様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 25 年　5 月 17 日現在

機関番号：１３３０１
研究種目：若手研究（Ｂ）
研究期間：2011～2012
課題番号：23700041
研究課題名（和文）トレース情報とプログラム解析による開発支援環境の研究開発

研究課題名（英文）Research of A New Environment for Supporting Software Developers by Trace Information and Program Analysis
研究代表者
櫻井 孝平　（SAKURAI KOUHEI）
金沢大学・電子情報学系・助教
研究者番号：80597021

研究成果の概要（和文）：本研究ではトレース情報に基づくデバッガを改良し、現実的なプログラムの実行履歴の取得を実現した。さらにこのデバッガ上に、デバッグ対象プログラムの解析を行う枠組みを開発・実装した。このデバッガと枠組みを応用して、潜在的に起こりうる欠陥の発見支援を可能にした。さらにソフトウェアテストにも応用し、プログラムの未実行箇所の解析によって品質向上につながるテストデータの生成を達成した。

研究成果の概要（英文）：In this research, we improved a trace-based debugger and achieved recording execution of realistic programs. Moreover, we developed the new framework for analyzing the debugged program on top of the debugger. We realized finding potential faults in the program by the framework and the debugger. It also applied to software testing, and we realized generating additional test inputs for improving software quality.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2011 年度	900,000	270,000	1,170,000
2012 年度	500,000	150,000	650,000
総　計	1,400,000	420,000	1,820,000

研究分野：プログラミング
科研費の分科・細目：情報学・ソフトウェア
キーワード：ソフトウェアデバッグ，プログラム解析，プログラムトレース

１．研究開始当初の背景

近年のソフトウェア工学の分野では既存のソフトウェアを分析、改善する研究が盛んである．代表的な目的は大規模化、複雑化したソフトウェアの品質や信頼性の向上であり、そのためにテストやプログラムの検証などの技術が適用される。

これまで，ソフトウェアの品質向上のためにトレース情報を扱うデバッガの研究開発がなされてきた。それらの多くはプログラムの実行履歴の情報をトレース情報としてすべて記録し、再生するデバッガツールであった。 通常、トレース情報は大量のデータになるが，近年の計算機性能の向上と実装の工夫によって実用的に扱うことが可能になった．トレース情報を再生することで、開発者はプログラムの実行を遡り欠陥を発見というデバッグの支援を受けることができる。

一方，実際にトレース情報を利用して欠陥を実際に発見することや、プログラムを改善するための支援は少なく、既存のデバッガは主に単純なデータの検索を用いて欠陥の発見を支援していた。

トレース情報はプログラムの動的な情報を全て含むため、それを解析することでよりよいデバッグの支援が可能なはずである。また、動的な情報だけでなく、プログラムのソースコードの解析(静的な解析)とも組合せることで、デバッグのみならず、プログラムの品質向上に対して幅広い有効な支援が期

待できた。

２．研究の目的

トレース情報を扱うデバッガ上に静的および動的なプログラム解析を行う枠組みを構築し、その上で実際の開発支援を行う機能を実装することで開発支援環境をツールとして開発する。このツールを実用的な事例(主にオープンソースソフトウェア)に対して適用し有効性を確認する。

具体的な開発支援としてはトレース情報とプログラム解析の組合せによって、(1)プログラム中の実行結果に影響を与えうる潜在的な(未実行の)コードの発見や、(2)プログラムの追加的なテストデータの生成を行う。

(1) データの流れの解析やポインタ解析を利用することで、プログラムの欠陥により実行されなかった関連する箇所を見つけ出すことができ、開発者にとってより自動化されたデバッグを可能にする。

(2) プログラムのテスト実行から得られたトレース情報を利用し、プログラムのデータや制御の流れの解析と組合せて、未実行部分を実行させるためのテストデータの生成を行う。これによってより多くの入力の場合を想定したソフトウェアテストが可能になり、品質向上に貢献する。

また、これらの支援を実現するために、トレースに基づくデバッガの基本的な性能の向上を行う。

有効性を示すには、実際のオープンソースソフトウェアを使って実験を行う。オープンソースソフトウェアの開発プロジェクトは大抵バグデータベースやテストプログラムを公開している。このデータベースを分析することで利用事例として適切な対象を選定できる。

