

ソフト分割の IoT サービス適用初期検討

An initial study of the IoT service application for software split

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]-[5]) 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[6]-[9].

これまで検証してきた環境適応の自動分割方式は, C 言語の計算系アプリに限られていた. 本稿では, IoT サービスに自動分割方式を適用することで, これまで事前に事業者の基本サービス提供が必要だった IoT サービスでも, ユーザが独自変更を容易化できるようにする.

2. 環境適応の自動分割適用先について

以前に著者は, 環境適応ソフトウェアの処理と全体像を提案した. 環境適応ソフトウェアの処理は, Step1-6 は, コード分析し, 配置環境に応じたコード変換, リソース量調整, 配置場所調整, 検証の一連を行い, アプリケーション運用を開始する. Step1-6 の流れを, GPU や FPGA 等のアクセラレータ, IoT 機器等の少リソース端末の両方で検証してきた. 更に, コード分析できるアプリケーションを増やすため, ユーザが行いたい処理を追加変更する際に, アプリケーションを関連処理に基づいて分割して, 変更を局所化して容易化する, 自動分割方式を提案している.

本稿の課題を整理する. 著者は環境適応ソフトウェアのコンセプトを以前提案し, GPU や FPGA 等アクセラレータにプログラムを自動オフロードする方式や, ユーザは基本サービスと処理したい計算処理コードと IoT 機器を指定すれば, 自動でサービス化する IoT 適応方式を検証してきた. また, 汎用的プログラムの, コードの処理関係を元に自動分割して, 変更を局所化することで, ユーザが機能追加や変更をしやすくする方式を検証してきた.

しかし, 汎用プログラムの自動分割はこれまで, C 言語の計算系アプリケーションの検証だけであり, 多くのユーザが使うと思われる IoT 等の新たなサービスは検証されていなく, 有効性が示されていなかった. そこで, 本稿は, IoT サービスでの自動分割を狙いとし, IoT ならではの考慮事項を踏まえて自動方式を微変更して, 検証する.

3. IoT 向け自動分割方式

IoT 向けのプログラムに, ユーザが行いたい独自処理を追加変更することを考える. 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ため, 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, 追加変更を容易化することを行う.

環境適応ソフトウェアの, GPU 自動オフロードの際,

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, IoT サービスの自動分割でも, 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方式をとることにする. IoT サービスでは, IoT データの集約が必ず必要となる特徴となるため, 特にプログラムを分析する際に, 通信が行われる処理の場合は, 必ず別グループに分割して, 別筐体となってよいようにする.

このようにすることで, 動的分析でユーザが指定するサンプルテストケースで呼び出し関係があるファイル群は必ず同じグループとなる. また, 静的分析で全ファイルを分析した際に一定度の回数である 3 回以上呼び出し関係があるファイル群も同じグループになる. Stand Alone 的で他の関数とあまり関わりがない関数のファイルは残りのグループになる. また, IoT で特徴となる IoT データ集約に関わる通信がある場合は別グループになる. 呼び出し関係に基づいて自動分割することで, ユーザが機能追加する際の確認範囲を小さくすることができる.

4. まとめ

本稿では, 環境適応新要素として, プログラムの自動分割方式を IoT サービスに適用する際の方式微変更を検討した. 具体的な分割方式を実装し, Azure IoT PF を用いる IoT サービスを自動分割し, 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., "Autonomous Device Identification Architecture for Internet of Things," IEEE WF-IoT 2018, 2018.
- [3] Y. Yamato, "Automatic Verification for Plural Virtual Machines Patches," ICUFN 2015, pp.837-838, July 2015.
- [4] Y. Yamato, et al., "Method of Service Template Generation on a Service Coordination Framework," UCS 2004, Nov. 2004.
- [5] Y. Yamato, et al., "Fast Restoration Method of Virtual Resources on OpenStack," IEEE CCNC 2015, pp.607-608, 2015.
- [6] Y. Yamato, "Automatic Offloading Method of Loop Statements of Software to FPGA," Int J Parallel Emergent Distrib Syst., Taylor and Francis, Apr. 2021.
- [7] Y. Yamato, "Improvement Proposal of Automatic GPU Offloading Technology," ICIT 2020, pp.242-246, Mar. 2020.
- [8] Y. Yamato, "Proposal of Automatic Offloading for Function Blocks of Applications," ICIAE 2020, pp.4-11, Mar. 2020.
- [9] Y. Yamato, "Study and Evaluation of Improved Automatic GPU Offloading Method," Int J Parallel Emergent Distrib Syst., Taylor and Francis, June 2021.