

Title	非対称型マルチプロセッサにおける動的分散リアルタイムスケジューリングに関する研究
Author(s)	石川, 智久
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3604
Rights	
Description	Supervisor: 田中 清史, 情報科学研究科, 修士

非対称型マルチプロセッサにおける動的分散 リアルタイムスケジューリングに関する研究

石川 智久 (510007)

北陸先端科学技術大学院大学 情報科学研究科

2007年2月8日

キーワード: 機能分散型マルチプロセッサ, 動的負荷分散, リアルタイムスケジューリング, 組み込みシステム.

1 はじめに

近年, 携帯電話などの組み込みシステムにおいて, システムの大規模化・複雑化が顕著である. それに伴い, プロセッサに要求される性能が高まっているが, 単一のプロセッサで消費電力を増やすことなく要求性能を満たすことが困難になってきた. このことから, 組み込みシステムにおいて, 処理性能の向上とともにシステム全体の電圧を下げるができるマルチプロセッサ構成が注目されている.

一般的に組み込みシステムでは, 動作するアプリケーションは事前に決められている. よって, DSP など特化したプロセッサが高負荷な処理を実行し, その他の雑多な処理を汎用プロセッサが実行することで効率化を行う. このように異種プロセッサを組み合わせた構成を機能分散マルチプロセッサ [1] (または非対称型マルチプロセッサ) という.

しかしながら, 機能分散型マルチプロセッサは異なる命令セットアーキテクチャ (ISA) から構成され, プロセッサごとに実行可能なバイナリが異なる. そのため, 局所的な高負荷が生じた場合に負荷を分散できず, デッドラインミスが発生する. 本研究では, 異種 ISA 間でタスクを移動する方式を提案し, 動的な負荷分散を行う. また, 提案方式によるデッドラインミス数の変化をシミュレータで計測し, 評価する.

2 リアルタイムシステムと機能分散型マルチプロセッサ

リアルタイムシステムでは, 処理がデッドライン時刻までに完了することを求められる. タスクがデッドラインミスした時の対処はリアルタイム性によって異なるが, 本研究では高負荷時にある程度のデッドラインミスが許容されるソフトリアルタイムシステムを

対象とする．また，本方式は各プロセッサにおいて静的優先度に基づいたスケジューリングを行う．

マルチプロセッサは対称型マルチプロセッサ (SMP) と機能分散型マルチプロセッサ (FDMP) に分類することができる．SMP は汎用システムにおいて利用されるマルチプロセッサ構成である．汎用システムでは，システムが実行されるまでどのような処理要求がいつ発生するかわからないため，全プロセッサが均質な SMP が用いられる．各処理は，処理要求発生時に OS によって決定されたプロセッサで実行される．

一方，組み込みシステムでは一般的に実行される処理は限定される．従って，処理に応じて特化したプロセッサを用意できる FDMP が利用されるが，各処理を実行可能なプロセッサは固定される．各処理はシステム設計者によって静的にプロセッサへ割り当てられるが，負荷の偏りが生じた場合，動的に負荷を分散することは難しい．本研究では，システム設計者の静的なタスク割り当てで対応できないケースにおいて，負荷分散を行う．

3 異種プロセッサ間のタスク移動方式

FDMP は ISA が異なるプロセッサから構成されるため，プロセッサ間で実行バイナリに互換性がない．そこで，各プロセッサがすべてのタスクの実行バイナリを事前に作成し，メモリに保持することで，タスクの移動が可能となる．

タスク移動の際に，プロセッサ間で移動対象となるタスクの情報を通信する必要がある．共有メモリ上に各プロセッサからアクセス可能な共有タスクプールを用意し，共有タスクプールに対して書き込み / 読み出しを行うことでプロセッサ間通信を実現する．

各プロセッサはスケジューラ実行時にデッドラインミス予測を行い，ミスが予測されるタスクが存在する場合は高負荷状態とする．ミスが予測されず，かつレディキュー上の実行タスク数が一定以下の場合は低負荷と判断される．高負荷，低負荷のどちらにも該当しない場合は通常状態とする．高負荷状態では，移動対象となったタスクを自身のレディキューから削除し，共有タスクプールに移動タスク情報を書き込む．低負荷状態では，共有タスクプールにスケジューリング可能なタスクが存在するかを確認する．該当するタスクが見つかった場合，共有タスクプールからそのタスク (の情報) を削除し，自身のレディキューに追加する．通常状態ではタスクの移動を行わず通常のスケジューリングを行う．

4 評価

評価環境として，機能分散型マルチプロセッサシミュレータを作成した．本シミュレータは，MIPS ISA32 シミュレータ [2] と TMS320C54x DSP ISA シミュレータ [3] から構成される．シミュレータの入力ファイルは各 ISA のバイナリである．タスクセットとスケジューラが記述されたソースファイルを MIPS コンパイラと DSP コンパイラでそれぞれ

コンパイルすることで、各シミュレータの入力バイナリが作成される。

シミュレータ上で提案スケジューラを実行し、デッドラインミス数と平均応答時間を計測した。MISP と DSP 間の負荷の偏りが異なるいくつかのタスクセットを用意し、それぞれの計測結果を比較した。その結果、2 倍以上の偏りがあるタスクセットにおいて、タスク移動によるデッドラインミス数の削減と平均応答時間の短縮を確認できた。

5 おわりに

本研究では、機能分散型マルチプロセッサにおいて、デッドラインミスが予測された場合、タスク移動を行う方式を提案・実装した。シミュレータ上でデッドラインミス数を計測し、提案方式を評価した結果、負荷の偏りがある状況でデッドラインミス数および平均応答時間が改善されることを確認した。

参考文献

- [1] “機能分散型マルチプロセッサ向けのリアルタイム OS”, 高田広章, 本田晋也, 情報処理 47 巻 1 号 2006 年
- [2] “MIPS32 Architecture For Programmers” Revision 2.50, MIPS Technologies Inc.
- [3] “TMS320C54x DSP Reference Set”, Texas Instruments Inc.