

平成 30 年 6 月 7 日現在

機関番号：14603

研究種目：基盤研究(B) (一般)

研究期間：2014～2017

課題番号：26280021

研究課題名(和文)クラウド環境で動作するソフトウェアに対するデバッグ技術の確立

研究課題名(英文) Debugging Methodologies for Software Running on Cloud Environment

研究代表者

石尾 隆 (Ishio, Takashi)

奈良先端科学技術大学院大学・情報科学研究科・准教授

研究者番号：60452413

交付決定額(研究期間全体)：(直接経費) 9,400,000円

研究成果の概要(和文)：本研究は、クラウド環境で動作するソフトウェア、特に企業の業務アプリケーションを主な対象として、ソフトウェアの実際の動作を観測し、それを用いてソフトウェアの動作の問題を分析する技術の研究を行った。ソフトウェアの複数回の実行を比較する技術、実行の中での特別な動きを検出する技術、業務データの計算手順を迅速に分析する技術などの研究を行い、ソフトウェアシステムの中でも特に開発者が注目すべき部分、動作を自動的に特定する技術を実現した。

研究成果の概要(英文)：This research investigated program analysis techniques for software running on cloud environment, in particular, enterprise applications. We developed various techniques including comparison of multiple program executions, detection of behavioral anomalies, and analysis of business data-flow. Those techniques enable developers to automatically identify important components and their behavior in a software system.

研究分野：ソフトウェア工学

キーワード：デバッグ 動的解析 プログラム理解 ソフトウェア可視化 Java

1. 研究開始当初の背景

クラウドコンピューティングは、ソフトウェアやそれを実行するためのハードウェアを開発者が利用者に譲渡するのではなく、開発者がソフトウェアを実行するための資源を保有し、利用者がネットワーク等を通じてそれを利用する、新しいソフトウェアの利用形態である。開発者がソフトウェアとその実行環境を保有し続けているため、開発者はソフトウェアの利用状況をモニタリングすることができ、バグ(欠陥)などによって障害が生じた場合は速やかに対処を行うことが可能である。クラウドは長期的に高品質のソフトウェアを提供するための基盤として、また大規模な計算機環境を多くの人々が安価に利用できる仕組みとして注目を集めているが、クラウドコンピューティングを活用したソフトウェアの増加を阻む要因の1つが、適切なデバッグ技術が開発者に提供されていないことである。

ソフトウェアのデバッグとして一般の開発者に広まっている技術は、ソフトウェアの実行を一時停止し、その状態を確認する対話的デバッグである。しかし、近年では複数のソフトウェアが並行で動作する場面が多くなってきたことから、対話的デバッグが利用できない場面が増加している。これに対して、1つのソフトウェアが外部とやり取りしたすべての入出力を「実行ログ」として記録しておき、後からソフトウェアの動作を他の計算機上で再現して分析する Capture & Replay という手法が活発に研究されている。ソフトウェア工学に関する様々な国際会議で研究発表が行われているだけでなく、商用の解析用ソフトウェアも登場しつつある。

Capture & Replay 技術は一般的なプログラムのデバッグには非常に有効であると考えられているが、クラウド環境で動作するソフトウェアに対しては直接適用することができない。クラウドは多数の計算機を並列に、可能な限り独立に動作させることで多数の利用者(あるいは大規模なデータ処理)に対応する仕組みであるため、その非同期な振舞いによって生じるバグを分析するために Capture & Replay を適用しようとする、クラウドの各計算機が行った入出力を順番に記録するための同期(時間的な待ち合わせ)が必要になり、クラウド本来の性能をまったく発揮することができず、バグが生じた実際の動作を観測、分析することができない。また、ソフトウェアが用いる情報はそれぞれ利用者の資産であり、それを実行ログとして開発者が記録することに対する懸念もある。さらに、一般的な Capture & Replay では、ソフトウェアが使用した計算機群の状態を完全に再現しようとする、単純にはすべての計算機が使用したのと同量のメモリ容量が必要になってしまう。大規模な計算を行うソフトウェアほどデバッグに必要なコストが増大してしまうことに加えて、開発者がそ

れらのデータを閲覧することも非現実的となってしまう。

これらの課題を解決し、クラウドコンピューティングのためのデバッグ技法を確立することは、社会に有用なソフトウェアを提供するために重要である。

2. 研究の目的

本研究では、クラウドコンピューティング環境上で実現されるソフトウェアシステムのデバッグ技術を確立する。クラウドコンピューティング環境では非同期的に多数の計算機を並列に動作させることで多数の利用者に対するサービス(あるいは大規模なデータの処理)を行うことから、まず、実行環境の性能を落とさないよう非同期的に実行ログを収集し、その情報からクラウド上でのソフトウェアの実行状況を推定する技術の構築を行う。