３．研究の方法

まず、既存のトレースに基づくデバッガ実装である Traceglasses に対して、基本性能の向上と、並行して、プログラム解析のための枠組みの開発を行った。

基本性能の向上のために、Traceglasses のバックエンド部分の再実装を行った。これは約1万8千行からなる Java のプログラムで、他の Java のプログラムにトレース情報取得のためのコードを挿入することを目的とする。バックエンド部分では ASM と呼ばれる既存の解析ライブラリを利用することで、コンパイルされた Java のバイトコードを変更する。その際、プログラムの解析によって、トレース情報として取得するデータの削減による最適化を行った。また記録されたトレース情報を効率よく展開するために、メモリマップファイルを利用し、それらをうまく扱うことで、大きなサイズのトレース情報を記録可能にする。

プログラム解析の枠組みは Soot と呼ばれる解析ライブラリを使い、Traceglasses を拡張することで実装する。Soot には SPARK と呼ばれるポインタ解析の機能が備わっている。これは Java のプログラムとその中の変数を入力として与えると、プログラムの解析によってその変数に代入されうるプログラムの要素を解析する機能である。この機能を Traceglasses 上で利用できるように、拡張を行う。Traceglasses のデバッグ表示中に、ユーザーの指定した実行の箇所と、対応するソースコードを探し出し、Soot の入力として与える。Soot によって解析が行われたら結果を受け取り、結果に対応するプログラムの箇所に最も近い実行された箇所を画面上に表示する。

プログラム解析の評価実験として、いくつかのオープンソースソフトウェアに手動で埋め込んだ欠陥を発見するユーザー実験を行う。具体的には Apache Foundation で開発・公開されている Java のソフトウェアプロジェクトであり、Commons SCXML(状態遷移図処理エンジン)、 Apache Ant(ビルドシステム)などである。これらのプログラムの特定の分岐条件を反転させる欠陥を埋め込み、Traceglasses でトレース情報を記録する。被験者はこれらのプログラムをデバッグし、Traceglasses のプログラム解析の枠組みを利用したグループと使用しないグループ、また既存のブレークポイントデバッガを利用するグループに分けて、デバッグにかかった時間を計測した。

テストデータの生成のためには、テストプログラムの実行を記録し、制御やデータの流れ(プログラム依存グラフ)に対応させる実装を行う。これは Soot を使った枠組みを応用する。さらに、プログラム依存グラフにあらわれる分岐の条件を抜き出し、それらのうち記録した実行がどの条件の組合わせをどのような値で満たしたかを解析する。その結果、まだ実行されてない分岐の条件の組合わせを得ることができる。それらを SMT (Satisfiability Modulo Theories)ソルバと呼ばれる種類の既存のツールに渡すことで、条件を満たす新たな組合せを導出することができる。その結果、新たなテストデータとなる。今回は Z3 と呼ばれる SMT ソルバの実装を利用した。

提案した方法により導き出されたテストデータが、テスト対象のプログラムのカバレッジの向上に貢献できるかを調査することと、提案手法の適応範囲の調査を行う。カバレッジは、テストプログラムが全体のプログラムのどの程度を実行済みかを示す被覆率であり、テストの基準となる。実験で使用するプログラムは JDK に付属している Java の

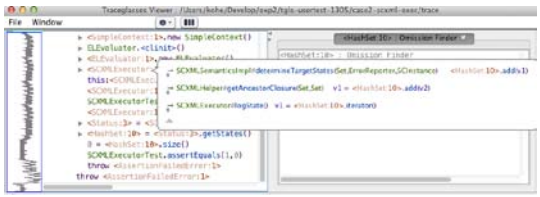
基本ライブラリのプログラムを利用する。

４．研究成果

基本性能を向上させる実装を行うことで、より幅広いプログラムに対してトレース情報を記録することが可能になった。具体的には、Dacapo ベンチマークのようなベンチマークプログラムで、トレース情報として 4 億個以上の実行時イベントを記録することが可能になった。プログラムの解析による取得データの削減によって、記録時に発生する速度低下は従来のおよそ 3 分の 1 に抑えられている。また、記録の方式を、Java エージェントと呼ばれる実装方式にしたことで、記録のためのプログラム埋め込みを実行時に行うことができるようになり、トレース情報の記録が利用者にとって容易になった。

