



Oracle Real Application Clusters Load Balancing and Failover Options

An IT Convergence presentation by Dan Norris

Agenda

Brief RAC and Grid Concepts Review

Services & Workload Management in 10g RAC

Fast Application Notification (FAN)

Fast Connection Failover (FCF)

Transparent Application Failover (TAF)

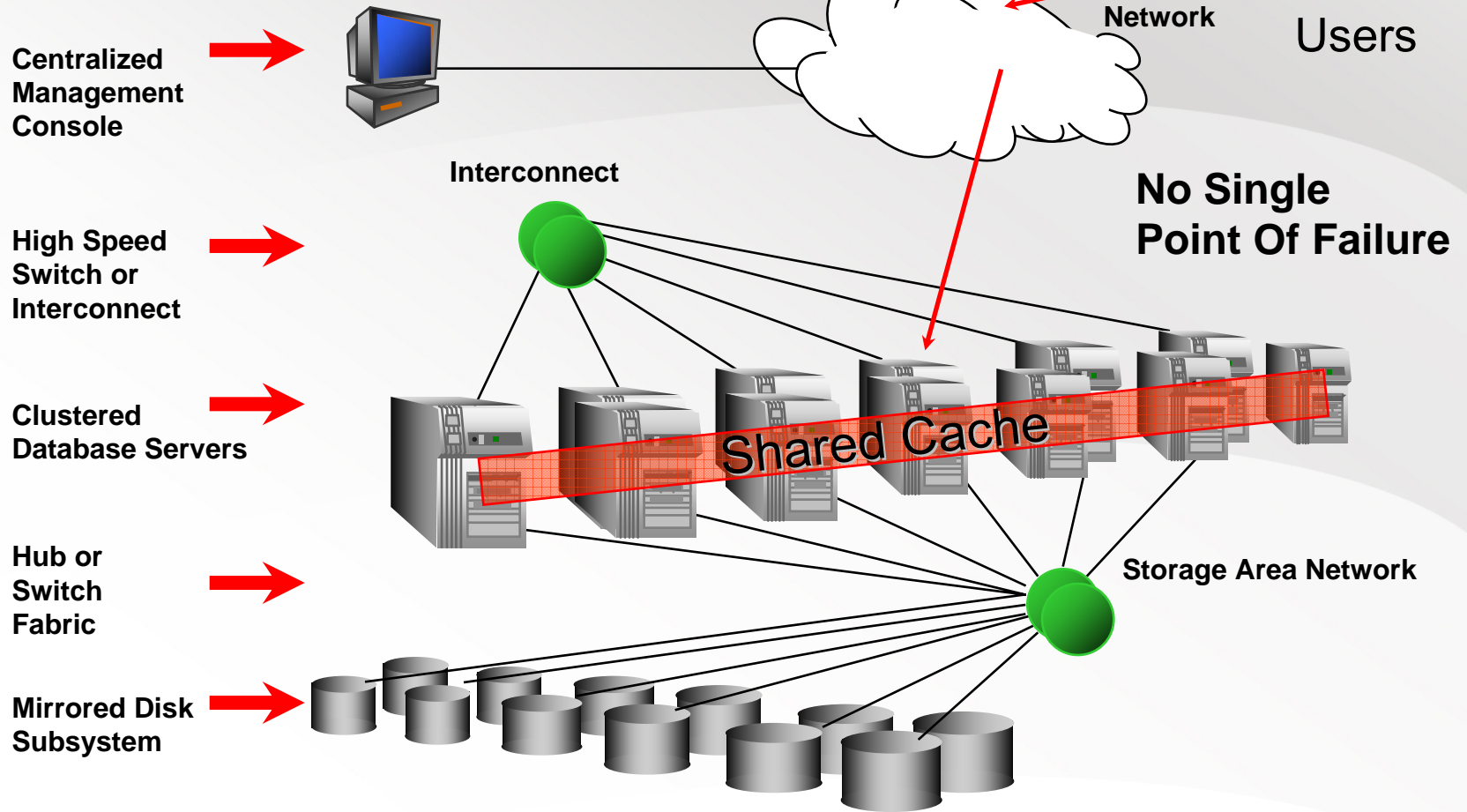
Load Balancing in RAC: Client, Server

Connection pools and load balancing with RAC 10g

Miscellaneous Configuration Tips

Next Steps & References

RAC: The Cluster Database

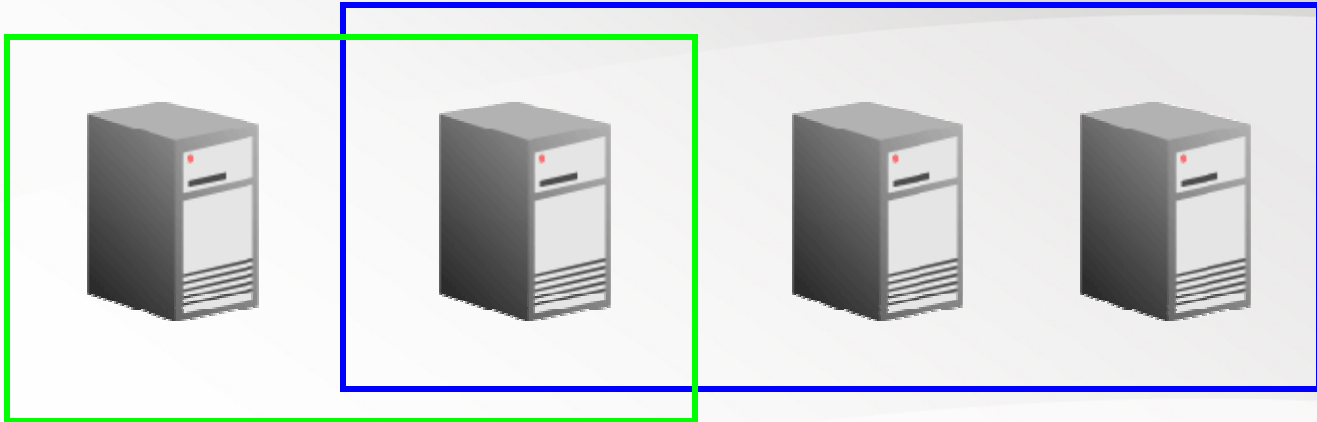


RAC Concepts

Finance Grid

HR Grid

Services



Nodes



Service
Members

VIPs: Why?

- ▶ Protects clients from long TCP/IP timeouts (can be >10 minutes)
- ▶ Transparent to clients during normal operations
- ▶ During failure event, doesn't accept or maintain connections, but allows clients to receive a response (a negative one) causing them to fail to next address in their ADDRESS_LIST
- ▶ Note that each node in a cluster has its own node VIP
- ▶ Node VIP ≠ Application VIP

Services & Workload Management

- ▶ A service is an entity to which users connect
- ▶ Usually designates a module or application used by a specific group of users
- ▶ Technically, a service is listed in the `service_name` parameter for an instance (Note: You should not edit the `service_name` parameter in a RAC environment.)
- ▶ Clusterware processes alter the `service_name` parameter on the fly to relocate services (according to policies)
- ▶ Stats in 10g are also gathered per service

Services & Workload Management



- ▶ Services can be available via one or more instances
- ▶ Failover policies and load balancing goals are set per service

Managing Services

- ▶ Services can be managed via:
 - ▶ Oracle Enterprise Manager
