

## 多段結合網 R-Clos における通信スケジューリング

森村 知 弘<sup>†</sup> 岩井 啓 輔<sup>††</sup> 天野 英 晴<sup>†</sup>

自動並列化コンパイラと協調して性能向上を目指すマルチプロセッサ ASCA 用の結合網として提案された R-Clos に対して、コンパイラ・スケジューラがコンパイル時に通信パケットの経路選択または発行時刻をスケジューリングする手法を提案する。本スケジューリング手法によって、静的解析可能な通信に対し無衝突な通信が可能である。

提案したスケジューリング法を、衝突回避のためにスケジューラが挿入する待機コード (NOP) によるオーバーヘッドによって評価する。

### Communication Scheduling on the Multi-Stage Interconnection Network R-Clos

TOMOHIRO MORIMURA,<sup>†</sup> KEISUKE IWAI<sup>††</sup>  
and HIDEHARU AMANO<sup>†</sup>

A scheduling scheme for packets transfer on the multi-stage interconnection network R-Clos is proposed. Using this scheme at compile time, the parallelizing compiler and scheduler can arrange the packet's routes, and schedule the time of issuing packets to avoid packets collisions.

Through the simulation, the performance of scheduling scheme with 8 heuristic algorithms is analyzed and discussed.

#### 1. ま え が き

マルチプロセッサシステムにおいて、システムのピーク性能はプロセッサ数を増やすことにより容易に向上させることができる一方、プログラムを実行させた場合の性能：実効性能は、プログラム中に存在する並列性を有効に抽出できなければ、その向上は困難である。このため、ユーザが書いた逐次実行型プログラムの潜在的な並列性を抽出する自動並列化コンパイラ<sup>1)2)</sup>と、並列化したプログラムをマルチプロセッサ上で実行する技術の実用化が進んでいる。

これら自動並列化コンパイラの効果を最大限に発揮するためには、マルチプロセッサのアーキテクチャ、すなわち、結合方式、ハードウェアの各要素が、コンパイラ・スケジューラにとって扱い易い構成をとることが望ましい。我々はこのようなアーキテクチャを確立することを目的に、マルチプロセッサシステム ASCA<sup>3)</sup>を開発している。

並列処理においては、プロセッサ間通信に要する時間と、混雑による遅れが性能を支配する場合がある。これらのプロセッサ間通信が、自動並列化コンパイラ・スケジューラによってスケジューリング可能で、さらに結合網をこのスケジューリングにより無衝突にすることができれば、並列処理における重要なボトルネックを解消することができる。

本報告では、自動並列化コンパイラに特化したマルチプロセッサシステム ASCA の相互結合網 R-Clos<sup>4)</sup>におけるデータ転送の静的スケジューリング法を提案する。提案するスケジューリング法は、データ転送の時期と相手があらかじめ予測可能な交信について、それぞれの交信パケットの発行時期を調整することで、結合網内での衝突を回避することが可能である。

本報告の構成は、2章で ASCA の相互結合網 R-Clos を紹介し、3章で R-Clos 上における転送スケジューリング手法について述べる。4章では、転送スケジューラによって生じるオーバーヘッドの評価を示し、最後にまとめと今後の課題について述べる。

#### 2. R-Clos

R-Clos は、近傍の PE 間を接続するのに転送能力が高い Clos 網<sup>5)</sup>を用いることにより、転送遅延一定で低

<sup>†</sup> 慶應義塾大学理工学部, 神奈川県横浜市  
Faculty of Science and Technology, Keio University,  
Hiyoshi 3-14-1, Yokohama, 223-8522 Japan  
<sup>††</sup> 防衛大学校, 神奈川県横須賀市  
National Defense Academy, Hashirimizu 1-10-20, Yokosuka, 239-8686 Japan

レイテンシ、高バンド幅のクラスタ内通信を可能にしている。また、階層再帰構造をとることにより、Clos網内で閉じた通信を可能とし、階層の頂上に共有メモリを接続することによって、共有メモリとのグローバルな通信を、低コストで実現している。ここでは、スケジューリング手法を示す前にR-Closの構造について説明する。

### 2.1 R-Clos の構成

R-Clos<sup>4)</sup>は、高い転送能力を持つ3ステージ構成のMIN (Multistage Interconnection Network) であるClos網を基本ネットワークとし、この中間段に再帰階層構造を導入した階層構成MINである。

