

# 組み込み向けマルチコアプロセッサ MPCore を用いた 応答性/機能性両立環境評価 —制御処理と情報処理の融合にむけて—

阿部 剛 酒井 淳嗣

日本電気株式会社 〒229-1198 神奈川県相模原市下九沢 1120

E-mail: t-abe@dr.jp.nec.com, jsakai@bc.jp.nec.com

あらまし 次世代組み込みシステムにおいては、従来求められてきた機器制御処理に加え、その機器から得られた情報に RMS(認識/抽出/合成)といった知的処理を施し人もしくは機器にフィードバックする情報処理能力が求められる。一方、それら増加する処理を支えるべきプロセッサに於いては、微細化に伴う消費電力問題から周波数向上が止まりマルチコアプロセッサ環境へ移行しつつある。この背景を鑑み、本論文では、マルチコアプロセッサ環境上に機器制御処理と情報処理を両立させる環境を複数提案し、その組み込み応答性能を評価した。また、評価によりマルチコア環境では、応答性能安定化には従来以上に割り込み処理に関わるメモリアクセス量が応答性能ジッタに影響を与えるため、機器制御処理 OS と情報処理 OS を両立させた HeteroOS 環境を構築する事が適切であるということが分った。

キーワード マルチコア 組み込みシステム 応答性能

## Coexistent environment of Responsiveness and Functionality on multi-core processor, MPCore.

— Toward harmonizing Control with Information processing —

Tsuyoshi Abe, Junji Sakai

NEC System IP Core Research Laboratories 1120 Shimokuzawa, Sagami-hara, Kanagawa, 229-1198 Japan

E-mail: t-abe@dr.jp.nec.com, jsakai@bc.jp.nec.com

**Abstract** The next generation embedded system will require not only the control processing but also the information processing ability like RMS(recognition/mining/synthesis). On the other hand, the frequency improvement is braked by the power consumption growth in spite of those increasing processing, and so it is shifting to the multi core processor. In this background, this paper proposed environments to coexist the control processing with information processing on the multi core processor environment, and evaluated the interruption performance. Moreover, this evaluation demonstrated HeteroOS environment which unites real-time OS with information processing OS is the best one, because the amount of the memory access to process interrupts influences the response performance and stabilization in the multi core environment.

**Keyword** MuliCore, Embedded System, Responsibility

### 1. はじめに

組み込みシステムには機器を安全確実に制御するという目的のために高い割り込み応答性能/性能予測性/信頼性が求められてきた。しかし、現在及び将来の組み込みシステムには、この機器が収集してきた情報を必要に応じて加工し、その情報を必要とする対象に対して適切な形で提供する情報処理能力も求められつつある。特に、車載システムではこの方向性が顕著であり、自動駐車などの自動運転、前方/後方から来る人や車の認識およびドライバへの警告など、外部の情報から必要

な情報を選び、加工し、人もしくは機器に提供するサービスが提案/実現されている。このような情報の選別や加工といった情報処理ソフトウェアは、非常に複雑になる。したがって、情報処理を実現するシステムには、ソフトウェア生産性を重視した環境が求められてきた。組み込みシステムでは、この機器制御処理と情報処理の両環境を一つのシステムとして動作させる必要がある。

この組み込みシステムは、機器制御処理と情報処理がそれぞれ要求する信頼性、応答性能、生産性を同時満

たす必要があるが、これらの関係にはトレードオフ関係がある。例えば、生産性の向上は、レイヤ化などの抽象化を用い実装を隠蔽することで達成される。一方、この抽象化は処理時間と言う観点では性能を落とす方向に向かう。また、実装を隠蔽するため、性能予測性も落ちる。つまり、性能と生産性はトレードオフ関係にあると言える。さらに、このような抽象化を重ねることによりブラックボックス化が進むと信頼性を落とすと言う見方も出来る。したがって、一つの OS でこれら要求全てを満たすのは難しい。