 - ▶ Can manage all service attributes
 - ▶ DBMS_SERVICE
 - ▶ Allows you to set service advisory goals + service failover attributes
 - ▶ DBCA
 - ▶ Allows you to set service failover type
 - ▶ srvctl
 - ▶ Allows you to set preferred and available instances + service failover type
- ▶ Set thresholds for service monitoring via DBMS_SERVER_ALERT

Oracle Notification Service (ONS)

- ▶ Publish/Subscribe Messaging System
- ▶ Allows both local and remote consumption
- ▶ Used by Fast Application Notification (FAN) to publish HA Events and Load Balancing Events
- ▶ Used by FAN clients to subscribe to events
- ▶ Automatically installed and configured during the Oracle Clusterware installation
- ▶ DO NOT TURN OFF – Required by Oracle Clusterware and RAC

FAN in a Nut Shell

- ▶ RAC posts
 - ▶ Node, service, instance, database status changes
 - ▶ Load balancing advice
- ▶ FAN events specify what changed, where, when
- ▶ FAN callouts execute server-side
- ▶ Subscribers receive FAN over AQ
- ▶ Packaged clients: ODP.NET, JDBC, OCI, Net

Fast Application Notification (FAN)

- ▶ Notification
 - ▶ First step for application recovery, diagnosis, repair, routing
- ▶ In-Band Events
 - ▶ Posted and processed synchronously in your session
 - ▶ Used when actively conversing with instances, ASM...
- ▶ Out-of-Band Events
 - ▶ Posted and processed asynchronously via another process
 - ▶ Used when nodes, public or private networks are down, application is doing something else, nodes are slow, hung...

FCF: Fast Connection Failover

- ▶ Introduced in 10g, Fast Connection Failover builds on the infrastructure created by FAN events and ONS.
- ▶ FCF can be easily integrated with existing application code. Once enabled, FCF's mechanism is automatic; no application intervention is needed.

