Developing and Deploying High Performance PHP Applications

http://joind.in/3396

php|tek 2011
Christopher Jones
Oracle Development

http://blogs.oracle.com/opal
christopher.jones@oracle.com
http://twitter.com/ghrd

This Talk - What to Expect

- “State of the Nation” overview of PHP & Oracle Technology
- (Some) Best Practices with Oracle Database
  - With live demos

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.

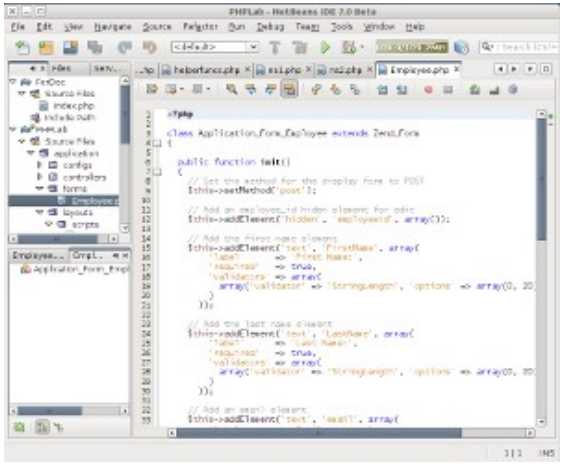
About Me

- Work in Oracle's Linux & Virtualization Group
- Focus on scripting languages – mostly PHP
  - Have PHP code check-in privileges
- Also work on Oracle features that help scripting languages

In The News

In the News: NetBeans IDE

- Free
  - GPL / CDDL
- NetBeans is very popular for PHP
  - Strong, passionate community of users
- Zend Framework, Symfony, XDebug, PHPUnit, PhpDoc
- New! NetBeans 7.0 just released
  - Generate PhpDoc
  - Rename refactoring, Safe delete refactoring
  - Support for PHP 5.3 namespace aliases
- See http://netbeans.org/features/php

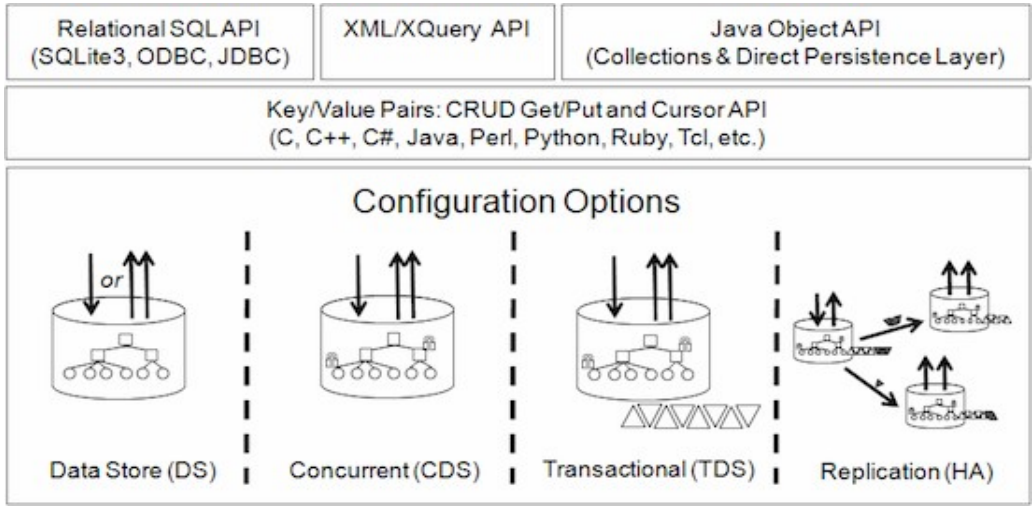


In the News: Oracle Database

- Beta Oracle 11.*2* “XE” is available
  - Free
  - Windows 32 & 64, Linux 64
  - Subset of Enterprise Edition features
  - Same code base as EE
  - XE applications can run unchanged against EE
  - Has DRCP connection pooling
- Oracle RDS on Amazon
  - Announced Tuesday

In the News: Oracle Berkeley DB

- Open source embedded database engine
  - Key/Value, Relational, XQuery, Java Object storage
  - Handles a few bytes to terabytes



- Existing PHP extensions
  - PHPs **DBA** extension
  - BDB ships source code for a **php_db4** extension

In the News: Oracle Berkeley DB

- New! BDB 5 has a SQLite-like API
  - Better than SQLite under load or heavy writes
  - Tunable
- BDB have joined SQLite Consortium
- Use BDB for PHP's **sqlite3** and **PDO_SQLITE**
  - Configure BDB 5's SQLite API:

```
configure –prefix=$HOME/bdb –enable-sql_compat
```

Configure PHP to use BDB:

```
configure --with-sqlite3=$HOME/bdb \
          --with-pdo-sqlite=$HOME/bdb
```

In the News: Oracle TimesTen In-Memory Database

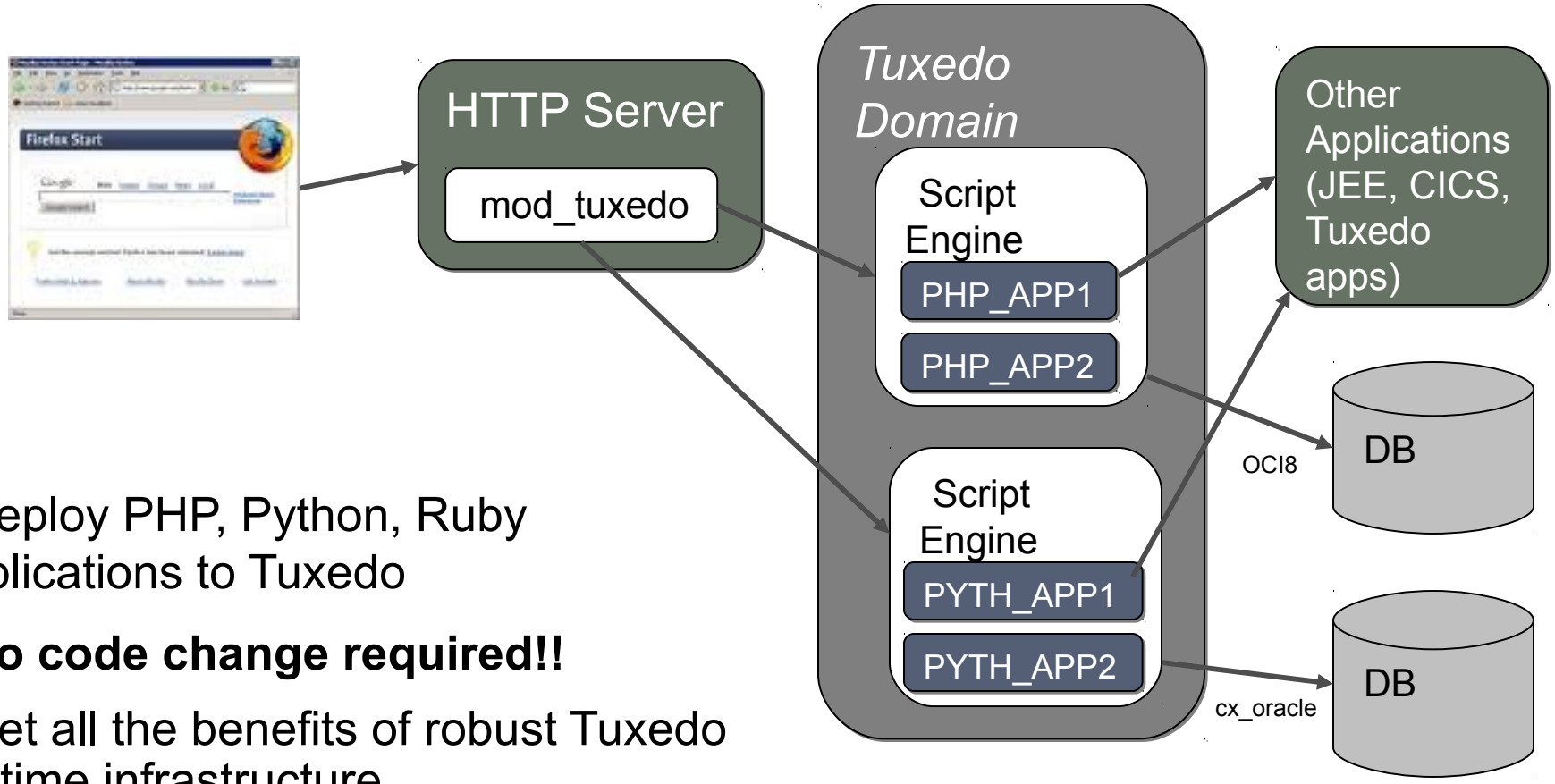
- TimesTen is an In-memory DB
  - Stand-alone DB with persistent storage, or
  - Cache for Oracle DB
- PHP OCI8 applications can connect to Oracle DB or TimesTen
  - New! PHP OCI8 with Oracle 11.2.0.2 libraries supports more TimesTen features

