Web Scale PHP Connection Broker

Kuassi Mensah -- http://db360.blogspot.com

Server Technologies - Database Access and Programming Interfaces

Agenda

✓ Motivations
- Traditional Oracle/PHP Connection
- Web Scale Connection Broker
- High Availability for PHP with RAC

Motivations

Web Scale Deployment Requirements

- Scaling from Zero to Hero
  - Optimize memory and CPU on low end commodity hardware
  - Sustain very high number of connection requests
  - Sustain unpredictable workload
  - Optimize database server resources
- High Availability
  - 24 x 7
  - Resilience to database instance failure

Agenda

- Motivation

✓Traditional Oracle/PHP Connection

  - Non Persistent Connections
  - Persistent Connections

- Web Scale Connection Broker
- High Availability for PHP with RAC

Traditional Oracle/PHP

Connection Management

Non Persistent Connections

Standard Connection

```
$c = oci_connect($username, $password, $dbname);
```

- Connection exists through the life of the script
- Second oci_connect() in script returns same DB connection

Multiple Unique Connections

```
$c = oci_new_connect($username, $password, $dbname);
```

- Connection exists through the life of the script
- Each oci_new_connect() returns a new DB connection
- Use for independent operations

What is Wrong with Non-Persistent Connections

- High CPU Overhead
  - Connection Establishment Time
  - Frequent and Excessive Connect/Disconnect
- Cannot Scale in Web Environments
  - Max Out Database Server CPU
    - 1 database server process per PHP user
  - Cannot handle the *Digg effect*

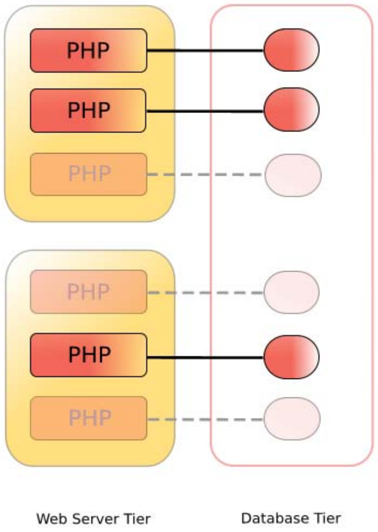
Persistent Connections

```
$c = oci_pconnect($username, $password, $dbname);
```

- Connection not automatically closed at script completion
- Fast for subsequent connections
  - But holds resources

- Configurable in *php.ini*

```
oci8.max_persistent
oci8.persistent_timeout
oci8.ping_interval
```



What is Wrong with Persistent Connections

- High Memory Overhead
  - Many Idle Cached Connections
- Limited Scalability in Web Environment
  - Max Out Database Server Memory
    - Idle Apache processes retain connections that are not pooled
  - Cannot handle the “*Digg effect*”

Agenda

- Requirement - Motivations
- Traditional Oracle/PHP Connection

✓Web Scale Connection Broker

  - Database Resident Connection Pool (DRCP)
  - DRCP: How It Works
  - DRCP: Applications Usage
  - Configuration and Sizing
  - Scalability Benchmark

- High Availability for PHP with RAC

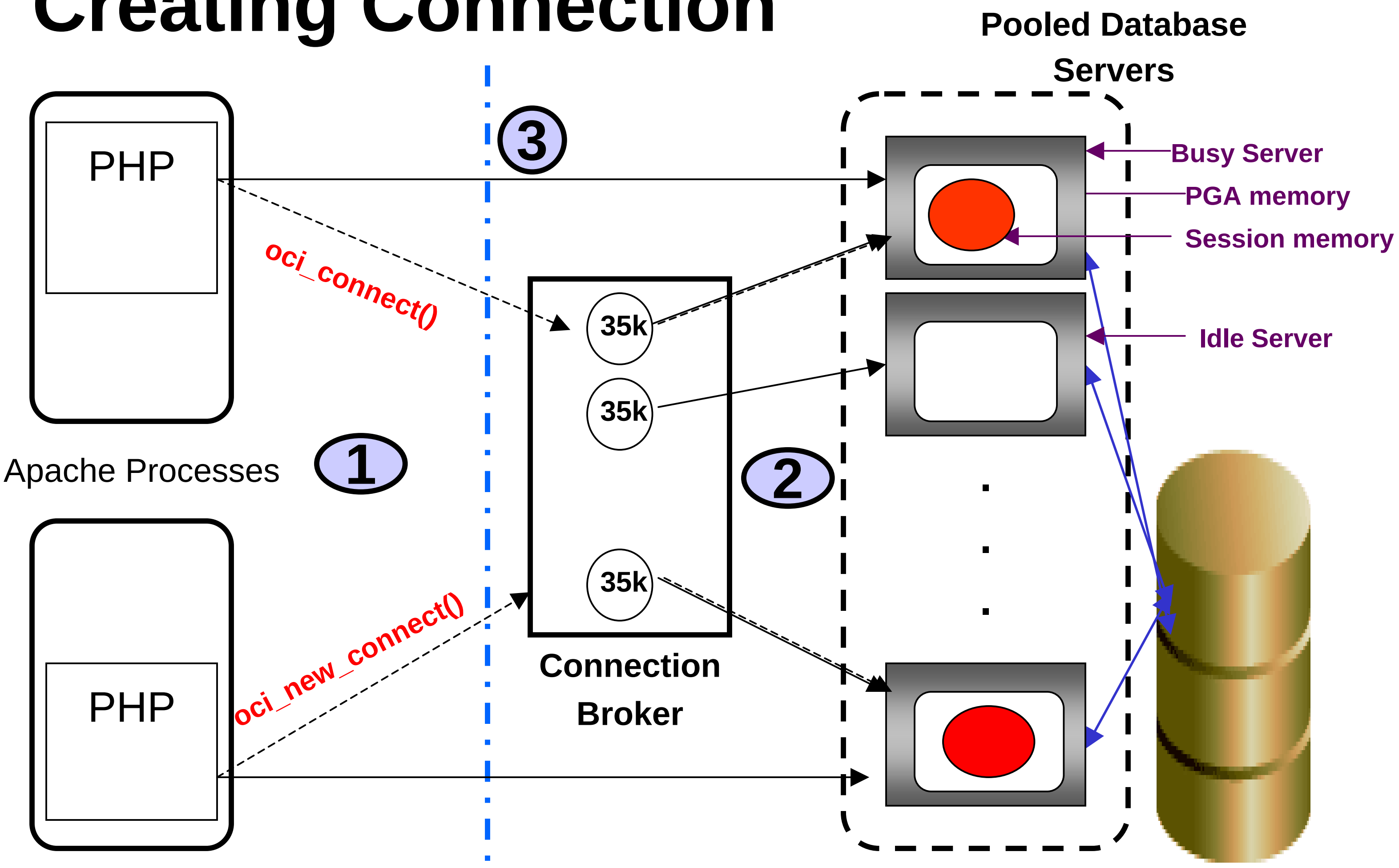
Database Resident Connection Pool

- Pools dedicated servers on database side
- Shares connections across apache processes
- Shares connections across middle tier nodes
- Co-exists in all database server configurations
  - Dedicated Servers, Shared Servers, RAC
- Exposed to PHP

