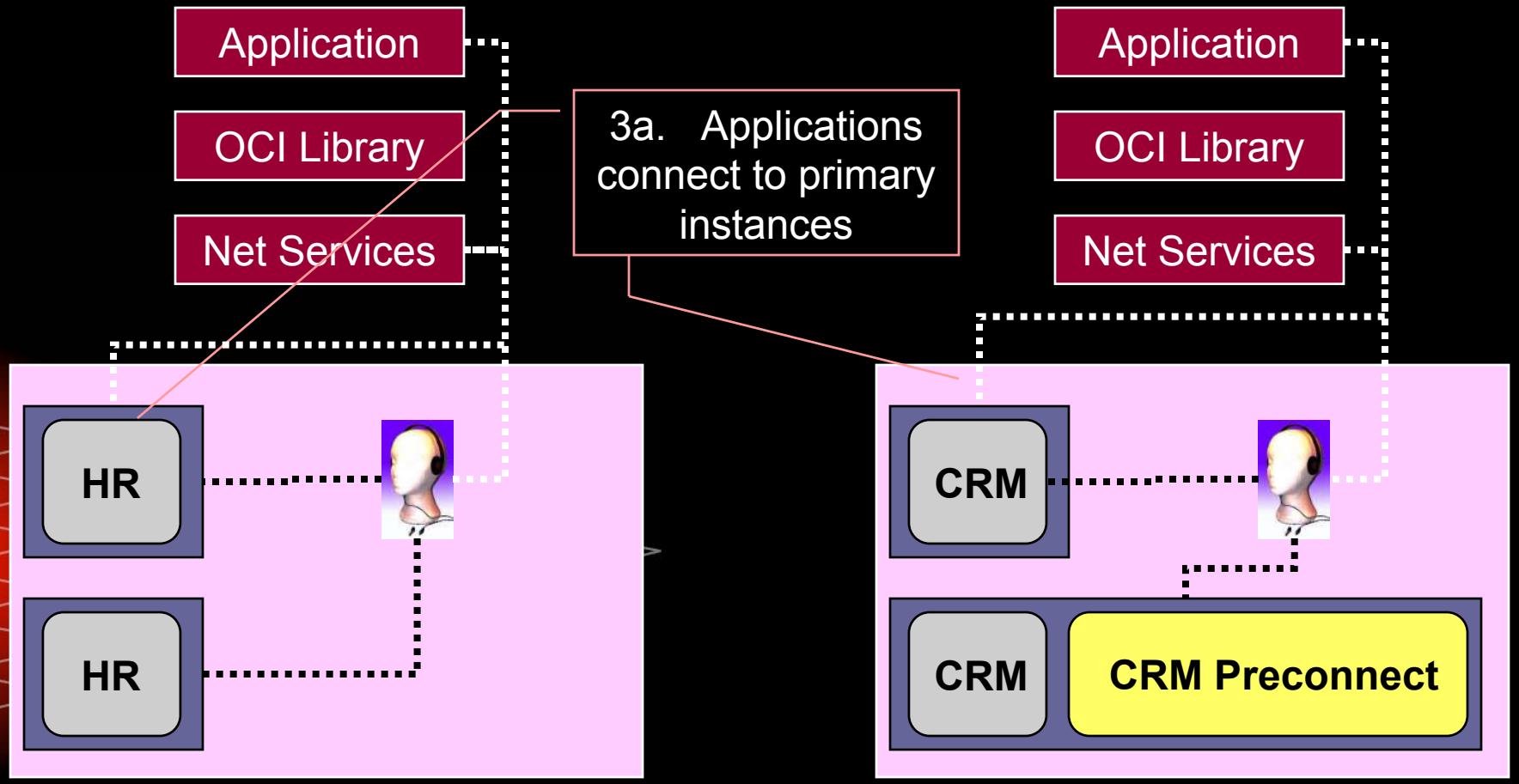
Transparent Application Failover: Overview



TAF Basic

TAF Preconnect

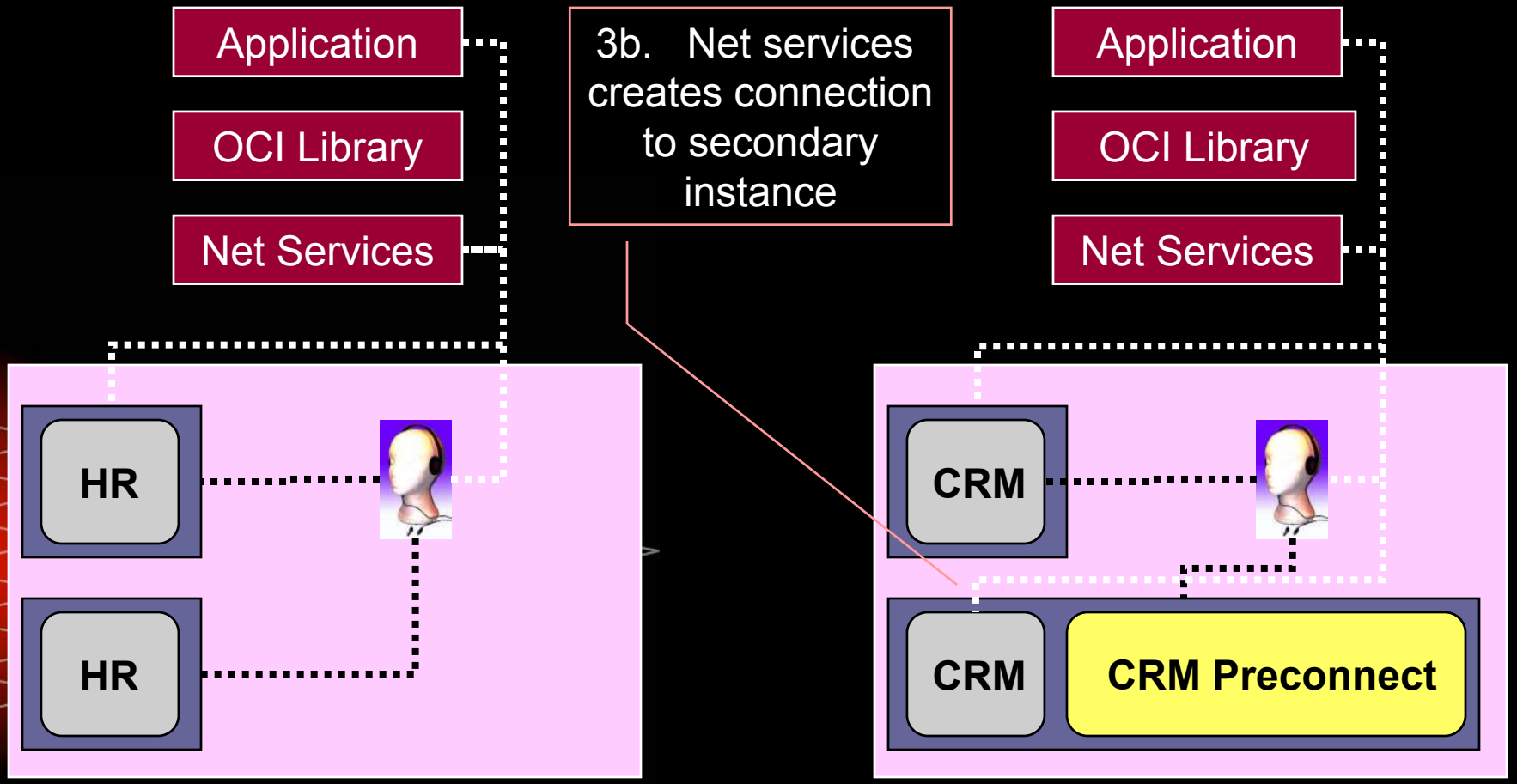
Transparent Application Failover: Overview



TAF Basic

TAF Preconnect

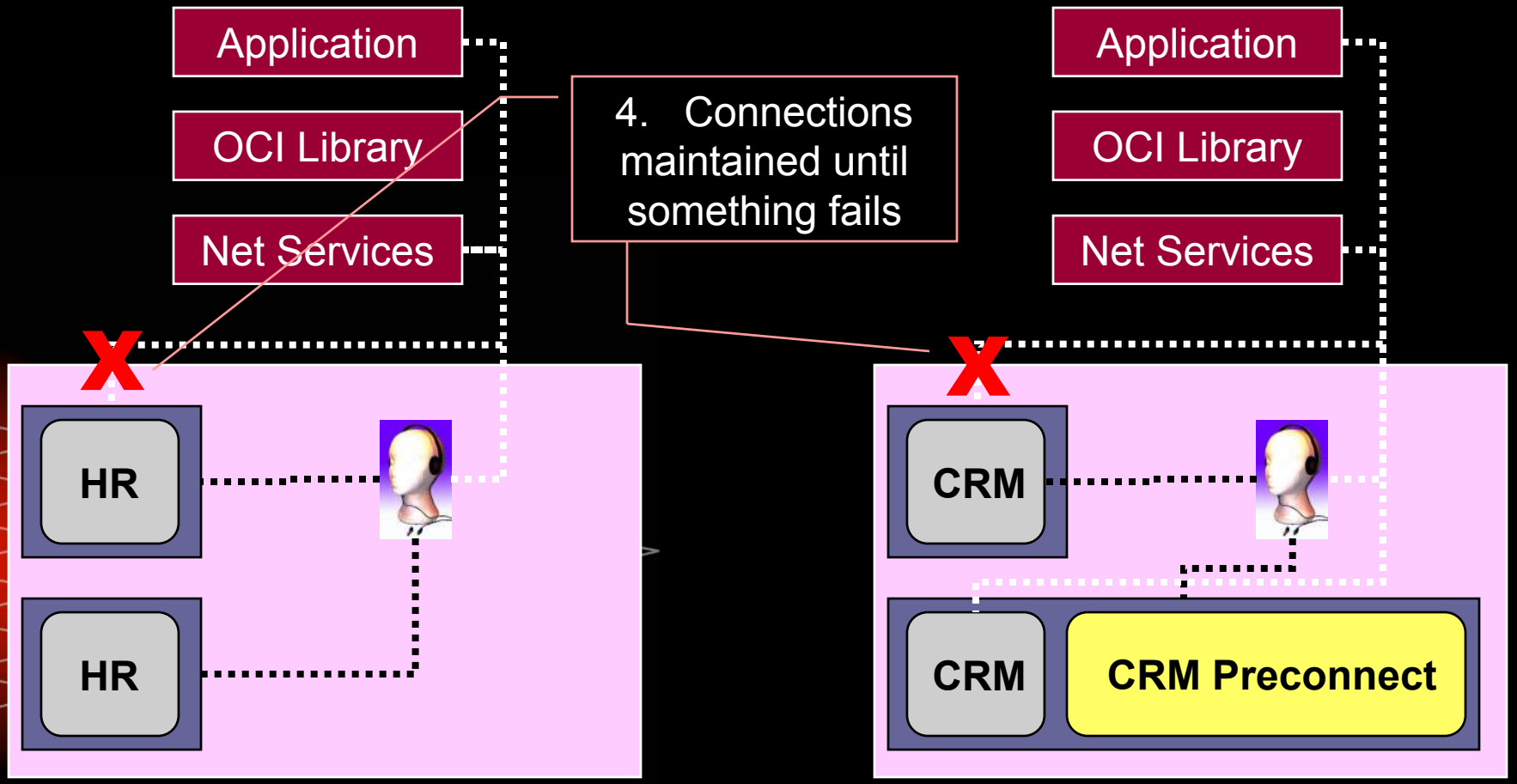
Transparent Application Failover: Overview



