

MAGENTO PERFORMANCE OPTIMIZATION

Get Optimal Performance from
Magento with Zend Server

July 2009



The PHP Company

Copyright © 2007, Zend Technologies, Inc.

Who We Are



What is Magento?

Magento is:

Developed on the
Zend Framework
(PHP5).

An open-source,
feature-rich
ecommerce platform.

First stable release
March 2008.

Flexible

Extendable

Modular

Upgradable

Scalable

Magento Flavors

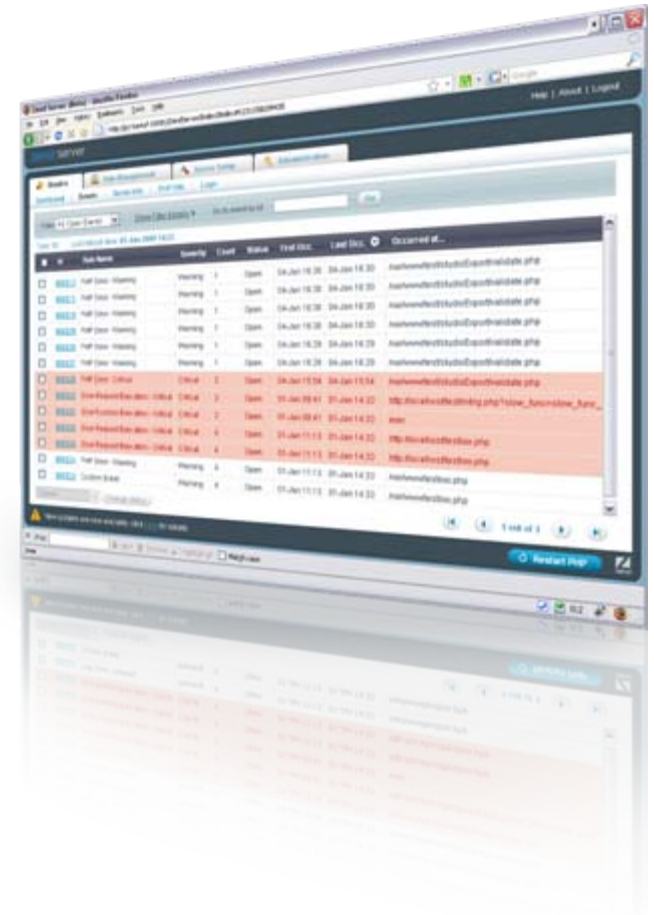
- **Magento Community Edition (CE)**
 - Free to download and use (<http://www.magentocommerce.com>).

- **Magento Enterprise Edition (EE)**
 - Annual Subscription Version
 - World Class Support, Warranty, and Indemnification
 - Additional Feature Set
 - <http://www.magentocommerce.com/product/enterprise-edition>

What is Zend Server?

- Production-ready, complete PHP application stack
- Application monitoring and problem diagnostics
- Multi-level performance enhancement capabilities
- Technical support, software updates and security hot fixes

www.zend.com/server



Zend Server Flavors

Zend Server Community Edition (CE)

- Free to download and use
- A certified PHP distribution with native installers
- Web based PHP Admin console (Apache/IIS)
- Debugging
- Out-of-box connectivity to all common databases (MySQL, Oracle, DB2, MSSQL,..)
- Performance optimization (data caching and bytecode acceleration)
- Linux, Windows, Mac OS X (only supported in CE)

Zend Server

Includes everything in CE plus the following:

- World-class technical support
- Native Installers, updaters and security patches
- Web-based application monitoring and alerting
- Problem reproduction and root cause analysis
- Page caching and JavaScript caching for extra performance improvement



Getting Best Performance from Magento



Zend Server Opcode Acceleration (Optimizer+)



Magento Native Caching



Database Access Optimization



Magento Compilation Module



Zend Server Page Caching

Opcode Acceleration (Optimizer+)

- Caching compiled PHP opcode in shared memory
- Automatically enabled when you run any application on Zend Server
- Usually provides a very significant performance boost



