Vol.2014-SLDM-165 No.37
Vol.2014-EMB-32 No.37
2014/3/16

組込み向け仮想化技術の評価

東山知彦[†1]　落合真一[†1]

近年、機能統合によるコスト削減や、新規プラットフォーム上での既存 S/W 実行環境の構築等の目的から、従来汎用 PC にて実現されていた仮想化技術を組込み機器でも適用する動きが高まりつつある。仮想化環境では、仮想化のオーバーヘッドによる性能劣化が懸念される。特に組込み機器はこれらの性能に対する要求が厳しい場合が多い。この流れを受け、組込み向け CPU に H/W の仮想化支援機能が追加されてきている。S/W で実装していた処理の一部を H/W で行うことにより仮想化のオーバーヘッドの軽減が見込まれる。そこで今回、ARM の仮想化支援機能を使用している ARM 向け KVM を対象として、H/W の仮想化支援技術の評価を行った。

本評価では、H/W の仮想化支援機能による性能向上が見込まれるメモリアクセス性能とリアルタイム応答性について評価を行った。その結果、メモリアクセス性能については、定常状態では仮想化による性能劣化は許容できる範囲内であった。一方、リアルタイム応答性能は H/W の仮想化支援機能だけでは性能劣化が抑えきれず課題であることが分かった。

1. はじめに

近年、機能統合によるコスト削減や、新規プラットフォーム上での既存 S/W 実行環境の構築等の目的から、従来汎用 PC にて実現されていた仮想化技術を組込み機器でも適用する動きが高まりつつある。

仮想化環境では、エミュレーションやリソースの競合等によるオーバーヘッドが発生し、リアルタイム応答性やスループット性能の劣化が懸念される。特に組込み機器はこれらの性能に対する要求が厳しい場合が多く、S/W のみで仮想化を実現するのは現実的でない。この流れを受け、ARM や PowerPC 等の組込み向け CPU に H/W の仮想化支援機能が追加されてきている。H/W で仮想化機能の一部を実現することにより、仮想化のオーバーヘッドが軽減されることが期待される。そこで今回、ARM の仮想化支援機能を使用している Linux ベースのハイパーバイザである ARM 向け KVM を対象として、H/W の仮想化支援技術の評価を行った。

以降、2 章にて仮想化技術の評価に関する関連研究について述べる。3 章で H/W の仮想化支援技術について述べ、4 章で評価の観点を検討する。5 章で具体的な評価方法を検討し、6 章で評価環境を述べる。7 章で評価結果を述べ、8 章で評価結果に対す考察を行う。最後に 9 章で本稿の内容をまとめる。

2. 関連研究

仮想環境における性能評価として、いくつかの先行研究事例がある。例えば、Miguel G. Xavier らは汎用 PC 上で仮想環境の性能評価を行っている。この研究では、Intel x86 アーキテクチャの PC 上に Linux をインストールし、Xen ハイパーバイザ[2]による仮想環境の性能評価を行っている。本評価では組込み向け仮想化技術について評価する。

†1 三菱電機株式会社　情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

3. H/W 仮想化支援技術

本章では H/W の仮想化支援技術について述べる。H/W の仮想化支援技術として、特権レベルの追加、ゲスト OS 用多段階アドレス変換、割込みの仮想化支援がある。以降、各節にて詳細を述べる。

3.1 特権レベルの追加

従来の CPU ではユーザモード、スーパーバイザモードという 2 つの特権レベルが用意されている。OS とアプリケーションの特権レベルを分けることで、アプリケーションが実行可能な命令を制限し、OS がアプリケーションを管理している。仮想環境では、ゲスト OS 間やゲスト OS とホスト OS 間が干渉しないよう、ゲスト OS の命令を制限し、ゲスト OS を管理する必要がある。そのため、スーパーバイザモードよりもさらに特権レベルの高いハイパーバイザモードが必要である。図 1 にハイパーバイザモードを使用したシステムの構成例を示す。