In the News: Oracle Tuxedo

- #1 Application Server for C, C++, COBOL, Python and Ruby
- New! PHP Support
- Tuxedo lets you scale up faster, with better resource monitoring, and great support for legacy applications.
- Preconfigured VirtualBox VM available:
  - http://www.oracle.com/technetwork/community/developer-vm/
- Whitepaper available
  - http://bit.ly/kqy5Gz

Tuxedo: Application Server for Dynamic Languages

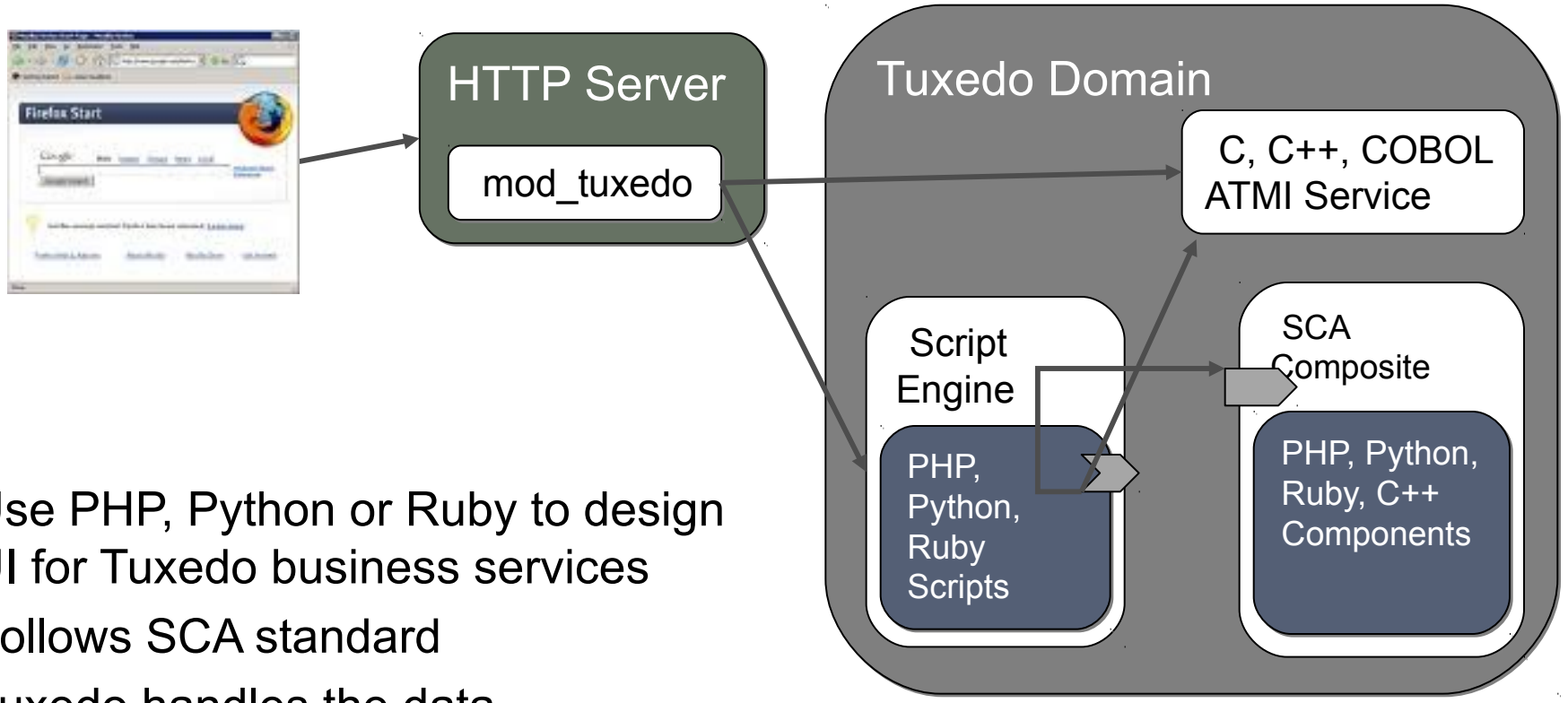
Robust Runtime for Deploying Web Applications



- Deploy PHP, Python, Ruby applications to Tuxedo
- **No code change required!!**
- Get all the benefits of robust Tuxedo runtime infrastructure
- FastCGI-like and much more

Existing Tuxedo Business Services

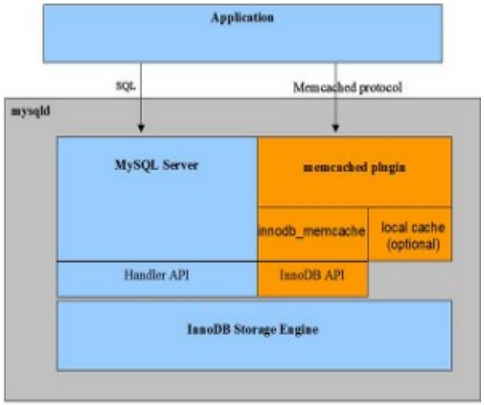
Dynamic Page Server



- Use PHP, Python or Ruby to design UI for Tuxedo business services
- Follows SCA standard
- Tuxedo handles the data-interchange

In the News: MySQL Database

- Query Cache Plugin for mysqlnd
  - PECL Stable
  - APC, Memcache, SQLite
- New! Replication & Load Balancing plugin for mysqlnd
  - PECL Alpha
- MySQL 5.5 available
  - InnoDB becomes default storage engine
- New! MySQL 5.6 Dev
  - NoSQL to InnoDB via memcached



In the News: Oracle Linux & the "UEK" kernel

- Oracle Linux is based on RHEL
  - New! Has PHP 5.3.3 RPMs
- Zend Server from Oracle Linux package site
- Linux 2.6.32 kernel optionally available for OL 5.5+
  - 75% faster on multi-core machines, better etc.
  - Standard kernel from RH is 2.6.18
  - Upgrade performance without having to reinstall OS
  - Applications run unchanged

Always In the News: SQL Injection

```
$user_input = '1234 OR TRUE';  // unfiltered form input
$sql = 'select secret from mytab where key = ' . $user_input;
```

- **A top security risk**
- Use prepared statements aka bind variables
- Filter input

In the News: Summary

php|tek USB has links

- NetBeans IDE – *7.0*
- Oracle Database – *Amazon RDS and XE 11gR2 Beta*
- Tuxedo Application Server – *PHP Support*
- Berkeley DB – *SQLite-like API*
- TimesTen In-Memory Cache – *OCI8 extension*
- MySQL – *5.5 GA*
- Oracle Linux – *PHP 5.3 and 2.6.32 kernel*

High Performance With Oracle Database

PHP OCI8 and Oracle Database

What is the PHP OCI8 Driver?

- Main Oracle Database extension for PHP
- Open source and part of PHP
- Current version is OCI8 1.4 – in PHP 5.3 and in PECL

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

PHP OCI8 Extension for Oracle Database

- Linux install with

```
$ configure … --with-oci8=instantclient,$HOME/instantclient_11_2
```

or

```
$ pecl install oci8
. . .
Please provide the path . . . :
instantclient,/home/cjones/instantclient_11_2
```

- Windows DLLs available with PHP
- From ULN for Oracle Linux
- Included in Zend Server

Some Caching & Buffering Features

OCI8 Insert/Fetch Best Practices Demo

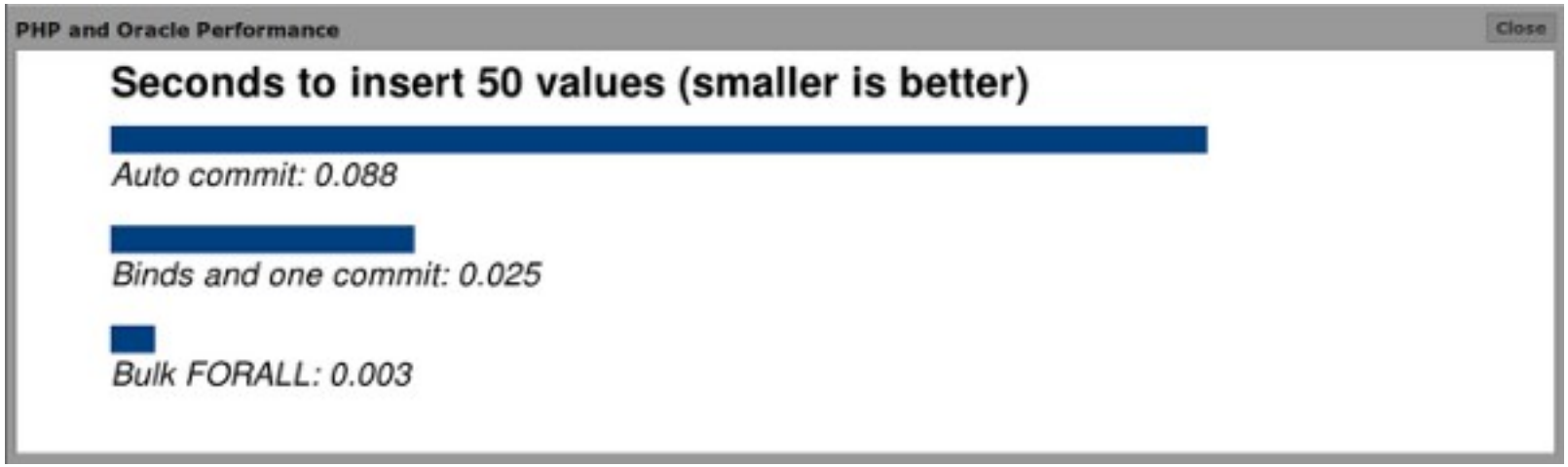
Insert Performance Tips Code (1)

```
function do_ins_basic($c, $a) {
  foreach ($a as $v) {
    $s = oci_parse($c,
              "insert into ptab (pdata) values ('".$v."')");
    $r = oci_execute($s);  // Auto commits
  }
}
function do_ins_bind_trans($c, $a) {
  $s = oci_parse($c, "insert into ptab (pdata) values (:bv)");
  oci_bind_by_name($s, ':bv', $v, 20, SQLT_CHR);
  foreach ($a as $v)
    $r = oci_execute($s, OCI_NO_AUTO_COMMIT);
  oci_commit($c);
}
```