そこで我々はこれら要求を一つのシステムで実現できる方法として、マルチコアプロセッサの各コアを機器制御処理ドメイン用と情報処理ドメイン用にそれぞれ配分することにより、全ての要求を同時に満たすことが可能なシステムを検討した。本論文でのドメインとは、関連処理をまとめて特定のコア上に隔離して動作させる単位のことを指す。このドメインをマルチコアプロセッサ上に実現する方式として、分離手段と使用する OS が異なる3つの案を提示し、それらの応答性能の比較評価を行った。

機器制御処理と情報処理が要求する応答性能はシステムによって異なるが、本稿では、達成すべき性能として以下のような目標を定めた。

● 機器制御処理

機器のデバイスを制御するための処理であることから、非常に高い応答性能とその安定性が求められる。割り込み応答時間の目安は用途に応じて様々だが、概ね数 usec~数十 usec に収まることが望まれるため、ここでは 50usec を目標とした。

● 情報処理

主要な対象が人であることから、割り込み応答時間としては数 msec~数十 msec オーダ程度が求められる。情報処理では複雑かつ多量な処理への対応が求められているため、短期的な応答性能を犠牲にする遅延処理などを用い、処理全体としてより高いスループット性能を実現することを重視している。

我々が目指すシステムは上記2つの処理の応答性能を満たした上で、両立させなければならないが、とりわけ、機器制御処理側の応答性能を達成することが重要である。この機器制御処理応答性能は、情報処理ドメイン側の動作状況に関わらず達成する必要がある。機器制御ドメインに影響を与える要因としては、CPU 的負荷に加え、マルチコアプロセッサ環境にて CPU 間で共有されるバスやメモリリソースによる性能干渉が考えられる。

以上をまとめると、本論文は以下について述べる。

- マルチコア環境上に機器制御処理と情報処理を

両立させるドメイン分離環境案をいくつか提案する

- 提案した各ドメイン環境の応答性能を各種負荷環境下で測定し、その最悪値が機器制御処理ドメインの目標性能である 50usec に収まるかどうかを評価する。
- 結果を解析することで、各ドメイン環境が提供する効果を明らかにし、さらなる性能向上案を提案する

2. ドメイン分離環境

我々が目指すシステムにおいて必要とされるドメインは、機器の開発生産性を重視する情報処理ドメインと、応答性能を重視する機器制御ドメインである。昨今、開発生産性の高い OS として、組み込み機器向けにも UNIX 系 OS が注目されており、その一つである Linux は携帯電話はじめ多様な情報系組み込み機器に利用されている。一方、機器制御ドメインには RTOS を用いることが通例であったが、ここでも開発生産性改善を求めて UNIX 系 OS が用いられることがある。そこで我々は、情報処理ドメイン OS として UNIX 系 OS、機器制御ドメイン OS として RTOS 又は UNIX 系 OS を想定することにした。

ドメインの分離方法には、スケジューラレベルでプロセスを特定の CPU にバインドさせ分離させる Affinity 手法と完全に別 OS を起動させる OS 分離が考えられる。本論文ではマルチコア環境を採用しており、これドメインは別々のプロセッサ上で動作する。

これら OS 及び分離方法を組み合わせて今回評価検討したドメイン分離環境を図 1 に示した。図中の APP とはユーザ空間で動作するアプリケーションを示している。ここで標準 SMP 環境はドメイン分離環境ではないが、他の3つのドメイン分離環境との比較評価のために用いる。

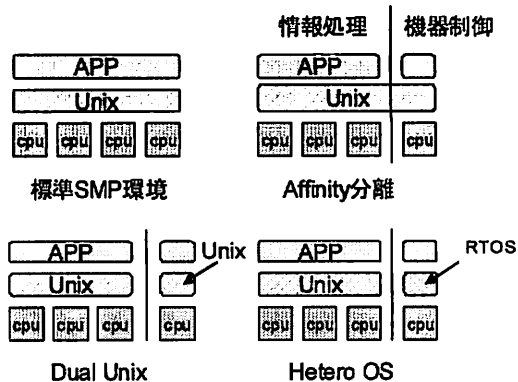


図 1 ドメイン分離環境

Affinity 環境とは、基盤 OS に SMP に対応した Unix 系 OS を用い、プロセスが実行の際に使用するプロセ

ッサを固定させることができる Affinity 機能を用いることにより機器制御処理ドメインと情報処理ドメインを分割する環境である。ドメイン分離が OS のスケジューリングレベルで行われているため、OS が使用する共有データによるドメイン間干渉が生じ、結果として応答性や信頼性を損ねることが懸念される。他方、もともと開発生産性が高いとされる Unix 環境をそのまま使え、またドメイン間で同じ環境を用いるため、最も実現が容易で生産性が高いドメイン分離環境である。

