Is Your Web Server Suffering from Undue Stress due to Duplicate Requests?

Fahad A. Arshad,
Amiya K. Maji, Sidharth Mudgal, Saurabh Bagchi

Dependable Computing Systems Laboratory (DCSL)

School of Electrical and Computer Engineering

PURDUE
UNIVERSITY

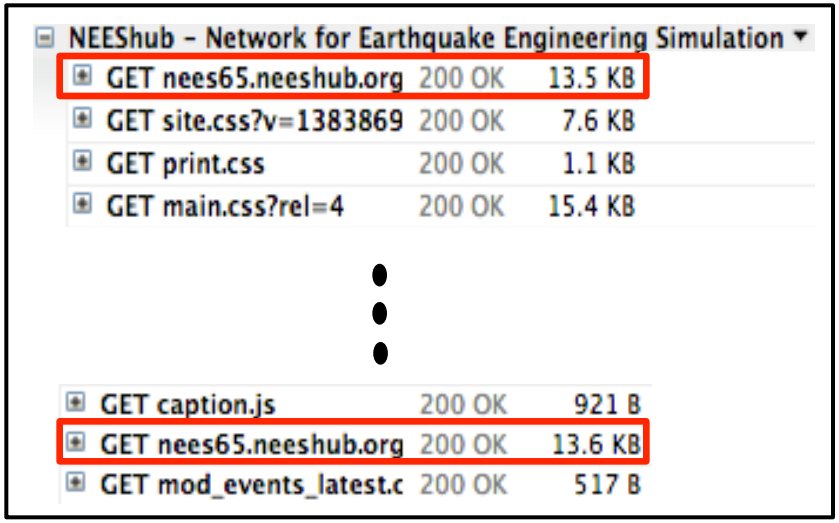
June 18th, 2014



PURDUE
UNIVERSITY

Motivation for Detecting Duplicated Requests

- What is a duplicated request?
  - A web-click resulting in the same HTTP request twice or more



- Consequences
  - Cause extra server load
  - Corrupt server state

- Frequency of Occurrence
  - Top sites CNN, YouTube
  - At-least 22 sites out of top 98 Alexa sites (Chrome)