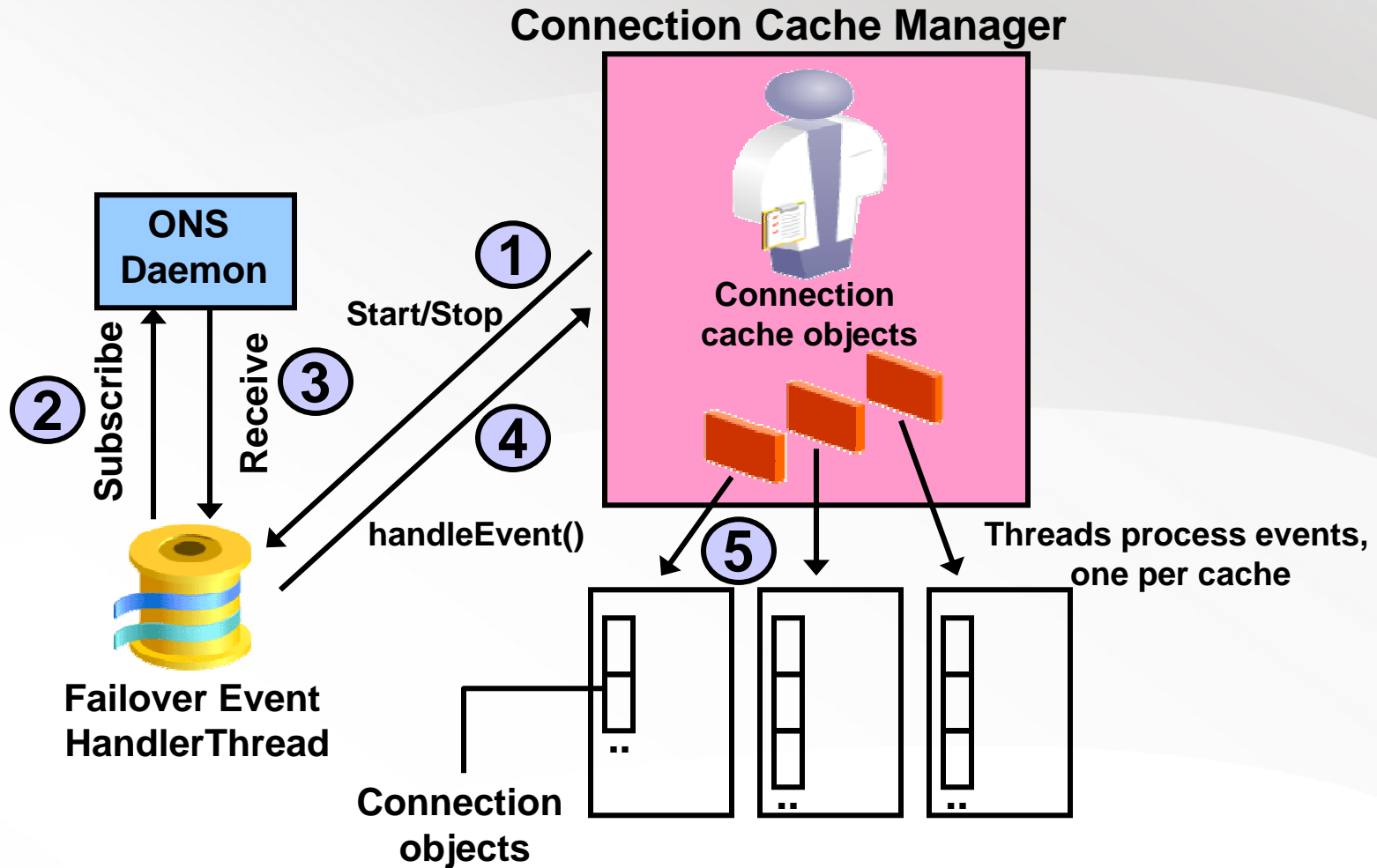
FCF Pre-Requisites

- ▶ There are a number of pre-requisites which must be met before FCF can be used. These include:
 - ▶ Implicit connection cache is enabled
 - ▶ Service names must be used for connecting to the database
 - ▶ If using the 10.1 release, the JVM in which the JDBC instance is running must have the `oracle.ons.oraclehome` property set to point to an ORACLE_HOME
 - ▶ For 10.1: Oracle Notification Service (ONS) is configured and available on the node where JDBC running
 - ▶ For 10.2 and higher, it is possible to subscribe to a remote ONS service

Fast Connection Failover How Does It Work?

- ▶ Applications continue to work while the RAC/HA events are processed asynchronously
- ▶ The pool uses an HA signature to selectively remove affected connections
- ▶ Process RAC/HA DOWN Events
- ▶ Process RAC/HA UP Events

Fast Connection Failover How Does It Work?



Fast Connection Failover Client Side Configuration

- ▶ Simplified Configuration (10.2 - Standalone)
*ods.setONSConfiguration("nodes=node1:4200,node2:4200
walletfile=/mydir/conf/wallet");*
- ▶ Ensure ons.jar file is in the CLASSPATH
- ▶ When starting Application, Specify system property
-Doracle.ons.oraclehome=<location-of-ORACLE_HOME>

Fast Connection Failover JDBC Usage

```
try {  
    conn = getConnection();  
    // do some work  
} catch (SQLException e) {  
    handleSQLException(e);  
}  
...  
handleSQLException (SQLException e)  
{  
    if (OracleConnectionCacheManager.isFatalConnectionError(e))  
        ConnRetry = true; // Fatal Connection error detected,  
}
```

Oracle Application Server handles connection retry transparently for container managed applications.

Fast Connection Failover ODP.NET Usage

- ▶ Down events monitored
 - ▶ Monitors service, service member, and node
- ▶ Cleans up sessions and connections in connection pool to downed instance
 - ▶ Prevents other incoming requests from getting bad connections
- ▶ No manual intervention required
 - `"user id=scott;password=tiger;data source=orcl;HA events=true;pooling=true"`

Fast Connection Failover **DOWN Event Processing**

- ▶ Under FCF, each connection in the cache maintains a mapping to a service, instance, database, and hostname.
- ▶ One worker thread per pool instance handles DOWN event processing – more efficient
 - ▶ First pass – Connections are marked as DOWN first, to efficiently disable bad connections
 - ▶ Second pass – Abort and remove connections that are marked as DOWN
- ▶ Active connections that may be in the middle of a transaction receive a `SQLException` instantly
- ▶ Application may replay transaction

Fast Connection Failover DOWN Event Processing

- ▶ A typical failover scenario might work like this:
 - ▶ A database instance fails, leaving several stale connections in the cache.
 - ▶ The RAC mechanism in the database generates a RAC event which is sent to the virtual machine containing JDBC.
 - ▶ The daemon thread inside the virtual machine finds all the connections affected by the RAC event, notifies them of the closed connection via SQL exceptions, and rolls back any open transactions.
 - ▶ Each individual connection receives a SQL exception and must retry (or not 😊).

