

科学研究費助成事業 研究成果報告書

平成 29 年 6 月 5 日現在

機関番号：13301

研究種目：若手研究(B)

研究期間：2014～2016

課題番号：26730034

研究課題名(和文)多様なアプリケーションのための実行履歴を扱うソフトウェア基盤の研究開発

研究課題名(英文) Research and development of a software infrastructure dealing with execution histories for various applications

研究代表者

櫻井 孝平 (Sakurai, Kohei)

金沢大学・電子情報学系・助教

研究者番号：80597021

交付決定額(研究期間全体)：(直接経費) 1,500,000円

研究成果の概要(和文)：本研究では、プログラムの実行履歴を記録し、活用する手法の開発と評価に取り組んだ。長時間実行するプログラムに対して、軽量のプロファイルデータを使った履歴の部分的な復元手法を提案し、さらに選択的に実行履歴を記録する実装を開発した。資源の制約が大きいモバイルアプリケーションに対しては、デバッグ用のログ出力とプログラムの解析からソースコードの対応箇所を特定する手法を開発した。また、実行履歴とプログラムの解析による自動的なバグ発見手法について、実際に有効性を評価した。

研究成果の概要(英文)：This study addressed development and evaluation of the technique using program execution histories. For long-running programs, we proposed the method using light-weight profile data for reconstructing histories, and developed an implementation of selective recording of histories. For mobile applications with limited resources, we developed a method for specifying source-code locations by using debug log outputs and program analysis. Moreover, we evaluated actual effectiveness of automatic bug finding technique by using execution histories with program analysis.

研究分野：プログラミング

キーワード：実行履歴 プログラム解析 ソフトウェアデバッグ

1. 研究開始当初の背景

(1) 計算機の性能向上に伴って、プログラムの実行履歴をデバッグ等に活用する技術の研究が盛んである。実行履歴は、実行されたプログラムの命令列とそこで生成された値を記録した情報である。ソフトウェアデバッグにプログラムの実行から記録した実行履歴を利用すると、欠陥(バグ)の出力結果から実行を遡ってその原因を特定することができ、デバッグ時間の短縮につながる。実行履歴を使ってプログラムを逆実行する研究は長らく行われてきたが、近年ではストレージの高速化(特にSSDの普及)やメモリの大容量化に伴って、実行全体の履歴を記録し、その実行の任意の時点に遡ることを可能にする手法が開発されている。

(2) しかし既存の実行履歴を扱う手法は、実用上では広く普及しているとはいえない。その理由として、

長時間の実行では大量の情報が記録され、ストレージのスペースと解析のための計算量が膨大になってしまう、
実行履歴を記録するオーバーヘッドが大きい、
実行履歴を高速にわかりやすく表示する技術が確立していないこと、

などの課題があげられる。特に近年ではクラウドコンピューティングのような分散並列アプリケーションと、スマートフォンなどのモバイル機器上のアプリケーションへの適用が期待される。前者は長期間停止せずに多量なデータを扱う分散したサーバープログラムに基づいており、後者は組み込みソフトウェアのように資源の制約が厳しい。結果的にこれらの対象に実行履歴を使った手法の適用は困難であった。

2. 研究の目的

本研究では長時間実行するプログラムに対して実用的な実行履歴の記録を可能にし、資源の制約が大きいプログラムに対しても適用可能な共通のソフトウェア基盤を開発することを目的とする。

本研究は産業界で主流となったオブジェクト指向プログラミング(OOP)に対して、逆戻りデバッグの実用化に取り組む。とくに近年注目されているクラウドコンピューティングや、ポストPCとされるタブレットやスマートフォン上のソフトウェアへの適用を目指す。

3. 研究の方法

(1) 長時間の実行履歴記録に対応するため、過去の不要な記録を削除する新しい記録方式を開発する。後のデバッグや解析に必要な履歴情報を適切に選択して記録するため、Javaのような典型的なオブジェクト指向プログラム(OOP)の実行の性質を利用する。制約が大き

