様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２９　年　　６　月　２１　日現在

機関番号：　１２６０８
研究種目：　若手研究(B)
研究期間：　2014～2016
課題番号：　２６７３００３３

研究課題名（和文）トランザクション並行制御に基づく動的記号実行方式

研究課題名（英文）Transactional Symbolic Execution

研究代表者

荒堀　喜貴（Arahori, Yoshitaka）

東京工業大学・情報理工学院・助教

研究者番号：５０６１３４６０

交付決定額（研究期間全体）：（直接経費）　　2,900,000円

研究成果の概要（和文）：本研究は、並行処理のバグを検出する動的記号実行系を実現した。従来の動的記号実行は逐次処理を対象とし、並行バグを効率的に探査することができなかった。この問題を解決すべく、本研究はまず、既存の逐次記号実行系に競合検出機構を実装し、共有データ上の潜在的競合アクセスを検知できるようにした。次に、これらの潜在的競合アクセスの中から真に競合となり得るアクセスを効率的に識別する手法を提案した。最後に、トランザクション処理の巻戻しに類似する機構を本研究の並行記号実行に統合することで、真に競合となる共有データアクセス順序を効率的に再現できるようにした。

研究成果の概要（英文）：We have extended dynamic symbolic execution (DSE) to find concurrency bugs. Traditional DSE systems are sequential and fail to explore concurrency bugs efficiently. To achieve our extension, we first incorporated a race detection scheme into an existing sequential DSE system so that it can find potential races on shared data. Next, we proposed methods to efficiently identify probably-real races among potential ones. Finally, inspired by transaction processing, we integrated a rollback-like scheme into our concurrent DSE system, in order to reproduce real racy-interleavings efficiently.

研究分野：　情報科学

キーワード：　プログラム解析　並行処理　記号実行

１．研究開始当初の背景

2005 年以降、ソフトウェア信頼性解析の分野では、動的記号実行（Dynamic Symbolic Execution、以下 DSE）と呼ばれる技術が活発に研究されてきた。この技術は、検査対象プログラムが実際に実行可能な経路を半網羅的に調べあげ、それらの経路上でバグを発見する。モデル検査やソースコード検証が対象プログラムを実行せずにバグを検出する静的解析であるのに対し、DSE は対象プログラムを実際に実行しながら、様々な実行経路上でバグを特定する動的解析である。DSE は複数の実行経路上でバグを検査でき、かつ、その検査が実行時情報に基づくため正確であるという特長を持つ。本研究の開始当初、ソフトウェア信頼性解析の分野で DSE は特に重要な技術として活発に研究が行われていた。

しかし、従来の DSE は並行処理のバグを効率良く網羅できないという問題があった。並行処理では、複数のスレッドが共有資源をアクセスしながら各自の仕事を並行して進める。ここで、各スレッドは互いに順序を調整して共有資源をアクセスするが、この順序調整のプログラミングは難しく、誤った順序がバグとなる。この種の並行処理のバグは並行バグ（concurrency bug）と呼ばれ、検出や再現が困難なことが知られていたが、従来の DSE は並行バグを効率良く検出することも再現することもできなかった。

２．研究の目的

本研究の目的は、従来の DSE では満足に扱うことのできない並行プログラムを対象とする新たな DSE の実現である。具体的には、従来の DSE を並行プログラムに適用した場合に問題となるバグの網羅率と再現効率の悪さを解決すべき課題とした。この課題に対し、トランザクション並行制御と幾つかの共通点を持つ要素技術を DSE に組み込むことで、並行バグを効率良く検出し再現できる DSE を確立することが本研究の目的である。

３．研究の方法

上記研究目的の達成に向けて、以下の方法で研究を行った。

(1)競合解析に基づく危うい経路・順序の計算方式の実現：
各スレッドの経路に加えスレッド間の共有資源アクセス順序を記録し、記録した順序の解析から並行バグの候補となる「危うい順序」を計算する手法を明らかにする。また、記録と異なる危うい順序を計算して再現することによる並行バグ検出能力の向上を実験で評価する。

(2)巻戻しに基づく並行バグ実行経路・順序の再現方式の実現：
計算した危うい順序及び経路を、特定のスレッド群のみを途中から実行スケジュールを変えて再実行することで効率良く再現する手法を明らかにする。また、この手法による並行バグ再現効率の向上実験で評価する。

４．研究成果

上記の方法に沿って研究を行い、以下の成果を得た。

(1)競合解析機構を備えた DSE 系の実現及び新規課題の特定
平成 26 年度は、逐次処理を対象とする既存の DSE 系を拡張し、検査対象プログラムのスレッド操作と共有資源アクセスを観測する機構を実現した。更に、トランザクション並行制御の競合解析に類似する技術に基づき、観測したアクセスイベント列から潜在的な競合を検出する機構を試作した。これにより、検査対象プログラムがどのような入力を受けたときに、並行処理を担うスレッド群がどのような順序で共有資源をアクセスすれば競合状態が発生し得るかを特定することが可能となった。

次に、試作した DSE 系を小規模なベンチマークプログラム群に適用し、競合検出の精度（再現率・適合率）と効率を計測した。その結果、小規模なプログラムの並行処理に対しては良い精度が得られるものの、効率面の改善には大きな課題が残ることが分かった。特に、高い精度を達成するには各スレッドの共有資源アクセス順序の観測と競合解析の予測に基づき観測とは異なる危ういアクセス順序を計算し再演しなければならないが、試作段階の DSE 系ではこの計算と再演に大きな時間を要することが判明した。更に、検査対象のプログラムおよび並行処理の規模が大きくなった場合にこの傾向がより顕著になるため、研究開始当初に計画していた並行バグ再現の高速化方式に加え、危ういアクセス順序の計算自体も効率化する技術の必要性が明らかになった。