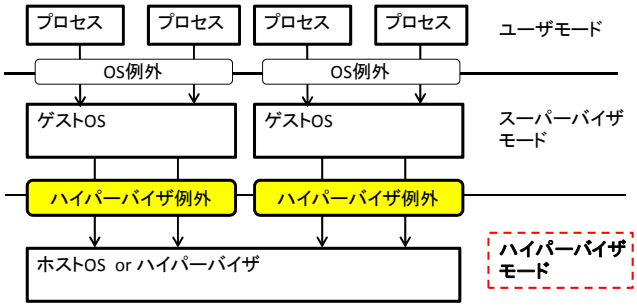

図 1：ハイパーバイザモードを使用したシステムの構成例

ハイパーバイザモードではホスト OS やハイパーバイザを動作させる。トラップするようにあらかじめ設定した命令をゲスト OS が実行した場合、ハイパーバイザ例外を発生させてより高い特権レベルであるハイパーバイザモードでトラップする。これにより、ゲスト OS 間やゲスト OS ーホスト OS 間が干渉しないよう、ホスト OS やハイパーバイザが調停することができる。

Vol.2014-SLDM-165 No.37
Vol.2014-EMB-32 No.37
2014/3/16

3.2 多段階アドレス変換

仮想環境では、ゲスト OS 同士、またはゲスト OS とホスト OS のアドレス空間を分離する必要がある。仮想化支援機能を備えた CPU は、ゲスト OS のアドレス変換を多段階で行うことで、OS ごとのアドレス空間を分離する。

図 2 にホスト OS とゲスト多段階アドレス変換の概略を示す。

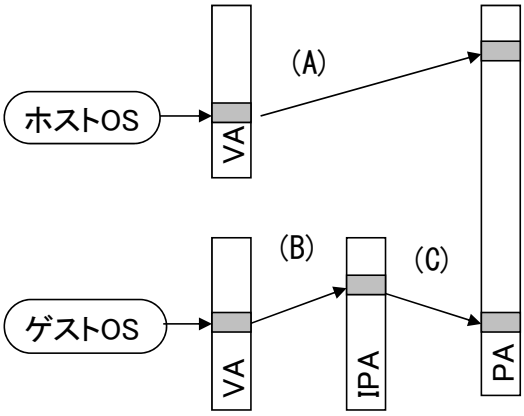

図 2：仮想環境でのアドレス変換

ホスト OS がメモリアクセスする場合は、仮想アドレス(VA)→物理アドレス(PA)というアドレス変換が行われる(図 2 中(A))。一方、ゲスト OS がメモリアクセスする場合、VA→中間アドレス(IPA)→物理アドレス(PA)という変換がおこなわれる(図 2 中(B)及び(C)) 。ゲスト OS は IPA を真の物理アドレスであると認識して動作する。

3.3 割込み仮想化支援

仮想化環境では、ゲスト OS に対する割込みをゲスト OS に振り分ける必要がある。発生した割込みが現在実行中のゲスト OS に対するものであるか否かで割込み仮想化支援機能が分けられる。

(I) 現在実行中のゲスト OS に対する割込みの場合

現在実行中の OS に対する割込みの場合、H/W の割込み仮想化支援機能を備えた割込みコントローラは直接割込みを通知する。

(II) 現在実行中でないゲスト OS に対する割込みの場合(図 3)

現在実行中でないゲスト OS に対する割込みの場合、ホスト OS やハイパーバイザが一旦割込みを受けつけ、割込みコントローラに割込みを登録する(図 3 中の割込みハンドリング機構)。そして、実行中のゲスト OS が遷移可能なタイミングで割込み通知先のゲスト OS に遷移する。仮想化支援機能を備えた割込みコントローラは、割込み通知先のゲスト OS に遷移したタイミングで自動的に仮想的な割込みを発生させる。複数の割込みをペンディングさせていた場合には、これらを管理し優先度に応じて割込みを発生させる。

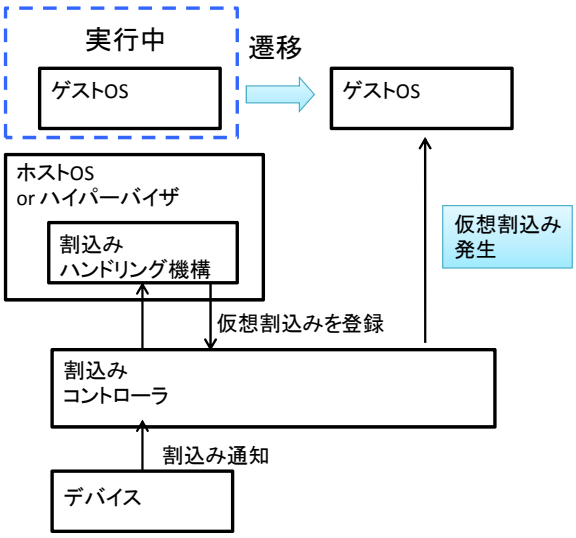

図 3：仮想割込み発生の流れ

4. 評価の観点

組込み機器は汎用 OS に比べて一般的に性能に対する要求が厳しい。そのため、仮想化のオーバーヘッドによる性能の劣化は重要な問題である。本章では H/W の仮想化支援技術を利用した場合に想定される性能への影響について述べる。本評価では、ここで述べた性能について評価を行った。