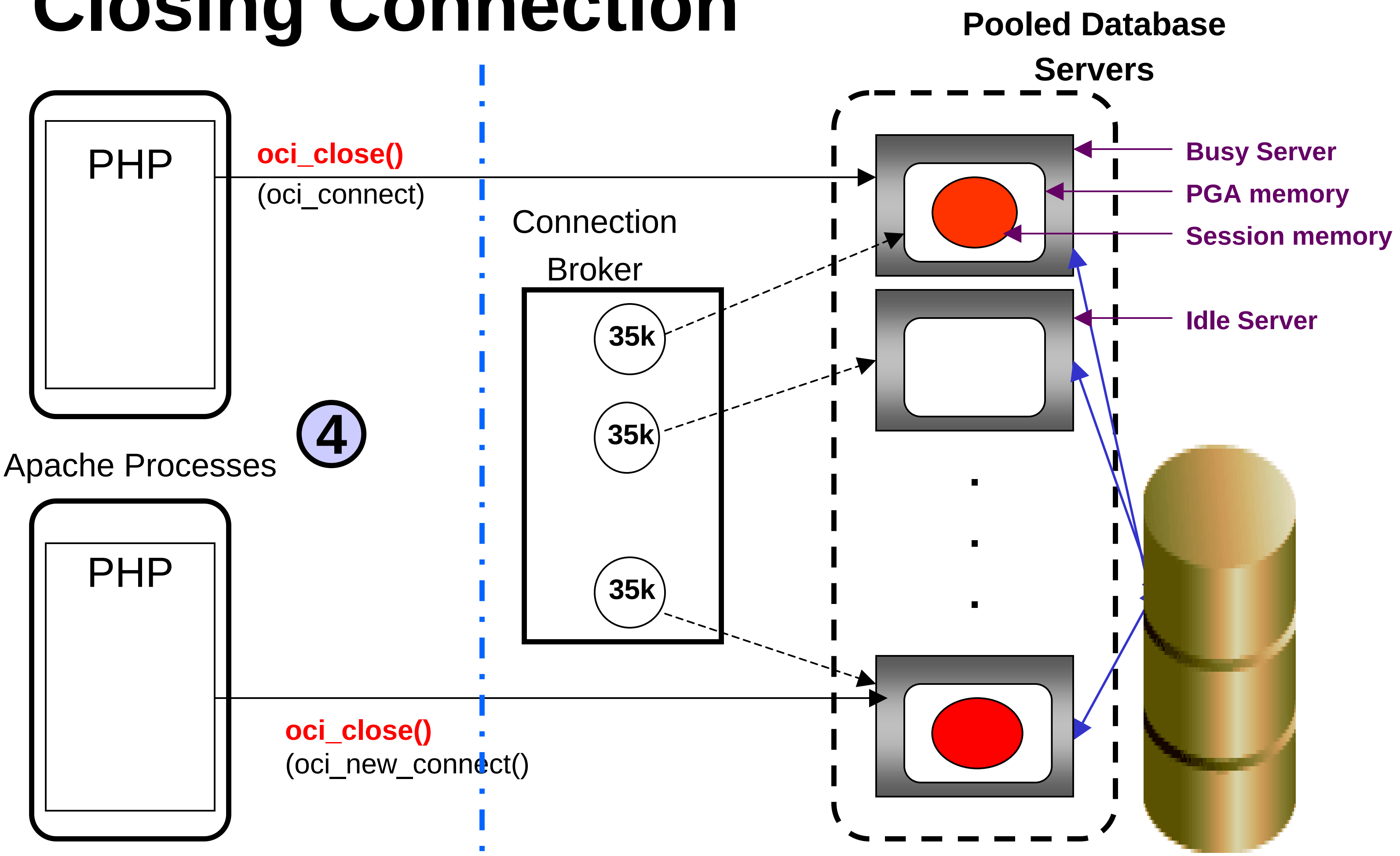
DRCP with Non-Persistent

Connections

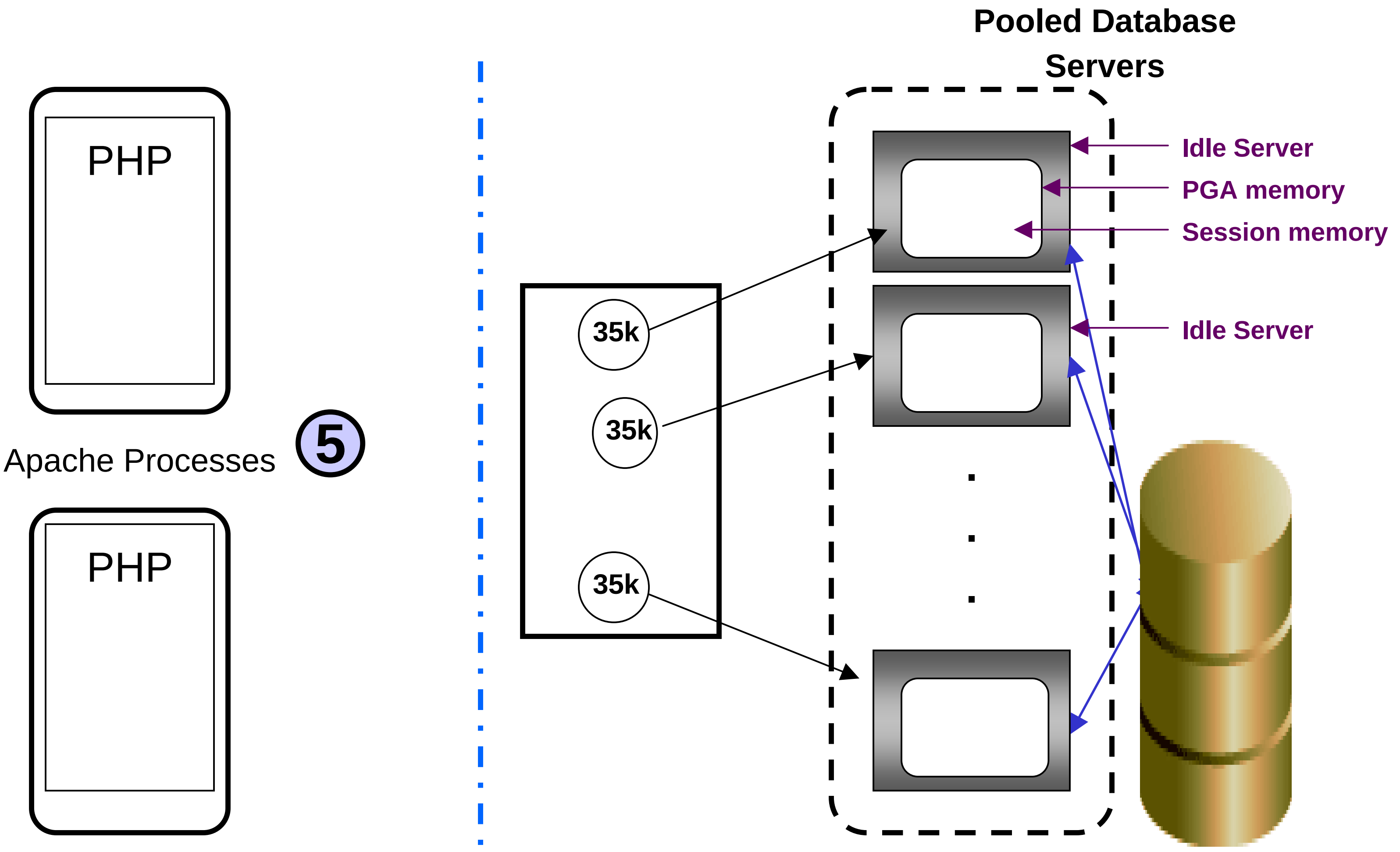
DRCP with Non-Persistent Connection Creating Connection



