

A Scalable and Efficient Approach for Compiling and Analyzing Commit History

Pooyan Behnamghader †, Patavee Meemeng †,
Iordanis Fostiropoulos †, Di Huang †,
Kamonphop Srisopha †, Barry Boehm †

University of Southern California †

紹介：戸田@福工大

概要

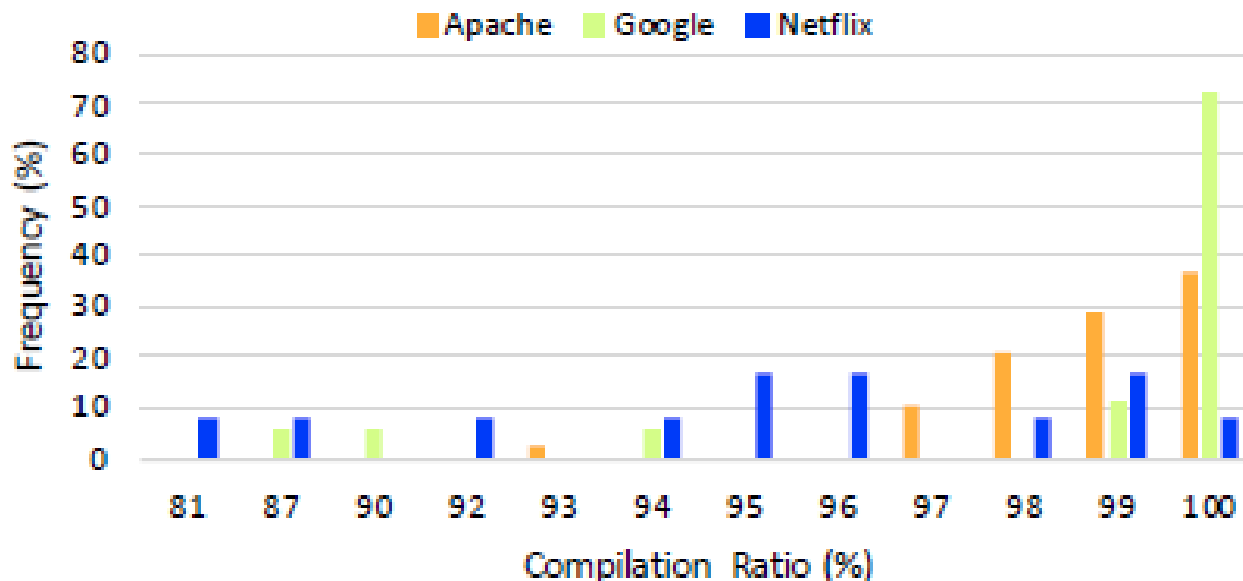
- Commit Historyの解析環境（システム）のデモンストレーション
 - 大規模なリポジトリに対しても適用可能
 - commitによる差分の解析ではなくコミットごとにシステム全体を解析する（詳細は次のスライド）

研究の立ち位置

- Repository Mining系の研究は変更されたファイルだけを分析対象にしている
- しかし正しく software evolution を評価したいのであれば、変更されなかったファイルを含めたシステム全体を分析対象にすべき
- というわけで、commitごとにコンパイルを回して解析するシステムを構築した
- で、Apache, Google, Netflixの68システムのコアモジュールを対象に分析を実施

デモンストレーション：コミット分類

- コミットごとにコンパイルを回せるので，コンパイルエラーを引き起こすコミットを特定可能
 - 同様にエラーを修正したコミットも特定可能
- コンパイルエラーを含むコミットがどの程度含まれているか調査した



デモンストレーション：それ以外

- コンパイルエラーを引き起こすコミットが投稿されてから、エラーを修正するコミットが投稿されるまでの時間の分析
 - コミットに対してコンパイルエラーを引き起こすかの予測
 - コンパイルエラーを引き起こすコミットとソースコードメトリクス（LoC, 複雑度, Code Smell等）との関係の分析
- 等々

感想

- 物量で殴ってる感
- ただし，コンパイルエラーとの関係の調査のために各プロジェクトで採用されているビルドツールの集計も行っており，細かいところまでフォロー済み
- 全体的に丁寧な仕事