Insert Performance Tips Code (2)

```
function do_ins_forall($c, $a)
{
  $s = oci_parse($c, "begin inspkg.insforall(:c1); end;");
  oci_bind_array_by_name($s, ":c1", $a, count($a), -1, SQLT_CHR);
  oci_execute($s);
}
```

Insert Performance Tips Results



Fetch Performance Tips Code (1)

```
function do_prefetch($c, $pf)
{
  $stid = oci_parse($c, "select mycol from bigtab");
  oci_set_prefetch($stid, $pf);
  oci_execute($stid);
  oci_fetch_all($stid, $res);
  return $res;
}
```

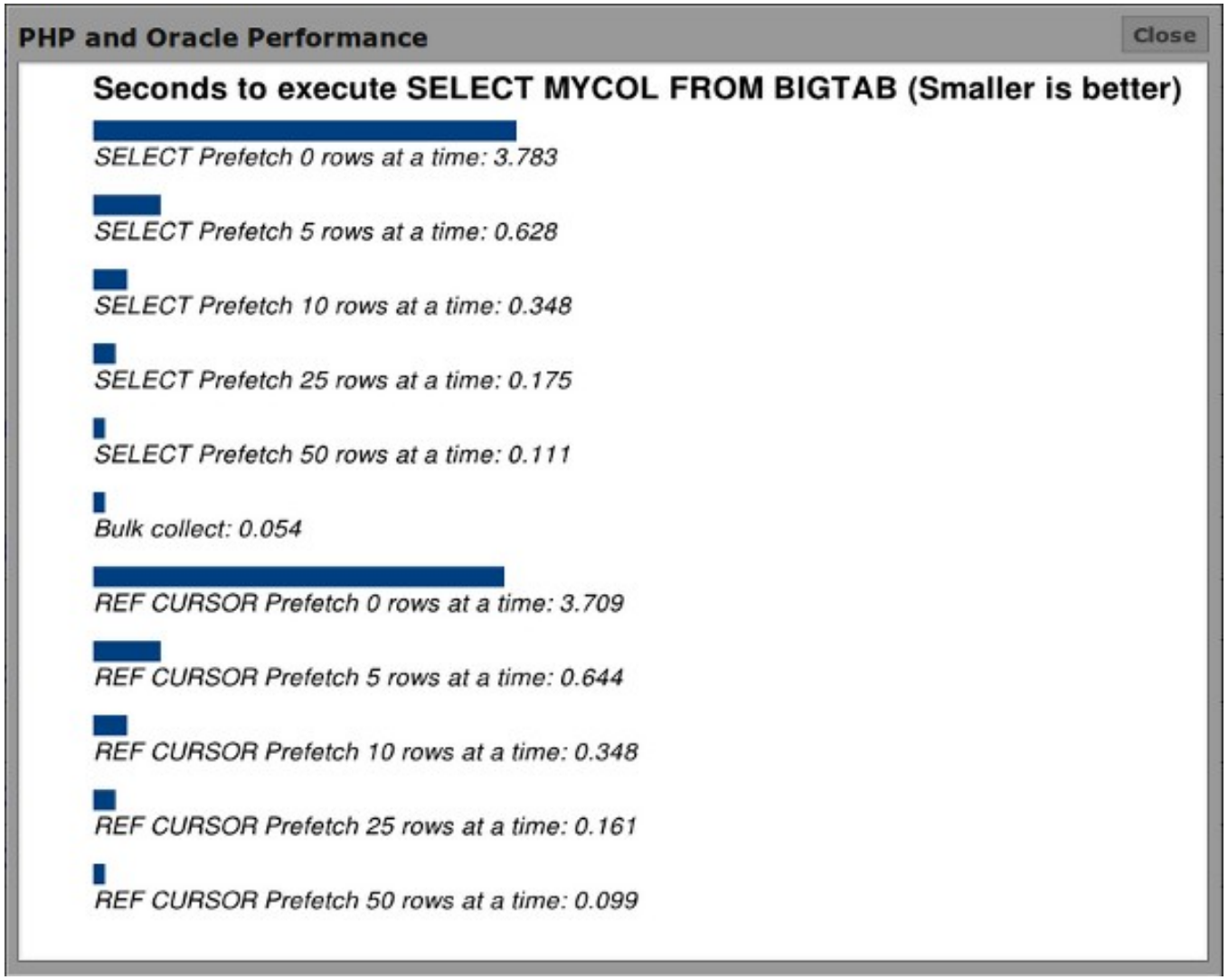
Fetch Performance Tips Code (2)

```
function do_ref_cur($c, $pf)
{
  $stid = oci_parse($c,
             "begin fetchperfpkg.refcurprc(:rc); end;");
  $refcur = oci_new_cursor($c);
  oci_bind_by_name($stid, ':rc', $refcur, -1, OCI_B_CURSOR);
  oci_execute($stid);
  oci_set_prefetch($refcur, $pf);
  oci_execute($refcur);
  oci_fetch_all($refcur, $res);
  return $res;
}
```

Fetch Performance Tips Code (3)

```
function do_sel_bulk($c)
{
  $s = oci_parse($c,
             "begin fetchperfpkg.bulkselectprc(:a1); end;");
  oci_bind_array_by_name($s, ":a1", $res, 20000, 20, SQLT_CHR);
  oci_execute($s);
  return($res);
}
```

Fetch Performance Tips Results



Oracle 11gR2 Client & Server Result Caches

- "Client" means query result caching in PHP process
- In Oracle 11gR1, developer adds hint to table query:

```
select /*+ result_cache */ last_name from employees
```

- In Oracle 11gR2 the DBA can choose tables or view to be cached:

```
create table sales (...) result_cache

alter table last_name result_cache

create view v2 as
   select /*+ result_cache */ col1, coln from t1
```

No Need to Change Application

Client Side Result Cache Demo

Client Result Cache Load Test

Client Side Result Cache Load Test

Running: `ab -n 1000 -c 10 http://localhost/crc.php`

Benchmark Output

Without Caching: **Requests per second: 485.45 [#/sec] (mean)**

With Caching: **Requests per second: 544.03 [#/sec] (mean)**

Execution Report for CRCPARTS

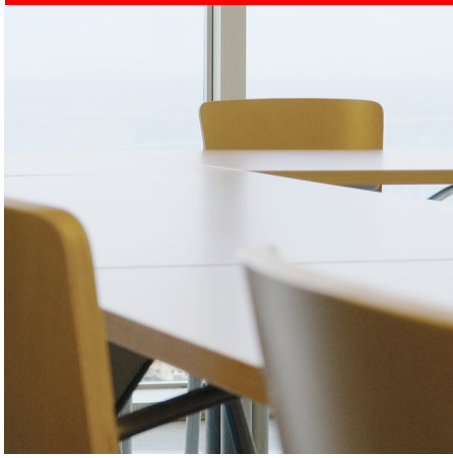
Parse Calls	Executions	SQL Text
1000	1000	SELECT /*+ NO_RESULT_CACHE */ * FROM CRCPARTS
8	8	SELECT * FROM CRCPARTS

This report shows what the DB recorded - PHP OCI8 did not need to do the underlying execution of statements that had cached results.