そして、デバッグ環境中に再現された実行状況の中から、バグに関係している可能性が高い計算機の振舞いと状態だけを自動的に特定し、開発者に効果的に提示する技術を実現する。

これらの技術を組み合わせることで、非同期・多並列の計算機環境で動作するソフトウェアの状況を開発者が効率的に分析し、バグの原因を迅速に特定することを可能とする。

3. 研究の方法

本研究では非同期に実行ログを記録するツールを開発し、それを基盤として技術構築を行っていく。

クラウド向けの実行ログ記録技術の開発については、まず初年度に単一計算機上で生じる並列性を扱うことでアルゴリズムの特性をつかみ、それからクラウド上の複数計算機で生じる並列性に対応するという順序で構築を行う。

デバッグに必要なシステムの部分状態を再現する技術の開発では、まず開発者が計算機群全体の状態を大域的に理解するための可視化技術を構築し、開発者が調査すべき範囲を絞り込むことを可能としてから、開発者が調査すべき情報を半自動的に削減していく技術の構築を行う。

4. 研究成果

本研究の成果は、新しい実行ログの記録技術とソフトウェアの振舞いを分析するための技術の2つに区分できる。いずれの研究成果も、ウェブアプリケーションとして企業で広く用いられるプログラミング言語 Java を対象として得られたものである。

(1) 新しい実行ログの記録技術

ソフトウェアの実行を高速に記録するた

めに、Java のソフトウェアに対してログ記録のための命令を自動的に追記し、プログラムの実行内容をディスクに記録するツール SELogger を実現し、オープンソースソフトウェアとして公開した。本研究では、このツールから得られた実行ログを対象に、様々な解析技術を確立した。

実行ログの収集に関しては、実行全体を記録するという標準的な方法だけでなく、あらかじめ興味のある問題のみに絞って情報収集を行う、低侵襲モニタリング手法も実現した [研究業績：国際会議]。クラウド上でソフトウェアが長期間稼働した場合などに起きる問題の分析に有効な方式である。この方式についても、SELogger の 1 機能として公開した。

また、ユーザの操作を伴う対話的なアプリケーションを対象として、「今、ソフトウェアの中で問題が起きているか」をリアルタイムに診断するための実行サンプリング技術を実現した。この成果については、ソフトウェア可視化に関する国際会議に採録されている [研究業績：国際会議]。

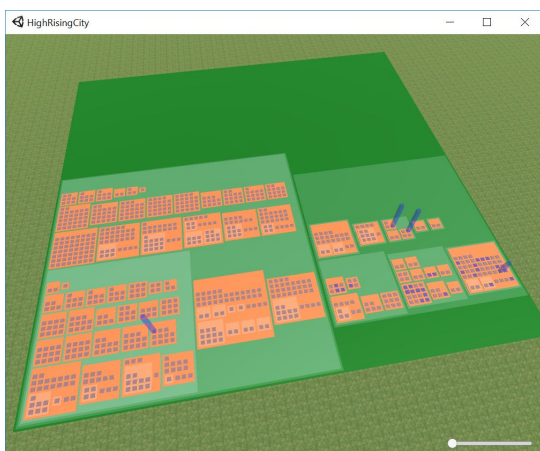
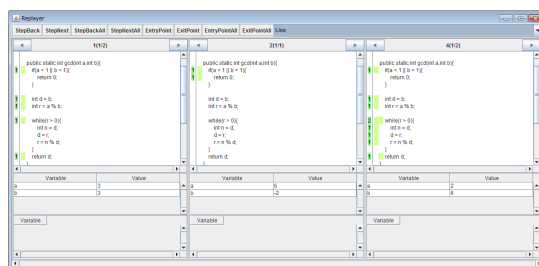


図 1. ソフトウェアの実行をリアルタイムサンプリングで観測し可視化するツールの実行例 [研究業績：国際会議]

(2) ソフトウェアの振舞いの分析技術

ソフトウェアの振舞いの分析技術として、開発者がソフトウェアの中身の多くを抽象化、すなわち重要な点以外は詳しく調査しなくても理解できる状態にするための技術を中心に研究を行った。

