



the  
**POWER**  
of  
**JAVA™**

***DOG 7***



JavaOne  
Part of the Network for Business Success

# Web Development: PHP Versus Java™ EE Web Tier Technologies

**Alexander Snaps**

CTO

DOG7

<http://www.dog-7.com>

TS-5672

# Goal

Lessons learned from a “real-world” experience developing, deploying and maintaining an internationalized commercial website using PHP and Java™ Platform, Enterprise Edition (Java EE) Web Tier Technologies

# Agenda

Introduction

PHP Development

First Deployment

Java EE Platform Development

Second Deployment

Comparison

Summary

Q&a

# Agenda

## Introduction

Commercial and Technical Context

PHP Development

First Deployment

Java EE Platform Development

Second Deployment

Comparison

Summary

Q&a

# Introduction

## Commercial Context

- <http://www.mtbikers.com>
  - 200,000 visitors/month
  - 3,000,000 page views/month
  - 2,000 orders/month
    - Average basket: \$200.00
- The “4 days of madness” period
  - Up to 70 transactions/second
  - 300 orders/day
  - No downtime nor denial of service acceptable

# Introduction

## Technical Context

- Previously using a Perl storefront application
  - Flat files for products and categories
  - No server side persistence
  - Duplication for internationalization
- Sync with the in-store solution: 4th Dimension
  - Developed and maintained externally
  - Average ADSL connection to the internet
- Keep previous housing solution provider
  - One server collocating all tiers
  - Scalable, maintainable and internationalized solution

# Agenda

Introduction

**PHP Development**

First Deployment

Java EE Platform Development

Second Deployment

Comparison

Summary

Q&a

# PHP Development

## What Is PHP?

- Server-side scripting
  - CGI or server module
  - Supports almost every web server
- Command line scripting
  - Parser binary
  - Cron (on \*nix) or Task Scheduler (on Windows)
- Desktop applications
  - Using PHP-GTK extension
- Runs on all major operating systems



# PHP Development

## The PHP Language

- Typing
  - Weakly typed
    - `"foo" + 2 = ?`
  - Dynamic typing
    - `$i = 10 / $j;`
- Procedural language
  - Perl-, C-like
- With OO features
  - Inheritance
  - Constructor

# PHP Development

## The PHP Language OO Features (Cont.)

- PHP 5 enhancements
  - Auto loads class definition file
  - Visibility: `public`, `protected`, `private`
  - `static`, `const`, `abstract`, `final` keywords
  - Pass objects by reference
  - Object cloning
  - Reflection API
  - Type hinting
    - `public function doStuff(array $target) {}`
  - try/throw/catch structured exception handling

# PHP Code Sample

```
<html>
  <head>
    <title>Almost i18n'zed example</title>
  </head>
  <body>

    <?php

    setlocale(LC_ALL, 'fr_FR');
    bindtextdomain("StoreFront", "./locale");

    printf( _("'Hi %s', says PHP!"), $_GET['name'] );
    ?>

  </body>
</html>
```

# PHP Development

## PHP Platform

- Web server integration
  - Shared or static module
  - No thread-safety concerns
- PHP modules
  - Shared or static at compile/installation time
- Session management
  - Explicitly or implicitly starting session tracking
  - Attributes saved as an associative array (map)
  - Serialized to disk
- Share nothing approach—well almost...

# PHP Development

## PHP Platform (Cont.)

- No connection pools...
- But persistent database connections
  - Available on multi-process/threaded web servers
    - Connection is bound to process/thread
  - Connection to DB survives the script lifecycle
  - No user session, transaction,... support
    - Still a transaction can outlive a script and, retaining locks, be propagated to the next script using the connection!
    - Solution 1: register a clean up function
    - Solution 2: use non-persistent connection when using locks

# PHP Development

## Tools

- Editors and IDEs
  - vi/emacs/Bluefish
  - Zend Studio 4 Professional
    - CVS, debugger, profiler, database and Zend server support
- Version control
  - CVS
- Development environment
  - Linux dev. boxes, with dedicated httpd/php install
    - Linux staging box over NFS
    - No server restart or packaging needed

# PHP Development

## Architecture

- Persistence
  - PostgreSQL RDBMS
  - DAO pattern
- Business logic
  - Implemented as PHP service classes
- Model 2 approach
  - Request comes in, access control comes in play
  - A PHP script manages business logic execution
  - Response is rendered, with presentation logic
    - No “PHP server pages” or equivalent available!