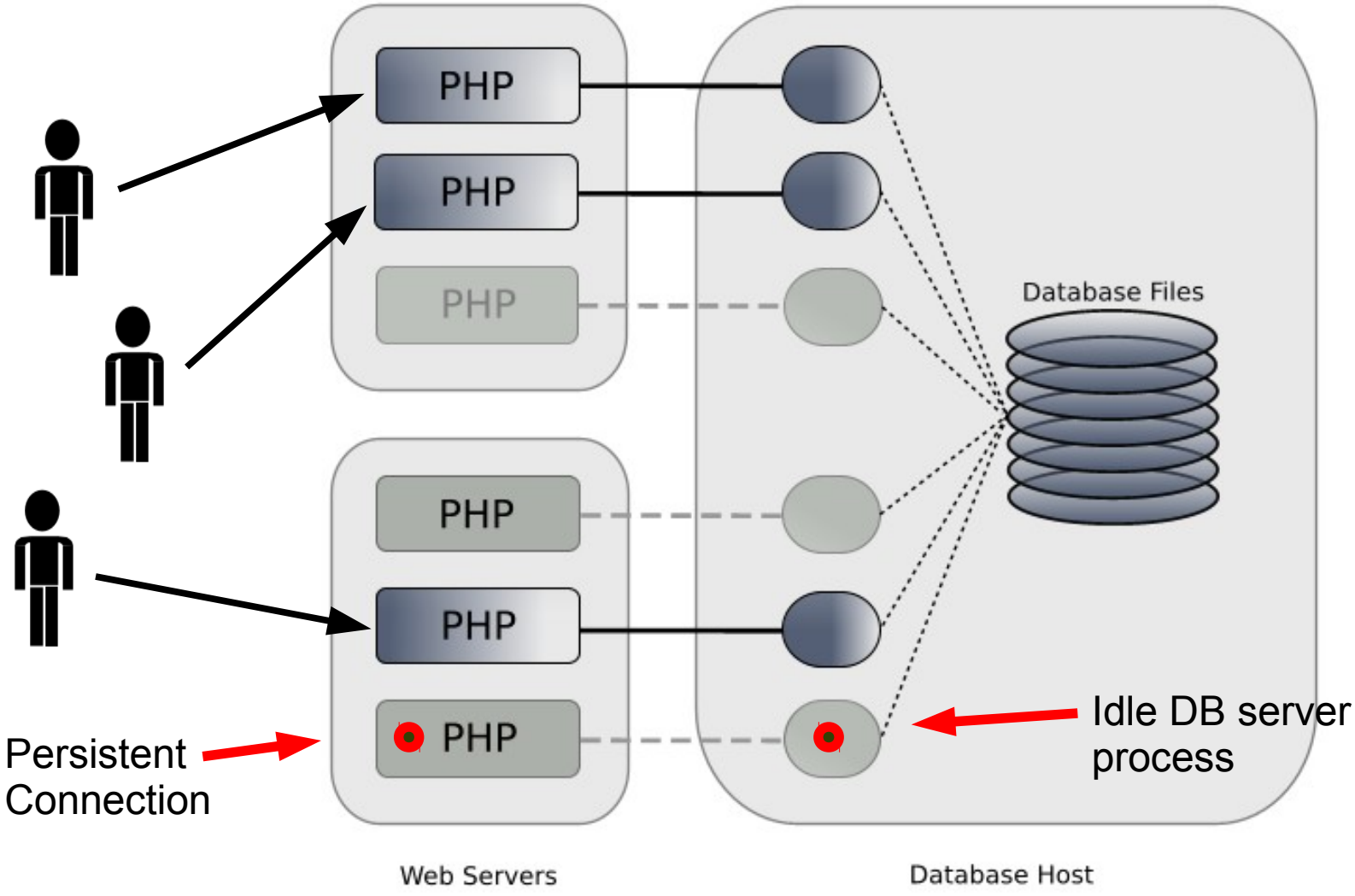
Using CRC

- **Improves PHP performance**
- **Reduces network load between PHP and Oracle Database**
- **Reduces DB server load**

Database Resident Connection Pooling in Oracle 11*g*

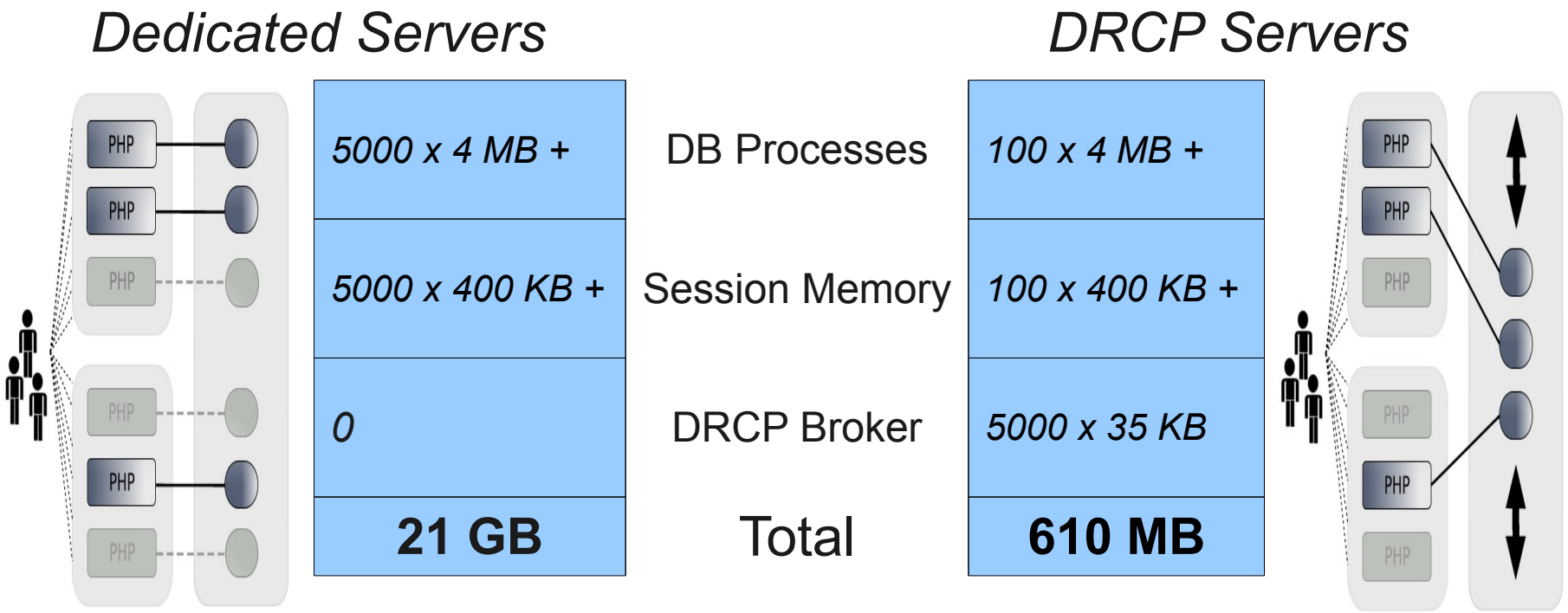


Database Processes Handle User Workload



Scaling up with DRCP

5000 users; DRCP pool size of 100



Configuring and Starting the Pool

- Configure the pool (optional)

```
SQL> execute dbms_connection_pool.configure_pool(
                              minsize  => 10,
                              maxsize  => 100,
                              incrsize => 2);
```

- Other options include

```
inactivity_timeout
max_think_time
```

- Start the Pool:

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

Applications Can Choose When to Use DRCP

- Developers/Administrators choose connection mode
- Pool is shared across all applications and web servers
- Supported in PHP, Python and Perl
  - PHP OCI8

```
ini_set('oci8.connection_class', 'MYCLASS');
$c = oci_pconnect('hr', 'welcome', 'localhost/orcl:pooled');
```

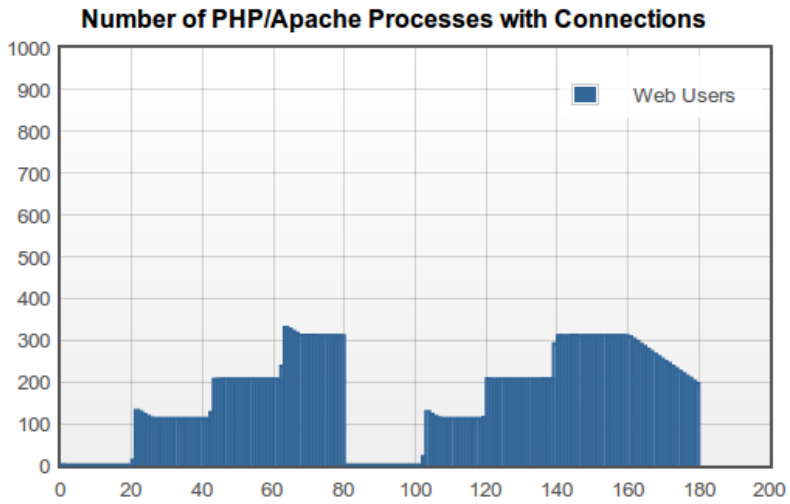
  - Python cx_Oracle

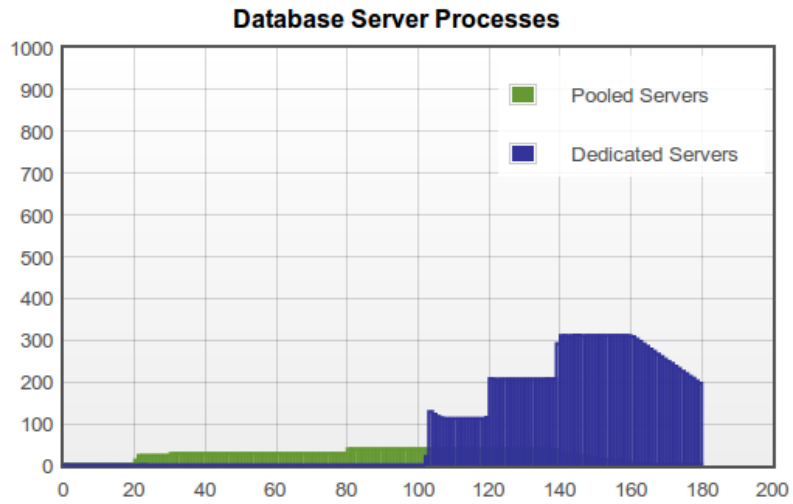
```
con = cx_Oracle.connect('hr', 'welcome', 'localhost/orcl:pooled',
                        cclass = "MYCLASS", purity = cx_Oracle.ATTR_PURITY_NEW)
```

