様式 C-19

科学研究費補助金研究成果報告書

平成 23 年　5 月 12 日現在

機関番号：１４４０１
研究種目：若手研究（B）
研究期間：2008　～　2010
課題番号：20700026
研究課題名（和文）ディペンダブル分散システム実現のための耐故障アルゴリズムのモデル検査
研究課題名（英文）　Model Checking of Fault-Tolerant Algorithms for the Dependability of Distributed Systems
研究代表者
土屋　達弘（TSUCHIYA TATSUHIRO）
大阪大学・大学院情報科学研究科・准教授
研究者番号：30283740

研究成果の概要（和文）：コンセンサスアルゴリズムと呼ばれる，複数の計算機から構成される分散システム上で耐故障性を実現するためのアルゴリズムに対し，その正しさを自動的に検証する手法を開発した．抽象度の高いシステムモデルを仮定した場合，対象システムが計算機 10 台程度の規模であれば，プログラムを用いて機械的に検証が可能ことを実験的に示した．

研究成果の概要（英文）：A mechanical verification approach is proposed to verify consensus algorithms, which are core algorithms that can be used in implementing fault-tolerant distributed systems. Experiment results show that when an abstract distributed system model is assumed, our approach can scale up to around ten computing nodes.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2008 年度	1,000,000	300,000	1,300,000
2009 年度	700,000	210,000	910,000
2010 年度	900,000	270,000	1,170,000
年度			
年度			
総　計	2,600,000	780,000	3,380,000

研究分野：総合領域
科研費の分科・細目：情報学・ソフトウエア
キーワード：仕様記述・仕様検証

１．研究開始当初の背景

ネットワーク上にディペンダブルな（高信頼な）情報システムを構築する場合，非同期の耐故障分散アルゴリズムが，極めて重要な役割を果たすことが知られている．ここで非同期とは，コンピュータの実行速度やメッセージ通信遅延のばらつきがどれだけ大きくなっても，正しく動作するという意味である．たとえば，典型的な耐故障手法であるリプリケーションでは，ネットワーク上の複数のコンピュータによって同一の処理を実行することで，障害が発生した場合でも処理を継続することができる．この時，すべてのコンピュータが，同じ順序で同じ処理を実行するという特性が保障されなければならない（そうでなければ状態の不整合が発生する）．ネットワークが不安定であっても，また故障が生じた場合でも，この特性は常に満たされる必要がある．コンセンサスアルゴリズムは，このような特性を実現する耐故障分散アルゴリズムである．近年実用化が進んでおり，Chubby システム(Google 社のファイルシステム)，Postgres-R（データベース PostgreSQL のリプリケーション版)などの応用が知られている．しかし，並行性，非同期性により，コンセンサスアルゴリズムを誤りなく設計

することは非常に難しい．

一方，アルゴリズムの正しさを検証する手法として，モデル検査が近年注目を浴びている．モデル検査は，現在最も広く利用されている形式的検証技術であり，システムの動作を状態機械で表現し，その状態を探索することで検証を行う．申請時において，我々はHO(Heard-Of)モデルという抽象度の高い分散システムのモデルを仮定して，非同期コンセンサスアルゴリズムのモデル検査に成功していた．しかし，検証可能なアルゴリズムの規模（システム中のプロセス（計算機）の数）は3プロセス程度であった．

２．研究の目的

これまでの研究成果を発展させることで，非同期耐故障分散アルゴリズムに対する自動的検証手法の確立を目指した．具体的には，以下の三つの目的を掲げた．

(1) スケーラビリティの向上：検証規模の拡大．

これまでに開発した方法では，アルゴリズムが想定しているシステムの規模が，コンピュータ3台もしくは4台という小規模な場合についてしか，検証を行うことができなかった．（この規模は，実質的に 1 台のコンピュータの故障に耐えられるシステムに相当する．）本研究では，状態機械へのモデル化手法を工夫することで，スケーラビリティの向上を図る．当初の目標としては，7台程度（この場合，3 台の故障まで耐えられる）の規模のシステムに対するアルゴリズムの検証を実現することを考えた．