# PHP Development

## Architecture (Cont.)

- Framework development
  - HttpRequest, HttpResponse and HttpSession
- Session and locale management
  - URL rewriting approach
    - `http://.../en/product.php;sessionId=...?id=5432`
- Response rendering
  - XML serialization of HttpResponse (and attributes)
    - Using reflection and... a few hacks
  - Two XSLTransformations to xHTML
    - One XSL per language
    - Sablotron: XSLT 1.0, DOM Level2 and XPath 1.0



# Adding a Product to Shopping Cart

```
include_once("framework.inc");
include_once("model/Product.inc");

$cart = &$httpSession->getAttribute("cart");

$productDao = new ProductDAO();
$product = productDao->getProductById($_GET["id"]);

$qty = 1;
if(isset($_POST["qty"]))
    $qty = $_POST["qty"];

$cart->addProduct($product, $qty);

$httpResponse->redirect("/viewCart");
```

# View User Cart

```
include_once("framework.inc");

if(isset($_POST["qtys"]))
    $httpSession->data["cart"]
        ->updateQtys($_POST["qtys"]);

$shippings = Order::getShippings();
$httpResponse->addObject($shippings, "shippings");
$httpResponse->writeXslTr("cart.xsl");
```

# Agenda

Introduction

PHP Development

**First Deployment**

Java EE Platform Development

Second Deployment

Comparison

Summary

Q&a

# PHP Deployment

## Going Live

- “LAPP” environment
  - Linux, Apache, PostgreSQL, PHP
- Fine tuned compilation
  - Minimal modules
    - --disable-all
  - Compiler and architecture optimization
    - -O3
    - -march= -mcpu= -msse -mmmx -mfpmath=
  - PHP and modules statically compiled into Apache
    - Up to 20% speed increase

# PHP Deployment

## Everything Runs Smoothly

- Some fine tunings
  - Caching to files
    - Bestsellers
    - Bargains
    - Root and “level 1” categories
  - Bulk inserts/updates
  - SQL fine tunings

# PHP Deployment

## Everything Runs Smoothly

- Some fine tunings
  - Caching to files
    - Bestsellers
    - Bargains
    - Root and “level 1” categories
  - Bulk inserts/updates
  - SQL fine tunings
- Well almost everything...

# PHP Deployment

## Maintenance

- Weak typing
  - Casting request parameters
    - `SELECT *`  
`FROM products`  
`WHERE id = $_GET["pid"]` **BAD!!!**
- Dynamic typing
  - Which is meant?
    - `$product→msrp;`
    - `$product→mrsp;`
- Pass by value

# PHP Deployment

## First Concerns

- 2nd or 3rd “days of madness”
  - Twice as much requests
  - Solution didn’t scale—at all: DoS!



# PHP Deployment

## First Concerns

- First issue
  - XSLTransformation
    - Low performance on hi concurrency
- Solution
  - Changed view rendering to PHP
    - Homepage
    - Brand pages
    - Category pages
- Drawback
  - Susceptible “spaghetti code” views

# PHP Deployment

## Trouble Continues...

- Second issue
  - Disk IO
    - Session serialization
- Solution
  - RAM disk for user session
- Drawback
  - Less flexible memory management

# PHP Deployment

## What's Next?

- Foreseeing troubles
  - RAM disk usage
    - Cached data files
  - Zend Performance Suite
    - Opcode caching
    - Script parse only once
    - Reduces disk IO
    - Optimized opcode
    - Content caching

# PHP Deployment

## Disappointments

- XSL vs. PHP views
  - Maintenance
  - Spaghetti code became reality
- RAM disk
  - High memory usage
- Zend Performance Suite
  - No content caching feasible
    - Reason: session management through URL rewriting
  - No “application scoped” cache

# PHP Deployment

## Disappointments

- XSL vs. PHP views
  - Maintenance
  - Spaghetti code became reality
- RAM disk
  - High memory usage
- Zend Performance Suite
  - No content caching feasible
    - Reason: session management through URL rewriting
  - No “**application scoped**” cache
    - Vulcan Logic SRM (last release 0.7.0 November 2004)