DualUnix 環境は、同じ Unix 系 OS を両ドメインで用いるが、異なる CPU 上に別々の OS インスタンスを起動させるものである。これにより、ソフトウェア的に完全に独立した OS 環境が作られるため、生産性や信頼性に優れた形態である。このような OS 分離による信頼性確保の有効性に関しては、論文[1]が詳しい。

最後に HeteroOS 環境は、機器制御処理ドメインの OS に RTOS を適用した環境である。ドメイン間の独立性の高さから信頼性が高く、また導入した RTOS よりもっとも高いレベルの応答性能を実現できる。他方、異なる OS に対するアプリケーション開発が必要となり、開発生産性面では若干劣る。この環境に関しては、文献[2],[3]に DualOS として詳しく書かれている。

以上3つの環境の定性比較を表 1 にまとめた。

表 1 定性比較

	信頼性	応答性	生産性
Affinity	×	△	○
DualUnix	○	△	○
HeteroOS	○	○	△

### 3. 応答性能評価方法

ここでは、応答性能として値を測定したポイントと応答測定時に用いた負荷に関して詳細に説明する。

#### 3.1. 測定ポイント

各環境の応答性能比較のためには、割り込みに対する割り込みハンドラ応答時間を用いた。割り込みに対する処理シーケンスは、用いる CPU 及び OS によって少々変化するが、概ね図 2 のような処理の流れで表現することが出来る。

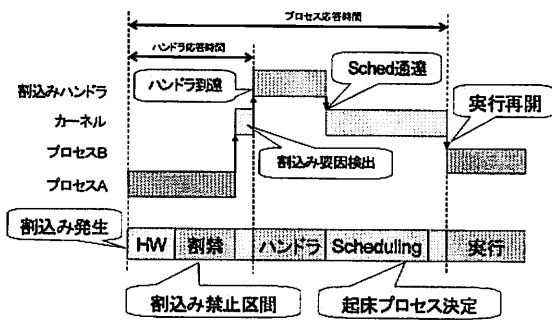


図 2 測定ポイント

割り込み応答時間についてはいくつか異なる定義があるが、この論文ではハードウェアによる割り込み発生から割り込みハンドラが呼びだされ、そのハンドラの最初に位置する時間測定コードが実行されるまでの時間を割り込み応答時間とした。

#### 3.2. 負荷

2章で提示した各ドメイン分離環境の応答性能の安定性は、情報処理ドメイン側に負荷アプリを走らせた場合の機器制御ドメインの割り込み応答時間を測定することで評価することにした。ここで情報処理ドメイン側負荷アプリとして次の2種類のものが考えられる。

- メモリに負荷を与えるもの
- スケジューラに負荷を与えるもの

メモリ負荷は、プロセッサ上の L1 キャッシュ及びそのラインサイズを考慮し、そのプロセッサから実施可能な最大量のメモリリード要求を永久に発行するプログラムである。両ドメインの共有資源であるメモリへ情報処理ドメイン側から負荷をかけることにより、機器制御ドメイン側の応答性能がどの程度変化するかを確認する目的で使用する。

スケジューリング負荷は、多数のスレッドを高速に切り替え続けるプログラムである。スケジューラなどのカーネル処理への負荷と、実行した CPU の L1 キャッシュを汚す効果を持つ。ドメイン分離効果を確認する目的で使用する。