Magento Native Caching

- Magento uses Zend_Cache (Zend Framework component) for data caching
- Supported backend models:
 - File system
 - APC
 - eAccelerator
 - Memcached
 - Zend Server
 - Zend Platform (coming...)
 - xCache (coming...)
- File system backend model used by default
- Cache backend model can be specified in configuration (app/etc/local.xml file)
- Cache configuration examples can be found in app/etc/local.xml.additional file

Magento Native Caching

```
<config>
  <global>
    <cache>
      <backend></backend><!-- apc /
memcached /eaccelerator/ empty=file -->
    </cache>
  </global>
</config>
```

Magento Native Caching

- Cache back-ends can be classified into two types:
 - Shared Memory Storage
 - File System Storage
- Shared memory storage is faster
- APC and eAccelerator can be used for deployments a single server
- Memcached is good solution for multiple web servers installations (cluster)

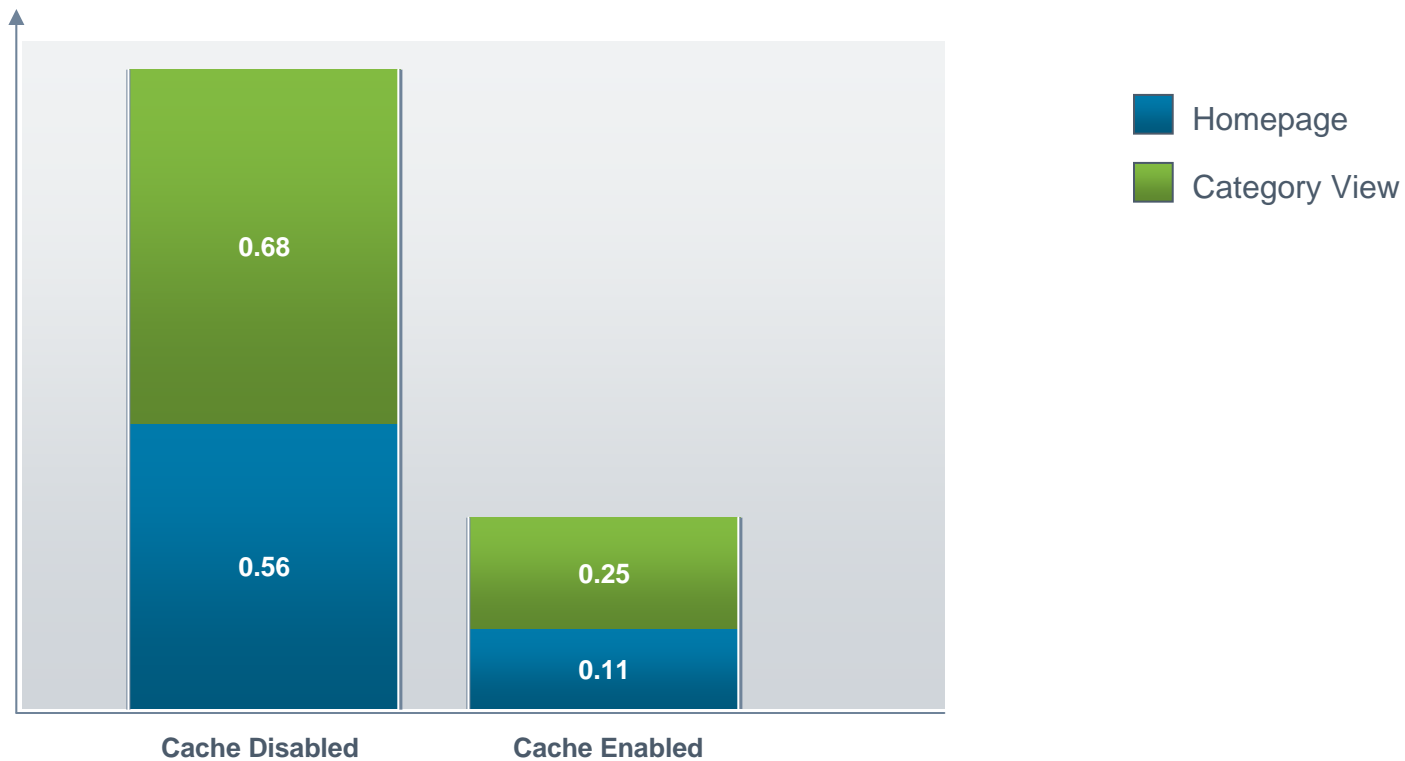
Magento Native Caching

Major Cached Entities:

- Configuration
- Page Layouts
- HTML Blocks (Top Navigation, Footer and plan to extend this to other HTML Blocks)
- Translations (each local creating an array of final translation and stored in cache)
- Data Collections (Websites Collection, Store Collection and Store View Collection will add more for example catalog)

Performance Improvement with Native Caching

Cache Disabled/Enabled



Database Read/Write Configuration

To configure read/write connections we need to modify app/etc/local.xml file:

```
<default_setup>
  <connection>
    <host><![CDATA[host]]></host>
    <username><![CDATA[user]]></username>
    <password><![CDATA[pass]]></password>
    <dbname><![CDATA[magento]]></dbname>
    <active>1</active>
  </connection>
</default_setup>
<default_read>
  <connection>
    <use></use>
    <host><![CDATA[readhost]]></host>
    <username><![CDATA[readuser]]></username>
    <password><![CDATA[readpass]]></password>
    <dbname><![CDATA[magento]]></dbname>
    <model>mysql4</model>
    <initStatements>SET NAMES utf8</initStatements>
    <type>pdo_mysql</type>
    <active>1</active>
  </connection>
</default_read>
```

Magento Compilation Module

- Goal: Decrease number of file system operations (I/O). Still in Beta!
- By default Magento uses 4 additional directories in `include_path` in the following order:
 - `app/code/local`
 - `app/code/community`
 - `app/code/core`
 - `lib`
- This schema provides extensibility and flexibility and allows to fully rewrite code of some classes but yet keeps upgradability.

Magento Compilation Module: Process

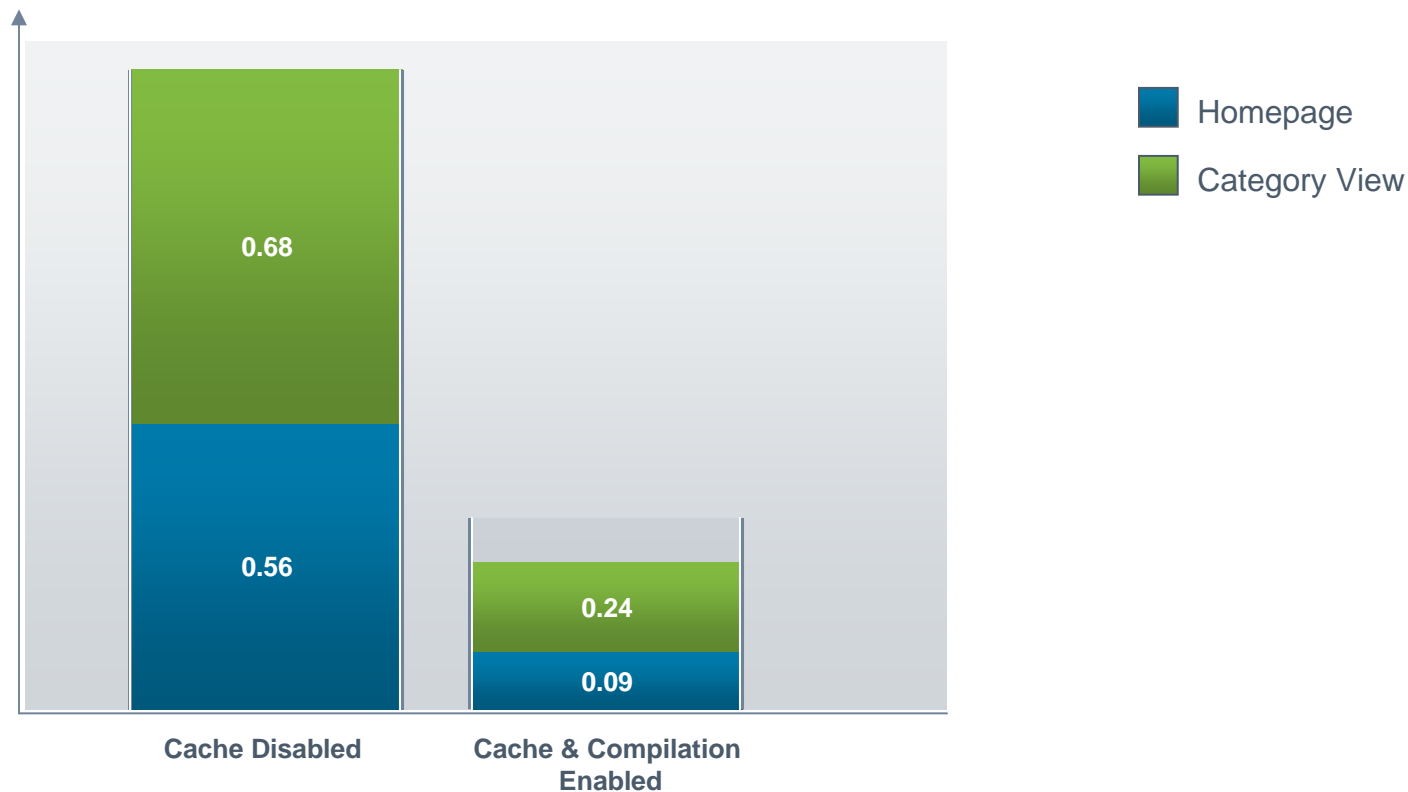
- Copy all php files from the four folders to one folder (include/src) and declare one directory for php include_path instead of the other four. This step give 10-25% improvement in performance.
- Aggregate code of most used php classes to “scope include” files.

