モバイルクラウドにおける最適データ分割と適応性能を有した高信頼化技術 の開発 (継続)

研究代表者 田村慶信 山口大学 大学院創成科学研究科 准教授 共同研究者 山田茂 鳥取大学 大学院工学研究科 教授

1 はじめに

現在,オープンソースソフトウェアは世界中の民間組織や公的組織などにおいて利活用されている.特に,データの一元管理,低コスト,保守・運用が容易といった観点から,OpenStackやEucalyptusなどのオープンソースソフトウェアを利用したクラウド環境の構築に注目が集まっている.

多くのオープンソースソフトウェアは、それぞれ異なるライセンスを有しているため、利用の妨げとなるケースもあるが、利用条件を満たしていれば、短納期、コスト削減、標準化といった観点から、非常に有用なツールとして活用できる。しかしながら、プロプライエタリ・ソフトウェアと比較して、その開発形態の違いから、品質上の問題点が指摘されている。オープンソースソフトウェアは、主に、バグトラッキングシステムと呼ばれるデータベース上でソフトウェアフォールトの管理が行われている場合が多い。一般的なプロプライエタリ・ソフトウェアにおいては、テスト工程においてある程度一定の品質が保たれた状態でユーザにリリースされている。一方で、オープンソースソフトウェアの場合は、その開発プロセスもオープンソースプロジェクトによって様々であり、テスト工程に類似したプロセスは存在しているものの、明確なテスト工程が存在していないケースもある。オープンソースソフトウェアの品質について考えた場合、最近では、企業主導の下でオープンソースプロジェクトが管理されるケースも増加しており、そのようなオープンソースソフトウェアでは一定の品質が保持されているケースも存在している。しかしながら、多くのオープンソースソフトウェアでは、依然バグトラッキングシステムに基づいて品質上におけるデータ管理が行われているのが現状である。

これまでに、ソフトウェアの信頼性を評価するための数理モデルとして、数百におよぶソフトウェア信頼性モデルが提案されてきた[1-3].また、オープンソースソフトウェアを対象としたソフトウェア信頼性モデルに関する研究も行われている[4].既存のソフトウェア信頼性モデルを適用するためには、バグトラッキングシステム上に登録されているデータから、ソフトウェア信頼性モデルに適用可能なフォールト発見数データやフォールト発生時間間隔データなどに編集する必要があった。さらに、バグトラッキングシステム上には、修正されたフォールトや未修正のフォールトなど様々なデータが混在しており、こうしたデータの全てをソフトウェア信頼性モデルに適用可能なように抽出する作業も必要となる。オープンソースソフトウェアの開発現場でソフトウェア信頼性モデルを適用することは、実利用上の観点から考えた場合、ソフトウェア信頼性モデルに知識のある開発管理者が分析作業を進める必要がある。特に、バグトラッキングシステム上には、開発管理者によって登録されたデータだけではなく、一般ユーザによって報告されたデータも登録されている。全てのデータが、オープンソースソフトウェアに精通したプロジェクト管理者によって登録されている。全てのデータが、オープンソースソフトウェアに精通したプロジェクト管理者によって登録・管理されていれば良いが、例えば、フォールト重要度やコンポーネントといったような分類データは、一般ユーザによって登録されたあとに修正されるケースもあり、様々な登録者が存在しているため、誤ったデータが存在しているケースも少なくない。

本研究では、クラウドを管理運用するソフトウェア管理者が数理モデルに関する知識がなくとも容易に利用できるように、過去に提案された確率モデルに基づく信頼性評価法をアプリケーションソフトウェアとして実装するとともに、フォールトビッグデータの解析に役立つソフトウェアツールを開発した.

