

Yes, its the mysqlnd talk!





mysqlnd: Asynchronous Queries and more ...

Ulf Wendel

Senior Software Engineer

Sun Microsystems



The MySQL PHP “Connectors”

Andrey Hristov (Development - mysqlnd),
Johannes Schlüter (Development),
Ulf Wendel (QA) and Georg Richter (everything, somehow...)



How PHP connects to MySQL

PHP API for PHP applications

PHP

MySQL native driver for PHP / MySQL Client Library

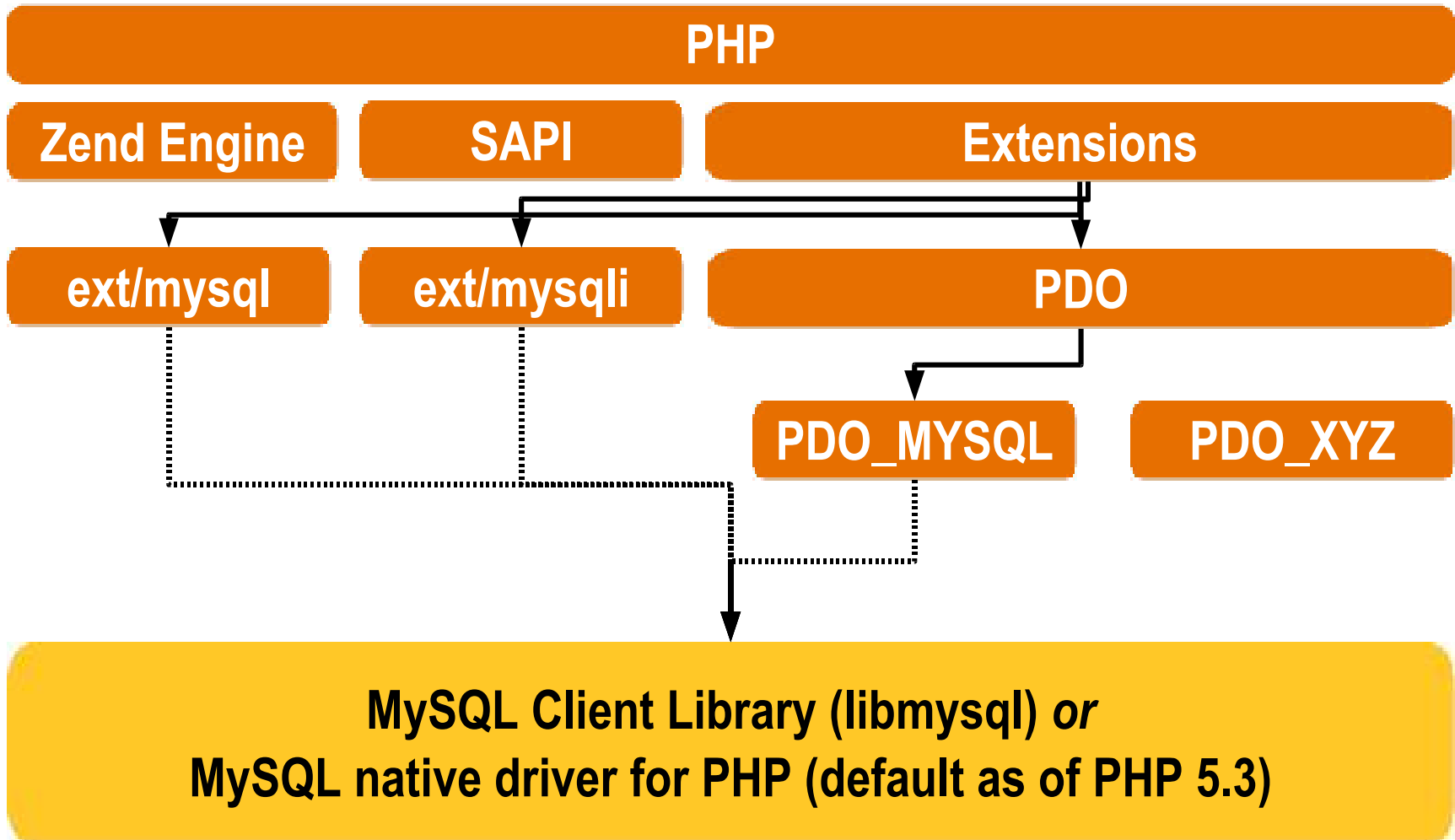
