

Improve Web Application Performance with Zend Platform

Shahar Evron

Zend Sr. PHP Specialist



The PHP Company

Copyright © 2007 Zend Technologies, Inc.

Agenda

- **Benchmark Setup**
- **Comprehensive Performance**
 - Multilayered Caching
 - Code Acceleration
 - Output Compression
 - Architecture Optimization
- **Performance Tuning**

Powering the Enterprise – Zend Platform



Performance

- Code Acceleration
- Multi-Layer Caching
 - Client-Side
 - Caching API
 - Page Cache
- Architecture Considerations
 - Download Optimization
 - Asynchronous Processing



Enterprise Reliability

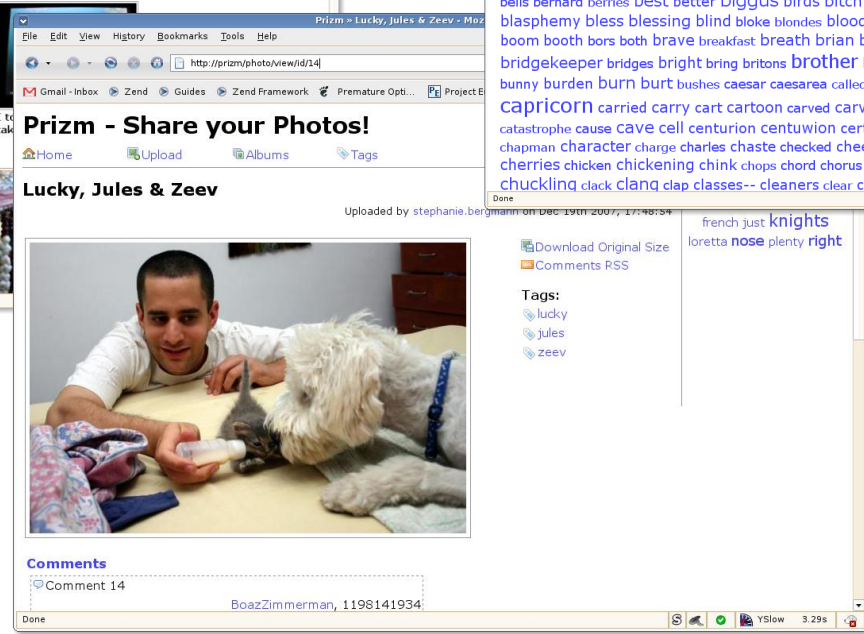
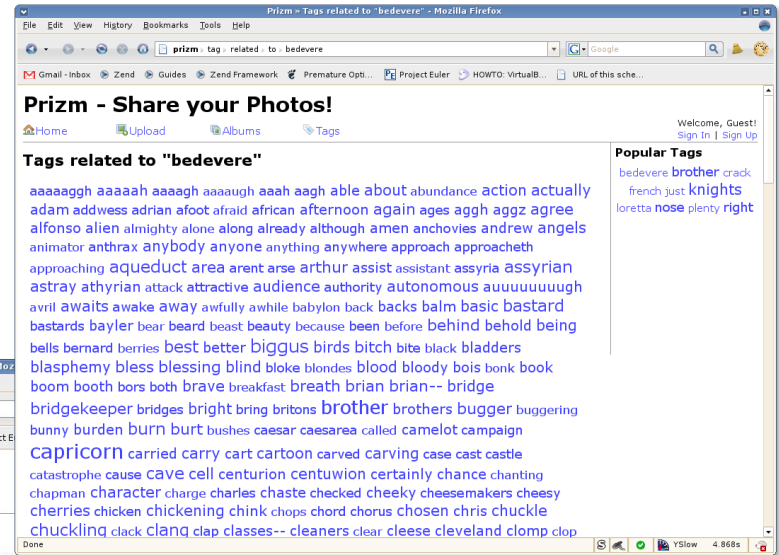
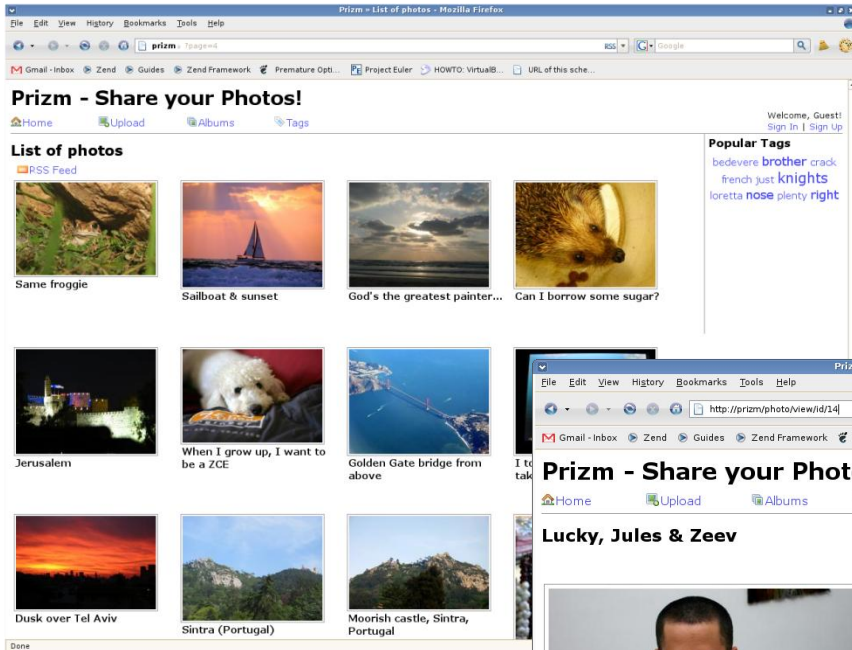
High-Availability & Scalability

- Session Clustering
- Event Consolidation
- Clone Configurations

Monitoring & Performance Tuning

- Detect, Identify and Tune
- Run-time Performance Tuning

Prizm – Application Overview



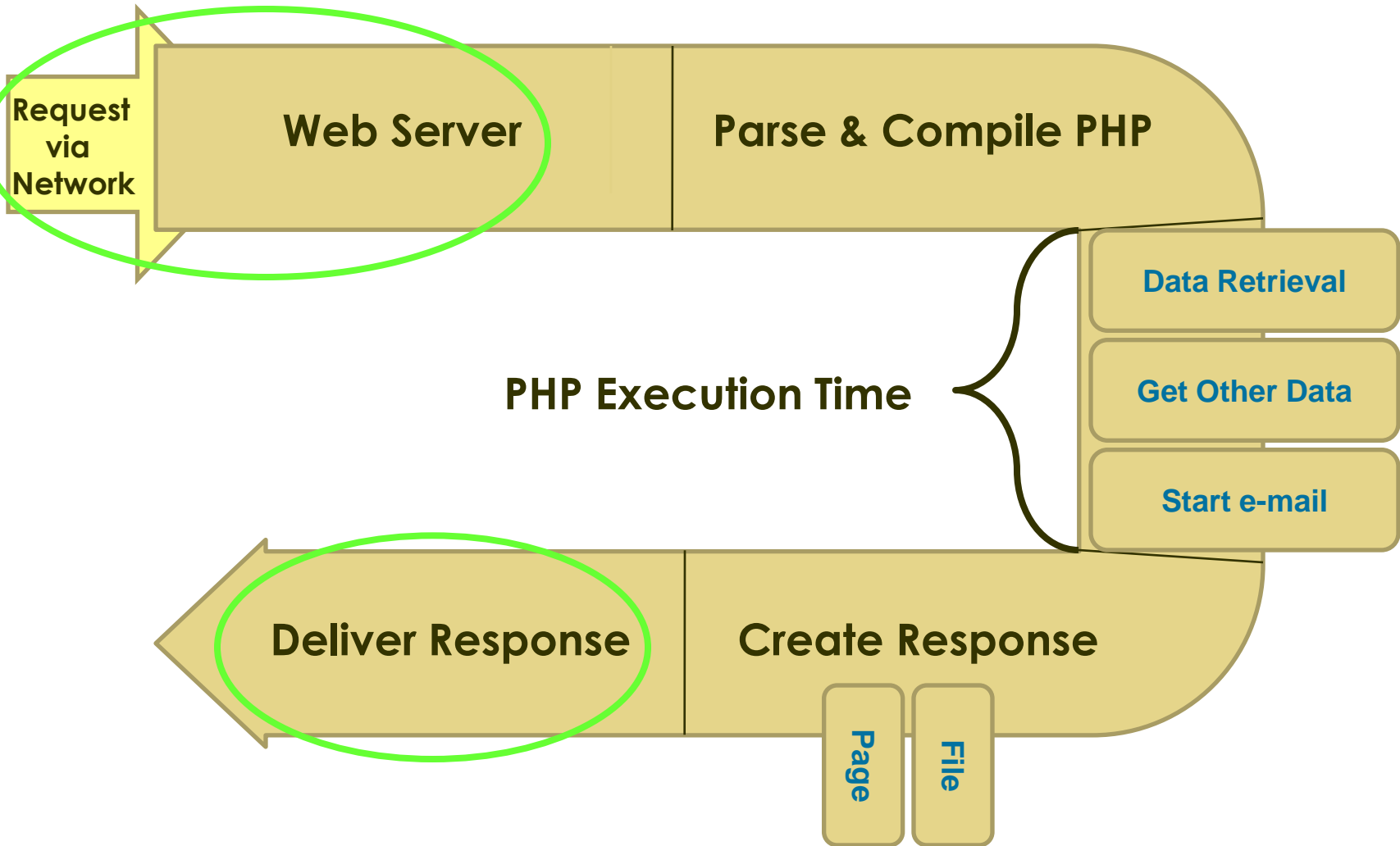
Benchmark Setup

- **Common Web Environment Setup**
 - Debian Linux based, average web servers
 - Powerful load-generation machine
 - Dedicated performance lab, on a dedicated LAN segment
- **Common Benchmarking Tools**
 - Apache Bench (“ab”) for very simple benchmarks
 - Apache JMeter for the more complex ones
- **A series of separate benchmarks to isolate the potential gain of different performance features**