# Agenda

Introduction

PHP Development

First Deployment

**Java EE Platform Development**

Second Deployment

Comparison

Summary

Q&a

# Java EE Platform Development

## Tools

- Editors and IDEs
  - Jetbrain's IntelliJ IDEA
    - Refactoring
- Version control
  - Subversion
- Development environment
  - Linux dev. boxes, with dedicated split install
  - MS Windows dev. boxes, only tomcat
    - Linux staging box

# Java EE Platform Development

## Architecture

- Persistence
  - Hibernate 2
- Business logic
  - POJO Services
- Model 2 approach
  - Struts
  - Tiles and JSTL
- Session management
  - Java EE servlet specs



# Java EE Platform Development

## Architecture

- Caching
  - Hibernate level 2 cache with ehcache
  - Hibernate query cache
  - Application scoped instances
- View
  - Precompiled classes: fast!
  - No duplication due to i18n
- Sessions
  - In memory, no disk IO
  - Sessions serialized on server restart

# Agenda

Introduction

PHP Development

First Deployment

Java EE Platform Development

**Second Deployment**

Comparison

Summary

Q&a

# Java Technology Deployment

## Going Live... Again

- Upgrades
  - Apache httpd 2.0
  - PostgreSQL 8.0
- Linux, Apache and Tomcat
  - mod\_jk statically linked
  - Removed rewrite rules
    - Except for redirects [R,L]
- BEA's JRockit VM 5.0

# Java Technology Deployment

## Fine Tunings

- C3P0 connection pool
  - Connection closed on exception
    - Fixed with release 0.9.x
- PostgreSQL JDBC™ API driver
  - Connection lost/errors
    - Solution: downgrade to 7.x drivers
- Hibernate mapping files issues

# Java Technology Deployment

## Maintenance

- Errors reported at compile time
  - Almost...
  - Jasper validation on JavaServer Pages™ (JSP™) specification-based pages
- Type casting minimal
  - Thanks to generics
- New features
  - Related products
  - Customer product reviews
  - WS (Axis) for stock and price updates

# Java Technology Deployment

## Days of Madness

- Less disk IO
  - Level 2 cache
  - Query cache
  - Application scoped containers
- More features
- More transactions per second
  - About 70 transactions/sec
- Load just above normal usage

# Agenda

Introduction

PHP Development

First Deployment

Java EE Platform Development

Second Deployment

**Comparison**

Summary

Q&a

# Comparison

## Life Cycle

### PHP

Share nothing approach

Script parsed and executed  
on each request

No “deploy” / packaging

### Java EE Platform

Application scope

Singleton servlets

Multi-threading left to  
programmer



# Comparison

## Development

### PHP

Dynamically typed

Weak typing

Frameworks emerging

Emerging tools and IDE

Eclipse, Zend (\$)

### Java EE Platform

Statically type

Strong typing

Enterprise level  
frameworks available

Hibernate, Spring, ...

Many Tools and IDEs

NetBeans™, IntelliJ,  
XDoclet, Junit, ...

# Comparison

## Maintenance

### PHP

No compile time check

Fast deploys

No downtime

### Java EE Platform

Error checked earlier

Longer deploy

Downtime

# Comparison

## Internationalization

### PHP

GNU Gettext library

Meaningful (almost) keys

Not appropriate for views

Duplicate pages and  
presentation logic

### Java EE Platform

Properties files

Cryptic keys

Usable in JSP pages

No duplication  
Longer page flows

# Summary

- PHP enables fast development
- But!
  - Lacks an application scope
  - Relies on high disk IO
- Partial solutions available
  - Zend (\$)
  - Vulcan Logic SRM (0.7.0 since November 2004)
- PHP misses enterprise tools, framework and IDEs
  - Code generation
  - ORM, MVC...
  - Zend Studio (\$), Eclipse emerging
- Java technology simply had it all and solved all issues!

# For More Information

## Websites

- <http://www.php.net>
- <http://www.vl-srm.net>
- <http://www.phpeclipse.net>
- <http://www.zend.com>

# Q&A



the  
**POWER**  
of  
**JAVA™**

***DOG 7***



# Web Development: PHP Versus Java™ EE Web Tier Technologies

**Alexander Snaps**

CTO

DOG7

<http://www.dog-7.com>

TS-5672