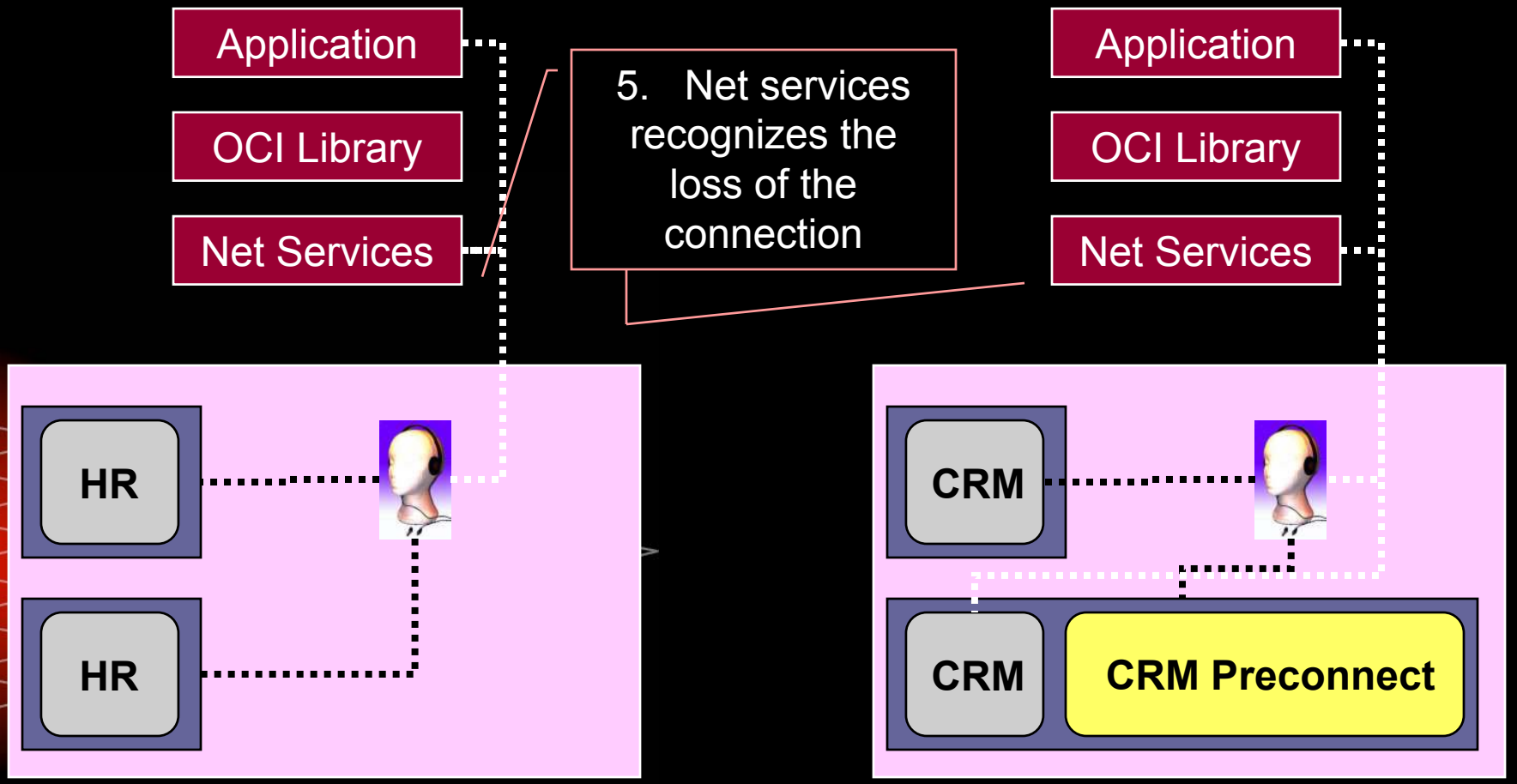
TAF Basic

TAF Preconnect

Transparent Application Failover: Overview



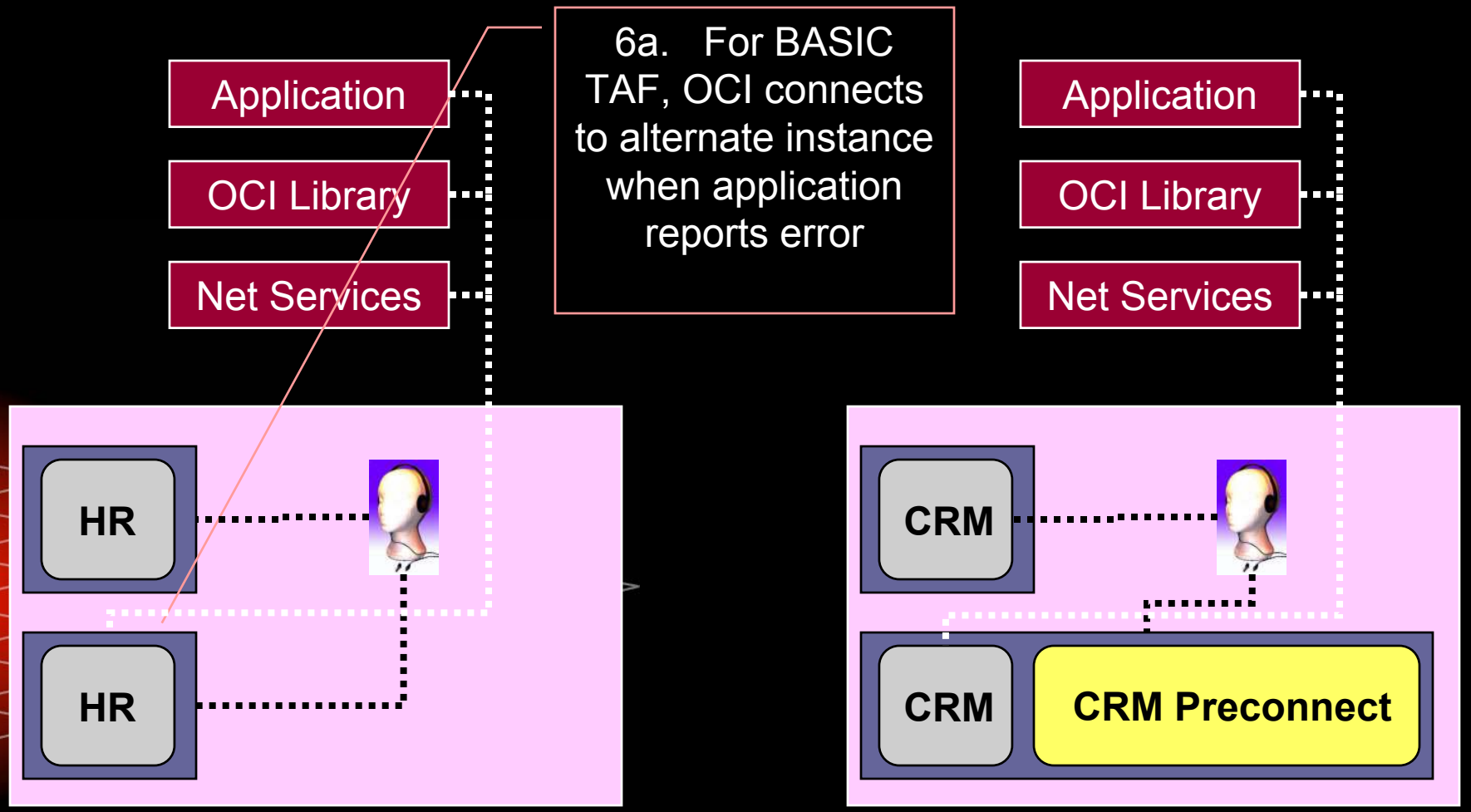
Transparent Application Failover: Overview



