Cache for Ca$h

Ilia Alshanetsky

Vancouver PHP Conference, Canada

Basic Fact

PHP generated content is by its nature dynamic. However, in most cases the content displayed is static or rarely changing.

How does caching help?

By caching static components of the page or even entire pages it is possible to reduce the processing time involved in generating a page.

Caching Approaches

- Complete page content caching
- Content pre-generation
- On-Demand caching
- Partial page caching
- SQL query caching
- Browser caching

"Blog" Scenario

```
<?php
require "/blog.inc";
load_header();
$qry = mysql_query("SELECT  * FROM blog_entries
ORDER BY date DESC LIMIT 15");
while ($entry = mysql_fetch_assoc($qry)) {
  show_blog_entry($entry);
}
load_footer();
```

Cache Handler

```
function cache() {
  list(,$dt) = mysql_fetch_row(mysql_query("SELECT
  MAX(date) FROM blog_entries"));
  if (filemtime("/blog/cache/index.html") > $dt) {
      // cache hit
      readfile("/blog/cache/index.html");
      exit;
  }
  init_cache();
}
```

```
function init_cache() {
  ob_start();
  register_shutdown_function('write_cache');
}

function write_cache() {
  echo $data = ob_get_clean();
  $fname = "/blog/cache/index.html";
  $tmp_fname = tempnam("/blog/cache", "blog");
  file_put_contents($tmp_fname, $data);
  rename($tmp_fname, $fname);
}
```

Cache Placement

```
<?php
require “/blog.inc”;
require “/cache.inc”;
cache();
load_header();
$qry = mysql_query(“SELECT  * FROM blog_entries
ORDER BY date DESC LIMIT 15”);
while ($entry = mysql_fetch_assoc($qry)) {
  show_blog_entry($entry);
}
load_footer();
```

Quick benchmark

	Before	After
Page Size	31,117 bytes	31,117 bytes
Latency	8.73 msec.	4.62 msec.
Req./Second	451	665

Serendipity Test

	Before	After
Page Size	25,853 bytes	25,853 bytes
Latency	199.3 msec.	42.37 msec.
Req./Second	24.7	112.2

Compressed Page Cache

```
if (!empty($_SERVER["HTTP_ACCEPT_ENCODING"])
&& strpos($_SERVER["HTTP_ACCEPT_ENCODING"],
'gzip') !== false) {
  header('Content-Encoding: gzip');
  readfile("/tmp/index.html.gz");
} else {
  readfile(“ /blog/cache/index.html”);
}
```

```
function write_cache() {
  echo $data = ob_get_clean();
  $fname = "/blog/cache/index.html";
  $tmp_fname = tempnam("/blog/cache", "blog");
  file_put_contents($tmp_fname, $data);
  rename($tmp_fname, $fname);
  $tmp_name = tempnam("/blog/cache", "blog");
  copy("/tmp/index.html",
    "compress.zlib://" . $tmp_name);
  rename($tmp_name, "/tmp/index.html.gz");
}
```

Content Pre-generation

- Simplifies caching code
  - No real-time work
  - Entire cache can be easily rebuilt
  - Can be used to bypass PHP entirely
  - Predictable operation

Let's Avoid PHP

- Concept
  - Make product pages be .html pages
  - Generate .html pages via PHP
  - Static pages are served by the web server, skipping the PHP overhead.

```
function content_generate($s, $e)
{
  while ($s <= $e) {
     $url = "http://store.com/product.php?id=".$s;
     $file = "/cache/product_".$s;

     $data = file_get_contents($url);
     file_put_contents($file, $data);

     copy($file, "compress.zlib://" . $file . ".gz");
     $s++;
  }
}
```

Livedocs Test

	Before	After
Page Size	30,896 bytes	30,896 bytes
Latency	611.2 msec.	2.37 msec.
Req./Second	5.29	1284.6

Pre-Generation Problems

- Disk utilization
  - 2 files per-page X number of pages
- Generates pages no one may visit
- Generating all pages at once may take too long.

On Demand Caching

- Advantages
  - Generate what you need, when you need it.
  - Simple to setup
  - Has all the speed advantages of pre-generation.

404 Handler

- Concept
  - Make all links lead to .html pages
  - When page does not exist, have PHP 404 handler be triggered.
  - The handler script generates the missing page
  - All future visitors access static content.

ErrorDocument 404 /generate_html.php

```
// sample URL http://site.com/products/123.html

$id = (int) basename($_SERVER['REDIRECT_URL']);

if (!$id || $id < 0) { exit; }

$url = "http://store.com/product.php?id=".$id;
$file = "./products/".$id."".html";

$data = file_get_contents($url);
file_put_contents($file, $data, LOCK_EX);
copy($file, "compress.zlib://" . $file . ".gz");
```

CACHING QUANDARY

My pages always have an ever changing content and cannot be cached in their entirety.

amazon.com | Your Store | See All 35 Product Categories | Your Account | Cart | Your Lists | Help

Gift Certificates | International | New Releases | Top Sellers | Today'S Deals | Sell Your Stuff

Search Amazon.com GO Find Gifts Web Search GO

Hello. Sign in to get personalized recommendations. New customer? Start here.

Browse

Books, Music & Movies
- Books
- DVD
- Magazines & Newspapers
- Music
- Textbooks
- Unbox Video Downloads
- VHS

Clothing & Accessories
- Apparel & Accessories
- Jewelry & Watches
- Shoes

Computer & Office
- Computers
- Office Products
- Software

Consumer Electronics
- Audio & Video
- Camera & Photo
- Cell Phones & Service
- Computer & Video Games
- Musical Instruments
- All Consumer Electronics

Shopping from Canada?

