

# PHP e Oracle

## Dicas de Performance

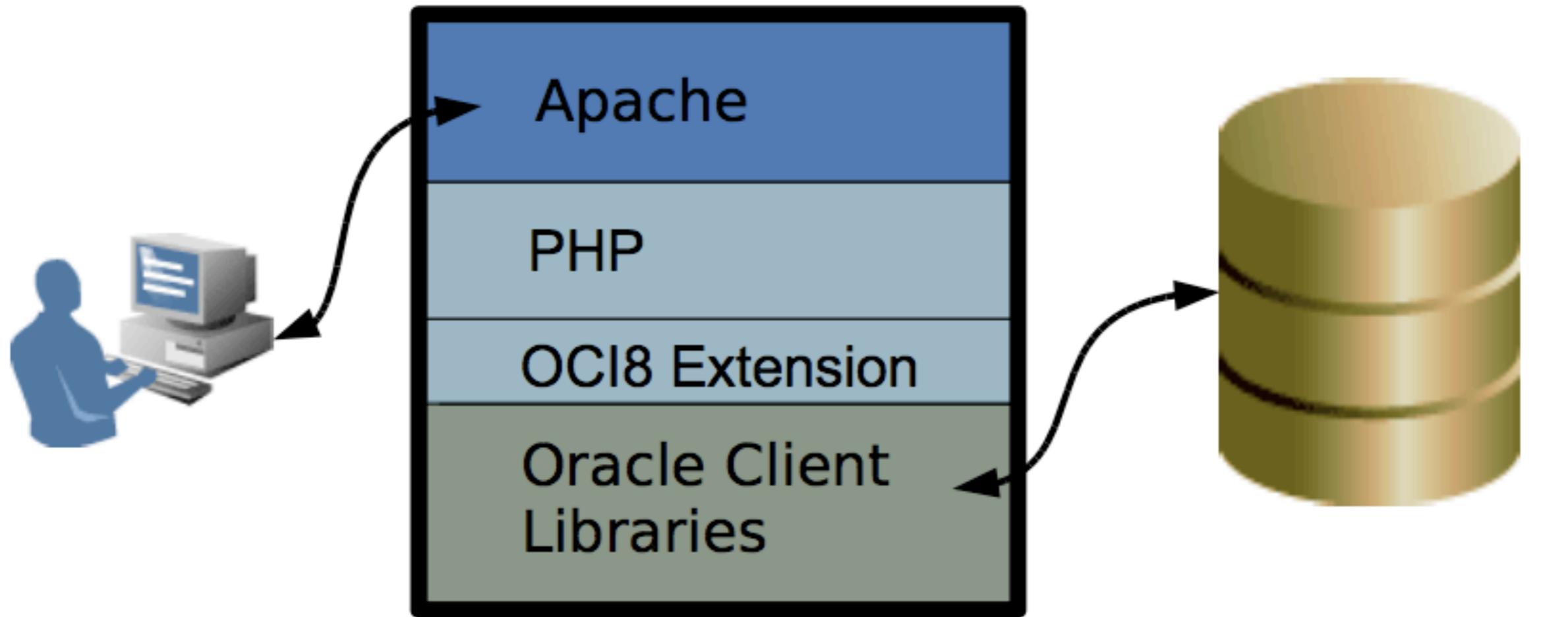
Elton Luís Minetto  
PHPSC Conf

# Quem?

- ✦ Graduado e pós-graduado em Ciência da Computação. Cursando MBA em Gerenciamento de Projetos
- ✦ Trabalha com PHP desde 2000
- ✦ Trabalha com Oracle desde 1999
- ✦ Autor do livro Frameworks para Desenvolvimento em PHP - Editora Novatec
- ✦ Membro do PHPSC

# OCI8

- ✦ Extensão desenvolvida pela Oracle e Zend (reescrita em 2005)
- ✦ Rápida e estável
- ✦ Parâmetros específicos no php.ini para o Oracle:
  - ✦ Gerenciamento de conexões persistentes
  - ✦ “Prefetching” de linhas
  - ✦ Cache no lado do cliente



Web User

Mid Tier  
9iR2, 10g, 11g  
Any platform

Oracle Database  
8i, 9i, 10g, 11g  
Any platform

# Exemplo

```
<?php
$u = "elton";
$p = "elton";
$conn = oci_connect($u,$p,"192.168.200.200:1521/db11g");
for($i=0;$i<10;$i++) {
    $stmt = oci_parse($conn, "insert into pessoa
values($i,'nome $i',sysdate)");
    oci_execute($stmt,OCI_DEFAULT);
}
oci_commit($conn);
oci_close($conn);
?>
```