TAF Basic

TAF Preconnect

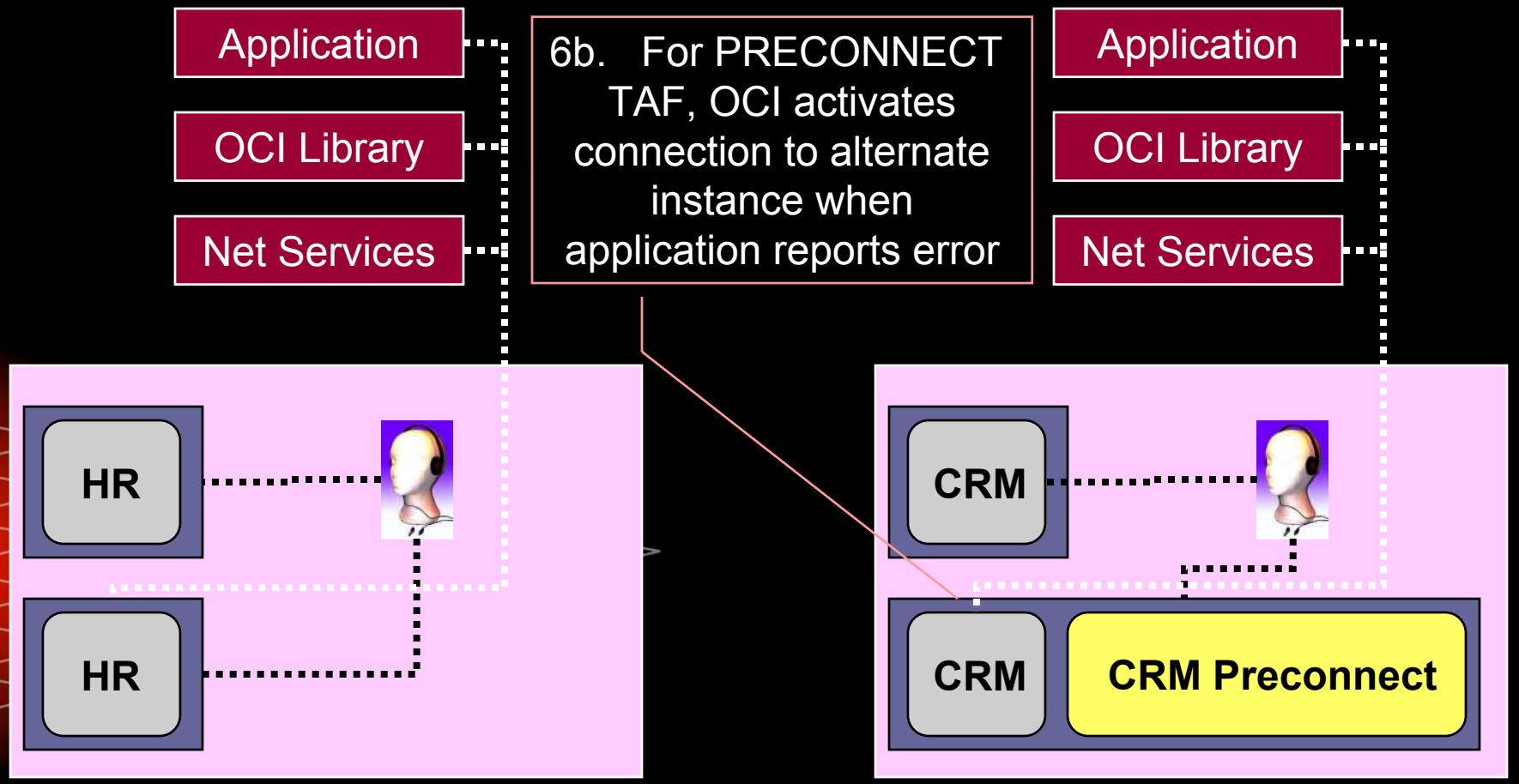
Transparent Application Failover: Overview



TAF Basic

TAF Preconnect

Transparent Application Failover: Overview

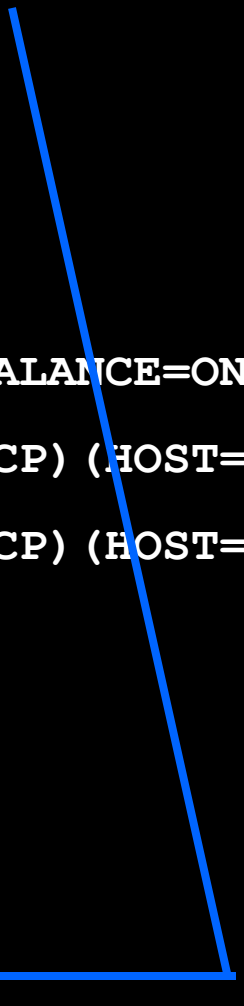


TAF Basic

TAF Preconnect

Basic TAF

```
HR =
 (DESCRIPTION =
  (FAILOVER=ON)(LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=TCP)(HOST=N1VIP)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=N2VIP)(PORT=1521))
  (CONNECT_DATA =
   (SERVICE_NAME = HR)
   (FAILOVER_MODE =
   (TYPE=SESSION)
   (METHOD=BASIC)
   (RETRIES=180)
   (DELAY=5))))
```



10g

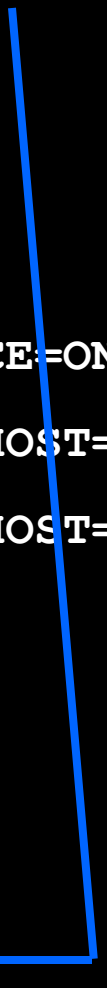
ORACLE

Considerations

- Reconnection time depends on:
  - Nature of failure
  - Net services delay
  - Number of connections to be made
- All applications switch to another instance
  - If only one, it must have reserve capacity for all users on primary instance
  - If multiple instances, logon storm may prevent load-balancing algorithms from working correctly

Preconnect TAF

```
CRM =
  (DESCRIPTION =
   (FAILOVER=ON)(LOAD_BALANCE=ON)
   (ADDRESS=(PROTOCOL=TCP)(HOST=N1VIP)(PORT=1521))
   (ADDRESS=(PROTOCOL=TCP)(HOST=N2VIP)(PORT=1521))
   (CONNECT_DATA =
    (SERVICE_NAME = CRM)
    (FAILOVER_MODE =
    (TYPE=SESSION)
    (METHOD=PRECONNECT)
    (RETRIES=180)
    (DELAY=5))))
```



10g

ORACLE

Considerations

- Reconnection time depends on:
  - Nature of failure
  - Net services delay

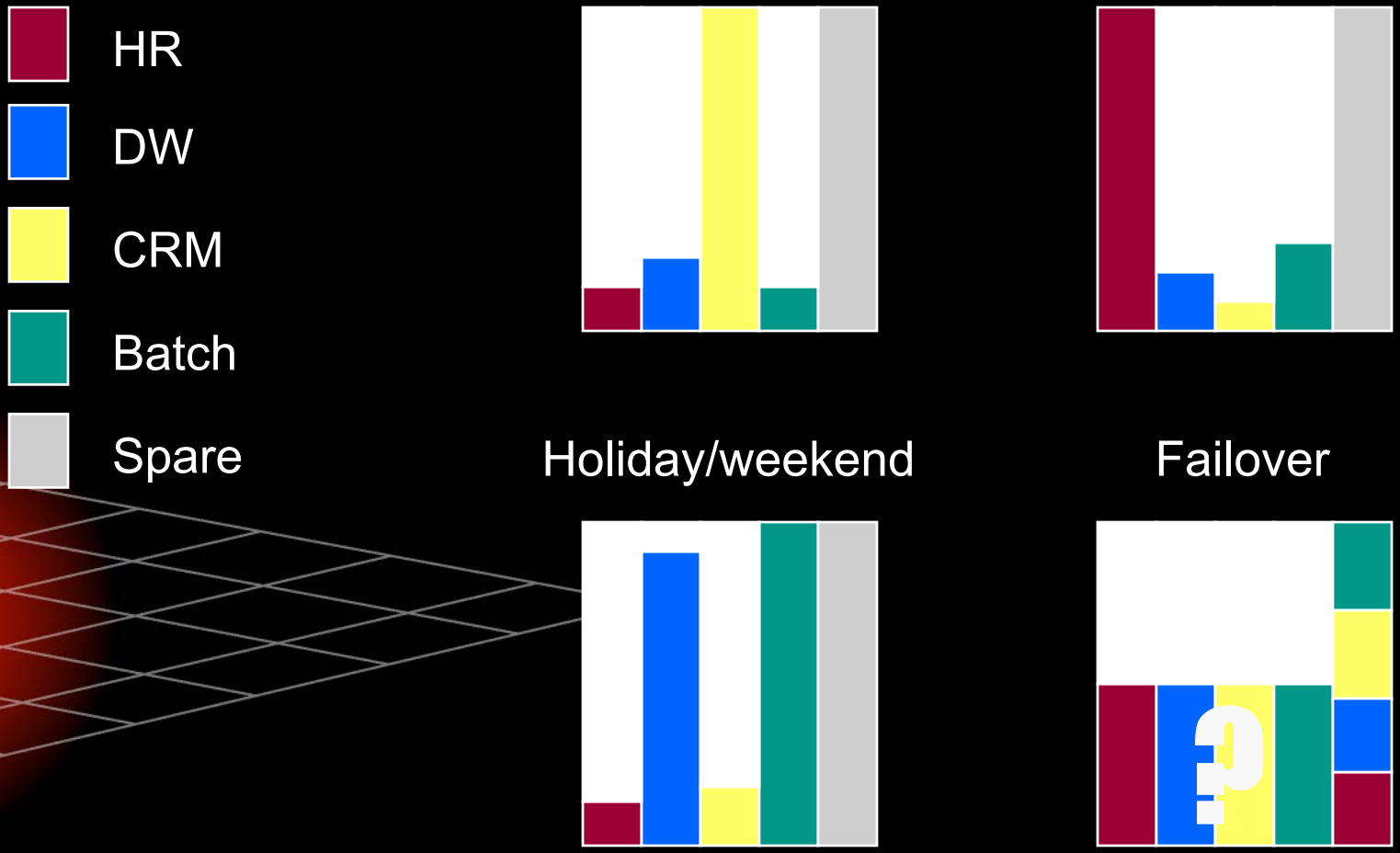
  NOTE: Not dependent on number of connections
- All applications switch to another instance
- Typically only useful in a two-instance system
- Secondary instance capacity is reduced by pre-connected sessions

How Can Services Help?