実行ログの利用に関しては、まず、複数回の実行を比較し、これまでと同じ実行であるかどうか、という観点から開発者に分析を許す技術を実現した [研究業績：雑誌論文]。また、開発者にとって意味のある機能の開始点を自動的に抽出する技術 [研究業績：国際会議]、実行において重要なデータを自動的に抽出する技術を実現した [研究業績：国際会議]。これらは、いずれもソフトウェアの実行の中で注目すべき情報に開発者を誘導する技術であり、いずれも査読付き



論文誌あるいは国際会議に採録された。

図 2. ソフトウェアの複数回の実行を比較するツールの実行例。異なる実行を左右に並べて閲覧できる [研究業績：雑誌論文]

実行ログ以外に利用できる情報として、ソフトウェアの命令自体に注目し、計算内容の分析を支援する技術の研究に取り組んだ。特に大きな成果として、業務アプリケーションにおけるビジネスロジック（業務データの計算手順）をソフトウェアから抽出する技術を実現した [研究業績：雑誌論文, 国際会議⑪]。この技術は実証的ソフトウェア工学に関する国際ワークショップで最優秀論文に選ばれた。また、同技術の応用として、特定の計算手順に対応する命令を検索する技術を実現し、第 23 回ソフトウェア工学の基礎ワークショップにて、フルペーパー貢献賞を受賞した [研究業績：国際会議]。これらの成果と合わせて、一群の命令が、データの追加や削除、検索といったどのような役割を担っているかを推定する技術を開発し [研究業績：雑誌論文]、ソフトウェア開発者が興味のある処理に素早く注目することを可能にした。

```

1 void action(int status,
                boolean member) {
2     if (hasError()) { return; }
3     int fee = 15;
4     if (status==CHILD) {
5         if (member) { return; }
6         fee = 5;
7     } else if (status==STUDENT) {
8         fee = 10;
9     }
10    setFee(fee);
11    setHour(2);
12    if (member) {
13        setHour(3);
14    }
15 }

```

分析

values	conditions
5	status==CHILD
10	status==STUDENT
15	!(status==CHILD) && !(status==STUDENT)

図 3. ソフトウェアの命令記述からのビジネスロジックの自動抽出 [研究業績：雑誌論文, 国際会議]

さらに本研究では、ソフトウェアの内部に組み込まれた既存のソフトウェアの部品を検出し、デバッグ作業の対象外とする技術を開発した。Java のバイナリ、ソースコー

ドの両方に対応する技術をそれぞれ開発し、いずれもソフトウェア工学のデータ分析に関するトップ国際会議の1つ Mining Software Repositories で発表した [研究業績：国際会議，②]。

これらの研究の実施を円滑に進めるために Java プログラム解析ライブラリ SOBA (Simple Objects for Bytecode Analysis) を開発しオープンソースソフトウェアとして公開したほか、従来のライブラリとの違いを比較した解説論文を執筆した [研究業績：雑誌論文]。現在、SOBA は国内の複数の研究グループに利用されている。

5. 主な発表論文等

[雑誌論文](計 13 件)

Kanyakorn Jewmeidang, Takashi Ishio, Akinori Ihara, Kenichi Matsumoto, Pattara Leelaprute. Extraction of Library Update History Using Source Code Reuse Detection. IEICE Transactions on Information and Systems, vol.E101-D, 査読有, 2018, pp.799-802, 10.1587/transinf.2017EDL8205.

竹之内 啓太, 石尾 隆, 井上 克郎. 変数のデータフローによる API 利用コード例の検索. コンピュータソフトウェア, Vol.34, 査読有, 2017, pp.4_68-4_74, 10.11309/jssst.34.4_68.

森 達也, Anders Hagward, 小林 隆志. 改版履歴の分析に基づく変更支援手法における時間的近接性の考慮と同一作業コミットの統合による影響. 情報処理学会論文誌, Vol.58, 査読有, 2017, pp.807-817, <http://id.nii.ac.jp/1001/00178569/>.

Kula Raula Gaikovina, Daniel M. German, Ali Ouni, Takashi Ishio, Katsuro Inoue. Do developers update their library dependencies? Empirical Software Engineering, Vol.23, 査読有, 2017, pp.384-417, 10.1007/s10664-017-9521-5.

秦野 智臣, 石尾 隆, 井上 克郎. SOBA: シンプルな Java バイトコード解析ツールキット. コンピュータソフトウェア, Vol.33, 査読有, 2016, pp.4-15, 10.11309/jssst.33.4_4.