MySQL Server



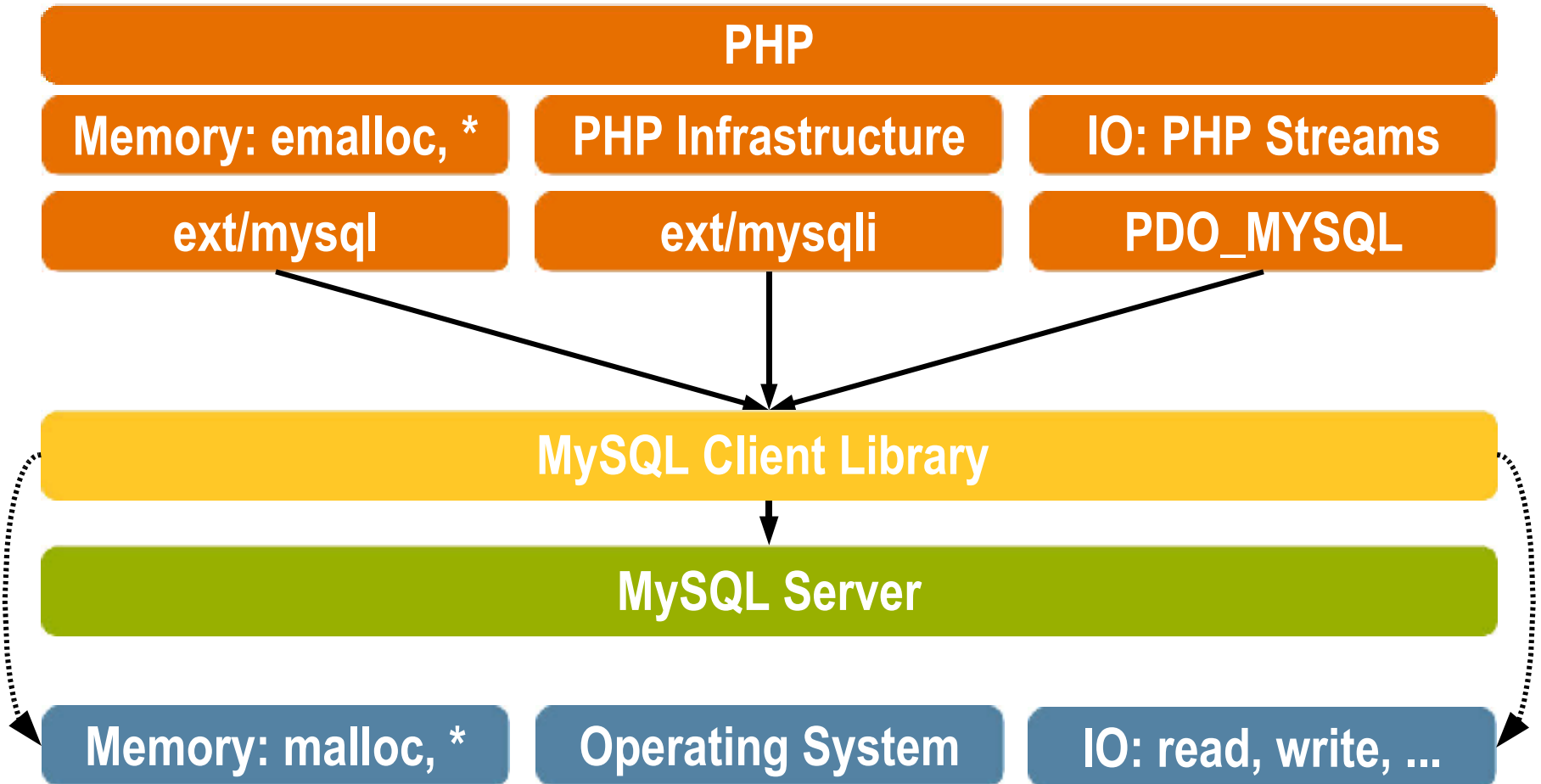
The “MySQL native driver for PHP”

- **Native?**
 - > Integrated tightly into PHP!
 - > NOT written in PHP, written in C
- **Driver?**
 - > A library that implements the communication protocol
 - > **NOT a new API for PHP users!**
- **For PHP?**
 - > Optimized for nothing but PHP!
 - > Easier to maintain: part of PHP, works with every MySQL
 - > Copyright PHP Group, 100% PHP license