4.1 メモリアクセス性能

アドレス変換時、仮想アドレスに該当する物理アドレスの対応(ページテーブルエントリ)を探す。図 4 にページテーブルエントリ探索を表した模式図を示す。

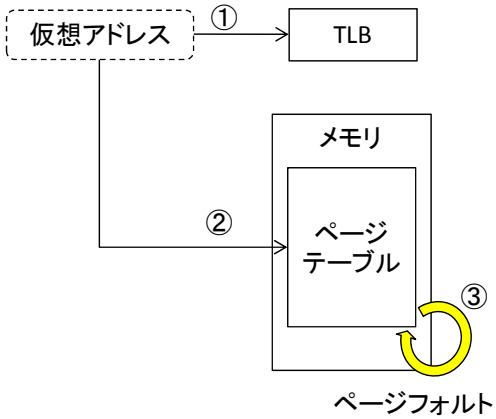

図 4：アドレス変換時のページテーブルエントリ探索

まず、CPU 内の TLB(Translation Lookaside Buffer)を参照する(図 4 中①)。TLB に該当するページテーブルエントリがない場合、通常メモリ上にあるページテーブルを参照する(図 4 中②)。ページテーブルにページテーブルエントリがない場合、ページフォルトを起こしてページテーブルにペ

ージテーブルエントリを追加する(図 4 中③)。①の場合ほとんどオーバーヘッドは生じないため、本評価では②と③の場合を評価の対象とする。本稿では、②を H/W のアドレス変換、③をページフォルトと定義する。ただし、ページフォルトは 2 次記憶にスワップされているアドレスに対するアクセス時の例外のことを指す場合もある。本稿では、メモリ上に存在するがページテーブルエントリが作成されていないアドレスに対するアクセス時の例外のことをページフォルトと定義する。

多段階アドレス変換を行った場合、図 2 に示したように、ホスト OS は 1 回であるアドレス変換がゲスト OS では 2 回行われる。これにより、メモリアクセス性能への影響が予想される。ページテーブルにヒットした場合、②の H/W のアドレス変換を 2 回行うことによる性能差がホスト OS とゲスト OS に生じる。ページテーブルにヒットしなかった場合、図 2 の(B)の変換で割り当てられた IPA が、PA とのページテーブルエントリが作成されていないものであった場合、(C)のページフォルトも同時に発生する可能性がある。よってページフォルトも 2 回発生する可能性があり、この性能差がホスト OS とゲスト OS の間に生じる。

以上から本評価では、ページフォルトと H/W によるアドレス変換について評価を行うこととした。

4.2 リアルタイム応答性

割込み仮想化支援を利用する場合、3.3 節の(II)の場合において、割込みを一度ハイパーバイザで受け、ハイパーバイザが割込みをハンドリングすることによるオーバーヘッドが見込まれる。これにより、ゲスト OS のリアルタイム応答性の劣化が予想される。

以上から本評価ではリアルタイム応答性について評価を行うこととした。

5. 評価方法

本評価では ARM 向け KVM を評価の対象とした。KVM は Linux をホスト OS とするハイパーバイザであり、ARM の仮想化支援機能を使用している。仮想化を行っていないネイティブな環境と、KVM による仮想環境でそれぞれ評価を行い、結果を比較した。

以下、各評価の詳細について説明する。

5.1 メモリアクセス性能

ホスト OS における図 2 の(A)の処理と、ゲスト OS における(B) →(C)の処理の差を比較する。4 章で述べたように、メモリアクセス性能はページフォルトと H/W のアドレス変換を評価する。以降、各項にてそれぞれの評価手法について説明する。