- Allow distribution of service across desired instances
- Choice of connection routing defined by type of workload
- Identify any instances for use in failovers
- Enable dynamic reallocation of application connections depending on workload

Traditional Workload Management

10g



ORACLE

Traditional Workload Management

10g

HR

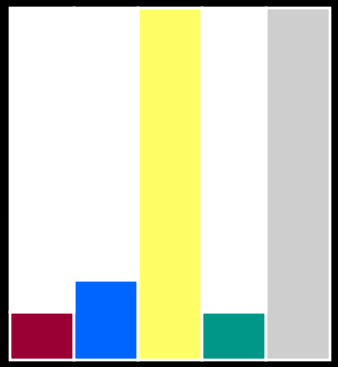
DW

CRM

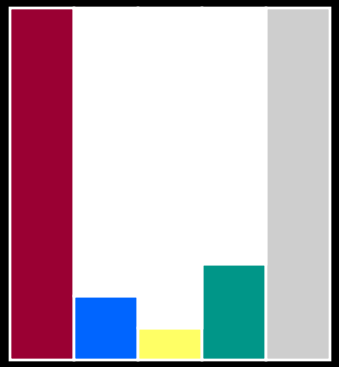
Batch

Spare

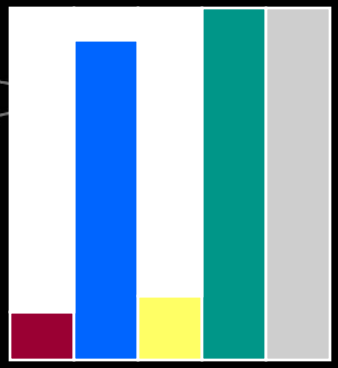
Day time



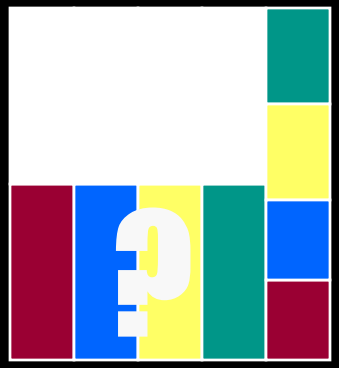
Payday



Holiday/weekend

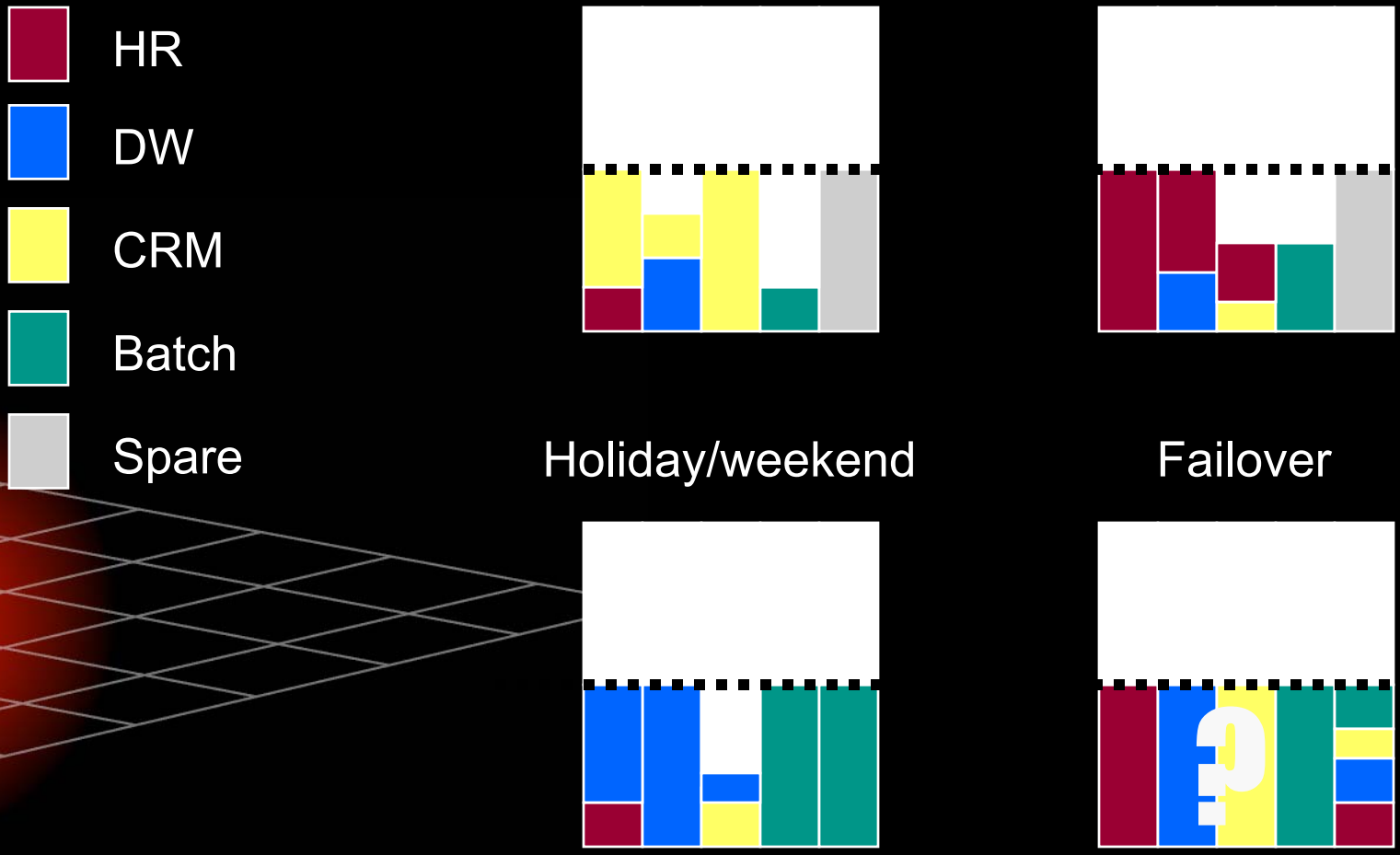


Failover



Have to over-allocate resources for each application's peak times

Grid Workload Management



Grid Workload Management

HR

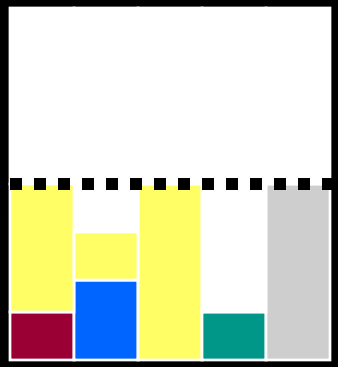
DW

CRM

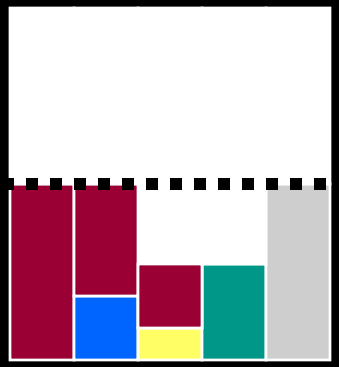
Batch

Spare

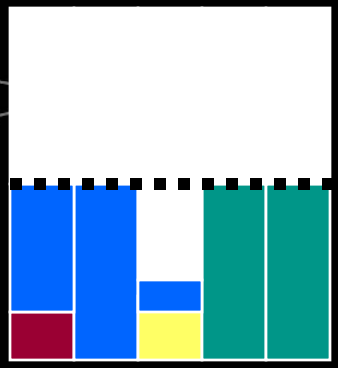
Day time



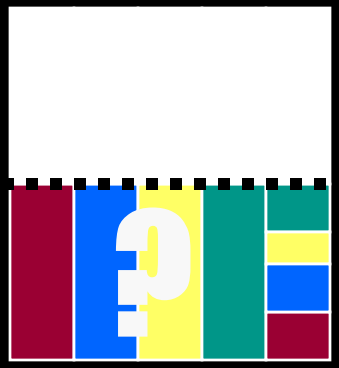
Payday



Holiday/weekend



Failover



Reduce waste but service workload with 1/2 the server resources

Workload is Managed with Services

- Are an integral part of Oracle Database 10*g*
- Group sessions doing similar work
- Provide new tuning options
- Manage multiple instances like a single instance
- Provide base for balanced and highly available connections

Attributes of a Service: Single Instance

- Identified with a globally unique name
- Network identifier to provide access by clients
- Threshold values (which you set) for response time and CPU consumption alerts
- Priority value (which you set) to determine relative resource use compared with competing services

Attributes of a Service: RAC Instances

10g

- All of single-instance attributes plus
  - Load balancing advisory goal to determine if connections are made for quality (service response time) or for best throughput (how much work is completed in a unit of time)
  - Flag to identify if service will use distributed transactions
  - Flag to turn on RAC high availability event propagation to OCI and ODP.NET clients via Advanced Queuing
  - Transparent application failover characteristics
  - Connection load balancing goal to determine if connections are made for open or closed workloads (that is, using load balancing advisory goal or using session counts)
  - High availability option by instance – primary (aka Preferred) or failover (aka Available)

Service Types and Characteristics

10g

- Application services
  - Default service created by DBCA (named after instance – full name for single instance, stem for RAC)
  - Others created as needed for workload management
  - Service names must be 64 characters or less
- Internal services
  - `SYS$BACKGROUND` (for use by background processes)
  - `SYS$USERS` (for use by sessions not assigned to an application service)
  - Cannot be deleted, changed, or disabled
- Limit of 64 application services per database (including the 2 internal services)

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basic,
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.clb_goal_long);
```

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basi
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.clb_goal_long);
```

Unique name and Net Service client connection are identical

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true
  failover_method =>
    dbms_service.failover_met
  failover_type =>
    dbms_service.failover_typ
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.cl
```

Load balancing advisory goal (quality, i.e. response time) – best for unpredictable workloads. Other options are `goal_throughput` – best for work with fixed execution times or serial processing, such as batch jobs, and `goal_none` – for applications that don't use active load balancing.

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_met
  failover_type =>
    dbms_service.failover_typ
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.clb_goal_long);
```

Indicates that the service is not for use with distributed transactions. This is a Boolean value, with `TRUE` being the other option.

