

環境適応のためのソフト自動分割初期検討

An initial study of automatic software division for environmental adaptation

山登庸次
Yoji Yamato

日本電信電話（株） ネットワークサービスシステム研究所
Network Service Systems Laboratories, NTT Corporation

1. はじめに

近年, IoT ([1]-[6]) 等様々な領域で, ヘテロなハード利用が増えているが, 活用の壁は高い. 私は, 環境に合わせて, 既存のコード変換等を自動で行い適切に動作させる, 環境適応ソフトウェアを提案してきた[7]-[9].

変換等は, 既存のコードを解析するため OSS 等がほぼ前提だった. 本稿では, 取り組む自動分割技術により, 汎用的プログラムに容易に機能追加できるようにする. これにより, OSS 等にユーザが独自の機能を追加したプログラム等, 環境適応の対象を大きく増やすことができる.

2. 汎用的プログラムへの機能追加について

環境適応は, 既存のコードを解析変換するため, OSS 等のソースコードが必要であった. しかし, OSS は機能不十分でユーザは自分用のカスタマイズをしたい場合も多い. 一方, 環境適応では, IoT サービスを対象に, ユーザは基本サービスとユーザ独自処理と IoT 機器を指定すれば, データ蓄積, 表示等の共通機能を探して組み合わせ, IoT サービス化する IoT 適応方式を実現したが, 対象は IoT サービスに限定されている.

そこで, 本稿は IoT 以外の汎用的プログラムにユーザが機能追加することを容易化することを目指す. それなりに行数があるアプリケーションは, 変更影響が広範囲に及ぶことが多く, 追加変更には関連する機能に影響がないかのチェックに大きな稼働がかかる. そこで, アプリケーションを関連する処理で分割して, 分割境界を元に変更を局所化することで, 適材適所利用できるようにするとともに, サービス追加変更を容易化する.

3. 汎用的プログラムの自動分割検討

IoT 適応方式では, コード分析し, IoT GW や IoT PF を適材適所で利用できるが, IoT サービス以外の汎用的プログラムを適材適所で利用することはできない. そこで, 分析を高度化し, 汎用的プログラムでも, ユーザが行いたい独自処理を追加変更できるようにする事が望まれる.

分析する手法で, アプリケーションを実際には動かさずに, ソースコードの関数の呼び出しや書き込み等のリレーションを見る, 静的分析手法がある. 静的分析では, 関数同士の呼び出し関係が把握できる, ある関数が同じデータに書き込む関数かどうか把握できる. 一般に, 呼び出し関係があったり, 同じデータに書き込みそれを使う関数は, 関連が深いので, 分割する際にはグループ化する必要がある. ただ, アプリケーションを動かさない静的分析は, ユーザに実際には使われないようなケースも分析される形となる.

一方, 分析する手法で, サンプルテストケースを用いてアプリケーションを実際に動かし, 実行されたログ等の情

報を見る, 動的分析手法がある. 動的分析では, 関数同士の呼び出し関係でも, ユーザが指定するサンプルテストケースで実際に使われる関数の呼び出し関係が抽出できる. DB を用いるアプリケーションの場合, DB アクセスログを抽出して, 連続的に実行しているデータアクセス命令を発見し, 一連処理として実行しているプログラム範囲を見つけることができる. また, DB でなくファイルアクセスの場合でも, ファイルアクセスログから, 一連の処理として実行しているプログラム範囲を見つけることができる. ただ, 動的分析で実際に動かすテストケースはサンプルの数に依存する形となる.

環境適応では今まで, GPU 自動オフロード等に取り組んできたが, GPU で計算処理可能かは静的分析で分かるが, GPU 処理した際の性能は実際に測定しないと分からないのが通常であり, 静的分析と動的分析を組み合わせ, オフロード部を自動探索していた. 動的分析は, ユーザが使うサンプルテストケースを実際に動かし性能測定するため, 個々のユーザ毎に異なる対応をするために重要な要素であった. そこで, 汎用的プログラムの自動分割でも, 個々のユーザに対応するために, 静的分析と動的分析を合わせた分析を行う.

4. まとめ

本稿では, 環境適応新要素として, 汎用的プログラムの自動分割方式を検討した. これにより, 環境適応対象を増やすことができる. 具体的な分割方式を検討し, 実装し, 分割による確認行数の変化等を比較し有効性を検証する.

参考文献

- [1] H. Noguchi, et al., "Distributed Search Architecture for Object Tracking in the Internet of Things," IEEE Access, 2018.
- [2] H. Noguchi, et al., "Device Identification Based on Communication Analysis for the Internet of Things," IEEE Access, DOI: 10.1109/ACCESS.2019.2910848, Apr. 2019.
- [3] H. Noguchi, et al., "Autonomous Device Identification Architecture for Internet of Things," IEEE WF-IoT 2018, 2018.
- [4] Y. Yamato, et al., "Proposal of Real Time Predictive Maintenance Platform with 3D Printer for Business Vehicles," International Journal of Information and Electronics Engineering, Vol.6, No.5, pp.289-293, 2016.
- [5] Y. Yamato, et al., "Method of Service Template Generation on a Service Coordination Framework," UCS 2004, Nov. 2004.
- [6] Y. Yamato, "Automatic Verification for Plural Virtual Machines Patches," ICUFN 2015, pp.837-838, 2015.
- [7] Y. Yamato, "Automatic Offloading Method of Loop Statements of Software to FPGA," Int J Parallel Emergent Distrib Syst., Taylor and Francis, Apr. 2021.
- [8] Y. Yamato, "Improvement Proposal of Automatic GPU Offloading Technology," ICIET 2020, pp.242-246, Mar. 2020.
- [9] Y. Yamato, "Proposal of Automatic Offloading for Function Blocks of Applications," ICIAE 2020, pp.4-11, Mar. 2020.

抄録：

私は、環境に合わせて、既存のコード変換等を自動で行い適切に動作させる、環境適応ソフトウェアを提案してきた。変換等は、既存のコードを解析するため OSS 等がほぼ前提だった。本稿では、取り組む自動分割技術により、汎用的プログラムに容易に機能追加できるようにする。

キーワード：

環境適応ソフトウェア、汎用的プログラム、機能追加、自動分割、動的分析

Keywords：

Environment-adaptive software, General-purpose programs, Function addition, Automatic division, Dynamic analysis

連絡先：

山登庸次

NTT ネットワークサービスシステム研究所

yoji.yamato@ntt.com