

ORACLE®




ORACLE
OPEN
WORLD

Your. Open. World.

Building and Deploying Web-scale Social Networking Application, Using PHP and Oracle Database

Srinath Krishnaswamy, Director, Oracle Corp.
Levi Dixon, Senior Architect, Community Connect
Nicolas Tang, VP Technical Operations, Community Connect

ORACLE®



The following 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.

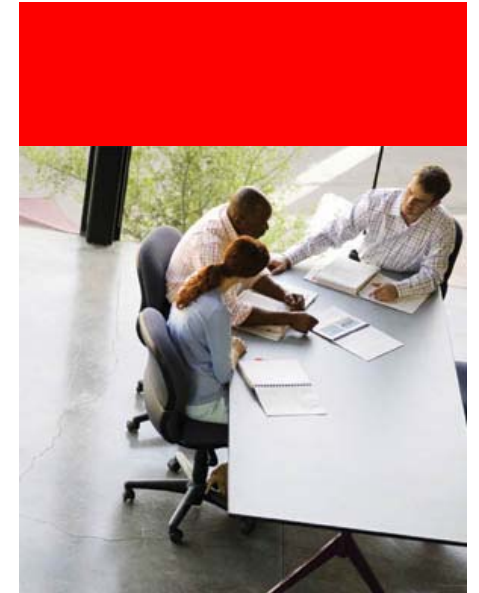


Overview

- Oracle Support for PHP
- Web-Scale PHP Application
- Oracle Database Resident Connection Pooling (DRCP)
- Horizontal Scalability and High Availability with RAC
- Community Connect – Building high-traffic Social Networking Web Application



PHP and the OCI8 Extension





What is OCI8?

- Main Oracle Database extension for PHP
- Open source and part of PHP

```
<?php
```

```
$c = oci_connect('un', 'pw', '//localhost/XE');  
$s = oci_parse($c, 'select * from employees');  
oci_execute($s);  
while ($row = oci_fetch_array($s))  
    foreach ($row as $item)  
        print $item;  
?>
```

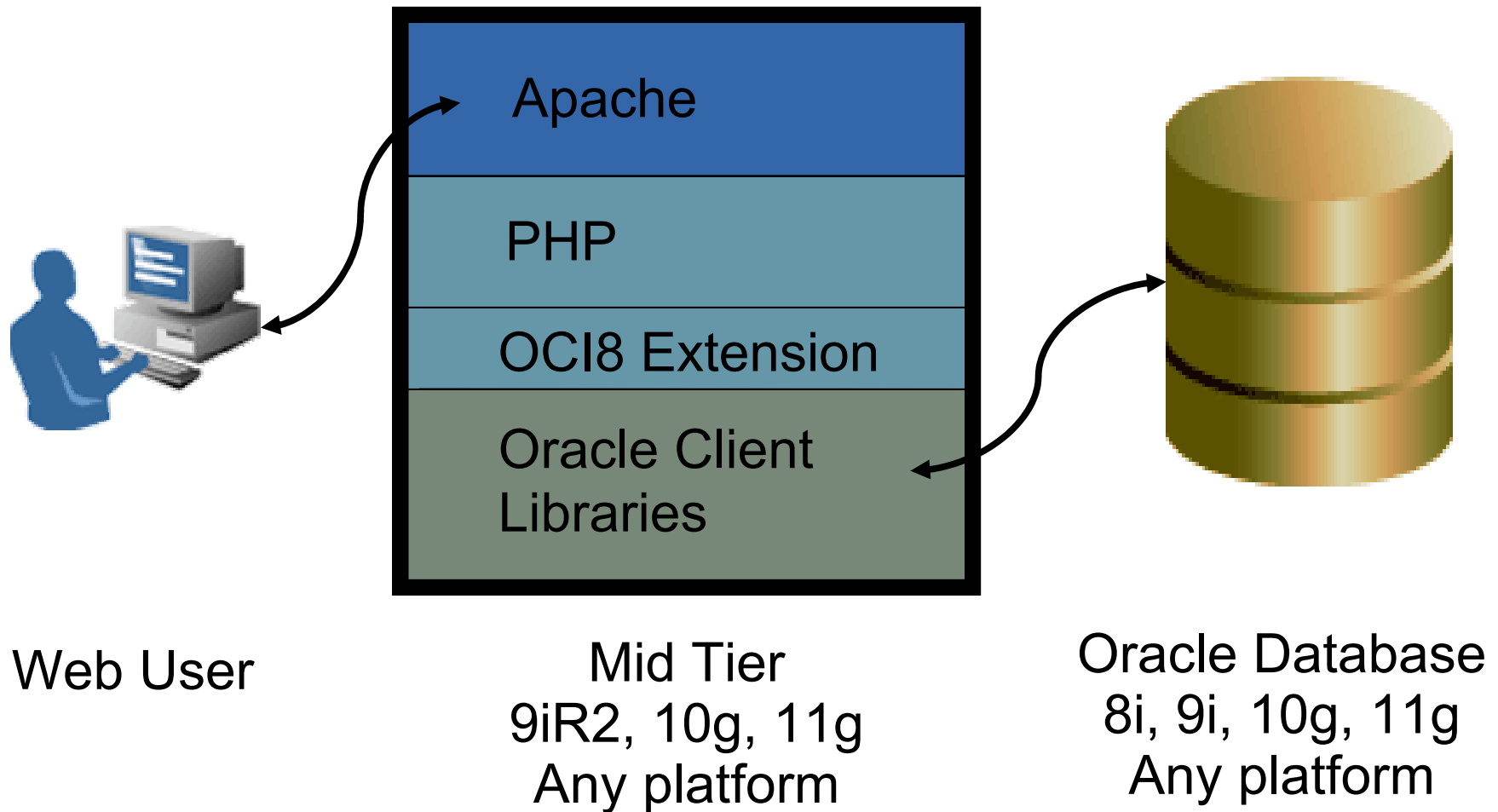


OCI8 Basic Oracle Features

- SQL and PL/SQL
- Large Objects (LOB), including Temporary LOBs
- Collections
- REFCURSORS
- Bind Data
- Privileged Connections
- Meta Data

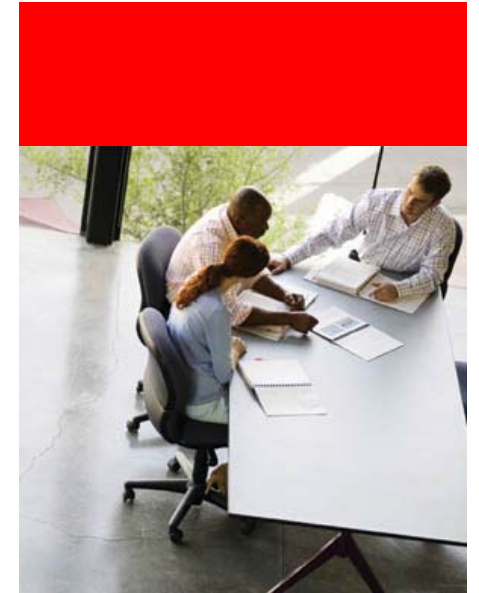


Three Tier Web Model





Web-Scale PHP Application





Scalability Requirement

- Middle tier expected to scale horizontally
 - Add more boxes
 - Add more `apache` processes per box
 - Adds more and more database connections
- Database tier expected to support tens of thousands of connections without degrading performance
- Middle tier connection pooling not an option
 - `apache` single threaded for PHP usage



