

## 1H-2

## 実時間用並列計算機アーキテクチャの検討\*

西田健次<sup>†</sup>、坂井修一、平木敬、戸田賢二、内堀義信、島田俊夫電子技術総合研究所<sup>‡</sup>

## 1 はじめに

センサフュージョンシステムシステム [1] は、多種類、大量のセンサから得られる情報を統一的に処理することにより、単一のセンサからは得られない新たなセンシング機能を付与しようというものである。センサフュージョンシステムシステムでは多数のセンサに対して情報収集/制御を行なうと同時に、各種センサからの情報を統合していくための高度な処理が行なうことが必要とされている。即ち、高度な実時間性と高度な計算能力の両立が求められている。

並列計算機は、高度な計算能力を実現するために非常に有効な方法であると考えられるが、実時間性を実現するのは困難であると考えられてきた。この主たる原因は、並列処理においては個々の計算、及び、個々の基本操作に必要な時間があらかじめ予測することが、極めて困難であるとされてきたことである。

本稿では、並列計算機上における予測可能性(最悪性能)を追求することで、並列計算機におけるハードデッドライン処理の実現について検討する。

## 2 実時間用並列計算機の目標

ハードデッドラインシステム [2] においては、一つのデッドラインミスがシステム全体のエラーにつながる。このため、デッドラインミスの発生する条件を解析可能なものとしなくてはならない。また、デッドラインを保証するために、スケジューリングは処理の最悪実行時間を仮定して行なわなくてはならない。このため、実時間用計算機においては最悪性能の向上がもっとも重要な目標となってくる。このためには、計算機上での個々の処理時間から不確定要素を除いていく必要がある。例えば、並列計算機の特有のプロセッサ間通信、同期コストなどの最悪値が予測可能なものでなくてはならない。また、プロセッサ上で実行中の処理が通信等によって妨害されることにより、処理時間が不確定となることも避けなくてはならない。

\*On Parallel Architecture for Hard Real-Time Applications

<sup>†</sup>Kenji NISHIDA, Shuichi SAKAI, Kei HIRAKI, Kenji TODA, Yoshinobu UCHIBORI, Toshio SHIMADA<sup>‡</sup>Electrotechnical Laboratory

実時間システムではデッドラインの迫っている処理を優先的に処理するため、優先度によるスケジューリングを導入している。逐次処理計算機上でのスケジューリングでは、最高優先度の処理に注目していれば実時間性を保証することができるが、並列計算機においては(プロセッサ台数を  $N$  として)上位  $N$  個の優先度を持つ処理についての実行が保証されなくては並列化によって実行効率が向上したとは言えない。そのため、システム全体では最高優先度でなくとも、上位  $N$  個の優先度に入る処理は最高優先度と同等に扱えるようにしなくてはならない。

従来の実時間用並列計算機の研究では、共有バス結合などが用いられてきた。しかし、通信負荷が増大した場合の共有バスの挙動には不確定要素が大きいため、並列化が最悪性能の向上に貢献していないと考えられる。通信ネットワークで結合された並列計算機では、通信負荷に対する挙動は予測し易い反面、ネットワーク上での衝突により通信時間の最悪値が予測困難になる。そこで、最悪通過時間を保証することのできるネットワークを用いれば、並列化により最悪性能を向上することが可能になり、高性能な実時間計算機を実現することが可能である。

## 3 スケジューリングと優先度処理

デッドラインの迫っているプロセスに高い優先度を与えることで、プロセス実行の実時間性を保証することができる。しかし、優先度のレベルが不足する場合はデッドラインとの正確な対応付けができなくなる。そのため、デッドラインが近づくにつれ、優先度を上昇させる必要が生じる。即ち動的な優先度処理が必要になるが、これはハードウェアでサポートするのは困難であり、また、ソフトウェア的に実現するのは非常に効率が悪い。特に並列計算機上で優先度を変化させる場合には、全体の同期を取る必要があり、処理効率を著しく落す可能性が高い。

そこで、プロセスのデッドラインを直接優先度として扱うことを考える。プロセスのデッドラインを絶対時刻で与えた場合、デッドラインが近づくにつれプロセスの優先度が上昇すると考えることができる。しか

し、一旦与えられた優先度の順位が逆転することはない。また、優先度の上昇していく速度は、全プロセスにわたって一定で良いと考えられる。そのため、プロセスに優先度を割り当てる際に、絶対時刻によるデッドラインを与えることができれば、その後は固定優先度と同等に扱うことができる。このような性質を持つ優先度のクラスをデッドラインクラスと呼び、スケジューリングの基本とする。

