eZ publish

Share your information

Welcome!

Performance Workshop

PHP Quebec

March 30th, 2005. Montréal, Canada

Derick Rethans and Marcus Börger
<internals@php.net>

eZ publish

Share your information

Questions

eZ publish

Share your information

Optimization Strategies

- Caching
- Optimizing Database Use
- Optimizing PHP Code
- Tuning the Webserver
- Reducing Bandwidth

eZ publish

Share your information

Caching

eZ publish

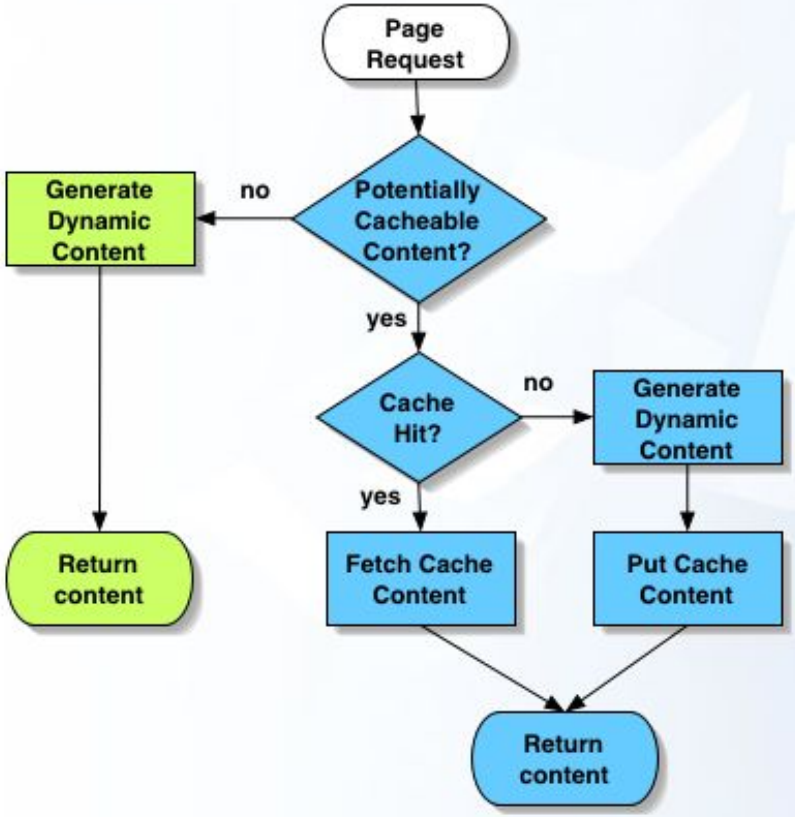
Share your information

What Is Caching?

Caching is the recognition and exploitation of the fact that most "dynamic" data does not change every time you request it.

eZ publish

How Caching Works



Share your information

eZ publish

Share your information

Caching Example

PEAR::Cache_Lite

```php
<?php
    /* Include the class */
    require_once('Cache/Lite.php');

    /* Set a key for this cache item */
    $id = 'newsitem1';

    /* Set a few options */
    $options = array(
        'cacheDir' => '/var/cache/news/',
        'lifeTime' => 3600
    );

    /* Create a Cache_Lite object */
    $Cache_Lite = new Cache_Lite($options);

    /* Test if there is a valid cache-entry for this key */
    if ($data = $Cache_Lite->get($id)) {
        /* Cache hit! */
    } else {
        /* Cache miss! */
        ob_start();
        /* Generate content */
        $data = ob_get_contents();
        $Cache_Lite->save($data);
    }
?>
```

eZ publish

Share your information

Pros and Cons of Caching

Pros:

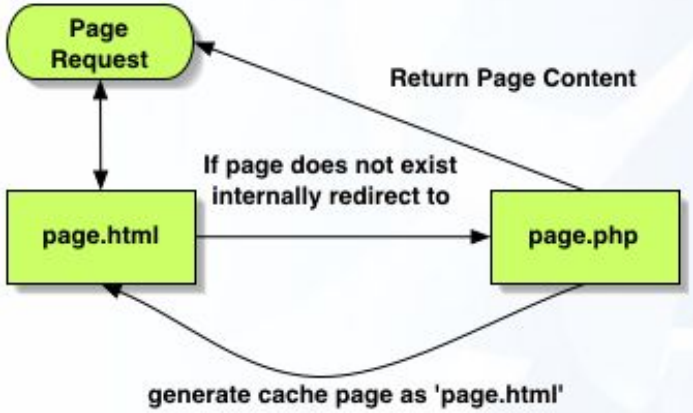
- Potentially Immense Speed-Up
- Decreased Resource Consumption

Cons:

- Increase in Architectural Complexity
- Potential for Stale or Inconsistent Data

eZ publish

Total Page Caching



Share your information

eZ publish

Share your information

Livedocs

An Example of On-Demand Page Generation

Set up a 404 error handler in .htaccess:

```
RewriteEngine on
RewriteRule /.*\.[^h][^t][^m][^l]$ /$1.html
ErrorDocument 404 /index.php
DirectoryIndex index.php
```

index.php:

```
<?php
if (!empty($_SERVER['REDIRECT_URL'])) {
    // This is the requested page that caused the error
    $current_page = substr($_SERVER['REDIRECT_URL'],
strlen(WEBBASE));
}

...

if (!FORCE_DYNAMIC) {
    $contents = ob_get_contents();
    ob_clean();
    $f = fopen($lang . "/" . $current_page . ".html", 'w');
    fwrite($f, $contents);
    fclose($f);
    echo $contents;
}
?>
```

eZ publish

Static Squids

- Reverse Proxy
- Tied into Apache
- Static Generated
- Huge Performance Boost
- No 'Dynamic' Content

Share your information

Squid

Some effort to use with PHP:

```
<?php
$timestamp = /* timestamp for last updated part */;
$tsstring = gmdate("D, d M Y H:i:s ", $timestamp)."GMT";

if ($_SERVER["HTTP_IF_MODIFIED_SINCE"] == $tsstring) {
    /* The UA has the exact same page we have. */
    header("HTTP/1.1 304 Not Modified");
    exit();
} else {
    header("Last-Modified: ".$tsstring);
}
?>
```

Set Cache-Control headers so external caches know how to respect your content:

```
<?php
  header("Last-Modified: Mon, 13 Se4 2002 09:44:13 GMT");
  header("Expires: Mon, 13 Sep 2004 10:44:13 GMT");
  header("Cache-Control: must-revalidate, max-age=15, s-maxage=0,
private");
?>
```

Share your information

eZ publish

Share your information

Static Cache in eZ publish

An Example of On-Publish Generation

```php
<?php
    function cacheURL($url, $nodeID = false, $skipExisting = false)
    {
        $fileName = "http://$hostname$dir$url";
        $content = @file_get_contents($fileName);
        $this->storeCachedFile($file, $content);
    }

    function storeCachedFile($file, $content)
    {
        $fp = fopen($tmpFileName, 'w');
        if ($fp) {
            fwrite($fp, $content);
            fclose($fp);
            rename($tmpFileName, $file);
        }
    }
```

Set up RewriteRules:

```
RewriteEngine on

RewriteCond   /dat/ez.no/static/index.html -f
RewriteRule   ^/$     /static/index.html [L]

RewriteCond   %{REQUEST_METHOD}       !^POST$
RewriteCond   /dat/ez.no/static$1/index.html -f
RewriteRule   ^(.*)$  /static$1/index.html [L]
```