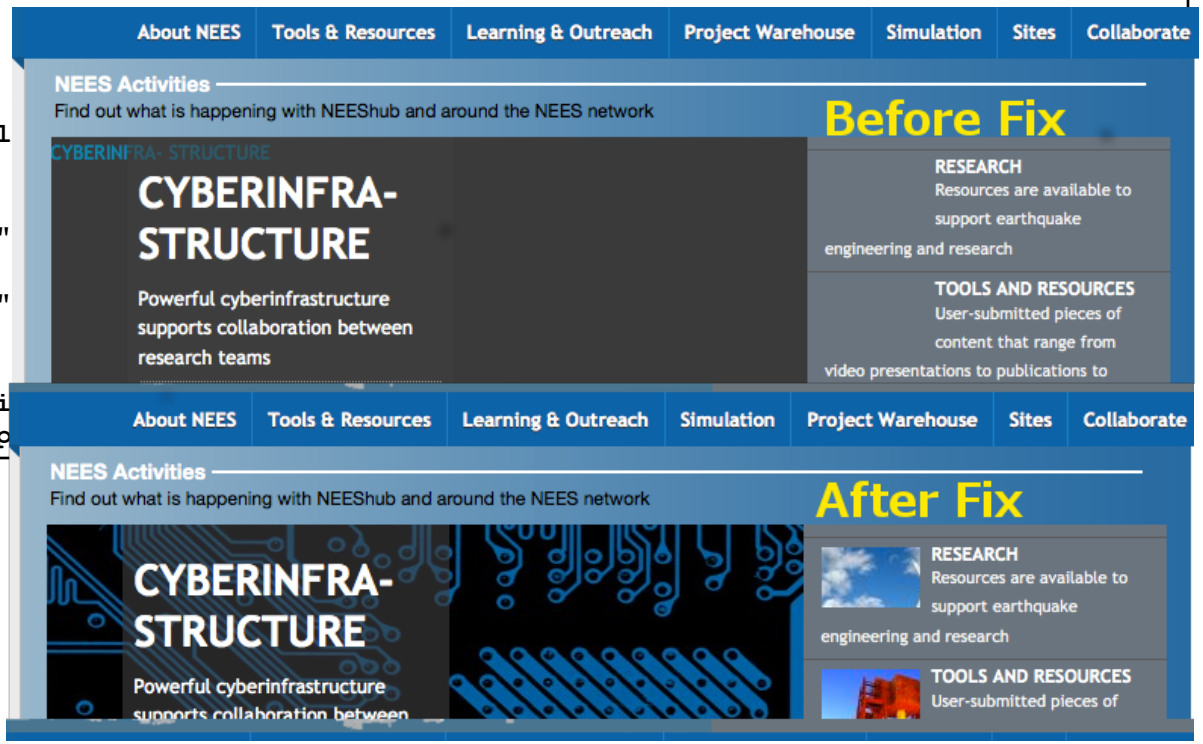
PURDUE UNIVERSITY

Root Causes of Duplicated Web Requests

- *Missing resource cause*

```
1    Var img = new Image();
2     img.src = “” //Code resolving to empty
```

```
5 -         <img src="<?php echo $slide->mainImage; ?>" alt="<?php echo $slide->altTitle; ?>" />
6 +     <span style="background:url(media/system/images/cc_button.jpg) no-repeat;">
7 +         <img src="media/system/images/cc_button.jpg" alt="<?php echo $slide->altTitle; ?>" />
8       </span>
9      </a>
10
@@ -59,7 +59,7 @@ defined('_JEXEC') or
11 <?php foreach($slides as $key => $sl
12  <li class="navigation-button">
13   <a href="<?php echo $slide->link;
14 - <span class="navigation-thumbnail"
repeat;"> </span>
15 + <span class="navigation-thumbnail"
repeat;"> </span>
16   <span class="navigation-info">
17    <?php if($slide->params->get('ti
28   <span class="navigation-title"><?p
```



- *Manifestation in browser*

PURDUE UNIVERSITY

Root Causes of Duplicated Web Requests

- *Duplicate Script Cause*

```
1 <script src="B.js"></script>
2 <script src="B.js"></script>
```

- *Manifestation in Browser*
  - None

PURDUE UNIVERSITY

Problem Statement and Design Goals

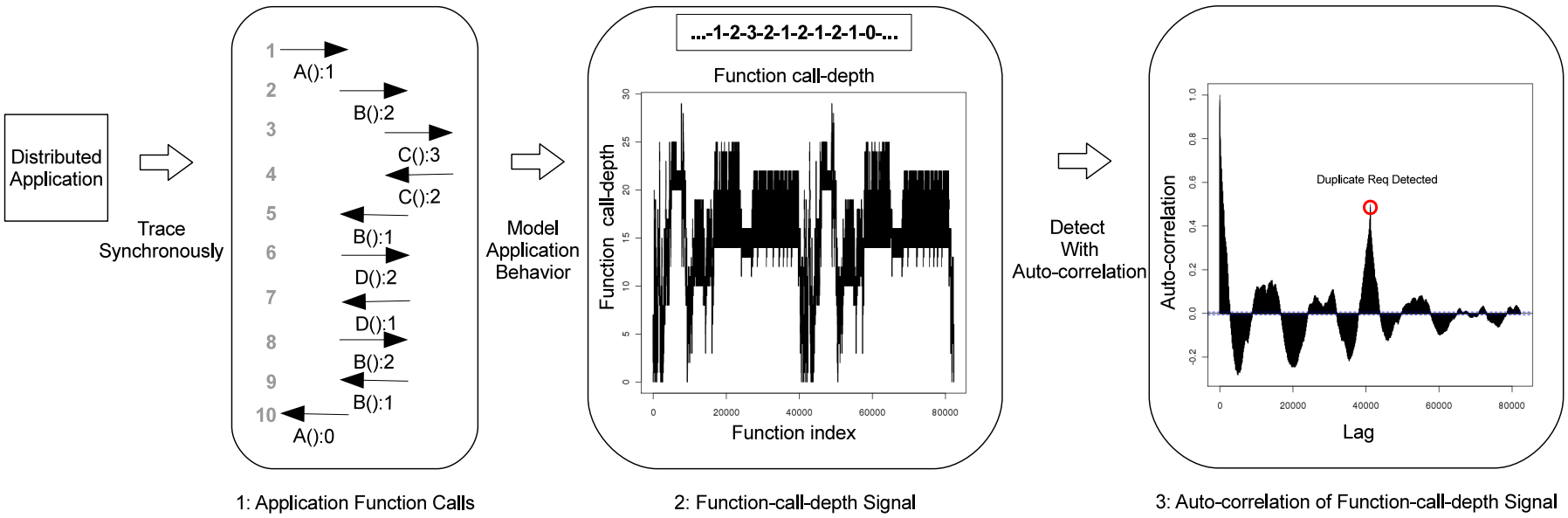
- How to automatically detect duplicated web-requests ?
- Design goals
  - General purpose solution
  - Low overhead
  - Low false-positive
  - High detection accuracy
  - Scope for diagnosis