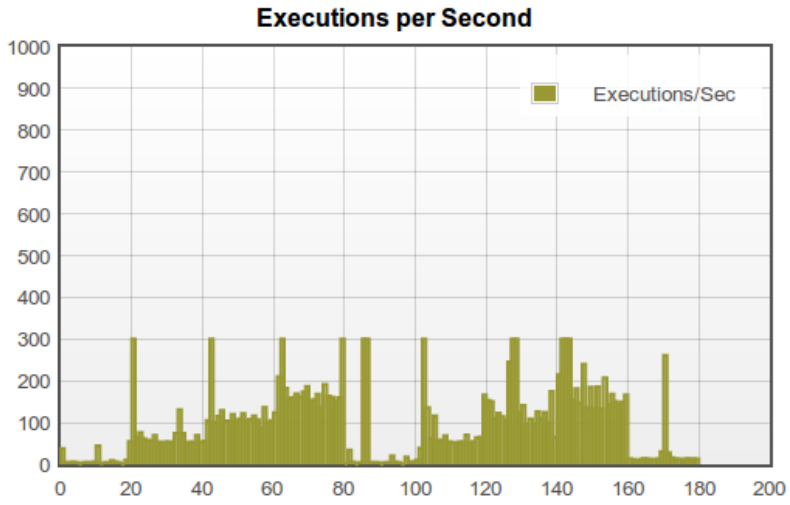
  - Perl DBD::Oracle

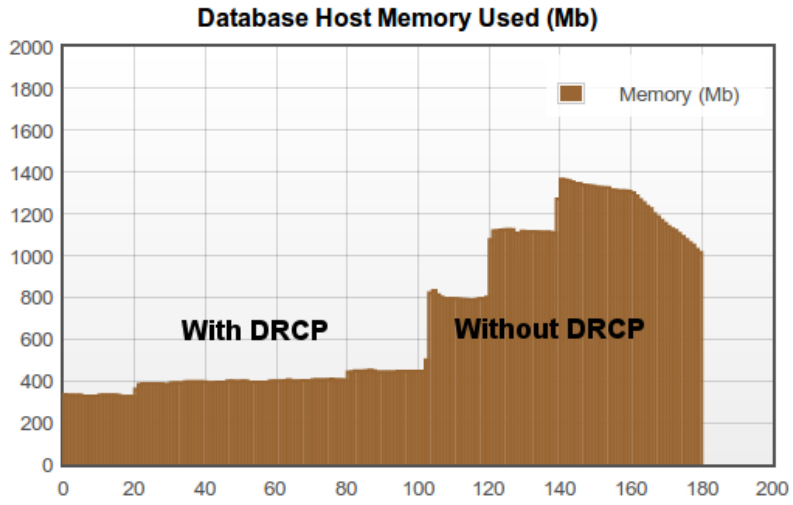
```
my $dbh = DBI->connect('dbi:Oracle:localhost/orcl:pooled',
                       'hr','welcome', {ora_drcp => 1, ora_drcp_class => "MYCLASS"})
```

DRCP Performance





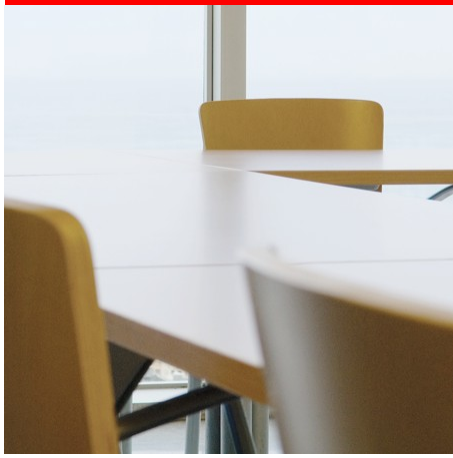




DRCP Recommendations.

- Read the PHP DRCP whitepaper
- Make sure `oci8.connection_class` is set
- Have > 1 Broker, but only a few
- Close connections when doing non-DB processing
- Explicitly control commits and rollbacks
  - Avoid unexpectedly open transactions when an `oci_close()` or end-of-scope occurs
  - Scripts coded like this can use `oci_close()` to take full advantage of DRCP but still be portable to older versions of the OCI8 extension
- Monitor `V$CPOOL_STATS` view to determine best pool size
- Don't use for long batch processes

Metadata for Authentication, Tracing and Diagnostics



The Problem: Database Tracing in a Typical PHP Application

- All web users run the same code:

```
$c = oci_connect('phpuser', 'welcome', 'localhost/orcl');
$s = oci_parse($c, 'select * from mytab');
oci_execute($s);
oci_fetch_all($s, $res);
```

- All database traces and logs show aggregated or non-identifying results
  - e.g. the Audit trail:

```
SQL> select username, extended_timestamp
     from dba_audit_trail;
PHPUSER 07-MAY-11 07.13.15.821472 PM -07:00
. . .
```

Client Identifiers for Auditing, Monitoring & VPD

- `oci_set_client_identifier($c, 'Chris');`
  - i.e. name of end web-user, not the table owner
- Executed SQL in DB views, Enterprise Manager and trace files are tagged with '**`Chris`**'
- Useful for web applications that
  - always connects with the one DB username e.g. '`phpuser`'
  - implement their own authentication
- Client IDs identify the actual web user for:
  - **Auditing**
  - **Monitoring**
  - **Virtual Private Database**

Using Client Identifiers for Auditing and VPD Demo

Auditing with Client Identifiers

- Run application

```
<?php
session_start();
$enduser = $_SESSION['username'];  // e.g. 'Chris'
$c = oci_connect('phpuser', 'welcome', 'localhost/orcl');
oci_set_client_identifier($c, $enduser);
$s = oci_parse($c, 'select * from mytab');
oci_execute($s);
oci_fetch_all($s, $res);
```

- View audit trail:

```
SQL> select client_id, username, extended_timestamp
     from dba_audit_trail;
Chris  PHPUSER 07-MAY-11 07.13.15.821472 PM -07:00
. . .
```

Virtual Private Databases

- Allow data to be protected in the DB by specifying arbitrary access rules in the DB
- In PHP, use

```
oci_set_client_identifier($c, 'Chris');
```

- In DB specify access rules based on the client identifier

VPD Policy function

```
create or replace function mypolicyfunc
            (schema in varchar2, tab in varchar2) return varchar2 as
   w varchar2(400);  // implicit WHERE clause
begin
   w := 'sys_context(''userenv'', ''client_identifier'') = ''Chris''';
   return w;
end;
/
execute dbms_rls.add_policy (
   object_schema   => 'HR', object_name => 'MYTABLENAME',
   policy_name     => 'Mypolicyname',  function_schema => 'HR',
   policy_function => 'MYPOLICYFUNC', policy_type => DBMS_RLS.STATIC);
```

VPD In Action

```
$c = oci_connect('hr', 'welcome', 'localhost/orcl');
oci_set_client_identifier($c, $cid);  // Chris or Alison
$s = oci_parse($c, "SELECT * FROM MYTABLENAME ORDER BY ID");
oci_execute($s);
oci_fetch_all($s, $res);
```

Inventory for Chris

ID	CATEGORY	NAME
1	electrical	lamp
2	electrical	wire
3	electrical	switch
4	plumbing	pipe
5	plumbing	sink
6	plumbing	toilet