い対象には既存の限定された実行履歴(ログ出力)情報から履歴を復元しデバッグ作業を可能にする。

(2) まず、実行履歴のデータサイズを抑える最適化として、プログラムの静的解析によって非決定的な実行のみを取得する。そして、実行時に実行履歴記録のためのコードを動的に埋め込み・取り外すことを可能にする。さらにオブジェクトの種類に基づく実行履歴の選択的な上書きを実現することで、長時間実行されるプログラムの実行履歴を現実的なサイズに抑える。

(3) 制約の大きい対象ではプログラムの追加的なコード埋込みを避け、一般的に出力されるログ情報とソースコードの解析によって実行履歴の可能な限りの復元を行う。

(4) ソースコードと対応したわかりやすい履歴の表示方式を開発・実装する。これは実行履歴と対応したソースコードの断片に実行時の値を埋め込むような表示を実現する。この表示方式は、長時間実行および資源制約が大きい場合の両方の対象で共通に利用できるスケラビリティをもち、実行履歴の解析手法を組合せて、デバッグの効率改善を達成する。

(5) これらのソフトウェア基盤と適用例をJavaプログラムのためのツールとして実装し、典型的なオープンソースソフトウェアなどの事例に対して評価実験を行うことで有効性を検証する。

4. 研究成果

(1) 長時間実行するプログラムの実行履歴を取得する方法として、軽量のソフトウェアプロファイルデータから復元する手法を提案した。従来手法による実行履歴の取得方法は、プログラムを改変して記録のための追加処理を書く命令に挿入するものであったが、これは長時間実行するプログラムに対しては膨大なデータと大きなオーバーヘッドをもたらすため、現実的ではなかった。この問題を解決するために検討を行い、プロファイルデータとプログラム解析結果から実行履歴を部分的に復元する手法を提案した。プロファイルデータはプログラム中の各手続きを実行した回数を記録した情報で、性能の分析などに利用されるが、本研究ではこの情報とプログラムの構造を自動的に解析することで、ある程度の履歴の復元を可能にする手法を開発した。プロファイルデータは実行時間が増えてもそのサイズの上限がプログラムの手続きの個数で決まるため、それほど増加しない。一方で実行順序の情報が落ちていることになるため、完全に復元することは難しいが、プロファイルデータの記録単位を並列処理の単位であるスレッドごとに増やし、プログラムの実行順

の候補となる解析手法を工夫することで、部分的に実行履歴を復元することに成功した。長時間実行を伴うプログラムとして、並列分散基盤であるApache Hadoopを想定し、実際に実験を行ない、適用が可能であることを確認した。

(2) 並行してプログラムの解析を使った自動的なバグ発見手法の開発と評価を行なった。これは期待通りに実行されなかった実行履歴に記録されない箇所から、本来実行されるべき箇所を求める手法で、ポイント解析と呼ぶプログラムの静的解析手法といくつかのヒューリスティックを利用する。ユーザー実験の結果からデバッグ時間が有意に短縮することを示し、実行履歴を応用したデバッグ手法としてプログラム解析を使った自動的なバグ発見の手法が、実際のソフトウェア開発に役に立つことを示した。

(3) 資源の制約が大きい環境のために、実行履歴の記録のためのプログラムを埋め込まずにアプリケーション開発を支援する手法を開発した。対象として、スマートフォンなどのモバイル端末上でのアプリケーション開発を想定し、一般的な開発者が利用するアプリケーションが出力するログデータから、その出力元のソースコード上の箇所と、プログラム要素へのデータの対応付を推測する。推測にはあらかじめプログラムの解析を行い、ソースコード上のログ出力箇所が参照する可能性のある文字列データから索引を作成し、実際のログデータに含まれた単語と比較することで、最適な候補を決定する。ログ出力の箇所を表示するには、ログ出力箇所参照される変数から生成されたとみられる部分文字列を特定し、ソースコード上の変数の値として表示を行う。実際にAndroidアプリケーションを対象とし、統合開発環境に対するプラグインLogChamberとして実装を行なった(図1)。既存の公開済みのアプリケーションに対して実験を行い、性能を評価したところ、比較的サイズの大きなアプリケーションに対しても実用的な環境と時間で動作することを確認した。また、実際のアプリケーションの欠陥の修正作業に適用し、有効に利用できることも確認した。