Fast Connection Failover UP Event Processing

- ▶ Service UP event initiates connections to be load balanced to all active RAC instances
- ▶ All stateless connections in the pool are load balanced
 - ▶ Connection creation depends on listener's placement of connections
- ▶ UP event load balancing is dampened
 - ▶ Improved scalability

Fast Connection Failover Best Practices

- ▶ Applications must catch SQLException and retry connection requests
- ▶ Fast Connection Failover relies on the Oracle Net Services for connection creation and placement
 - ▶ In 10.1, listener's config should contain
`PREFER_LEAST_LOADED_NODE_[<listener_name>]=OFF`
 - ▶ In 10.2 and above, Service CLB_GOAL must be set
- ▶ Use Oracle RAC VIP in connect strings for all client connections

FCF vs. TAF

▶ FCF

- ▶ Proactive
- ▶ Uses ONS to receive information about cluster status
- ▶ Cleans up connections immediately, before they are used
- ▶ Application-tier (connection pool) technology

▶ TAF

- ▶ Reactive
- ▶ Network layer technology (OCI/Net)
- ▶ Connections discovered disconnected when they are next used
- ▶ Less efficient
- ▶ Useful for non-pooling applications

TAF: Transparent Application Failover

- ▶ TAF allows applications and processes to automatically reconnect to another instance in event of failure.
- ▶ Failover is transparent and the new connection will be established. However, any uncommitted transactions will be rolled back.
- ▶ TAF occurs at the OCI layer which means an application does not need to be modified in order to take advantage of TAF.

TAF Configuration

▶ Here is an example tnsnames.ora file.

```
ORCLSERV =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = nap-rac01-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = nap-rac02-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orclserv.itconvergence.com)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
        (RETRIES = 180)
        (DELAY = 5)
      )
    )
  )
)
```

Failover with TAF

TAF can be configured to perform two *methods* of failover:

▶ **METHOD=BASIC**

- ▶ The session to a surviving node is created at failover time
- ▶ Delay of creating new process on surviving node is incurred at failure time
- ▶ In 10.2, can be configured entirely server-side, per service.

▶ **METHOD=PRECONNECT**

- ▶ During original connection, a backup connection is established to another cluster instance.
- ▶ Requires client-side configuration
- ▶ Faster failover time, shorter delay, more overhead to manage

Failover with TAF

TAF can be configured to perform two *types* of failover:

▶ TYPE=SESSION

- ▶ Any uncommitted transactions will be rolled back
- ▶ The session will be connected to another instance

▶ TYPE=SELECT

- ▶ Any uncommitted transactions will be rolled back
- ▶ The session will be connected to another instance
- ▶ A SELECT statement that was executing at the time of failure will be re-executed by the new session using the same SCN and fetched rows will be discarded up to the point that the original query failed.

What To Expect After TAF Failover

▶ TYPE=SESSION

- ▶ If any uncommitted transaction existed in the session
 - ▶ ORA-25402: transaction must roll back
 - ▶ rollback;
 - ▶ re-issue statement (requires application to know how/what to resubmit)
- ▶ If no active transaction
 - ▶ SELECT statements will get ORA-25401: can not continue fetches
 - ▶ Even though error may be returned, session will be connected to surviving instance and work may continue

▶ TYPE=SELECT

- ▶ Similar behavior to above, but SELECT statements will continue fetches without interruption if no transaction in progress

What TAF Cannot Do

- ▶ TAF may cause issues for applications which modify the database.
- ▶ Applications must be capable of detecting DML failure and, if necessary, reapplying statements up to the point of failure.
- ▶ Session state is lost. Through callbacks, it is possible to restore session variables, PL/SQL package variables, and instances of user-defined types.

Callbacks

- ▶ When there is a failure, callback functions are initiated on the client-side via OCI callbacks.
- ▶ This can be implemented in Java by implementing the OracleOCIFailover interface. This interface is invoked on failure of a node or lost connection.

Summary: What Does Oracle Provide For Client Failover?

- ▶ Failure notification: FAN
- ▶ Previous connection cleanup: FCF
- ▶ Automatic reconnection: FCF, TAF
- ▶ Possibly query replay: TAF, your application-level handling

Note that FCF and TAF may be utilized from JDBC, ODP.NET, and OCI.

Questions?

**(Please don't leave—I'm only about
1/2 way through!)
Load balancing is coming up next...**

Load Balancing Available

- ▶ **Client-side**
 - ▶ Uses TNS configuration to balance connections to various listeners in cluster
 - ▶ Challenging to maintain and update
 - ▶ Not "intelligent"—purely random distribution
- ▶ **Server-side**
 - ▶ More "powerful" by incorporating metrics into decision-making process
 - ▶ No maintenance required—"automagic"
 - ▶ Will redirect clients according to policies and configuration

Connect-time (Client-Side) Load Balancing

- ▶ Uses TNS entry attribute `LOAD_BALANCE=ON`
- ▶ Client chooses randomly from the `ADDRESSES` in the `ADDRESS_LIST`
- ▶ Client may then be redirected based on server-side load balancing
- ▶ Useful to spread initial connection load among all listeners in the cluster (especially during major failover events)

Runtime Connection Load Balancing

Why Do We Need It?