Griffin's High-level Detection Scheme

Trace Synchronously
1

Extract Function-Call Depth Signal
2

Compute Autocorrelation and Detect on Threshold
3



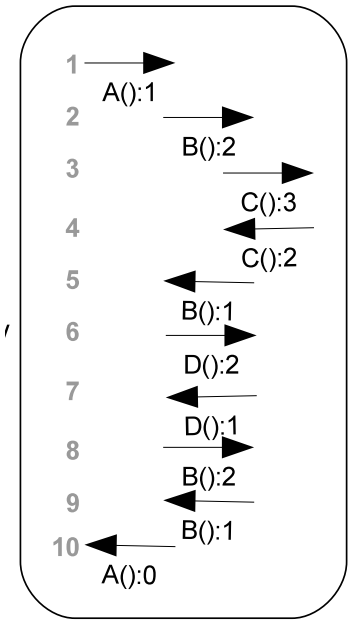
1: Application Function Calls

2: Function-call-depth Signal

3: Auto-correlation of Function-call-depth Signal

Function-call-depth to Autocorrelation Example





1: Application Function Calls

Autocorrelation => *shift* + *multiply* + *sum*

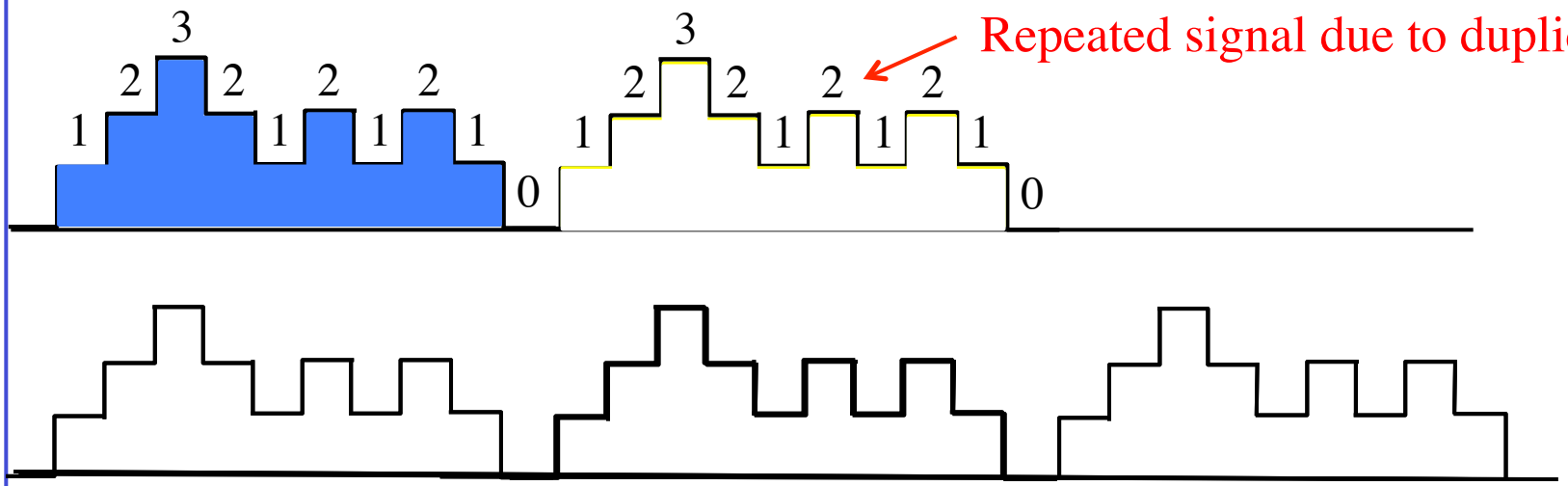
$C_0=1x1+2x2+\ldots+1x1+0x0=28$ $R_0=C_0/C_0=1$