eZ publish

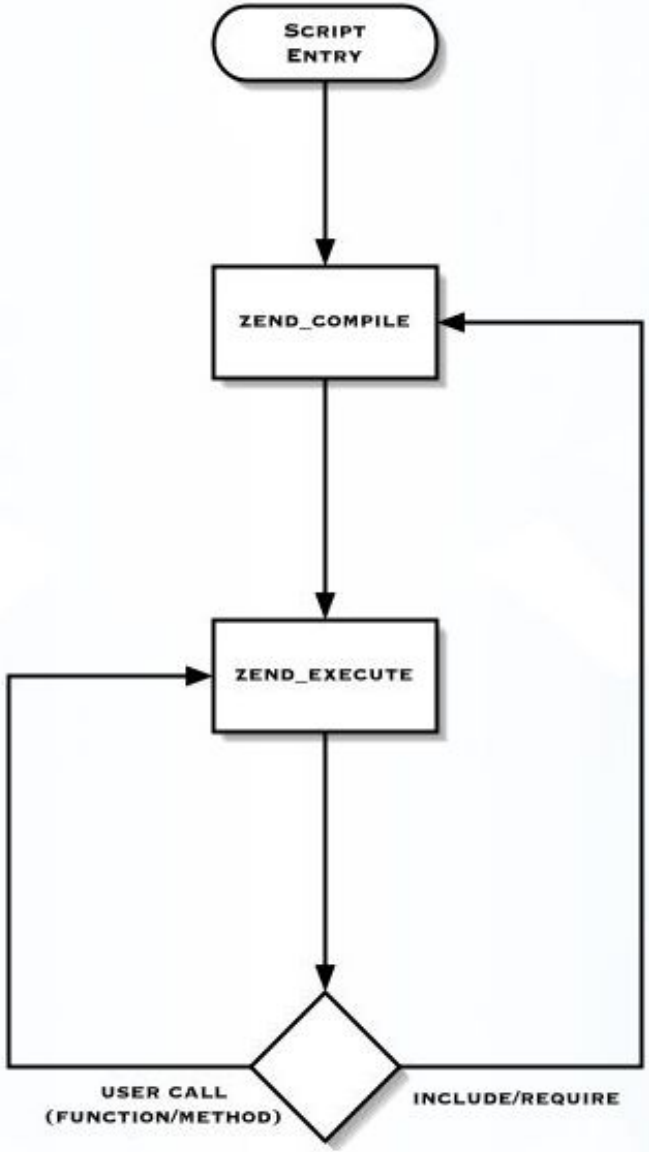
Share your information

Server Tuning

eZ publish

Compiler Caches

What can be optimized



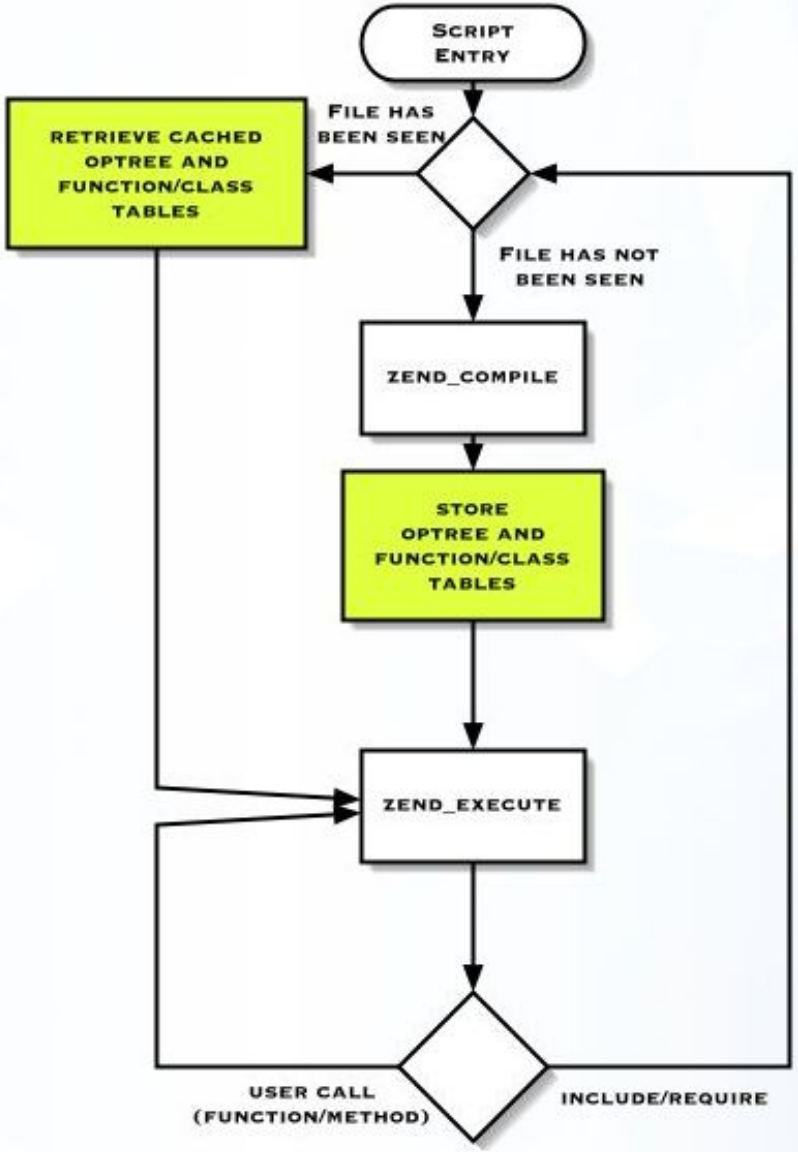
- This cycle happens for every include file, not just for the "main" script
- Compilation can easily consume more time than execution