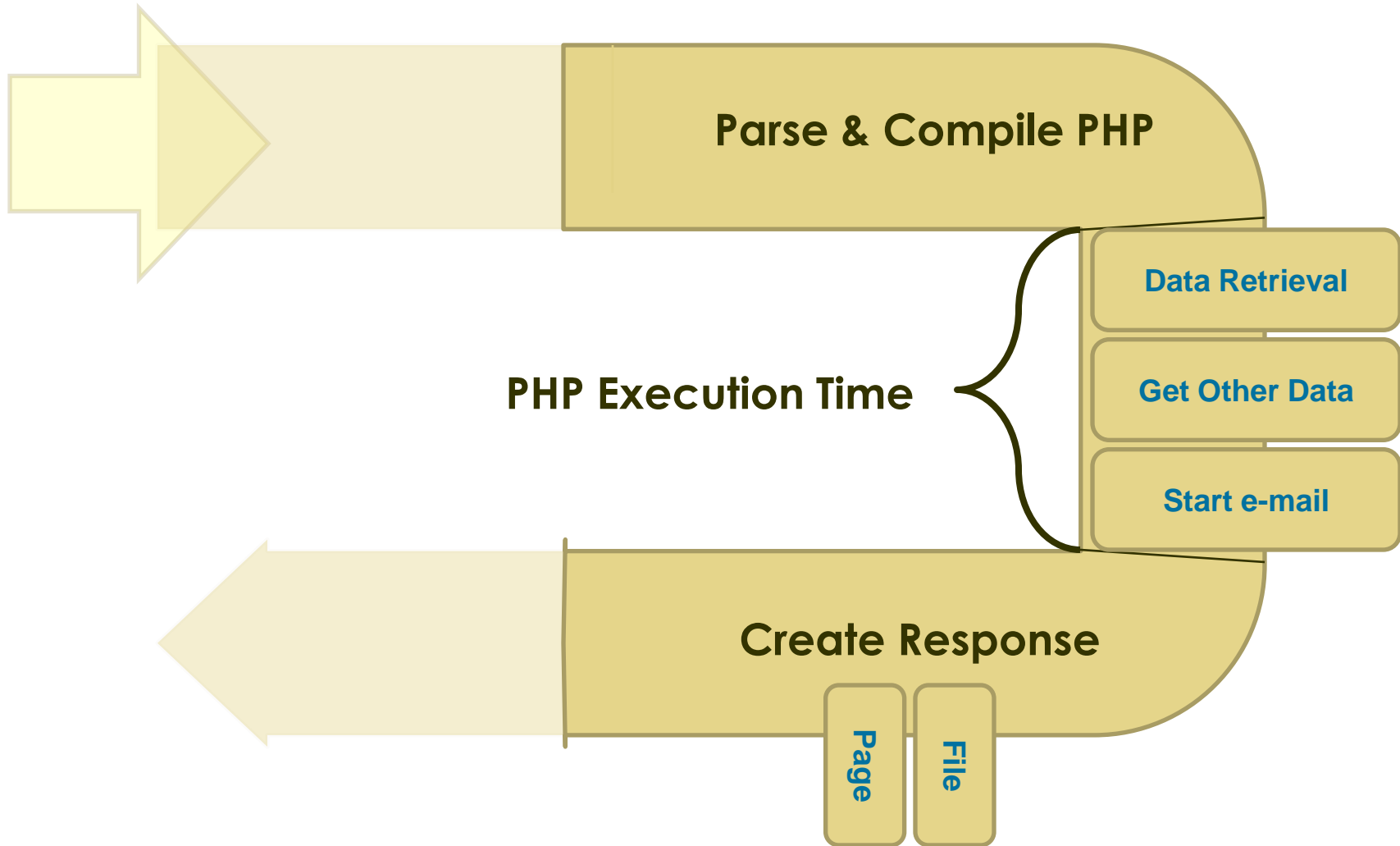
Comprehensive Performance

- **Multilayered Caching**
Multiple levels of caching capability
- **Code Acceleration**
Ability to bypass the time consuming PHP compilation phase
- **Output Compression**
Compressing the entire content sent in the response
- **Architecture Considerations**
Asynchronous processing and dedicated download