# Alternativas ao OCI8

- ✦ ADODB e PEAR MBD2
  - ✦ Bibliotecas de abstração de banco de dados
  - ✦ Escritas em PHP: usam chamadas ao OCI8
- ✦ PDO (PHP Data Objects)
  - ✦ Abstração de acesso a base de dados
  - ✦ Extensão separada escrita em C, não relacionada ao OCI8
  - ✦ Menos estável que o OCI8

# Grandes erros de Performance em DB

- ✦ Gerenciamento ineficiente das conexões
- ✦ Mal uso de cursores
- ✦ SQLs mal construídos
- ✦ “Full table scans”

# Conexões

- ✦ `oci_connect()`
- ✦ `oci_new_connect()`
- ✦ `oci_pconnect()`

# oci\_connect()

- ✦ A segunda chamada em um script retorna a mesma conexão
- ✦ Grande uso de CPU. Criação da conexão é lenta
- ✦ Pouco escalável
- ✦ A conexão é finalizada no fim do script ou explicitamente com o `oci_close()`

# oci\_new\_connect()

- Cada chamada retorna uma nova conexão com o banco. Recomendada para operações independentes
- A conexão é finalizada no fim do script ou explicitamente com o `oci_close()`

# oci\_pconnect()

- Conexões persistentes. Não são fechadas no fim do script
- Rápido para conexões sequenciais
- Ocupa recursos quando a aplicação está em espera
- Configurações no php.ini:
  - `oci8.max_persistent`
  - `oci8.persistent_timeout`
  - `oci8.ping_interval`

# Parâmetros

- ✦ `oci8.max_persistent = -1`
  - ✦ Número de conexões persistentes em cache para cada processo do Apache.
  - ✦ -1 significa sem limites

# Parâmetros

- ✦ `oci8.persistent_timeout = -1`
  - ✦ Timeout para conexões não usadas. Tempo em segundos para que uma conexão parada seja descartada
  - ✦ -1 significa sem limite

# Parâmetros

- ✦ `oci8.ping_interval = 60`
  - ✦ Intervalo de tempo em segundos para que a conexão seja testada
  - ✦ Uma nova conexão é criada se nenhuma está disponível
  - ✦ Para escalabilidade é recomendado desabilitar com `-1`

# Fechando Conexões

- ✦ É recomendado usar o `oci_close()` para fechar conexões abertas com o `oci_connect()` e `oci_new_connect()`
- ✦ Principalmente em scripts demorados e com pouca interação com o banco

# “Prefetch” de linhas

- ❖ `oci8.default_prefetch = 10`
  - ❖ Número de linhas que o Oracle guarda em cache além das solicitadas
  - ❖ Aumenta a performance das consultas
  - ❖ O valor também pode ser mudado em tempo de execução:
    - ❖ `oci_set_prefetch($s, 100);`

# Bind de Variáveis

- ✦ Aumenta a performance porque o banco não precisa “compilar” as consultas todas as vezes
- ✦ Aplicações podem usar a mesma consulta com diferentes valores
- ✦ Diminui o risco de ataques de SQL Injection

# Bind - Exemplo

```
$s = oci_parse($c,"select nome from pessoas where id = :p_id");  
$tmp_id = 101;  
oci_bind_by_name($s, ":P_ID", $tmp_id);  
oci_execute($s);  
oci_fetch_all($s, $res);  
echo "Nome:". $res['NOME'][0]. "<br>\n";  
$tmp_id = 102;  
oci_execute($s); // Não precisa o parse de novo  
oci_fetch_all($s, $res);  
echo "Nome: " . $res['NOME'][0] . "<br>\n";
```

# OCI8 Statement Cache

- ✦ `oci8_statement_cache_size = 20`
  - ✦ Número de consultas que são armazenadas. Move a tarefa de armazenar as consultas para o PHP poupando o banco.