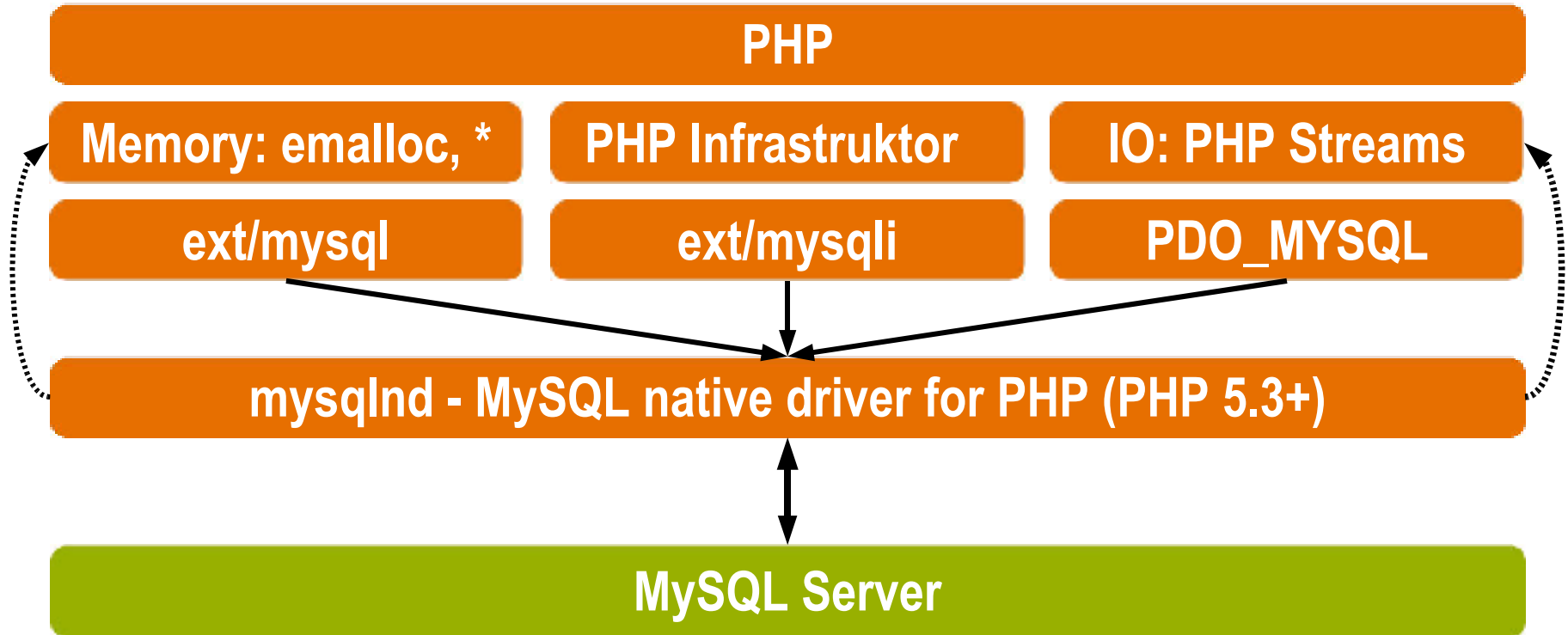
Inside PHP (on the C level!)



PHP and the MySQL Client Library



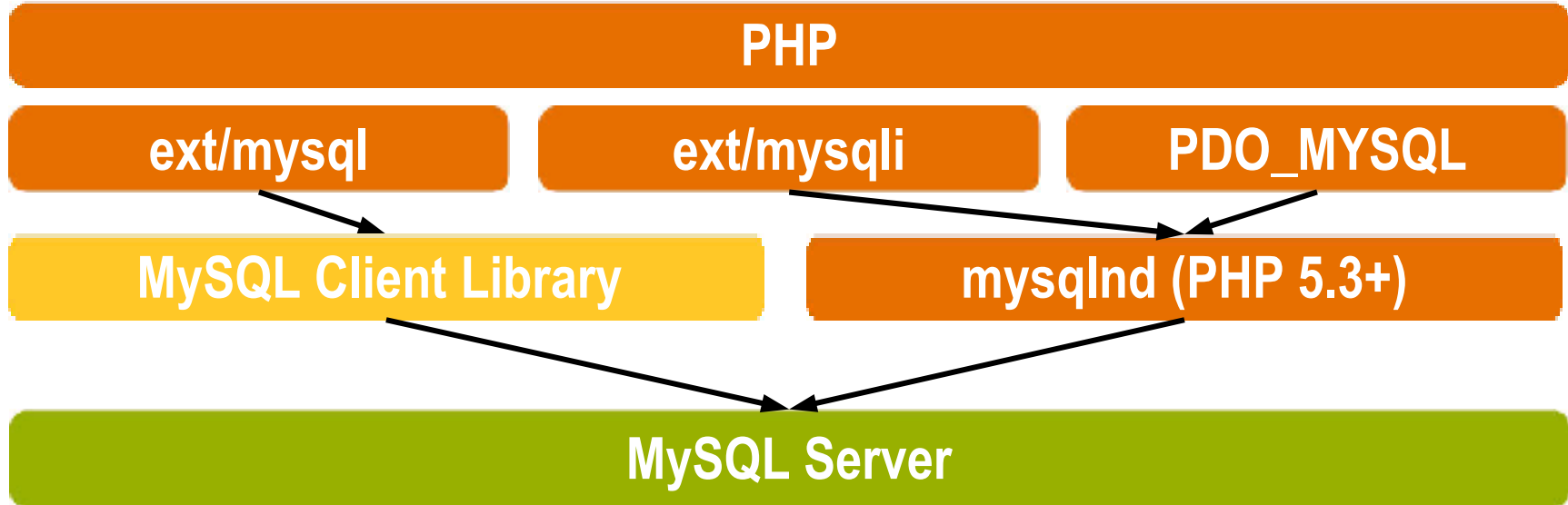
PHP and mysqlnd



Which API would you like?

| | ext/mysql | ext/mysql_i | PDO_MYSQL |
|-----------------------------|------------|-------------|------------|
| Maintained by MySQL | yes | yes | yes |
| Future additions from MySQL | no | yes | yes |
| Comes with PHP 4 | yes | no | no |
| Comes with PHP 5 | yes | yes | yes |
| Comes with PHP 6 | yes | yes | yes |
| Support of MySQL < 4.1 | yes | no | yes |
| Support of MySQL >= 4.1 | incomplete | yes | incomplete |
| MySQL Client Library | yes | yes | yes |
| MySQL native driver for PHP | yes | yes | yes |