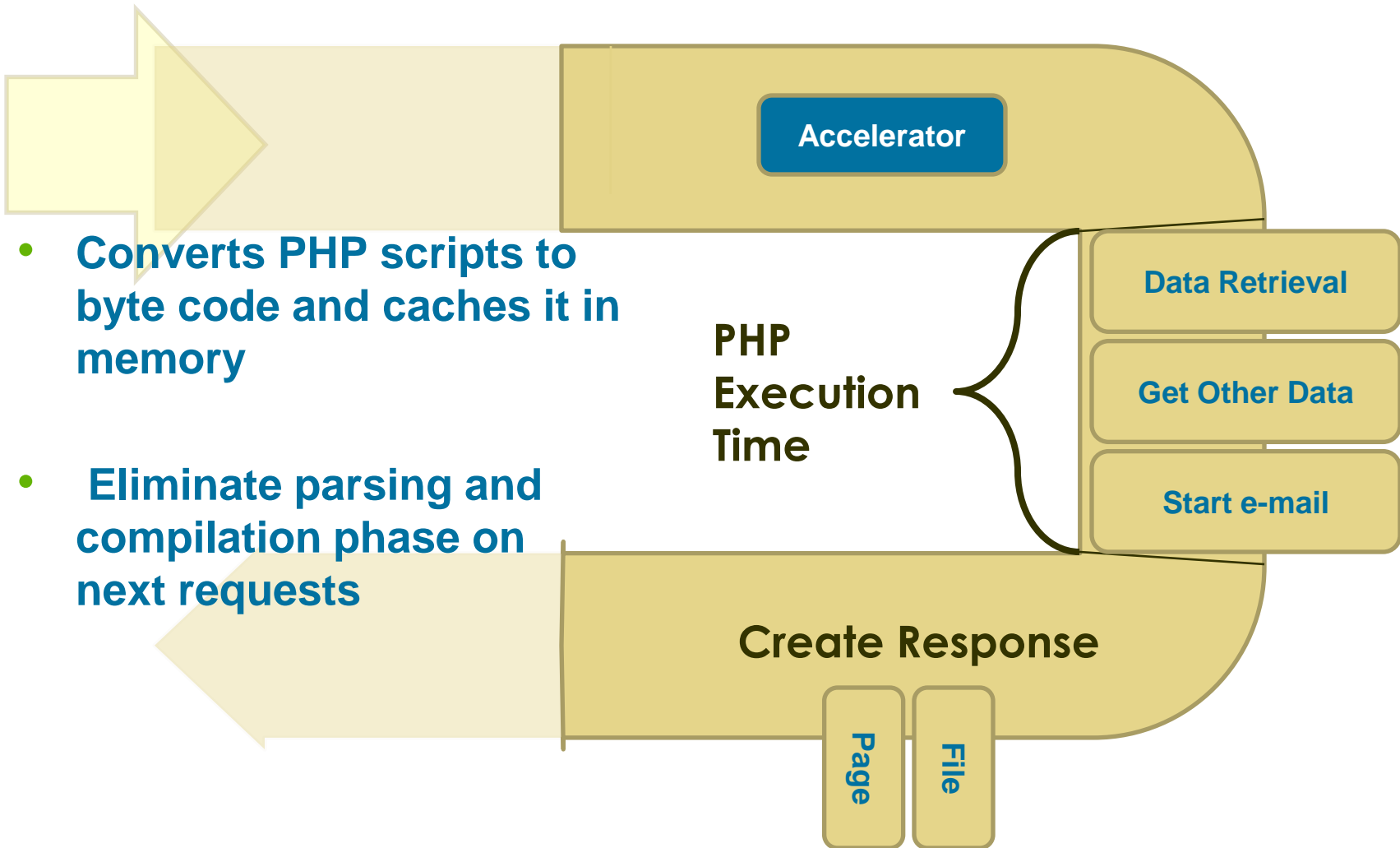
Optimize Entire Request-Response Cycle



Code Acceleration – Double Performance

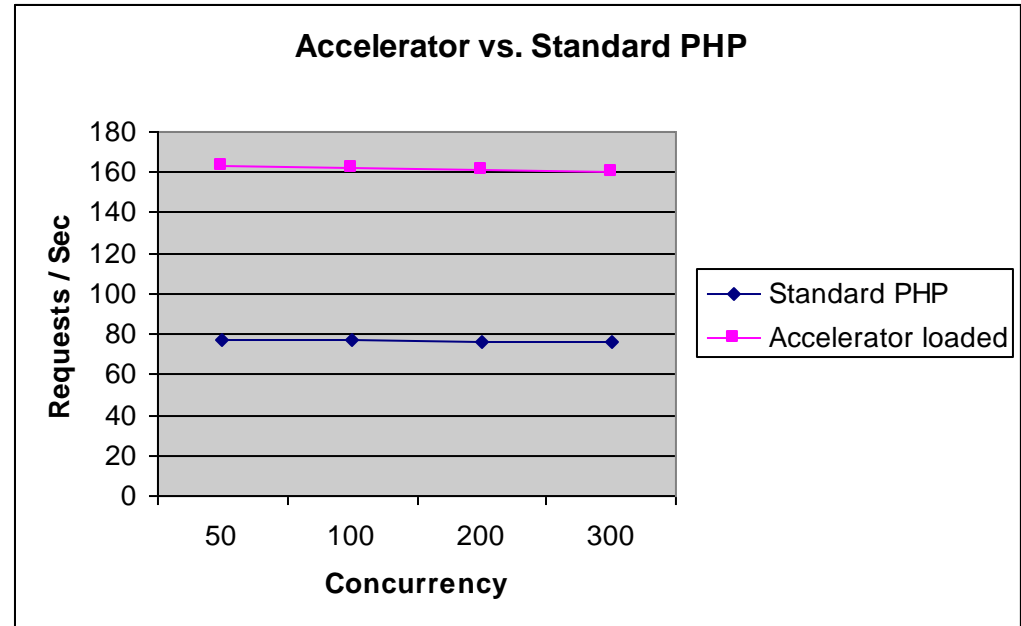


Code Acceleration – Double Performance



110 % Average increase in request/sec

- Works transparently
- Requires no code changes
- Can be used on any PHP script
- Blacklist files if appropriate



Zend Platform ENTERPRISE SERVER

Platform | PHP Intelligence | Performance | Configuration | Session Clustering | Job Q

Console | Settings | File View | Testing | Tuning

Settings Save Clone Settings Help

- Reset Dynamic Content Caching
- Reset Code Accelerator
- Update Virtual Hosts list

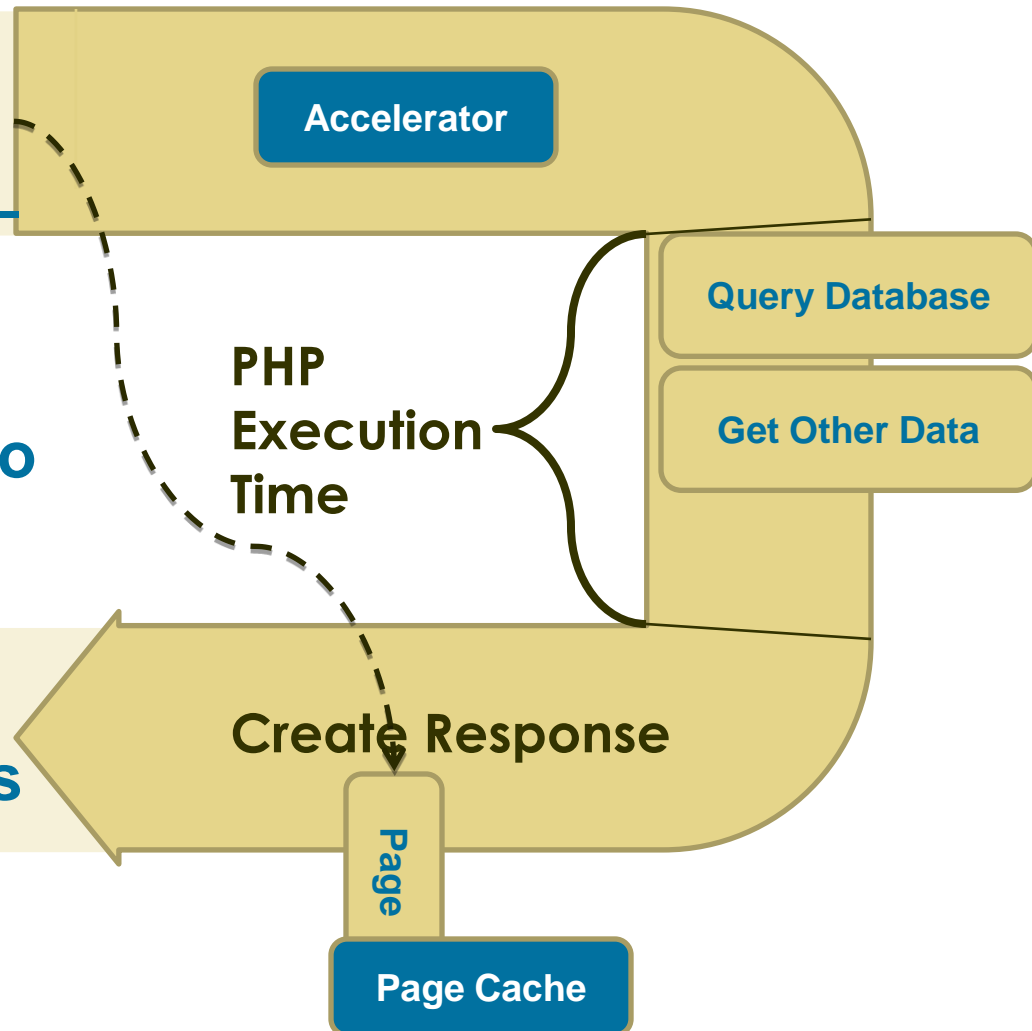
Code Acceleration		Current Settings	New Settings
<input checked="" type="checkbox"/> Code Acceleration Enabled	On		<input checked="" type="radio"/> On <input type="radio"/> Off
Accelerator Memory	64 MB	In use:	<input type="text" value="64"/> MB
Memory Reclaim Threshold	5%		<input type="text" value="5"/> %
Maximum Accelerated Files	2000	In use:	<input type="text" value="2000"/>
Extensions For PHP Files	php		<input type="text" value="php"/>

Multilayer Caching

- Page Caching
 - Cache the entire output of a PHP script
 - Configurable from GUI – no code change required
- Client-Side
 - Reduce bandwidth consumption and network latency
- Data Caching API
 - Cache data items or parts of a script's output through a set of flexible and easy to apply APIs

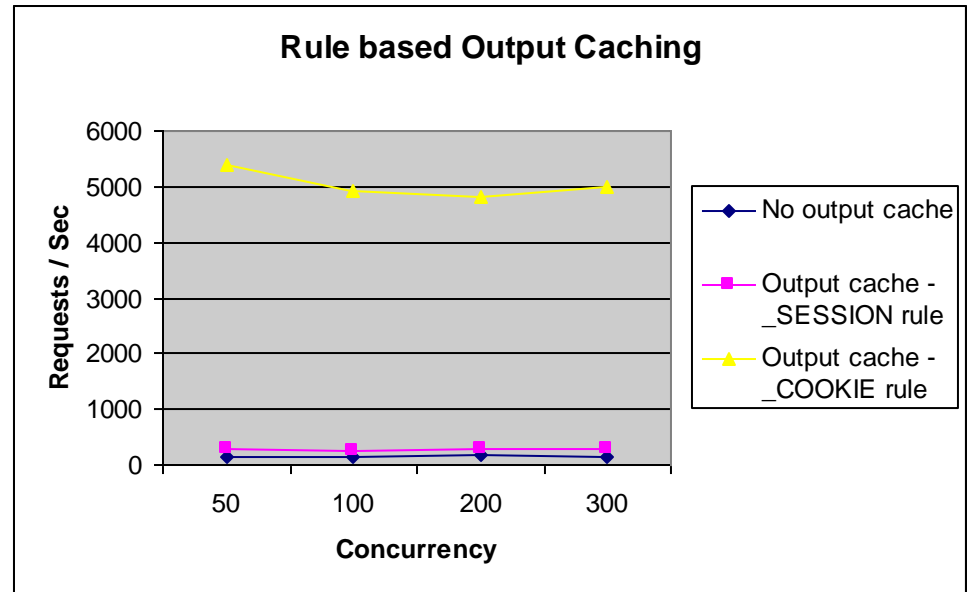
Multilayer Caching – Page Caching

- Caches the full response on disk – eliminates entire execution time
- GUI configured - no code changes required
- Supports flexible caching conditions and rules