Novidades no Oracle 11g e  
avanços no PHP

OCI8 Beta - Outubro 2007

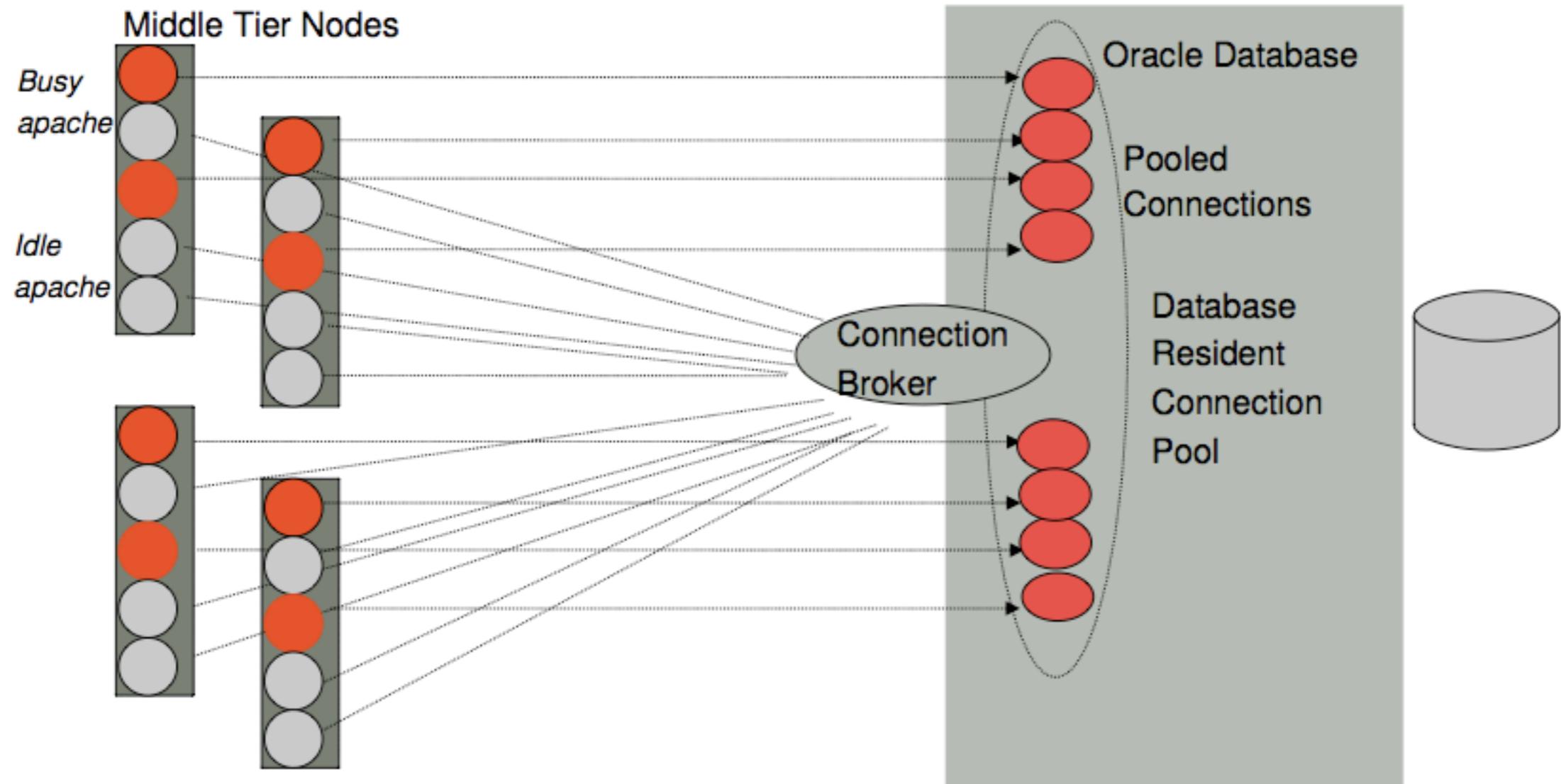
PHP 5.3 e PECL para 4.3.9+

# Conexões

- ✦ Conexões Não Persistentes
  - ✦ Grande tempo gasto para as conexões
  - ✦ Carga de CPU para cada processo connect/disconnect
- ✦ Conexões Persistentes
  - ✦ Muitas conexões abertas
  - ✦ Muito consumo de memória

# Pooling de Conexões

- ✦ Database Resident Connection Pool (DRCP)
  - ✦ Cria um pooling de conexões com o banco, compartilhando conexões entre processos do Apache



# 5000 clientes

	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>
<b>Total Memory</b>	<b>21 GB</b>	<b>2.3 GB</b>	<b>610 MB</b>

# Como funciona

- Pooling é opcionalmente ativado pelo DBA no Oracle: Min, Max, Timeout
- Cliente (PHP) altera a string de conexão
- Cliente conecta no DRCP
- Uma conexão é colocada em “lock” para o cliente
- Conexão é devolvida ao pool quando liberada pelo cliente

# Compartilhamento

- ✦ Apenas clientes conectados com usuário “elton” podem compartilhar outras conexões do usuário “elton” no pool
- ✦ Pode-se configurar uma `CONNECTION_CLASS` permitindo que somente clientes com a mesma classe compartilhem suas conexões.