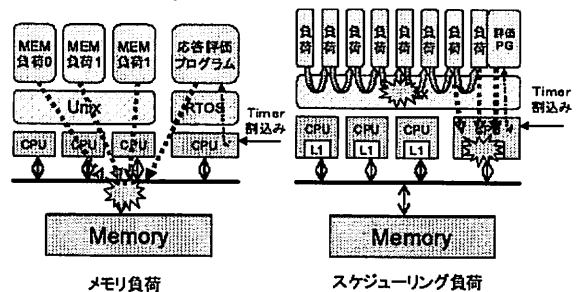


図 3 メモリとスケジューリング負荷

以上のような応答性能評価環境を用意し、各ドメイン分離環境の比較評価を行った。

### 4. 評価

#### 4.1. 評価環境

マルチコアプロセッサには、ARM MPCore[4]を使用した(図 4)。MPCore は 4 つの MP11 Processing Element (PE) を持ち、割り込み通知を各 PE 単位に分配できる割り込み分散機構を持つ。各 PE には命令/データ共に 32KByte の L1 キャッシュがあり、L1 キャッシュ内のデータは Snoop Control Unit(SCU)によってコヒーレンスが保たれる。また、SCU から AXI bus にアクセス要求が出力される際には、各要求は要求元の PE の ID を元にラウンドロビンに処理される。AXI bus の先には DDR

SDRAM が接続されており、OS を始め関連プログラムは全てここに配置される。

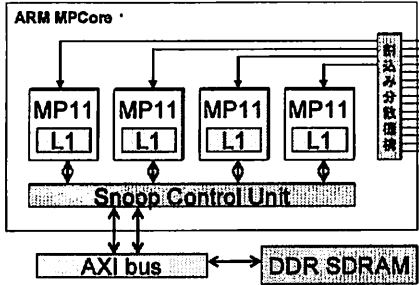


図 4 MPCore のブロック図

システム評価環境には ARM 社の Realview Emulation Board[4]を使用した。動作周波数は、MPCore が 210MHz、AXI bus 及び DRAM コントローラが各々 30MHz となっている。

MPCore が有する 4 つの PE のうち、PE3 を機器制御処理ドメインとし、残り PE0~PE2 を情報処理ドメインとして想定した。前者には RTOS として TOPPERS/JSP-1.4.2[5]を、後者には Unix 系 OS として Linux-2.6.22.1[6]を搭載した。

応答性能測定評価には、rtc\_wakeup プログラム[7]に割り込みハンドラ応答時間測定機能を加えたプログラムを用いた。この測定プログラムは割り込みハンドラとそのハンドラが呼ばれるのを待つプロセスからなる。割り込みハンドラは、1msec 毎の Timer 割り込みにより起動され、Timer のカウント値から応答時間を測定し、前記プロセスがその値を回収する。この測定プログラムを PE3 に配置し(図 3)この PE を機器制御ドメインと想定した。評価に使用する Timer 割り込みは、割り込み分散機構により PE3 だけに分配される。各環境の最小値/最悪値を測定する際に実施した割り込み回数は 4096 回である。

メモリ負荷は MPCore の L1 キャッシュサイズ 32KByte 及びラインサイズ 32Byte を考慮することで、その PE から実現可能な最大量のメモリリードアクセス負荷を実現した。メモリ負荷適用時には、このプログラムを情報処理ドメインの PE0, PE1, PE2 上で同時に実行させた。

スケジューリング負荷は、各スレッド専用排他変数とその排他変数の変化を待つスレッドを 50 生成し、排他変数を順に操作することにより多量のコンテキストスイッチを発生させてカーネルへの負荷を実現した。

#### 4.2. 評価結果

本論文では、割り込み応答時間の最悪値(最大値)と最小値、及び平均値について論じる。性能の達成は主に最悪値によって判断する。

まず最初に、図 1 に示した 4 つのドメイン分離環

境に対して、無負荷の場合、及び前記メモリ負荷、スケジューリング負荷を与えた際の機器制御ドメインにおける割り込みハンドラ応答時間を測定した。得られたハンドラ応答時間の最悪値の比較結果を図 5 に示す。