DRCP with Non-Persistent Connection Closing Connection

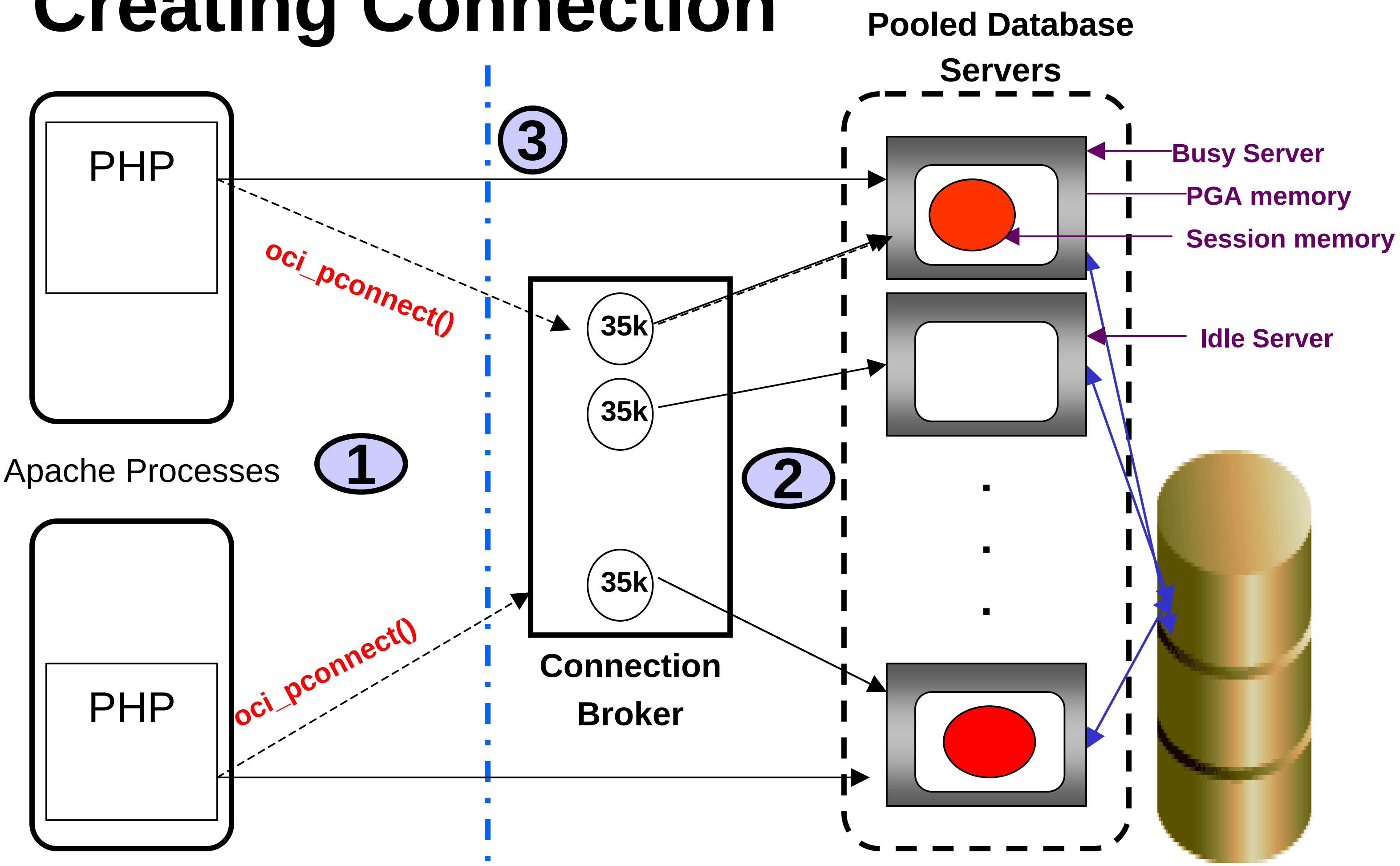


DRCP with Non-Persistent Connection

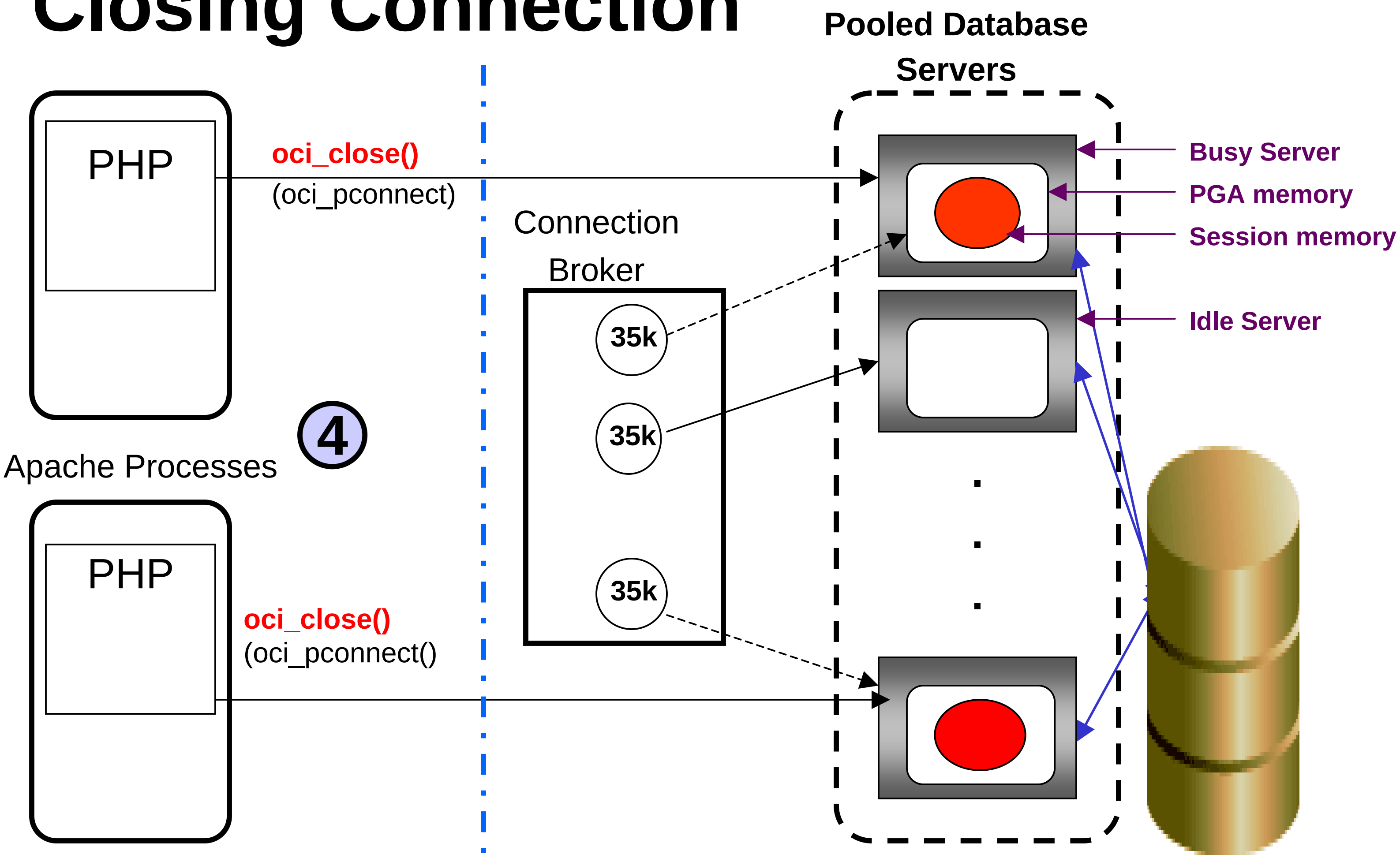


DRCP with Persistent Connections