将来的な改良の余地として、更なるトレース情報の最適化による削減や、取得するとレース情報を一定期間に制限する等で、サーバープログラム等の長時間実行されるプログラムの記録に適用が期待される。

プログラム解析の枠組みを実装することで、潜在的なプログラムの欠陥の発見が可能になった。例えば、配列オブジェクトを使うプログラムにおいて、配列に要素を追加し忘れる、という欠陥の発見支援を行うことができる。このような欠陥による不具合を実行の欠落(Execution Omission)と呼ぶ。この不具合によってプログラムの特定の分岐に誤りが存在したときに、分岐先の命令が実行されなくなり、プログラムが不正な挙動となる。このような欠落を発見する解析として、配列オブジェクトに関する潜在的な利用箇所を、プログラムのソースコード上からポインタ解析によって解析し、実際に実行されなかった箇所(トレース情報に現れなかった箇所)を特定する。さらにそれらの箇所をプログラムの制御の流れの解析により、もっともちかい実行済みの分岐の箇所を特定する。これによって実行の欠落を発生させる可能性のある箇所を特定でき、その箇所を表示する機能として実装した(下図)。



利用者はその箇所が何故実行されなかったかを直近のトレース情報を分析することで、欠陥を特定できるようになる。解析によって特定する潜在的な利用箇所はトレース情報には現れないので、従来のトレース情報を検索する手法では不可能な支援である。

実際に Apache Ant および SCXML において、それらのバグデータベースから支援が可能であったケースを発見している。例えば、Apache Ant のケースでは、プログラムの入力ファイルの特定の部分に対して、実行の欠落による欠陥が存在した。欠陥の原因箇所は 134 個の分岐に関連する 655 個の実行時イベントの候補があり、通常ではこれらの中から開発者が実際の原因箇所を特定する必要がある。一方、提案手法を適用することで、これらの候補が絞り込まれ、13 のイベントを探索することで欠陥の原因箇所にたどり着くことができた。

また、ユーザー実験の結果、3 つのケースにおいてブレークポイントデバッガと比べてデバッグ速度の向上が確認されている。また 1 つのケースにおいて、プログラム解析の枠組みの支援を利用した場合の速度向上が確認された。ただしこれは予備的な実験であるので、将来的にはより規模の大きい実験を行う必要がある。

テストデータの生成に関する提要に関しては、提案手法から得られた抽出した実行ハパス、生成されたテストケース数、生成されたテストケースがカバレッジを向上させているかを計測した。また、実行速度も計測し現実的な実行時間で求められるかを調査した。その結果、60 のテストケースに対して、カバレッジ向上に貢献可能な 120 のテストデータを生成することを確認した。実行時間はおよそ 50 秒(うち 49 秒は Z3 による計算時間)で、現実的な時間で結果が得られることを確認した。なお、これらの結果は、基本的なメソッドに対するものであり、更にネイティブメソッドのサポートの問題など解決すべき課題が残っている。

５．主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計 1 件)

① Hiroyuki Ebihara, Seiichi Komiya, Kouhei Sakurai, A PROPOSAL OF A METHOD FOR GENERATING NEW TEST CASES BY USING PROGRAM TRACE DATA, Proc. of South East Asian Technical University Consortium Symposium, Vol. 7, 2013, (査読有)

〔学会発表〕(計 2 件)

① Yusuke Shimizu, Kouhei Sakurai, Satoshi Yamane, Trace-mining Profile for Dependable Large-Scale Distributed Framework Hadoop, The 18th IEEE Pacific Rim International Symposium on Dependable Computing

(PRDC 2012) (Fast Abstract), 2012 Nov.19, Niigata Convention Center (Niigata)

② 海老原裕之, 古宮誠一, 櫻井孝平, プログラムトレース情報を利用したテストケースの前提条件の生成, 第29回 日本ソフトウエア科学会大会, 2012年 8月22日, 法政大学（東京）

〔その他〕
ホームページ等
http://www.graco.c.u-tokyo.ac.jp/~sakurai/traceglasses/index.html

６．研究組織
(1)研究代表者
櫻井 孝平 (SAKURAI KOUHEI)
金沢大学・理工研究域・助教
研究者番号：80597021

(2)研究分担者

該当なし

(3)連携研究者

該当なし