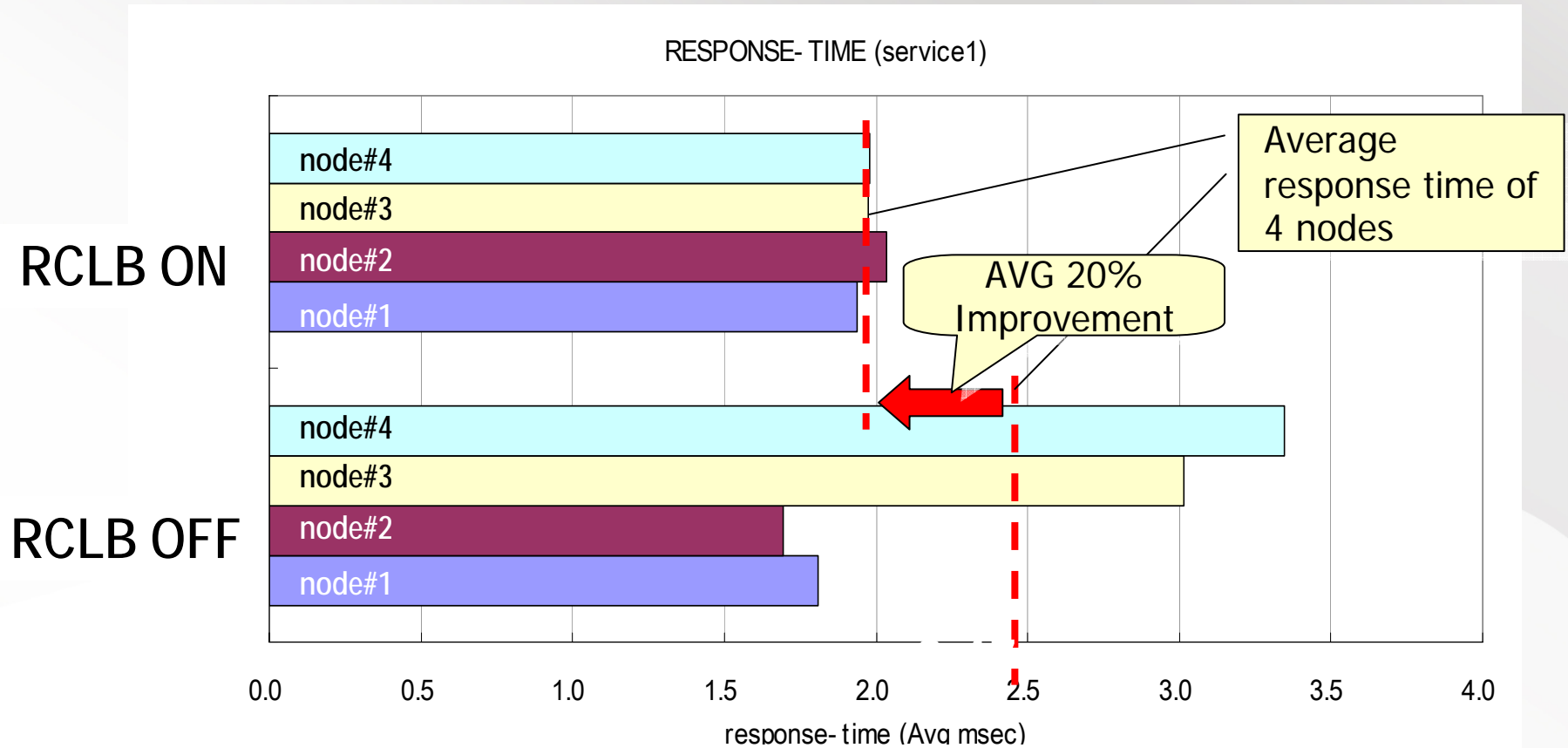
- ▶ Connections in pool last a long time. *Balancing at connect time is no longer enough.*
- ▶ Need to do best routing of work requests over existing connections to make best use of resources
- ▶ Capacity may differ across the cluster, over time as things change
- ▶ Want to avoid sending work to slow, hung, dead nodes

Runtime Connection Load Balancing

What Does It Do?

