1A-04

組込みソフトウェア向け実機レス開発環境の検討

加藤　寿和†　　山本　整†　　桐村　昌行†
三菱電機株式会社　情報技術総合研究所†

1 はじめに

製品開発では，競争力強化のために Time to market を意識した市場投入が重要となる．特に組込みシステムの開発では，開発期間を短縮させるため，ハードウェアとソフトウェアを並行して開発することが一般的に行われている．この場合，専用のハードウェア（以下，実機）が完成するまで，組込みソフトウェアの実行や試験を行うことができないという課題がある．

上記の課題に対して，パソコンやサーバなどで広く用いられている仮想化技術を活用し，実機を用いずに組込みソフトウェアの開発が可能な実機レス開発環境が検討されつつある[1]．

本稿では，仮想化技術を利用した実機レス開発環境として，仮想化するリソースレイヤに着目した 3 つの実現方式を検討し，評価を行った結果について報告する．なお，実機レス開発環境を動作させるパソコンの OS と実機に搭載する OS については，ともに Linux を想定する．

2 仮想化技術

仮想化技術は，コンピュータのリソースを抽象化することで，単一マシン上で複数の仮想マシンや仮想 OS を動作させる技術である．パソコンやサーバの分野では，コストやスペースの削減，耐障害性の向上などを目的として，従来から盛んに研究や実用化が行われている．

仮想化技術には，仮想化するリソースレイヤや実現方法の違いから複数の方式が存在する．主な技術分類として，下記の 3 つが挙げられる．

(1) コンテナ技術
(2) ハイパーバイザ技術
(3) エミュレータ技術

3 章において，技術分類毎に検討した実機レス開発環境の実現方式を説明する．

The study of device-less development environment for embedded software
Toshikazu Kato†, Hitoshi Yamamoto†, Masayuki Kirimura†
† Information Technology R&D Center, Mitsubishi Electric Corporation

3 実機レス開発環境の実現方式

3.1 コンテナ方式

本方式は，OS レベルでの仮想化を行うコンテナ技術を利用して，ホスト OS 上に仮想の OS 実行環境（以下，コンテナ）を構築し，そのコンテナ上に組込みソフトウェアを実行する実機レス開発環境を構築する方式である．本方式におけるソフトウェア構成例を図 1(a)に示す．コンテナを管理する主なソフトウェアとして，Docker[2]が挙げられる．

アプリケーションとライブラリをひとまとめにした実行環境を，ホスト OS の実行環境から独立して構築することが可能なため，実機環境と同一バージョンのライブラリを利用することが可能である．また，仮想マシンを構築する他の実現方式と比較して，パソコンのリソース消費を抑えることが可能である．一方，コンテナはパソコンのハードウェアとホスト OS のカーネルを直接使用するため，実機環境におけるカーネルや CPU アーキテクチャを仮想化することは不可能であり，ハードウェアに依存するソフトウェアの開発環境としては適さないと考えられる．

3.2 ハイパーバイザ方式

本方式は，ハードウェアレベルでの仮想化を行うハイパーバイザ技術を利用して，パソコン上に仮想マシンを構築し，その仮想マシン上に組込みソフトウェアを実行する実機レス開発環境を構築する方式である．本方式におけるソフトウェア構成例を図 1(b)に示す．主なハイパーバイザとして，VirtualBox[3]が挙げられる．

ホスト OS とは異なる OS を隔離した状態で動作させることができるため，実機環境と同一バージョンのライブラリやカーネルを利用することが可能である．一方，仮想マシンはパソコンの CPU アーキテクチャを使用するため，実機環境における CPU アーキテクチャを仮想化することは不可能であり，コンテナ方式と同様に，ハードウェアに依存するソフトウェアの開発環境としては適さないと考えられる．

Copyright ©2021 Information Processing Society of Japan.
All Rights Reserved.

