Exploring Variability-Aware Execution for Testing Plugin-Based Web Applications

Hung Viet Nguyen
Iowa State University

Christian Kästner
Carnegie Mellon University

Tien N. Nguyen
Iowa State University

ICSE 2014, May 31 - June 7, 2014
Hyderabad, India
Plugin-Based Systems
Plugin-Based Systems
Plugin Conflicts

- Occur when one plugin interferes with another plugin’s behavior
“I've been running WP-Facebox for a while now, and it's worked great. However, I recently downloaded and activated WP-Syntax and now it won't work.”

http://wordpress.org/support/topic/wp-syntax-wp-facebox-conflict
“The number one reason people give us for not upgrading to the latest version of WordPress is fear that their plugins won’t be compatible.”

—WordPress News
Integration testing to detect plugin conflicts is still very limited.

Test Confessions: A Study of Testing Practices for Plug-In Systems

Michaela Greiler, Arie van Deursen
Delft University of Technology
{m.s.greiler|arie.vanDeursen}@tudelft.nl

Margaret-Anne Storey
University of Victoria, BC, Canada
mstorey@uvic.ca

Abstract—Testing plug-in-based systems is challenging due to complex interactions among many different plug-ins, and variations in version and configuration. The objective of this paper is to increase our understanding of what testers and developers think and do when it comes to testing plug-in-based systems. To that end, we conduct a qualitative (grounded

Answering questions like these calls for an in-depth study of test practices in a community of people working on plug-in-based applications. In this paper, we present such a study, revealing what Eclipse community practitioners think and do when it comes to testing plug-in based systems.
Example of Plugin Conflict

Plugin Activations

Weather  Smiley

Weather  Smiley

Website Display

**Example**

**Weather Updates:**
Mostly cloudy today. It's currently 61°F.

Temperature not shown due to Smiley plugin
Example of Plugin Conflict

```php
<h4>Weather Updates:</h4>
<p><?php echo $msg?></p>
It's currently [:Temperature:]
```

**Weather plugin:**
Replace “[:Temperature:]” with a temperature

**Smiley plugin:**
Replace “:]” with a smiley

Example:
```
Weather Updates:
Mostly cloudy today. It's currently 61°F.
```

Example:
```
Weather Updates:
Mostly cloudy today. It's currently [:Temperature:

Example:
```
Weather Updates:
Mostly cloudy today. It's currently [:Temperature:
Detecting Plugin Conflicts

• **Feature Interaction Community:**
  - Formal methods or requirement engineering are hard to apply on large-scale software

• **Software Testing Community:**
  - Integration testing for plugin conflicts faces combinatorial explosion
Brute-Force Testing

Program + Plugins
Program with optional plugins

configure program

Program + Plugins
Result for all plugin configurations

aggregate results

P1  P2
P3  P2^n

Programs with concrete selections of plugins
brute-force execution

Results for individual plugin configurations
Brute-Force Testing

System  N plugins  2^N Products

| Product 1 | 0 | 0 | 0 |
| Product 2 | 0 | 0 | 1 |
| Product 3 | 0 | 1 | 0 |
| Product 4 | 0 | 1 | 1 |
| Product 5 | 1 | 0 | 0 |
| Product 6 | 1 | 0 | 1 |
| Product 7 | 1 | 1 | 0 |
| Product 8 | 1 | 1 | 1 |

N plugins ⇄ 2^N configurations to be tested
50 plugins

35 million years to run
320 plugins

more configurations than estimated atoms in the universe
Sampling Strategies for Testing
(e.g., Combinatorial Testing)

• Exploring only a \textit{subset} of the configuration space

• \textit{These approaches are incomplete} by nature
Is There a Shortcut?

Program + Plugins
Program with optional plugins

Program + Plugins
Result for all plugin configurations

Program + Plugins
configure program

Program + Plugins
aggregated results

brute-force execution
Programs with concrete selections of plugins

variability-aware execution
Results for individual plugin configurations

P1
P2
P3
P2^n
P1
P2
P3
P2^n
Variability-Aware Execution

- **Single run** to explore the *entire configuration space*
- Yields *equivalent results* with brute-force execution on all configs
Approach
Key Idea of Variability-Aware Execution

• Taking advantage of sharing

```php
// $foo represents plugin Foo
1 initialize();
2 if ($foo) {
3   ...
4   $content = ‘Running [Foo]’;
5 } else {
6   ...
7   $content = ‘Welcome’;
8 }
9 $status = ‘’;...
10 echo $content . $status;
```