# Ativar - Visão do DBA

- ✦ Ativar o pool de conexões
  - ✦ `dbms_connection_pool.start_pool();`
  - ✦ `dbms_connection_pool.stop_pool();`
- ✦ Configurar
  - ✦ `dbms_connection_pool.configure(min=>10, max=>300);`

# Ativar - Visão do PHP

- ✦ Compilar o PHP com a nova versão da extensão OCI8 que tem suporte a pooling (ou usar PHP 5.3+)
- ✦ Pedir para o DBA ativar no banco
- ✦ Alterar a string de conexão
- ✦ Colocar no php.ini:
  - ✦ `oci8.connection_class = "PHP"`

# O que muda?

- ✦ `oci_pconnect()`
  - ✦ Rapidamente faz a ligação com uma conexão existente no pool
- ✦ `oci_connect()/oci_new_connect()`
  - ✦ Rapidamente cria novas conexões
- ✦ `oci_close()`
  - ✦ Libera uma conexão de volta ao pool

# Testes de Performance

Apache ab

```
ab -c10 -t 60 http://localhost/php_oracle/
```

# Conexão Normal

```
<?php
$u = "elton";
$p = "elton";
$conn = oci_pconnect($u,$p,"192.168.200.200:1521/db11g");
for($i=0;$i<10;$i++) {
    $stid = oci_parse($conn, "insert into pessoa
values($i,'nome $i',sysdate)");
    oci_execute($stid,OCI_DEFAULT);
}
oci_commit($conn);
oci_close();
?>
```

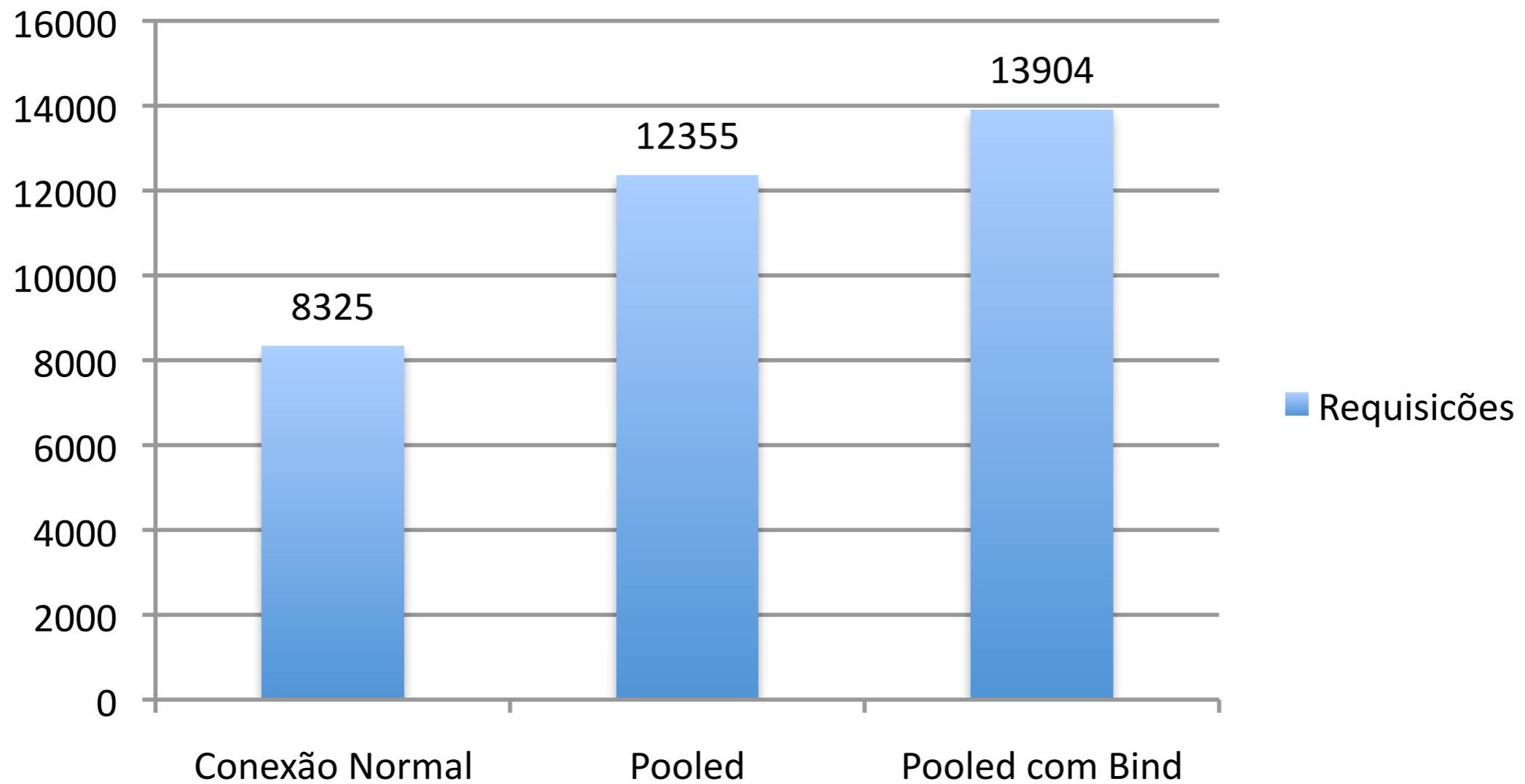
# Pooled

```
<?php
$u = "elton";
$p = "elton";
$conn = oci_pconnect($u,$p,"192.168.200.200:1521/
db11g:POOLED");
for($i=0;$i<10;$i++) {
    $stmt = oci_parse($conn, "insert into pessoa
values($i,'nome $i',sysdate)");
    oci_execute($stmt,OCI_DEFAULT);
}
oci_commit($conn);
oci_close();
?>
```