Multilayer Caching – Page Caching

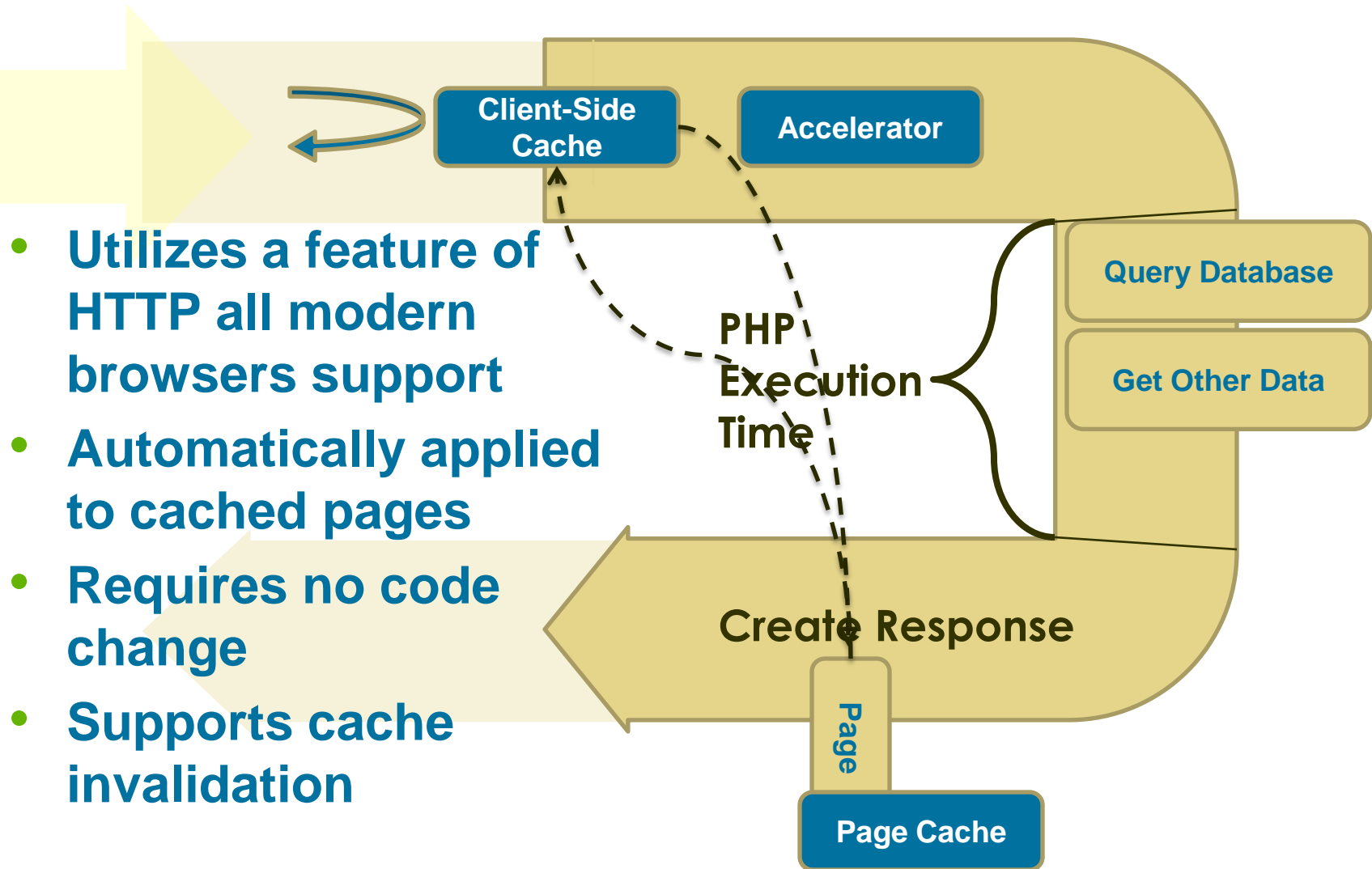
- Improvement 80% - 3,000% depending on implementation
- Can be used on any PHP script
- User need to select files or URLs to be cached



The screenshot shows the 'Dynamic Content Caching' settings in the Zend Platform. The 'Dynamic Caching Enabled' checkbox is checked. The 'Maximum Cache Size' is set to 'Unlimited'. The 'Minimum Free Diskspace' is 159 MB. The 'Maximum Cached File Size' is 500 KB. The 'Default Cache Lifetime' is 360 Seconds. The 'Default Dynamic Caching Conditions' are set to 'ALLGET'. The 'Extensions For PHP Files' is set to 'php'.

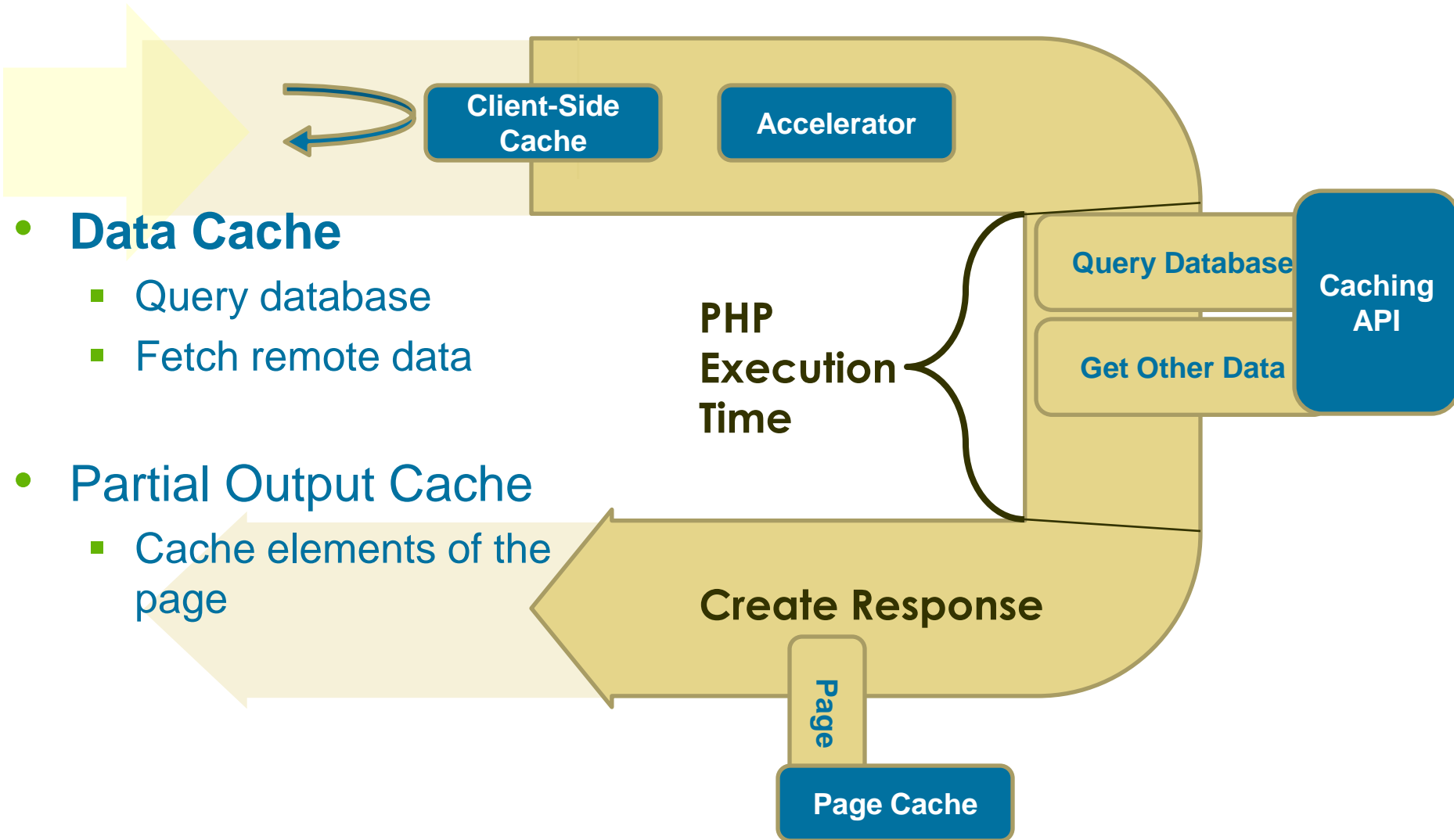
Setting	Current Settings	New Settings
Code Acceleration	On	On
Code Acceleration Memory	64 MB	64 MB
Memory Reclaim Threshold	5%	5%
Extensions For PHP Files	php	php
Dynamic Caching Enabled	On	On
Maximum Cache Size	Unlimited	0 MB
Minimum Free Diskspace	159 MB	159 MB
Maximum Cached File Size	500 KB	500 KB
Default Cache Lifetime	360 Seconds	360 Seconds
Default Dynamic Caching Conditions	ALLGET	Change Default Conditions

Multilayer Caching – Client Side Cache



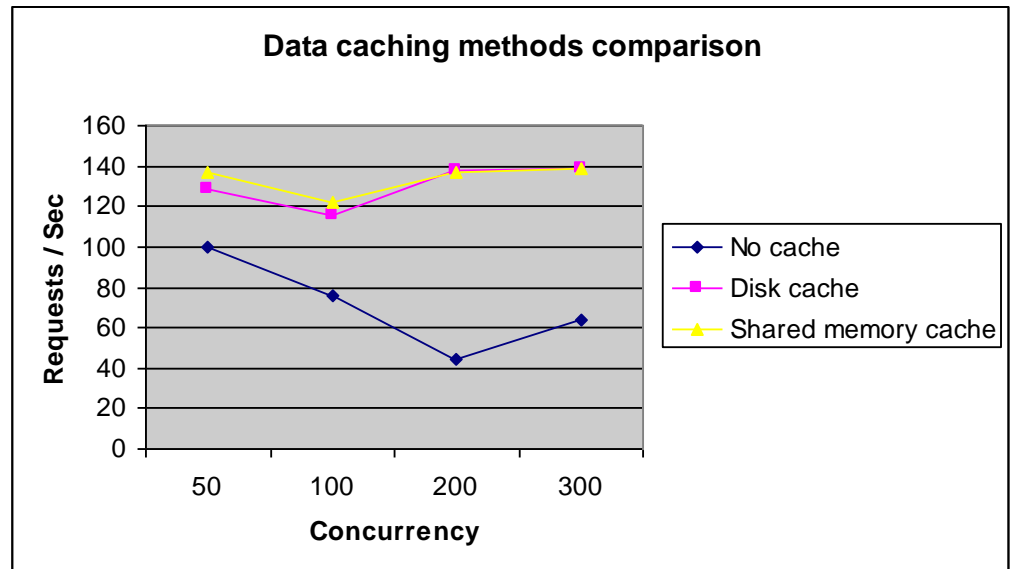
- Utilizes a feature of HTTP all modern browsers support
- Automatically applied to cached pages
- Requires no code change
- Supports cache invalidation