- ▶ Manages pooled connections for high performance and scalability
- ▶ Receives "continuous" recommendations on the percentage of work to route to database instances
- ▶ Adjusts distribution of work based on different backend node capacities such as CPU capacity or response time
- ▶ Reacts "quickly" (not immediately) to changes in cluster reconfiguration, application workload, overworked nodes or hangs

Runtime Connection Load Balancing Better Response Time

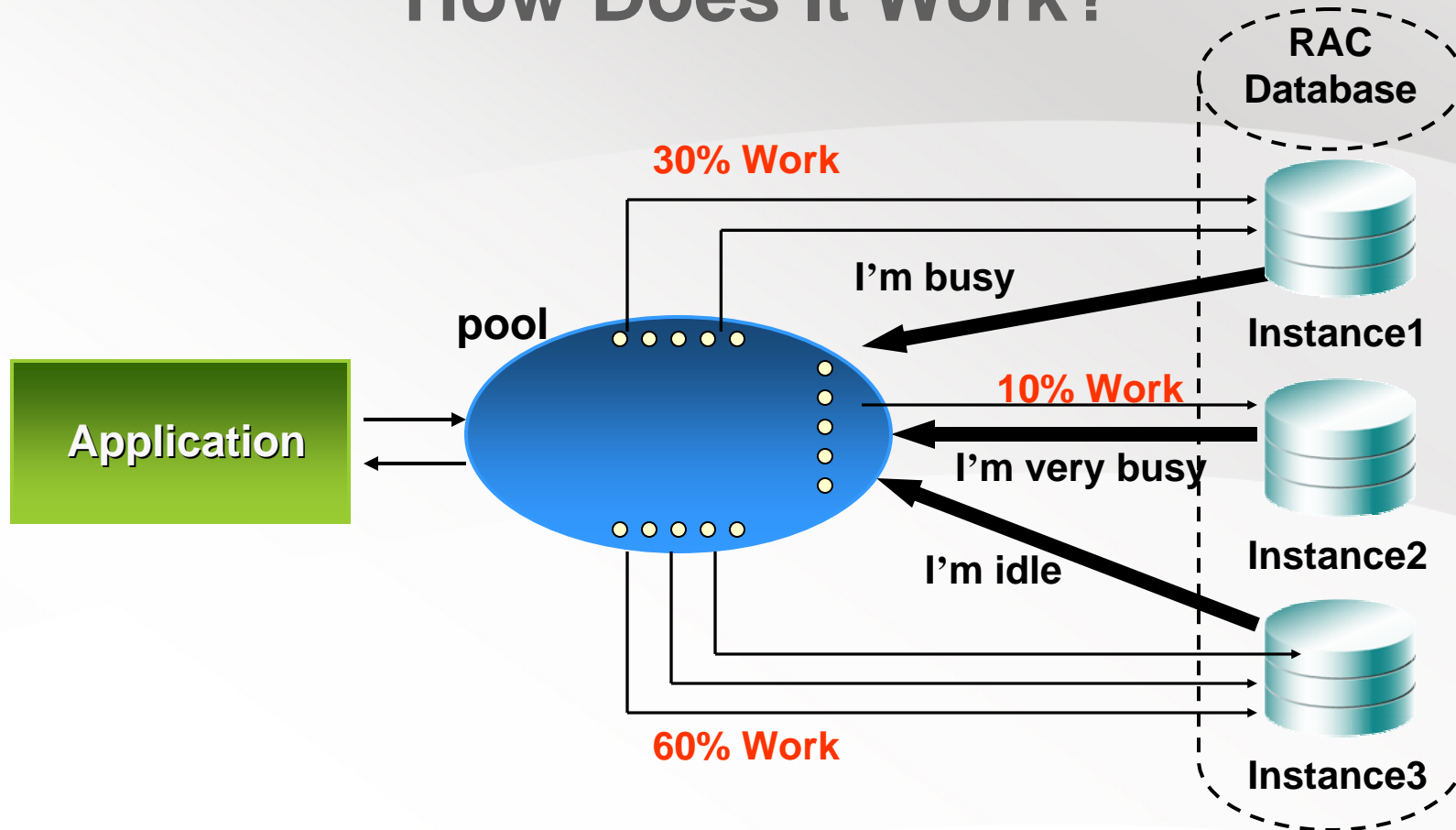


Tests run by NS Solutions Corporation, Japan

Runtime Connection Load Balancing **How Does It Work?**

- ▶ Uses connections to well performing instances more often
- ▶ Gets distribution metrics from RAC Load Balancing Advisory
- ▶ New and unused connections to "bad" instances will gravitate away over time
- ▶ When distribution metrics are not received, connection selection falls back to random choice

Runtime Connection Load Balancing How Does It Work?



Runtime Connection Load Balancing

How Do I Use It?

- ▶ Tight integration with RAC
- ▶ When not enabled, connection request distribution is based on default policy – random
- ▶ No additional client side configuration necessary – just ensure Fast Connection Failover is enabled

RAC Load Balancing Advisory

- ▶ Monitors workload activity for a service across all instances in the cluster
- ▶ Analyzes the service level for each instance based on defined metric goal
 - ▶ Metric: service time (GOAL_SERVICE_TIME)
 - ▶ Metric: throughput (GOAL_THROUGHPUT)
- ▶ Publishes FAN events recommending amount of work to be sent to each instance and data quality flag
- ▶ Default is Off. Set via EM or DBMS_SERVICE

Load Balancing Advisory Goals

- ▶ **THROUGHPUT** – Work requests are directed based on throughput
 - ▶ Best when work completes at homogenous rates
 - ▶ Measures how quickly DB tasks complete
 - ▶ Example: trading system where work requests are similar lengths
- ▶ **SERVICE_TIME** – Work requests are directed based on response time
 - ▶ Used when the work in a service completes at various rates
 - ▶ Measures CPU usage
 - ▶ Example: internet shopping system where work requests are various lengths
- ▶ **NONE** – LBAs use a formula based on CPU utilization to create advisories