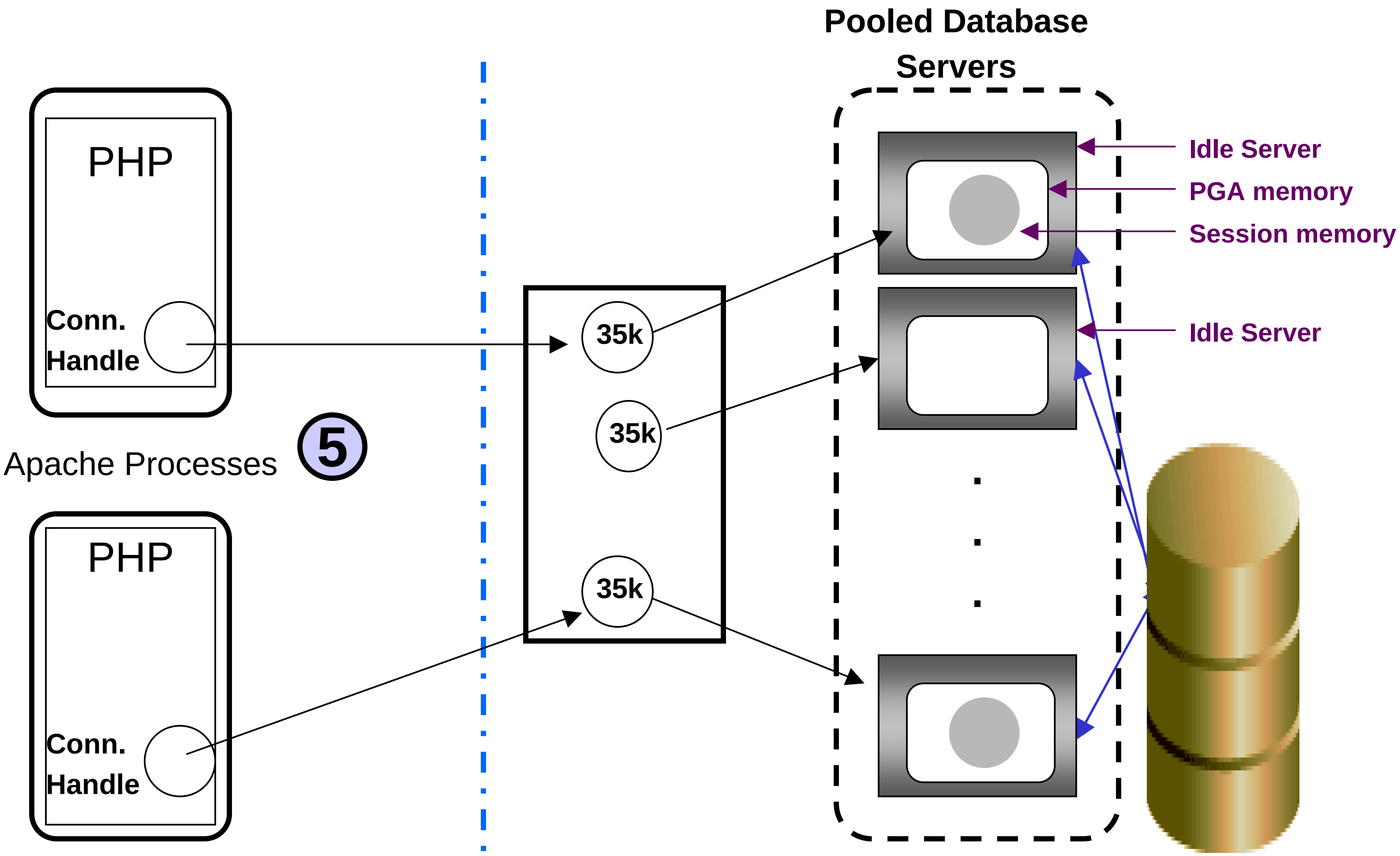
DRCP with Persistent Connections
Creating Connection



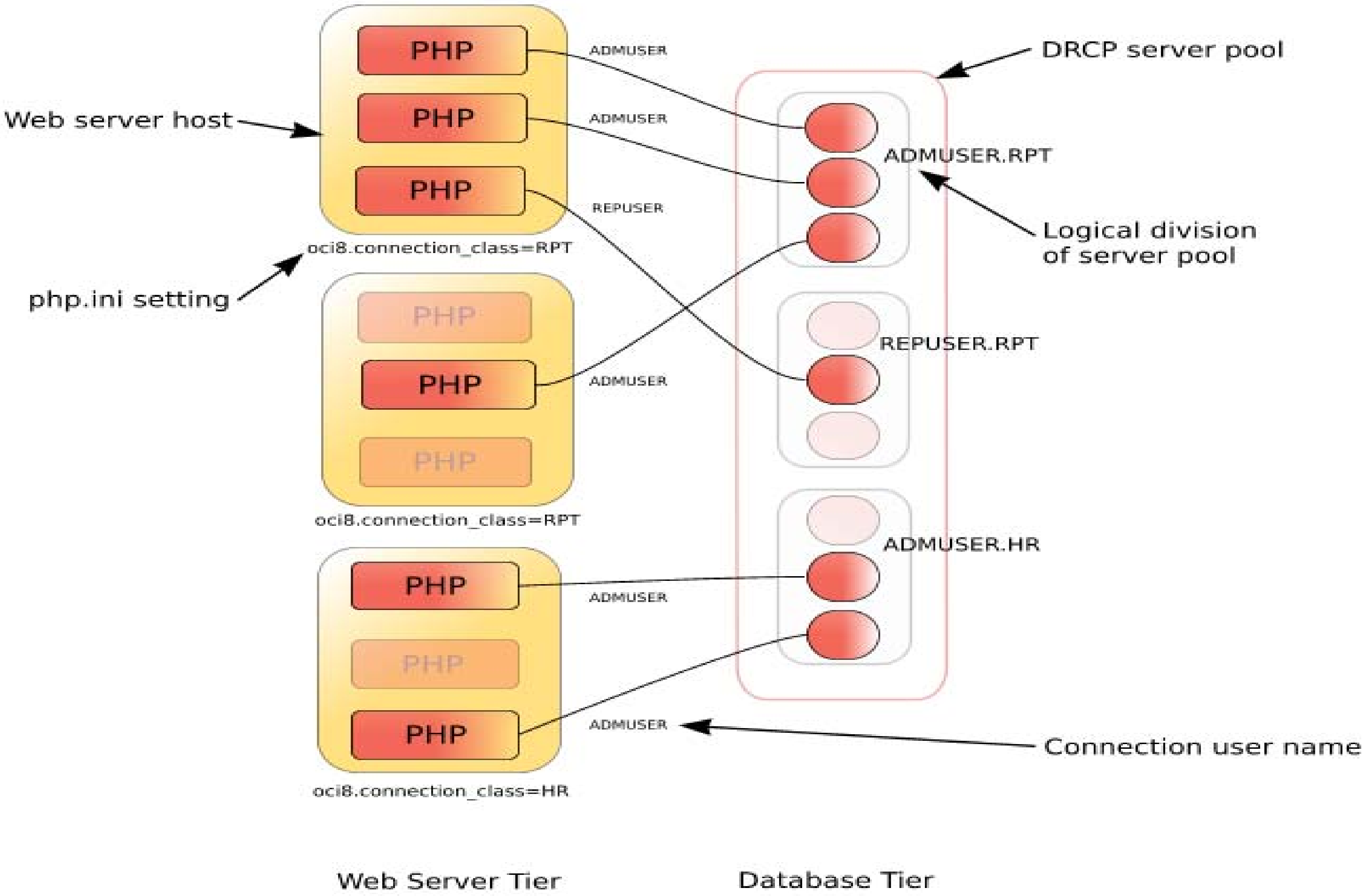
DRCP with Persistent Connections
Closing Connection