5.1.1 ページフォルト

ページフォルトは LMbench のページフォルトレイテンシテスト(ファイル名：lat_pagefault)にて測定した。このテストは試行ごとにテスト領域のアンマップ→マップを行い、当該領域を参照することによってページフォルトを発生させるテストである。このテストにて、図 2 の(B)のページフォルトを発生させ、4 章で述べた(C)のページフォルトの同時発生による性能差が確認できるかを評価した。

5.1.2 H/W のアドレス変換

H/W のアドレス変換については、LMbench のメモリアクセスレイテンシテスト(ファイル名：lat_mem_rd)にて測定した。このテストは、あらかじめ参照用データを書き込んでおき、テスト実行時に次々と読み出していくテストである。読み出しは参照用データ作成直後に実行される。参照用データ書き込み時間はテスト実行時間に含まれない。参照用データ書き込み時にページフォルトが発生するが、テスト実行時(読み出し時)には既に参照用データのページテーブルエントリが出来上がっているため、ページフォルトは発生しないと考えられる。純粋な H/W のアドレス変換のレイテンシが測定できると考え、本評価では、このメモリアクセスレイテンシテストにて H/W のアドレス変換のレイテンシを評価することとした。

測定するメモリ領域は H/W に搭載されているキャッシュ(後述)より十分大きな 256MB とし、キャッシュミスを発生させてメモリにアクセスするようにした。

5.2 リアルタイム応答性

リアルタイム応答性はタイマ割込みに対する応答性で評価した。3.3 節で述べたように、ゲスト OS の実行状態に応じて、(I)ゲスト OS に直接割込みを通知する場合と、(II)ハイパーバイザがハンドリングする場合がある。リアルタイム応答性への影響が大きいと考えられるのは(II)が発生した場合である。ただし、KVM はゲスト OS が実行中の場合も一旦 KVM が割込みを受け取る仕組みになっている。そのため、割込み発生時のゲスト OS の実行状態に関わらず(II)のオーバーヘッドの影響が測定可能である。

本評価では割込みが発生してから、アプリケーションが実行されるまでを測定対象とした。周期的に発生するタイマ割込みによってアプリケーションを起床させ、起床時間の遅延を測定した。測定にはリアルタイム応答性評価ベンチマークである cyclictest を用いた。cyclictest は割込みが発生してからアプリケーションが起床するまでの遅延時間を測定する。

表 1 に本評価での測定パラメータを示す。優先度及びスケジューリングポリシーは cyclictest の優先度、スケジューリングポリシーを表す。また、cyclictest で使用するメモリ領域は mlcokall 関数によりメモリ上にロックした。

Vol.2014-SLDM-165 No.37
Vol.2014-EMB-32 No.37
2014/3/16

表 1：測定パラメータ

優先度	リアルタイム優先度(99)
スケジューリングポリシー	FIFO
ループ回数	12 万回
起床周期	30ms

6. 評価環境

評価に用いた H/W 環境、S/W 環境を表 2、表 3 にそれぞれ示す。ホスト OS、ゲスト OS のカーネル、ユーザランドは全て参考文献[4]を使用し、環境構築は全て参考文献[5]に記載の手順で行っている。

表 2：H/W 環境

評価ボード	Arndale Board (Sumsung) [1]
CPU	・ Cortex-A15(1.7GHz)×2 コア ・ 32KB(instruction)/32KB(DATA)L1 Cache and 1MB l2 Cache
主記憶	32-bit 800Mhz DDR3(L)/DDR3 1Gbytes × 2
2 次記憶 (micro SDHC)	TS8GUSDU1 (Transcend) ・Class 10 ・8G

表 3：S/W 環境

ホスト OS	Virtual Open Systems 社提供[4] ・カーネル Linux-3.8.0-rc4. ・ユーザランド Ubuntu12.04LTS ベース
ゲスト OS	Virtual Open Systems 社提供[4] ・カーネル Linux-3.8.0-rc4. ・ユーザランド Ubuntu12.04LTS ベース
eglibc	eglibc-2.15
cyclictest	cyclictest 0.87 git://git.kernel.org/pub/scm/linux/kernel/git/clrkwllms/rt-tests.git/
LMbench	LMbench-3.0-a9 http://sourceforge.net/projects/lmbench/

7. 結果

本章は、本評価の結果について述べる。0 節にて CPU 演算性能、7.2 節にてリアルタイム応答性、7.1.1 節にてメモリアクセス性能の評価結果をそれぞれ示す。