2 章で示したように本稿で目標とする割り込み応答時間は 50usec であるが、どのような負荷を与えてもこの規準を達成できたのは HeteroOS 環境だけであった。Affinity 環境や DualUnix 環境は、スケジューリング負荷に対してドメイン分離効果を示すことが出来ているが、最悪値は 100usec を超えており、機器制御ドメインの要求に達していない。また、メモリ負荷に対してはドメイン分離効果が見られなかった。

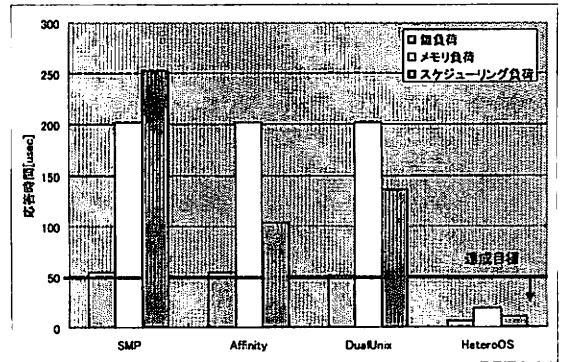


図 5 ハンドラ応答時間の最悪値

各環境の安定度の目安となる最小値と最悪値の差を次の図 6 にまとめた。

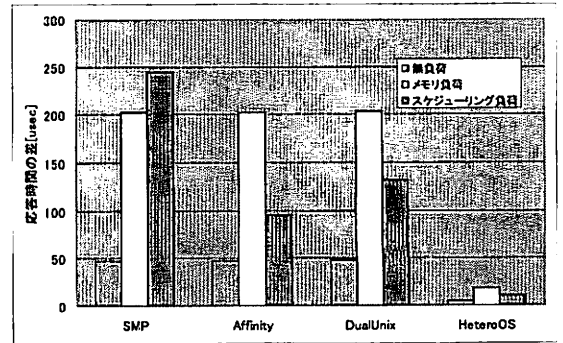


図 6 応答時間の最小/最悪値の差(安定度)

HeteroOS 環境では応答時間の最小値と最悪値の差が小さい、すなわち応答時間バラツキが小さく安定しているが、他の方式では応答時間バラツキが非常に大きく安定していないことがわかる。次節はこの原因を別の評価結果を用いながら解析する。

#### 4.3. 解析と改善検討

ここではメモリ負荷とスケジューリング負荷に対してドメイン分離環境間に差をもたらした原因を解析する。

### 4.3.1. メモリ負荷の影響

割込みハンドラ応答処理の最悪値の中でどの程度の割合をメモリアクセス時間が占めているのかを調べることで、メモリ負荷によって受ける影響の程度を明らかにできる。このメモリアクセス時間は、メモリ負荷プログラムを実行する PE 数を変化させて応答時間の最悪値を測定し、その負荷数に対する応答時間の増分を見ることでわかるはずである。なぜならば、今回用いている環境では、4章で述べたように各 PE からメモリへの要求は、要求元の PE の ID を元にラウンドロビンに処理されるからである。つまり、メモリ負荷として参加する PE 数が1つ増加すると、単位時間当たりには受理される要求は半減するため、要求しているメモリアクセス時間が倍となり、それが増分として評価値に現れると考えられる。

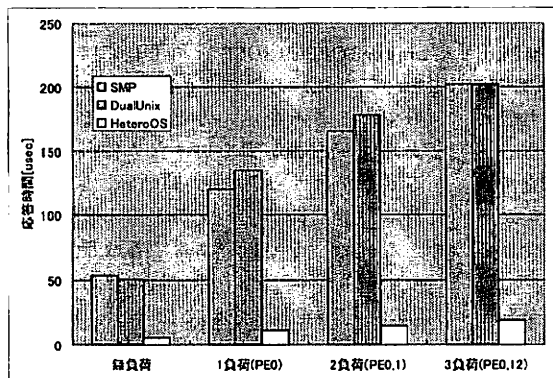


図 7 メモリ負荷数に対する応答時間の最悪値

図 7が負荷数を変化させたグラフである。メモリ負荷評価では負荷を適用する際に使用する PE を固定することが重要であるため、メモリ負荷に Affinity 機能を適用している。よって、SMP 環境と Affinity 環境での結果は同じとなるため、Affinity 環境をグラフから除外している。