DRCP with Persistent Connections



Sub-Partitioning the Pool



Configuring and Using DRCP

Download the new PHP OCI8 1.3.1 (currently beta)

1. Configure & Start the Pool

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

2. Set oci8.connection_class = “<*your application*>"

   ```
   oci8.connection_class = MY_APP_CC
   ```

3. Specify “*:POOLED*” in the connect string

   ```
   $c = oci_pconnect('myuser', 'mypassword',
          'myhost/sales:POOLED');
   ```

That’s it!

Configuring the POOL

```
SQL> execute dbms_connection_pool.configure_pool(
  2         pool_name =>
   'SYS_DEFAULT_CONNECTION_POOL',
  3         minsize                => 4,
  4         maxsize                => 40,
  5         incrsize               => 2,
  6         session_cached_cursors => 20,
  7         inactivity_timeout     => 300,
  8         max_think_time         => 600,
  9         max_use_session        => 500000,
 10         max_lifetime_session   => 86400);
```

DRCP Availability in PHP

- DRCP is available in the new OCI8 1.3.1 beta
- Must be linked with Oracle database 11g client libraries against an Oracle database 11g
- Default in PHP 5.3 and PHP 6
- PHP 5.2.4+
  - Build PHP with DRCP-enabled OCI8 1.3.1 Beta from PECL under PHP 5.2.4+ ext/oci8; configure, build, and install PHP as normal
  - Alternatively use *phpize* and build OCI8 as a shared module

- Note: The new OCI8 extension is backward compatible with older Oracle clients libraries

Sample Sizing for 5000 Clients

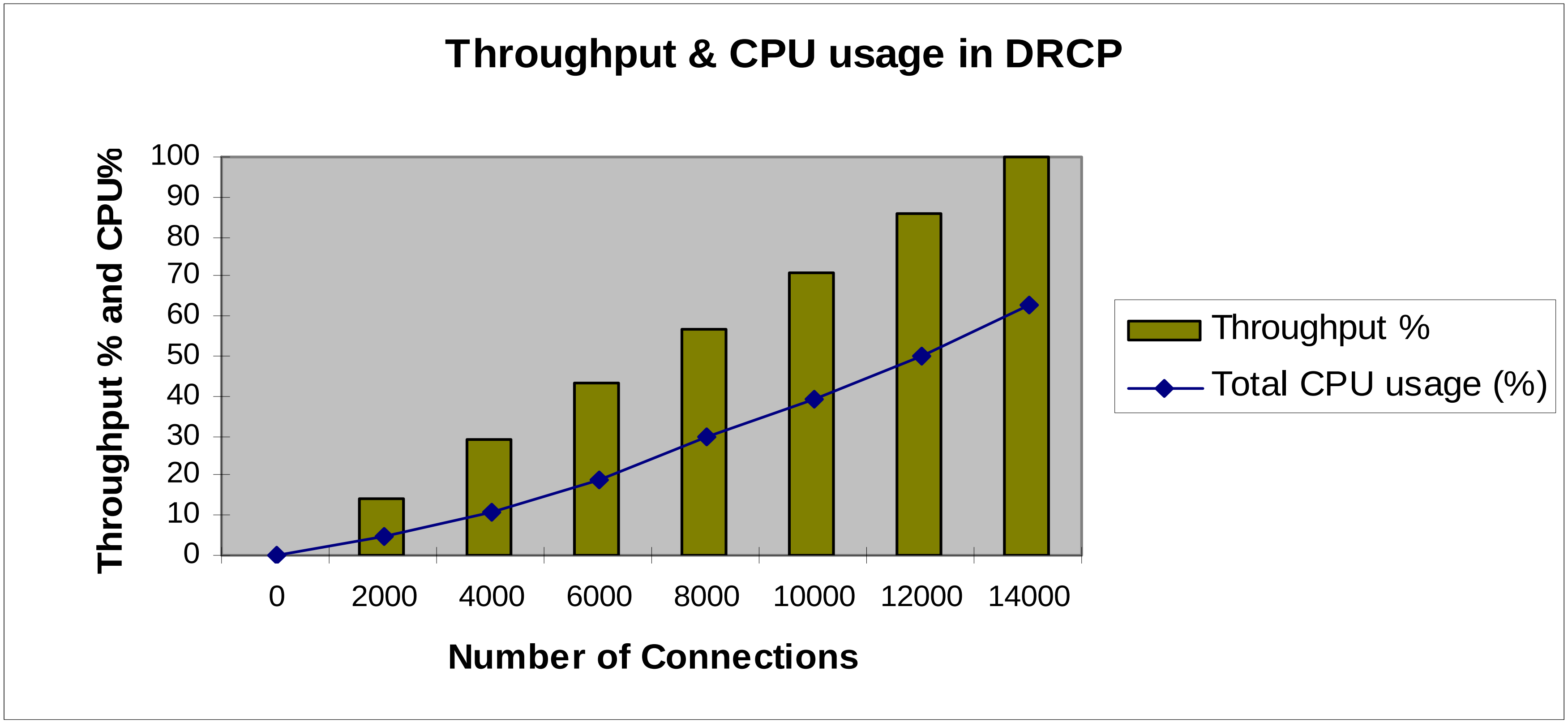
	Dedicated Servers	Shared Servers	_DRCP Servers_
Database Servers	_5000 * 4 MB_	_100 * 4 MB_	_100 * 4 MB_
Session Memory	_5000 * 400 KB_	_5000 * 400 KB_	_100 * 400 KB_
DRCP Memory Overhead			_5000 * 35 KB_
Total Memory	22 GB	2.5 GB	_615 MB_

