

コンテナ型仮想化によるモデリング教育向け コンパイルサーバでの micro:bit 対応機能の評価

Evaluation of the compilation function for micro:bit on the compile server for modeling education with container-based virtualization

大宅 剛生^{*1}, 香山 瑞恵^{*2}, 永井 孝^{*3}
Kouki Ootaku^{*1}, Mizue Kayama^{*2}, Takashi Nagai^{*3}

^{*1}信州大学大学院総合理工学研究科
^{*1}Graduate school of
Science & Technology,
Shinshu University

*2 信州大学
*2Shinshu University

*³ものづくり大学
*³Institute of Technologist

Email: 20w2013k@shinshu-u.ac.jp

あらまし: 本研究の目的は、モデリング教育支援環境内におけるモデル駆動開発ツールのコンパイルサーバでの `micro:bit` 対応機能の開発である。これまでに、サーバ管理の拡張性確保と効率化向上を目指して、コンパイル環境の `Docker` によるコンテナ型仮想化を行ってきた。本稿では支援環境の概要を述べ、`micro:bit` 対応機能を有したコンパイル環境の評価の結果を示す。

キーワード： コンパイルサーバ, Docker, コンテナ型仮想化, モデル駆動開発, モデリング

1. はじめに

現在，大学の情報系学科や情報の専門学校において，ソフトウェアエンジニアリング領域において「考える力」や「捉える力」をなどの向上をカリキュラムとして設定されている⁽¹⁾．その際，モデル化の能力が重要とされている．本研究では MDD（モデル駆動開発）ツールの clooca⁽²⁾を Lego Mindstorms NXT や Studuino 等のデバイスを対象とした独自コンパイラサーバを付け加えた S-clooca を開発してきた^[3-5]．現在，S-clooca は中学校技術科の正規授業を中心に運用されている．本稿では，このコンパイラサーバにおける micro:bit 対応機能の評価結果を示す．

2. MDD ツールについて

2.1 S-clooca

S-clooca は著者ら⁽³⁻⁵⁾が独自に開発したコンパイルサーバを clooca と統合したモデリング教育学習環境である。S-clooca の動作概要を図1に示す。clooca は

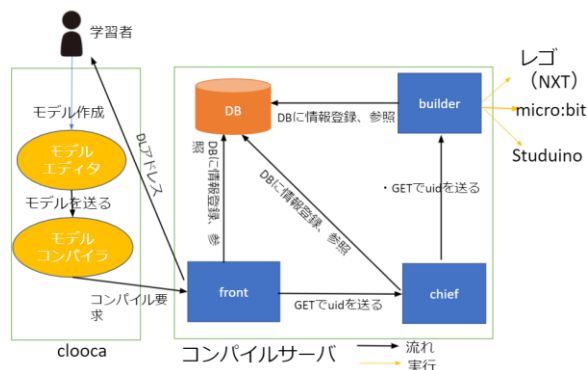


図 1 S-clooca の動作概要図

学習者のモデルから生成されたソースコードのコンパイルをコンパイルサーバに要求する。コンパイルサーバは各デバイス用のコンパイル環境を用いて実行コードを生成する。生成された実行コードをダウンロードするためのアドレスが学習者に返される。

コンパイルサーバは、3 種のアプリケーション：
front, chief, builder からなる。front はコンパイル要求
の受付を担当する。chief はコンパイル処理の負荷分
散を担当する。builder はコンパイル処理を担当する。
front, chief, builder の情報は専用 DB で管理される。
また、clooca は仕様上 front での受付から 6 秒経過す
ると実行コードのダウンロードアドレスが学習者に
返されなくなる。

2.2 コンパイラサーバの問題点と解決策

現状のコンパイラサーバは 3 種のデバイス: Lego Mindstorms NXT, Studuino, micro:bit を対象としている. しかし, 各デバイスのコンパイル環境を共存させるには問題点が 2 つある. 1 つ目はコンパイル環境の拡張性が低い点, 2 つ目は管理負担が大きい点である. そこで, 先行研究⁶⁾において, コンパイラサーバの Studuino 対応の builder に Docker を用いたコンテナ型仮想化を導入し, 問題点の解決を図った.

2.3 Studuino 対応の nDocker 環境と 1Docker 環境

Studuino 対応の builder において、コンパイル要求一つに対して一つコンテナを作る nDocker 環境 (図 2-1 参照) を用いることによって、個々のデバイスのコンパイル環境の独立管理が可能となった。しかし、Docker 導入前の環境と比較して 10 倍以上のコンパイル時間が必要になった。その原因は、コンテナを起動から破棄の部分にあることが分かった。よって、

