Welcome!

Enterprise PHP Bananas

Intl. PHP Conference

Nov 10, 2004. Frankfurt, Germany

Derick Rethans <derick@php.net>

eZ systems

Share your information

SRM: Script Running Machine

- Functions to work with persistent Objects
- Bridge between PHP Client and Objects
- Objects run in threads
- Transparent Object Marshalling

eZ systems

Share your information

Bananas

- Persistent
- A bit like Java beans
- Compiled only once

eZ systems

Share your information

Users on-line

Before SRM

Typical script to count the users 'online':

```php
<?php
$timeoutseconds = 300;

$timestamp = time();
$timeout = $timestamp - $timeoutseconds;

mysql_connect();
mysql_select_db('users');

$insert = mysql_query(
  "INSERT INTO online VALUES ('$timestamp', '$REMOTE_ADDR', '$PHP_SELF')");

$delete = mysql_query(
  "DELETE FROM online WHERE timestamp < $timeout");

$result = mysql_query(
  "SELECT DISTINCT ip FROM online WHERE file = '$PHP_SELF'");

$users = mysql_num_rows($result);
?>
```

eZ systems

Users on-line

With SRM

Script utilizing SRM to count the users 'online':

```php
<?php
    $srm = new SRM ('/var/srm.socket');
    $srm->user_active($_COOKIE['PHP_SESSID']);
    echo $srm->get_active_user_count();
?>
```

Daemon side:

```php
<?php
$active_users = array();

function user_active($id) {
    $GLOBALS['active_users'][$id] = time();
}

function cleanup_inactive_users() {
    foreach($GLOBALS['active_users'] as $id => $time) {
        if ($time < time() - 300) {
            unset($GLOBALS['active_users'][$id]);
        }
    }
}

function get_active_user_count() {
    cleanup_inactive_users();
    return count($GLOBALS['active_users']);
}
?>
```

Share your information

Persistent objects

File Class - Before SRM

__sleep and __wakeup:

```php
// file.class.php
<?php
    class File {
        function File($filename) {
            $this->filename = $filename;
            $this->fp = fopen($filename, 'rb');
        }

        function seek($pos) {
            fseek($this->fp, $pos);
        }

        function __sleep() {
            $this->pos = ftell($this->fp);
            return array('fp', 'pos', 'filename');
        }

        function __wakeup() {
            $this->fp = fopen($this->filename, 'rb');
            fseek($this->fp, $this->pos);
        }
    }
? >
```

Persistent objects

Examples - Before SRM

__sleep and __wakeup example:

```php
// example1.php
<?php
    require 'file.class.php';
    session_start();

    $f = new File('/etc/hosts');
    $f->seek(20);

    $_SESSION['f'] = $f;
? >
```

```php
// example2.php
<?php
    require 'file.class.php';
    session_start();

    var_dump($_SESSION['f']);
? >
```

eZ systems

Share your information

Trees

Before SRM

```php
<?php
    class node {
        var $key; var $value; var $left; var $right;
    }

    class binsearch {
        var $root = NULL;

        function binsearch($elements) {
            foreach ($elements as $key => $value) {
                $this->add_value($key, $value);
            }
            echo "<pre>", substr(var_export($this->root, true), 0, 190), "\n...",
                 substr(var_export($this->root, true), -40), "</pre>";
        }

        function add_value($key, $value) {
            $ptr = &$this->root;

            while ($ptr != NULL) {
                if ($key < $ptr->key) {
                    $ptr = &$ptr->left;
                } else {
                    $ptr = &$ptr->right;
                }
            }
            $ptr = new node;
            $ptr->key = $key;
            $ptr->value = $value;
            $ptr->left = NULL;
            $ptr->right = NULL;
        }

        function get_value($key) {
            $ptr = &$this->root;

            while ($key != $ptr->key && $ptr != NULL) {
                if ($key < $ptr->key) {
                    $ptr = &$ptr->left;
                } else {
                    $ptr = &$ptr->right;
                }
            }
            return $ptr ? $ptr->value : FALSE;
        }
    }
?>
```