Inventory for Alison

ID	CATEGORY	NAME

No rows returned

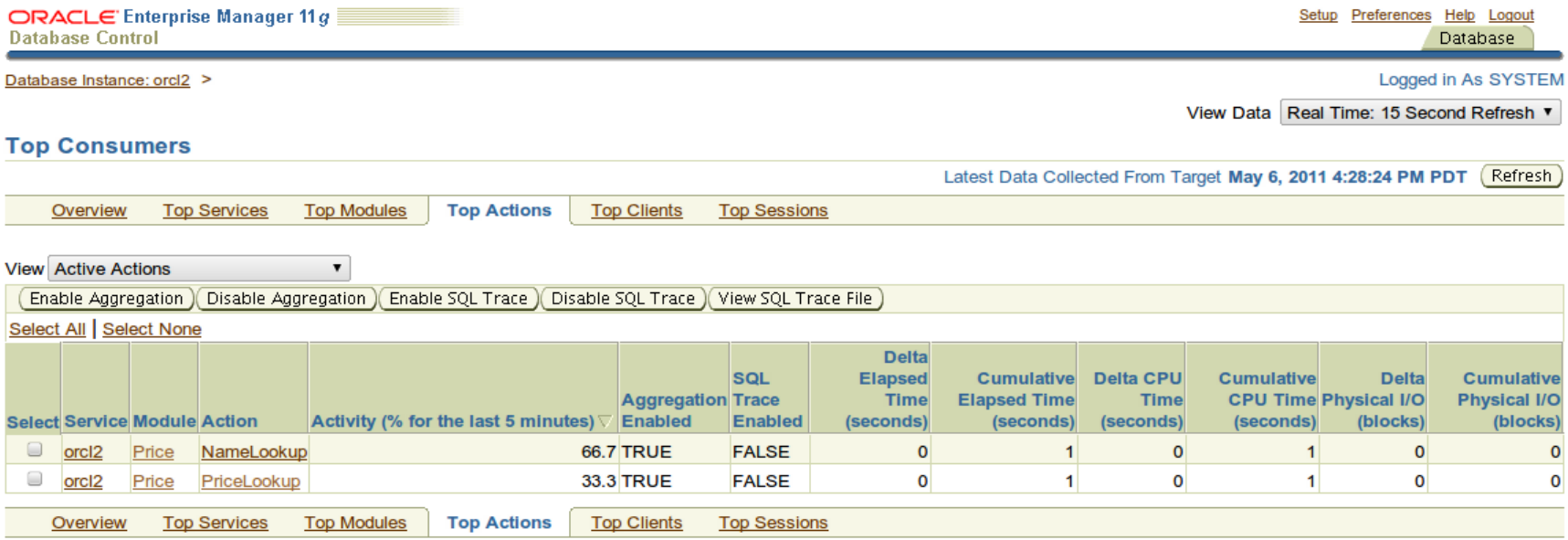
Metadata Values for Tracing & Diagnostics

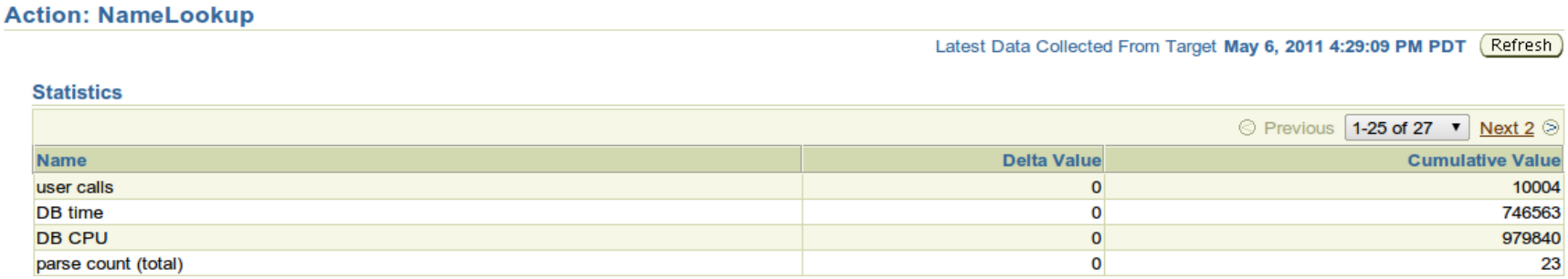
- Scripts can also set three levels of metadata identifiers
- Used by DBAs to locate PHP code executing SQL

```
$c = oci_connect('phpuser', 'welcome', 'localhost/orcl');
oci_set_client_identifier($c, 'Chris');
oci_set_client_info($c, 'My Application Version 2');
oci_set_module_name($c, 'Price');
oci_set_action($c, 'NameLookup');
$s = oci_parse($c, 'select * from mytab');
oci_execute($s);
oci_fetch_all($s, $res);
```

Monitoring Demo

Monitoring: V$ Views or Enterprise Manager





Tips on Using the oci_set_* Attribute Functions

- Exist in PHP 5.3 / OCI8 1.4
- Values are piggy-backed on next round-trip
- Before OCI8 1.4 use `DBMS_SESSION` or `DBMS_APPLICATION_INFO` but needs a round-trip:

```
$s = oci_parse($c,
 "begin
     DBMS_APPLICATION_INFO.SET_ACTION('NameLookup');
  end");
oci_execute($s);
```

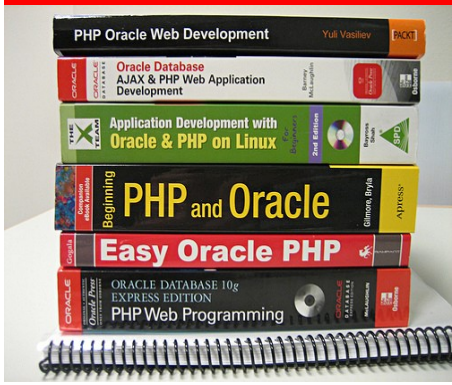
Summary

- IDE: NetBeans
- Databases
  - Berkeley DB
  - TimesTen In-Memory Cache
  - MySQL
  - Oracle Database
- Mid-tier: Tuxedo Application Server
- Operating System: Oracle Linux

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.

Wrap Up



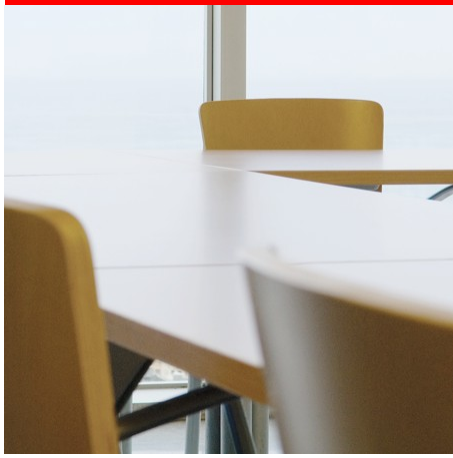
http://joind.in/3396

http://blogs.oracle.com/opal
http://twitter.com/ghrd
christopher.jones@oracle.com

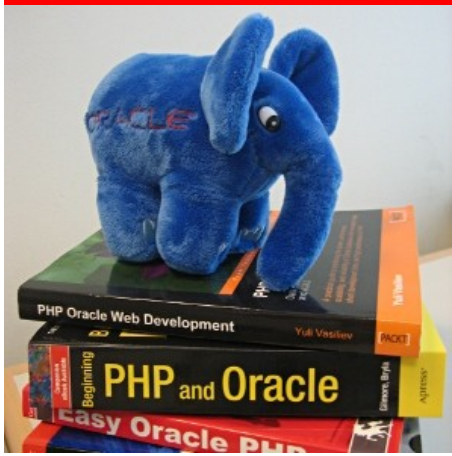
http://otn.oracle.com/php
PHP Developer Center on Oracle Technology Network