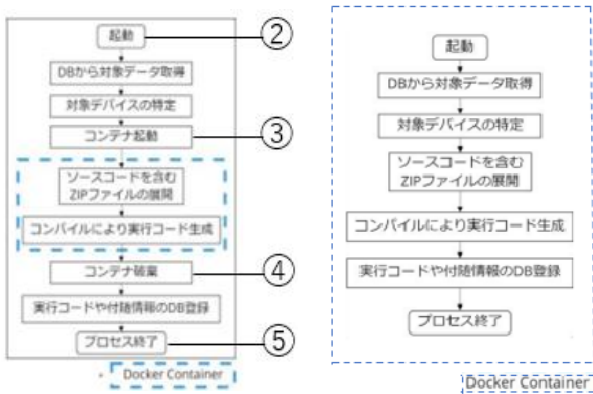


図 2-1 nDocker 環境の処理 図 2-2 1Docker 環境の処理

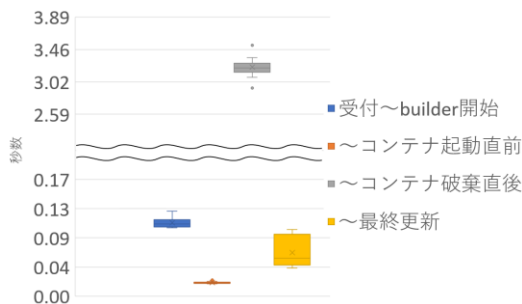


図 3 nDocker 環境のコンパイル総処理時間

コンテナを常時立て builder を仮想化することによって、コンテナ破棄の必要性がない 1Docker 環境(図 2-2 参照)を導入することで、コンパイル時間を短くする改善を図った。

3. microbit 対応の 1Docker 環境の評価

3.1 micro:bit 対応の nDocker 環境と 1Docker 環境

micro:bit 対応の builder において nDocker 環境を用いた場合、利用者から Studuino に比べてコンパイルに時間がかかるという指摘があった。各デバイスのコンパイル時間を整理した結果、Studuino は平均 3.04 秒、micro:bit は平均 12.92 秒であった。

micro:bit 対応の nDocker 環境の builder とコンテナによるコンパイル処理は図 2-1 と同様である。前述の問題点を解決するため、micro:bit における nDocker 環境の処理時間を 5 点(①front 受付、②builder 開始、③コンテナ起動直前、④コンテナ破棄直後、⑤最終更新)で計測した(図 2-1 中丸数字参照)。図中にない①は front がコンパイル要求を受け付けた点とした。

中学校での課題と同程度の複雑度を持つモデル図を同時に 5 つコンパイル要求する動作を 4 回(計 20 回)計測した。図 3 に計測点間の時間を示す。「受付～builder 開始」の平均処理時間が平均 0.11 秒、「～コンテナ起動直前」が平均 0.02 秒、「～コンテナ破棄直後」が平均 3.22 秒、「～最終更新」が平均 0.06 秒であった。コンテナ起動からコンテナ破棄までの時間で総処理時間の 94%を占めていた。

そこで、コンテナの起動・破棄を極力少なくした micro:bit 対応の 1Docker 環境を作成した。1Docker 環

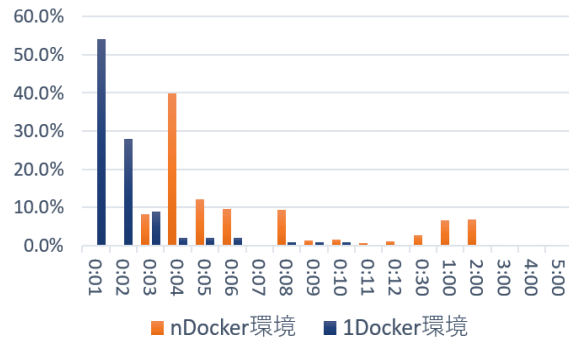


図 4 1Docker 環境と nDocker 環境のコンパイル時間の比較

境の処理フローは図 2-2 と同様である。

3.2 コンパイル時間の評価

micro:bit 対応の 1Docker 環境でのコンパイル時間を計測した。ここでは、3.1 の計測と同程度の複雑度を持つモデル図を用いて 10 人で各 10 回コンパイルした結果を対象とした。

1Docker 環境と nDocker 環境のコンパイル時間の比較結果を図 4 に示す。コンパイル後の実行コードのアドレスの返り率は nDocker 環境 60.2%、1Docker 環境 95.0%と約 35%向上した。しかし、1Docker 環境でもサーバに負荷がかかった際、コンパイル時間が最大で 10 秒かかる場合もあった。

4. おわりに

本稿では、コンパイルサーバの一部を仮想化した環境の運用の評価と結果を示した。具体的には、micro:bit 対応の nDocker 環境においてコンパイル時間が長くなる要因を特定し、1Docker 環境にすることでコンパイル時間を 85%短縮した。今後は、1Docker 環境の運用を通して有用性の検証を行う。また、コンパイルサーバのさらなる改良を行いコンパイル時間の改善を目指す。

謝辞 本研究は科研費基盤研究 B:16H03074 の支援を受けた。

参考文献

- (1) 情報処理学会, “カリキュラム標準 J17 カリキュラム標準ソフトウェアエンジニアリング領域(SE) J17-SE-報告書-20180319,” (2018).
- (2) Hiya, S., Hisazumi, K., Fukuda, A. et al., “clooca: Web based tool for Domain Specific Modeling,” Proceedings of 16th ACM/IEEE MoDELS Demonstrations, pp.31-35, (2013).
- (3) 香山 瑞恵, 小形 真平, 永井 孝: “モデル駆動開発方法論に基づく UML プログラミング教育環境,” 教育システム情報学会論文誌, 第 36 巻, 2 号, pp.118-130 (2019).
- (4) 増田壮志, “大学での利用を想定したモデル駆動開発用コンパイルサーバの構築,” 卒業研究発表資料, (2012).
- (5) 但馬将貴, “モデル評価機能を有した初学者向け概念モデリング学習環境の提案,” 修士学位論文, (2017).
- (6) 中野敬久, “コンテナ型仮想化によるモデリング教育用コンパイルサーバ利用の考察,” 修士学位論文 (2020).