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basic,
                                    >
                                    failover_type_select,
                                    s => 180,
                                    => 5.
                                    s_service.clb_goal_long);
```

HA events are to be sent via Advanced Queuing to OCI and ODP.NET clients. Set to FALSE to prevent event propagation.

Create a Service with PL/SQL

```
execute dbms_service.create_servic
  service_name => 'hr.mycompany.co
  network_name => 'hr.mycompany.co
  goal => dbms_service.goal_servic
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basic,
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.clb_goal_long);
```

This TAF method is the only one supported – Preconnected sessions are not valid for services created with PL/SQL

Create a Service with PL/SQL

```
execute dbms_service.create_ser
  service_name => 'hr.mycompany
  network_name => 'hr.mycompany
  goal => dbms_service.goal_ser
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basic,
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5.
  clb_goal => dbms_service.clb_goal_long);
```

Use this TAF type or FAILOVER_TYPE_SESSION to enable TAF for the session, which overrides client connection TAF settings. Note that TAF works only with OCI. Session type just connects, Select type resumes interrupted query.

Create a Service with PL/SQL

```
execute dbms_service.create_service(
  service_name => 'hr.mycompany.com',
  network_name => 'hr.mycompany.com',
  goal => dbms_service.goal_service_time,
  dtp => false,
  aq_ha_notifications => true,
  failover_method =>
    dbms_service.failover_method_basic,
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5,
  clb_goal => dbms_service.clb_goal_long);
```

Number of times TAF tries to reconnect and time (in seconds) to wait between each attempt.

Create a Service with PL/SQL

```
execute dbms_service.crea
  service_name => 'hr.myc
  network_name => 'hr.myc
  goal => dbms_service.go
  dtp => false,
  aq_ha_notifications =>
  failover_method =>
    dbms_service.failove
  failover_type =>
    dbms_service.failover_type_select,
  failover_retries => 180,
  failover_delay => 5,
  clb_goal => dbms_service.clb_goal_long);
```

Connection goal for long-lasting connections, such as connection pools and SQL*Forms applications. The other option is `CLB_GOAL_SHORT` for activities that will not stay connected long enough to be impacted by changing loads on the connected instance. NOTE: the `tnsnames.ora LOAD_BALANCE=ON` entry is required for these goals to be instantiated.

Other Ways to Create and Manage Services

- Additional procedures in the `DBMS_SERVICE` PL/SQL package
- Database Configuration Assistant (DBCA)
  - Create during database creation
  - Create and manage after installation from the Service Management screen under the Oracle Real Application Clusters option
- Enterprise Manager
  - Create and manage from the Cluster Managed Database Services page
- The SRVCTL tool
  - Create and manage services by database or instance
  - Requires Oracle Clusterware

SRVCTL Commands for Services: 10g Add Syntax

```
srvctl add service -d db_unique_name -s service_name
-r preferred_list [-a available_list] [-P TAF_policy]
```

where

`-d db_unique_name`	identifies the unique name for the database
`-s service_name`	identifies the service name
`-r preferred_list`	identifies the list of preferred instances
`-a available_list`	identifies the list of available instances
`-P TAF_policy`	identifies the TAF policy (NONE, BASIC, or PRECONNECT). The BASIC and PRECONNECT settings affect the content of the TNS string that Oracle generates automatically when the command is executed

SRVCTL Commands for Services: Add Example

10g

```
srvctl add service -d db_unique_name -s service_name
-r preferred_list [-a available_list] [-P TAF_policy]
```

where

`-d db_unique_name`	identifies the unique name for the database
`-s service_name`	identifies the service name
`-r preferred_list`	identifies the list of preferred instances
`-a available_list`	identifies the list of available instances
`-P TAF_policy`	identifies the TAF polic… PRECONNECT). The … settings affect the content of the TNS string that Oracle generates automatically when the command is executed

These options are identical to those in the `dbms_service` package

SRVCTL Commands for Services: 10g Add Example

```
srvctl add service -d db_unique_name -s service_name
-r preferred_list [-a ava
```

where

-d db_unique_name	identi
-s service_name	identifies the service name
-r preferred_list	identifies the list of preferred instances
-a available_list	identifies the list of available instances
-P TAF_policy	identifies the TAF policy (NONE, BASIC, or PRECONNECT). The BASIC and PRECONNECT settings affect the content of the TNS string that Oracle generates automatically when the command is executed

These options are not provided in the dbms_services package.
Preferred instances are where service always runs on startup; available instances are used for failover when preferred instances fail.

SRVCTL Commands for Services: 10g Add Syntax

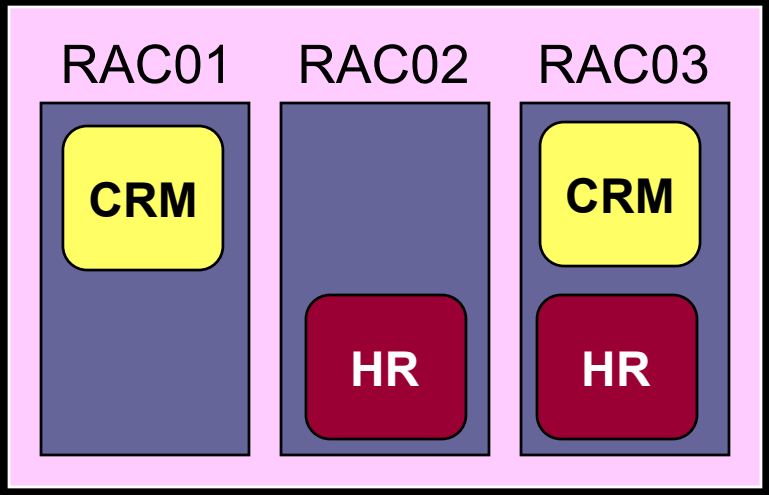
```
srvctl add service -d db_unique_name -s service_name
-r preferred_list [-a availabl
```

where

`-d db_unique_name`	identifies the
`-s service_name`	identifies the service name
`-r preferred_list`	identifies the list of preferred instances
`-a available_list`	identifies the list of available instances
`-P TAF_policy`	identifies the TAF policy (NONE, BASIC, or PRECONNECT). The BASIC and PRECONNECT settings affect the content of the TNS string that Oracle generates automatically when the command is executed

This option is provided in the `dbms_service` package but the Preconnect value is not; it is used here to create failover-ready connections to the available instances.

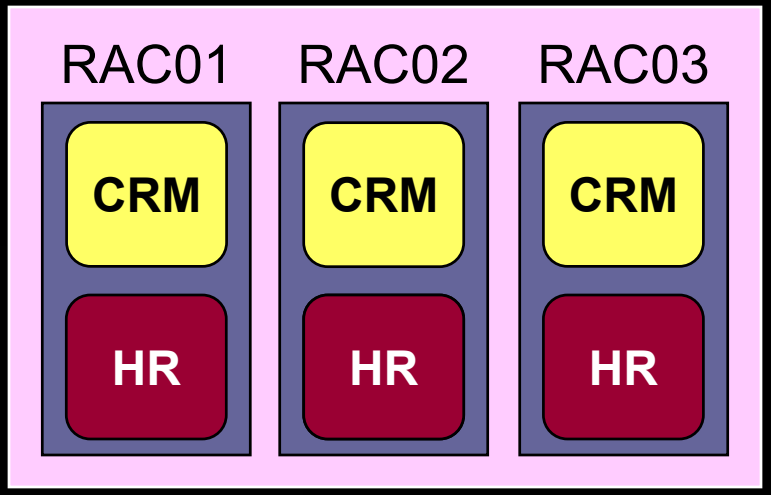
Create Active/Spare Configuration




```
srvctl add service -d ACME_DB -s CRM
-r RAC01 -a RAC03 -P PRECONNECT
```

```
srvctl add service -d ACME_DB -s HR
-r RAC02 -a RAC03 -P PRECONNECT
```

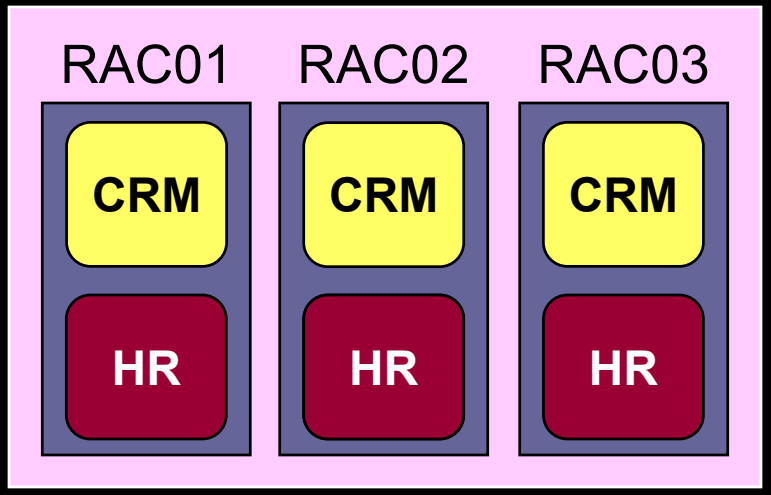
Create Active/Symmetric Configuration




```
srvctl add service -d ACME_DB -s CRM
-r ("RAC01,RAC02,RAC03")
```

```
srvctl add service -d ACME_DB -s HR
-r ("RAC01,RAC02,RAC03")
```

Create Active/Asymmetric Configuration




```
srvctl add service -d ACME_DB -s CRM -r RAC01
-a ("RAC02,RAC03") -P PRECONNECT
```

```
srvctl add service -d ACME_DB -s CRM
-r ("RAC02,RAC03") -a RAC01 -P PRECONNECT
```

Now You Have Services…

- Sessions are tracked by the services to which they connect
- Automatic Workload Repository (AWR) manages the performance of services and records the service performance, including
  - SQL execution times
  - Wait classes
  - Resources consumed by service
- AWR sends alerts when service response time thresholds are exceeded
- Database Resource Manager can prioritize application workloads within an instance by service

Now You Have Services…

- Sessions are tracked by the services to which they connect

```
CONNECT APPUSR1/pass1@CRM
```

```
CONNECT APPUSR2/pass2@HR
```

Now You Have Services…

- Sessions are tracked by the services to which they connect
- Automatic Workload Repository (AWR) manages the performance of services and records the service performance, including
  - SQL execution times
  - Wait classes
  - Resources consumed by service
- AWR sends alerts when service response time thresholds are exceeded
- Database Resource Manager can prioritize application workloads within an instance by service

Now You Have Services…

- Sessions are tracked by the services to which they connect
- Automatic Workload Repository (AWR) manages the performance of services and records the service performance, including
  - SQL execution times
  - Wait classes
  - Resources consumed by service
- AWR sends alerts when service response time thresholds are exceeded
- Database Resource Manager can prioritize application workloads within an instance by service

Services and Alerts

Set thresholds on each instance supporting the service

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

Can also be CPU_TIME_PER_CALL

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

Can also be CPU_TIME_PER_CALL

Warning when call time exceeds 0.5 secs (500000 msecs)

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

10g

Can also be CPU_TIME_PER_CALL

Warning when call time exceeds 0.5 secs (500000 msecs)

Averaged over 15 minutes

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

Can also be CPU_TIME_PER_CALL

Warning when call time exceeds 0.5 secs. (500000 msecs.)

Averaged over 15 minutes

Three times in a row

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

Critical alert when 15 minute avg call time exceeds 0.75 secs. In three consecutive periods