Existing Options

- Use Non Persistent Connections
 - High connect times
 - Burns CPU
 - Throughput hits a wall on a commodity database box soon
- Use Persistent Connections
 - Majority idle
 - Excessive swapping, eventually exhausts RAM
- Both strategies run into problems as we get into thousands of database connections

The Solution: Database Resident Connection Pooling (DRCP)





DRCP Overview

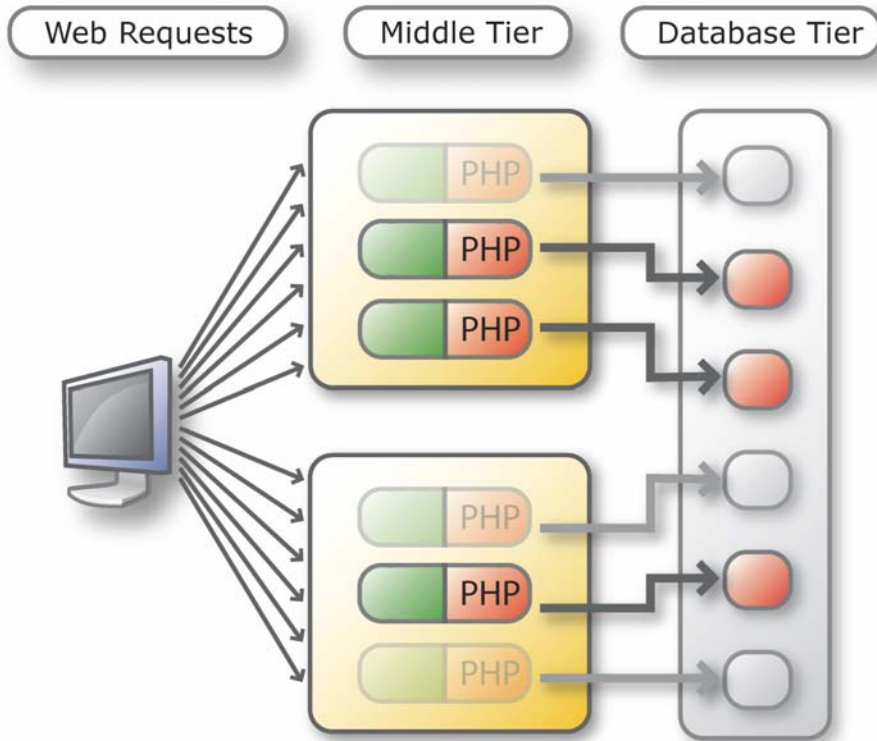
- Oracle Database 11g Feature
 - Not just for PHP
- Database tier connection pool
 - Pool of dedicated servers on database machine
 - Pool shared across Apache processes and middle-tier nodes
- Scales to tens of thousands of persistent connections
- Speeds up non-persistent connection creation too!
- Co-exists in all database server configurations
 - Single instance, RAC



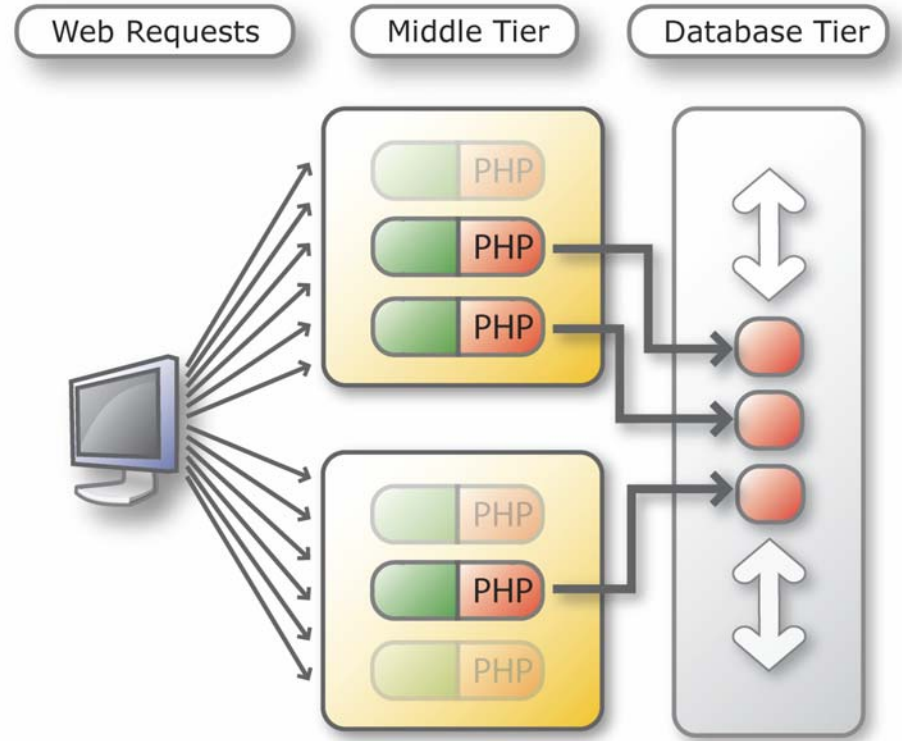
Basic Functionality

- Pooling is optionally enabled by DBA on server
 - Min, Max, Timeout etc. for Pool
- Client connect string:
 - `hostname/service:POOLED`
 - `(SERVER=POOLED)`
- Client directed to Database Resident Pool
- Pooled Server “locked” when connection requested by client
- Pooled Server “released” back to pool when client disconnects

Dedicated Servers vs DRCP



No Connection Pooling



11g Database Resident Connection Pooling

Sample Sizing for 5000 Clients

	Dedicated Servers	Shared Servers	<i>DRCP Servers</i>
Database Servers	<i>5000 * 4 MB</i>	<i>100 * 4 MB</i>	<i>100 * 4 MB</i>
Session Memory	<i>5000 * 400 KB</i>	<i>5000 * 400 KB</i>	<i>100 * 400 KB</i>
<i>DRCP Connection Broker Overhead</i>			<i>5000 * 35 KB</i>
Total Memory	21 GB	2.3 GB	610 MB