Tomomi Hatano, Takashi Ishio, Joji Okada, Yuji Sakata, Katsuro Inoue. Dependency-Based Extraction of Conditional Statements for Understanding Business Rules. IEICE Transactions on Information and Systems, Vol.E99-D, 査読有, 2016, pp.1117-1126, 10.1587/transinf.2015EDP7202.

Yu Kashima, Takashi Ishio, Shogo Etsuda, Katsuro Inoue. Variable Data-Flow Graph for Lightweight Program Slicing and Visualization. IEICE Transactions on

Information and Systems, vol.E98-D, 査読有, 2015, pp.1194-1205, 10.1587/transinf.2014EDP7395.

柏原 由紀, 石尾 隆, 井上 克郎. Java メソッドの動作を表現する同士の自動推薦手法の評価. 情報処理学会論文誌, vol.56, 査読有, 2015, pp.1900-1904, <http://id.nii.ac.jp/1001/00145044/>.

Yuki Kashiwabara, Takashi Ishio, Katsuro Inoue. Improvement in Method Verb Recommendation Technique using Association Rule Mining. IEICE Transactions on Information and Systems, vol.E98-D, 査読有, 2015, pp.1982-1985, 10.1587/transinf.2015EDL8069

松村 俊徳, 石尾 隆, 鹿島 悠, 井上 克郎. REMViewer: 複数回実行された Java メソッドの実行経路可視化ツール, コンピュータソフトウェア, 査読有, 2015, pp.3_137-3_148, 10.11309/jssst.32.3_137.

Yuki Kashiwabara, Takashi Ishio, Hideaki Hata, Katsuro Inoue. Method Verb Recommendation Using Association Rule Mining in a Set of Existing Projects. IEICE Transactions on Information and Systems, 査読有, Vol.E98-D, 2015, pp.627-636, 10.1587/transinf.2014EDP7276.

石尾 隆, 伊達 浩典, 井上 克郎. メソッド呼び出しの差異に基づくコードクロンの分類手法. 情報処理学会論文誌, vol.56, 査読有, 2015, pp.1471-1480, <http://id.nii.ac.jp/1001/00142292/>.

Hironori Date, Takashi Ishio, Makoto Matsushita, Katsuro Inoue. Analysis of Coding Patterns over Version History. コンピュータソフトウェア, Vol.32, 査読有, 2015, pp.220-226, 10.11185/imt.10.226.

[学会発表](計 33 件)

中野 真明貴. 実行トレースの共通性分析に基づく機能開始点の特定. IEICE ソフトウェアサイエンス研究会 2018 年 3 月研究会, 2018 年.

Raula Gaikovina Kula. A Generalized Model for Visualizing Library Popularity, Adoption, and Diffusion within a Software Ecosystem. 25th IEEE International Conference on Software Analysis, Evolution and Reengineering, 2018 年.

嶋利 一真. ライブラリのバージョン更新支援のための実行トレースからのテストケース生成. IPSJ/SIGSE ウィンターワークショップ 2018・イン・宮島. 2018 年.

Katsuya Ogami. Using High-Rising Cities to Visualize Performance in Real-Time. 5th IEEE Working Conference on Software

Visualization, 2017 年.
Raula Gaikovina Kula. Do Developers Update their Library Dependencies? 2017 European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, 2017 年.
嶋利 一真. ソフトウェアの実行を分析するための低侵襲なモニタリングツールの試作. IPSJ/SIGSE ソフトウェアエンジニアリングシンポジウム 2017, 2017 年.
藤原 拓海. 行動履歴分析に基づく変更支援における部分履歴抽出の影響調査. IEICE ソフトウェアサイエンス研究会 2017 年 7 月研究会, 2017 年.
Kunihiro Noda. Identifying Core Objects for Trace Summarization Using Reference Relations and Access Analysis. 41st IEEE Computer Society International Conference on Computers, Software and Applications, 2017 年.
Maaki Nakano. Mediating Turf Battles! Prioritizing Shared Modules in Locating Multiple Features. 41st IEEE Computer Society International Conference on Computers, Software and Applications, 2017 年.
Akihiro Yamamori. Can Developers' Interaction Data Improve Change Recommendation? 41st IEEE Computer Society International Conference on Computers, Software and Applications, 2017 年.
Takashi Ishio. Source File Set Search for Clone-and-Own Reuse Analysis. IEEE/ACM 14th International Conference on Mining Software Repositories, 2017 年.
野田 訓広. 実行トレース抽象化を目的とした参照関係・アクセス解析によるコアオブジェクト特定. 情報処理学会 第 195 回 SE 研究発表会, 2017 年.
坂口 雄亮. ソースファイル群の類似性を用いたソフトウェア再利用元の検索. 情報処理学会 第 195 回 SE 研究発表会, 2017 年.
矢野 裕貴. Java バイトコード比較を用いたライブラリ再利用検出ツールの提案. 情報処理学会 第 195 回 SE 研究発表会, 2017 年.
Raula Gaikovina Kula. An Exploratory Study on Library Aging by Monitoring Client Usage in a Software Ecosystem. 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering, 2017 年.
Naoya Ujihara. c-JRefRec: Change-Based Identification of Move Method Refactoring Opportunities. 24th IEEE International Conference on Software