(2) 通常の分散計算モデルへの拡張：抽象度の高い分散システムのモデル(HO モデル)から，抽象度の低いソースコードレベルのモデルへの検証手法の適用．

申請時に開発していた検証手法では，HO(Heard-Of)モデルという分散計算モデルに検証対象を制限していた．この制約を外し，通常の分散システムモデル上のアルゴリズムを検証できるように手法を拡張することを目標とした．

(3) 擬似コードからの自動モデル生成：分散アルゴリズムの開発者が，容易に検証を行う環境の実現．

一般に，モデル検査ツールでは，独自の入力言語を用いている．これらの言語は，必ずしも，アルゴリズムの設計者が使いやすいものとはいえない．そこで，一般によく使われている擬似コードで記述された検証対象のアルゴリズムを，自動的に検証するツールを作成することで，開発した手法の実用性を高めることを目標とした．

３．研究の方法

(1) スケーラビリティの向上：アルゴリズムの動作全体を一度に検証する従来の方法を改め，モデル検査とインダクションを組み合わせることで，モデル検査する動作の範囲を限定する手法を考案した．限定した動作範囲の探索は，SMT ソルバによって充足可能性判定を行うことで実行する．

より具体的には，コンセンサスアルゴリズムは，ラウンドとよばれる一連の動作を繰り返すことで動作するので，この機構に注目した．一つのラウンドは，数ステップ程度の動作からなるので，これらの動作を数式として表すことが可能である．また，数式における変数として上下限のない整数変数を用いることで，無限の状態を表現することが可能となる．このような考え方に基づき，任意の１ラウンドの動作を数式として表現することを試みた．一度，このように数式として動作が表現できれば，あるラウンドで与えられた性質が成り立つかどうかは，数式の充足可能性を判定することで機械的に検証（モデル検査）できる．この判定にはSMT ソルバとよばれるツールが利用できる．

インダクションは，以下のように適用する．まず，第1ラウンドで望ましい性質が成り立つかどうか，上記のモデル検査によって判定する．次に，前のラウンドで性質が成り立っていたことを仮定したときに，１ラウンドの実行で，その性質が成り立ち続けるかどうかを，同様にモデル検査で検証する．最初のラウンドで性質がなりたつこと，および，前のラウンドで性質が成り立つなら次のラウンドでも成り立つことが判定できれば，機能的にすべての動作でその性質が成り立つことが判定できる．

(2) 通常の分散計算モデルへの拡張：(1)で考案した，モデル検査とインダクションを組み合わせを，通常の非同期分散システムのモデルに適用した．この適用には，抽象度の高い HO モデルにおける仮定に依存した部分の改変が必要であった．

具体的には，HOモデルでは，アルゴリズムはメッセージを待ってブロックすることはないという仮定をおいており，この仮定によりアルゴリズムの進行性，すなわち，プロセスがブロックされず，ラウンドは必ず有限ステップで終了することが保証されるので，モデル化が容易になっていた．この仮定がない場合でも検証を行えるようにすることを試みた．

たとえば，プロセスの状態としてブロック中であることをあらわす状態を導入するなどし，最終的に，抽象度の低い，通常の分散システムモデル上のアルゴリズムを検証で

きるようにした.

(3) 擬似コードからの自動モデル生成:擬似コードをそのまま表現できる計算機言語を設計し，擬似コードで記述された検証対象のアルゴリズムを入力することで，自動的に正しさを検証することができる手法を開発した．分散システムのモデルとしては，HOモデルを仮定した.
ただし，任意の擬似コードを機械的に処理することは不可能なので，擬似コードに近い，繰り返し構造のない手続き的な言語を設計した．繰り返し構造がないのは，HOモデルでは，任意のラウンドがあらかじめ定められた有限数のステップで実行されることが保証されているためである.