SMP 環境における負荷数の増加に対する応答時間最悪値の変化に注目すると、負荷数が増加に伴い、平均 49μsec 応答時間が増加していることがわかる。先に述べたこの増分がメモリアクセス時間にあたるという推測に基づくと、最悪値が出る状況において、無負荷時では 49μsec、1 負荷時では 98μsec、2 負荷時では 147μsec、3 負荷時では 196μsec 分をメモリアクセスに費やしていることになる。他方、図 7エラー! 参照元が見つかりません。を参照すると、SMP 環境での最悪値は無負荷時で 54μsec、1 負荷時では 120μsec、2 負荷では 165μsec、3 負荷時では 202μsec である。これらを比較すると、SMP 環境での応答時間最悪値の 8 割～9 割がメモリアクセス時間によって占められていると考えられる。DualUnix 環境もまた同様の傾向を示して

いる。

また、HeteroOS 環境の負荷数増加時の応答時間最悪値増分は平均 4μsec である。よって、最悪値が出る状況において、無負荷時では 4μsec、1 負荷時では 8μsec、2 負荷時では 12μsec、3 負荷時では 16μsec 分をメモリアクセスに費やしていることになる。他方、HeteroOS 環境での各負荷数に対する最悪値は、無負荷時では 6μsec、1 負荷時では 11μsec、2 負荷時では 15μsec、3 負荷時では 19μsec であることから、最悪値の 6 割～8 割がメモリアクセス時間によって占めているといえる。

よって、ハンドラ応答時間の最悪値は、そのハンドラが呼ばれるまでに L1 キャッシュから外れてメインメモリへアクセスした量によってほぼ決まると考えることが出来る。

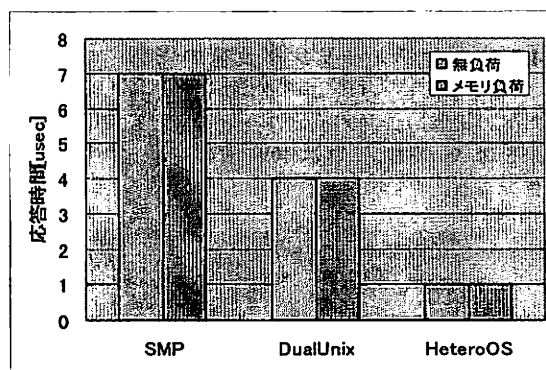


図 8 メモリ負荷に対する応答時間の最小値

一方、メモリ負荷に対する各環境の応答時間の最小値を調べると、負荷によって変動していないことがわかった(図 8)。このことにより、最小値が出る状況においては、外部メモリアクセスが実施されていないことがわかる。つまり、割込み処理に必要なとされるデータが L1 キャッシュに収まっていたと考えられる。

以上をまとめると

- 割込み応答時間の最悪値の大部分はメモリアクセス時間で占められている
- そのため、最小値と最悪値の差は、ハンドラが呼ばれるまでに L1 キャッシュから外れてメインメモリへアクセスした量によって決まる
- HeteroOS 環境とその他の環境との差は、その OS が応答処理に使用するメモリ量によって生じている

ことがわかった。このことから、このメモリ負荷影響の低減策は以下になる。

- メモリアクセス量を減らすこと
- メモリアクセスレイテンシを短くすること
- L1 キャッシュなどのローカルメモリを増やすこと

メモリアクセス量を減らす第一の方策は、多くの RTOS が行っているように必要最小限機能提供による footprint の削減である。したがって、マルチコア上のドメイン分離環境に RTOS を用いることは応答性能向上とその最小値と最悪値の幅を抑えることに対して非常に効果がある。

メモリアクセスレイテンシの向上のためには、バス及びメモリコントローラの周波数を向上させる必要がある。本評価はこの部分が30MHzと非常に遅い環境となっているが、これを高速化すればレイテンシの改善により応答時間の最悪値の大半を占めるメモリアクセス時間を改善できると考えられる。