```
exec DBMS_SERVER_ALERT.SET_THRESHOLD
(METRICS_ID => dbms_server_alert.elapsed_time_per_call,
WARNING_OPERATOR => dbms_server_alert.operator_ge,
WARNING_VALUE => '500000',
CRITICAL_OPERATOR => dbms_server_alert.operator_ge,
CRITICAL_VALUE => '750000',
OBSERVATION_PERIOD => 15,
CONSECUTIVE_OCCURRENCES => 3,
INSTANCE_NAME => 'APP1',
OBJECT_TYPE => dbms_server_alert.object_type_service,
OBJECT_NAME => 'CRM');
```

Services and Alerts

- Alert messages written to log
- Visible through Enterprise Manager screens
- Use alert information to change service allocation, for example by
  - Relocating service to faster server
  - Adding new instances for service
  - Removing lower priority service from server
  - Changing priority of workload (discussed in next section)

Now You Have Services…

- Sessions are tracked by the services to which they connect
- Automatic Workload Repository (AWR) manages the performance of services and records the service performance, including
  - SQL execution times
  - Wait classes
  - Resources consumed by service
- AWR sends alerts when service response time thresholds are exceeded
- Database Resource Manager can prioritize application workloads within an instance by service

Services with Resource Manager

10g

```
execute DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
  (CONSUMER_GROUP => 'HIGH_PRIORITY')
exec DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
  (ATTRIBUTE => DBMS_RESOURCE_MANAGER.SERVICE_NAME,
   VALUE => 'CRM',CONSUMER_GROUP => 'HIGH_PRIORITY')
```

```
execute DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
  (CONSUMER_GROUP => 'LOW_PRIORITY')
exec DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
  (ATTRIBUTE => DBMS_RESOURCE_MANAGER.SERVICE_NAME,
   VALUE => 'HR',CONSUMER_GROUP => 'LOW_PRIORITY')
```

Services with Resource Manager

10g

```
CRM =
  (DESCRIPTION =
   (ADDRESS=(PROTOCOL=TCP)(HOST=ACME1)(PORT=1521))
   (CONNECT_DATA = (SERVICE_NAME = CRM)))
HR =
  (DESCRIPTION =
   (ADDRESS=(PROTOCOL=TCP)(HOST=ACME1)(PORT=1521))
   (CONNECT_DATA = (SERVICE_NAME = HR)))
```

```
CONNECT APPUSR1/pass1@CRM
```

```
CONNECT APPUSR2/pass2@HR
```

CRM — HIGH_PRIORITY

HR — LOW_PRIORITY

Now You Have Services…

- Jobs can now run under a service, as opposed to a specific instance
- Parallel slave processes inherit the service of their coordinator
- The RAC High Availability framework keeps services available within a site
- Dynamic performance views report current service status with one hour of history
- Performance-related statistics and wait events tracked by services
- Data Guard Broker migrates the primary service across Data Guard sites for disaster tolerance

Services with Job Scheduler

10g

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
SERVICE => 'CRM',
RESOURCE_CONSUMER_GROUP => 'HIGH_PRIORITY',
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGING_RUNS,
LOG_HISTORY => 30, COMMENTS => 'P1 batch');
```

```
DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_CLASS => 'HOT_BATCH_CLASS', ENABLED => TRUE,
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.my_proc();',
NUMBER_OF_ARGUMENTS => 4, START_DATE =>SYSDATE+1,
REPEAT_INTERVAL => 5, END_DATE => SYSDATE+30,
AUTO_DROP => false, COMMENTS => 'daily status');
```

Services with Job Scheduler

10g

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
SERVICE => 'CRM',
RESOURCE_CONSUMER_GROUP => 'HIGH_PRIORITY',
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGING_RUNS,
LOG_HISTORY => 30, COMMENTS => 'P1 batch');
```

```
DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_CLASS => 'HOT_BATCH_CLASS', ENABLED => TRUE,
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.
NUMBER_OF_ARGUMENTS =>
REPEAT_INTERVAL => 5, E
AUTO_DROP => false, COM
```

JOB TABLE

JOB	JOB CLASS	SERVICE
Job 1	HOT_BATCH_CLASS	CRM
Job 2	HOT_BATCH_CLASS	CRM
Job 3	LOW_BATCH_CLASS	HR

Services with Job Scheduler

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
SERVICE => 'CRM',
RESOURCE_CONSUMER_GROUP => 'HIGH_PRIO
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGING_RUNS,
LOG_HISTORY => 30, COMMENTS => 'P1 batch');
```

```
DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_CLASS => 'HOT_BATCH_CLASS', ENABLED => TRUE,
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.
NUMBER_OF_ARGUMENTS =>
REPEAT_INTERVAL => 5, E
AUTO_DROP => false, CO
```

A job is associated with a service

JOB TABLE

JOB	JOB CLASS	SERVICE
Job 1	HOT_BATCH_CLASS	CRM
Job 2	HOT_BATCH_CLASS	CRM
Job 3	LOW_BATCH_CLASS	HR

Services with Job Scheduler

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
SERVICE => 'CRM',
RESOURCE_CONSUMER_GROUP => 'HIGH_PRIO
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGI
LOG_HISTORY => 30, COMMENTS => 'P1 ba
```

```
DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_CLASS => 'HOT_BATCH_CLASS', ENABLED => TRUE,
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.
NUMBER_OF_ARGUMENTS =>
REPEAT_INTERVAL => 5,
AUTO_DROP => false, CO
```

A job is associated with a service through its affiliation with a job class

JOB TABLE

JOB	JOB CLASS	SERVICE
Job 1	HOT_BATCH_CLASS	CRM
Job 2	HOT_BATCH_CLASS	CRM
Job 3	LOW_BATCH_CLASS	HR

Services with Job Scheduler

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
SERVICE => 'CRM',
RESOURCE_CONSUMER_GROUP => 'HIGH_PRIO
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGI
LOG_HISTORY => 30, COMMENTS => 'P1 ba

DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_CLASS => 'HOT_BATCH_CLASS', ENABL
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.
NUMBER_OF_ARGUMENTS =>
REPEAT_INTERVAL => 5, E
AUTO_DROP => false, COM
```

A job is associated with a service

through its affiliation with a job class

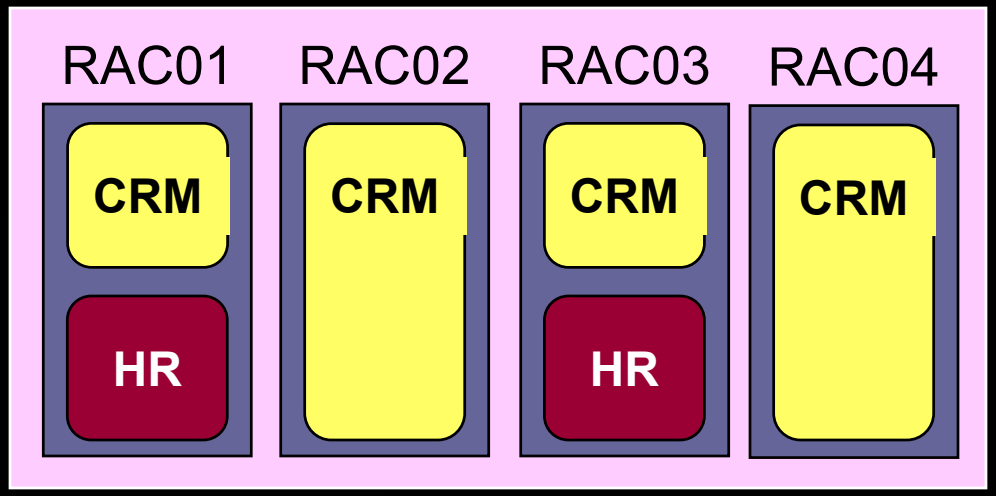
which was defined on a service

JOB	JOB	
Job 1	HOT_BATCH_CLASS	CRM
Job 2	HOT_BATCH_CLASS	CRM
Job 3	LOW_BATCH_CLASS	HR

Now You Have Services…

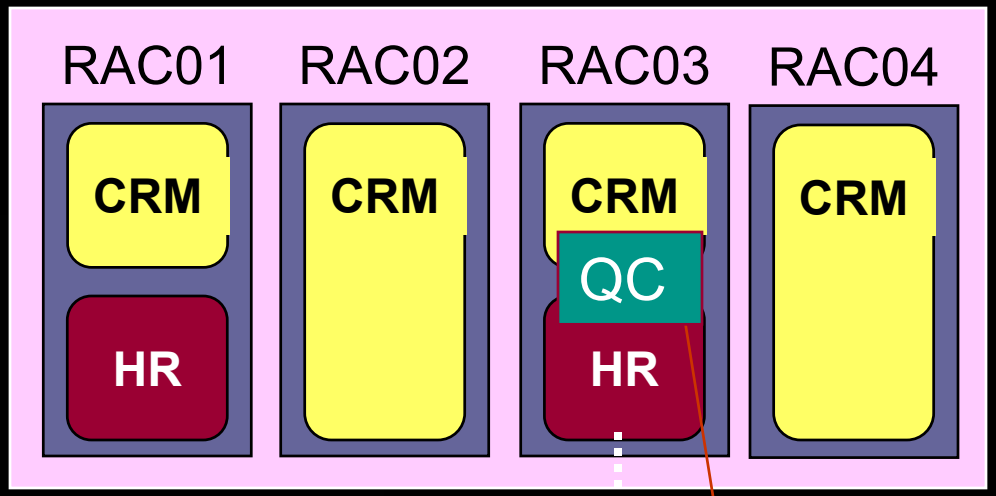
- Jobs can now run under a service, as opposed to a specific instance
- Parallel slave processes inherit the service of their coordinator
- The RAC High Availability framework keeps services available within a site
- Dynamic performance views report current service status with one hour of history
- Performance-related statistics and wait events tracked by services
- Data Guard Broker migrates the primary service across Data Guard sites for disaster tolerance