$C_1=1x2+2x3+\ldots+2x1+1x2=24$ $R_1=C_1/C_0=0.85$

$C_{10}=1x0+2x0+\ldots+2x0+1x0=0$ $R_{10}=0/C_0=0.0$

PURDUE UNIVERSITY

Autocorrelation Example with Duplicate requests

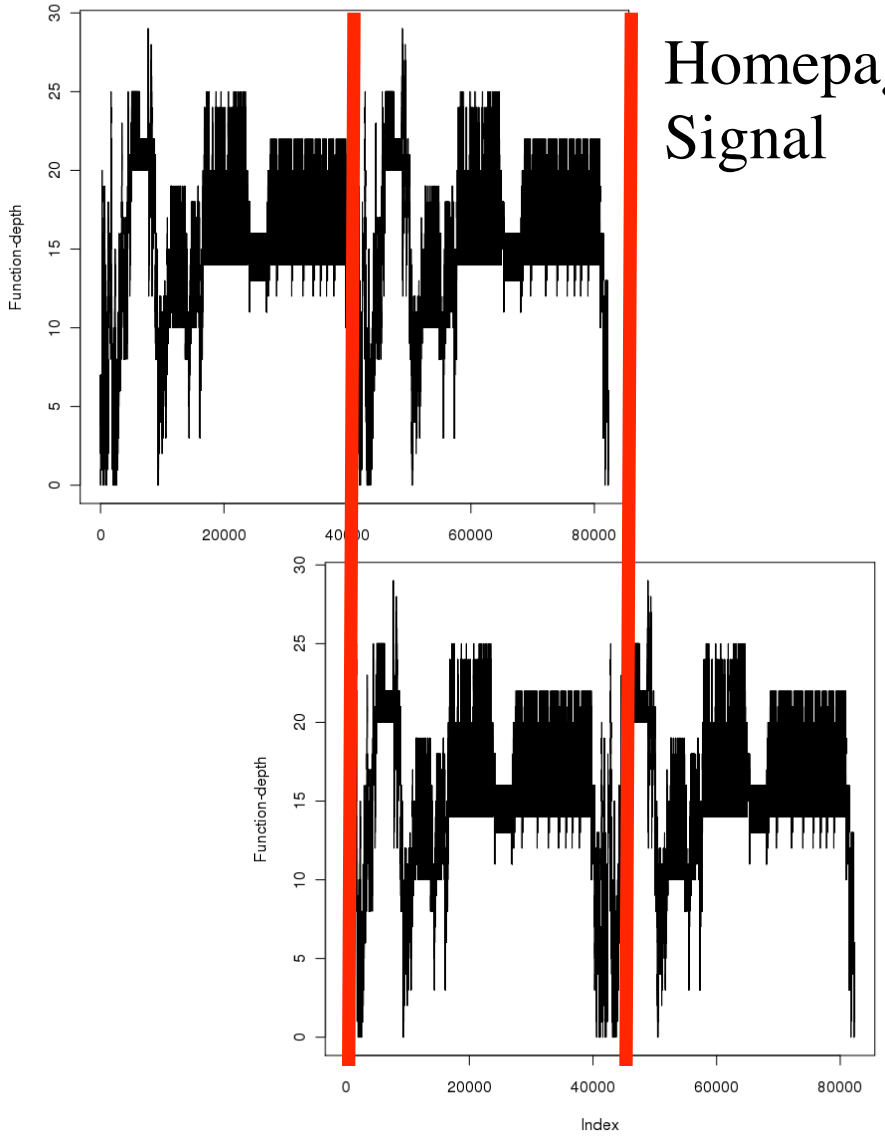


$C_0=1x1+2x2+\ldots+1x1+0x0=56$ $R_0=C_0/C_0=1$

$C_{10}=1x1+2x2+\ldots+1x1+0x0=28$ $R_{10}=C_{10}/C_0=0.5$

$C_{20}=1x0+2x0+\ldots+2x0+1x0=0$ $R_{20}=0/C_0=0.0$

Detection Algorithm Example in NEEShub

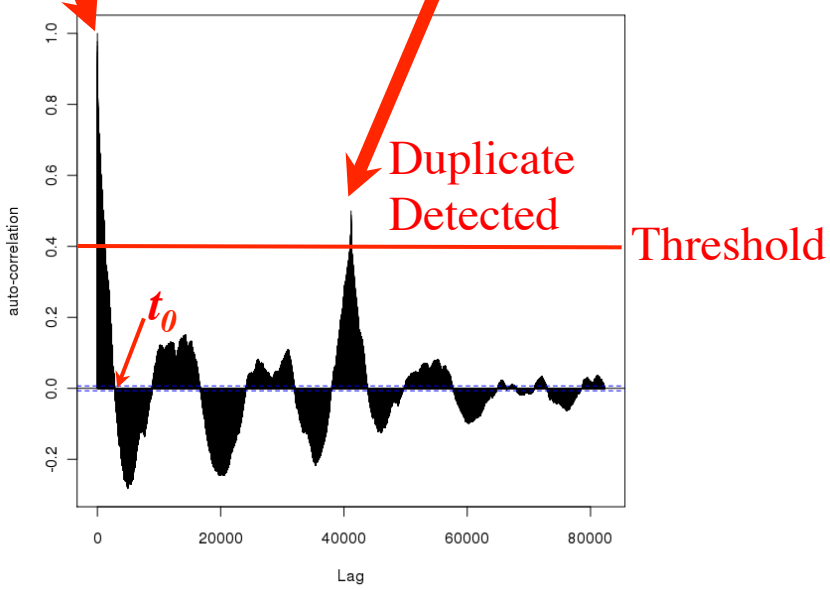


Homepage Signal

$$R_{xx}[t] = \frac{C_t}{C_0} \text{ where t=0,...,n}$$

$$C_t = \frac{1}{n} \sum_{s=max(1,-t)}^{min(n-t,n)} [X_{s+t} - \bar{X}][X_s - \bar{X}]$$

$R_{xx}[0]=C_0/C_0=1$ $R_{xx}[40000]=C_{40000}/C_0=0.49$



PURDUE UNIVERSITY

Evaluation

- HUBZERO: Infrastructure for building dynamic websites
  - www.nees.org (web server, backend database)

- Accuracy

$$\frac{True_Positives + True_Negatives}{True_Positives + True_Negatives + False_Positives + False_Negatives}$$

- Precision

$$\frac{True_Positives}{True_Positives + False_Positives}$$

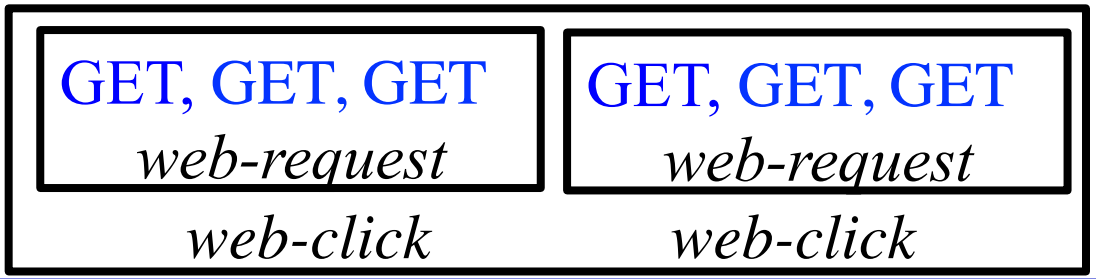
- Overhead
  - Percentage Tracing Overhead
  - Detection Latency (seconds)

PURDUE UNIVERSITY