Share your information

eZ publish

Compiler Caches

How it works



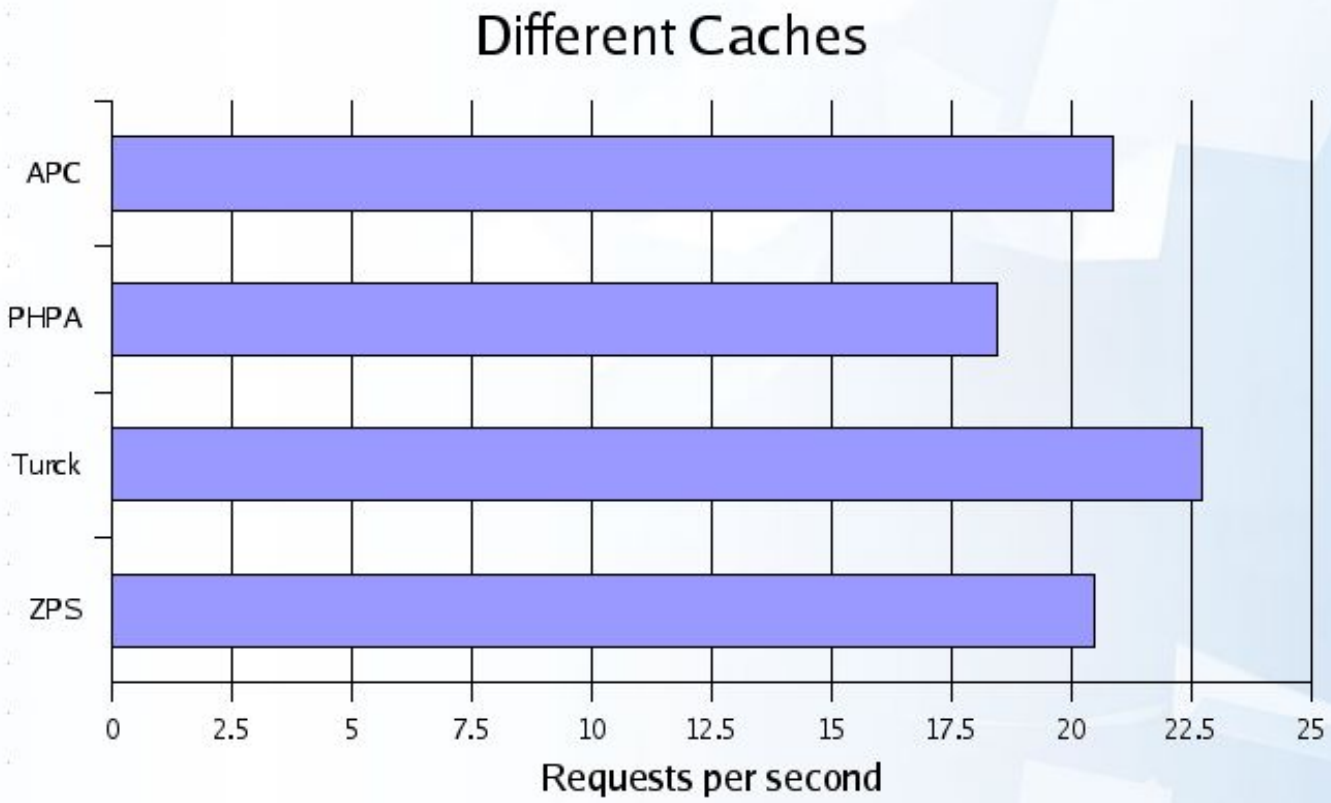
- In general, each source file is compiled once
- Compilation overhead becomes inconsequential
- Cache introduces its own overhead due to dynamic nature of includes

Share your information

eZ publish

PHP Accelerators

- APC
- Zend Performance Suite
- Turck MM Cache
- PHP Accelerator



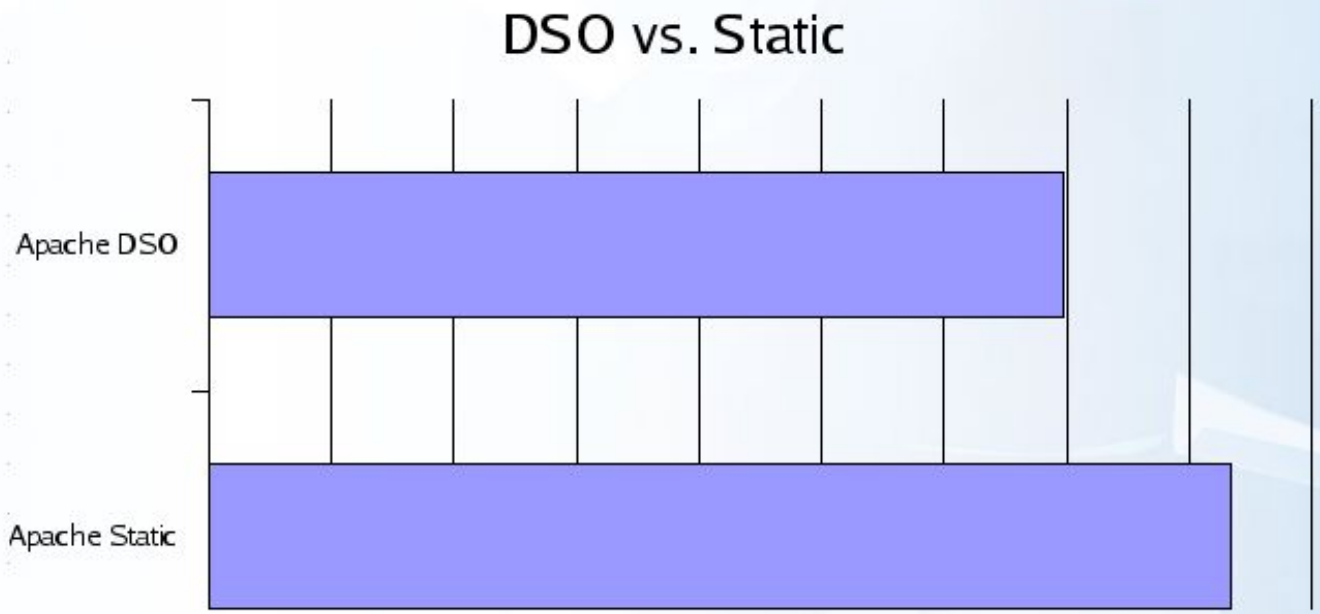
Share your information

eZ publish

Share your information

Apache Is Slow!