RAC Load Balancing Advisory

```
dbms_service.modify_service(  
    service_name =>      'DAN',  
    goal =>              GOAL_SERVICE_TIME,  
    failover_method =>   FAILOVER_METHOD_BASIC,  
    failover_type =>     FAILOVER_TYPE_SELECT,  
    clb_goal =>          CLB_GOAL_LONG);
```

Load Balancing Advisory

- ▶ Load Balancing Advisory is an advisory for balancing work across RAC instances
- ▶ Load balancing advice
 - ▶ Is available to ALL applications that send work
 - ▶ Directs work to where services are executing well and resources are available
 - ▶ Adjusts distribution for different power nodes, different priority and shape workloads, changing demand
 - ▶ Stops sending work to slow, hung, failed nodes early

Load Balancing Advisory

- ▶ **Automatic Workload Repository**
 - ▶ Calculates goodness locally, forwards to master MMON process
 - ▶ Master MMON builds advisory for distribution of work
 - ▶ Records advice to SYS\$SERVICE_METRICS
 - ▶ Posts FAN event to AQ, PMON, ONS

JDBC Load Balancing Tips

- ▶ Must be using Oracle JDBC drivers
- ▶ By configuring FCF, your pool will also receive LBAs from ONS
- ▶ The LBAs will be processed locally by the pool to ensure that getConnection will return the most favorable connection in the pool
- ▶ Use remote ONS subscription in order to avoid running a local ONS daemon on the app tier
- ▶ Can be used with JDBC OCI or thin drivers

JDBC Load Balancing

```
OracleDataSource ods = new OracleDataSource()  
...  
ods.setUser("Scott");  
ods.setPassword("tiger");  
ods.setConnectionCachingEnabled(True);  
ods.setFastConnectionFailoverEnabled(True);  
ods.setConnectionCacheName("MyCache");  
ods.setConnectionCacheProperties(cp);  
// configure remote ONS subscription over SSL  
ods.setONSConfiguration("nodes=racnode1:6200,racnode2:6200  
walletfile=/mydir/conf/wallet");
```

.NET Integration with Grids

- ▶ **Automatic connection cleanup**
 - ▶ Via FCF
 - ▶ CP Attribute: HA Events = true

- ▶ **Automatic connection load balancing**
 - ▶ Via RLB
 - ▶ CP Attribute: Load Balancing = true

- ▶ **Must be using ODP.NET connection pool**

ODP.NET Connection Load Balancing

- ▶ ODP.NET connection pool is integrated with RAC load balancing advisory
- ▶ When application calls `OracleConnection.Open`, connection with best service given
- ▶ Policy defined by service goal

```
"user id=scott;password=tiger;data  
source=orcl;load  
balancing=true;pooling=true"
```

Miscellaneous Tips

- ▶ Test everything. Rinse. Repeat.
- ▶ See #1 above.
- ▶ Java clients should set the `oracle.net.ns.SQLnetDef.TCP_CONNTIMEOUT_STR` property on the data source
- ▶ Do not use TAF with JDBC OCI clients as TAF will interfere with FCF.

Miscellaneous Tips

- ▶ JDBC thin clients can use FCF
- ▶ Configure remote ONS subscriptions to enable FCF—also, configure ONS communication via SSL
- ▶ Configure `SQLNET.OUTBOUND_CONNECT_TIMEOUT` on clients in `sqlnet.ora` file

Next Steps

- ▶ Read "Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 10g Release 2" (January 2007) found at <http://www.oracle.com/technology/deploy/availability/>
- ▶ Read " Workload Management with Oracle Real Application Clusters" found at <http://www.oracle.com/technology/products/database/clustering/>
- ▶ Join the RAC SIG at <http://www.oracleracsig.org/>
- ▶ Test, TEST, **TEST**.

References, page 1

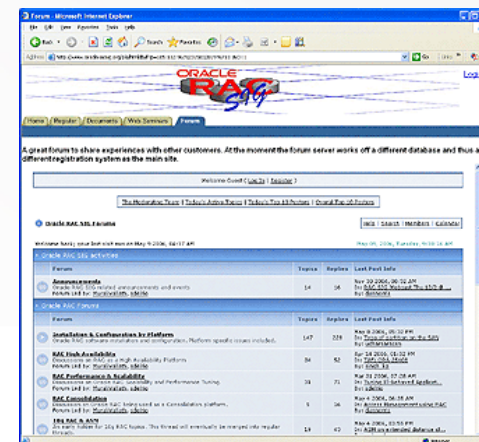
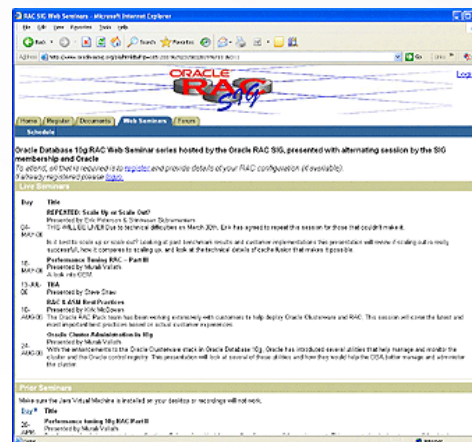
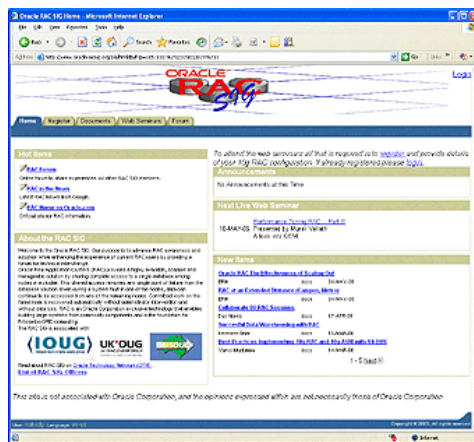
- ▶ Padraig O'Sullivan, IT Convergence contributed many slides and samples in this presentation
- ▶ Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 10g Release 2 (January 2007), <http://www.oracle.com/technology/deploy/availability/>
- ▶ Several documents by Barb Lundhild, Oracle: conference presentations, whitepapers on <http://otn.oracle.com/rac>
- ▶ Alex Keh, Oracle, presentation "Leveraging Oracle Grid Computing with .NET"