Mixed Salad

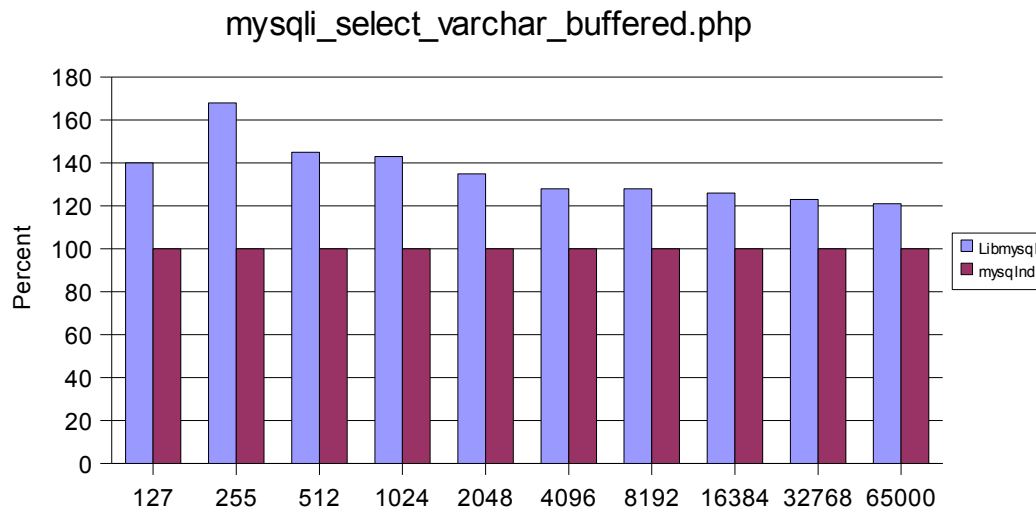


```

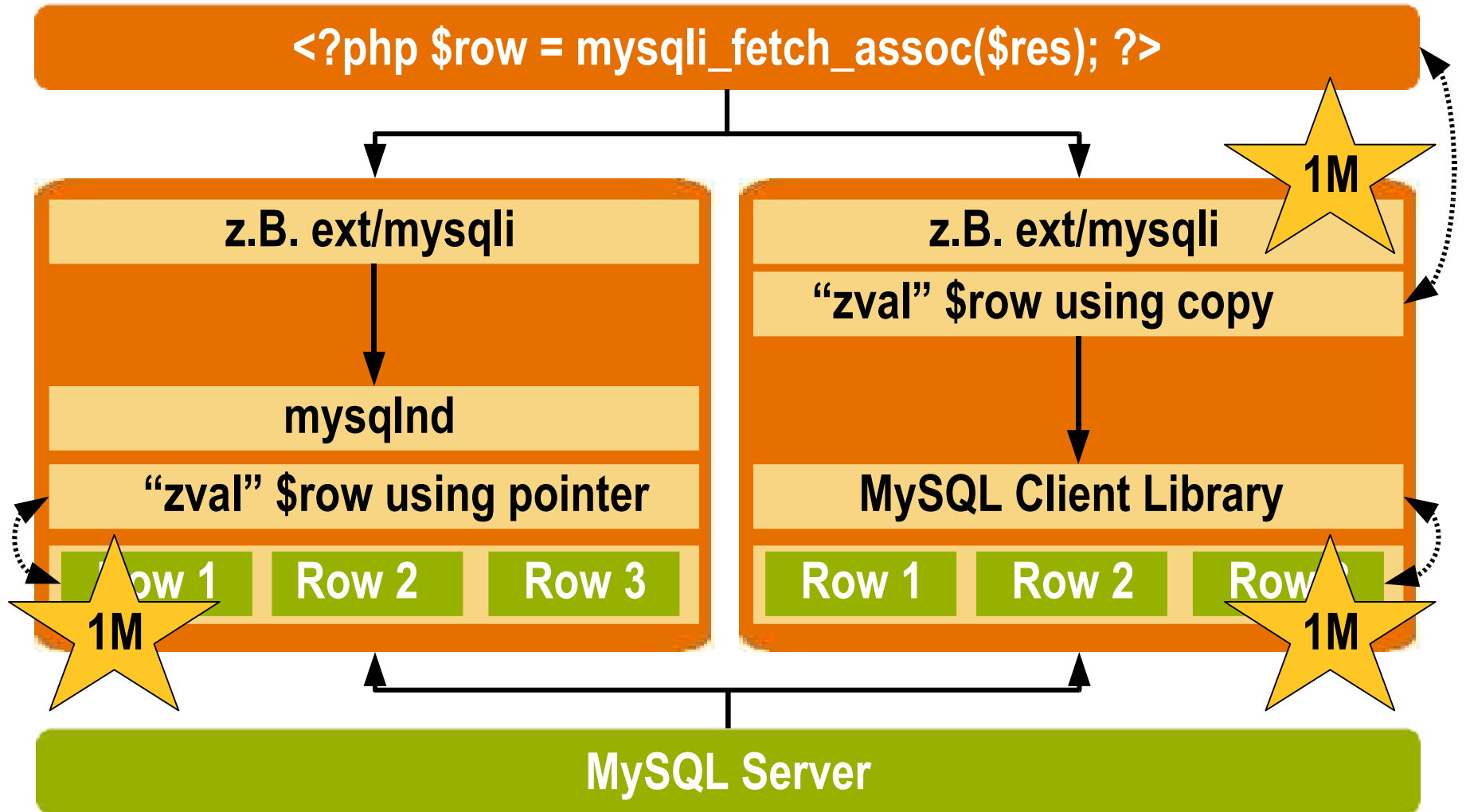
./configure --with-mysql=/path/to/mysql_config \
  --with-mysqli=mysqlnd \
  --with-pdo-mysql=mysqlnd
  
