

混在環境におけるアプリケーション配置の初期検討

An initial study of application deployment in a mixed environment

山登庸次
Yoji Yamato

NTT
NTT Corporation

1. はじめに

近年, IoT ([1]-[4]) 等種々領域で, ヘテロジニアスハードウェア利用が増えているが, 活用の壁は高い. 私は, 配置環境に合わせ, 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[5]-[9].

これまで検証してきた自動オフロードは, GPU や FPGA, 量子コンピュータ等個々ハードウェアに対してだった. 本稿では, それらの混在環境においてアプリケーションを適材適所で利用するための, 自動オフロード方式を検討する.

2. 量子計算機含む混在環境の利用について

著者は以前に, 既存ソフトウェアコードを, 配置先環境で利用できる様, FPGA や GPU 向変換, リソース量設定等自動でし, アプリケーション高速化する, 環境適応ソフトウェアを提案した. 環境適応ソフトウェア要素として, 既存コードのループ文や機能ブロックを, FPGA, GPU, 量子コンピュータ等に自動オフロードする方式等も提案した.

本稿の課題を整理する. ヘテロジニアスハードウェア用いた高速化は専門家の手動が必要である. 私は以前に環境適応ソフトウェアを提案し, FPGA や GPU, 量子コンピュータの自動オフロード方式も実現してきた. しかし, 今までは個別のハードウェアオフロード自動化が目的で, 混在環境への適切なオフロードは十分検討されていなかった. 例えば, GPU で性能 10 倍, FPGA で性能 20 倍も, 利用料で FPGA が GPU の 3 倍ならば, GPU が適切となる. そこで, 本稿は, 通常 CPU, マルチコア CPU, FPGA, GPU, 量子コンピュータが存在する環境の中で, 適切なハードウェアへの自動オフロードを対象とする. 混在環境での適切な自動オフロード方式を提案し, 実ヘテロジニアスハードウェア混在環境を用いて, 提案方式を確認する.

3. 混在環境への適切な自動オフロード

混在環境への自動オフロードは以下で行う. 検証は全て検証環境で行われる. ユーザからアプリケーションのオフロード依頼があったら, 構文解析ライブラリを用いて, アプリケーションを分析する. 個別ハードウェアの自動オフロード試行自体は著者提案の以前方式を用いる. まず, 機能ブロックオフロードである. 抽象構文木類似性元に, 機能ブロックオフロード試行をマルチコア CPU, GPU, FPGA, 量子コンピュータに対して行う. オフロード可能な機能ブロックをコードパターン DB との照合により Deckard で検索し, 見つかった場合ハードウェア処理部分を置換し (Qiskit 実装, CUDA ライブラリ等), 機能ブロックオフロード後の性能を測定する. 性能測定では, 通常 CPU に対して何倍高速になったかを各ハードウェアに対し

取得する. 次に, 機能ブロックオフロードの可否によらずループ文オフロード試行をマルチコア CPU, GPU, FPGA に対して行う. マルチコア CPU, GPU では, 性能測定を反復し, 遺伝的アルゴリズムで高速のループ文オフロードパターンを組み替えながら検索し, 最終解の OpenMP, OpenACC ファイルを見つける. FPGA では, 算術強度やリソース効率等でオフロード候補ループ文を絞り込んでから, 複数パターン性能測定を行い, 最高性能パターンの OpenCL ファイルを見つける. それぞれの最終解に対して性能測定し, 通常 CPU に対する改善度を取得する. 最後に, 全オフロードパターンの何倍かの改善度を, 各ハードウェアの月利用料で割ることで, コストパフォーマンスを求める. 通常 CPU VM の場合よりも高く, 最もコストパフォーマンスが良いハードウェアに対して, 該当アプリケーションを自動オフロードするよう利用料含めてユーザに提案する. ユーザがオフロード利用を了承後, アプリケーションは商用環境に配置され, 実利用が開始される.

提案方式を実装し, 固有値解析問題, NAS.BT, 姫野ベンチマーク, MRI-Q の 4 アプリケーションを, 分析, 性能測定して, それぞれ, 量子コンピュータ, マルチコア CPU, GPU, FPGA に自動オフロードし, 方式有効性を確認する.

4. まとめ

本稿では, 環境適応ソフトウェアの新要素として, 混在環境向けの自動オフロード方式を初期検討した. ヘテロジニアスハードウェア種類が増えている中, 適材適所を行う本技術は有効と考える. Azure Quantum 等を用いて実際のオフロード性能等を確認する.

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, et al., "Fast Restoration Method of Virtual Resources on OpenStack," IEEE CCNC 2015, pp.607-608, 2015.
- [4] Y. Yamato, "Automatic Verification for Plural Virtual Machines Patches," ICUFN 2015, pp.837-838, 2015.
- [5] Y. Yamato, "Automatic Offloading Method of Loop Statements of Software to FPGA," Int J Parallel Emergent Distrib Syst., Taylor and Francis, Apr. 2021.
- [6] Y. Yamato, "Improvement Proposal of Automatic GPU Offloading Technology," ICJET 2020, pp.242-246, Mar. 2020.
- [7] Y. Yamato, "Proposal of Automatic Offloading for Function Blocks of Applications," ICIAE 2020, pp.4-11, Mar. 2020.
- [8] Y. Yamato, "Study and Evaluation of Automatic GPU Offloading Method from Various Language Applications," Int J Parallel Emergent Distrib Syst., Taylor and Francis, Sep. 2021.
- [9] Y. Yamato, "Study and Evaluation of Improved Automatic GPU Offloading Method," Int J Parallel Emergent Distrib Syst., Taylor and Francis, June 2021.