References, page 2

- ▶ <http://www.oracle.com/technology/products/database/clustering/>
- ▶ Demystifying RAC Internals, Barb Lundhild & Kirk McGowan, Oracle, OOW 2006
- ▶ Under the Covers of High Availability and Load Balancing for JDBC Connections in Real Application Clusters/Grid Environments, Rajkumar Irudayaraj, Oracle
- ▶ Workload Management with Oracle Real Application Clusters, Barb Lundhild, Oracle

RAC SIG Events

- ▶ See www.oracleracsig.org for details
 - ▶ **Monday, April 16 @ 10:30 am: RAC SIG Expert Panel**
 - ▶ **Tuesday, April 17 @ 1:45 pm: RAC SIG Birds of a Feather Mixer**
 - ▶ **Wednesday, April 18 @ 11:00 am: RAC SIG Customer Panel**
- ▶ Join the RAC SIG at www.oracleracsig.org!



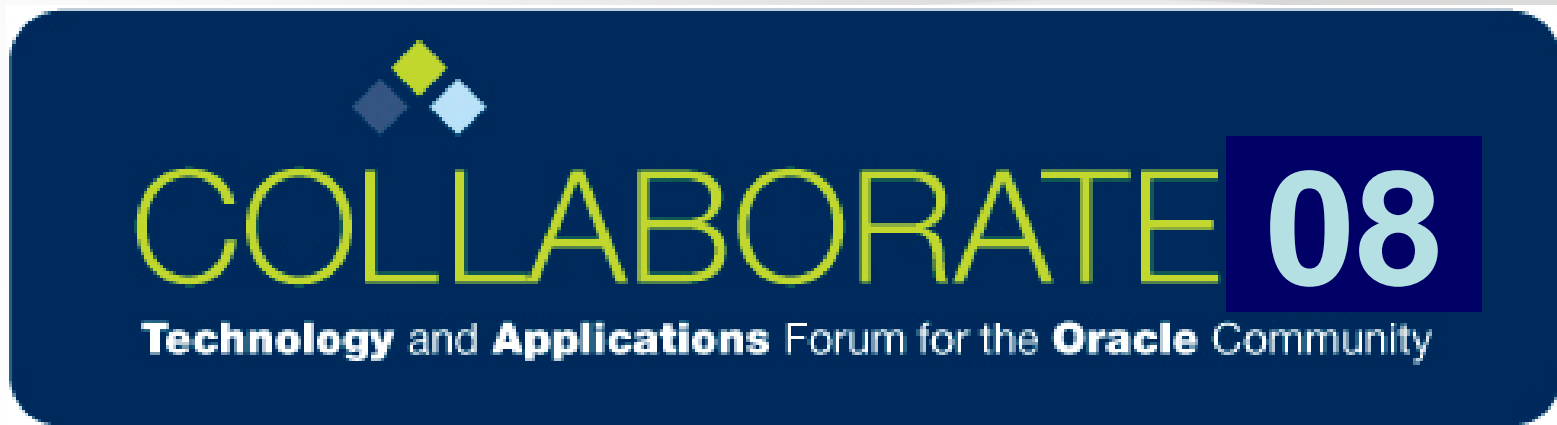
Save the Date!



Oracle OpenWorld San Francisco
November 11-15, 2007

See you there!

Save the Date!



April 13 - 17, 2008
Colorado Convention Center
Denver, CO

Submit to present for the IOUG!

- ▶ Share your expertise with the greater Oracle community. Solidify your reputation as an Oracle expert! The IOUG is looking for presentations in the following tracks: **Architecture, Database, Development, and Middleware.**
- ▶ Submit your abstracts no later than (not yet announced—usually mid-November).
- ▶ If selected, you will receive a FREE COLLABORATE 08 conference registration, industry and peer recognition and much more!

Submit today at www.ioug.org



COLLABORATE 08
Technology and Applications Forum for the Oracle Community



Thank You!



Oracle Real Application Clusters Load Balancing and Failover Options

An IT Convergence presentation by Dan Norris

Legal

The information contained herein should be deemed reliable but not guaranteed. The author has made every attempt to provide current and accurate information. If you have any comments or suggestions, please contact the author at:

dnorris@itconvergence.com

Only Collaborate 07 has been granted permission to reprint and distribute this presentation. Others may request redistribution permission from dnorris@itconvergence.com.