When to Use DRCP

- DRCP can be useful when:
 - Large number of connections need to be supported with minimum memory usage on database host
 - Applications mostly use same database credentials for all connections
 - Applications acquire a database connection, work on it for a relatively short duration, and then release it
 - Multiple web server hosts
 - Connections look identical in terms of session settings, for example date format settings and PL/SQL package state
- Generally true for majority of web applications



Starting and Configuring DRCP

- Start the pool:

```
SQL> execute dbms_connection_pool.start_pool();
```

- Optionally Configure the Pool:

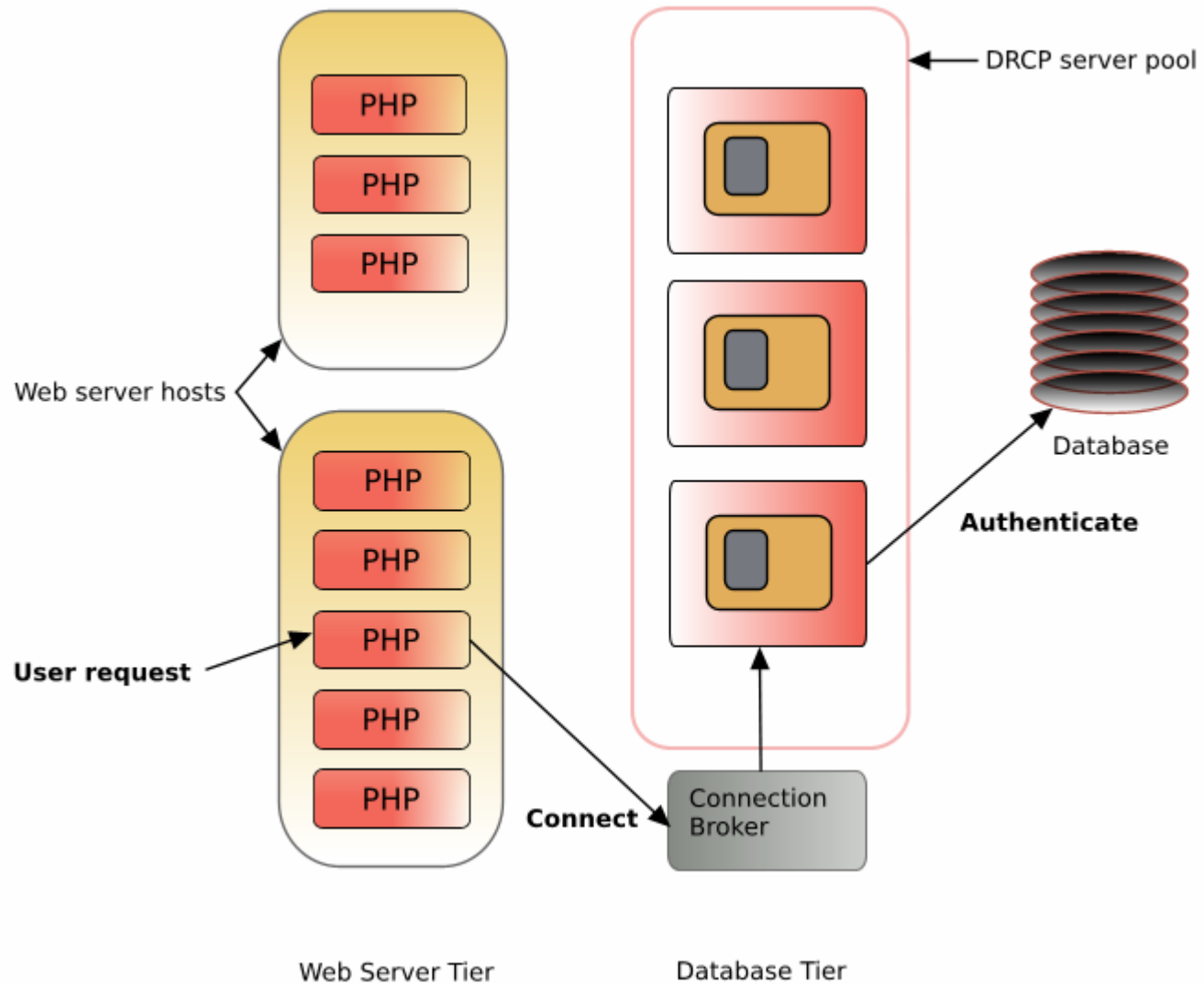
```
SQL> execute dbms_connection_pool.configure_pool(  
    pool_name           =>  
    'SYS_DEFAULT_CONNECTION_POOL',  
    minsize             => 4,  
    maxsize             => 40,  
    incrsiz             => 2,  
    session_cached_cursors => 20,  
    inactivity_timeout  => 300,  
    max_think_time      => 600,  
    max_use_session     => 500000,  
    max_lifetime_session => 86400);
```