ORACLE®

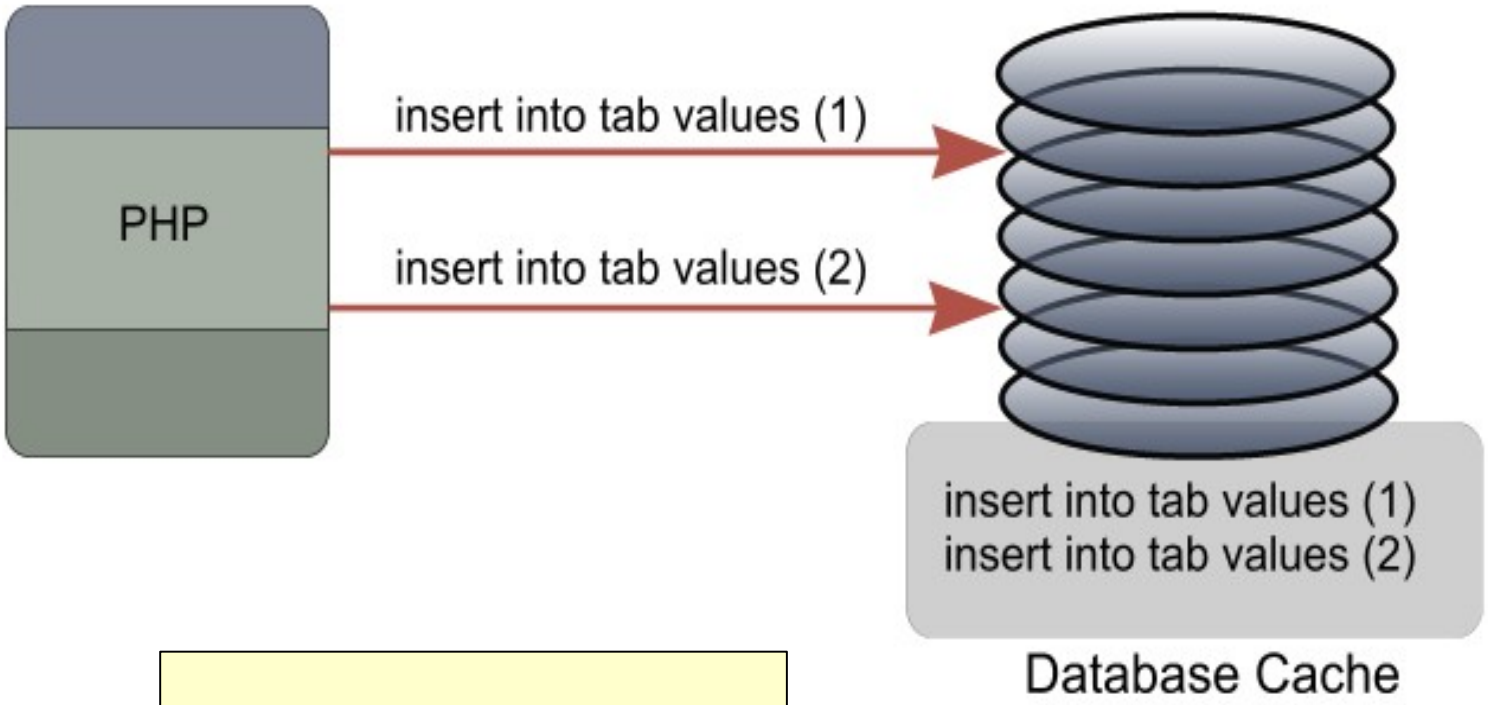
Extra Slides



Binding with SQL Statements

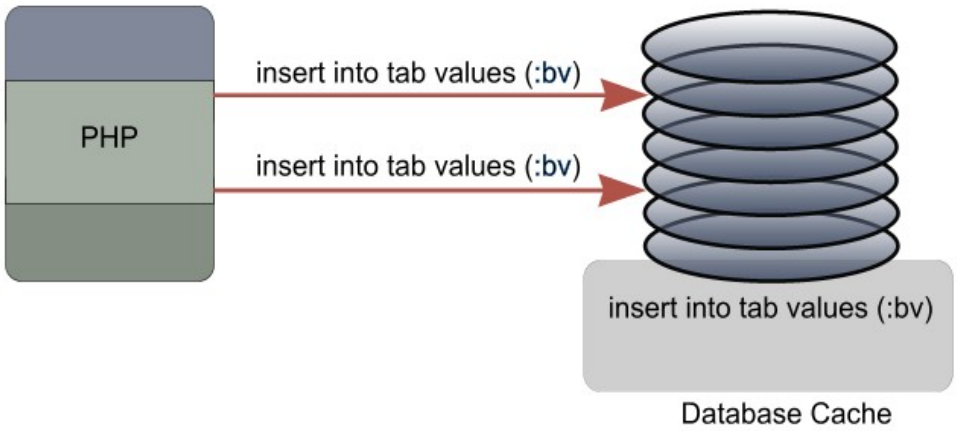


Not Binding Gives Poor Cache Use



Poor use of cache

Binding Scalars




```
$c = oci_connect('hr', 'hrpwd', 'localhost/orcl');
$s = oci_parse($c,'insert into tab values (:bv)');
$name = 'Jones';
oci_bind_by_name($s, ':bv', $name);
oci_execute($s);
```

Binding Benefits

From a query example by Tom Kyte:

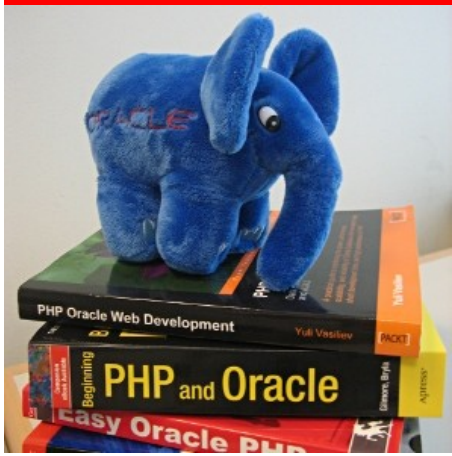
	Without	With
Parse count (hard)	5,000	1
Parse time elapsed	485	35
Latches	328,496	118,614

- Can change bind values and re-execute without re-parsing
- No SQL Injection worry
- Easier to write than adding quote escaping
- Overall system is more efficient
- PHP user elapsed time directly benefits

Binding Best Practices

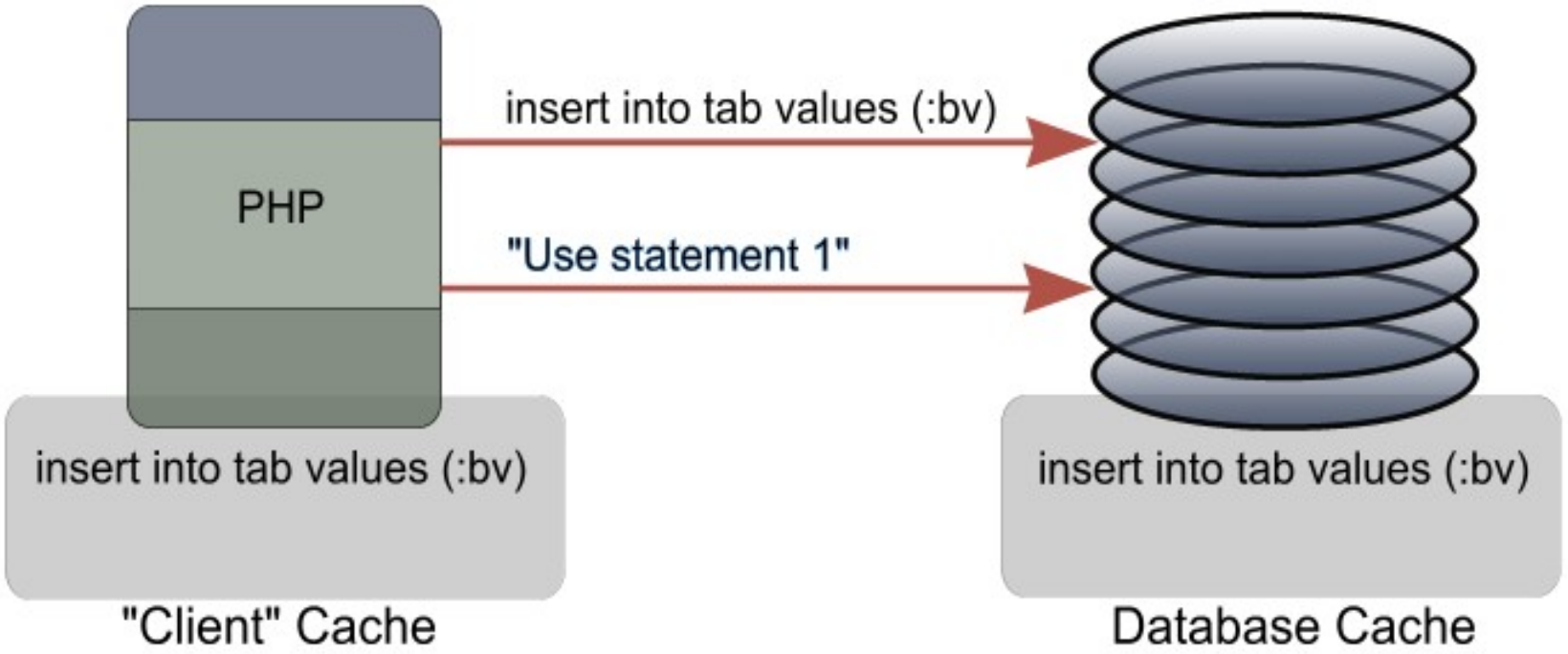
- Set length parameter to your upper data size for re-executed IN binds `oci_bind_by_name($s, ":b", $b, 40);`
- Don't bind constants
  - Let the optimizer see them
- Long running unique queries may not benefit
  - Parse time is a relatively small cost
- CURSOR_SHARING parameter
  - Set in "session" or database init.ora
  - Makes every statement appear to have bound data, but optimizer now doesn't see constants
  - For bind-unfriendly applications
- Oracle 11g has Adaptive Cursor Sharing
  - Can have multiple execution plans for same statement

Statement Caching



Client (aka PHP) Statement Caching

Oracle Client library cache of statement text & meta data



Less traffic and DB CPU

Statement Caching Best Practices

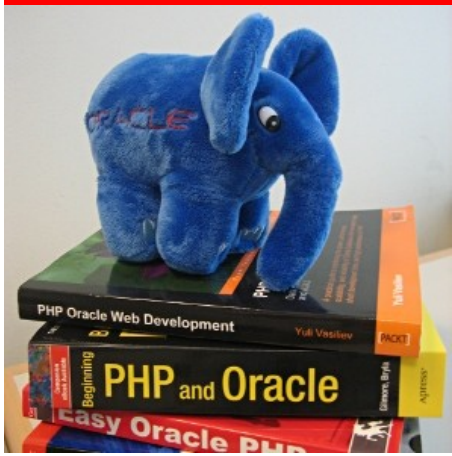
- Enabled by default in php.ini

```
oci8.statement_cache_size = 20
```

  Unit is number of statements

- Set it big enough for working set of statements

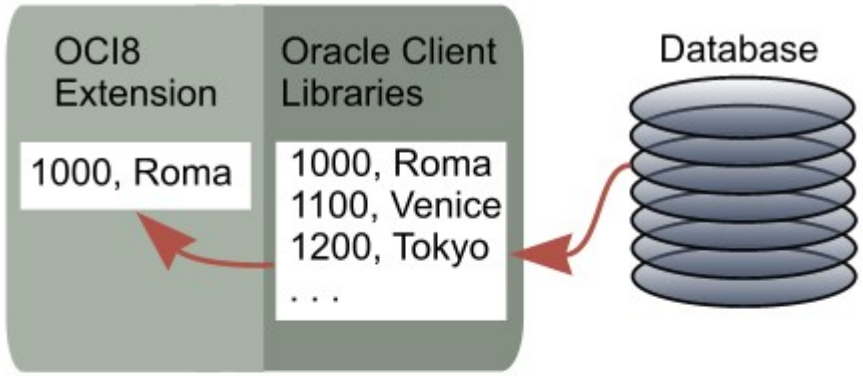
Row Prefetching



Prefetching Reduces Roundtrips

Temporary buffer cache for query duration

```
$r = oci_fetch_array(...);
var_dump($r);
// array('1000', 'Roma')
```