eZ systems

Share your information

Trees

Example - Before SRM

```php
<?php
    include "/home/httpd/presentations/slides/srm/binsearch.class.php";

    $elements = array (
        'jan'    => 'Dorsten',       'derick' => 'Dieren',
        'stig'   => 'Trondheim',     'dan'    => 'Portland',
        'rasmus' => 'San Francisco', 'ilia'   => 'Montreal'
    );

    $tree =& new binsearch($elements);
    echo $tree->get_value('rasmus');
?>
```

output:

```
class node {
  public $key = 'jan';
  public $value = 'Dorsten';
  public $left =
  class node {
    public $key = 'derick';
    public $value = 'Dieren';
    public $left =
    class node
...
    };
    public $right = NULL;
  };
}

San Francisco
```

eZ systems

Persistent Tree

Bananized Class

```
<?php
    class node {
        var $key; var $value; var $left; var $right;
    }

    class binsearch extends Banana {
        var $root = NULL;

        function binsearch($elements) {
            foreach ($elements as $key => $value) {
                $this->add_value($key, $value);
            }
        }

        function add_value($key, $value) {
            /* Same as before */
        }

        function get_value($key) {
            /* Same as before */
        }

        function get_all() {
            return $this->root;
        }
    }

    $tree =& new binsearch(array());
    $tree->run();
?>
```

Share your information

eZ systems

Share your information

Persistent Tree

Clients to add and search elements

Adding a value to the binairy tree:

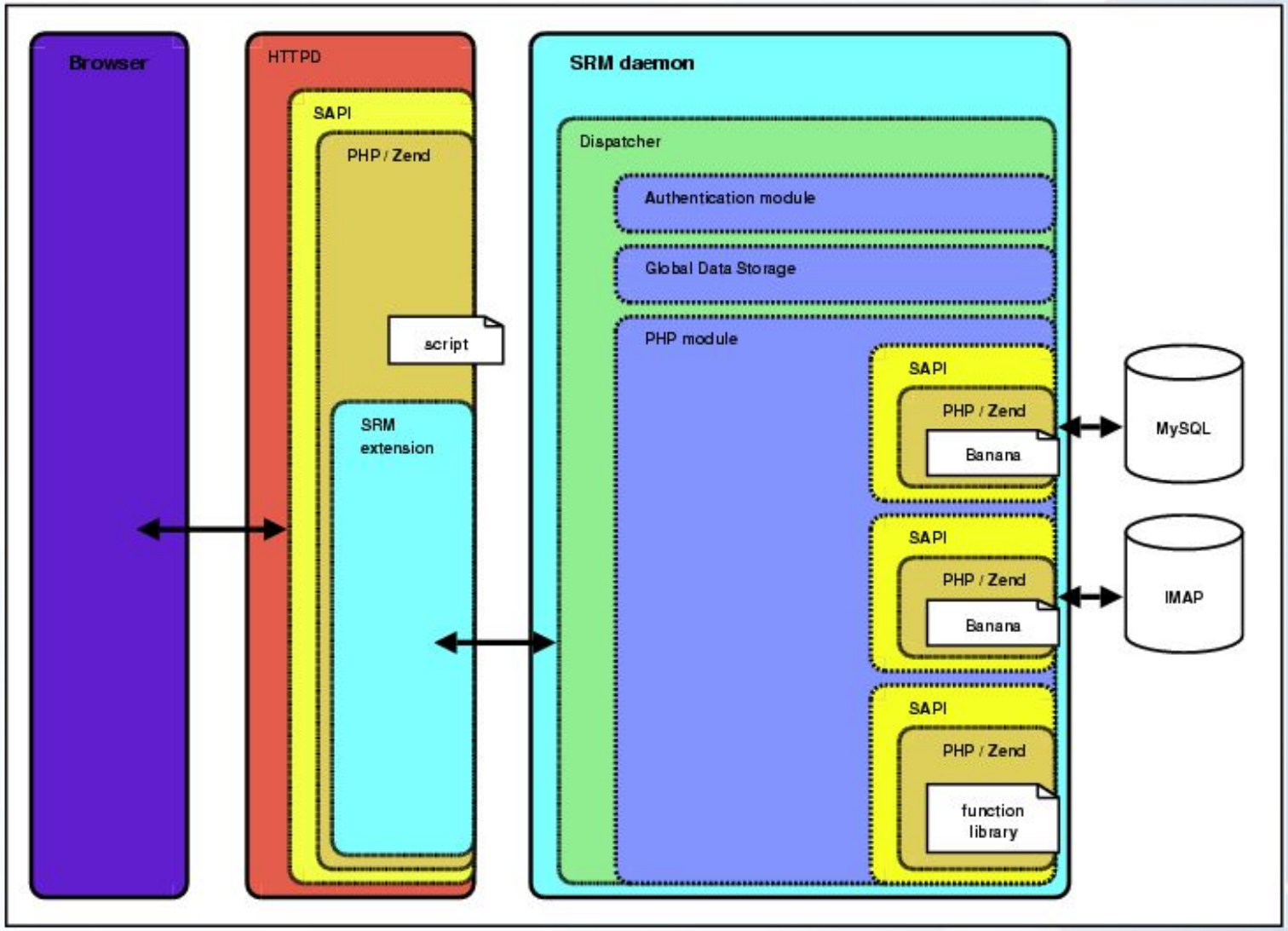
```php
<?php
    $s = new SRM('/tmp/srm.socket');
    $t = new SRMApp($s, 'binsearch');

    $t->add_value($argv[1], $argv[2]);
    $all = $t->get_all();

    var_dump($all);
?>
```