```

Advantage mysqlnd!

- 0% to 5% faster
- Microbenchmarks: -5% to +1200% faster
- 0% to 40% lower memory usage
- 120+ performance statistics – phpinfo()



Read-Only Variablen (Copy on Write)



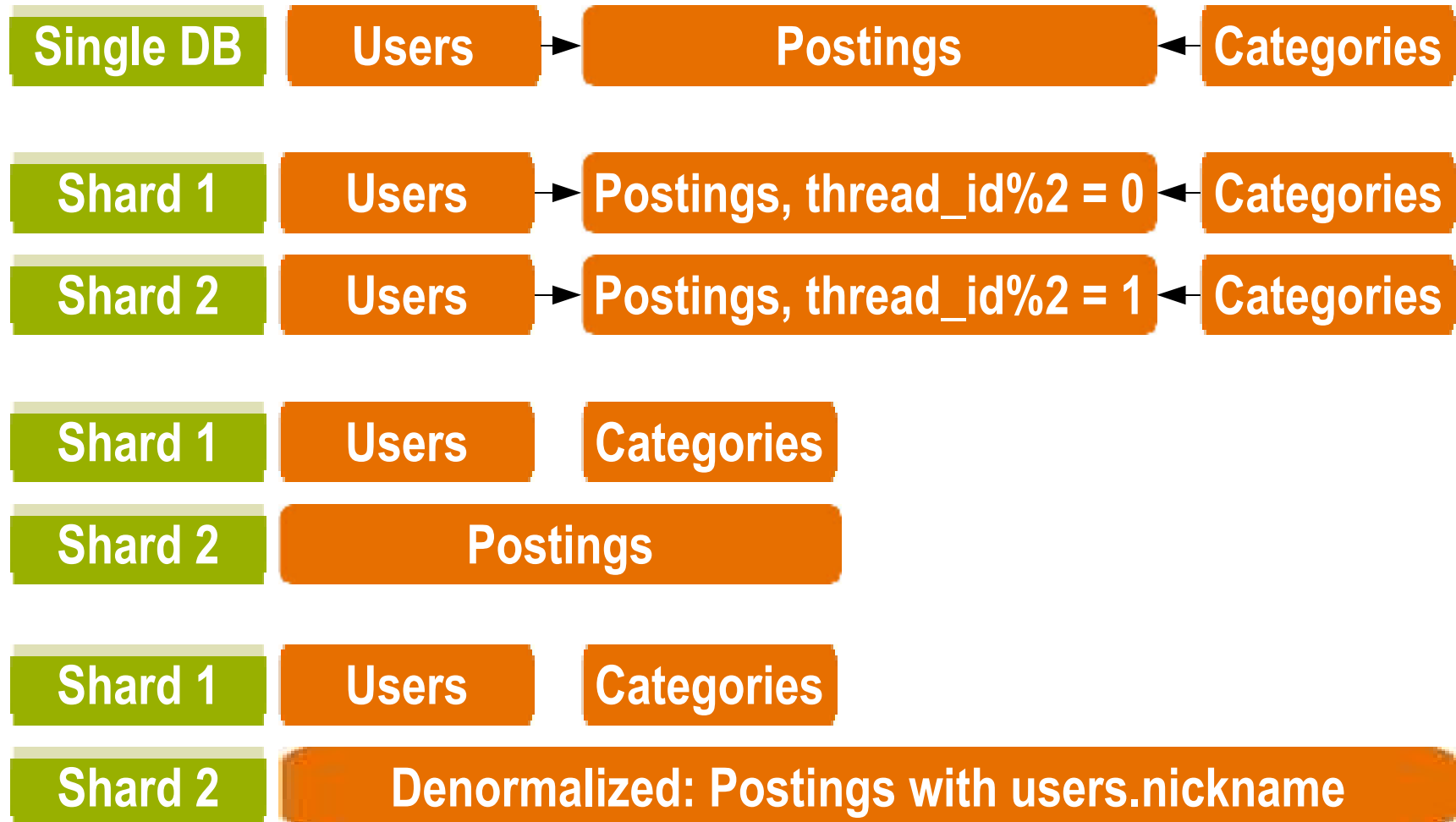
Cheers – mysqlnd rocks!