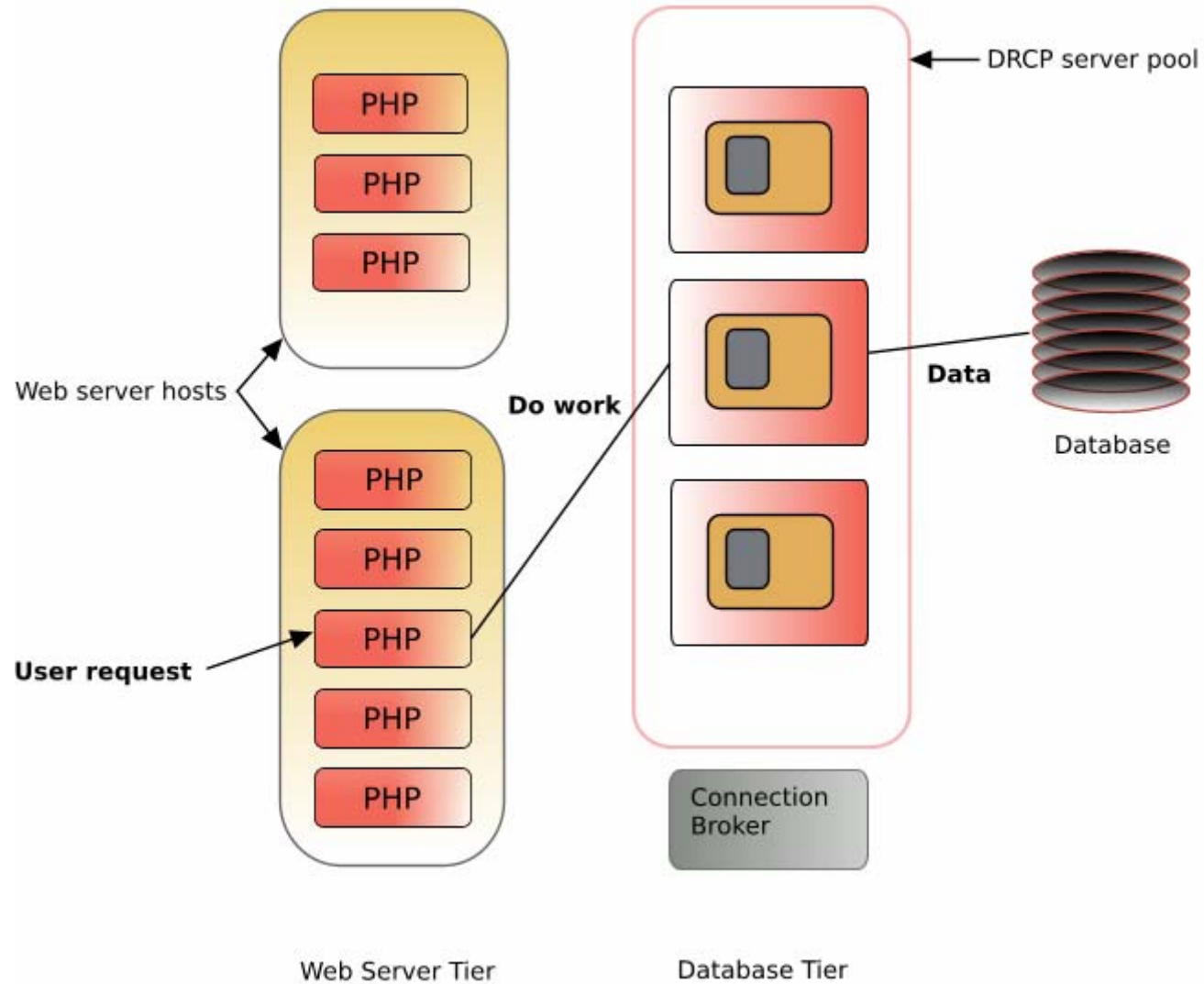
DRCP: System Components

- **Connection Broker**
 - New in Oracle Database 11g
 - Oracle instance background daemon
 - Handles initial authentication
 - Handles subsequent connect/disconnect requests
- **Pooled Servers**
 - New in Oracle Database 11g
 - Oracle instance background slave processes
- **Oracle 11g OCI Client library**
 - DRCP aware

