6G-2

マルチコアシステムにおけるLinuxカーネル2.6のリアルタイム性評価

徳丸　潤一†　攝津　敦†　落合　真一†　松本　利夫†
三菱電機（株）　情報技術総合研究所†

1. はじめに

デジタル家電や携帯電話などの組込み機器では、高機能化に伴いハードウェアのマルチコア化が進んでいる。その為の制御用OSとしてLinuxの採用が増加しており、近年では1ミリ秒以下のオーダの応答性を必要とする領域に対しても適用が広がってきている。Linuxカーネル2.6では、リアルタイム優先度の導入やプリエンプション機能など、Linuxカーネル単体でのリアルタイム性の向上が図られてきており、また従来からマルチプロセッサのサポートを考慮したSymmetric Multiprocessing（以下SMP）機能を実装しマルチコアとの親和も高い。しかしながらマルチコアとリアルタイム性との整合性については評価できていない。そこで今回、マルチコアシステムにおけるLinuxカーネル2.6のリアルタイム性について評価を実施した。本稿ではその結果について述べる。

2. Linuxのマルチコア/リアルタイム機能

2.1. Linuxのマルチコア機能

Linuxは複数のコアが対等な立場で処理分担するSMPの構成で動作することが可能である。SMPでは、複数のコアで1つのメモリ空間を使用し情報を共有している。一方SMPに対して、コアごとに役割を固定化し、複数のコアが独立したメモリ空間を確保し動作するAsymmetric Multiprocessing(以下AMP)という構成があるが、Linuxではこの構成をとることは出来ない。しかしLinuxでは、プロセスを特定のコアに割付けるCPUアフィニティ機能を有しており、これを用いることで擬似的なAMPを構成する事が可能である。このCPUアフィニティを設定するのは、Control Group(以下cgroup)機能を用いる。cgroupは、複数のプロセスが所属するグループ単位で、管理するCPUコアを指定することができる機能である。

2.2. Linuxのリアルタイム機能

Linuxカーネル2.6のリアルタイム機能として、O(1)スケジューラを用いた固定優先度のサポート、システムコールの途中であっても実行中のプロセスをプリエンプトし、優先度の高いプロセスを実行できるプリエンプション機能などを備えている。

3. 評価方法

3.1. 測定環境

評価対象の構成を表1に示す。

表1　評価対象の構成

ハードウェア	
CPU	Xeon(R) CPU X5355 2.66GHz(8コア)
メモリ	4GB
ファイルシステム	ハードディスク　ext3:40GB
ソフトウェア	
カーネル	Linux Kernel 2.6.34 CONFIG_SMP=y CONFIG_HZ=1000 CONFIG_PREEMPT=y　など
ユーザランド	CentOS 5

3.2. 測定方法

リアルタイム性評価として、低優先度プロセスで負荷をかけた状態における高優先度プロセスの周期起動遅延時間を、プロセス応答遅延時間として測定した。

負荷プログラムはカーネル負荷(共有メモリ、スケジューラ、タイマ、メモリ)とI/O負荷(ハードディスクに対する読出し書込み)で構成している。

3.3. 評価の観点

以下の観点で評価を行った

(1) 動作コア数とリアルタイム性の関係
　動作コア数の増減が、リアルタイム性に与える影響を評価するために、Linuxカーネルの動作CPUコア数(1，4，8)を変化させ測定を行った。

(2) アフィニティとリアルタイム性の関係
　CPUアフィニティの設定が、リアルタイム性に与える影響を評価するため、cgroup機能を利用し高優先度のプロセスと低優先度のプロセス(及びデーモン)を別々のコアに割付け、測定を行った。

(3) I/O系の負荷とリアルタイム性の関係
　I/O系の負荷が、リアルタイム性に与える影響を評価するため、ハードディスクに対する読出し書込みの負荷を追加し、測定を行った。

4. 測定結果

3.3(1)～(3)の測定結果をそれぞれ次ページの図1～図3に示す。これらの結果から以下のことが判明した。

4.1. 動作コア数とリアルタイム性の関係(図1)