Multilayer Caching – Caching API



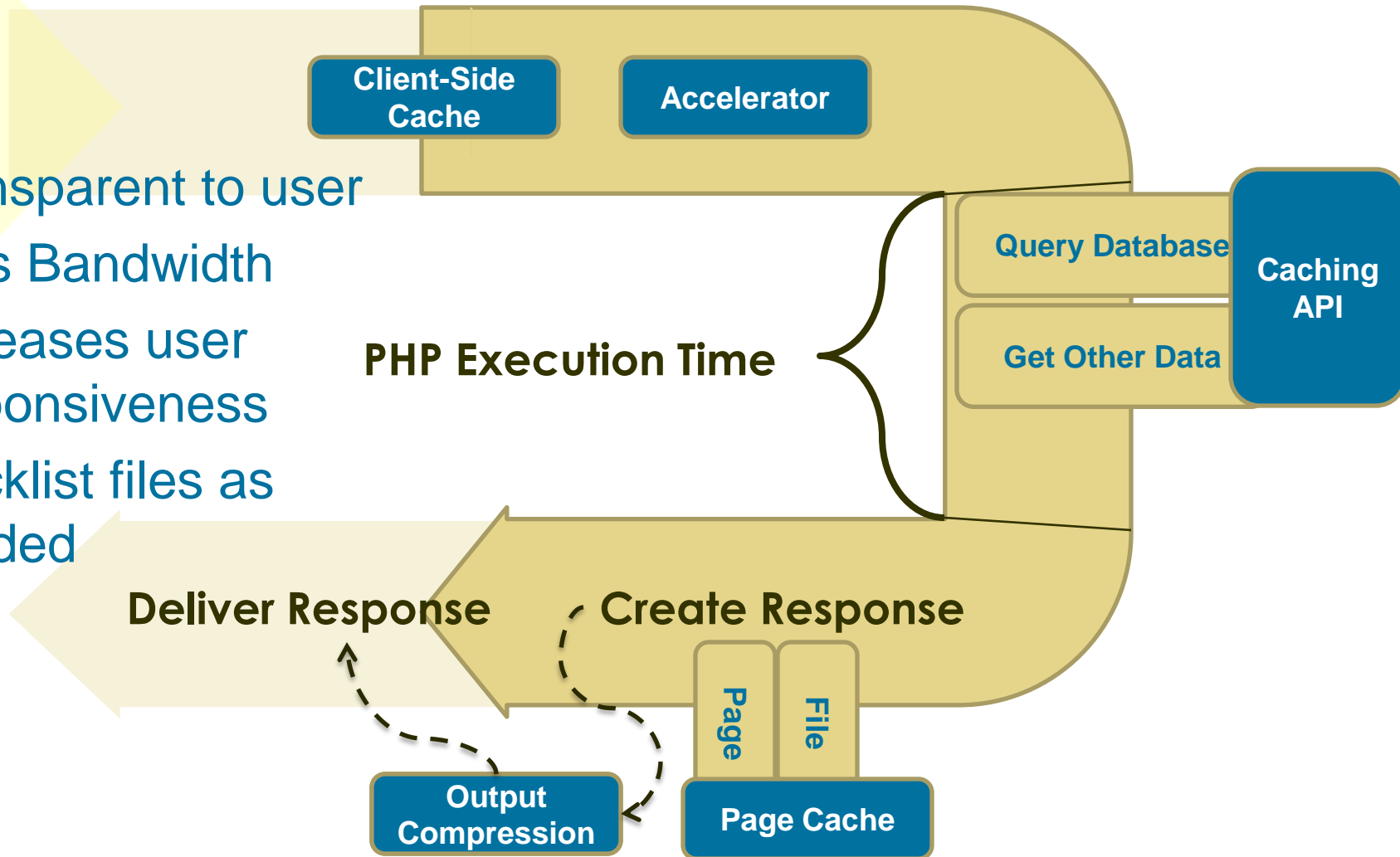
Multilayer Caching – Caching API

- **Up to 100% improvement – depending on the cached code**
- **Easy to use API – can be easily applied on existing code**
- **Supports both memory and disk storage**
- **Namespace support simplifies cache management**



Output Compression

- Transparent to user
- Less Bandwidth
- Increases user responsiveness
- Blacklist files as needed



Output Compression

The screenshot shows the Zend Platform Performance page. The browser address bar is `http://localhost/ZendPlatform/server/in`. The page has a navigation menu with tabs for Platform, PHP Intelligence, Performance (selected), Configuration, and Session Clustering. Below the navigation is a sub-menu with Console (selected), Settings, File View, Testing, and Tuning. The server information is `localhost [Change server] | User: Admin [Logout] | 10 Jul 2007, 20:19:37`. The main content area is titled "Console" and features a "Help" icon. A performance summary box shows "Overall Performance Gain: **x1.37**" valid for 10 Jul 07, 20:06, with an "Update" link. Below this is a button to "Get Latest Detailed Performance Gain". A table lists performance settings:

Code Acceleration	On	Reset	230 files accelerated 15.85 of 64MB used	Settings
Dynamic Content Caching	On	Reset	Default Cache Lifetime: 360 Default Dynamic Caching Conditions: ALLGET Add/Remove specific files to Content Caching Add/Remove	Settings
File Compression	On		Compressing all files	Settings

At the bottom of the table, there are two buttons: "Run Performance Test" and "Run Compression Test". The "File Compression" row and its corresponding button are highlighted with a green border.

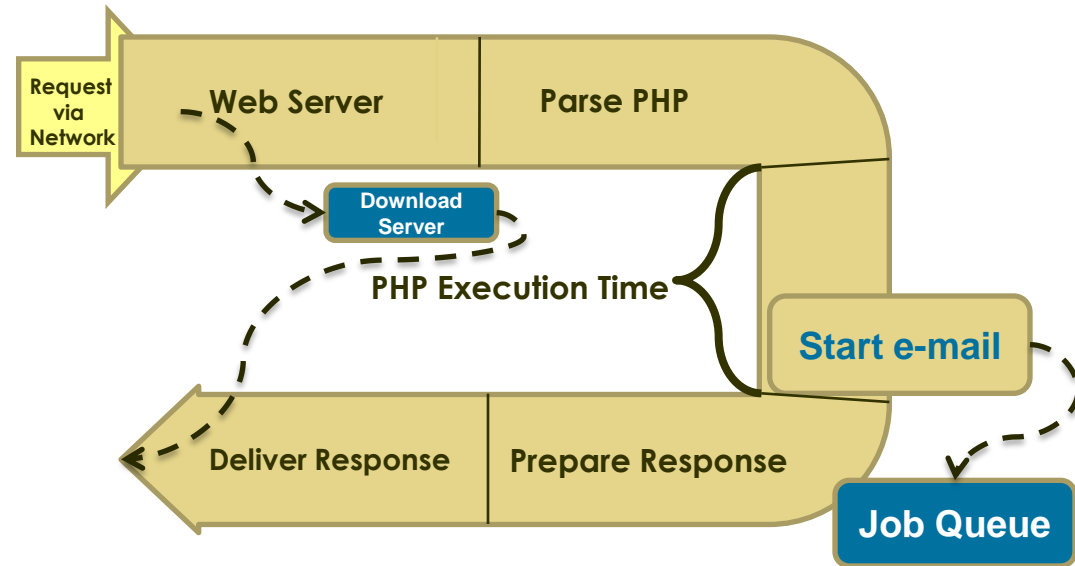
Architecture Considerations

Job Queue

- Off load asynchronous tasks
 - e-mail
 - Credit card authorization
 - Batch jobs
- Monitor Jobs like Web Apps

Download Server

- For large files (Video, .pdf, etc.)