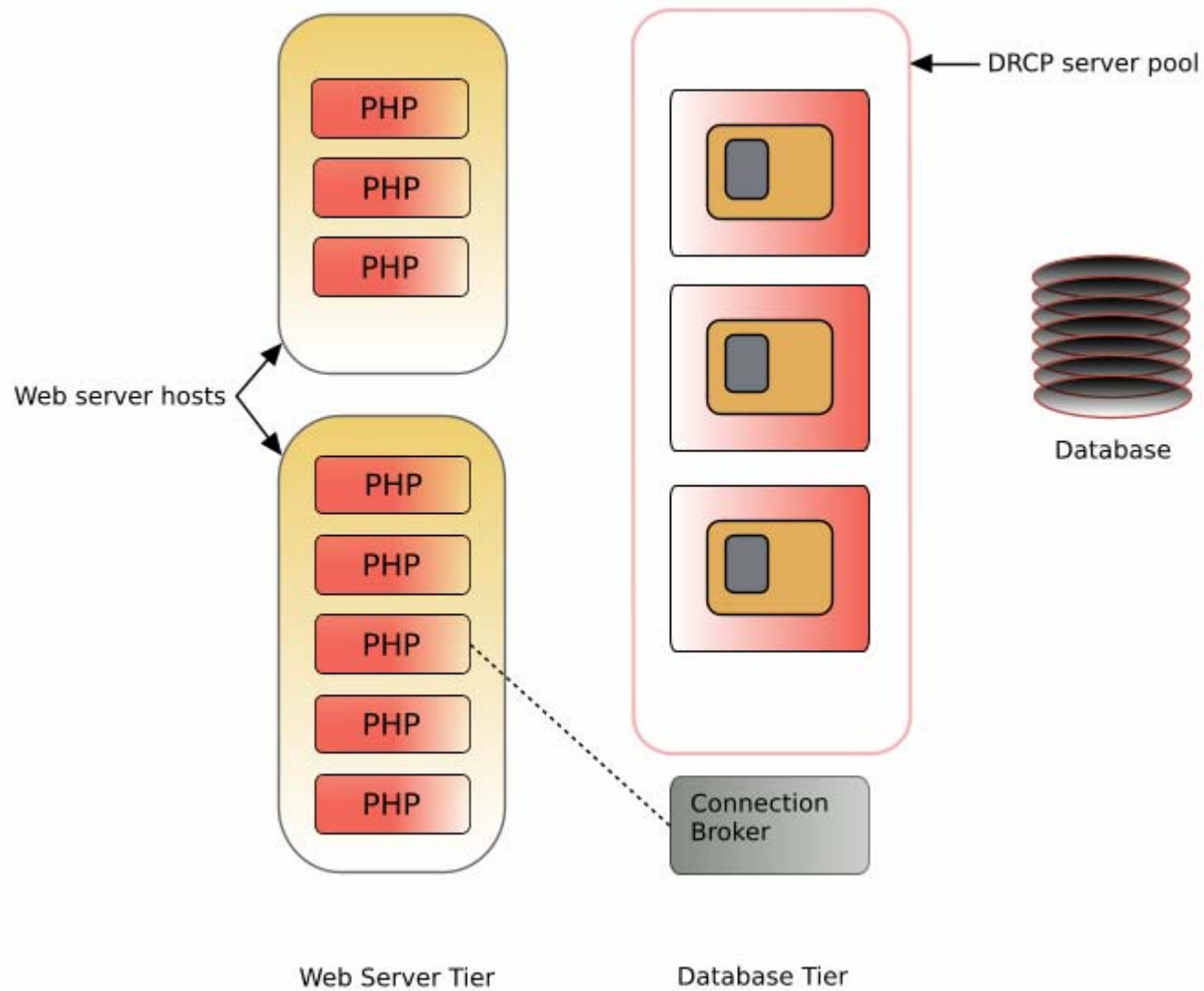
DRCP: Connecting



DRCP: Doing Work



DRCP: After Disconnecting

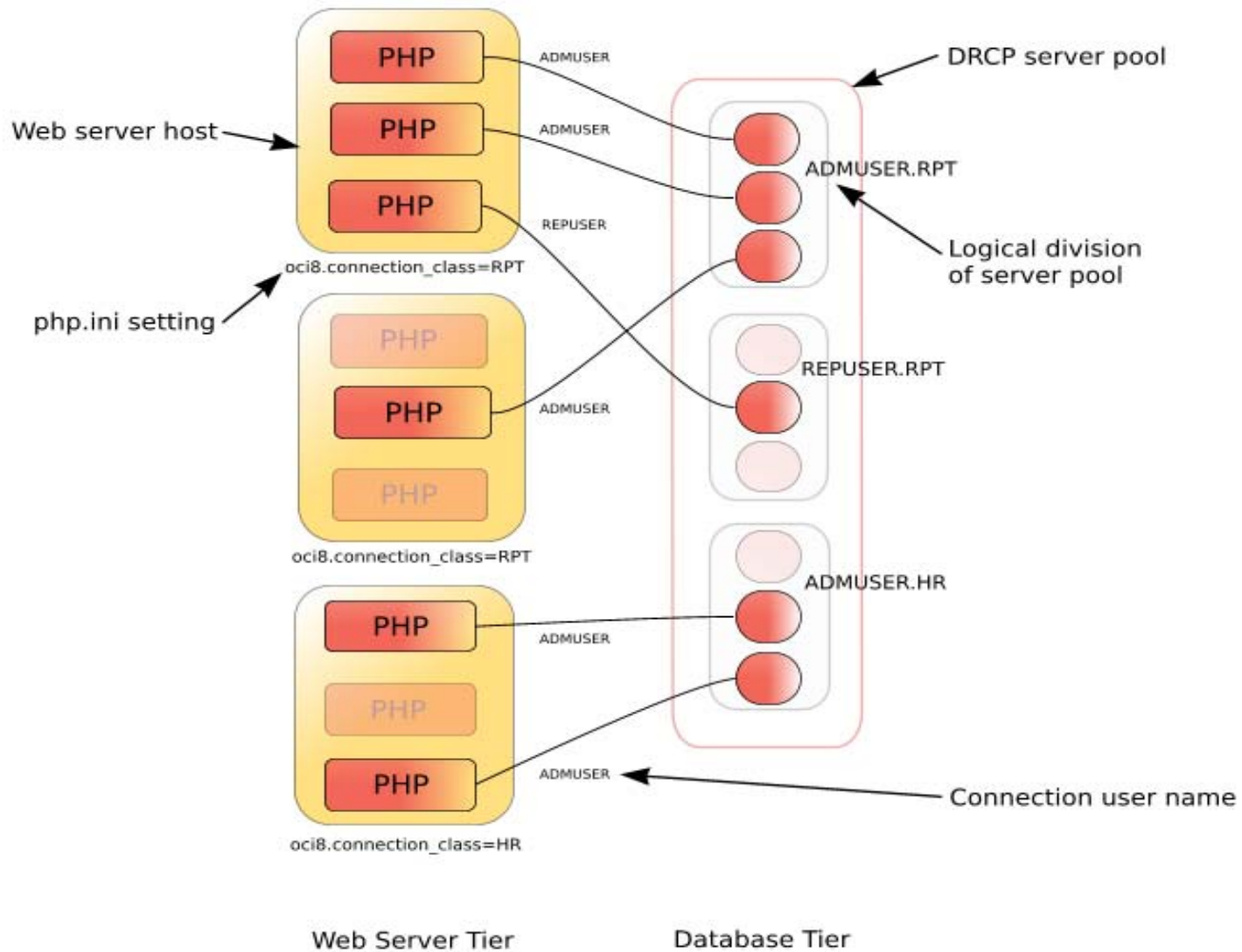




DRCP: Pool Sharing

- There is one physical pool in Oracle 11g
- Connections are never shared across different database usernames for security
- CONNECTION_CLASS
 - Allows for multiple logical sub pools per user
 - Allows clients to identify the “type” of connection required
 - Only clients with same “username.connection_class” share connections

DRCP: Pool Sharing

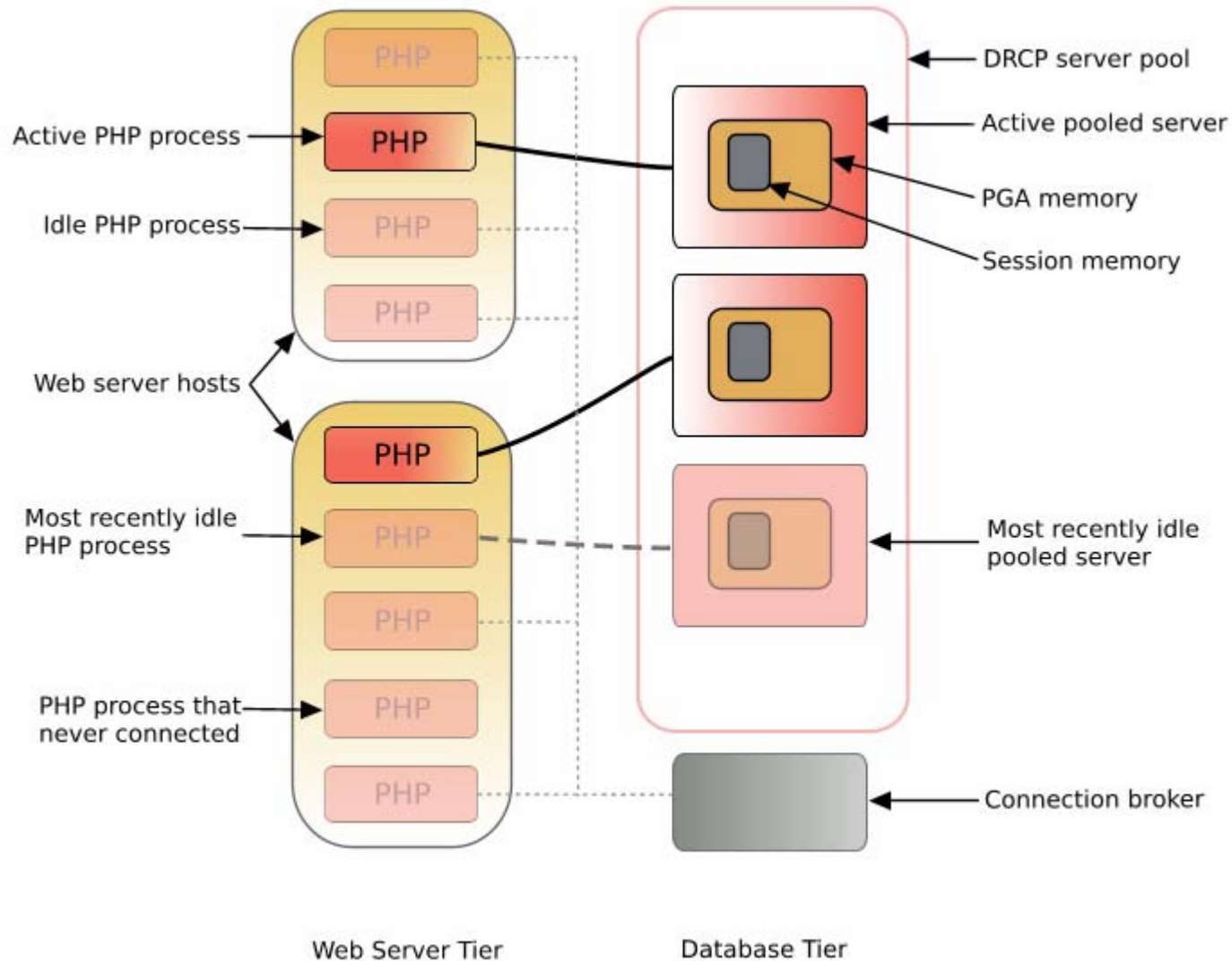




Configuring DRCP for a Very Large #Connections

- Bump up O/S file descriptor limits
 - E.g on Linux: `/etc/security/limits.conf`
 - `oracle HARD NOFILE 40000`
- Configuring additional brokers
 - Helps if O/S has a small per process file descriptor limit
 - `dbms_connection_pool.alter_param()`
 - `NUM_CBROK (default=1)`
- Configure the max number of connections per broker
 - Helps distribute connections across multiple brokers
 - `dbms_connection_pool.alter_param()`
 - `MAXCONN_CBROK (default=40,000)`