Clos網は、入力段のスイッチ (distributor) の入力数  $n$ 、中間段のスイッチ (exchanger) 数  $m$ 、そして入力/出力段のスイッチ数  $r$  の3つの要素によって  $V(m, n, r)$  として表され、この3つの要素の関係によってブロッキング・リアレンジブル・ノンブロッキングと性質を変えること<sup>5)</sup>が可能である。R-Closに基本ネットワークとして用いているClos網は、これら3つの要素が全て等しい  $k$  である  $V(k, k, k)$  で構成され、この場合はリアレンジブルで<sup>5)</sup>、Clos網1つに収まるPE数は  $k^2$  個である。

R-Closは、このリアレンジブルClos網の中間段のみを再帰的に階層化することによって、これらClos網を複数接続し、より大規模なシステムを接続・構成することが可能である。階層化によって接続されるClos網の数は、基本ネットワークであるClos網  $V(k, k, k)$  の  $k$  を構成基数として、階層数  $R$  のときの接続Clos網数は  $k^{(R-1)}$  である。ただし階層は、最小構成単位の  $k \times k$  のクロスバを階層0 (level 0) とし、基本ネットワークのClos網を階層1 (level 1) と定義する。またClos網の中間ステージを level 1 ステージと呼び、このステージのスイッチングエレメントを level-1 exchanger と呼ぶ。

ここでは、階層1から階層2のR-Closへの拡張を例に説明する。 $k$  個のClos網を接続するために、 $k$  個の  $k \times k$  のスイッチングエレメントで構成する1段の拡張ステージを作成する。これを level 2 ステージと呼び、このステージのスイッチングエレメントを level-2 exchanger と呼ぶ。各 level-1 exchanger に出力ポートを1つ追加し、これを level-2 exchanger の入力に接続することにより、 $k$  個の基本Clos網間を接続する。

同様にして階層  $i$  ( $i > 1$ ) を、level  $i - 1$  のステージを  $k$  個集めて形成することによって、 $k$  を基数として再帰的に階層化する。したがって、R-Closに接続される全PE数を  $P$  とすると、形成される階層の最大数  $R$  は  $\log_k P - 1$  で表され、このとき拡張ネットワーク部のステージ数  $S$  は  $R$  が2以上で  $R - 1$  となる。

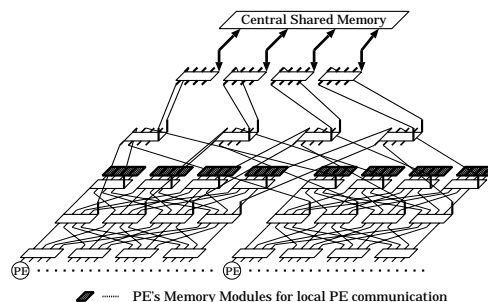


図1 R-Closを用いた共有メモリ型並列計算機の接続例

クラスタ構成基数の  $k$  を4、階層数を3として256個のPEをR-Closを用いて接続した場合の共有メモリ型並列計算機の接続例を図1に示す。明確化のため図1は、R-Closの一部のみを記述している。

### 2.2 ルーティング

R-Closのルーティングは、転送が基本ネットワーク部である送信元Clos網内で閉じている場合は、

$$T = \{*, d_1, d_0\}, (* = \forall z | z = 0, 1, \dots, k-1) \quad (1)$$

したがって、式1から、distributorではどの出力端子を使用しても正しくルーティングされる。

一方、転送が拡張ネットワーク部を経由した自Clos網外への転送の場合は、

$$T = \{d_1, \underbrace{k, \dots, k}_r, \underbrace{d_{r+1}, d_r, \dots, d_2, d_0}_r\} \quad (2)$$

ただし、 $k$  は、extended portを表す。式2から、拡張ネットワーク部を使用するデータ転送は、distributorの出力によって静的にただ1通りに定まる。

## 3. R-Closにおけるパケット転送スケジューリング手法

提案するスケジューリング手法は、R-Closの次の2つの性質を利用している：

1. 基本ネットワーク部のClos網の経路選択冗長性
2. 拡張ネットワーク部の経路の一意決定性

上記の2.によって送信元Clos網から外へ出るパケット転送に関しての経路を決定し、それから1.のClos網内の経路選択冗長性を活かして、効率の良い転送を実現する経路選択アルゴリズムを適用する。