４．研究成果
(1) HO モデルを仮定し，モデル検査とインダクションを組み合わせる手法を，4 種類のアルゴリズムに対して適用し，包括的な実験的結果を得ることができた．具体的には，アルゴリズムによって異なるが，7 プロセス(コンピュータ)から 13 プロセスの規模のシステム上での動作に関して，アルゴリズムの正しさを検証することができた．また，活性という性質の検証に限定すれば，どのアルゴリズムについても 14 プロセスの規模の検証が可能であった.
初期の結果については，雑誌論文④で，最終的な結果については雑誌論文①にて報告している.

(2) 現実のプログラムレベルに近い言語（具体的には PROMELA）でアルゴリズムを記述し，より抽象度の低いモデルでのアルゴリズムの動作を表現した．このようにしてモデル化した 2 つのよく知られたアルゴリズム(Chandra-Toueg アルゴリズムと Mostefaoui-Raynal アルゴリズム)を対象に，3 プロセスからなるシステムを仮定し検証を行い正しさを示した．また，不具合を混入したアルゴリズムを用いて，実際に不具合の検出が可能なことを示した.
抽象度の低いモデルに対するこれらの成果は，学会発表①，②において発表した.

(3) 設計した言語に従って記述されたアルゴリズムを，モデル検査システム SPIN の入力言語 PROMELA，および，SMT ソルバ Yices の入力言語に自動的に変換するプログラムを開発した．前者の言語に変換した場合，SPIN を適用することで，アルゴリズムの自動検証が可能になる．後者の場合，スケーラビリティの向上を目的に開発したモデル検査とインダクションを組み合わせる手法が利用できる.

これらの成果については，雑誌論文②，③，初期の結果については，学会発表③にて発表した.

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 4 件）

① Tatsuhiro Tsuchiya, Andre Schiper, Verification of Consensus Algorithms Using Satisfiability Solving, Distributed Computing, 査読有, 23 (2011), 341-358.

②Takahiro Minamikawa, Tatsuhiro Tsuchiya, Tohru Kikuno, Towards Automated Verification of Distributed Consensus Protocols, Proceedings of 16th Asia-Pacific Software Engineering Conference (APSEC 2009), 査読有 (2009), 499-506.

③ Takahiro Minamikawa, Tatsuhiro Tsuchiya, Tohru Kikuno, Language and Tool Support for Model Checking of Fault-Tolerant Distributed Algorithms, Proceedings of 11th International Symposium on Pacific Rim Dependable Computing, 査読有, (2008), 40-47.

④ Tatsuhiro Tsuchiya, Andre Schiper, Using Bounded Model Checking to Verify Consensus Algorithms, Lecture Note on Computer Science, 査読有, 5218(2008), 466-480.

〔学会発表〕（計３ 件）

① Tatsuya Noguchi, Tatsuhiro Tsuchiya, Tohru Kikuno, Model Checking of Unbounded Rounds of Asynchronous Consensus Protocols, Workshop on Dependability of Network Software Applications 2010, 2010.11.18, 広島大学（広島県）.

② Tatsuya Noguchi, Tatsuhiro Tsuchiya, Tohru Kikuno, Safety Verification of Asynchronous Consensus Algorithms Using Model Checking, 2nd International Workshop on Reliability, Availability, and Security (WRAS), 2009.12.11, 広島大学（広島県）.

③野口達也，土屋達弘，菊野亨，モデル検査を用いたコンセンサスアルゴリズムの合意性検証，電子情報通信学会ディペンダブルコンピューティング研究会，2009.10.13，機械振興会館（東京）.

６．研究組織

(1)研究代表者

土屋　達弘（TSUCHIYA TATSUHIRO）
大阪大学・大学院情報科学研究科・准教授
研究者番号：30283740

(2)研究分担者

（　　）

研究者番号：

(3)連携研究者

（　　）

研究者番号：