なお、ホスト OS とゲスト OS で CPU の性能がないことを確認するため、別途 dhrystone と whetstone による評価を行いスコアがほぼ同一であることを確認した。

7.1 メモリアクセス性能評価

7.1.1 ページフォルト

表 4 に各 OS における LMbench のページフォルトレイテンシテストの結果を示す。

表 4：ページフォルトレイテンシ評価結果(µs)

対象OS	スコア
ホストOS	2.40
ゲストOS	2.38

ページフォルトレイテンシは、ホスト OS とゲスト OS でほとんど差が見られなかった。これについては考察で論じる。

7.1.2 アドレス変換

表 5 に各 OS における LMbench のメモリアクセスレイテンシテストの結果を示す。

表 5：メモリアクセスレイテンシ測定結果(ns)

対象OS	シーケンシャル	ランダム
ホストOS	95.60	122.30
ゲストOS	96.20	273.75

シーケンシャルアクセスでは差が出なかったが、ランダムアクセスで 2.24 倍の性能差がでた。これについては考察で論じる。

7.2 リアルタイム応答性評価

表 6 にリアルタイム応答性評価における応答遅延の最大値、最小値、平均値を示す。また、このときの遅延時間の分布を図 5、図 6 に示す。

表 6：リアルタイム応答性評価結果(µs)

対象OS	最大遅延時間	95%信頼時間
ホストOS	185	108
ゲストOS	8939	251

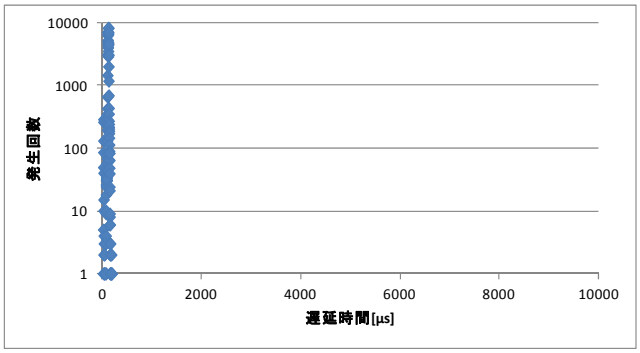

図 5：ホスト OS の応答遅延と発生回数

Vol.2014-SLDM-165 No.37
Vol.2014-EMB-32 No.37
2014/3/16

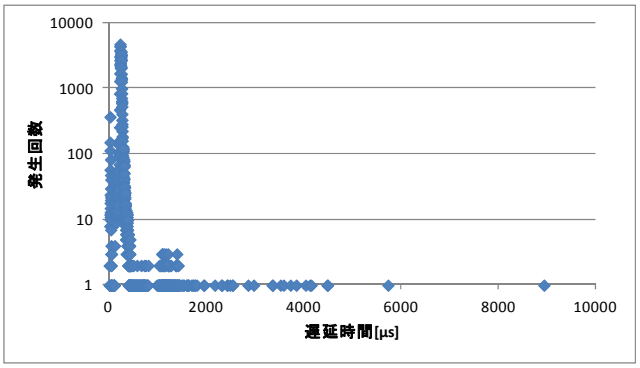

図 6：ゲスト OS の応答遅延と発生回数

最大応答遅延について、ホスト OS が 185μsであるのに対し、ゲスト OS は 8939μsの遅延が発生した。ゲスト OS はホスト OS の 48 倍もの最大応答遅延が発生することが分かった。

最大応答遅延はハードリアルタイムシステムにとって重要な指標である。ソフトリアルタイムシステムでは、デットラインを超過する場合があってもシステムの価値が 0 になるわけではない。そこで、5%の超過を許容し、95%が応答できる時間を 95%信頼時間と定義し、この時間を計算した。すると、ホスト OS が 119μs、ゲスト OS が 276μsであった。95%信頼時間で比較すると、ゲスト OS はホスト OS に比べ 2.3 倍の値であることが分かった。