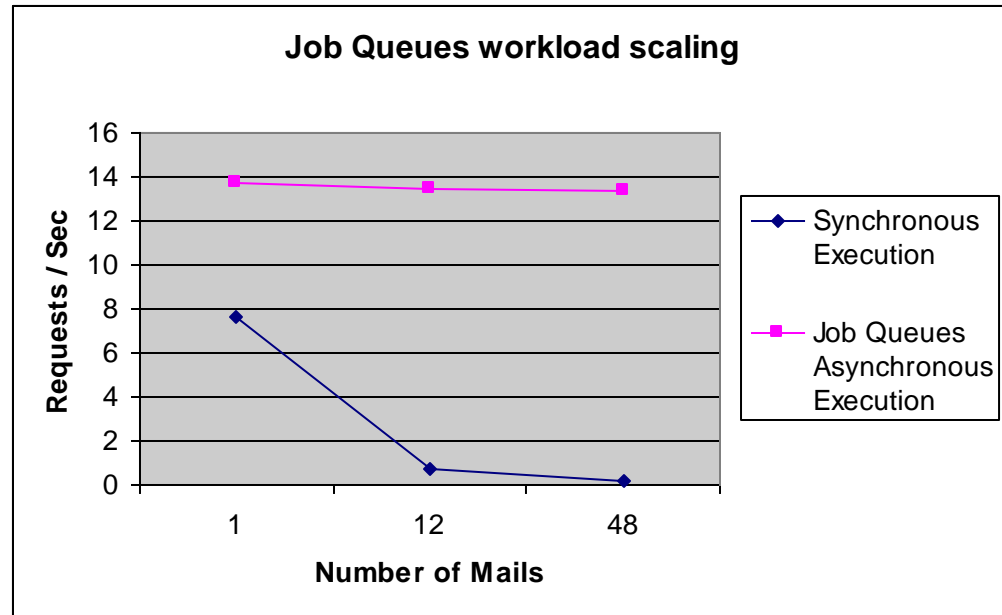


The screenshot shows the 'Job Queues' management page in the Zend Platform. It includes a navigation menu, a user status bar, and a 'Statistics' section with a table of queue metrics.

Queue Name	Completed Jobs		Current Jobs				Load Statistics				Add Job	Su R
	Successful	Failed	Recurring	Ready to Run	Scheduled	Dependent	Averaged Waiting Time	Averaged Time In Queue	# of Requested Jobs	# of Served Jobs		
zendqueue	0	2	0	0	3	0	0	0	1	0		
Total	0	2	0	0	3	0	0	0	1	0		

Architecture Considerations – Job Queue

- Improvement of 1623% on average – depending on complexity of job to be executed
- Job can be scheduled through API or GUI
- Can offload to a remote server
- Can schedule jobs for deferred execution
- Can create dependant jobs



Jobs

Filter By: Queue: local | Status: Ready to Run | Priority: All | Host: All | Application: All | Recurrence: All | Name: []

Go

Find Job by Id: [] Find

Displaying 'Ready to Run' Jobs for Queue 'local'

Jobs 1 - 300 of 4,339 (Please refine your filter criteria or modify the Jobs settings in the P)

Id	Name	Status	Priority	Application	Host
<input type="checkbox"/>	119280	notify_subscribers.php	Ready to Run	Normal	127.0.0.0
<input type="checkbox"/>	119279	notify_subscribers.php	Ready to Run	Normal	127.0.0.0
<input type="checkbox"/>	119278	notify_subscribers.php	Ready to Run	Normal	127.0.0.0
<input type="checkbox"/>	119277	notify_subscribers.php	Ready to Run	Normal	127.0.0.0
<input type="checkbox"/>	119276	notify_subscribers.php	Ready to Run	Normal	127.0.0.0

Job Details

Job #119280

Name: notify_subscribers.php

Status: Ready to Run

Priority: Normal

Host: 127.0.0.1

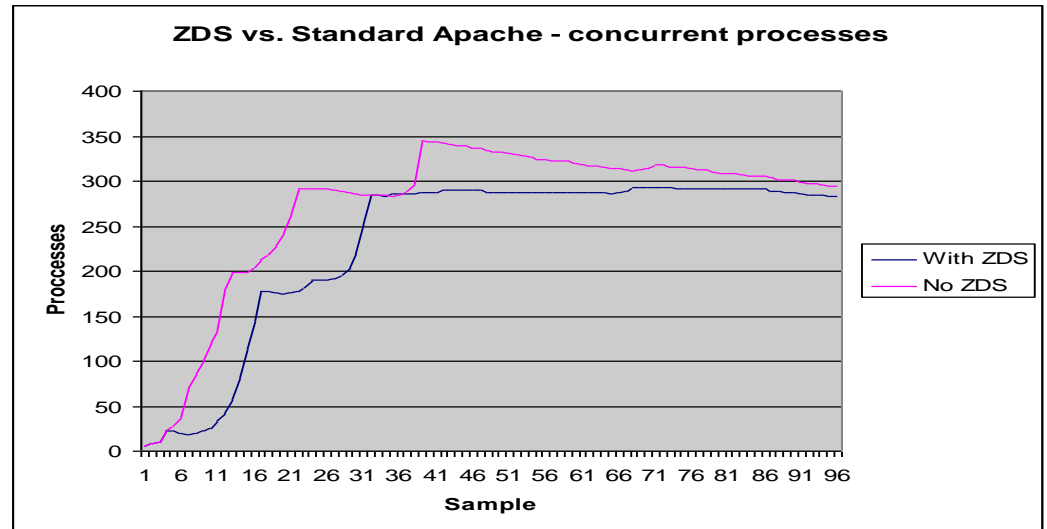
Script File: notify_subscribers.php

Context:

- Variables
- Globals
- User Variables

Architecture Considerations – Download

- Reduction of ~15% in the number of concurrent Apache processes
- Offload static downloads from Apache – allow Apache to handle the dynamic requests
- Works transparently or through API
- Allows for better scalability
- Improves performance during traffic peaks



Advanced Performance Tuning Options

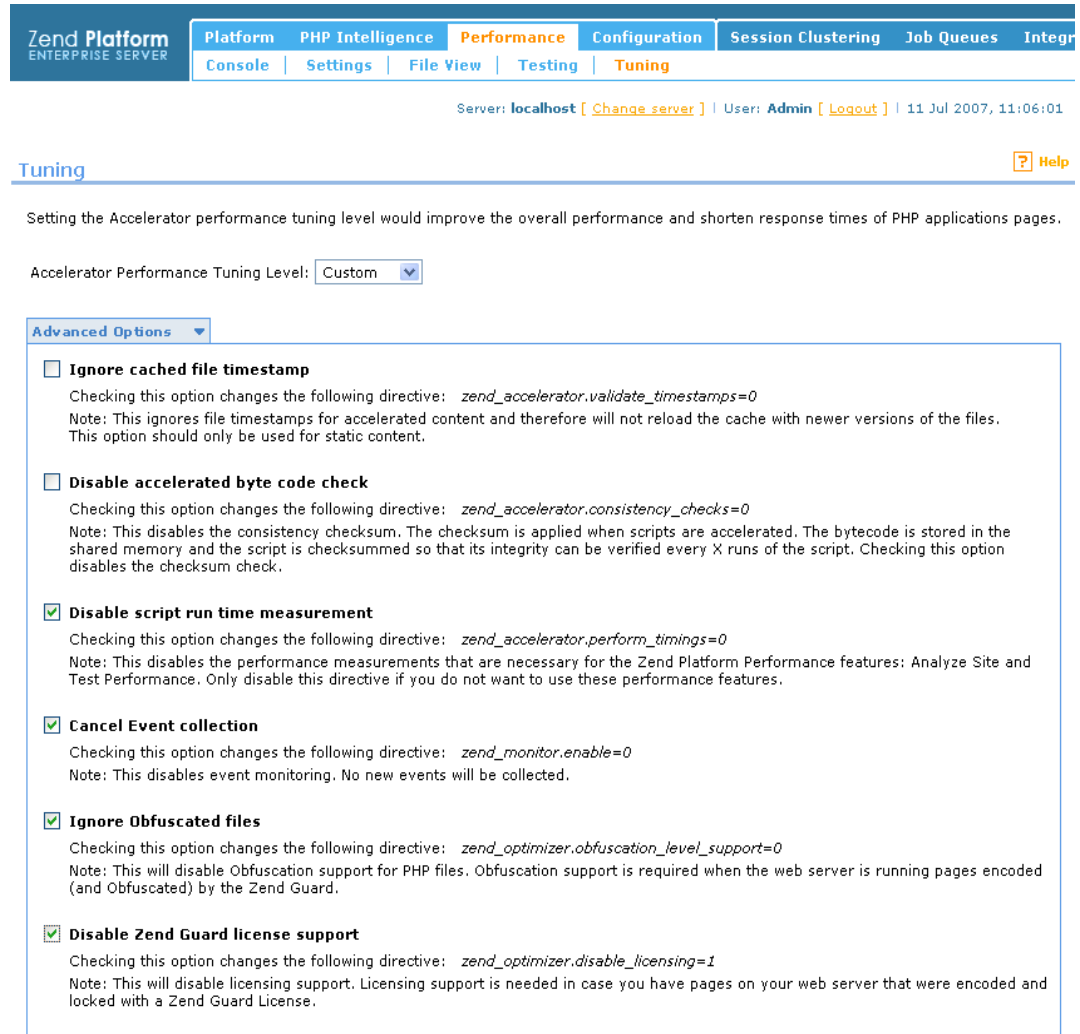
Custom tuning

- **X – Disable run time measurements**
- **X – Ignore Obfuscation**

+ 16% performance improvement

And/or:

- **X – Cancel Event collection**
- + 10% performance improvement



Zend Platform
ENTERPRISE SERVER

Platform | PHP Intelligence | **Performance** | Configuration | Session Clustering | Job Queues | Integrations