さらに、クラウドやビッグデータに関しても、最近では、OpenStack や Hadoop に代表されるようにオープンソースソフトウェアが活用されている。こうしたオープンソースソフトウェアは、バグトラッキングシステムにおける比較的大規模なフォールトデータに基づいて管理されていることから、これらのフォールトデータを分析することは、ビッグデータを有するクラウドソフトウェアの高信頼化に必要不可欠である。本研究では、こうした問題にも着目し、フォールトビッグデータの可視化を実現するために、以下のような特徴をもつアプリケーションソフトウェアを開発してきた。

2 最適メンテナンス問題

過去の提案モデルに基づいた最適メンテナンス問題を提案し、ソフトウェアツールとして実装してきた. 具体的には、従来の最適リリース問題[5,6]を応用した最適メンテナンス問題について議論した.特に、ビッグデータを想定したクラウドコンピューティングに対するジャンプ拡散過程モデルに基づく総ソフトウェアコストを定式化し、コンテンツデータとクラウドベースデータとの関係性を考慮した最適メンテナンス時刻の推定法を提案した.

3 フォールトビッグデータの分析

一般的に、オープンソースソフトウェアはバグトラッキングシステムと呼ばれるデータベース上でフォールトデータが管理されている。バグトラッキングシステムには、Bugzillaに代表されるオープンソース系のものと、BugListerのようなプロプライエタリなものが存在している。フォールトデータとして一般的に登録されているデータ項目としては、以下のようなものが挙げられる。

【データ項目の一例】

バグ ID, ハードウェア, 0S, プロダクト名, コンポーネント名, 報告者名, 報告者のニックネーム, 修正者名, 修正者のニックネーム, フォールト状況, 修正状態, 登録日時, 修正日時, キーワード, コメント数, 優先度, フォールト内容の要約, フォールトの詳細内容, ターゲットマイルストーン, バージョン情報, 投票情報, 重要度

一般的なバグトラッキングシステム上には、上述したようなデータが登録されている. これらのデータをフォールト分析のために可視化することができれば、開発管理者だけでなくユーザにとっても有益な情報を提供できるものと考える.

これまでに、特定の組織に限定されたバグ管理ツールや、それをグラフ化するものはいくつか存在しているが、Bugzillaに代表されるオープンソース系のバグトラッキングシステムを対象としたフォールト分析ツールは存在していない。特に、データ項目別の分類表示や、あるコンポーネントのみを対象とした時系列分析のように、詳細に分析できるツールはなく、こうしたデータの可視化が容易にできれば、開発者やユーザにとっても、以下のような判断材料として利用できるものと考える。

- オープンソースソフトウェアを開発しているオープンソースプロジェクトサイトには、複数のバージョンのソフトウェアが登録されており、これらのバージョンの中からどれを選べば良いかといった選択の問題.
- 複数のコンポーネントから構成されているオープンソースソフトウェアの場合, その中から複数のパッケージを選択して利用するケースにおいて, 特定コンポーネントのフォールト出現頻度から判断し, パッケージの選択肢から除外するなどして運用することが可能.
- 重要度の高いフォールトの出現傾向や頻度に応じたパッケージの取捨選択やバージョン選択が可能.

4 アプリケーションの開発

4-1 要求仕様定義

開発されたツールの要求仕様の一例を以下に示す.

- 本アプリケーションソフトウェアにおいて使用するデータは、オープンソースプロジェクトの バグトラッキングシステム上において登録されているデータを用いる.
- バグトラッキングシステム上おいて登録されているデータに基づき,各登録データに対する時系列分析結果をグラフ表示する.
- データ処理を行うために統計言語 R を使用する.
- アプリケーション開発言語として HTML5, CSS3, および JavaScript を用いる.

- アプリケーションの動作にアニメーションを適用する.
- アプリケーションの操作には GUI を使用し、マウスを用いて行う.
- アプリケーション開発の際には、NW. js[7]によりクロスプラットフォーム実行環境を構築する. NW. js は、MIT License の下で開発・公開されている.
- アプリケーションのグラフ描画には, JavaScript ライブラリとして知られている NVD3[8]チャートコンポーネントを利用する. NVD3 は, Apache License (Version 2.0) の下で開発・公開されている.
- 本ソフトウェアは、Web アプリケーションとしてだけではなく、クロスプラットフォームによるネイティブアプリケーションとしても動作する.

4-2 実行手順

本アプリケーションソフトウェアの実行手順を以下に示す.

- バグトラッキングシステム上からデータを取得する.
- 本アプリケーションソフトウェア上でデータを読み込む.
- アプリケーションソフトウェア上でデータ処理を行う.
- バグトラッキングシステムから取得された各データ項目に対するグラフを表示する.