以上のように、特に最大応答遅延について、ゲスト OS のリアルタイム応答性能はホスト OS に比べて大きく劣化することが分かった。これについては考察で論じる。

8. 考察

8.1 メモリアクセス性能評価

8.1.1 ページフォルト

ホスト OS とゲスト OS でページフォルトレイテンシにほとんど差が見られなかった原因について考察する。

ゲスト OS でのアドレス変換について、ARM 向け KVM では図 2 の(B)の変換時に発生するページフォルトはゲスト OS が処理し、(C)の変換時に発生するページフォルトは KVM が処理する。(C)のページフォルトが発生するタイミングを確認するため、KVM の該当する処理内にログ出力処理を追加し、ゲスト OS を起動しページフォルトテストを実行した。その結果、ゲスト OS 起動時にログが多数出力され、ページフォルトテスト実行時はログが出力されなかった。このことから、(C)の変換時のページテーブルエントリは、その大部分がゲスト OS 起動時に作成され、LMbench 実行時には(B)のページフォルトしか発生していないことが分かった。ゲスト OS とホスト OS のページフォルトレイテンシは、互いに 1 段階のページフォルトしか発生しないため差がでなかったといえる。

ただし、ゲスト OS 起動時にページフォルトが大量に発生していることから、起動時間等の起動時の性能に影響があることが予想される。この影響の評価については、今後の課題とする。

8.1.2 H/W のアドレス変換

8.1.2.1. 原因分析

ランダムアクセスにおいてゲスト OS のレイテンシが増加した原因について検討する。ページフォルトの発生の有無について、Linux ではルートファイルファイルシステム中の/proc/{PID}/stat 内に、実行中のプロセスのページフォルト発生回数が記録されている({PID}は対象プロセスのプロセス ID を表す)。5.1.2 で述べたように、LMbench のメモリアクセスレイテンシテストは、あらかじめ参照用データを書き込んでおき、テスト実行時に次々と読み出していくテストである。このテストを参照用データ作成時とテスト実行時を区別できるように変更し、/proc/{PID}/stat からページフォルト発生回数を確認した。すると、参照用データ作成時にページフォルトが頻発し、テスト実行時はページフォルトが発生していないことが分かった。よって、本テストにて確認されたランダムアクセスに関するゲスト OS とホスト OS の性能差は、H./W の多段階アドレス変換によるものであると考えられる。つまり、ホスト OS は 1 回の変換であるのに対し、ゲスト OS は 2 回の変換を行うため、レイテンシが増加したと考えられる。

ただし、シーケンシャルアクセスについては、ゲスト OS とホスト OS で差が見られなかった。シーケンシャルアクセスでは同一ページ内のアドレスを連続して参照するため、ランダムアクセスに比べて TLB にヒットする回数が多くなる。TLB にヒットすると図 2 のアドレス変換が行われないため、両 PF で差が出なかったと考えられる。一方、ランダムアクセスでは都度異なるページへアクセスするため、TLB ミスが発生し、図 2 のアドレス変換が行われていると考えられる。

8.1.2.2. 影響分析

ゲスト OS のレイテンシ増加が及ぼす影響度について述べる。今回の測定結果では、多段階アドレス変換時のメモリアクセスレイテンシは、1 段階の時に比べ、TLB ミスが発生した場合 2.24 倍になり、TLB にヒットした場合変わらないという結果が出た。ホスト OS と TLB ミスが発生する頻度は一般的に 0.01~1%程度である。仮に 1%のミス率だった場合、多段階変換を行った際のメモリアクセス性能は、1 段階のアドレス変換に比べて以下のようになる。

$$1 \times 0.99 + 2.24 \times 0.01 = 1.014(\text{倍})$$

1.4%程度の性能劣化であり、多段階アドレス変換の性能への影響は小さいと判断する。

8.2 リアルタイム応答性評価

8.2.1 原因分析

ゲスト OS の応答性能が劣化した原因について考察する。ARM 向け KVM では、GIC(Generic Interrupt Controller)という割込みコントローラが全ての IRQ(Interrupt Request)を KVM に通知し、KVM が割込みをハンドリングしている。

各 OS に対する割込み処理の流れを図 7 に示す。ホスト OS に対する割込みは以下のような流れで行う。

(1) タイマ割込みを GIC が受信
(2) GIC がホスト OS に割込みを通知
(3) ホスト OS が通常の割込み処理を実行

ゲスト OS に対する割込みは以下のような流れで行う。

(4) タイマ割込みを GIC が受信
(5) GIC が KVM に割込みを通知
(6) KVM がゲスト OS への仮想割込みを GIC に登録
(7) ゲスト OS の実行を再開
(8) (7)ゲスト OS の実行再開と同時に GIC がゲスト OS に対して仮想割込みを発生
(9) ゲスト OS が通常の割込み処理を実行