- Use static compiled modules or compile with --prefer-non-pic
- Enable all compiler optimizations with -O3
- Tell the compiler what CPU you use via -m*arch* -m*cpu*
- Use CPU specfic features -msse -mmmx -mfpmath=sse
- Other options: -fomit-frame-pointer



Say Squeeze

- Optimized `./configure`
- Memory Limit
- Shorter Path
- Use *include()* in a smarter way
- Ramdisk

Share your information

eZ publish

Share your information

Application Performance

eZ publish

Share your information

CPU Usage

- Avoid Using frequent array accesses, remember, each array access is a hash lookup.

```
<?php
$ar = array();
$ar['name'] = 'Optimization';

for ($i = 0; $i < count($entries); $i++) {
    print $ar['name'];
}
?>
```

- Shorten "required" executions, such as loops.
- Avoid rebuilding large data structures.
- Look for alternatives to regular expressions

eZ publish

Share your information

CPU Usage (2)

- Everything has a cost: include_path, open_basedir, variables_order, magic_quotes, always_populate_raw_post_data
- Cache expensive operations, like generating images
- Function calls are expensive
- User defined functions are more expensive
- Passing parameters take time

eZ publish

Bandwidth Optimizations

Optimizing HTML

- Remove All Comments!
- Use CSS
- Fully qualify links (ie, personal/images/ not personal/images)
- `ext/tidy`

```
<?php
    $opts = array("clean" => true,
                  "drop-proprietary-attributes" => true,
                  "drop-font-tags" => true,
                  "drop-empty-paras" => true,
                  "hide-comments" => true,
                  "join-classes" => true,
                  "join-styles" => true);

    $tidy = tidy_parse_file("php.html", $opts);
    tidy_clean_repair($tidy);

    echo $tidy;
?>
```

Share your information

File I/O

File I/O Problems

- Disk access is slower than memory access
- Simultaneous write/write operations cause data corruption. Therefore locking must occur—which is slow

Tips:

- Use the disk in concentrated bits, meaning do all your writing together.
- Try and put write operations where they will cause the least loss in user experience (remember, web pages are chunked!)
- Close the session handler ASAP with *session_write_close()*

Share your information

eZ publish

Share your information

The Hunt for Speed

- Xdebug: a tool to debug PHP
- Tracing function calls
- Profiling Applications

eZ publish

Tracing the Problems

- Xdebug's Trace Features
- Trace function calls
- Provides details on memory usage

```
/dat/ez.no-33/var/ezno
13:52:21]
-> {main}() /dat/ez.no-33/index.php:0
  -> microtime() /dat/ez.no-33/index.php:24
  -> ob_start() /dat/ez.no-33/index.php:35
  -> error_reporti /dat/ez.no-33/index.php:74
  -> include_once(/dat/ez.no-33/lib/ezutils/classes/ezsys.php) /dat/ez.no-33/index.php:77
    -> include_once(/dat/ez.no-33/lib/ezutils/classes/ezsys.php) /dat/ez.no-33/lib/ezutils/cl
      -> define('EZ_SYS_DEBUG_INTERNALS', FALSE) /dat/ez.no-33/lib/ezutils/classes/ezsys.php:
    -> define('EZ_LEVEL_NOTICE', 1) /dat/ez.no-33/lib/ezutils/classes/ezdebug.php:85
    -> define('EZ_LEVEL_WARNING', 2) /dat/ez.no-33/lib/ezutils/classes/ezdebug.php:86
t [RO]                                                                    2,1          Top
    -> eztextcodec::internalcharset /dat/ez.no-33/lib/ezi18n/classes/eztextcodec.php:581
      -> ezcharsetinfo::realchar de('iso-8859-15') /dat/ez.no-33/lib/ezi18n/classes/eztex
        -> ezcharsetinfo::aliastab /dat/ez.no-33/lib/ezi18n/classes/ezcharsetinfo.php:212
          -> is_array(array ('ascii' => 'us-ascii', 'latin1' => 'iso-8859-1', 'latin2' => 'is
        -> strtolower('iso-8859-15') /dat/ez.no-33/lib/ezi18n/classes/ezcharsetinfo.php:213
  -> include_once(/dat/ez.no-33/lib/ezlocale/classes/ezlocale.php) /dat/ez.no-33/index.php:24
    -> define('EZ_LOCALE_DEBUG_INTERNALS', FALSE) /dat/ez.no-33/lib/ezlocale/classes/ezlocale
  -> ezini::instance() /dat/ez.no-33/index.php:246
    -> eval('class ezlocale { var $Charset = 'iso-8859-1'; var $BlockValues = array ('Databa
  -> ezini->variable('RegionalSettings', 'Debug') /dat/ez.no-33/index.php:247
t [RO]                                                                    349,5        1%
```