Console | Settings | File View | Testing | **Tuning**

Server: localhost [[Change server](#)] | User: Admin [[Logout](#)] | 11 Jul 2007, 11:06:01

[Help](#)

Tuning

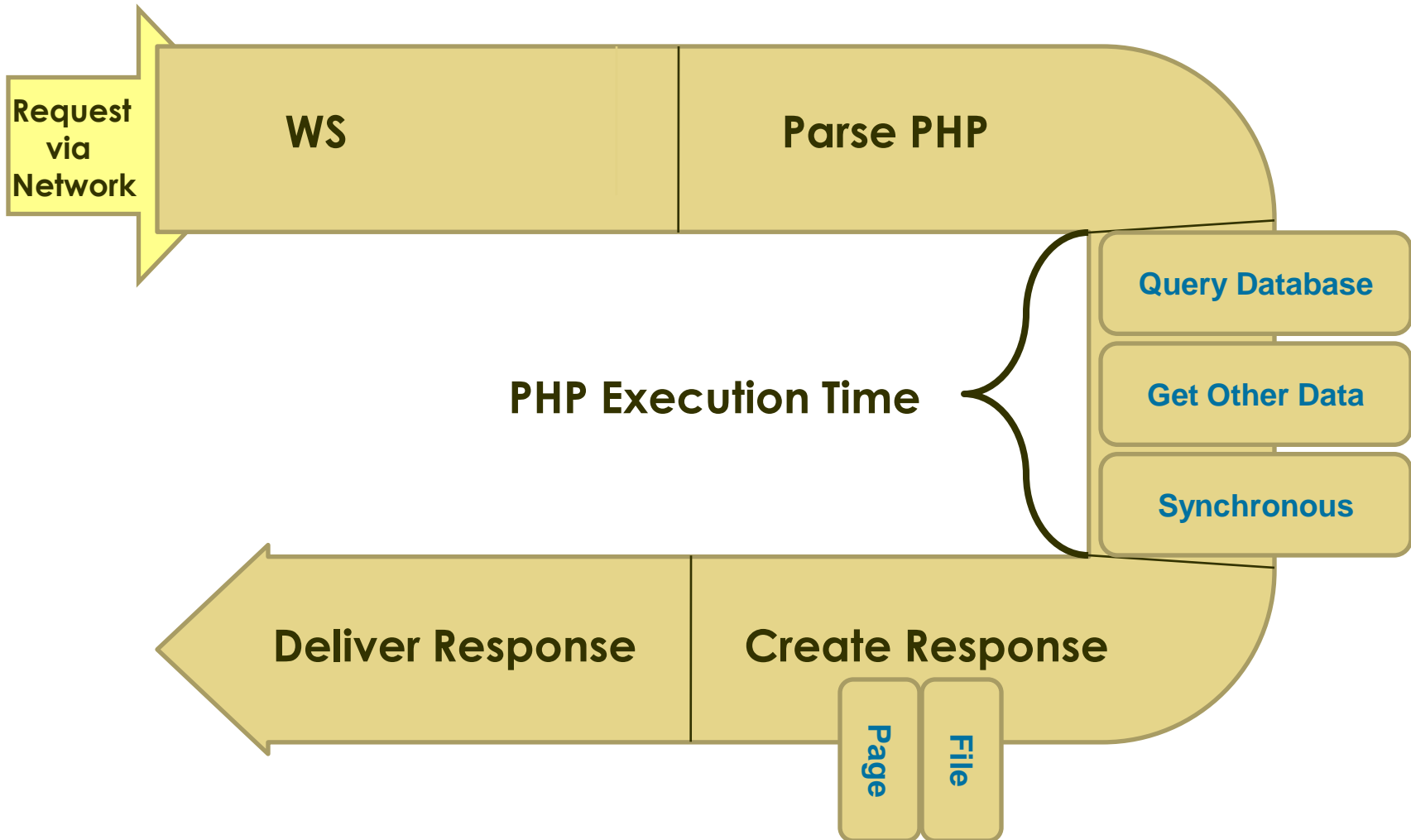
Setting the Accelerator performance tuning level would improve the overall performance and shorten response times of PHP applications pages.

Accelerator Performance Tuning Level:

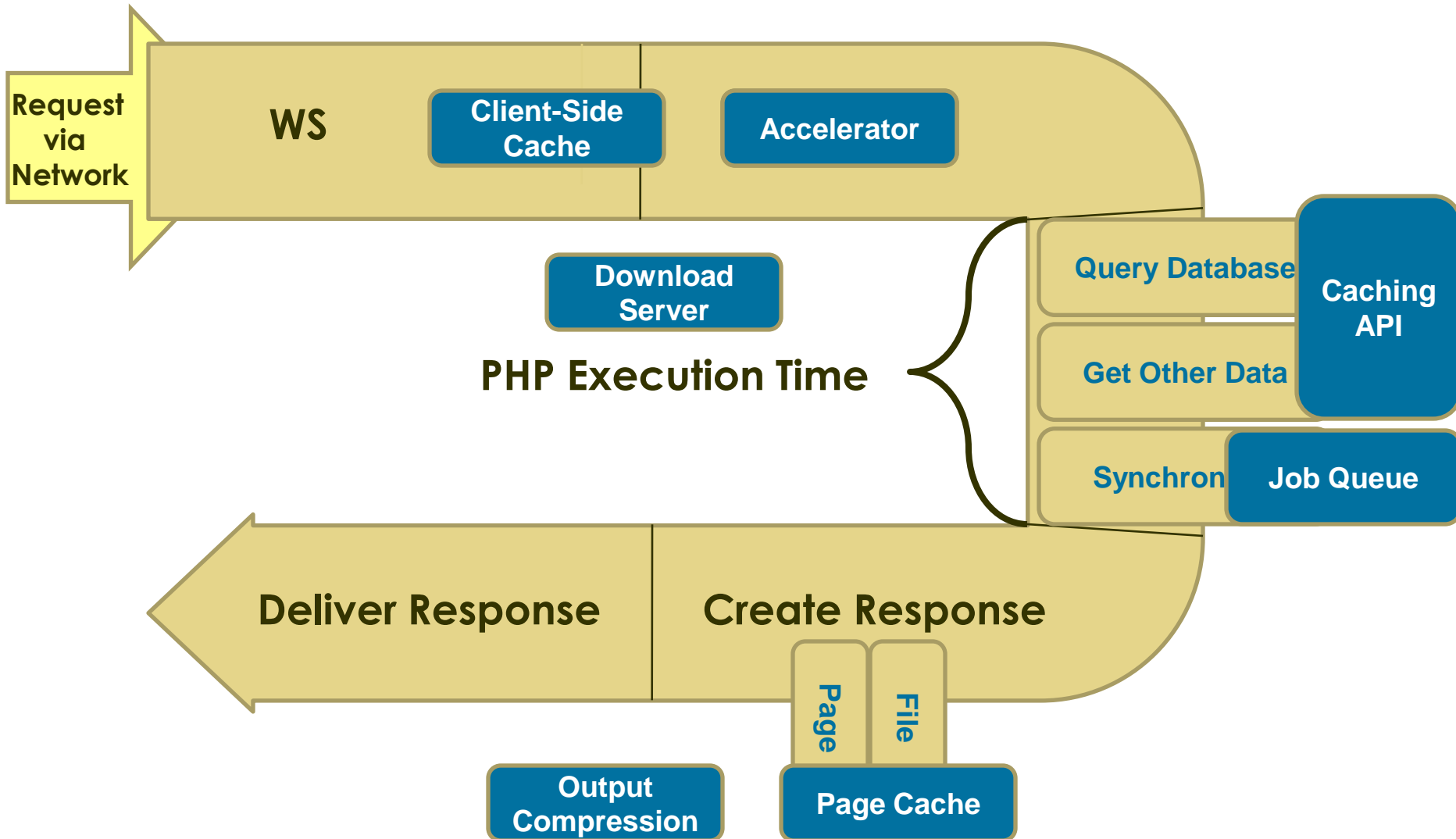
Advanced Options

- Ignore cached file timestamp**
Checking this option changes the following directive: `zend_accelerator.validate_timestamps=0`
Note: This ignores file timestamps for accelerated content and therefore will not reload the cache with newer versions of the files. This option should only be used for static content.
- Disable accelerated byte code check**
Checking this option changes the following directive: `zend_accelerator.consistency_checks=0`
Note: This disables the consistency checksum. The checksum is applied when scripts are accelerated. The bytecode is stored in the shared memory and the script is checksummed so that its integrity can be verified every X runs of the script. Checking this option disables the checksum check.
- Disable script run time measurement**
Checking this option changes the following directive: `zend_accelerator.perform_timings=0`
Note: This disables the performance measurements that are necessary for the Zend Platform Performance features: Analyze Site and Test Performance. Only disable this directive if you do not want to use these performance features.
- Cancel Event collection**
Checking this option changes the following directive: `zend_monitor.enable=0`
Note: This disables event monitoring. No new events will be collected.
- Ignore Obfuscated files**
Checking this option changes the following directive: `zend_optimizer.obfuscation_level_support=0`
Note: This will disable Obfuscation support for PHP files. Obfuscation support is required when the web server is running pages encoded (and Obfuscated) by the Zend Guard.
- Disable Zend Guard license support**
Checking this option changes the following directive: `zend_optimizer.disable_licensing=1`
Note: This will disable licensing support. Licensing support is needed in case you have pages on your web server that were encoded and locked with a Zend Guard License.