### 3.1 スケジューリングの概要

パケット転送スケジューラの全体の大まかな流れについて説明する。ここでは、あるPE (Processing Element) から、あるパケットが時刻  $t_{sch}$  に発行されるとする。この発行時刻と受信相手は、コンパイラがあらかじめ予測可能と仮定する。本報告で対象とするASCA<sup>3)</sup>などコンパイラの予測可能性を高める設計が行なわれているマルチプロセッサにとっても全ての通信が予測可能ではないため、R-Closを構成するスイッ

チのアーキテクチャ・サポートが不可欠であるが、これについては紙面の都合上、割愛する。詳細については、文献<sup>6)</sup>を参照されたい。

ここでは発行時刻があらかじめ予測されているパケットを対象とし、発行PEのプログラム中に待機コード(NOP)を挿入して発行時刻を調整することで結合網内での衝突を回避する。スケジュールの流れは以下の通りである。

1. スケジューラは、 $t_{sch}$  に発行される全パケット  $P_{sch}$  について、それらの経路情報から經由する最高階層数  $l$  でパケットを分類し、分類したパケットの集合を  $P_{sch}(l)$  とする。例えば、宛先が送信元 Clos 網内である場合は、經由する経路最高階層は1で、そうでない場合で經由する最高階層が level-2 exchanger である場合は経路最高階層  $l$  は2である
2.  $l$  を1に設定する
3. 全  $P_{sch}$  について自由度のある distributor におけるルーティングを決定する。
4.  $P_{sch}(l)$  に対して、R-Clos のシミュレータ上で時刻  $t_{sch}$  から1つつ進めるとともにパケットを前進させていく。この際、各段の出力が複数パケットで競合する場合は調停を行い、調停の結果敗れたパケットを  $P_{sch}$  から取り去り、次の時刻に発行時刻を遅らせる。敗れたパケットの発行時刻はこれによって1つつ後ろにずらされる
5.  $P_{sch}(l)$  が目的地に到着したら、 $l$  を増やす
6.  $l$  が、システムの最高階層以下であれば5へ戻り、そうでなければ  $t_{sch}$  におけるスケジューリングを終了する

上記のようにパケットの発行時刻について各  $P_{sch}(l)$  をスケジューリングし、 $t_{sch}$  における R-Clos 上の全階層にわたるパケットの転送をスケジュールする。各階層、目的地まで到着したパケットは、スケジューリング成功パケットとして  $t_{sch}$  に発行されるパケットとしてコードを生成する。

また各段における調停の際に、各時刻におけるスイッチの状態(出力の予約状況、入力パケットなど)についてスケジュール表で管理する。スケジューラは、各時刻のスケジュール表を FIFO で保持し  $t_{sch}$  以降のスケジュールにこれらの情報を反映させている。しかしながら、全ての時刻のスケジュール表を持つのはメモリの無駄であるため、 $t_{sch}$  のスケジュールが終わると、以降のスケジューリングに必要な  $t_{sch}$  のスケジュール表を廃棄する。

### 3.2 distributor における経路決定

1段目の distributor における出力ポートの選択は、目的地と独立して行えるため自由度がある。したがってスケジューラは、1段目においてどの出力ポートを

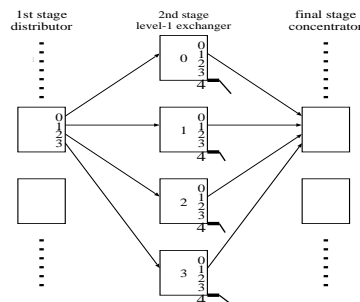


図2 concentrator に対する有効な経路

選択するかを調整することが可能である。

R-Clos の基本ネットワーク部が  $V(n, n, n)$  の Clos 網である場合、目的の concentrator に対して図2のように  $n$  個(図2では  $n$  個)の有効な経路が存在する。スケジューリング手法は、各パケット転送に対して適切な level-1 exchanger の割り当て (distributor の出力選択) を行うことによって、最大  $n$  個の経路を利用できるようにする。まずはじめに、経路冗長性のない送信元 Clos 網から外の Clos 網へ転送される Clos 網外パケット転送について level-1 exchanger の割り当てを行い、割り当てのない経路に対して送信元 Clos 網内のパケット転送についての level-1 exchanger の割り当てを行う。