Share your information

Xgui

- Simple Tracefile Browser

xGUI profiling frontend for xDebug

File Edit Help

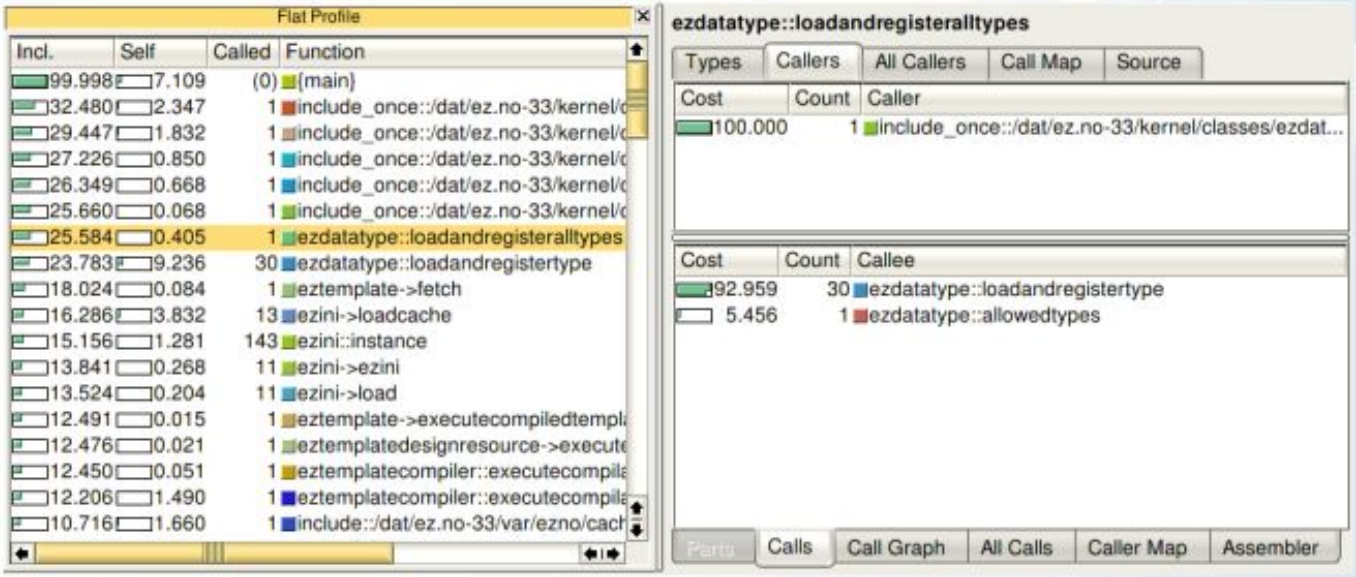
Function	Nr.	Memory	Time	Filename
strlen	8	0	5.6e-05	/home/httpd/pres2/show.php:11
require_once	9	1194408	0.054128	/home/httpd/pres2/show.php:16
require_once	10	104552	0.005591	/home/httpd/pres2/XML_Presentation.php:23
require_once	22	926400	0.03976	/home/httpd/pres2/XML_Presentation.php:24
require_once	23	0	0.000131998	/home/httpd/pres2/objects.php:4
require_once	24	40	0.000185002	/home/httpd/pres2/objects.php:6
_pear_call_destructors	489			/usr/local/lib/php/PEAR.php:0
is_array	490			/usr/local/lib/php/PEAR.php:505
sizeof	491			/usr/local/lib/php/PEAR.php:506

Share your information

eZ publish

Finding the Bottleneck

- Xdebug's Profiling Features
- GUI analysing with KCachegrind
- Provides details on CPU usage



Share your information

Regular Expressions

Its often beneficial to replace regular expressions with simpler functions that have the same effect.

With Regular Expression:

```
<?php
if (preg_match ('/(.*)\@(.*)/', $data, $matches)) {
    print $matches[2];
}
?>
```

Without Regular Expression:

```
<?php
$pos = strrpos ('@', $data);
if ($pos !== false) {
    print substr ($data, $pos);
}
?>
```

In other cases it may help to use regular expressions instead of custom parse routines in your PHP code.

Share your information

File I/O

File I/O Problems

- Disk access is slower than memory access
- Simultaneous write/write operations cause data corruption. Therefore locking must occur—which is slow

Tips:

- Use the disk in concentrated bits, meaning do all your writing together.
- Try and put write operations where they will cause the least loss in user experience (remember, web pages are chunked!)
- Close the session handler ASAP with *session_write_close()*

Share your information

eZ publish

Share your information

Break

eZ publish

Share your information

Database Performance

eZ publish

Share your information

MySQL Table Types

MyISAM

- Efficient For **either** High Volume writes **or** reads
- Table level locking
- No Transaction support

InnoDB

- Efficient locking (Row-level update, non-locking read)
- Foreign Keys, Transactions
- High Concurrency

eZ publish

Visitation model 1

Share your information

Directory structure:

Stored as:

Id	Parent	Name
3	1	Countries
4	3	Belgium
5	3	Netherlands
6	3	Germany
7	5	Business and Economy