DRCP Scalability Benchmark

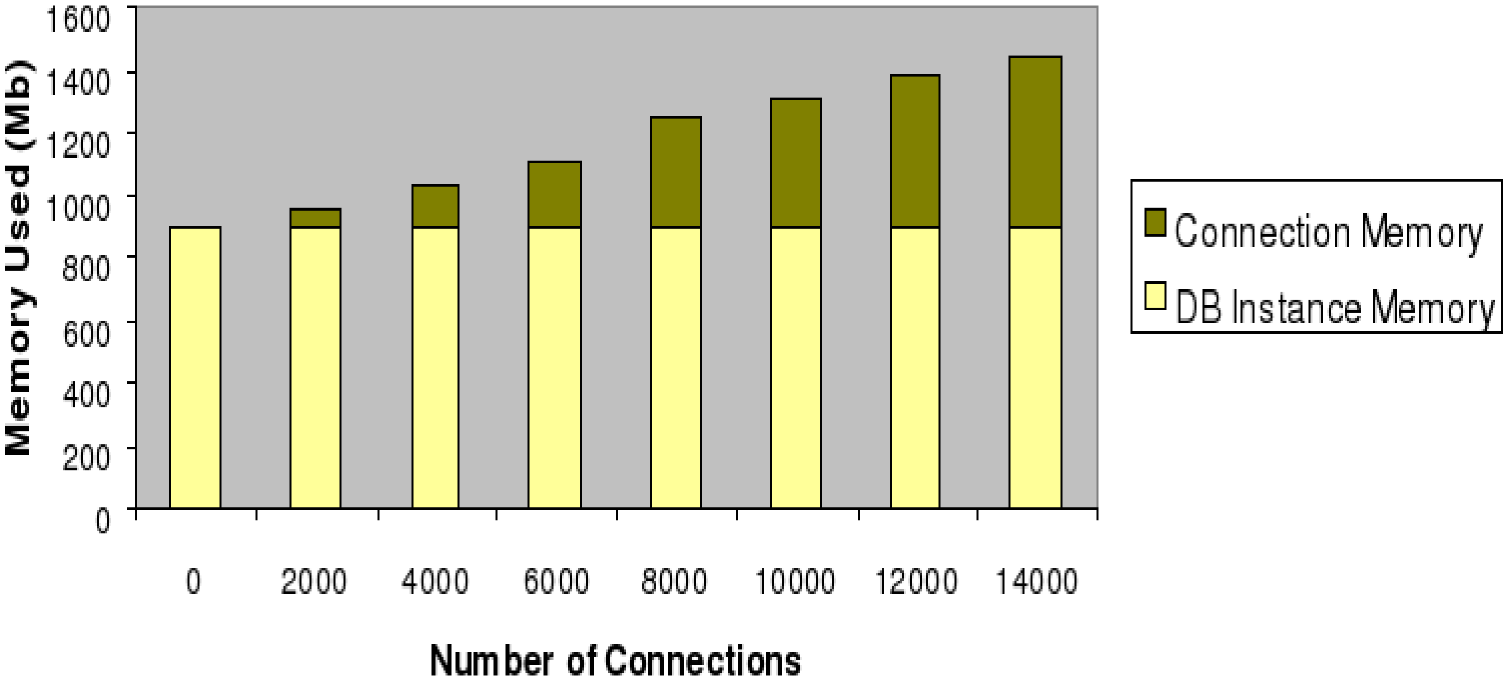
Database Server

- 4 CPU Intel Xeon MP 2.80GHZ
- 2GB RAM
- Red Hat Enterprise Linux 4
- Oracle database 11*g* R1
- DRCP
  - 1 Connection Broker
  - 100 Pooled Servers.

DRCP Scalability Benchmark



DRCP Scalability Benchmark



Agenda

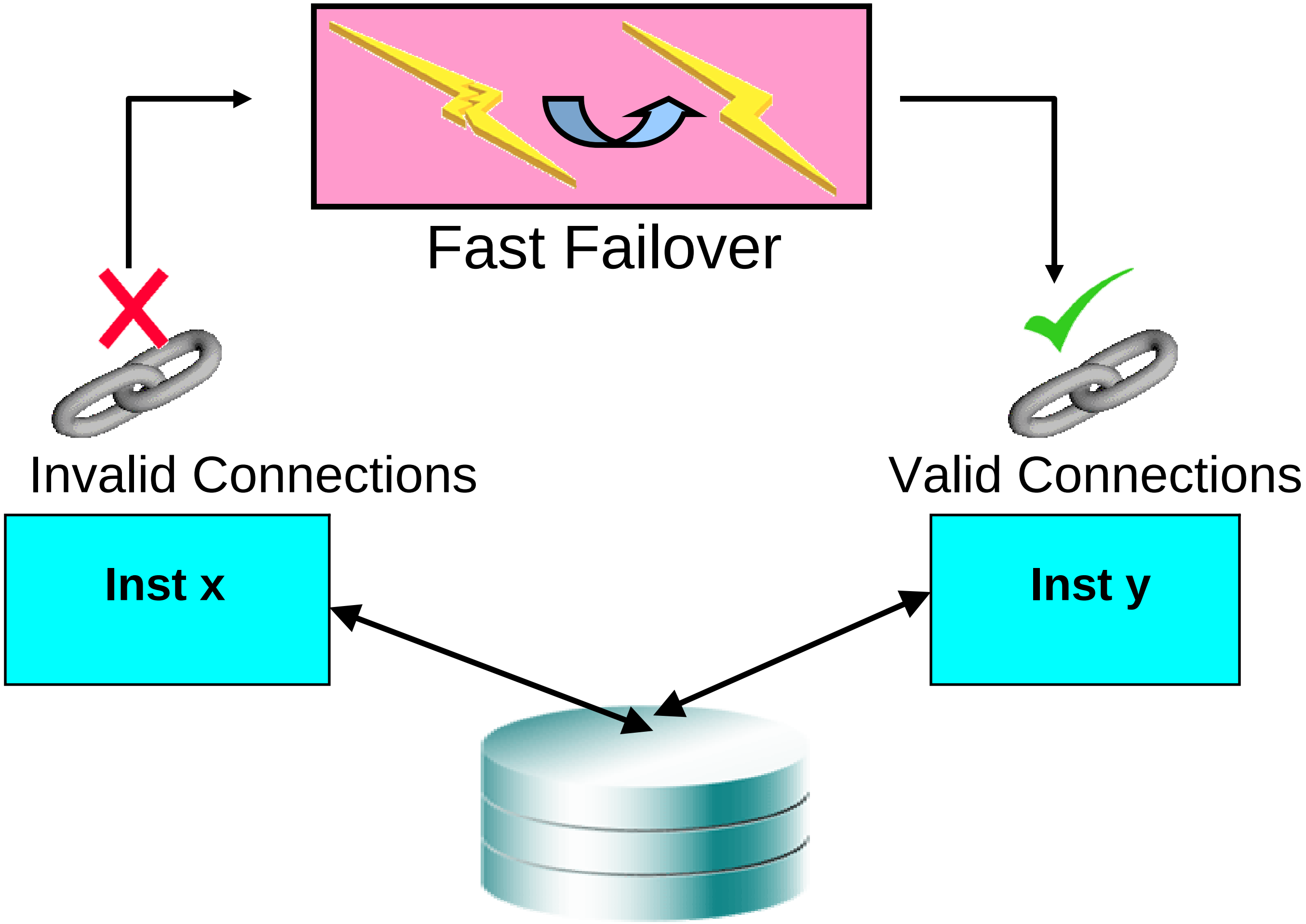
- Requirements - Motivations
- Traditional Oracle/PHP Connection
- Web Scale Connection Broker