### 3.3 Clos 網外転送のスケジューリング

Clos 網の外へ送信されるパケット転送に対してのスケジューリングについて説明する。R-Clos を構成する Clos 網が  $V(n, n, n)$  で構成されるとき、Clos 網から外へのリンクは  $n$  本である。この  $n$  本は、それぞれ level-1 exchanger 拡張出力ポートから1本ずつ level-2 exchanger の入力に接続される。それゆえスケジューラは、同時に最大  $n$  個のパケットが衝突しないで発行元 Clos 網の外へ送信されるように調整する。R-Clos では、経路が distributor の出力選択によって一意に定まるため、これより上の階層で衝突が起きた場合は、単にパケットの発行時間を遅らせることによって衝突を回避する。

#### 3.3.1 導入

説明のために、以下の用語を定義する:

##### 定義 1 $OAL_i$ (Outer Access List)

$distributor i$  から送信元 Clos 網外へ出るパケットが使用する level-1 exchanger の番号をそのパケットの  $age$  (アクセス発行されてからの時刻) の大きい順に並べたリスト

##### 定義 2 $OCA$ (Outer Connection Array)

level-1 exchanger の拡張出力 (上位の level-2 exchanger に接続) の予約状況を示す  $OCA_i$ , ( $0 \leq i \leq n$ ) を要素とする配列。  $OCA_i$  には、 $i$  番目の level-1

exchanger の拡張出力端子を予約した distributor の番号が入る

### 3.3.2 スケジューリング手順

スケジューラは、OAL (Outer Access List) 内のアクセスを distributor のスイッチ単位で順番に level-1 exchanger の拡張出力に割り当てていく。割り当てが完了すると、割り当てる拡張出力に対応する OCA に、割り当てられる distributor の番号を書き込み、後で行う Clos 網内転送スケジューリングで、distributor の出力選択の重複が起らないようにする。

どの distributor から割り当てを開始するかは、様々な方法が考えられるが、ここでは以下の要素によって、distributor の Clos 網外アクセスを重みづけすることとした。

- Age ...  $OAL_i$  を構成するパケットの age (最初に発行された時刻からの経過時刻) の最大値
- Nums...  $OAL_i$  を構成するパケットの数
- NodeAge ... パケットの送信元ノードの age (累積遅延)

これらを用いた選択手順としては、表 1 に挙げた 8 つのヒューリスティックなアルゴリズムが考えられる。これらのヒューリスティックなアルゴリズムに基づくスケジューリング法の評価については、4.1 節にて述べる。実際のスケジューラにおいては、R-Clos 内の  $V(n, n, n)$  の Clos 網の  $n$  が小さい場合はこれら全てについてスケジューリングを行い、最も小さいものを選択する。

表 1 割り当てリスト選択アルゴリズム

アルゴリズム	評価する順番
Nums-RR	1. Nums、2. RR
Nums-Age-RR	1. Nums、2. Age、3. RR
Age-RR	1. Age、2. RR
Age-Nums-RR	1. Age、2. Nums、3. RR
RR	1. RR
Nums-NodeAge-RR	1. Nums、2. NodeAge、3. RR
NodeAge-RR	1. NodeAge、2. RR
NodeAge-Nums-RR	1. NodeAge、2. Nums、3. RR

### 3.4 Clos 網内に対するスケジューリング

Clos 網を対象として、packing strategy と呼ばれる経路再形成の方法が提案されている<sup>7)</sup>。この方法は、Clos 網においてパケットの経路形成時の中間ステージの exchanger を選ぶ手法の 1 つで、これによってノンブロッキング転送を実現する中間ステージの数  $m$  をより小さい数で実現する。Clos 網内における衝突の全ては、2 段階目の level-1 exchanger の出線競合である。これは目的とする出力段 (level-0 concentrator) への接続を要求するパケットが level-1 exchanger で複数存在する場合である。そこでスケジューラは、level-1 exchanger の出力ポートを、そのスイッチに接続して

いる distributor によって予約し、競合しないように distributor の出力 (経路選択) を調整する。

#### 3.4.1 導入

説明のため、いくつかの用語の定義を行う。

#### 定義 3 $AL_i$ (Access List Set)

Clos 網内の  $i$  番目の distributor からのパケットの目的の出力段 (level-0 concentrator) の番号を age (アクセス発行時刻からの経過時刻) が大きい順に並べたリスト。

例えば、0 番目の distributor から、それぞれ 0, 1, 2, 2 への level-0 concentrator にデータが転送され、それぞれ age が 0, 3, 1, 2 である場合は、 $AL_0 = \{1, 2, 2, 0\}$  となる。