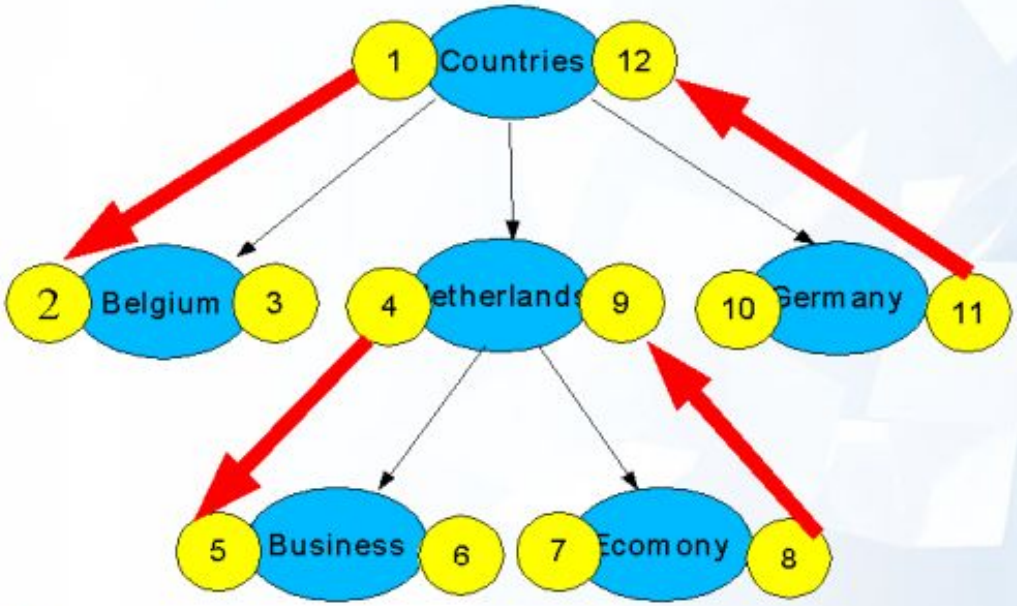
Retrieve list with:

```
<?php
function gimmeParent($id) {
    global $dir;
    $res = mysql_query("SELECT parent, name FROM dir WHERE
id=$id");
    if (mysql_num_rows($res) > 0) {
        $dir[] = ($row = mysql_fetch_row($res));
        gimmeParent($row['parent']);
    }
}
gimmeParent(7);
```

eZ publish

Visitation model 2

Visitation:



Stored as:

Id	Parent	Name	Left	Right
3	1	Countries	1	12
4	3	Belgium	2	3
5	3	Netherlands	4	9
6	3	Germany	10	11
7	5	Business	5	6
8	5	Economy	7	8

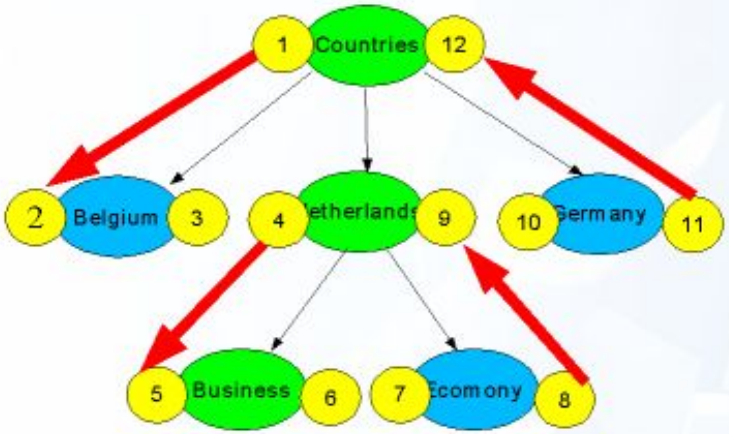
Share your information

eZ publish

Share your information

Visitation model 3

Select the path:



Query:

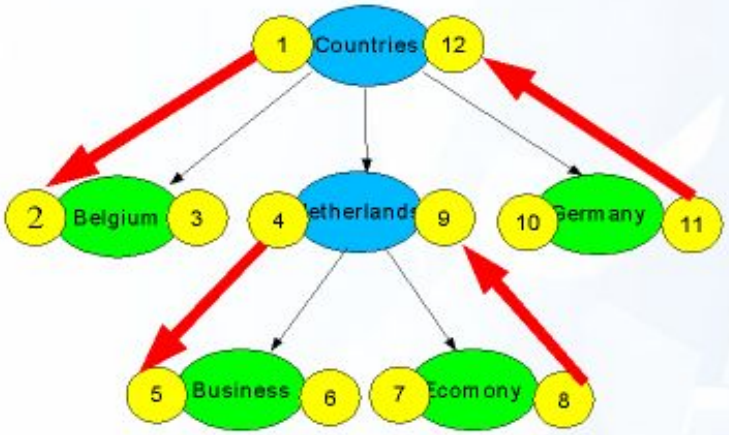
```
  SELECT * FROM directory
WHERE
    left <= 5 AND right >= 6
```

eZ publish

Share your information

Visitation model 4

Select all end points:



Query:

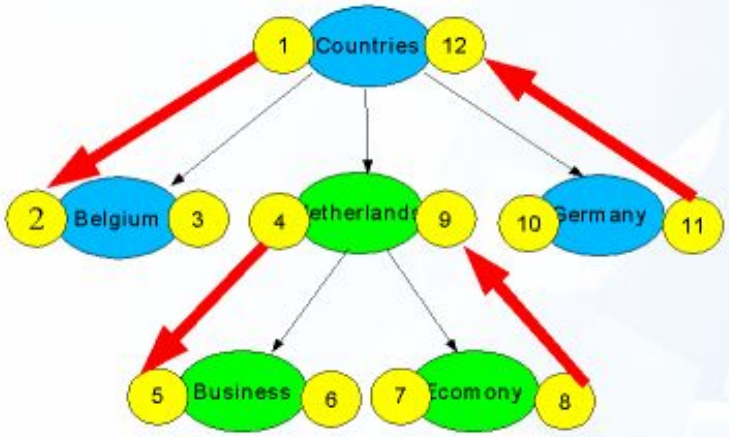
```
  SELECT * FROM directory
 WHERE
     right - left = 1
```

eZ publish

Share your information

Visitation model 5

Select a subtree:

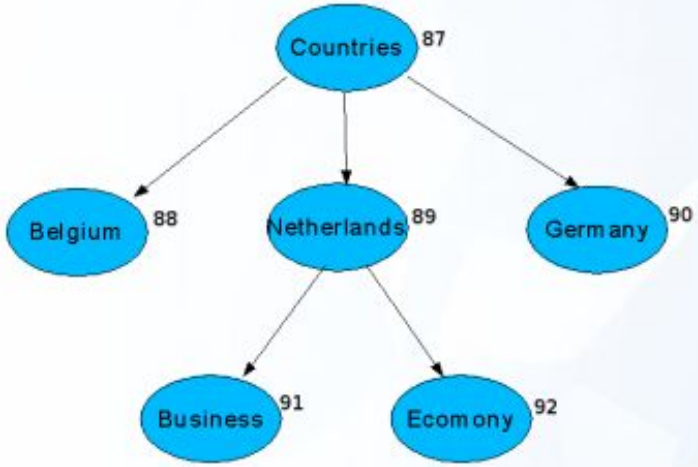


Query:

```
  SELECT * FROM directory
 WHERE
     left >= 4 AND right <= 9
```

eZ publish

Different Approach



Path strings:

Id	ParentId	PathString	Path Identification String	Depth
87	2	/87/	countries	2
88	87	/87/88/	countries/belgium	3
89	87	/87/89/	countries/netherlands	3
91	89	/87/89/91/	countries/netherlands/business	4
92	89	/87/89/92/	countries/netherlands/economy	4
90	87	/87/90/	countries/germany	3

Share your information

eZ publish

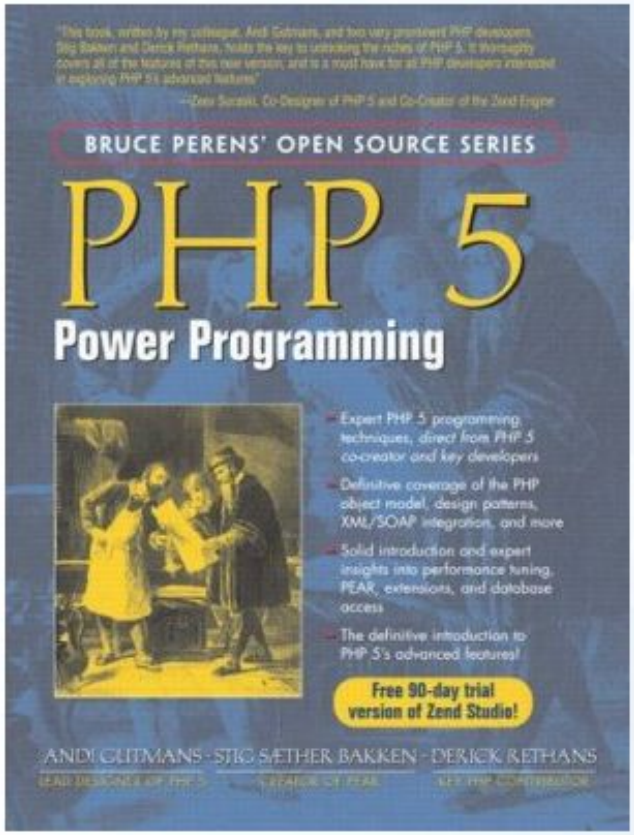
Share your information

Break

eZ publish

Share your information

PHP 5 Power Programming



- Written by Andi Gutmans, Stig Sæther Bakken and Derick Rethans
- Meant for people familiar with PHP (or another language) wanting to know how to use PHP in the most efficient way
- ISBN: 0-13-147149-X