DRCP: Key Optimizations





Installation Steps for DRCP with PHP

- Install Oracle Database 11.1.0.6 (with patches for bug 6474441, 7240473 (for SE support))
- Download
 - PHP 5.3 (has OCI8 1.3)
OR
 - PHP 4.3.9 – 5.2 and OCI8 1.3.4 from Pecl
- Build PHP using Oracle 11g client libraries from ORACLE_HOME or Instant Client



Using DRCP with PHP

- No application code change required
 - Unchanged PHP API
 - Deployment decision to use DRCP
 - Application can still talk to other Oracle versions
- Configure and start DRCP on Oracle Database 11g
 - Setup O/S file descriptor limits
 - Configure pool limits, timeouts, #brokers etc as required

dbms_connection_pool.start_pool
- Set php.ini parameters
 - oci8.connection_class = MYAPP**
 - oci8.old_oci_close_semantics = Off** (which is default)



Some DRCP Best Practices for PHP

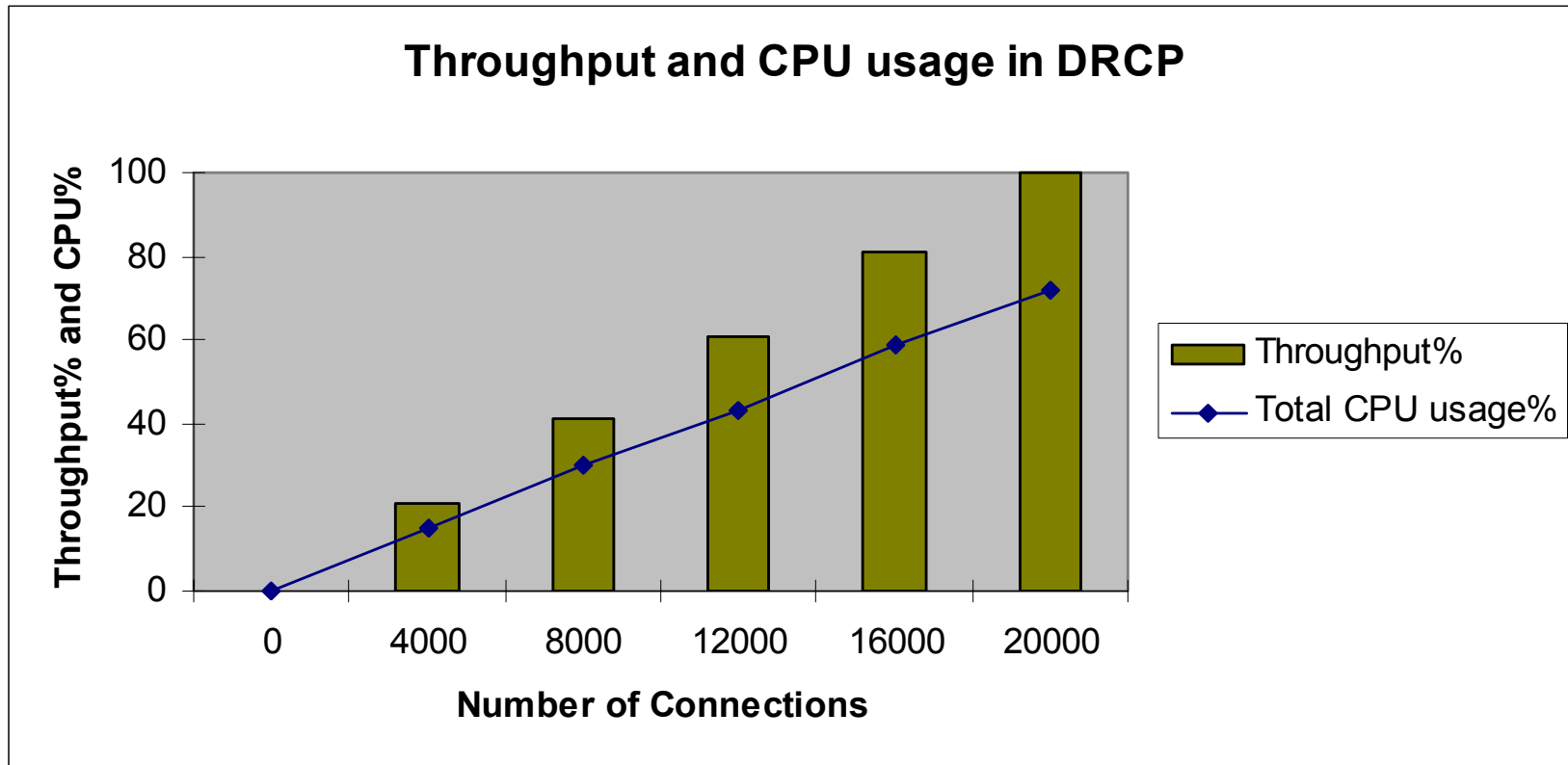
- Close connections when doing non-DB processing
 - Allows DRCP to serve more clients with the same pool size
- Setting `oci8.connection_class`
 - Ensure that applications under a `connection_class` have uniform session state expectations
 - Use the same `oci8.connection_class` value across machines hosting the same app to maximize DRCP sharing
- Monitor DRCP performance with `V$CPOOL_STATS`