Searching for a key in the binairy tree:

```php
<?php
    $s = new SRM('/tmp/srm.socket');
    $t = new SRMApp($s, 'binsearch');

    echo $t->get_value($argv[1])."\n";
?>
```

eZ systems

Design diagram



Share your information

Compiling

Setting up the sources

Download sources:

```
$ mkdir -p /dat/dev/php/srm
$ cd /dat/dev/php/srm
$ wget http://www.php.net/distributions/php-4.3.7.tar.bz2

$ cvs -d :pserver:srmread@cvs.xdebug.org:/repository login
Password: srmread
$ cvs -d :pserver:srmread@cvs.xdebug.org:/repository co srm
$ cvs -d :pserver:srmread@cvs.xdebug.org:/repository co sapi_srm
$ cvs -d :pserver:srmread@cvs.xdebug.org:/repository co php_srm
```

Preparing sources:

```
$ tar -xvjf php-4.3.7.tar.bz2
$ cp -R php-4.3.7 php-4.3.7-sapi

$ cd srm
$ ./buildconf

$ cd php-4.3.7-sapi/sapi
$ ln -s ../../sapi_srm srm
$ cd ../ext
$ ln -s ../../php_srm srm
$ cd ..
$ rm configure && ./buildconf

$ cd ../php-4.3.7/ext
$ ln -s ../../php_srm srm
$ cd ..
$ rm configure && ./buildconf
$ cd ..
```

eZ systems

Share your information

Compiling

Three Parts: Daemon, PHP Client and PHP SAPI

Daemon:

```
$ ./configure
$ make
$ sudo make install
```

SAPI for PHP:

```
$ cd php-4.3.7-sapi
$ ./configure --with-srm-sapi --disable-cli --disable-pear --with-srm
$ make clean
$ rm ext/srm/*.o
$ rm ext/srm/*.lo
$ rm sapi/srm/*.o
$ rm sapi/srm/*.lo
$ make
$ sudo make install-sapi
```

Client in PHP:

```
$ cd php-4.3.7
$ ../configure --with-srm ..other options..
$ make clean
$ rm ext/srm/*.o
$ rm ext/srm/*.lo
$ rm sapi/srm/*.o
$ rm sapi/srm/*.lo
$ make
$ sudo make install
```

Configuring and Running SRM

/etc/srm.ini:

```
modules {
    path =  "/usr/local/srm/libexec/";
    load    "php4";
}

banana {
    class_path       = "/usr/local/srm/banana";
    function_library = "/usr/local/srm/banana/lib/library.php";
}
```

Running the daemon:

```
Usage:  srmd [-S] [-L] [-X]
             [-v|-V|-h]
Options:
        -S  Disable signal handling
        -L  Log to screen instead of the logfile
        -X  Do not fork into background

        -v  Display version information and exit
        -V  Display detailed version information and exit
        -h  Display this help message and exit
```