Magasinez-vous depuis le Canada ?

Visit | Visitez amazon.ca

Shop now

INTRODUCING amazonunbox™ Video Downloads

Try it now with a FREE TV show!*

*restrictions apply

Ergonomic Massage Cushion

The contoured, orthopedic design relieves tension and weariness for lower back fatigue & with 2 D-cell batteries (not included) operated massage mode... Find out more

› Find similar items

Life Is Good at Amazon.com

Check out our wide selection of fun and quirky Life Is Good T-shirts and hats at Amazon.com.

› Shop now

Save up to $50--Get Your Home and Lawn Projects Done Now

Use promotional code ALLTOOLS at checkout to get instant discounts of $25 off $150 or $50 off $300 or more on Tools, Lawn & Garden, or Home Improvement products. Don't get left out in the cold. Save now on

- Power tools, hand tools, and accessories
- Lawn & Garden products

MP3 Player Perfection?

Are the new Sansa MP3 players really as good as everyone says? With FM radios, bigger screens, and longer battery life, they're mighty impressive.

*Dynamic Content

**Semi-Dynamic Content

APC to the Rescue!

Alternative PHP Cache (APC) is an open source opcode cache. It happens to provide handy PHP variable storage mechanism we can use.

APC API

- apc_store([key], [variable], [time to live])
  - key - cache entry identifier
  - variable - PHP variable to store
  - time to live - for long to store the data
- [variable] = apc_fetch([key])
- apc_delete([key])

```
if (!($promo = apc_fetch('promo'))) {
  $promo = fetch_all_promos();
  apc_store('promo', $promo, 3600); // 1 hour
}

if (!($deals = apc_fetch('deal'))) {
  $deals = fetch_all_deals();
  apc_store('promo', $promo, 600); // 10 mins
}

if (!($final_tmpl = apc_fetch('main_template'))) {
  $final_tmpl = generate_front_page();
  apc_store('main_template', $final_tmpl, 86400);
}                                  // 1 day
```

```
// fetch one promo & deal to display
$data[0] = $promo[array_rand($promo)];
$data[1] = $deals[array_rand($deals)];
// get the always dynamic sections
$data[2] = get_shopping_cart();
$data[3] = generate_greeting_msg();
// in the finalized page substitute semi-static content
echo str_replace(
  array(
    '%%deals%%', '%%promo%%',
    '%%cart%%', '%%greeting%%'
  ), $data, $final_tmpl);
```

SQL Caching

- In most applications the primary bottleneck can often be traced to "database work".

- Caching of SQL can drastically reduce the load caused by unavoidable, complex queries.

Cache Tables

```
CREATE TABLE search_cache
(
  skey      CHAR(32) PRIMARY KEY,
  rec_id    INT NOT NULL,
  age       INT UNSIGNED,

  INDEX(skey),
  INDEX(age)
);
```

Basic Search Cache

```
$key = md5("user supplied search query");
$tm = time(); $max_age = $tm - 3600 * 4; // 4h

$qry = "SELECT count(*) FROM search_cache
WHERE key='{$key}' AND age>".$max_age;

if (!$pdo->query($qry)->fetchColumn()) {
  // cache generation code is here
}
// Standard join against search_cache table to get
// the relevant records
```

Cache Generation Code

```
// clear old cache entries
$pdo->exec("DELETE FROM search_cache
      WHERE key='{$key}'
      AND age<=".$max_age);

$qry  = "INSERT INTO search_cache";
$qry .= " (key,age,rec_id) ";
$qry .= " SELECT '{$key}', {$tm}, rec_id";
$qry .= " FROM [search qry] LIMIT 1000";

$pdo->exec($qry);
```

On-Disk SQL Caching

```
$file = "./qry.cache"
if (filemtime($file) > time() - 600) {
   $res = unserialize(file_get_contents($file));
} else {
   $res = $pdo->query($qry)->fetchAll
   (PDO::FETCH_ASSOC);

   file_put_contents($file, serialize($res), LOCK_EX);
}
```

In-memory SQL Cache

```
$key = crc32($qry);
$s = shmop_open($key, 'a');
$age =  (int)shmop_read($s, 0, 10);
if ($age > $max_age) {
    $size = shmop_size($s) - 10;
    $res  = unserialize(shmop_read($s, 10, $size));
} else {
  // cache miss
}
```

```
$res = $pdo->query($qry)->fetchAll
(PDO::FETCH_ASSOC);

// obtain lock
$l = shmop_open($key, 'n', 0600, 1);
if ($l) { // got lock
  $val  = str_pad(time(), 10, '0', STR_PAD_LEFT);
  $val .= serialize($res);
  shmop_write($s, $val, 0); // write new data
  shmop_close($l); // release lock
}
shmop_close($s);
```

Browser Cache

- Advantages
  - Reduces data sent to virtually zero
  - Requires next to no server resources
- Disadvantages
  - Flimsy control over cache age
  - Not guaranteed to work

How to implement?

```
// Time to Cache
header("Expires: ".date(DATE_RFC822));

// Based on modification date
header("Last-Modified: ".date(DATE_RFC822));

// Content hash
header("Etag: ".md5_file("./page_cache"));
```

Thank You For Listening

- Resources
  - These slides: http://ilia.ws/
  - APC - http://pecl.php.net/apc/
  - SHMOP - http://php.net/shmop
  - http://pear.php.net/cache