Services with Parallel Operations




```
CONNECT APPUSR2/pass2@HR
SELECT * FROM huge_table...
```

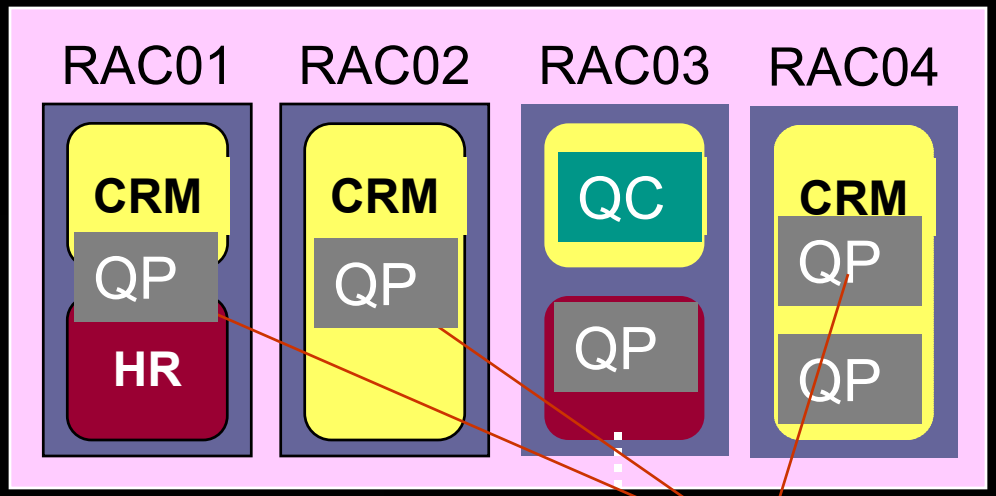
Services with Parallel Operations




```
CONNECT APPUSR2/pass2@HR
SELECT * FROM huge_table...
```

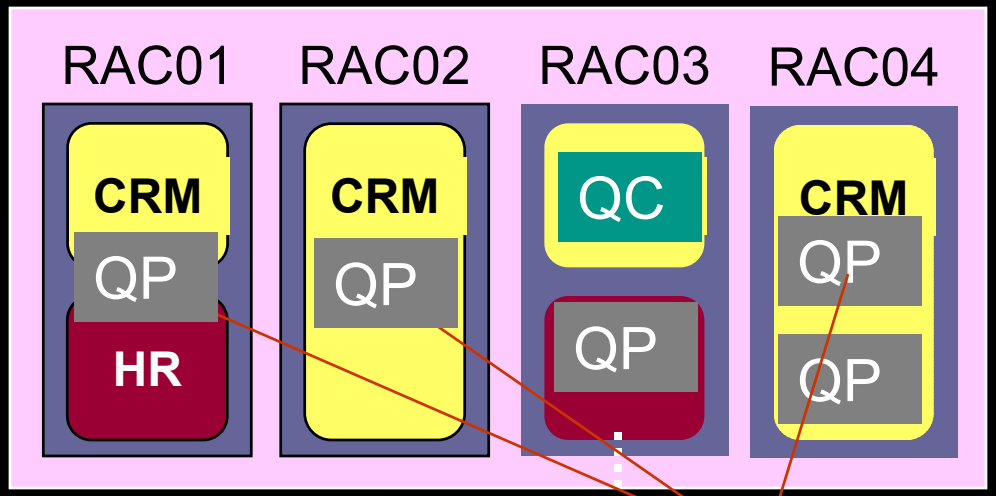
Connects to preferred instance for HR where query coordinator will run

Services with Parallel Operations



Query coordinator can spawn query processes on any running instance – whether it is preferred, available or neither for the attached service

Services with Parallel Operations



This is because the query process inherits the service of the coordinator, so the service need not be present on the target instance

Now You Have Services…

- Jobs can now run under a service, as opposed to a specific instance
- Parallel slave processes inherit the service of their coordinator
- Performance-related statistics and wait events tracked by services
- Dynamic performance views report current service status with one hour of history
- The RAC High Availability framework keeps services available within a site
- Data Guard Broker migrates the primary service across Data Guard sites for disaster tolerance

Services with Tracing

10g

- **Database-wide tracing:**

```
dbms_monitor.DATABASE_TRACE_ENABLE(
      waits                IN BOOLEAN DEFAULT TRUE,
      binds                IN BOOLEAN DEFAULT FALSE,
      instance_name        IN VARCHAR2 DEFAULT NULL)
```

```
dbms_monitor.DATABASE_TRACE_DISABLE(
      instance_name        IN VARCHAR2 DEFAULT NULL)
```

Trace waits

Trace binds

Apply only to this RAC instance

Services with Tracing

10g

- Database-wide tracing:

```
DBMS_MONITOR.DATABASE_TRACE_ENABLE(
      waits                IN BOOLEAN DEFAULT TRUE,
      binds                IN BOOLEAN DEFAULT FALSE,
      instance_name        IN VARCHAR2 DEFAULT NULL)
```

```
DBMS_MONITOR.DATABASE_TRACE_DISABLE(
      instance_name        IN VARCHAR2 D
```

- Session level tracing:

```
DBMS_MONITOR.DATABASE_TRACE_ENABLE(
      session_id     IN BINARY_INTEGER DEFAULT NULL,
      serial_num     IN BINARY_INTEGER DEFAULT NULL,
      waits          IN BOOLEAN DEFAULT TRUE,
      binds          IN BOOLEAN DEFAULT FALSE)
```

Trace waits

Trace binds

Apply only to this RAC instance

Disable requires only session ID and serial number

Services with Tracing

10g

Trace waits

Trace binds

Apply only to this RAC instance

- **Service level tracing:**

```
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE(
      service_name  IN VARCHAR2)
```

```
DBMS_MONITOR.SERV_MOD_ACT_TRACE_DISABLE(
      service_name  IN VARCHAR2)
```

- **Service level tracing with module & action:**

```
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE(
      service_name   IN VARCHAR2,
      module_name    IN VARCHAR2 DEFAULT ANY_MODULE,
      action_name    IN VARCHAR2 DEFAULT ANY_ACTION,
      waits          IN BOOLEAN DEFAULT TRUE,
      binds          IN BOOLEAN DEFAULT FALSE,
      instance_name  IN VARCHAR2 DEFAULT NULL)
```

Services with Tracing

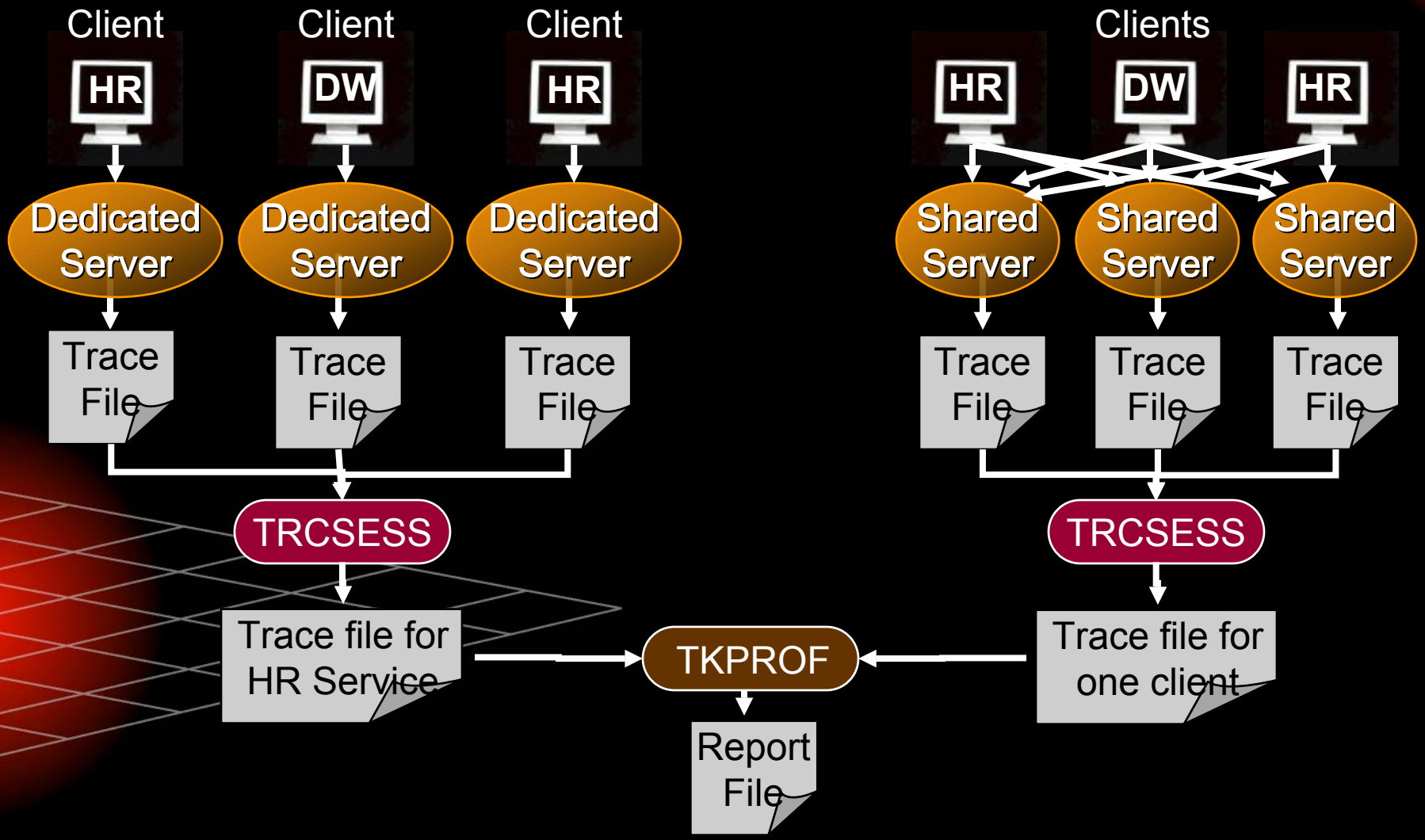
10g

- Hierarchical statistics gathering under a service:

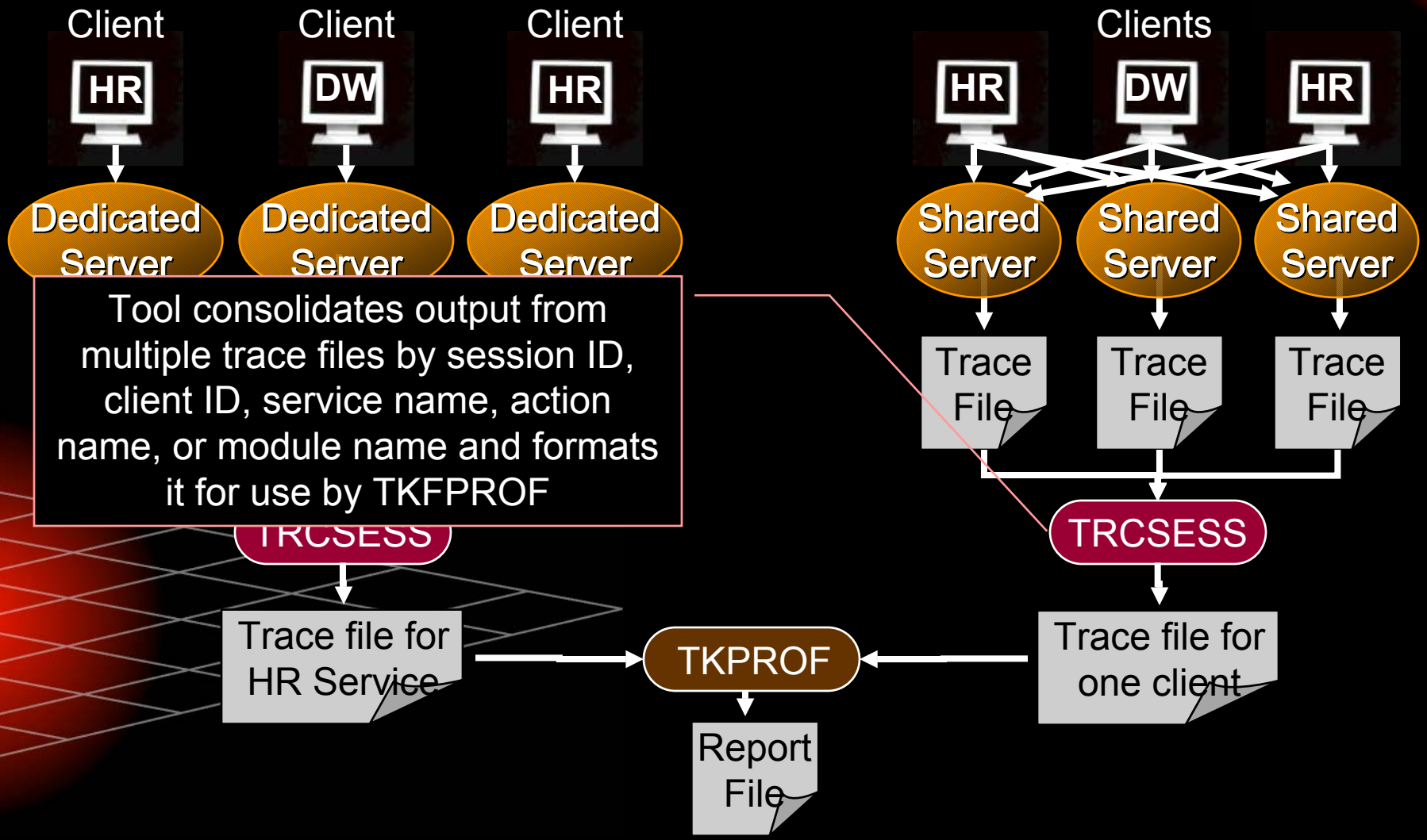
```
DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(
       service_name  IN VARCHAR2,
       module_name   IN VARCHAR2,
       action_name   IN VARCHAR2 DEFAULT ALL_ACTIONS)
```

- Useful for tuning systems using shared sessions
- `MODULE` and `ACTION` names are set by the application by using the `DBMS_APPLICATION_INFO` package or OCI calls
- To analyze performance from aggregated statistics, use the `trcsess` tool

Services with Tracing: trcsess



Services with Tracing: trcsess



Now You Have Services…

- Jobs can now run under a service, as opposed to a specific instance
- Parallel slave processes inherit the service of their coordinator
- Performance-related statistics and wait events tracked by services
- Dynamic performance views report current service status with one hour of history
- The RAC High Availability framework keeps services available within a site
- Data Guard Broker migrates the primary service across Data Guard sites for disaster tolerance

Services with Performance Views

10g

- Service, module, and action information in:
  - V$SESSION
  - V$ACTIVE_SESSION_HISTORY
- Call times and performance statistics listed in:
  - V$SERVICE_STATS
  - V$SERVICE_EVENT
  - V$SERVICE_WAIT_CLASS
  - V$SERVICEMETRIC
  - V$SERVICEMETRIC_HISTORY
- V$SERV_MOD_ACT_STATS shows performance measures for each instance when statistics are collected by module and action

Services with Performance Views

10g

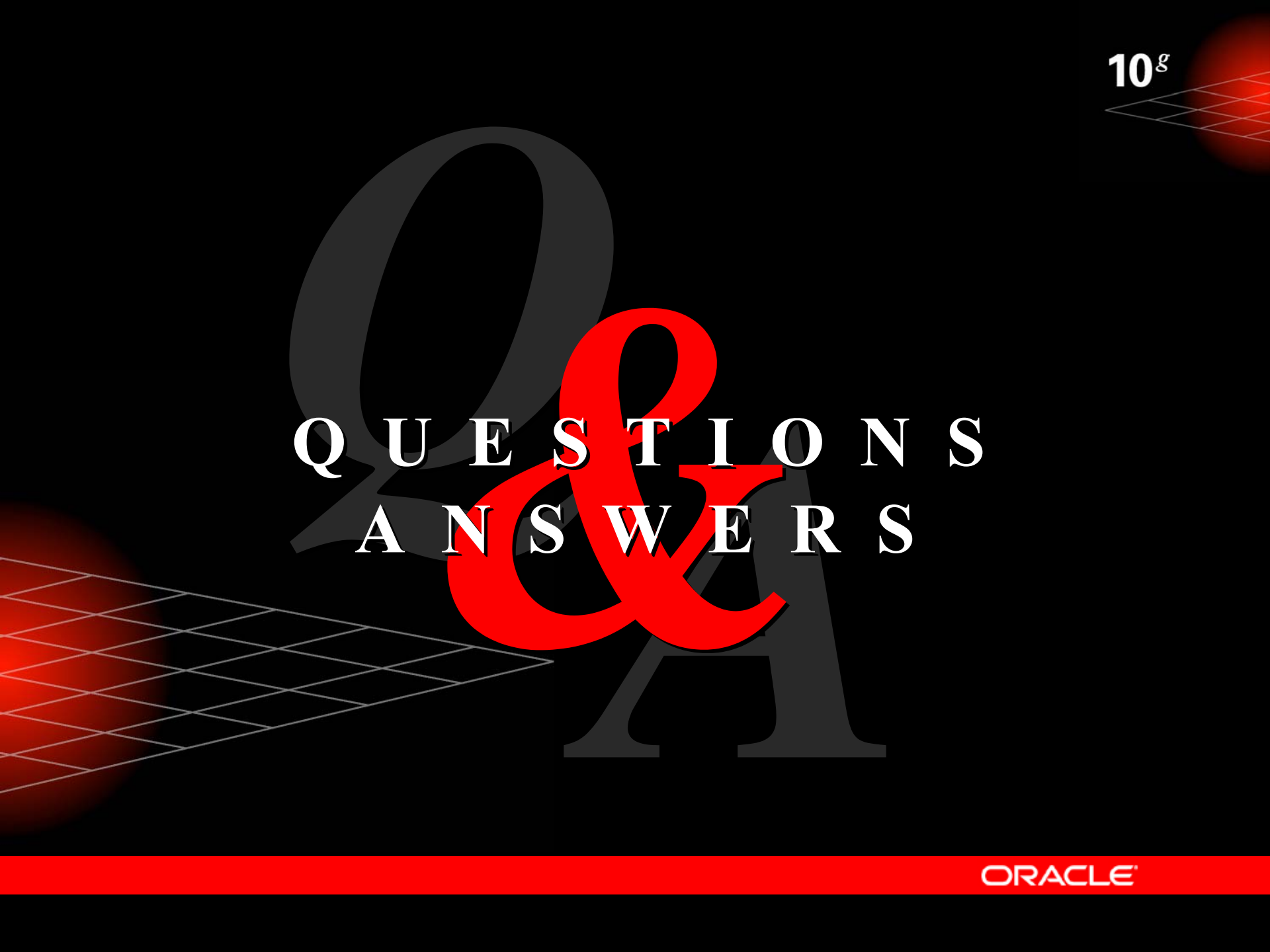
- V$SYSSTAT
  - Of its 300 performance-related statistics 28 are tracked for services
  - To see the statistics measured for services, run

```
SELECT DISTINCT stat_name
FROM v$service_stats
```

- DBA_ENABLED_AGGREGATIONS **displays information about enabled on-demand statistic aggregation**
- DBA_ENABLED_TRACES **displays information about enabled traces**

Now You Have Services…

- Jobs can now run under a service, as opposed to a specific instance
- Parallel slave processes inherit the service of their coordinator
- Performance-related statistics and wait events tracked by services
- Dynamic performance views report current service status with one hour of history
- The RAC High Availability framework keeps services available within a site
- Data Guard Broker migrates the primary service across Data Guard sites for disaster tolerance


10g
QUESTIONS
&
ANSWERS
ORACLE