# Pooled com Bind

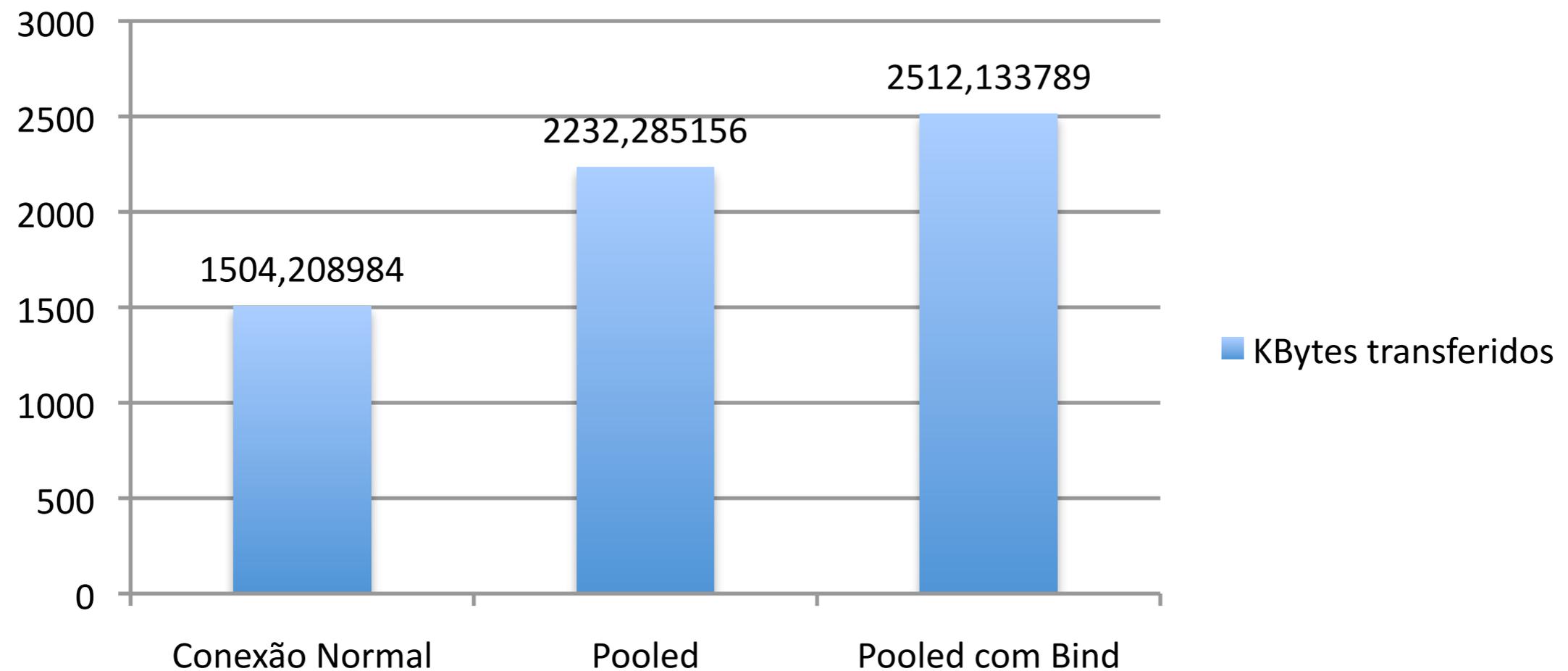
```
<?php
$u = "elton";
$p = "elton";
$conn = oci_pconnect($u,$p,"192.168.200.200:1521/
db11g:POOLED");
$stmt = oci_parse($conn, "insert into pessoa
values(:codpes, :nompes, sysdate)");
for($i=0;$i<10;$i++) {
    oci_bind_by_name($stmt,":CODPES",$i);
    $nompes = "nome $i";
    oci_bind_by_name($stmt,":NOMPES",$nompes);
    oci_execute($stmt,OCI_DEFAULT);
}
oci_commit($conn);
oci_close();
?>
```

