

An Empirical Study on Performance Bugs for Highly Configurable Software Systems

The 10th ACM/IEEE International Symposium on
Empirical Software Engineering and Measurement (ESEM'16)



Xue Han

Department of Computer Science University of Kentucky, USA



Tingting Yu

背景

- 近年のシステムは様々な設定（オプション）を持つ
 - Apacheには設定が1000以上ある
- 複雑な設定と高度な制約は、
パフォーマンส์バグの原因になりやすい
- 機能に関する不具合と比べ、
パフォーマンส์バグは扱いが難しい
 - 出現するタイミングが限られる
 - » 入力が大きいのときのみ、ある特定の実行環境のみ
 - 従来のテスト（カバレッジベースなど）は役に立たない

既存研究

- パフォーマンスバグを発見するための研究
 - パフォーマンスの異常を検知する手法
 - » ルールベースのパターンマッチング
 - パフォーマンス低下を確認するテストケース生成
 - » 大きく負荷をかける入力値の生成

ソフトウェアの設定が
デフォルトであることを想定しており、
様々な設定下での影響を考慮していない

目的とアプローチ

- 目的
 - 設定に関するパフォーマンスバグの特徴を明らかにする
 - » パフォーマンスバグに関するテストツールやデバッグツールの改善に貢献
- アプローチ
 - 193件のパフォーマンスバグを目視
 - そのうち、設定が関係する113件のバグを詳細に分析

Research Questions

RQ1. 設定に関連する不具合は、
どの程度出現するか？

RQ2. 設定に関連する不具合は、
どのような種類が存在するか？

RQ3. 設定に関連する不具合は、
何が原因か？

RQ4. 設定に関連する不具合は、
修正がどれだけ複雑か？

データセット

- 以下のプロジェクトからパフォーマンスに関する
キーワードを含むバグを取得
 - slow, performance, latency, throughput, etc..