Interaction

With the SRM daemon:

```php
<?php
        // UNIX domain sockets
        $srm = new SRM ('/var/srm.socket');
        // TCP/IP sockets
        $srm = new SRM ('localhost', 7777);

        $srm->get_loaded();
?>
```

With Bananas:

```php
<?php
        $binsearch = new SRMApp($srm, 'binsearch');
        $binsearch->method('param1', 2);
?>
```

Moving Objects

foo.class.php:

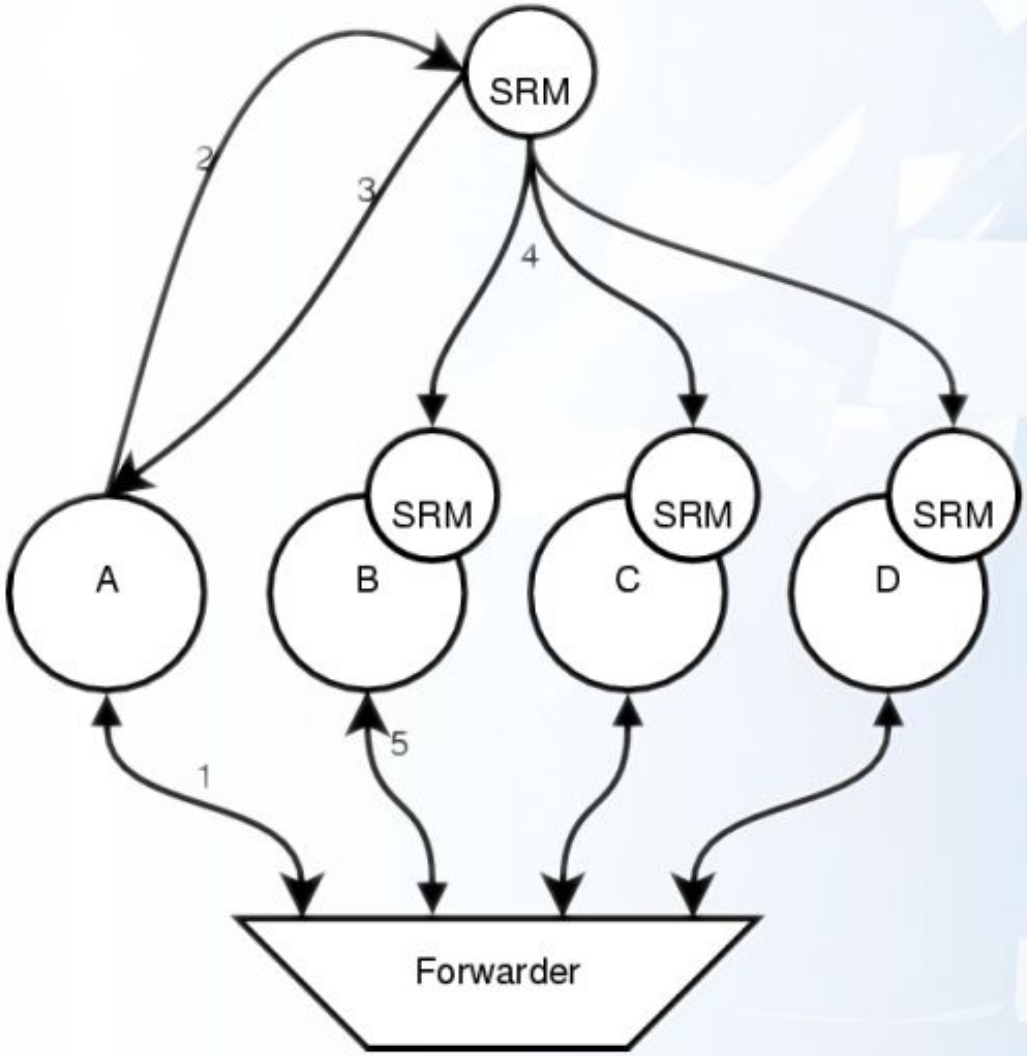
```php
<?php
    class foo {
        var $bar = '';

        function foo42() {
            $this->bar = 'foo';
            return func_get_args();
        }

        function echo_bar() {
            echo $this->bar;
        }
    }
?>
```