Definitions

- Web-request
  - GET, POST
- Web-click
  - mouse clicks generating multiple web-requests
  - Homepage, Login, LoggingIn
- Http-transaction
  - Multiple web-clicks by a human user
  - Homepage➔Login➔LoggingIn (size=3)
  - Homepage➔Register (size=2)

http-transaction



PURDUE UNIVERSITY

Detection Results

- Tested 60 unique http-transactions
  - 20 http-transactions of size 1,2,3
- Ground-truth established by manual testing from browser
  - Duplicate requests found in seven unique web-clicks

	Accuracy	Precision
one-click	$90\% = \frac{18}{20}$	$100\% = \frac{3}{3}$
two-clicks	$70\% = \frac{14}{20}$	$100\% = \frac{4}{4}$
three-clicks	$75\% = \frac{15}{20}$	$0\% = \frac{0}{4}$

- Tracing Overheard
  - 1.29X
- Detection Latency: < 30 sec for sequence length of 100K

PURDUE UNIVERSITY

GRIFFIN'S Summary

- General solution for duplicate detection using autocorrelation
  - Trace function calls and returns
  - Extract function call-depth signal
  - Autocorrelation-based detection using only one threshold (0.4)
- Zero-false positives with 78% accuracy
- Low-overhead of tracing and detection

QUESTIONS ?

PURDUE UNIVERSITY