Apache HTTPD

- » Webサーバー
- » 取得：323件
- » 対象： 64件



MySQL

- » 関係データベース
- » 取得：241件
- » 対象： 77件



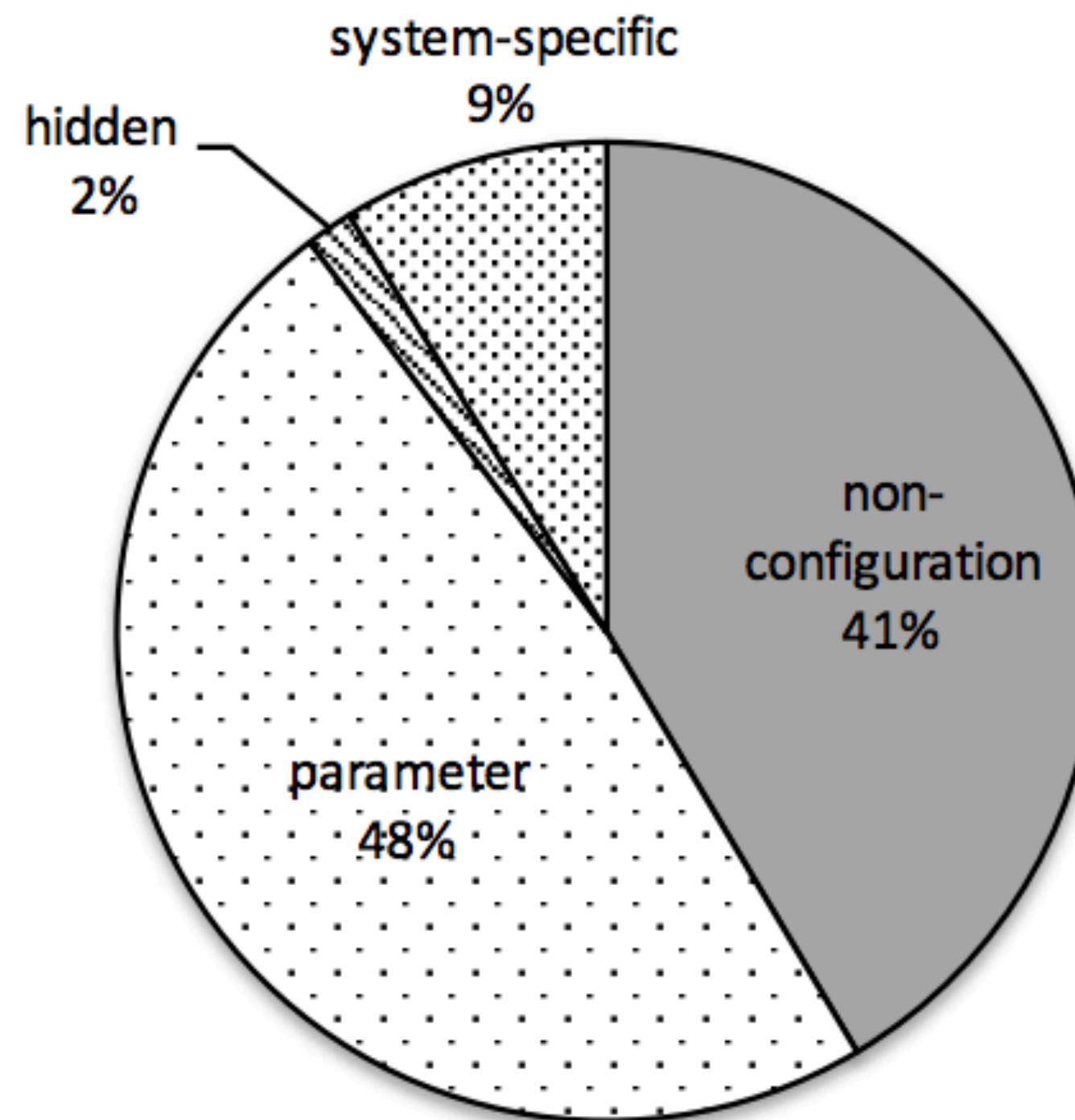
Firefox

- » Webブラウザ
- » 取得：323件
- » 対象： 53件

RQ1 : どの程度出現するか

7

- 目視で設定に関するものか, 否かを確認
 - パフォーマンスバグの59%が設定に関連する



設定には関係しないもの
(プログラムの書き方など)
例: SQLの結果がなかなか帰ってこない

RQ 2 : どんな種類が存在するか

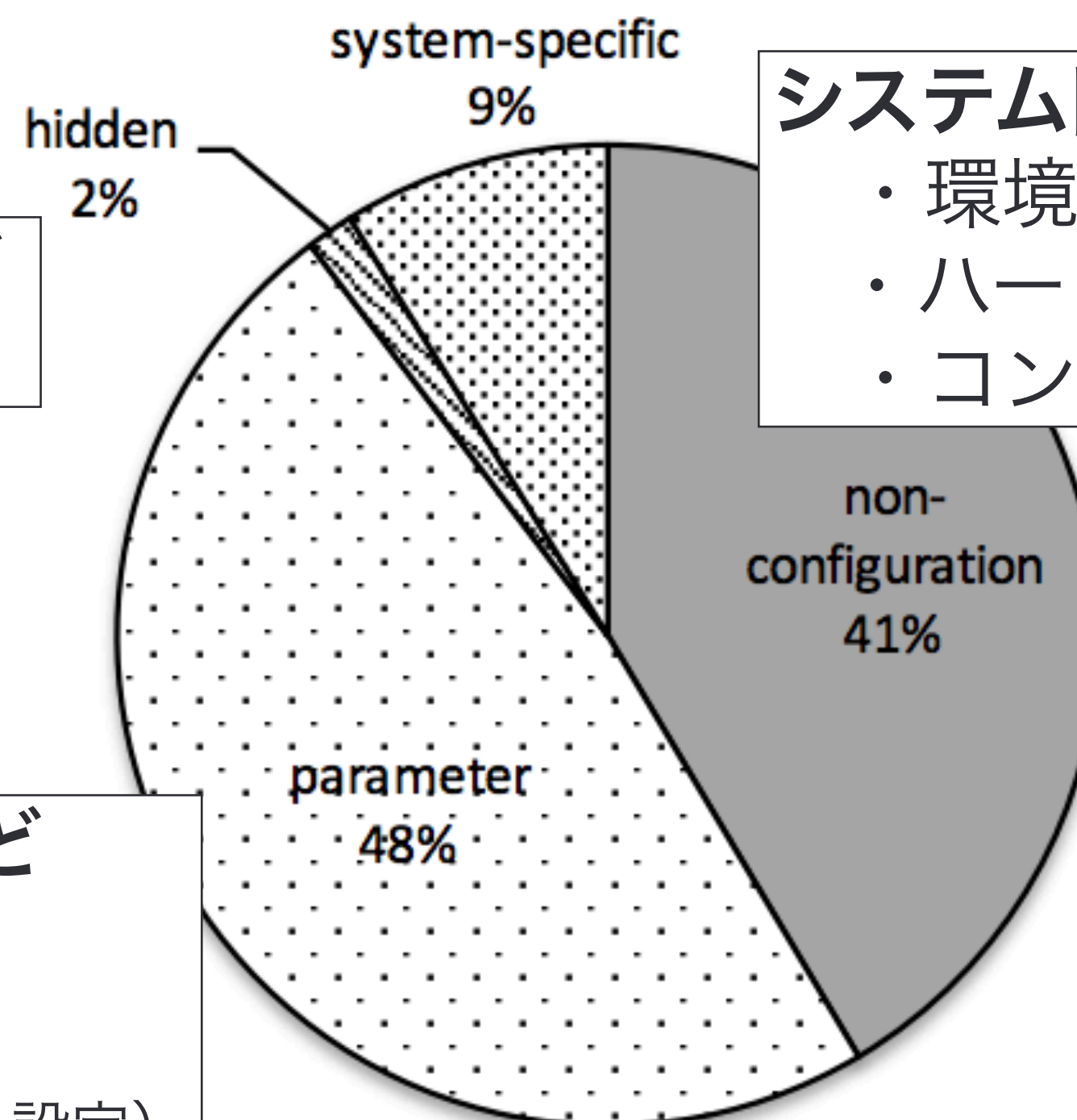
8

- 設定に関する不具合をさらに目視で分類
 - 78%から92%がパラメータに関する不具合
 - 8%から17%がシステム固有の設定に関連する不具合

ハードコーディングされている設定など
例：ハッシュサイズが小さいなど

パラメータのフォーマットや文法など

- オプションの不正な利用
- パラメータ数の間違い
- ローカル由来（搭載メモリより大きく設定）



システム固有の設定など

- 環境依存（ある設定だとLDAPが動作しない）
- ハードウェア依存（CPUのスレッド機能が無効）
- コンポーネント依存（一部OSでは別の設定が必要）

Table 2: Types of Configurations

Project	Parameter	Hidden	Sys-spec	Total
Apache	50	1	9	60
MySQL	32	2	7	41
Firefox	11	0	1	12

72%-92%

8%-17%

RQ 3 : 原因は何か

- いつ発生するのか

- 起動時

- 全プロジェクトで多く発生
 - 製品に関する知識不足やドキュメントの間違が多い

- 起動中の設定変更時

- Firefoxのみで発生
 - ＊ 起動中の設定変更はFirefoxでしかできない
 - タイムアウトの設定変更など

- バージョンアップ時

- 全てのプロジェクトで発生
 - コネクションのTTL機能が追加など