Sharding – split and distribute

- **Problem**
 - > CPU bound: too much work for one DB system
 - > Disk bound: too large entities for one DB system
- **Solution**
 - > Split schema and distribute data
 - > Use 1, 2, 4, 8, 16, 32, 64, 128, ... 16384 blades

How to split and distribute?



Your problems... not mine...

- Joins, Unions, Intersections
- Grouping
- Selection and projection on groups
- Aggregation
- Primary Keys
- Referential integrity (Foreign Keys)
- (De-)Normalization

Where to split and distribute?

- **Application, DAO, ...**
 - > New shard? Expensive programming to follow
- **Framework, SOA, ...**
 - > Ask Rasmus...
- **Driver**
 - > Which PHP driver can do it? mysqlnd?
- **(Transparent) Proxy**
 - > For example, MySQL Proxy, HSCALE

“Transparent” Proxy with mysqlnd?

bzr clone lp:~johannes-s/php-mysqlnd/mysqlnd-to-stream

```
$mysqli = mysqli_connect("host", "user", "pw", "db");
```

```
$stream = mysqli_conn_to_stream($mysqli);
```

```
stream_filter_register("rewrite", "rewrite_filter");  
stream_filter_append($stream, "rewrite");
```

```
$res = mysqli_query($mysqli, "SELECT 1 AS _one");  
while ($row = mysqli_fetch_assoc($res))  
    var_dump($row);
```

```
array(1) { ["_one"]=> string(1) "2" }
```

Query Rewriting with mysqlnd

100% experimental – no packet decoders exported to PHP

```
class rewrite_filter extends php_user_filter {
    function filter($in, $out, &$consumed, $closing) {
        while ($bucket = stream_bucket_make_writeable($in)) {
            if (strstr($bucket->data, 'SELECT 1')) {
                $bucket->data = str_replace(
                    'SELECT 1', 'SELECT 2', $bucket->data);
            }
            $consumed += $bucket->datalen;
            stream_bucket_append($out, $bucket);
        }
        return PSFS_PASS_ON;
    }
}
```

Sharding - a forum example

- **Distribution logic**
 - > Implemented inside the PHP application
 - > “Users click on categories to read postings”
- **ER-Model, 3 shards**
 - > Split postings by categories.id
 - > Denormalize postings: add users.nickname

Shard 1

Users

Categories

Shard 2

Postings with users.nickname, category_id % 2 = 0

Shard 3

Postings with users.nickname, category_id % 2 = 1

Your new problems...

- **Show all postings of a user**
 - > Union operation over shard 2 and shard 3
 - > Fetch user information from shard 1
- **Calculate the total number of postings**
 - > Aggregation on shard 2 and shard 3

Shard 1

Users

Categories

Shard 2

Postings with users.nickname, category_id % 2 = 0

Shard 3

Postings with users.nickname, category_id % 2 = 1

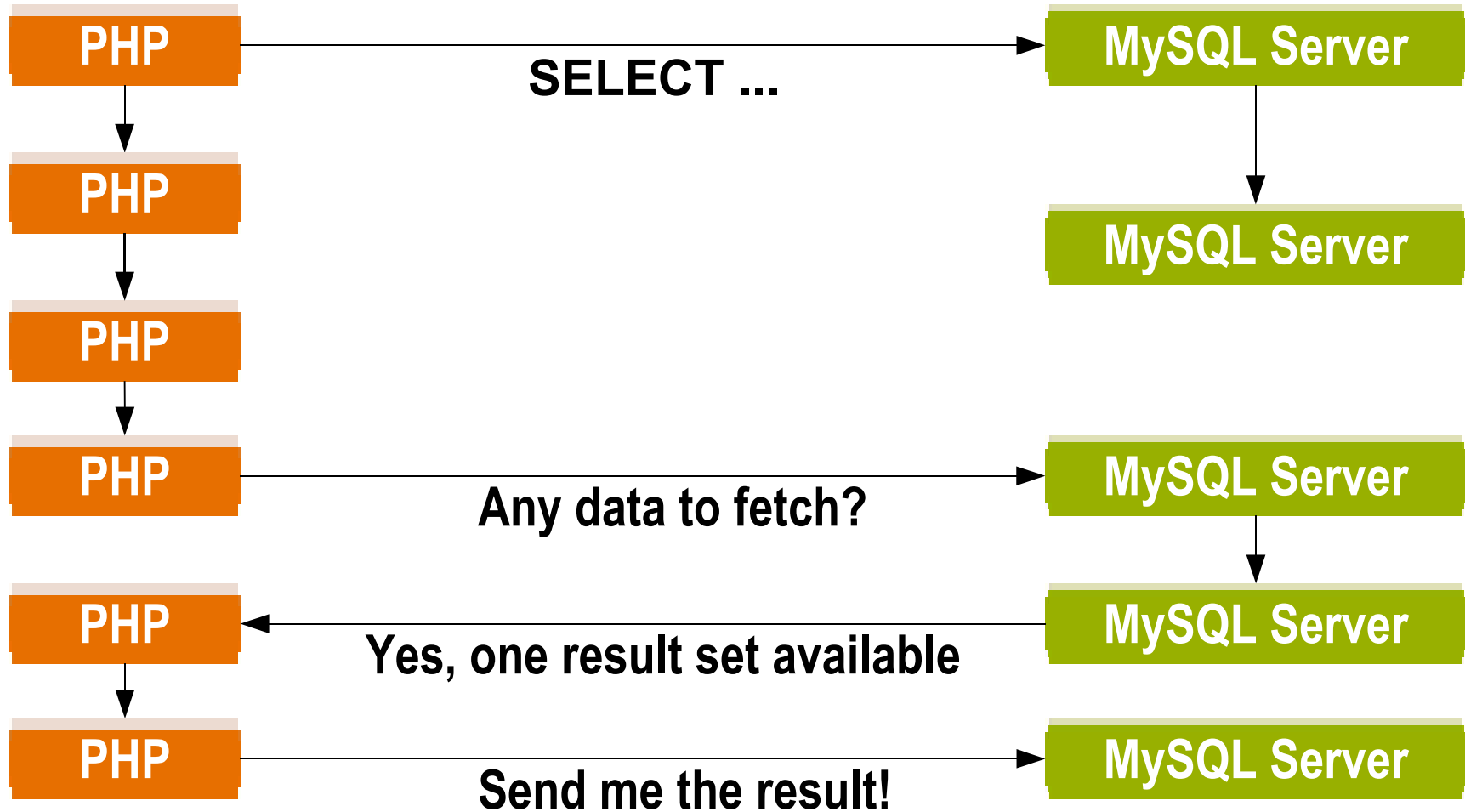
Show all postings of a user

```
$shard1 = mysqli_connect('shard1', ...);  
$res = $shard1->query('SELECT ... FROM users WHERE id = ...');  
display_user($res);  
$res->free_result(); $shard1->close();
```

```
$shard2 = mysqli_connect('shard2', ...);  
$res = $shard2->query('SELECT ... FROM postings WHERE ...');  
display_postings($res);  
$res->free_result(); $shard2->close();
```

```
$shard3 = mysqli_connect('shard3', ...);  
$res = $shard3->query('SELECT ... FROM postings WHERE ...');  
display_postings($res);  
$res->free_result(); $shard3->close();
```