#### 定義 4 $VPC_i$ (Valid Port Counter)

concentrator  $i$  に対する現在の有効な経路数を示す。level-1 exchanger の出力  $i$  が割り当てられる度に値が 1 つ減る。すなわち、 $VPC_i = 0$  である場合、 $i$  番目の出力段 (level-0 concentrator) に接続可能な出力端子は一つも存在しないことを意味する。

#### 定義 5 予約行列: RM

$V(m, n, r)$  で表現される Clos 網における level-1 exchanger の出力端子の予約状況を  $m$  行  $r$  列で表した行列を予約行列  $RM$  と定義する。このとき  $RM$  は、以下のように表される:

$$RM = [RM_{i,j}], \quad (0 \leq i \leq m, 0 \leq j \leq r)$$

ただし、 $RM_{i,j}$  は、Clos 網内の  $i$  番目の level-1 exchanger の出力ポート  $j$  を予約した distributor の番号を示す。

#### 定義 6 出線競合検出行列: OM

$V(m, n, r)$  で表現される Clos 網における最終段 concentrator の出力端子の予約状況を  $n$  行  $r$  列で表した行列を出線競合検出行列  $OM$  と定義する。 $OM[i][j]$  は、 $i$  番目の concentrator の  $j$  番目の出力端子を予約しているパケットの情報を保持している。

出線競合検出行列  $OM$  は、最終段での出線競合が起こるのを予め検出し、Clos 網内パケット転送スケジューリングに出線競合を引き起こすパケットのうち最も高い優先度を持つものだけを取り扱わせ、競合して調停に負けた他のパケットが  $RM$  において level-1 exchanger を無駄に占有する可能性を排除している。

スケジューラは、この予約行列  $RM$  をもとに level-1 exchanger の割り当てを行う。

#### 3.4.2 スケジューリング手順

スケジューリングは、各 distributor が順番に空いてい

る出力ポートをスイッチ番号の小さい方から予約し、一通りの予約が行われると、全  $AL$  が空になるまで同じことを繰り返す。以下に詳細を示す。

アルゴリズム 1: Clos 網内パケット転送スケジューリング

1. ある時刻  $t_{sch}$  に発行されるパケットのルーティングタグから  $AL_i$  を生成し、以下の初期化を行う:
  - $VPC_i$  の値を  $r$  に設定
  - $CT_i$  の値を  $0$  に設定
  - $RM$  の要素、全てを負の値に設定
  - $OM$  の要素、全て無効な状態に設定
2. 全  $AL_i$  を走査することによって  $OM$  を作成し、衝突を検出したら調停に基づき、負けたアクセスを  $AL_i$  から取り除く
3.  $CT_i$  が  $1$  でないかつ空でない  $AL_i$  から  $AL_x$  を  $1$  つ選択
4.  $AL_x$  から、先頭のアクセス  $y$  (ただし  $y \neq x$ ) を一つとり出す
5. 空いている level-1 exchanger  $as$  の出力ポート ( $VPC_y > 0$  かつ  $OCA_y \neq as$  である) に  $y$  を割り当て、予約行列  $RM$  の該当する  $RM_{as,y}$  に  $x$  を、 $CT_x$  を  $1$  にセットし、 $VPC_y$  の値を一つ減らし  $7$  に進む
6.  $VPC_y = 0$  であれば、既に経路は全て埋まっているため  $t_{sch}$  における転送は不可能であるため、対応するパケットの発行時刻を遅らし  $8$  に進む
7.  $CT_i$  が  $1$  でない  $AL_{i,cs}$  が存在すれば  $3$  へ、そうでなければ次へ
8.  $CT_i$  を全て  $0$  にリセット ( $\forall i | 0 \leq i \leq m$ )
9.  $AL_{i,cs} \{ \forall i | 0 \leq i \leq m \}$  が空でなければ  $3$  へ、そうであれば次へ

上記の手順でスケジューリングが行われると、level-1 exchanger の出力の予約状況が  $VPC$  と  $RM$  と  $OCA$  によって管理されるため、衝突が起こることは無い。

Clos 網内転送スケジューリング法も Clos 網外転送スケジューリング手法と同様にして  $AL$  の選択について表 1 の 8 つのヒューリスティックなアルゴリズムが考えられるが、4.1 節で、評価・比較を行っている。スケジューラがこれらのうちどれを用いるかについては Clos 網外転送スケジュールと同様に、全てを試行して最も良かったものを採用する。