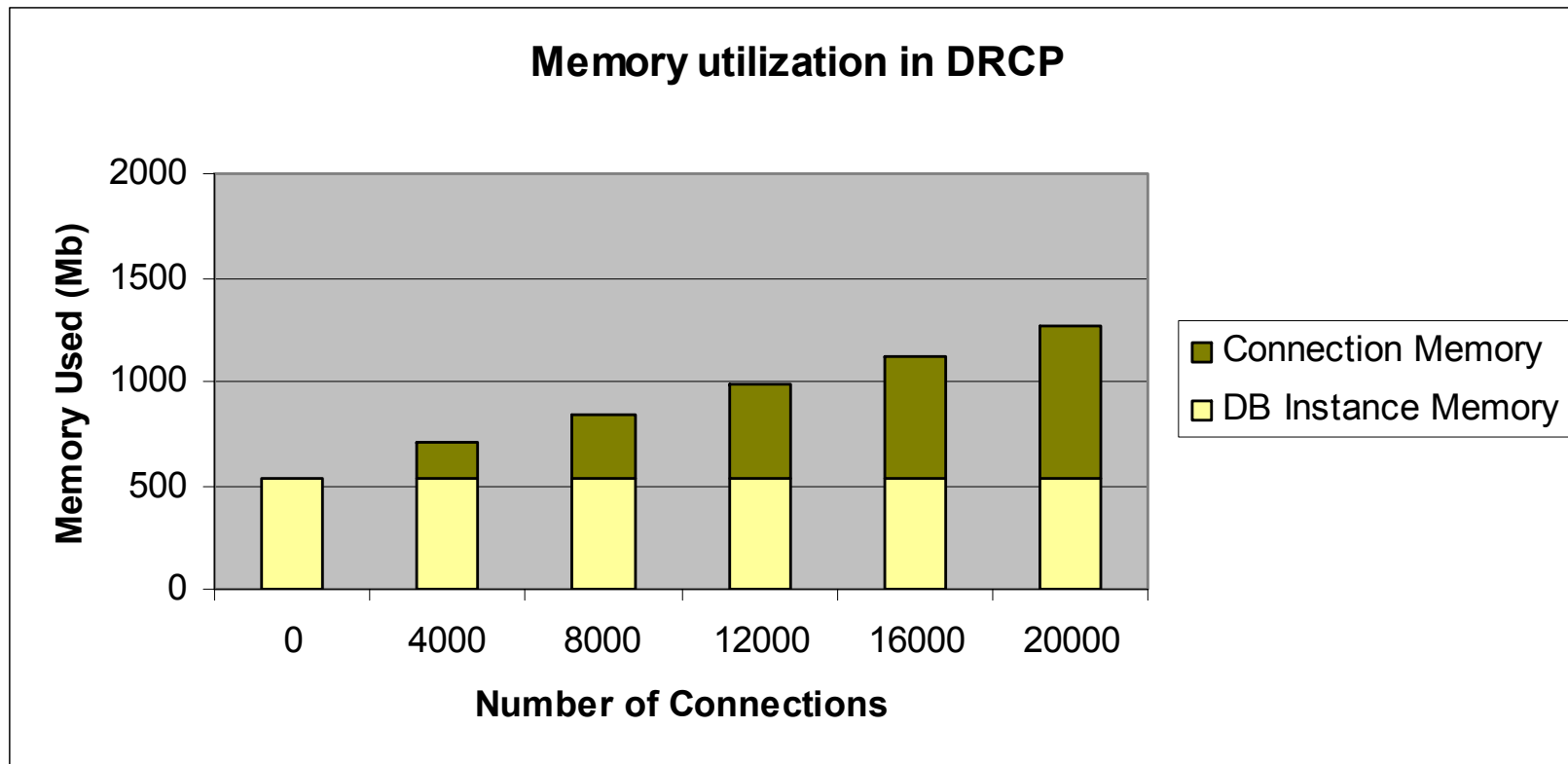
A DRCP Benchmark

- PHP script
 - connect, query, disconnect, sleep 1 second
- Server
 - Dual CPU Intel P4/Xeon 3.00GHz
 - 2GB RAM
 - 32bit Red Hat Enterprise Linux 4
- DRCP
 - 100 pooled servers, one connection broker
- Clients
 - 3 similar machines
 - Apache
- See PHP DRCP Whitepaper for details

Database Resident Connection Pool Performance



Database Resident Connection Pool Memory Utilization



Horizontal Scalability and High Availability with Real Application Clusters (RAC)





DRCP on RAC

- Horizontal scaling as your database load increases
- DRCP starts on all RAC instances
- Same pool limits apply to each individual RAC instance
 - min, max
 - number of brokers
 - max connections per broker
- DRCP connections benefit from TNS Listener connection load balancing across RAC instances



High Availability

- Fast Application Notification (FAN)
- When DB node or network fails
 - Database generates FAN events
 - **Oracle error returned without TCP timeout delay**
 - PHP application is not blocked for TCP timeout – it can immediately reconnect to surviving DB instance

```
$conn = doConnect();  
$err = doSomeWork($conn);  
if (isConnectionError($err)) {  
    // reconnect, find what was committed, and retry  
    $conn = doConnect();  
    $err = checkApplicationStateAndContinueWork($conn);  
}  
if ($err)  
    handleError($err);
```



Fast Application Notification

- High Availability feature for PHP with RAC or Data Guard with physical standby
- Usable with or without DRCP
- Available from Oracle 10gR2
- OCI8 1.3 supports FAN