プロセス応答時間の最大値について、1コアの場合では90マイクロ秒、4コア540マイクロ秒、8コアで600マイクロ秒の遅延が発生し、シングルコアかマルチコアかで顕著な差が見られた。プロセス応答時間の分布については、1コアの場合は10マイクロ秒付近にサンプル数の98%が集中している。同様に4コアでは0マイクロ秒～60マイクロ秒の間、8コアの場合は0マイクロ秒～110マイクロ秒の間となっており、コア数に応じて分布が広がっていることが判明した。

4.2. アフィニティとリアルタイム性の関係(図2)

CPUアフィニティを設定することによって最大遅延時間は4コアの場合では90マイクロ秒、8コアの場合で140マイクロ秒と改善が見られた。プロセス応答時間のばらつきについても小さくなり4コアの場合0～40マイクロ秒の間に98%以上のサンプルが集中した。また、8コアの場合も同様に0～90マイクロ秒の間に集中するようになった。

4.3. I/O系の負荷とリアルタイム性の関係(図3)

ハードディスクに対して負荷をかけた場合、非常に大きなプロセス応答時間の遅延が発生し、1コアで720マイクロ秒、4コアで1,350マイクロ秒、8コアで3,440マイクロ秒となった。また、遅延時間のばらつきも大きくなった。

4.4. 測定結果まとめ

以上の結果より次の3点が判明した。

- コア数が少ない方が、プロセス応答性能の最大遅延が小さく、遅延のばらつきも少ない。
- 周期起動プログラムにCPUアフィニティを設定す

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.

ることでプロセス応答性のばらつきが少なくなり、プロセス応答性が改善される。

・ I/O に対して負荷をかけることでプロセス応答性の最大遅延時間が増大する。

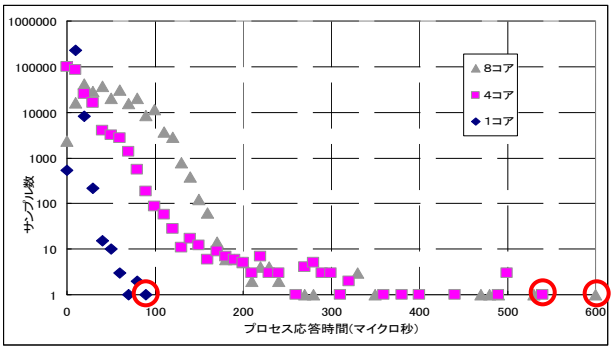

図 1　測定結果(CPU アフィニティ設定：無、ハードディスク負荷：無)

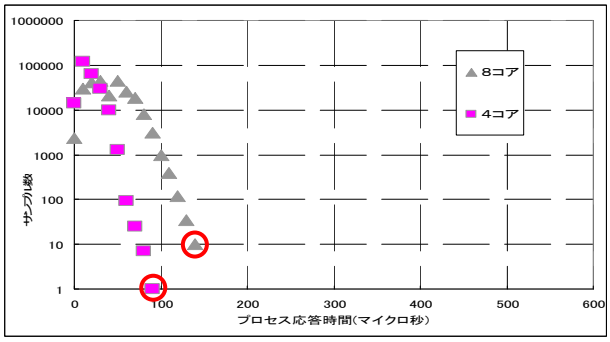

図 2　測定結果(CPU アフィニティ設定：有、ハードディスク負荷：無)

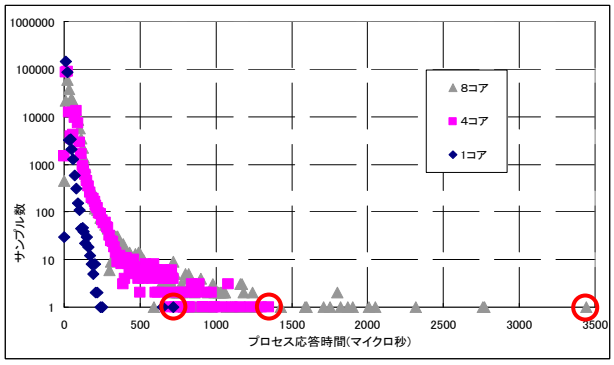

図 3　測定結果(CPU アフィニティ設定：無、ハードディスク負荷：有)