### 3.5 スケジュールアルゴリズムの計算量

対象のネットワークの全接続ノード (PE) 数を  $N$  とし、R-Clos の基本ネットワーク部の Clos 網を  $V(n,n,n)$  として、提案したスケジューリング手法の最悪の場合の計算量を求める。

Clos 網 ( $V(n,n,n)$ ) 内の転送に関するスケジューリングにおいては、出線競合の検出・排除に  $O(n^2)$  要する。さらに各 distributor の  $AL$  を作成するのに

$O(n \log_2 n)$  要し、 $AL$  を選択してポートを割り当てる動作を  $1$  巡すると最悪  $O(n^2)$  の計算量を要する。 $n$  巡するとすべてのアクセス割り当てが完了するので、Clos 網内転送スケジュールに要する計算量は  $O(n^3)$  となる。

一方、Clos 網外の転送についても、アルゴリズムの動作は同じであるためこの計算量以下で求まる。よって、R-Clos 全体では  $O(\frac{N}{n}n^3)$  の計算量で求まる。

## 4. 評価

ここでは提案したスケジューリング手法について、C 言語の random 関数を用いてランダムに生成されたアクセスパターンを用意し、これに対し表 1 で述べた Nums-RR、Nums-Age-RR、Age-RR、Age-Nums-RR、RR、Nums-NodeAge-RR、NodeAge-RR、NodeAge-Nums-RR の 8 つのヒューリスティックなアルゴリズムに基づいてスケジュールし、衝突回避のために挿入される待機コード (NOP) によってどの程度、完了時刻が増加するかを評価した。

### 4.1 スケジューリングによるオーバーヘッドの評価

はじめに、Clos 網内におけるスケジューリング手法の評価を示す。評価環境は、i). ネットワークサイズは 4PE 1 クラスタ、16 PE (1 階層) ii). アクセスパターンの生成ステップ数は 10000 ステップである。

評価の結果を図 3 に示す。グラフの横軸は、生成されるアクセスの発行率で、縦軸は全てのアクセスデータの発行が終了するステップ数を示している。アクセス発行率が最大の 100% の場合でアクセス完了ステップ数は、スケジューリング前のステップ数のおよそ 1.68 倍 (Nums-NodeAge-RR) から 1.79 倍 (Nums-Age-RR) になる。アルゴリズムの比較では、Age よりも NodeAge を優先して重みづけを行った場合の方が、全体的に約 4% から 6% 程向上する。これは、Age のようなアクセス単位の時間局所性による重みづけよりも送信元ノード単位の NodeAge で重みづけを行っ

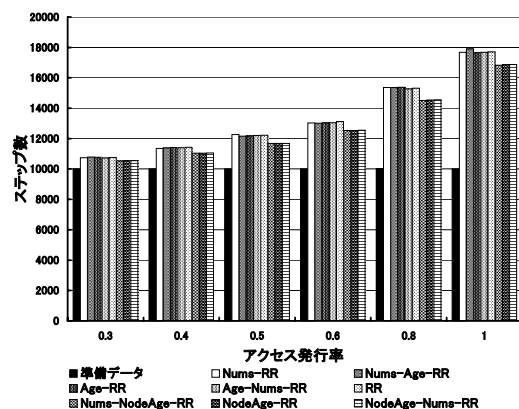


図 3 1 階層 16PE におけるスケジューリングオーバーヘッド

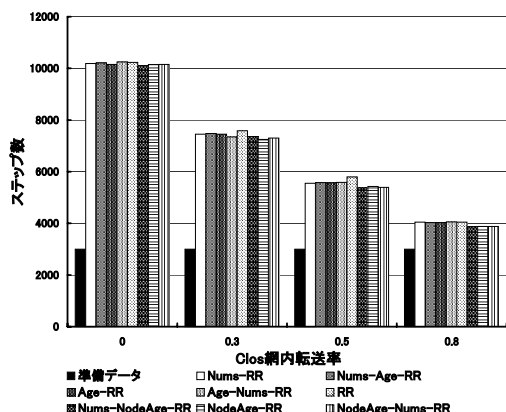


図4 2階層64PEにおけるスケジューリングオーバーヘッドの方が効果大きいことを示している。それ以外のアルゴリズムの差が出ないのは、衝突のほとんどが最終段における出線競合によるためである。