Config: $foo = True

```
1
2
3
4
5
6
7
8
9
10
```

Config: $foo = False

```
1
2
3
4
5
6
7
8
9
10
```
Key Idea of Variability-Aware Execution

• Taking advantage of sharing

// $foo represents plugin Foo
1 initialize();
2 if ($foo) {
3   ...
4   $content = 'Running [Foo]';
5 } else {
6   ...
7   $content = 'Welcome';
8 }
9 $status = ''; ...
10 echo $content . $status;

$foo = True
1 2 3 4 9 10
$foo = False
1 2 3 4 9 10

Variability-aware execution
1 2 3 6 9 10
Key Idea of Variability-Aware Execution

- Taking advantage of sharing

```php
// $foo represents plugin Foo
initialize();
if ($foo) {
    ...
    $content = 'Running [Foo]';
} else {
    ...
    $content = 'Welcome';
}
$status = ''; ...
echo $content . $status;
```

<table>
<thead>
<tr>
<th>$foo = True</th>
<th>$foo = False</th>
<th>Variability-aware execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multi-Value Data Representation

```php
// $foo represents plugin Foo
initialize();
if ($foo) {
    ...
    $content = 'Running [Foo]';
} else {
    ...
    $content = 'Welcome';
}

$status = ''; ...
echo $content . $status;
```

\$content = CHOICE(\texttt{Foo}, ‘Running [Foo]’, ‘Welcome’)

Symbolic config option

Concrete values

CHOICE(\texttt{Cond}, \texttt{value1}, \texttt{value2})

Condition

Value when Cond=True

Value when Cond=False
Local Variations

\[
\text{CHOICE}(\text{Foo}, 1, 1) \quad \rightarrow \quad 1
\]

\[
\text{CHOICE} (\text{Foo},
\begin{align*}
\text{Object}(x &= 1, y = 2), \\
\text{Object}(x &= 1, y = 3)
\end{align*}
) \quad \rightarrow \quad \text{Object (}
\begin{align*}
x &= 1, \\
y &= \text{CHOICE}(\text{Foo}, 2, 3)
\end{align*}
)
// $foo represents plugin Foo
// $bar represents plugin Bar

1  ...
2  if ($foo) {
3      ...
4      $content = 'Running [Foo]';
5  } else {
6      ...
7      $content = 'Welcome';
8  }
9  $status = ''; ...
10 echo $content . $status;

Variability-Aware Computations

Program

$foo: CHOICE(Foo, True, False)
$bar: CHOICE(Bar, True, False)

Memory

$content: CHOICE(Foo, 'Running [Foo]', null)
$content: CHOICE(Foo, 'Running [Foo]', 'Welcome')

Condition

Foo
~Foo
True
Testing Framework

Test case → Variability-Aware Execution → Test results for all configs

function testWeather() {
    output = runWebPage('index.php') // Execute the web page for all configs
    if (WEATHER)
        assertContains (output.getEleByXPath('/html/body/div[1]'), 'Temperature');
}
Variability-Aware Execution

• Exploring the entire configuration space

• Different than *symbolic execution*
  
  • Always using **concrete values**
  
  • Path conditions are always decidable

\[ \text{Symbolic config option} \]

\[ \text{Concrete values} \]
Related Work
Related Work

- Experiments performed on small programs only

Does it scale in large real-world scenarios?
Does it scale?

• It’s in the interactions

a) Low interaction

```php
function foo($x, $y) {
    if ($x > 10)
        echo 'warning';
    return ($y > 4);
}
```

→ More sharing

b) High interaction

```php
function foo($x, $y) {
    return $x + $y;
}
```

→ Less sharing

How many interactions are there in real-world scenarios?
An Experiment
Plugins can extend WordPress to do almost anything you can imagine. In the directory you can find, download, rate, and comment on all the best plugins the WordPress community has to offer.

31,418 Plugins 669,149,370 Downloads, and Counting

Featured Plugins

Jetpack by WordPress.com
Supercharge your WordPress site with powerful features previously only available to WordPress.com users.

BuddyPress
Social networking in a box. Build a social network for your company, school,

Most Popular

- Contact Form 7
  Downloaded 17,682,768 times
- Jetpack by WordPress.com
  Downloaded 10,445,021 times
- WordPress SEO by Yoast
  Downloaded 10,642,066 times
- Akismet
  Downloaded 20,851,189 times
- Google XML Sitemaps
  Downloaded 13,637,761 times
- All in One SEO Pack
  Downloaded 18,559,006 times
- Wordfence Security
  Downloaded 1,966,136 times
- MailPoet Newsletters
Implementation

- Extended the open-source full-scale PHP 5 interpreter **Quercus**, written in Java.
Research Questions

• (RQ1) Sharing and interaction among plugins
  • How much of the output and computations are shared?
  • How often do plugins interact?