✓High Availability for PHP

  - Fast Application Notification of Events (FAN)
  - Fast Failover

High Availability for PHP

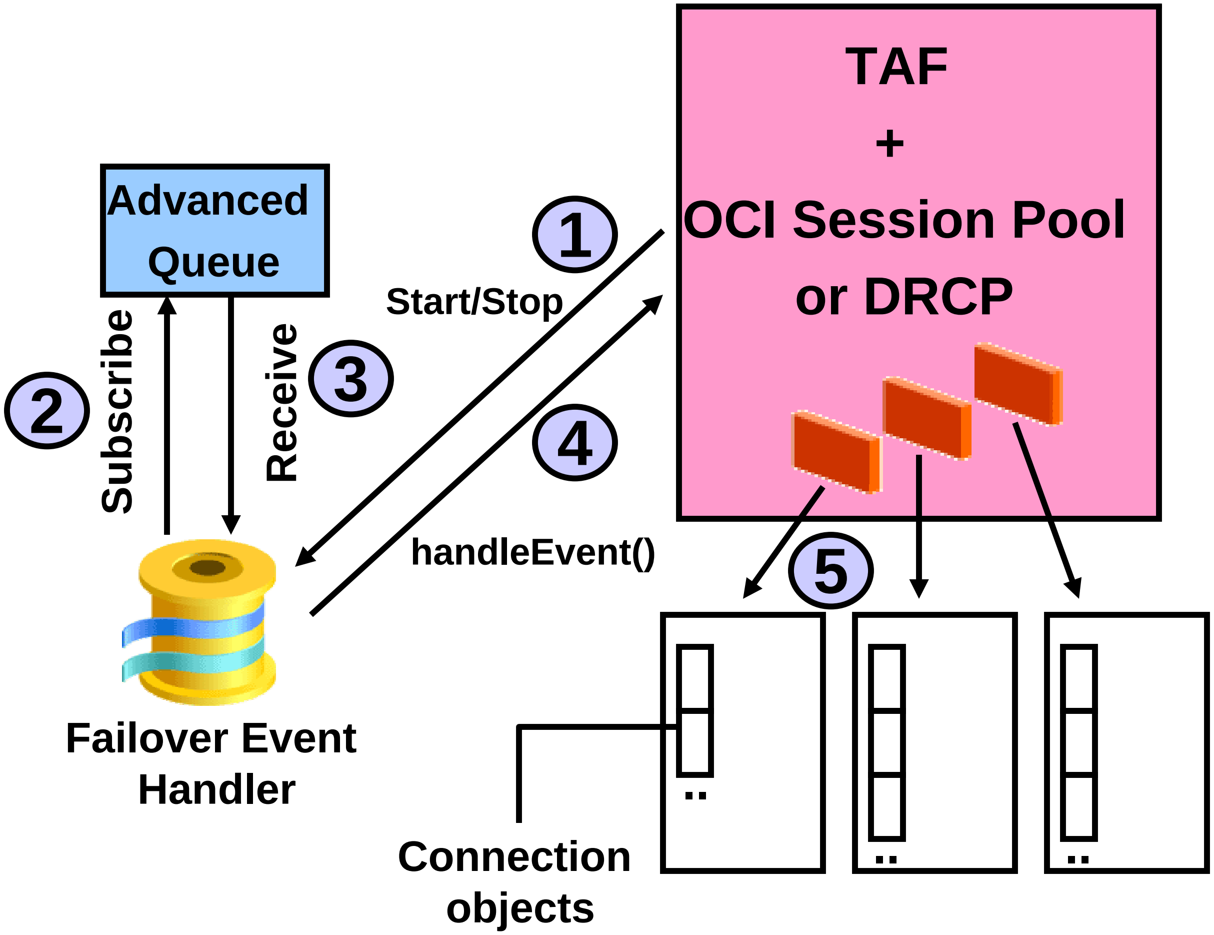
High Availability – Application View



High Availability –
Fast Application Notification of Events

- Pub/Sub event notification
- High Availability feature for PHP with RAC or Data-Guard-with-physical-standby
- When DB node or network fails
  - Database generates FAN events
  - PHP error returned without TCP timeout delay
  - PHP application can reconnect to surviving instance
- OCI8 1.3.1 Beta supports FAN

High Availability – Fast Fail-Over



Fast Failover – Configuration

- Subscription to RAC Events

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

- Configure PHP (php.ini)

```
oci8.events = On
```

- OCI Session Pool transparently removes invalid connections from the pool
- Works with and without DRCP

Fast Failover – PHP

The Application

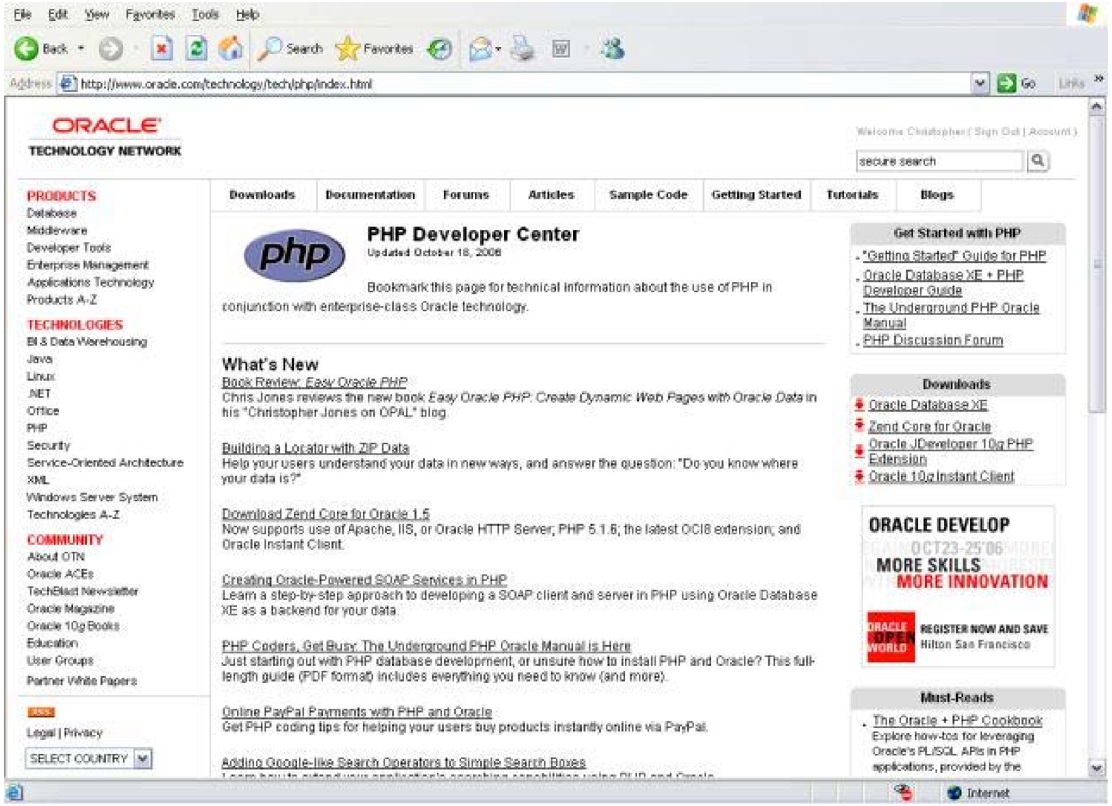
```
. . .
$conn = doConnect();
$err = doSomeWork($conn);
. . .
if (isConnectionError($err)) {
   // reconnect and retry
   $conn = doConnect();
   $err = doMoreWork($conn);
}
if ($err) {
  // probably some other issue
  handleError($err);
}
```