For example Varien_Object, Mage_Core_Model_Abstract, Zend_Cache etc, are classes that are included in every request. The classes are aggregated to the “default scope include” file. So now only one file will have to be included for all these classes. The same is done for other scopes like checkout, catalog, cms etc.

* Compilation Needs to be disabled when upgrading or developing and should be enabled in production

Performance Improvement with Native Compilation

Compilation Disabled/Enabled



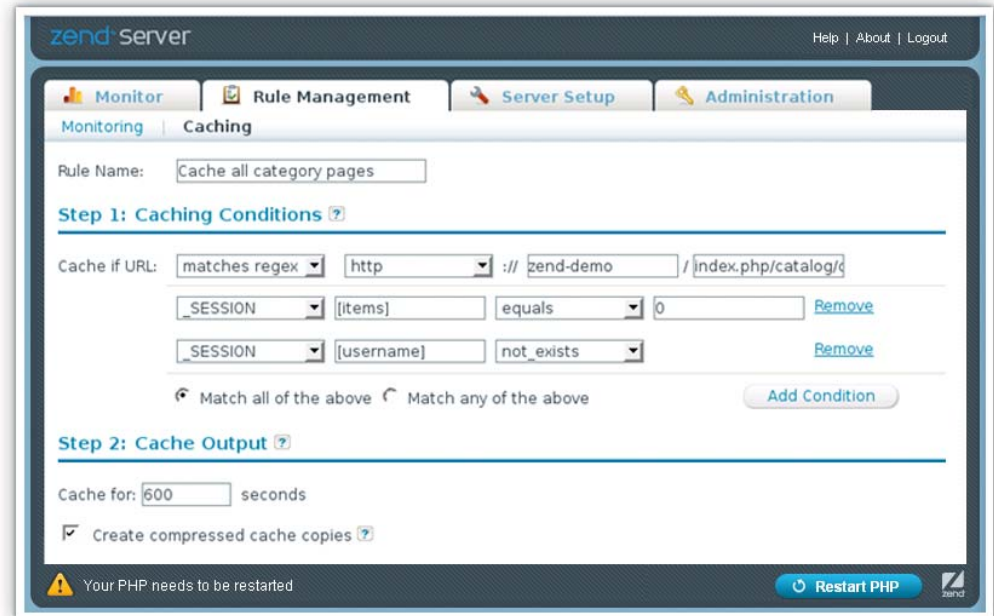
Diagnosing Performance Problems

- Built-in application monitoring alerts on script execution slowdowns
- Problem diagnostics enables quick problem isolation
- Code-level profiling and analysis performed in Zend Studio