FAN Configuration

- Tell DB to broadcast FAN Events

SQL> execute

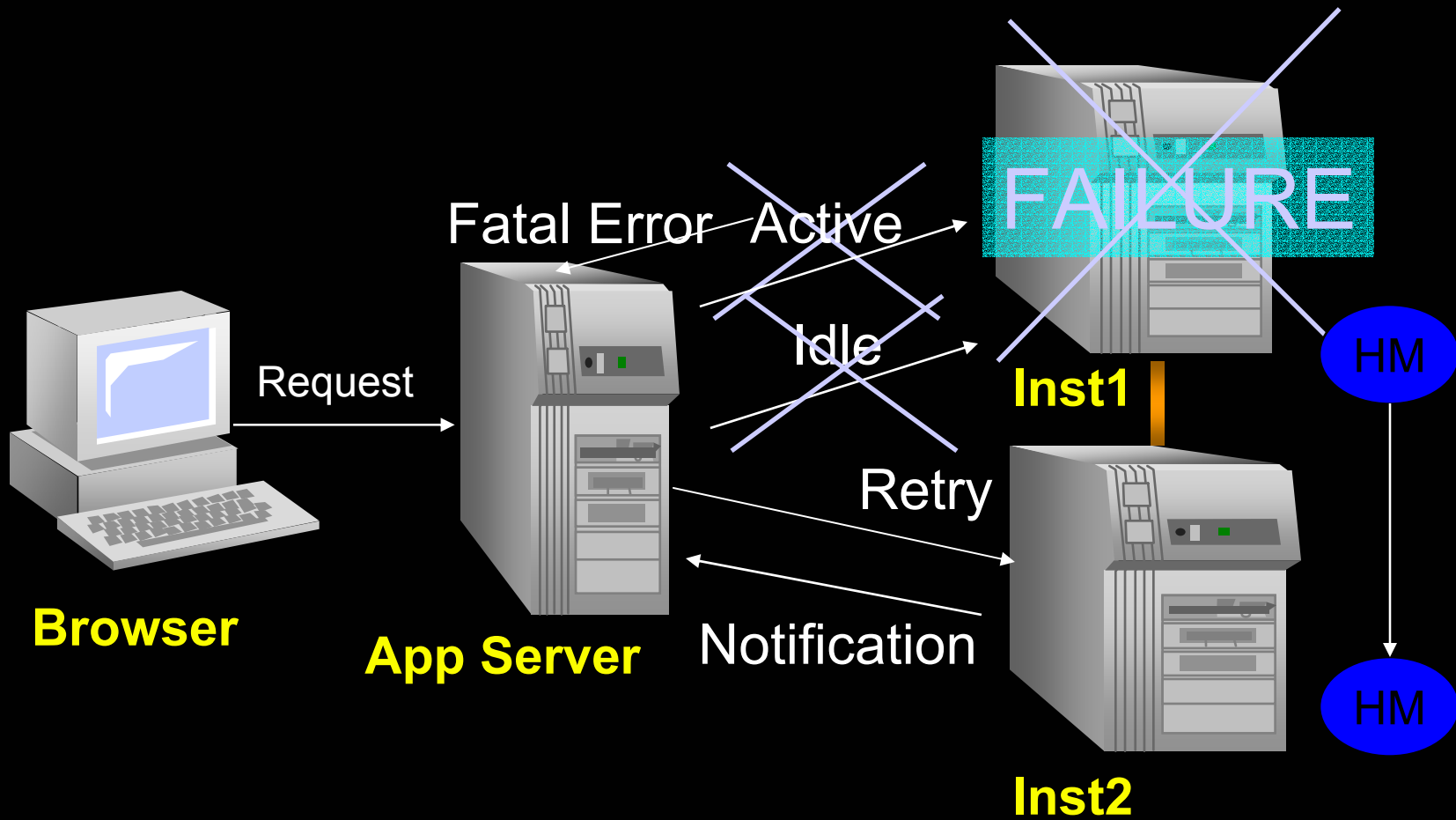
```
dbms_service.modify_service(service_name =>'SALES',  
aq_ha_notifications =>TRUE);
```

- Configure PHP's php.ini so OCI8 listens for FAN events

oci8.events = On

- Optionally add re-connection code to PHP application

Solution Overview – Stages





Other OCI8 1.3 Changes

- Fixed PHP OCI8 connection ref-counting edge cases
- Better detection of dead sessions and bounced DBs
- Automatic cleanup of dead, idle connections
- Closing persistent connections can now rollback



Get the Latest OCI8 1.3

- php.net
 - Source code, Windows binaries
- PECL - PHP Extension Community Library
 - Useful for updating PHP 4 with new OCI8

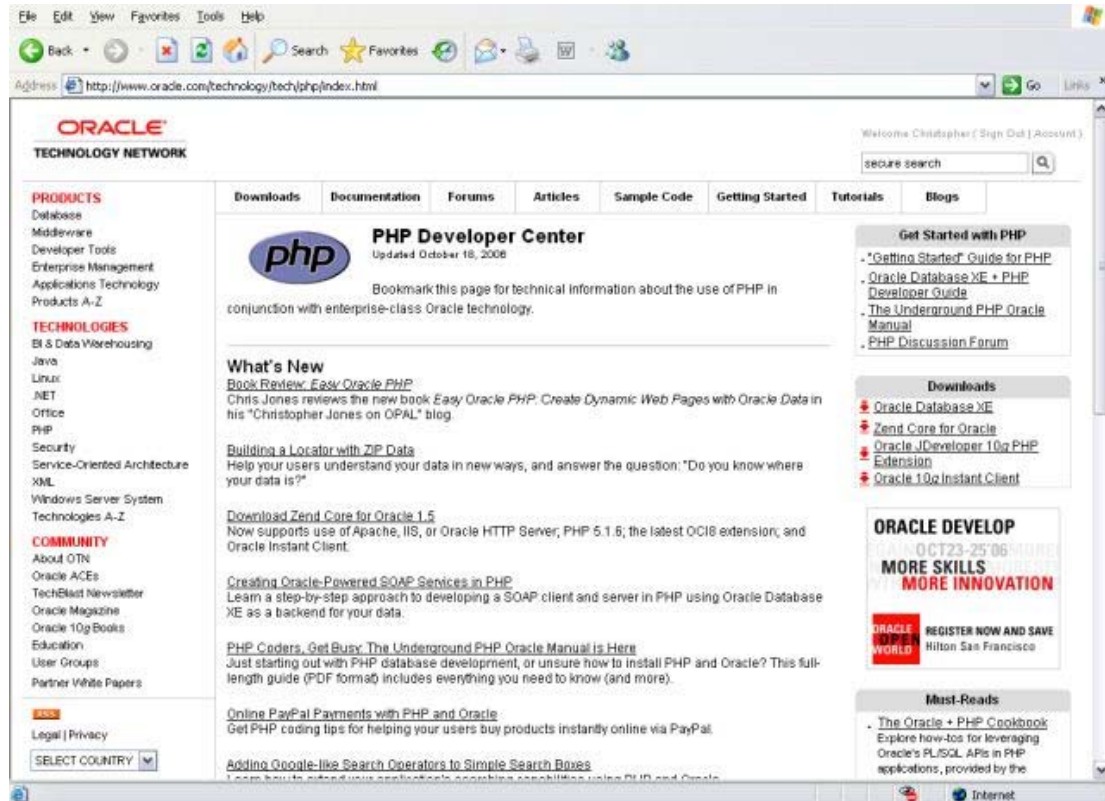


Oracle Resources

- Free Oracle Technology Network (OTN) PHP Developer Center
 - otn.oracle.com/php
 - Underground PHP and Oracle Manual
 - Whitepapers, Articles, FAQs, links to blogs, Jdeveloper PHP Extension, PHP RPMs
- Information
 - christopher.jones@oracle.com, kuassi.mensah@oracle.com
 - blogs.oracle.com/opal
- Excellent introduction to PHP with DRCP and FAN
 - www.oracle.com/technology/tech/php/pdf/php-scalability-ha-twp.pdf

Oracle Technology Network PHP Developer Center


- Free
- Articles
- Install guides
- Underground PHP and Oracle Manual
- Online forum
- PHP RPMs
- Oracle JDeveloper 10g PHP extension



The screenshot shows a web browser window displaying the Oracle Technology Network PHP Developer Center. The page features a navigation menu with categories like Downloads, Documentation, Forums, Articles, Sample Code, Getting Started, Tutorials, and Blogs. The main content area is titled "PHP Developer Center" and includes a "What's New" section with articles such as "Book Review: Easy Oracle PHP" and "Building a Locator with ZIP Data". There are also sections for "Get Started with PHP" and "Downloads". The page is dated "Updated October 18, 2008".

otn.oracle.com/php

ORACLE



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 remain at the sole discretion of Oracle.

ORACLE®