(4) 長時間の実行履歴の記録に対応するための手法と、その利用法の開発に取り組んだ。具体的には、長時間実行されるサーバープログラムで利用されるJava言語に対して、実行履歴データを記録するためのツールを開発した。膨大なデータ量となる実行履歴をうまく記録できるように、既存のデータの選択的な上書きを実装した。この実装では、実行履歴データはプログラムの解析によってプログラムの基本ブロックごとに局所変数(レジスタ)のまとめりで記録する。さらにそれらを実行

文脈としてメモリ上のスタックに保持し、関数呼出しを含む基本ブロックの遷移によって補助記憶装置に出力を行う。また、実行履歴のサイズ上限に対応するメモリマップファイルを出力先とすることで、上書きがメモリ上で行われ、プログラム終了時に一括して補助記憶装置に保存されることで、高速な実行履歴の記録を実現できる。記録したデータは復元時に解析を行って索引付けをすることで、高速な読み込みが可能である。

選択的な記録のために、実行時のオブジェクトの寿命を利用する手法を開発した。基本ブロックのデータごとに代表となるオブジェクト(多くの場合オブジェクト指向プログラムのメソッド実行文脈オブジェクト)を決め、記録時に生存している(解放されていない)古い世代オブジェクトかどうかによって、長寿のオブジェクトに関連する履歴を上書き対象から除外し、なるべく記録されるようにする。このようにすることで、膨大なデータの中からよりシステムの動作に関わると思われる実行履歴を選択できる。このような実装をSSDなどの高速な補助記憶装置を使うことで、サーバープログラムなどの長時間の実行履歴の記録に有効であると期待できる。

また、読み込みを行うための表示プログラムも新たに実装した。この実装では、サーバープログラムでは遠隔操作の端末によって作業を行うことが多いため、文字端末で動作する。

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 2件)

Kouhei Sakurai and Hidehiko Masuhara, The Omission Finder for Debugging What-should-have-happened Bugs in Object-oriented Programs, in Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC'15), (査読有) Vol 30, 2015, pp.1962--1969, DOI:10.1145/2695664.2695735

Yuki Ono, Kouhei Sakurai, Satoshi Yamane, LogChamber: Inferring Source Code Locations Corresponding to Mobile Applications Run-time Logs, IPSJ, Journal of Information Processing (JIP) (査読有), Vol. 24, 2016, pp.700-710, DOI:10.2197/ipsjjip.24.700

〔学会発表〕(計 2件)

Yosuke Nakagawa, Kouhei Sakurai, Satoshi Yamane, A method of generating traces of Hadoop YARN by lightweight profile data, Annual Meeting on Advanced Computing System and Infrastructure (ACSI) 2015, 2015年01月26日--2015年01月28日, つくば国際会議場(茨城県つくば市)

小野 祐貴, 櫻井 孝平, 山根
智, LogChamber:実行時ログから対応する
ソースコード箇所を推測するモバイルア
プリケーションの開発支援ツール, 第106
回プログラミング研究発表会, 2015年11
月05日, 国立情報学研究所 学術総合セン
ター(東京都 千代田区)

〔その他〕

ホームページ

<http://csl.ec.t.kanazawa-u.ac.jp/~sakurai/project/tgls/>

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 電子情報学系・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

(4)研究協力者

山根 智 (YAMANE Satoshi)

金沢大学 理工研究域 電子情報学系・教授

増原 英彦 (MASUHARA Hidehiko)

東京工業大学大学院 情報理工学研究科・教授