Script (objmove.php):

```php
<?php
    include '/usr/local/srm/banana/lib/foo.class.php';

    $f = new foo;
    $s = new SRM('/tmp/srm.socket');
    $f = new SRMApp($s, $f, 'key2');

    var_dump($f->foo42(1, 2, 'bar', 3, 4));

    /* Properties don't work yet */
    echo $f->bar;
?>
```

Demonstration - Cluster Management

Overview



Demonstration - Cluster Management

Configuration

Main Server configuration:

```
srm {
    sockname       =  "/tmp/srm.socket.main";
    logfile        =  "/var/log/srm";
    disable_tcp_ip = 1;
}

banana {
    class_path       = "/dat/dev/php/srm/demo/banana";
    function_library = "/dat/dev/php/srm/demo/library/main.php";
}
```

Client Server configuration:

```
srm {
    sockname       =  "/tmp/srm.socket.client1";
    logfile        =  "/var/log/srm";
    disable_tcp_ip = 1;
}

banana {
    class_path       = "/dat/dev/php/srm/demo/banana";
    function_library = "/dat/dev/php/srm/demo/library/client.php";
}
```

eZ systems

Share your information

Demonstration - Cluster Management

Master Code

```php
<?php
$client_last_nr = 0; $task_last_nr = 0;
$clients = array();  $client_tasks = array();

function register($socket) {
    $GLOBALS['client_last_nr']++;
    $GLOBALS['clients'][$GLOBALS['client_last_nr']] = $socket;
    $GLOBALS['client_tasks'][$client_last_nr] = array();

    echo "Registered client with socket '$socket'\n";
    return $GLOBALS['client_last_nr'];
}

function unregister($id) {
    unset($GLOBALS['clients'][$id], $GLOBALS['client_tasks'][$id]);
    echo "Unregistered client #$id";
}

function handle_uploaded_file($hostname, $path, $task_title) {
    $task_last_nr++;

    foreach ($GLOBALS['clients'] as $id => $obj) {
        if (is_string($obj)) {
            $obj = new srm($obj);
            $GLOBALS['clients'][$id] = $obj;
        }
        $GLOBALS['client_tasks'][$id][$task_last_nr] = true;
        $method = "execute_task_copyfile";
        $obj->$method($hostname, $path, $task_title);
    }
}
?>
```

eZ systems

Demonstration - Cluster Management

Client Code

```php
<?php
include './paths.php';
$client_id = false;

register($_SERVER['SRM_SOCKET']);
register_shutdown_function('unregister');

function register($sockname) {
    global $client_id;

    $srm = new SRM(TMP_PATH . 'srm.socket.main');
    $client_id = $srm->register($sockname);
    if (!$client_id) {
        return false;
    }
    echo "Registered with ID #$client_id\n";
    return true;
}

function unregister() {
    global $client_id;

    $srm = new SRM(TMP_PATH . 'srm.socket.main');
    echo "Unregistering client";
    $srm->unregister($client_id);
}

function execute_task_copyfile($hostname, $path, $task_title) {
    echo "scp $hostname:$path /tmp/t.part\nmv /tmp/t.part $path\n";
    sleep(2);
}
?>
```

Share your information

eZ systems

Demonstration - Cluster Management

Usage

```php
<?php
    echo "This is a simulation of uploading a file.\nHere we go:\n\n";

    echo "Uploading file"; flush();
    for ($i = 0; $i < 3; $i++) {
        sleep(1);
        echo "..."; flush();
    }
    echo "Done\n\n\n";

    echo "Now we register the uploaded file with the SRM daemon... ";
    $s = new SRM('/tmp/srm.socket.main');
    $s->handle_uploaded_file('host', '/path/to/file', 'Our New File');
    echo "Done!\n";
?>
```

Share your information

Demonstration - Cluster Management

Improvements

- Provide better detection of down servers
- Provide asynchronus distribution of tasks

Future

- Periodic calling of functions
- Support for PHP 5 / Zend Engine 2 (exceptions...)
- Automatic saving of state
- ACLs to Bananas
- Load balancing
- ...

eZ systems

Share your information

Questions

eZ systems

Share your information

Resources

SRM: http://www.vl-srm.net

PHP: http://www.php.net