次に2階層のR-Closでのスケジューリング法の評価を示す。評価環境は、以下のとおりである: i). ネットワークサイズは4PE1クラスタ、64PE(2階層) ii). 生成ステップ数は3000 iii). 有効アクセス発行率は0.6 iv). アクセス中の発行元Clos網内転送発行率は0、0.3、0.5、0.8の4種類とした。

評価の結果を図4に示す。グラフの横軸は、アクセス中の送信元Clos網内転送の割合を示し、縦軸はアクセス完了時刻を示す。全てがClos網外への転送である場合はアクセス完了時刻は、3.36倍(Nums-NodeAge-RR)から3.41倍(Age-Nums-RR)に増加するが、Clos網内転送の割合が増加するにつれ減少し、80%がClos網内転送である場合では1.28倍(Nums-NodeAge-RR)にまで減少する。これはClos網外への転送スケジューリングがネットワークの上位階層のリンク数に依存し、時刻をずらすことによるのみ衝突が回避できるのに対し、Clos網内の転送では最終段での出線競合以外の衝突は、経路制御で回避できるためである。この結果から、近接アクセスの割合を高めるようにデータの配置を適切に行うことによってスケジューリングのオーバーヘッドを低減できることがわかる。アルゴリズムによる差は、16PEの場合と同様で、AgeよりもNodeAgeを利用している方が最大で8%程、オーバーヘッドを抑えることができる。その他のアルゴリズムの差異は微小であるが、わずかながらRRだけの場合のオーバーヘッドが大きくなる。

## 5. 結論と今後の課題

自動並列化コンパイラを行う静的スケジューリングを支援するために提案されたマルチプロセッサASCA用プロセッサ間相互結合網R-Closにおけるパケット転送スケジューリング手法を提案した。

本手法は、R-Clos内の基本ネットワーク部であるClos網内に関するパケット転送に対しては、経路選択冗長性を活かして経路をスケジューリングすることによって衝突を回避し、それ以外の転送の衝突や出線競合などの避けられない衝突についてはアクセス発行時刻を調整することによって無衝突なデータ転送を実現する。

提案したスケジューリング法によって衝突回避のために挿入される待機コード(NOP)によるアクセス完了時刻の増加は、Clos網内転送においては最大で1.67倍、Clos網間を含めた2階層のR-Closでは最大で3.36倍程になることが判った。

今後は、提案したスケジューリングアルゴリズムのさらなる性能向上を目指すとともに、ASCA用の並列化コンパイラが完成次第、マルチプロセッサASCAのシミュレータ上で実プログラムを動かすことによって、スケジューリング法の実効性能を評価する予定である。

## 謝 辞

本研究の一部は、半導体理工学研究センター(STAR-C)のプログラム”スタティックスケジューリング可能なマルチプロセッサ”の一環として行なわれた。

## 参 考 文 献

- 岡本 雅巳, 合田 憲人, 吉田 明正, 笠原 博徳, 成田 誠之助: ”実用レベルのマルチグレインFORTRANコンパイラの開発”, 情報処理学会研究報告96-ARC-106, pp. 43-48 (1996).
- H.Saito, N.Stavrakos, S.Carroll, C.D.Polychronopoulos and A.Nicolau: ”The Design of the PROMIS Compiler”, *Computational Complexity*, pp. 214-228 (1999).
- 岩井 啓輔, 森村 知弘, 川口 貴裕, 酒井 敦, 阿部 剛, 天野 英晴: ”コンパイラ主導型マルチプロセッサASCAのアーキテクチャ”, 並列処理シンポジウムJSPP2000論文集, pp. 3-11 (2000).
- Tomohiro Morimura, Keisuke Iwai and Hideharu Amano: ”Multistage Interconnection Network Recursive-Clos(R-Clos) : Emulating the hierarchical multi-bus”, *PDCS '98*, pp. 99-104 (1998).
- C. Clos: ”A study of non-blocking switching networks”, *Bell system Tech.J.*, Vol.32, pp.406-424 (1953).
- 森村 知弘, 田中 健介, 岩井 啓輔, 天野 英晴: ”スケジューリングを考慮した多段結合網スイッチチップの実装”, 電子情報通信学会 [VLSI設計技術] VLD2001, Vol. 101, No. 46, pp. 43-50 (2001).
- Y.Yang and J.Wang: ”Wide-Sense Nonblocking Clos Networks Under Packing Strategy”, *IEEE Trans. on Computers*, Vol. 48, No. 3, pp. 265-284 (1999).