開発されたアプリケーションソフトウェアの特徴としては、NW. js によりクロスプラットフォーム実行環境により構築されている点にある。これにより、Web アプリケーションとしてブラウザから実行でき、バグトラッキングシステム上における可視化されたデータを Web 経由で公開することが可能となる。さらに、同じソースコードを利用して、クロスプラットフォームなデスクトップアプリケーションとしても利用可能であることから、様々な環境で利用することができる。

5 アプリケーションの実行例

ジャンプ拡散過程モデルに基づくディペンダビリティ評価ツールを実行した際におけるメイン画面を図1に示す. さらに、機能拡張としての最適メンテナンス問題に関する実行例を示す. 推定された総ソフトウェアコストのサンプルパスを図2に示す. 図2から、最適メンテナンス時刻は153.3 目となり、そのときの総ソフトウェアコストは1049.2 であることが確認できる. また、図2からフォールト報告終了時点以降から200日目までは雑音が大きく、その後は時間の経過とともに徐々に小さくなる様子が確認できる. このことから、従来の期待値に基づく最適メンテナンス時刻[5,6]よりも50日経過後にメンテナンスを行うことが望ましいといえる.

次に、開発されたフォールトビッグデータ分析のためのアプリケーションソフトウェアによる実行結果を示す. ソフトウェアコンポーネント名、ソフトウェアバージョン、フォールト報告者、およびフォールト修正者に対する分析結果を図 3~図 6 に示す. また、Apache HTTP サーバ[9]における主要バージョンのリリース時期を以下に示す.

- バージョン 2.0 (2002年4月6日)
- バージョン 2.2 (2005年12月1日)
- バージョン 2.4 (2012年2月21日)

図3~図6から,全体的には同じような検出傾向である様子が確認できる.主要バージョンにおけるリリース時期と対比させた場合,各バージョンのリリース日時の前後でフォールト発見頻度が比較的大きくなっている様子が分かる.特に,バージョン 2.2 のリリース前おいては,バージョン 2.2 に関係するフォールトがリリースへ向けて継続的に多く検出されている.

さらに、フォールト重要度に関する分析結果を図7に示す。図7から、normal レベルのフォールトが多くの割合を占めている様子が確認できる。特に、major レベルのフォールトが、バージョン 2.2のリリースされた 2005 年 12 月 1 日以降に集中的に検出されている様子が分かる。さらに、 critical レベルのフォールトに関しては、2005 年 12 月 1 日にバージョン 2.2 が公開されて以降から、徐々にフォールト検出数が減少している様子が確認できる。

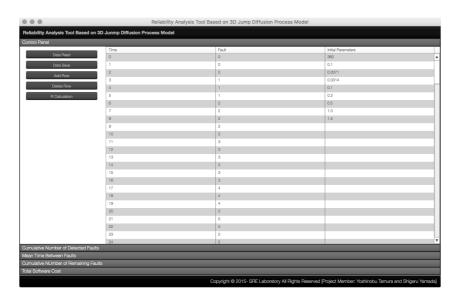


図1:ツールのデータ入力画面.

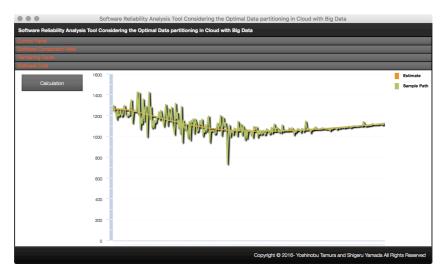


図2:総期待ソフトウェアコストの推定結果.

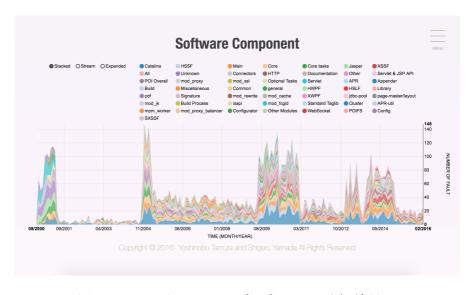


図3:ソフトウェアコンポーネントの分析結果.