• (RQ2) Scalability
  • How much time does it take to run variability-aware execution?
Experiment Setup

<table>
<thead>
<tr>
<th>No.</th>
<th>Plugin Name</th>
<th>Version</th>
<th>Files</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jetpack</td>
<td>2.2.5</td>
<td>123</td>
<td>50,605</td>
</tr>
<tr>
<td>2</td>
<td>Types</td>
<td>1.4.0.1</td>
<td>164</td>
<td>46,071</td>
</tr>
<tr>
<td>3</td>
<td>Google Analyticator</td>
<td>6.4.4.3</td>
<td>64</td>
<td>41,287</td>
</tr>
<tr>
<td>4</td>
<td>WP Photo Album Plus</td>
<td>5.1.2</td>
<td>56</td>
<td>30,156</td>
</tr>
<tr>
<td>12</td>
<td>My Calendar</td>
<td>2.1.3</td>
<td>27</td>
<td>12,614</td>
</tr>
<tr>
<td>25</td>
<td>WP SlimStat</td>
<td>3.3.2</td>
<td>15</td>
<td>4,733</td>
</tr>
<tr>
<td>48</td>
<td>Really Simple Captcha</td>
<td>1.6</td>
<td>2</td>
<td>293</td>
</tr>
<tr>
<td>49</td>
<td>WP Facebox</td>
<td>1.2.2</td>
<td>1</td>
<td>109</td>
</tr>
<tr>
<td>50</td>
<td>Lazy Load</td>
<td>0.5</td>
<td>1</td>
<td>64</td>
</tr>
</tbody>
</table>

Total size of plugins: 1,478
Size of WordPress 3.4.2: 388
Sharing and Interactions Observed in Output

CAL (Calendar plugin)
<abbr title="Monday">Mon</abbr>
<abbr title="Tuesday">Tue</abbr>
...
GAL (Gallery plugin)
<link ... href='.../gallery-plugin/css/stylesheet.css?ver=3.4.2' ... />

CAR | CON | GAL | LAZ | CAL | FAC | WPP
<script type='text/javascript'
src='http://.../jquery.js?ver=1.7.2'></script>
Sharing and Interactions in Computations

16 plugins call function `get_locale()` of WordPress:

```php
function get_locale() {
    global $locale; …
    if (empty($locale))
        $locale = 'en_US';
}
```
Implications

1. High-degree plugin interactions exist (involving up to 16 plugins).
   
   Combinatorial testing struggles at interactions > 5. Variability-aware execution offers a solution.

2. However, interactions are rare enough. Most values/computations are shared.

   Key property that allows variability-aware execution to scale
Scalability

![Graph showing the relationship between execution time (in seconds) and the number of plugins. Execution time increases as the number of plugins increases. The graph includes data points for 0, 10, 20, and 30 plugins, with execution times of 1.4, 20.8, 62.4, and 154.1 seconds respectively. At 40 plugins, the execution time is approximately 267.0 seconds. At 50 plugins, the execution time is approximately 413.2 seconds.]

38
Comparison with Sampling Strategies

- **Brute-force/10-wise sampling**
- **Variability-aware execution**
- **5-wise sampling**
- **Pairwise sampling**

The graph shows the execution time (in seconds) on the y-axis and the number of plugins on the x-axis. Different sampling strategies exhibit varying execution times as the number of plugins increases.
Plugin Conflict between **Contact Form 7** and **Really Simple CAPTCHA**

Fatal Error: ‘win_is_writeable’ is an unknown function.
Plugin Conflict between My Calendar and WP to Twitter

// Plugin CAL: my-calendar/my-calendar.php
Line 87: $wp_plugin_url = plugin_dir_url(FILE);

// Plugin WPT: wp-to-twitter/wp-to-twitter.php
Line 39: $wp_plugin_url = plugins_url();

Plugin CAL only

<img src="..plugins/my-calendar/icons/event.png" ... />

Plugin CAL + WPT incorrect URL

<img src="..plugins///my-calendar/icons/event.png" ... />
Summary

320 plugins
more configurations than estimated
atoms in the universe

Key Idea of Variability-Aware Execution
• Taking advantage of sharing

Sharing and Interactions Observed in Output

Comparison with Sampling Strategies

10-wise sampling
Variability-aware execution
5-wise sampling
Pairwise sampling