# Requisições



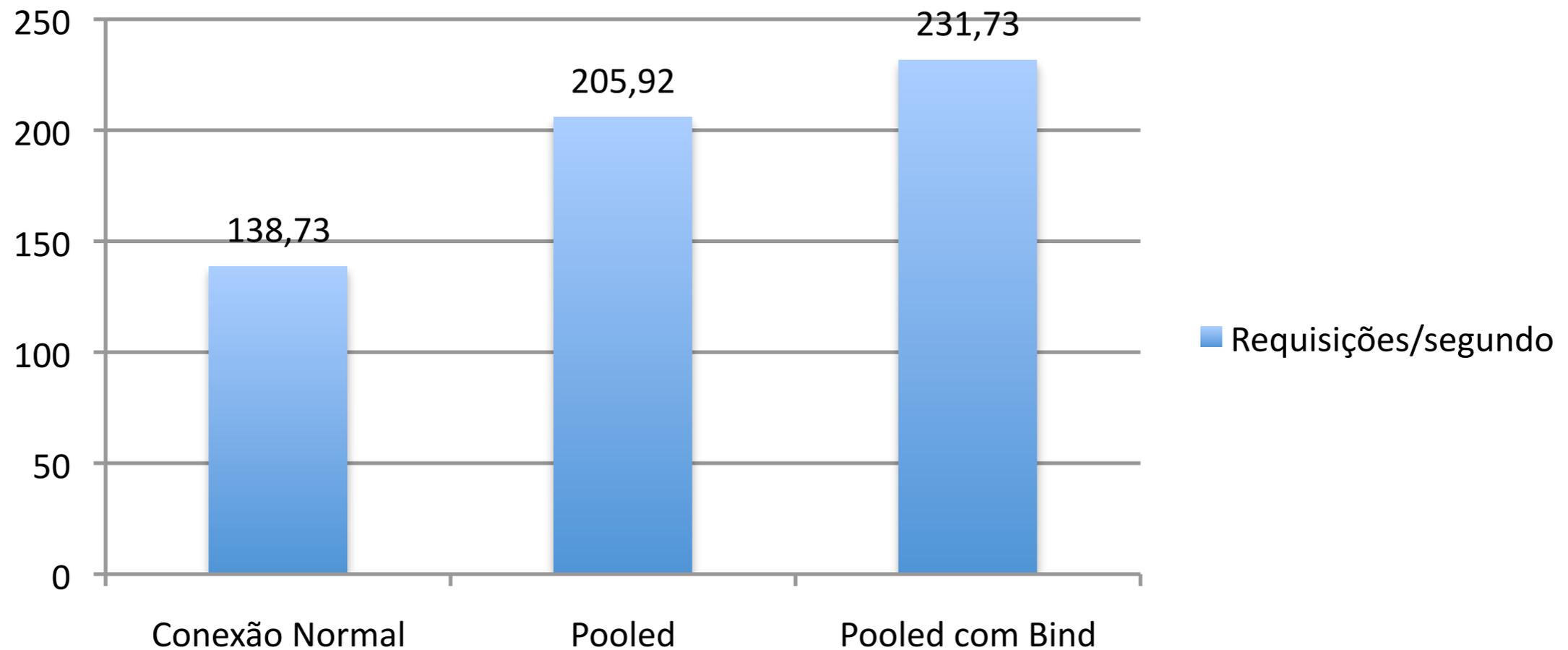
Maior é melhor. Mais requisições atendidas