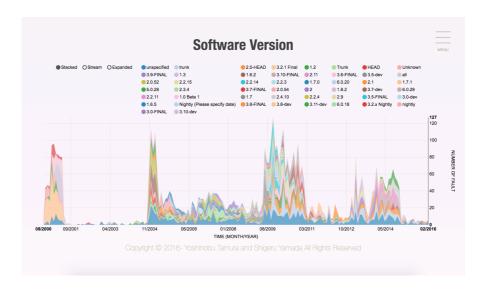


図4:ソフトウェアバージョンの分析結果.

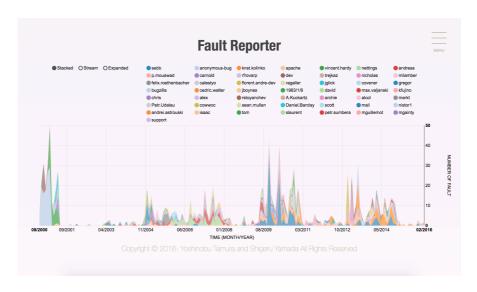


図5:フォールト報告者の分析結果.

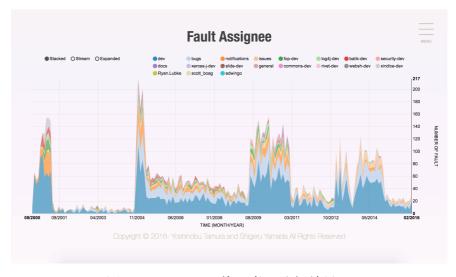


図6:フォールト修正者の分析結果.

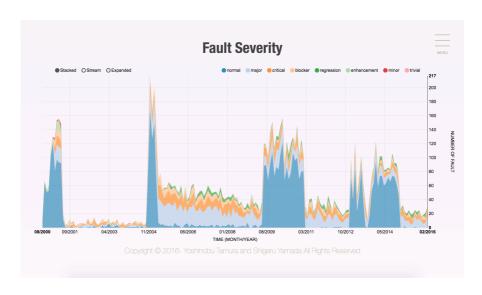


図7:フォールト重要度の分析結果.

6 おわりに

本研究では、これまでに提案されたジャンプ拡散過程モデルに基づく最適メンテナンス問題について議論し、信頼性評価ツールの追加機能として実装した。また、実測データに基づくソフトウェアツールの実行例を示すとともにツールの有効性について考察した。さらに、フォールトビッグデータ分析のためのアプリケーションソフトウェアも実装した。

特に、多くのオープンソースソフトウェアは、バグトラッキングシステムのようなフォールトデータベースを利用して開発が行われている。バグトラッキングシステム上には、多くのフォールトデータが蓄積されており、これらのフォールトデータの全体像をその分析のために可視化できるようになれば、オープンソースソフトウェアの開発者だけではなく、それを利用するユーザ側にも有益な情報を提供することが可能となる。こうしたフォールトデータを分析することは、ビッグデータを有するクラウドソフトウェアの高信頼化にも必要不可欠である。本研究では、実際のオープンソースソフトウェアのバグトラッキングシステム上に登録されているデータから、種々のフォールト要因に対するアプリケーションソフトウェアの実行例を示した。これらの結果から、主要バージョンのリリース時期とフォールト発見における関連性について考察した。

【参考文献】

- [1] M.R. Lyu, ed., Handbook of Software Reliability Engineering, IEEE Computer Society Press, Los Alamitos, CA, 1996.
- [2] S. Yamada, Software Reliability Modeling: Fundamentals and Applications, Springer-Verlag, Tokyo/Heidelberg, 2014.
- [3] P.K. Kapur, H. Pham, A. Gupta, and P.C. Jha, Software Reliability Assessment with OR Applications, Springer-Verlag, London, 2011.
- [4] S. Yamada and Y. Tamura, OSS Reliability Measurement and Assessment, Springer-Verlag, London, 2016.
- [5] S. Yamada and S. Osaki, Cost-reliability optimal software release policies for software systems, IEEE Transactions on Reliability, vol. R-34, no. 5, pp. 422-424, 1985.