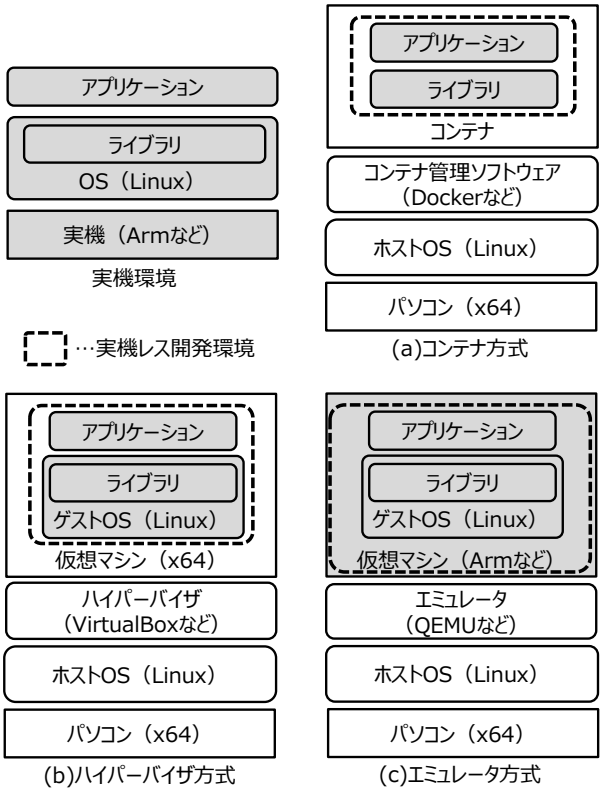

図 1 検討方式によるソフトウェア構成例

3.3 エミュレータ方式

本方式は，CPU アーキテクチャを含めたハードウェアレベルでの仮想化を行うエミュレータ（命令セットシミュレータ）技術を利用して，パソコン上に仮想マシンを構築し，その仮想マシン上に組込みソフトウェアを実行する実機レス開発環境を構築する方式である．本方式におけるソフトウェア構成例を図 1(c)に示す．エミュレータ方式を実現する主なソフトウェアとして，QEMU[4]が挙げられる．

前述したように，CPU アーキテクチャを含めたハードウェアレベルでの仮想化を行うため，実機環境と同一のライブラリ，カーネル，CPU アーキテクチャを利用することが可能である．一方，命令セットをシミュレーションする際のパソコンの処理負荷が大きく，メモリ使用量も比較的大きいことや，他の方式と比較して導入に関わる開発量が増えることが課題になると考えられる．

4 評価

3 章で述べた検討方式について，下記の項目に関する定性的かつ相対的な評価を実施した．

（ア） 試験可能なソフトウェアレイヤ
（イ） 実行速度

表 1 検討方式の評価結果

評価項目	コンテナ方式	ハイパーバイザ方式	エミュレータ方式
試験可能なソフトウェアレイヤ	アプリ ライブラリ	アプリ ライブラリ カーネル	アプリ ライブラリ カーネル ドライバ
実行速度	○	△	×
メモリ使用量	○	△	×
導入コスト	○	×	×
配布サイズ	○	×	×

※○△×：方式間での相対比較

（ウ） メモリ使用量
（エ） 導入コスト
（オ） 配布サイズ

項目(オ)については，組込みソフトウェアを複数人で試験することを想定し，実機レス開発環境を配布する際のイメージサイズについて評価する．

評価結果を表 1 に示す．表 1 の結果のとおり，実機環境の再現性が高いほど，実機レス開発環境の実行性能は低下する傾向にあると言える．そのため，組込みシステムの開発毎に，実機レス開発環境に要求される実行環境を考慮して，使用する仮想化技術を判断する必要があると考える．例えば，CPU アーキテクチャに依存するソフトウェアの試験では，エミュレータ方式を用いる必要がある．一方で，アプリケーションとライブラリのみが試験対象である場合には，コンテナ方式を用いることが最適となる可能性がある．

5 おわりに

本稿では，実機を用いずに組込みソフトウェアの開発が可能な実機レス開発環境について，複数の仮想化技術による実現方式を検討し，評価を実施した．結果として，組込みソフトウェアの試験対象に応じて適切な方式を選択していくことが重要であると考える．

今後は，検討方式に対する定量的な評価を実施していく予定である．

参考文献

[1] 原嶋 他,“仮想化技術による実機レステスト環境の構築”,東芝レビュー,67,8,p.31-34,2012
[2] Docker Inc., “Docker” https://www.docker.com/
[3] Oracle Corp.,”VirtualBox”, https://www.virtualbox.org/
[4] Fabrice Bellard,“QEMU”, https://www.qemu.org/

Copyright ©2021 Information Processing Society of Japan.
All Rights Reserved.