5.　考察

5.1. コア数増加による応答性劣化の要因

今回の測定結果から判明したコア数増加による応答性劣化の要因として、以下のような事が考えられる。

・資源の獲得解放時のロックによる遅延

カーネル負荷によって各コアがカーネル資源を頻繁に獲得解放している。周期起動プロセスが動作するときに、資源獲得待ちが発生することで、割込み処理やプロセス処理が遅延すると考えられる。

・割込み/プロセススイッチのロックによる遅延

周期起動プログラムが動作するコアとは別のコアにタイマ割込みが上げられるような場合、割込み処理実行側のコアでクリティカルな処理が実行されていると、割込み処理に切換わらず、割り込み処理が遅延すると考えられる。

・キャッシュフラッシュによる処理の遅延

メモリ負荷の影響によりキャッシュ上のデータが頻繁に排出される。周期起動プロセスが動作するときにキャッシュのデータ取込み、キャッシュの同期などが発生しプロセス動作が遅延すると考えられる。

これらの要因がコア数の増加により大きくなりプロセス応答時間の劣化に影響したと思われる。

5.2. アフィニティ効果による応答性向上の要因

CPU アフィニティを設定することで以下の効果が得られたと考えられる。

・「資源の獲得解放時のロックによる遅延」の減少

利用できるコアを制限することで、スケジューラなどのカーネル資源競合が減っているため、割り込み処理やプロセス処理が短くなったと考えられる。

・「キャッシュフラッシュによる処理遅延」の減少

特定コアに周期起動プロセスのみを割付けていることにより、そのコアが管理しているキャッシュに周期起動プロセスのデータが常に取り込まれている状態で動作できるため、プロセス動作が速くなったと考えられる。

これらの効果によりプロセス応答性が向上したと考えられる。

5.3. I/O 系の負荷による応答性劣化の要因

I/O 系負荷を設定したことで以下のような要因が発生し、プロセス応答性が劣化したと考えられる。

・資源の獲得解放時のロックによる遅延

カーネル資源の獲得解放に加えて I/O 資源の獲得解放が加わることで、割り込み処理やプロセス処理が大幅に遅延していると考えられる。

・割込み/プロセススイッチのロックによる遅延

ディスク I/O の完了を受取るコアと割込み処理を実行するコアが異なる場合、割込み処理実行側のコアで I/O 獲得解放待ちが発生していると、割込み処理に切替わらず、割込み処理が遅延すると考えられる。

要因としては 5.1. で述べた要因に加えて上記のようなディスク I/O 待ちが発生しているためプロセス応答時間が劣化していると考えられる。

6.　おわりに

本稿では、マルチコアシステムにおける Linux カーネル 2.6 のリアルタイム性評価結果について述べた。評価の結果、マルチコアシステムではコア数が増えるにしたがってリアルタイム性が低下していることが判明した。マルチコアシステムにおいてリアルタイム性を向上させるためには、動作プログラムに CPU に対するアフィニティを設定する、I/O 負荷を取り除く、コア数を少なくするなどの設定を行う必要があると考える。

今後は更なるリアルタイム性の向上を目指し、プロセス応答時間が遅延する要因の調査を続けていく予定である。

参考文献

[1] 出原章雄, 攝津　敦, 伊藤孝之, 落合真一：組込み機器における Linux カーネル 2.6 のリアルタイム性評価(情報処理学会第 68 回全国大会)

Evaluation of Linux Kernel 2.6 as a “realtime” Operating System for Multi-core System

† Junichi Tokumaru , Atsushi Settsu , Shinichi Ochiai , Toshio Matsumoto, Information Technology R&D Center , Mitsubishi Electric Corporation

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.