The basic idea



New asynchronous API

```
boolean mysqli_query(  
    string query,  
    MYSQLI_ASYNC)
```

```
int mysqli_poll(  
    array $connections,  
    array $except,  
    array $rejected,  
    int $tv_sec  
    [, int tv_usec])
```

```
mixed mysqli_reap_async_query(  
    mysqli $connection)
```

Asynchronous “Show all ...” - I

```
$shard1 = mysqli_connect('shard1', ...);
```

```
$shard2 = mysqli_connect('shard2', ...);
```

```
$shard3 = mysqli_connect('shard2', ...);
```

```
$shard1->query('... FROM users ...', MYSQLI_ASYNC);
```

```
$shard2->query('... FROM postings ...', MYSQLI_ASYNC);
```

```
$shard3->query('... FROM postings ...', MYSQLI_ASYNC);
```

Asynchronous “Show all ...” - II

```
$all_links = array($shard1, $shard2, $shard3);
$processed = 0;
do {
    $links = $errors = $reject = array();
    foreach ($all_links as $link)
        $links[] = $errors[] = $reject[] = $link;
    if (0 == ($ready = mysqli_poll($links, $errors, $reject, 1, 0))
        continue;

    foreach ($links as $k => $link) {
        if ($res = mysqli_reap_async_query($link)) {
            mysqli_free_result($res);
            $processed++;
        }
    }
} while ($processed < count($all_links));
```

Synchronous vs. asynchronous

1000ms

500ms

600ms

- Time required: $\text{sum}(t_1 + t_2 + \dots + t_n)$
Example: $1000 \text{ ms} + 500\text{ms} + 600\text{ms} = 2100\text{ms}$

1000ms

500ms

600ms

- Time required: $\text{max}(t_1 + t_2 + \dots + t_n)$
Example: $\text{max}(1000\text{ms}, 500\text{ms}, 600\text{ms}) = 1000\text{ms}$

Is it faster?

```

$start = microtime(true);
$m1 = mysqli_connect('host', 'user', 'password', 'schema');
$m2 = mysqli_connect('host', 'user', 'password', 'schema');
mysqli_query($m1, 'SELECT SLEEP(0.10)', MYSQLI_ASYNC);
mysqli_query($m2, 'SELECT SLEEP(0.25)', MYSQLI_ASYNC);
printf("Query   : %2.2fs\n", microtime(true) - $start);
while ($processed < 2) {
    $links = array($m1, $m2);
    if (mysqli_poll($links, array(), array(), 0, 50000)) {
        foreach ($links as $k => $link)
            if ($res = mysqli_reap_async_query($link)) {
                mysqli_free_result($res);
                printf("Fetch %d : %2.2fs\n", ++$processed,
                    microtime(true) - $start);
            }
    }
    printf("Poll    : %2.2fs\n", microtime(true) - $start);
}