## KBytes transferidos



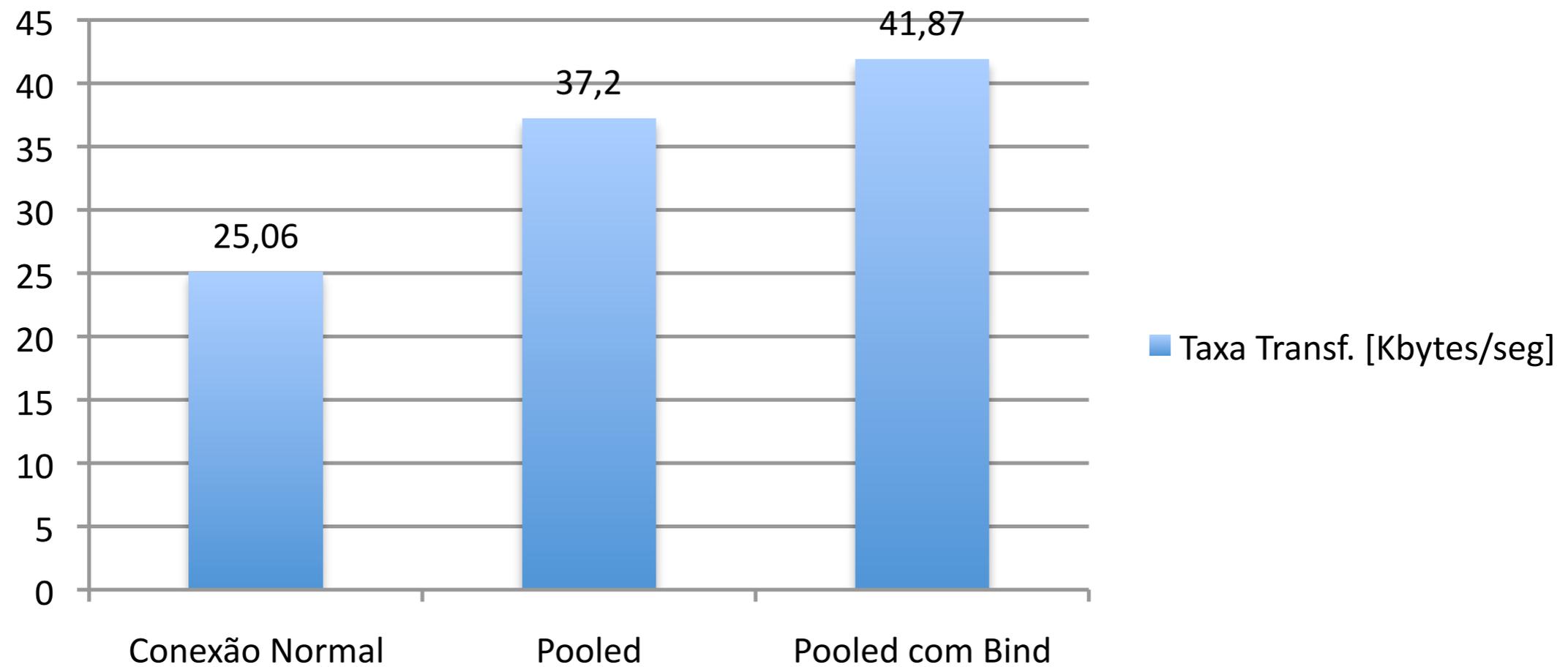
Maior é melhor. Mais KB transferidos

## Requisições/segundo



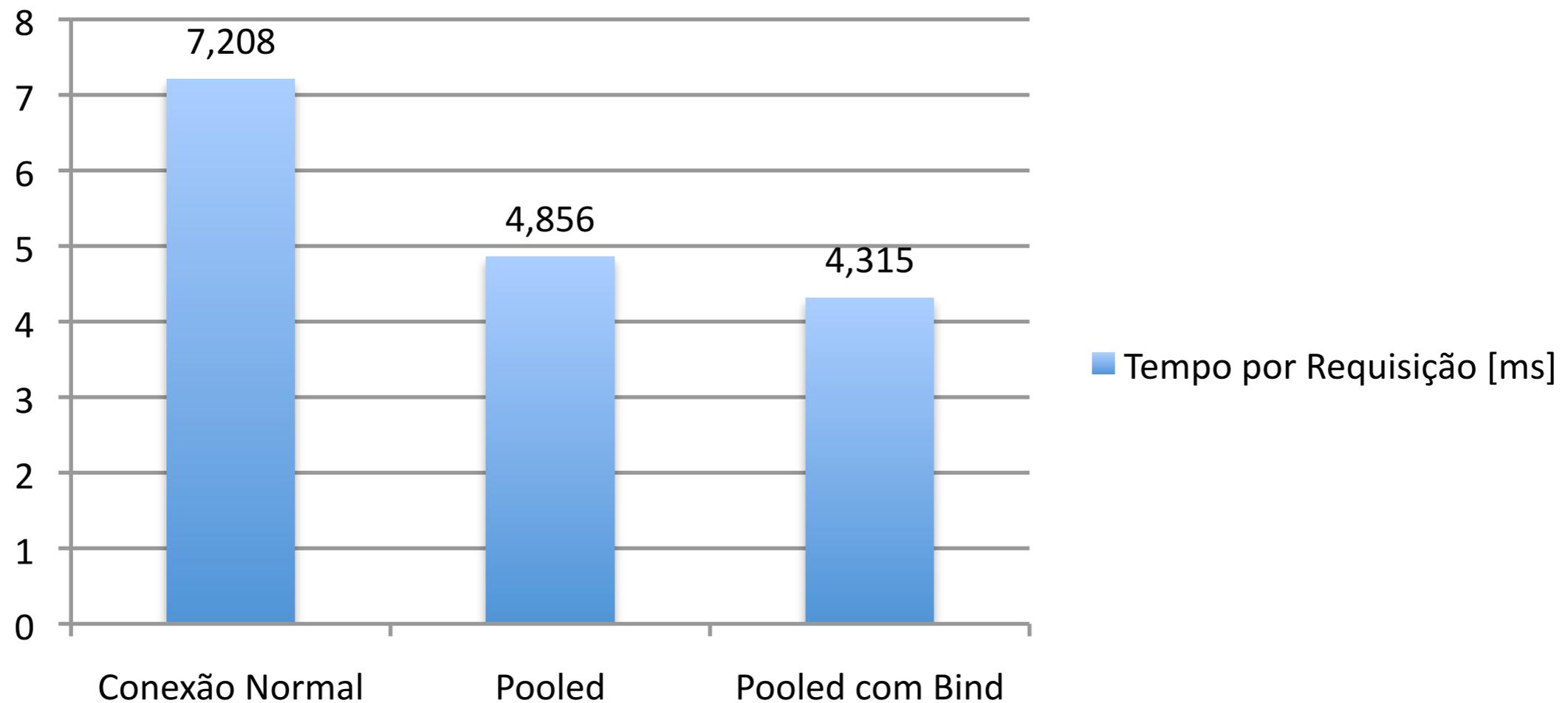
Maior é melhor. Mais requisições atendidas por segundo

## Taxa Transf. [Kbytes/seg]



Maior é melhor. Maior taxa de transferência

## Tempo por Requisição [ms]



Menor é melhor. Menos tempo para responder

# Observação

- ✦ Segundo Christopher Jones (Oracle):
  - ✦ “...when the DB machine is loaded that pooling has benefits, because you can do more on the machine. DRCP is for scalability and is not a pure performance feature.”

# Boas práticas

- ✦ Não segurar uma conexão do pool quando não é mais necessária
- ✦ Não esperar até o fim do script para liberar uma conexão
- ✦ Não intercalar trabalho com o banco com outros trabalhos dentro do script quando estiver com uma conexão em aberto

# Referências

- ✦ <http://blogs.oracle.com/opal>
- ✦ <http://www.oracle.com/technology/php>
- ✦ <http://otn.oracle.com>
- ✦ <http://www.php.net>

# Contato

Elton Luís Minetto

<http://www.eltonminetto.net>

[eminetto@gmail.com](mailto:eminetto@gmail.com)