- [6] S. Yamada and S. Osaki, Optimal software release policies with simultaneous cost and reliability requirements, European Journal of Operational Research, vol. 31, no. 1, pp. 46-51, 1987.
- [7] NW.js community, NW.js, http://nwjs.io/
- [8] Novus Partners, NVD3, http://nvd3.org/
- [9] The Apache Software Foundation, The Apache HTTP Server Project, http://httpd.apache.org/

〈発表資料〉

題 名	掲載誌・学会名等	発表年月
Software reliability model selection based on deep learning	Proceedings of the International Conference on Industrial Engineering, Management Science and Application 2016, Jeju Island, Korea, pp. 77-81	May 23-26, 2016
オープンソースソフトウェアに対するディープラーニングに基づくフォールト識別法	電子情報通信学会 技術研究報告 [信頼性], ウインクあいち, Vol. 116, No. 69, pp. 33-38	2016年5月28日
Identification method of fault level based on deep learning for open source software	Software Engineering Research, Management and Applications, Studies in Computational Intelligence, Springer International Publishing Switzerland, pp. 65-76	June 2016
Reliability and interaction analysis based on fault data clustering and neural network for cloud software with big data	Amity Journal of Interdisciplinary Research, Amity University Press, pp. 75-80	July 2016
オープンソースソフトウェアに対するフォ ールトデータ分析ツールの開発	電子情報通信学会 技術研究報告 [信頼性], 小樽経済センター, Vol. 116, No. 168, pp. 7-12	2016年7月29日
Deep learning approach for reliability assessment of cloud software	Proceedings of the 22nd ISSAT International Conference on Reliability and Quality in Design, Los Angeles, California, USA, pp. 138-142	August 4-6, 2016
Jump diffusion process model considering the optimal data partitioning for cloud with big data	Proceedings of the 22nd ISSAT International Conference on Reliability and Quality in Design, Los Angeles, California, USA, pp. 167-171	August 4-6, 2016
Open source software reliability assessment based on deep learning	Proceedings of the 7th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling, Seoul, Korea, pp. 509-516	August 24-26, 2016
OSS フォールトデータ分析ツールの開発と その考察	プロジェクトマネジメント学会 2016 年度 秋季研究発表大会予稿 集,広島修道大学,pp. 207-208	2016年9月1-2日
オープンソースソフトウェアに対するディープラーニングに基づくフォールト分析に 関する比較と考察	プロジェクトマネジメント学会 2016 年度 秋季研究発表大会予稿 集,広島修道大学,pp. 224-225	2016年9月1-2日

モバイルクラウドに対する影響要因を考慮 したハザードレートモデルに基づく信頼性 管理	プロジェクトマネジメント学会 2016 年度 秋季研究発表大会予稿 集,広島修道大学,pp. 233-234	2016年9月1-2日
Reliability analysis based on deep learning for fault big data on bug tracking system	Proceedings of the IEEE International Conference on Reliability, Infocom Technology and Optimization, Amity University, Uttar Pradesh, Noida, India, pp. 37-42	September 7-9, 2016
Quantitative interdependency analysis based on deep learning for cloud and database software	Proceedings of the Thirteenth International Conference on Industrial Management, Hiroshima, Japan, pp. 325-331	September 21-23, 2016
Software reliability model selection based on deep learning with application to the optimal release problem	Journal of Industrial Engineering and Management Science, DOI: 10.13052/jiems2446-1822.2016.0 03, Vol. 1, pp. 43-58	October 2016
ソフトウェア信頼性評価と最適リリース時 刻推定のためのモバイルアプリケーション 開発	第18回 IEEE 広島支部 学生シンポ ジウム論文集,山口大学工学部 常 盤キャンパス,pp. 188-191	2016年11月19-20日
クラウドデータの運用環境を考慮したソフ トウェア信頼性評価法	第18回 IEEE 広島支部 学生シンポ ジウム論文集,山口大学工学部 常 盤キャンパス,pp. 362-365	2016年11月19-20日
Comparison of big data analyses for reliable open source software	Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Bali, Indonesia, CD-ROM (Reliability and Maintenance Engineering 3)	December 4-7, 2016
Software reliability and cost analysis considering service user for cloud with big data	International Journal of Reliability, Quality and Safety Engineering, World Scientific	to be published