```

| | |
|---------|----------|
| Query | : 0.00s, |
| Poll | : 0.05s |
| Fetch 1 | : 0.11s |
| Poll | : 0.11s |
| Poll | : 0.15s |
| Poll | : 0.21s |
| Fetch 2 | : 0.26s |
| Poll | : 0.26s |

Mixing SELECT and INSERT

```
$m1 = mysqli_connect('host', 'user', 'passwd', 'database');
$m2 = mysqli_connect('host', 'user', 'passwd', 'database');
mysqli_query($m1, 'SELECT SLEEP(0.10)', MYSQLI_ASYNC);
mysqli_query($m2, 'INSERT INTO users(id) VALUES (100)', MYSQLI_ASYNC);

while ($processed < 2) {
    $links = array($m1, $m2);
    if (mysqli_poll($links, array(), array(), 0, 50000)) {
        foreach ($links as $link)
            if (is_object($res = mysqli_reap_async_query($link))) {
                $processed++;
                mysqli_free_result($res);
            } else {
                $processed++;
            }
    }
}
```

Handling Server errors

```

$m1 = mysqli_connect('localhost', 'user', 'password', 'schema');
$m2 = mysqli_connect("localhost", "user", "password", "schema");
mysqli_query($m1, 'SELECT NIXNUTZ FOR PREDISENT', MYSQLI_ASYNC);
mysqli_query($m2, "SELECT 1", MYSQLI_ASYNC | MYSQLI_USE_RESULT);
while ($processed < 2) {
    $links = array($m1, $m2);
    if (mysqli_poll($links, array(), array(), 0, 50000))
        foreach ($links as $k => $link) {
            if (is_object($res = mysqli_reap_async_query($link))) {
                var_dump(mysqli_fetch_assoc($res)); mysqli_free_result($res);
            } else if (mysqli_errno($link))
                printf("[%d] %s\n", mysqli_errno($link), mysqli_error($link));
            else
                printf("no error, no result\n");
            $processed++;
        }
    }
}

```

```

array(1) {
    [1]=>
        string(1) "1"
}

```

```

[1064] You have an error in your SQL
syntax; check the manual that
corresponds to your MySQL server version
for the right syntax to use near
'PREDISENT' at line 1

```

Detecting invalid handles

```

$m1 = mysqli_connect('host', 'user', 'password', 'schema');
$m2 = mysqli_connect('host', 'user', 'password', 'schema');
printf("Connection %d: no query\n", mysqli_thread_id($m1));
mysqli_query($m2, 'SELECT 1', MYSQLI_ASYNC | MYSQLI_USE_RESULT);
printf("Connection %d: SELECT 1\n", mysqli_thread_id($m2));
while ($processed < 2) {
    $links = array($m1, $m2);
    $rejected = array($m1, $m2);
    if (0 == ($ready = mysqli_poll($links, array(), $rejected, 0, 50000)))
        continue;
    foreach ($rejected as $link)
        printf("Connection %d: rejected\n", mysqli_thread_id($link));
    $processed += count($rejected);
    foreach ($links as $link)
        printf("Connection %d: accepted\n", mysqli_thread_id($link));
    $processed += count($links);
}

```

```

Connection 205: no query
Connection 206: SELECT 1
Connection 205: rejected
Connection 206: accepted

```

Daily bulk INSERT - ./ me! (Part1)

```

if (mysqli_poll($links, array(), array(), 0, 5000))
    foreach ($links as $link) {

        mysqli_reap_async_query($link);
        if (mysqli_errno($link))
            die(mysqli_error($link));

        $all_links[mysqli_thread_id($link)]['inserted']++;

        if ($all_links[mysqli_thread_id($link)]['inserted'] < $rows) {
            if (mysqli_query($link, $query, MYSQLI_ASYNC))
                $i++;
            else
                die(mysqli_error($link));
        }
    }
}

```

| | |
|---|-------|
| Sequential INSERT (2 shards, 1000 rows) | 4.22s |
| 2000 rows deleted | |
| 'Parallel' INSERT (2 shards, 1000 rows) | 1.98s |
| 2000 rows deleted | |

Andrey suffers from Insomnia

Hi Ulf,

I did a small modification to mysqlnd, locally, that **enables it to send UPSERT queries in a batch, without reading the result from the query.**

[...]

Results are amazing (see total! - **ASYNC INSERTs take less than 60% of the SYNC**, if not less). You can show a slide tomorrow about it.

Andrey suffers from Insomnia II

100% experimental!

Don't trust the performance figures!

Where to get mysqlnd with async?

If still possible to commit into 5.3 tree: PHP 5.3+ CVS

```
// the super secret Launchpad repository with all raw-bin ideas

bzip clone lp:~andrey-mysql/php-mysqlnd/trunk/

// Get PHP 5.3 from cvs.php.net

cd php5/ext
rm -rf mysqlnd mysqlnd
cp -R /path/to/bzip_clone/trunk/mysqlnd mysqlnd
cp -R /path/to/bzip_clone/trunk/php5/ext/mysqlnd mysqlnd
cd ..
./buildconf --force ; ./configure --with-mysqlnd=mysqlnd
make clean; make
```



The End
Feedback: ulf.wendel@sun.com