(2)真に危うい順序の効率的計算手法の実現及び評価
平成 27 年度は、実用規模の並行プログラムに対し特定の入力を与えて実行し観測した共有資源アクセス順序から実際に並行バグが起きうる危ういアクセス順序を効率良く計算する手法を実現した。現実の並行プログラムでは、共有資源へのアクセス順序は多様な同期操作によって制御される。この同期操作の誤りが並行バグの原因となる一方で、多

様な同期操作の効果を正確に認識できない検査はバグの誤検出を生む。前年度の研究成果から、観測した共有資源アクセス順序から観測とは異なる危うい順序を正確に計算しようとすると、単純な方式では多大な計算時間を要することが判明していた。そこで、この課題を実効的に解消する方式として、共有資源アクセスと多様な同期処理の間の依存関係を正確かつ高速に計算することで真に調べるべき危うい順序を効率的に計算する手法を形式化した。

次に、形式化した真に危うい順序の効率的計算法を前年度試作したDSE系に実装し、小規模なベンチマークプログラム群および現実規模の並行プログラム群に適用する実験を行った。その結果、真に危うい順序の計算手法は、実験の範囲内で、競合検出の精度を十分に保ちつつ真に危うい順序を効率良く計算できることが分かった。本年度の成果の一部は[学会発表 4]と[学会発表 5]で発表した。

(3)巻戻しに基づく危うい順序の効率的再現手法の実現及び評価

平成 28 年度は、まず、共有資源アクセスの観測順序をもとに計算した危ういアクセス順序を並行制御の巻戻しに類似する技術を応用することで効率的に再現する手法を検討した。次に、この手法を前年度までに試作したDSE系に実装し、競合状態を引き起こしうる小規模なベンチマークプログラム群及び大規模な実用並行プログラム群に適用し、競合検出精度・効率を計測した。その結果、競合検出の精度・効率の両面において良い結果が得られた。

次に、競合に起因する原子性違反や単一スレッドによる割込み競合や複数プロセス間のファイルアクセス競合などの並行バグを扱えるようDSE系を拡張する方式を検討した。その内、競合に起因する原子性違反の検出機構をこれまでに試作したDSE系に実装し、その検出精度・効率を予備評価した。その結果、概ね良い検出精度・効率が得られたが、原子性を保つべき共有資源アクセス列の識別精度には改善の余地が残されていることが分かった。

最後に、本研究で実現した並行制御機構付きDSEの応用および展開として、以下を検討した：

①並行バグの発現過程の可視化によるデバッグ支援方式[学会発表 1]

②バイナリ形式プログラムを対象とする脆弱性検出方式[学会発表 2]

③パッチの影響範囲に限定した高効率な並行処理デバッグ方式[学会発表 3]

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　0 件）

〔学会発表〕（計　5 件）

(1)大村裕, 荒堀喜貴, 権藤克彦, 渡部卓雄, Code Oriented Diagram Editorを用いた並行バグの可視化, 日本ソフトウェア科学会第 14 回ディペンダブルシステムワークショップ（DSW2016）
https://sites.google.com/site/jssstdsw/dsw2016, セッション 2-3, 2016/12/15, 花びしホテル（北海道・函館市）

(2)長田晃太朗, 荒堀喜貴, 権藤克彦, 動的バイナリ計装に基づく正確なヒープ境界検査, 日本ソフトウェア科学会第 14 回ディペンダブルシステムワークショップ（DSW2016）
https://sites.google.com/site/jssstdsw/dsw2016, セッション 1-5, 2016/12/14, 花びしホテル（北海道・函館市）

(3)佐々木俊亮, 荒堀喜貴, 権藤克彦, 静的コード解析に基づくプルリクエスト品質即時計測, 日本ソフトウェア科学会第 14 回ディペンダブルシステムワークショップ（DSW2016）
https://sites.google.com/site/jssstdsw/dsw2016, セッション 1-4, 2016/12/14, 花びしホテル（北海道・函館市）

(4)荒堀喜貴, 並行バグの高精度かつ高効率な検出方式の検討, 日本ソフトウェア科学会第 14 回ディペンダブルシステムワークショップ（DSW2015）
https://sites.google.com/site/jssstdsw/dsw2015, セッション 3-9, 2015/12/17, ホテル水葉亭（静岡県・熱海市）

(5)荒堀喜貴, 並行ソフトウェア実行時検証のアクセラレーション技術, 日本エレクトロニクスショー協会 Design Solution Forum 2015（DSF2015）,
http://dsforum.jp/2015/timetable_soft.html#c3, セッション C-3, 2015/10/02, 新横浜国際ホテル（神奈川県・横浜市）

〔図書〕（計　0 件）

〔産業財産権〕

○出願状況（計　0 件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
国内外の別：

○取得状況（計　0 件）

名称：

発明者：
権利者：
種類：
番号：
取得年月日：
国内外の別：

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者
荒堀　喜貴（Arahori Yoshitaka）
東京工業大学・情報理工学院・助教
研究者番号：50613460

(2)研究分担者
（　　）

研究者番号：

(3)連携研究者
（　　）

研究者番号：

(4)研究協力者
（　　）