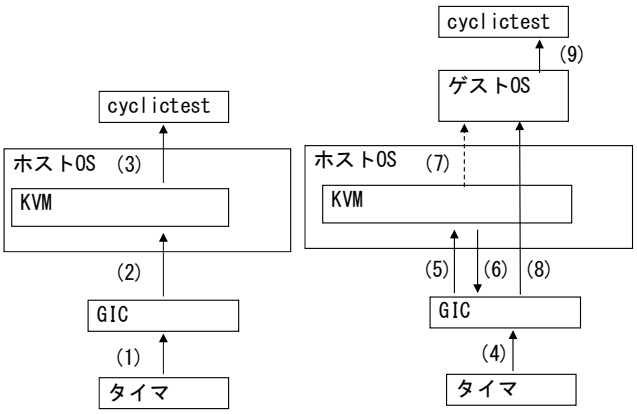

図 7：各 OS に対する割込みの流れ

ゲスト OS に対する割込み処理のうち、(4)、(5)、(9)はホスト OS に対する割込み処理と同等である。(6)、(7)、(8)がホスト OS に対する割込み処理との差分である。このうち、特にオーバーヘッドが大きいのは(7)である。ゲスト OS への割込みを KVM が受けた後、実際にゲスト OS に仮想割込みが発生するのは、ゲスト OS がスケジューリングされる時である。ゲスト OS の応答遅延には、このゲスト OS にスケジューリングされるまでの時間が反映される。

以上で述べたオーバーヘッドにより、ゲスト OS のリアルタイム応答性が劣化していると考えられる。

8.2.2 改善方法検討

改善方法について、検討する。8.2.1 の考察から、ゲスト OS 実行中にゲスト OS への割込みが発生した場合、ゲスト OS に直接通知するようにするようにすることで性能改善が見込まれる。ARM では HVC(Hypervisor Configuration Register)というレジスタの設定により、ゲスト OS への割込み通知を直接ゲスト OS に通知することが可能である。

ホスト OS 実行中にゲスト OS への割込みが発生した場合、割込みは一旦ホスト OS に通知される。ゲスト OS にリアルタイム応答性が必要な処理がある場合、その処理に関連するデバイスからの割込みを優先的にゲスト OS に通知する仕組みを組み込むことで性能改善が見込まれる。ただしこの仕組みはホスト OS の性能とのトレードオフになるため、システムの要件に応じて設計を行う必要がある。

9. おわりに

今回、ARM 向け KVM を対象に H/W の仮想化支援機能のオーバーヘッドを評価した。評価は、メモリアクセス性能及びリアルタイム応答性について行った。メモリアクセス性能は、ページフォルトと H/W のアドレス変換のレイテンシを測定した。ページフォルトレイテンシはホスト OS とゲスト OS で差が見られなかった。H/W のアドレス変換レイテンシは、ゲスト OS はホスト OS の 2.24 倍のレイテンシが発生したが、TLB ミスの発生頻度を考慮すると影響は小さいと判断した。ただし、起動時にページフォルトが頻発していることから、起動時の性能に影響があると考えられる。この評価については今後の課題とする。リアルタイム応答性について、ゲスト OS ではホスト OS の 48 倍もの最大応答遅延が発生した。これはゲスト OS に対する割込みを一旦 KVM で受け、ゲスト OS にスケジューリングする際に仮想割込みを発生させていることが原因である。

本結果から、ゲスト OS のリアルタイム応答性が H/W の仮想化支援機能の課題であるといえる。本評価では、測定時に負荷をかけていない。ホスト OS に負荷をかけた場合、さらなる性能劣化が予測される。今後はホスト OS に様々な負荷をかけた環境でのゲスト OS のリアルタイム応答性について評価していく予定である。また、考察にて述べた性能改善方法の実装についても検討していく予定である。

参考文献

[1] Miguel G. Xavier et al. “Performance Evaluation of Container-based Virtualization for High Performance Computing Environments”, 21st Euromicro International Conference on Parallel Distributed, and Network-Based Processing (2013)
[2] Xen: http://www.xen.org
[3] Arndale Board : http://www.arndaleboard.org/wiki/index.php/Main_Page
[4] Virtual Open Systems: https://github.com/virtualopensystems
[5] A step by step guide for linux kvm virtualization on embedded systems: http://www.virtualopensystems.com/en/solutions/guides/kvm-on-arm/