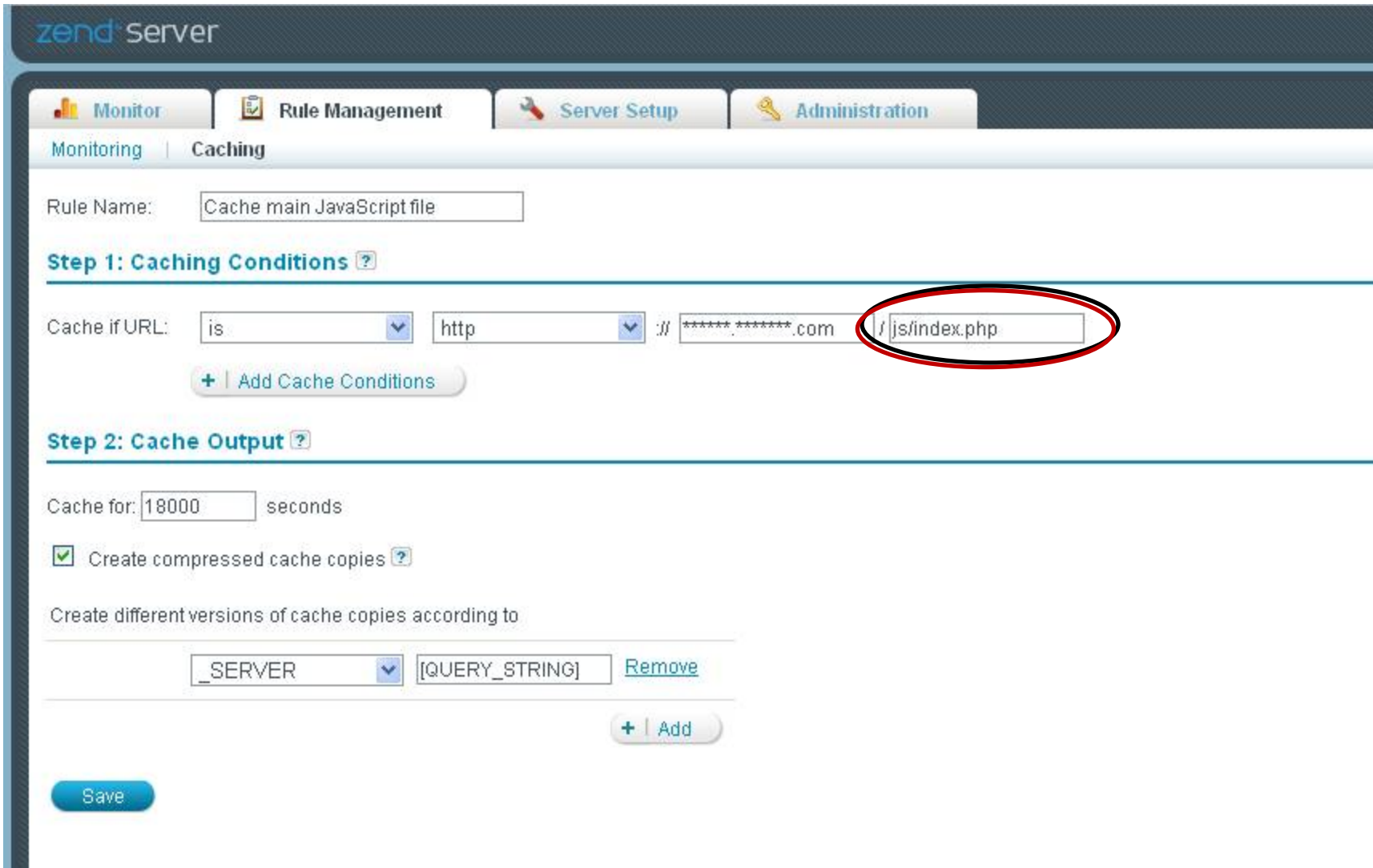
Zend Server Page Caching

- Caches entire HTTP responses
- Performance gains are usually **very** high
- Typically doesn't require code changes



May require code changes for pages where user-specific information (e.g. shopping cart, recent purchases list) is used