Optimize Entire Request-Response Cycle



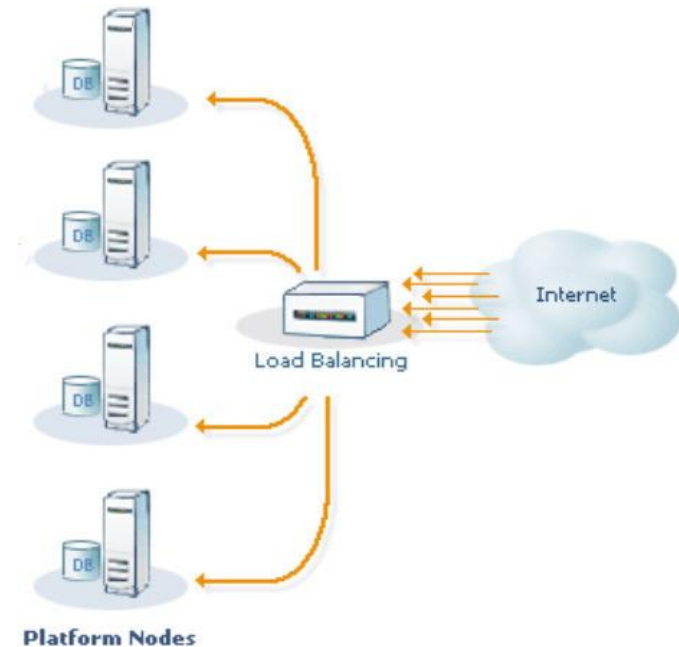
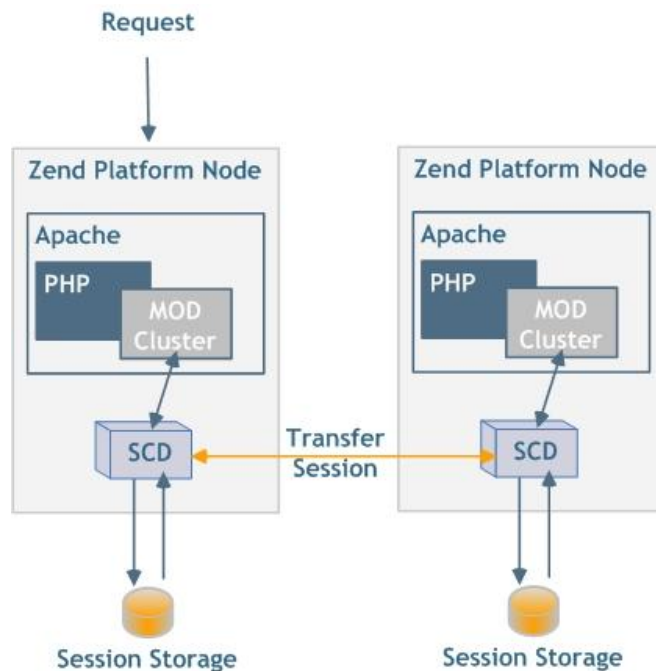
Optimize Entire Request-Response Cycle



Optimize Entire Request-Response Cycle

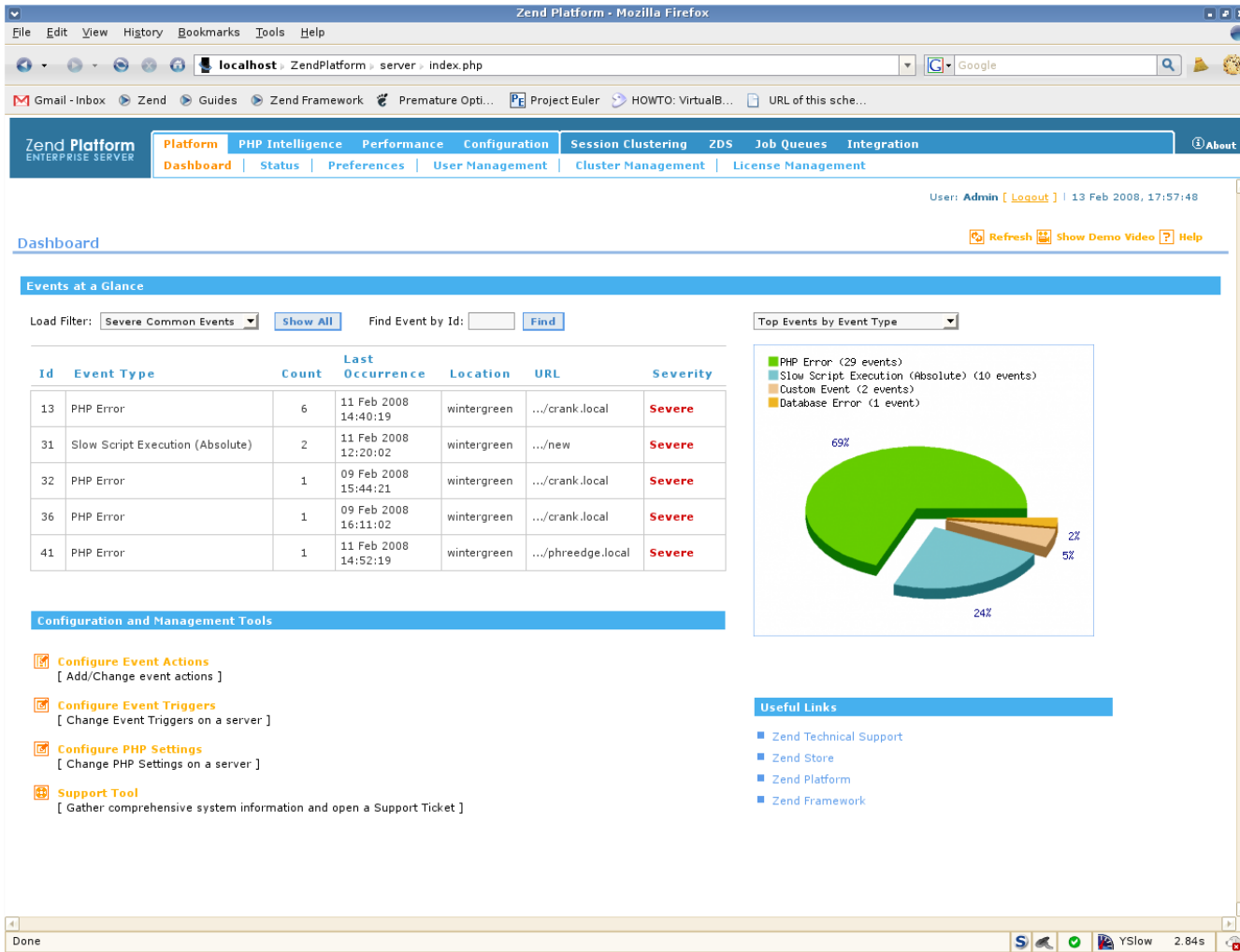
- **Accelerator up to 100%**
- **Page caching 80% to 3000%**
- **Data caching 100% +**
- **Architecture considerations**
 - Job Queue - up to 1623%
 - Download Server - 15% less Apache processes

Scale your application with -- Session Clustering



- Sessions “reside” in computers where first created -- with HA cluster mode a backup is automatically stored on secondary - Each accessible by all in cluster
- No single point of failure
- Fast Performance (up to 4 times faster than DB or NFS stored sessions)

Performance Bottlenecks: Detect, Identify & Tune



The screenshot displays the Zend Platform Performance dashboard. The main content area is titled "Events at a Glance" and includes a table of recent events and a pie chart showing the distribution of event types.

Id	Event Type	Count	Last Occurrence	Location	URL	Severity
13	PHP Error	6	11 Feb 2008 14:40:19	wintergreen	.../crank.local	Severe
31	Slow Script Execution (Absolute)	2	11 Feb 2008 12:20:02	wintergreen	.../new	Severe
32	PHP Error	1	09 Feb 2008 15:44:21	wintergreen	.../crank.local	Severe
36	PHP Error	1	09 Feb 2008 16:11:02	wintergreen	.../crank.local	Severe
41	PHP Error	1	11 Feb 2008 14:52:19	wintergreen	.../phreedge.local	Severe

Top Events by Event Type

- PHP Error (29 events) - 69%
- Slow Script Execution (Absolute) (10 events) - 24%
- Custom Event (2 events) - 5%
- Database Error (1 event) - 2%

- Worst bottlenecks are usually in the code

- PHP Intelligence collects information about run-time bottlenecks and provides easy root-cause analysis

Zend Stack

- Professional developer tools (Zend Studio)
- PHP Application Servers (Zend Platform)
- Training, support, consulting (Zend Professional Services)
- Channels support, strategic alliances (Oracle, IBM, Microsoft, ...)

Next Steps & Contact Information

- Download a Trial version of Zend Platform
- For additional information on Zend's solutions to support your PHP development and production applications:
 - Email: nasales@zend.com
 - Phone: 888-PHP-ZEND (888-747-9363)