Fast Failover – PHP

```
function isConnectionError($err)
{
  switch($err)
  {
    case 3113: // ORA-3113
    case 3114:
    case 3125:
    case 3122:
    case 1033:
    case 1034:
    ...
    case 12153:
  return true;
   }
  return false;
}
```

Oracle Technology Network
PHP Developer Center

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



PHP Developer Center

Updated October 18, 2006

Bookmark this page for technical information about the use of PHP in conjunction with enterprise-class Oracle technology.

What's New

Book Review: *Easy Oracle PHP*
Chris Jones reviews the new book *Easy Oracle PHP: Create Dynamic Web Pages with Oracle Data* in his "Christopher Jones on OPAL" blog.

Building a Locator with ZIP Data
Help your users understand your data in new ways, and answer the question: "Do you know where your data is?"

Download Zend Core for Oracle 1.5
Now supports use of Apache, IIS, or Oracle HTTP Server; PHP 5.1.6; the latest OCI8 extension; and Oracle Instant Client.

Creating Oracle-Powered SOAP Services in PHP
Learn a step-by-step approach to developing a SOAP client and server in PHP using Oracle Database XE as a backend for your data.

PHP Coders, Get Busy: The Underground PHP Oracle Manual is Here
Just starting out with PHP database development, or unsure how to install PHP and Oracle? This full-length guide (PDF format) includes everything you need to know (and more).

Online PayPal Payments with PHP and Oracle
Get PHP coding tips for helping your users buy products instantly online via PayPal.

Adding Google-like Search Operators to Simple Search Boxes

Get Started with PHP

- "Getting Started" Guide for PHP
- Oracle Database XE + PHP Developer Guide
- The Underground PHP Oracle Manual
- PHP Discussion Forum

Downloads

- Oracle Database XE
- Zend Core for Oracle
- Oracle JDeveloper 10g PHP Extension
- Oracle 10g Instant Client

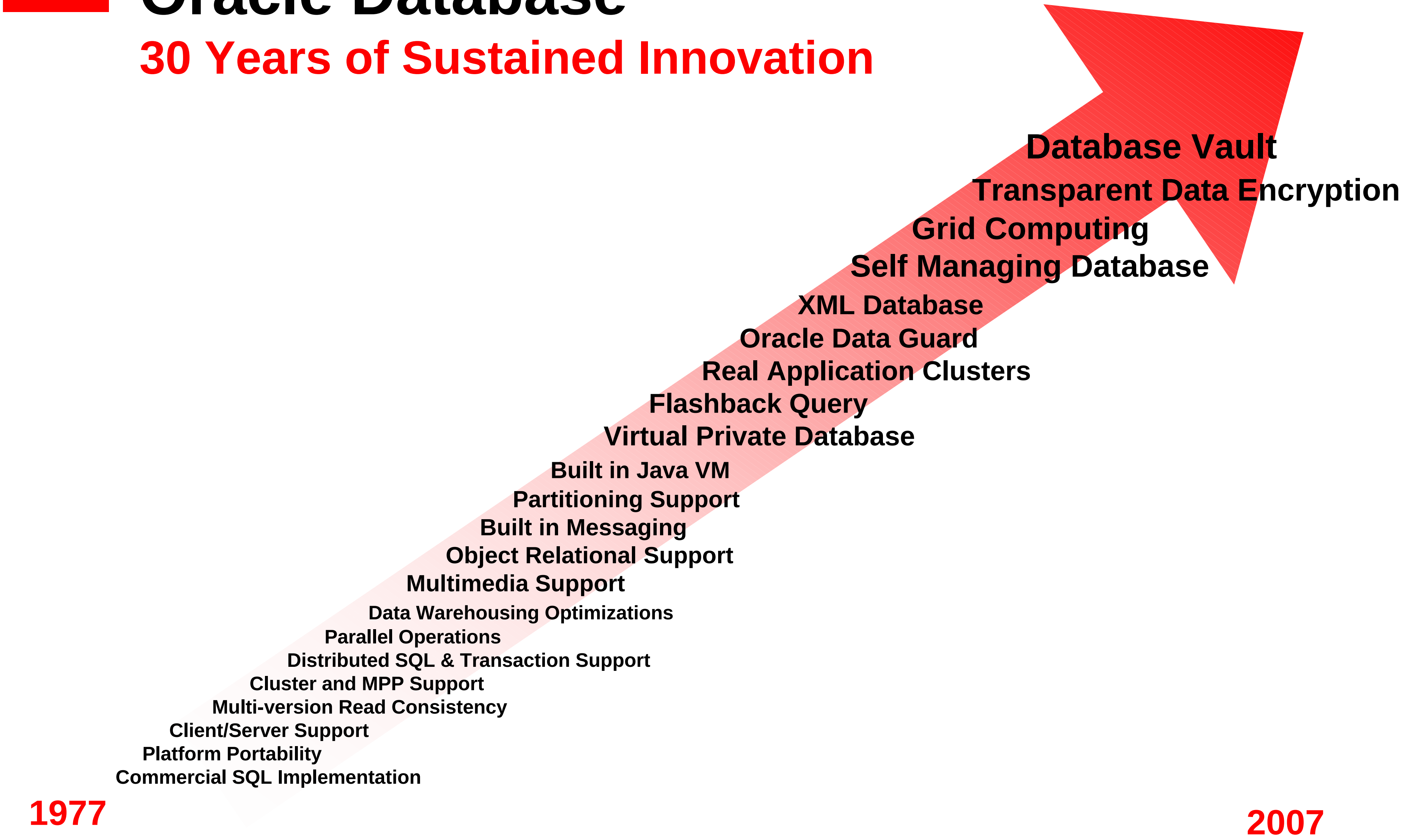
Must-Reads

- The Oracle + PHP Cookbook
  Explore how-tos for leveraging Oracle's PL/SQL APIs in PHP applications, provided by the

oracle.com/technology/php

Oracle Database

30 Years of Sustained Innovation



Database Vault

Transparent Data Encryption

Grid Computing

Self Managing Database

XML Database

Oracle Data Guard

Real Application Clusters

Flashback Query

Virtual Private Database

Built in Java VM

Partitioning Support

Built in Messaging

Object Relational Support

Multimedia Support

Data Warehousing Optimizations

Parallel Operations

Distributed SQL & Transaction Support

Cluster and MPP Support

Multi-version Read Consistency

Client/Server Support

Platform Portability

Commercial SQL Implementation

1977

2007