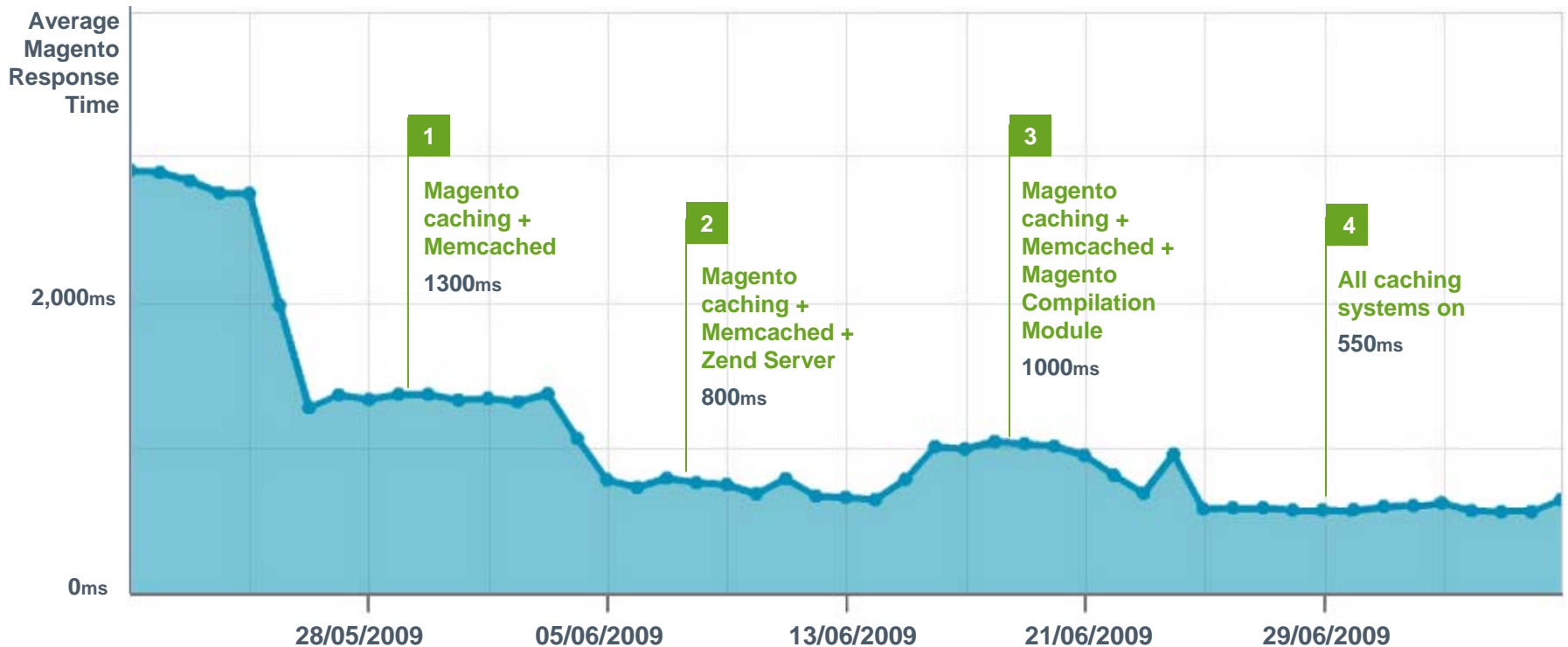
“Quick Win”: Cache /js/index.php



The screenshot shows the Zend Server administration interface for configuring a caching rule. The interface is divided into several sections:

- Navigation:** Monitor, Rule Management, Server Setup, Administration.
- Monitoring | Caching:** The active section.
- Rule Name:** Cache main JavaScript file
- Step 1: Caching Conditions:**
 - Cache if URL: is http :// *****.*****.com **/js/index.php**
 - + Add Cache Conditions
- Step 2: Cache Output:**
 - Cache for: 18000 seconds
 - Create compressed cache copies
 - Create different versions of cache copies according to:
 - [Remove](#)
 - + Add
- Save**

Real-Life Results: Zend's Online Store Project



Summary



+



= Optimal User Experience

What Do I Do Now?

- Download Magento
<http://www.magentocommerce.com/>
- Download Zend Server Community Edition
<http://www.zend.com/server>



Q&A