 - ＊ 旧バージョンでは, TTL=無限でデフォルト設定

Table 6: Stages of Bugs

<i>Project</i>	<i>First-time</i>	<i>Runtime</i>	<i>Upgrade</i>
Apache	60	0	4
MySQL	37	0	4
Firefox	11	12	2
Total	108	12	10

RQ3：原因は何か

- なぜ発生するのか（内部症状ごと）
 - メモリー
 - » 不要になったメモリ領域が開放されない
 - » キャッシュヒット率が低い
 - CPU
 - » 無限ループが発生
 - 同期処理
 - » スレッドの競合

Table 7: Causes of Bugs

<i>Project</i>	<i>Memory</i>	<i>CPU</i>	<i>Synchronization</i>
Apache	24	11	1
MySQL	13	3	15
Firefox	2	5	0

RQ4：修正は複雑か

11

- どのように修正しているか

- 設定値の変更
- ソースコードの変更
 - » 最も多い(88%)
 - » if文の変更や追加などが行われる
- 上記2つ共に変更

Table 8: Bug Fix

<i>Project</i>	<i>Options</i>	<i>Source</i>	<i>Option&Source</i>	Total
Apache	5	54	1	60
MySQL	2	39	0	41
Firefox	5	6	1	12

- パッチは複雑か

- 61%のバグは30行以上も変更されている
 - » ただし，既存研究では8行程度のシンプルな変更が多いと報告

所感

- ⚙ 実体験から背景にはかなり共感
 - 設定で全然パフォーマンスが違う (lp_solve)
 - この設定が最善なのか？といつも思うが、そんな研究ない？
- ⚙ この論文が役に立つ範囲は限られるが、重要な研究
- ⚙ RQがどう立てれば、役に立てるかを考えている
 - 背景でしっかり述べられている (背景がよく書けている)
- ☁ 『59%のパフォーマンスバグが設定に関するもの』
は信じがたい. . . ←目視は院生が担当. . .