Analysis, Evolution, and Reengineering, 2017 年.
Kaoru Ito. Web-Service for Finding Cloned Files using b-Bit Minwise Hashing. 11th International Workshop on Software Clones, 2017 年.
野田 訓広. リバースエンジニアリングによる実用的な設計情報復元に向けて. 情報処理学会 SIGSE ウィンターワークショップ 2017・イン・飛騨高山, 2017 年.
相澤 遥也. インライン展開と制御構造の系列要素化を組み合わせた API 利用パターン抽出. 第 23 回ソフトウェア工学の基礎ワークショップ, 2016 年.
竹之内 啓太. プログラミング言語の構造を考慮した API 利用例検索ツール. 第 23 回ソフトウェア工学の基礎ワークショップ, 2016 年.
⑳中野 真明貴. 動的機能検索における関連度と探索戦略. 電子情報通信学会ソフトウェアサイエンス研究会, 2016 年.
㉑富永 真司. ステップ実行時に注目すべき変数を提示するデバッガの開発. 情報処理学会 第 192 回 SE・第 141 回 EMB 合同研究発表会, 2016 年.
㉒Takashi Ishio. Software Ingredients: Detection of Third-party Component Reuse in Java Software Release. 13th IEEE International Conference on Mining Software Repositories, 2016 年.
㉓松村 俊徳. 動的スライスを用いたバグ修正前後の実行系列の差分検出手法の提案. 第 191 回ソフトウェア工学研究発表会, 2016 年.
㉔竹之内 啓太. 変数の型を考慮したメソッド間の実行経路の検索. 第 22 回ソフトウェア工学の基礎ワークショップ, 2015 年.
㉕石尾 隆. SOBA: Java バイトコード解析ツールキット. ソフトウェアエンジニアリングシンポジウム 2015, 2015 年.
㉖中野 瑞樹. 動的データ依存集合の発生確率を用いた欠陥箇所特定支援手法の実装及び評価. 電子情報通信学会ソフトウェアサイエンス研究会, 2015 年.
㉗ブヤンネメフ オドファー. 動的依存グラフの 3-gram を用いた実行トレースの比較手法, 第 187 回ソフトウェア工学研究発表会, 2015 年.
㉘柏原 由紀. 動詞に着目した相関ルールを利用するメソッド名の命名支援手法の評価, 第 187 回ソフトウェア工学研究発表会, 2015 年.
㉙Shinpei Hayashi. Toward Understanding How Developers Recognize Features in Source Code from Descriptions, 9th International Workshop on Advanced Modularization Techniques, 2014 年.
㉚Tomomi Hatano. Extraction of Conditional Statements for Understanding Business Rules. 6th International Workshop on Empirical Software Engineering in Practice, 2014 年.

- ③② Raula Gaikovina Kula. Visualizing the Evolution of Systems and their Library Dependencies. 2nd IEEE Working Conference on Software Visualization, 2014 年.
- ③③ 松村 俊徳. REMViewer: 複数回実行された Java メソッドの実行経路可視化ツール. 第 31 回ソフトウェア科学会大会, 2014 年.

〔図書〕(計 0 件)

〔産業財産権〕

出願状況 (計 0 件)

取得状況 (計 0 件)

〔その他〕

SELogger プロジェクトページ

<https://github.com/takashi-ishio/selogger>

SOBA プロジェクトページ

<https://osdn.jp/projects/soba/>

6. 研究組織

(1) 研究代表者

石尾 隆 (Takashi Ishio)

奈良先端科学技術大学院大学・情報科学研究科・准教授

研究者番号: 60452413

(2) 研究分担者

小林 隆志 (Takashi Kobayashi)

東京工業大学・情報理工学院・准教授

研究者番号: 50345386