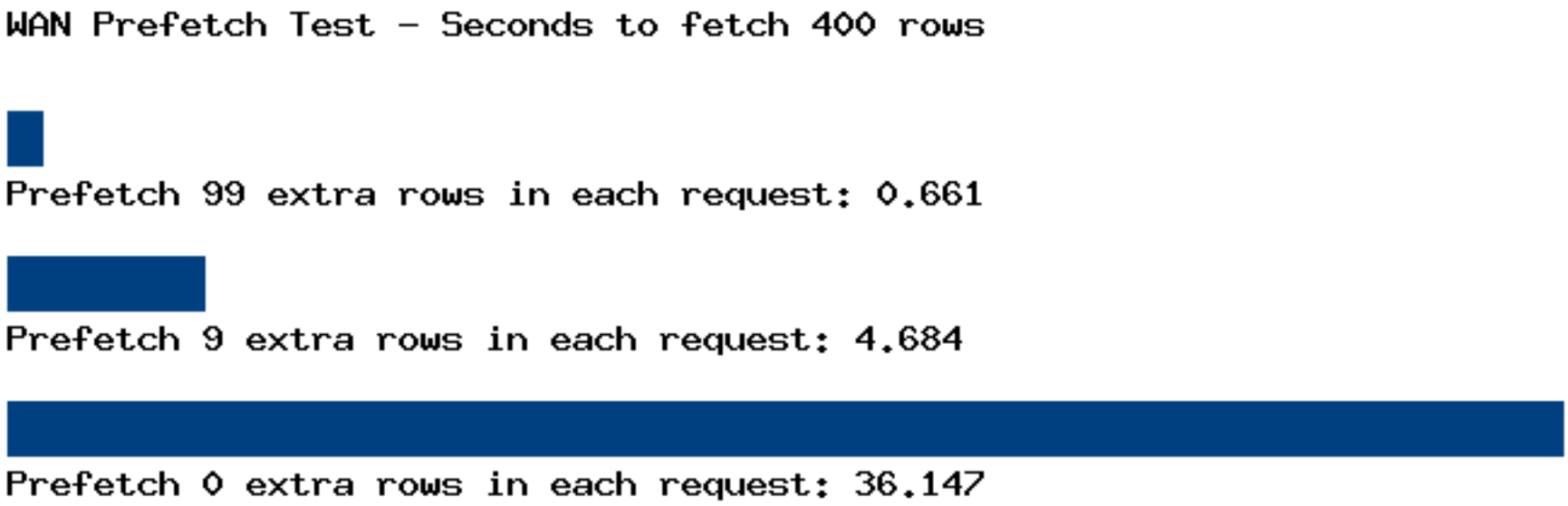



```
$r = oci_fetch_array(...);
var_dump($r);
// array('1100', 'Venice')
```

No DB access for next fetch

Reduces round trips

Prefetching Reduces Query Times



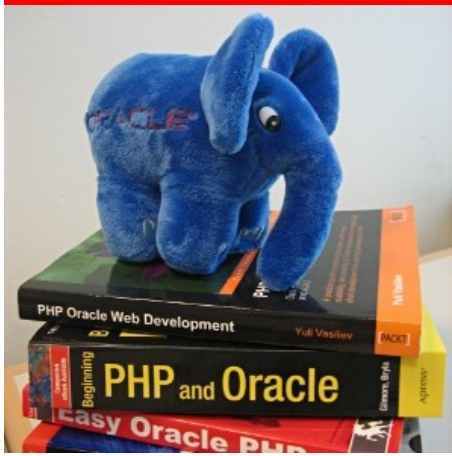
Your results may vary

Prefetching is Enabled by Default

- Enabled by default `oci8.default_prefetch = 100` rows
  - Was 10 rows in OCI8 1.2
- Tuning goal: Reduce round trips
  - but transfer reasonable chunks, not huge sets
- Can tune per statement:

```
$s = oci_parse($c, 'select city from locations');
oci_set_prefetch($s, 87);
oci_execute($s);
while (($row = oci_fetch_array($s, OCI_ASSOC)) != false)
  foreach ($row as $item)
    print $item;
```

Edition Based Redefinition



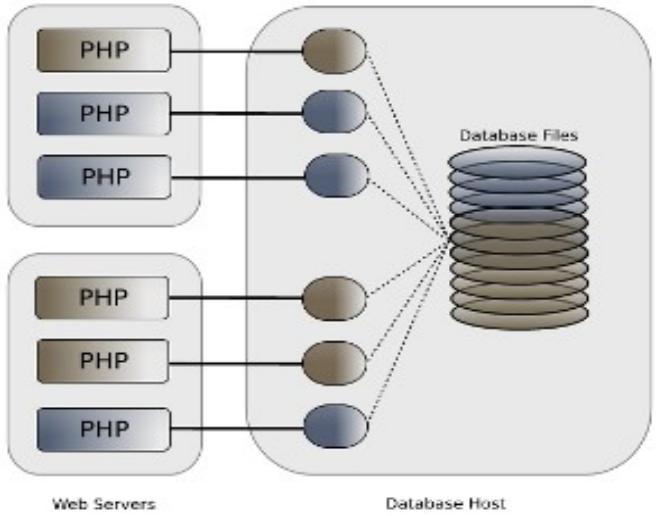
Edition Based Redefinition in Oracle 11gR2

- EBR allows multiple versions of DB objects to be used concurrently
- EBR allows application upgrade and testing

  while still in production use
- PHP OCI8 1.4 has `oci_set_client_edition()`

EBR and PHP Applications

- Use EBR in conjunction with PHP code changes
  - Load balancer/web server needs to call correct version of PHP scripts
- Use EBR for
  - A/B testing
  - Application migration and release testing



Application Version 1: DB Objects

Requirement: Show prices for house supplies

```
SQL> create table parts (name  varchar2(10),
                         price number);

SQL> insert into parts values ('lamp', 40);
...

SQL> create or replace function discount(name in varchar2)
return number as
begin
  return 4;
end;
```

Application Version 1: PHP Code

```
<?php // edition1.php

oci_set_edition('ora$base');  // OCI8 1.4
$c = oci_connect('cj','welcome','localhost/orcl');

$s = oci_parse($c,
      "select name, price, discount(id) as d from parts");
oci_execute($s);
$r = oci_fetch_array($s);
echo $r['NAME'] . ' costs $' . $r['PRICE'] - $r['D'];
?>

$ php edition1.php
lamp costs $36
```

EBR Recap: Application Version 1

- One **`edition1.php`** script
- One **`discount`** PL/SQL function
- One **`parts`** table
- One set of users accessing the application

Application Version 2: DB Schema

New Requirement: Alter the discount rate calculation

```
SQL> create edition e2;
SQL> alter session set edition = e2;

SQL> create or replace function discount(name in varchar2)
return number as
begin
  return 8;  // was 4 in previous edition
end;
```

Application Version 2: PHP Code

```
<?php // edition2.php

oci_set_edition('e2');  // OCI8 1.4
$c = oci_connect('cj','welcome','localhost/orcl');

$s = oci_parse($c,
   “select name, price, discount(id) as d from parts”);
oci_execute($s);
$r = oci_fetch_array($s);
echo $r['NAME'] . ' costs $' . $r['PRICE'] - $r['D'];
?>

$ php edition2.php
lamp costs $32
```

Summary: Edition Based Redefinition

- Two scripts: **`edition1.php`** and **`edition2.php`**
- Two **`discount`** stored functions in same schema
- One **`parts`** table
- Two sets of web users running different versions **concurrently**

```
$ php edition1.php
lamp costs $36
$ php edition2.php
lamp costs $32
```

- When migration completed, use `DROP EDITION`
- Use `oci_set_edition()` not `ALTER SESSION` in PHP

DBMS_XA: Transactions Across Requests

- Useful for
  - Stateful web applications
  - Migrating two-tier applications to the web
- HTTP Request #1:

```
rc := DBMS_XA.XA_START(DBMS_XA_XID(123), DBMS_XA.TMNOFLAGS);
UPDATE employees SET salary = salary * 1.1 WHERE id = 1;
rc := DBMS_XA.XA_END(DBMS_XA_XID(123), DBMS_XA.TMSUSPEND);
```

- HTTP Request #2:

```
rc := DBMS_XA.XA_START(DBMS_XA_XID(123), DBMS_XA.TMRESUME);
UPDATE employees SET salary = salary * 3 WHERE id = 2;
rc := DBMS_XA.XA_END(DBMS_XA_XID(123), DBMS_XA.TMSUSPEND);
```

- HTTP Request #3:

```
rc := DBMS_XA.XA_COMMIT(DBMS_XA_XID(123), TRUE);
```