L1 キャッシュの増量は、自ドメインへの負荷影響を低減するだけでなく、他ドメインへの影響低減効果もある。

### 4.3.2. スケジューリング負荷の影響

情報処理ドメインにスケジューリング負荷を与えた際の機器制御ドメインのハンドラ応答時間の測定結果を図 9 に示す。これは 4096 回のハンドラ実行の平均値である。

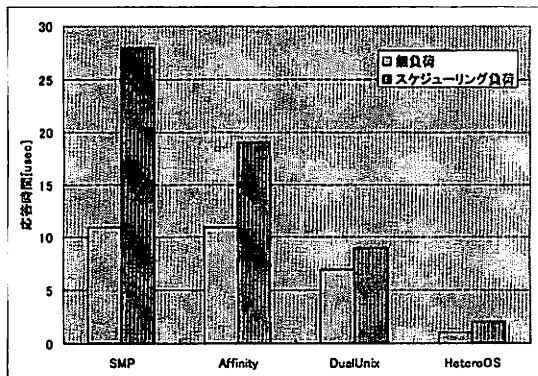


図 9 スケジューリング負荷時の平均応答時間

SMP 環境と Affinity 環境との差は、スケジューリング負荷を機器制御ドメインと隔離したことから生まれる。ドメイン分離により機器制御ドメインの L1 キャッシュヒット率が向上するため、平均応答時間が改善すると考えられる。

Affinity 環境と DualUnix 環境の差を考えてみると、Affinity 環境では一つの SMP OS が両方のドメインを管理しているため、一部カーネルデータなどを共有している可能性があるほか、クリティカルセクションによる影響が伝播する等、ドメイン分離が不完全による相互干渉があると考えられる。他方 DualUnix 環境はそのような問題を回避可能であり、その効果が Affinity 環境と DualUnix 環境の平均値に現れていると考えている。

また、DualUnix 環境や HeteroOS 環境の無負荷とス

ケジューリング負荷時を比較すると、OS 分離を実施したにも関わらず、数 usec 影響を受けていることがわかる。この環境ではまったく OS を共有していないので、スケジューリング負荷がメモリ負荷として作用しているものと考えられる。

OS 分離が負荷変動に対して安定な割込み応答性をもたらすということは前節までに分かったが、その結果をもたらす要因については、上記の通り様々な要素が複雑に関与しており、まだ完全に解析できていない。今後も引き続き調査する予定である。

## 5. 結論

今後の組込みシステムでは、機器制御処理と情報処理を両立させる必要があることを示し、その実現性を評価するために3つの形態(Affinity 環境, DualUnix 環境, HeteroOS 環境)のドメイン分離環境を提案した。また、そのドメイン分離環境をマルチコアプロセッサ MPCore と Linux, TOPPERS/JSP を利用して実装し、その応答性能を評価した。その結果、HeteroOS 環境が最も優れた環境であることを示すとともに、その割込み応答性能と安定性が機器制御を担うために十分な性能を持っていることがわかった。さらに情報処理ドメインからの負荷に機器制御ドメインがどのような影響を受けるのかを解析し、その影響に対する改善として、メモリアクセスの高速化と affinity による分離より完全な OS 分離の実施が重要であることを示した。

## 文献

- [1] Dynamic Security Domain Scaling on Symmetric Multiprocessors for Future High-End Embedded Systems. In *Proceedings of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, September 2007, 39-44.
- [2] 高信頼かつ高機能を実現する組込みシステム向け OS 共存システム, NEC 技術報告 Vol.60, No.2, 2007 年 4 月, 14-17.
- [3] 高信頼かつ高機能を実現する組込み向け OS 共存システム, SWEST9, 2007 年 8 月
- [4] ARM, <http://www.arm.com/>
- [5] TOPPERS プロジェクト, <http://www.toppers.jp/>
- [6] Linux, <http://www.kernel.org/>
- [7] rtc\_wakeup, <http://tapas.affenbande.org/wordpress/>