デッドラインクラスの優先度を実装するために必要なハードウェア要素は、十分な精度を持つシステムクロックと十分な幅を持つ優先度フィールドである。現在、システムクロックは10 $\mu$ sec刻みで32bit幅、優先度フィールドも32bit幅程度を想定している。

#### 4 実時間用並列計算機アーキテクチャ

実時間用並列計算機アーキテクチャは、前述のデッドラインクラスの優先度によるスケジューリングをサポートするためのハードウェア要素と、個々の処理の最悪実行時間を予測可能とするためのハードウェア要素を統合したものでなくてはならない。

プロセッシングエレメント(PE)で、命令実行時間を不確定にする可能性のある主な要素は、PE間通信及びプロセス切替えであると考えられる。低位優先度のプロセスの通信によって最高優先度の処理が妨害されてはならない、しかし、同時にシステム全体でN個の優先度を実行するためには、たとえ最高優先度のプロセスであっても低位の優先度のプロセスの通信を妨げてはならない。このため、命令実行を行なう実行パイプラインと、外部との通信を行なうパケット制御部を、PEの中で分離する。そして、実行パイプラインはプロセスに対する優先度キューを管理し、パケット制御部は入出力パケットに対する優先度キューを管理する。

実行中のプロセスよりも優先度の高いプロセスが実行可能になった場合には、実行中のプロセスを中断し高い優先度の処理の実行を開始しなくてはならない。このプリエンブションを含め実時間処理においては、プロセス切替えが頻繁に発生すると考えられる。そこで、プロセス切替えを高速化するために、レジスタファイルとメモリ間でのデータ転送幅を広くとる。また、レジスタセットを複数(2セット)持つことにより、次に実行可能になると予測されるプロセスをあらかじめロードしておくことで、更に高速なプロセス切替えを実現する。

キャッシュはメモリアクセスの平均的速度を向上させるには有効な手法であるが、ミスヒットすることを考えるとメモリアクセスの最悪時間を向上しているとは言えない。そこで、優先度の高いプロセスからアクセスされるデータを優先的にキャッシュへステージン

グすることを考える。これにより、高優先度のプロセスのミスヒットをなくし、メモリアクセス時間の最悪値を向上させる。

#### 5 現状

我々は、これまでCODA[3]を実時間用計算機とすることを検討してきた。CODAは命令実行とパケット通信機能が効率的に融合されたアーキテクチャを持つ。しかし、パケット処理を行なう際に、実行パイプラインに命令挿入を行なうため、パケット通信により命令実行が妨げられることになる。また、命令挿入が連続した場合(パケットが連続して到着した場合)には、命令の実行時間が予測できなくなる。

CODAでも複数のレジスタセットを持つことによりプロセス切替えの高速化を図っている。しかし、優先度による制御を行なっていないため、レジスタ上にあらかじめロードされているという保証が得られず、最悪値で考えなくてはならない実時間処理に対しては有効に活用することができない。

#### 6 おわりに

実時間用並列計算機を構成していくうえでの要件について検討した。最大性能や平均性能ではなく、最悪性能を向上させなくてはならないという点で、実時間用並列計算機は従来の計算機アーキテクチャとは異なる視点を待たなくてはならない。今後、我々はメモリシステムを中心とした最悪性能の向上とネットワークにおけるN個の優先度に対しての通過時間を保証する手法についての検討を進めていく予定である。

#### 謝辞

本研究を遂行するにあたり御指導、御討論いただいた棟上情報アーキテクチャ部長ならびに計算機方式研究室の同僚諸氏に感謝致します。

#### 参考文献

- [1] M. Ishikawa. Sensor Fusion System -Mechanism for Integration of Sensory Information-. *Journal of the Robotics Society of Japan*, 6(3):46-53, June 1989. (in Japanese).
- [2] J.A. Stankovic and K. Ramamritham, editors. *Hard Real-Time Systems*. IEEE Computer Society Press, 1988. ISBN 0-8186-0819-6.
- [3] 西田健次, 戸田賢二, 内堀義信, 坂井修一, 島田俊夫. 並列実時間実時間計算機 CODA の概要. 電子情報通信学会技術研究報告, vol. 90, no. 274, pages 45-50, Oct. 1990.