1　はじめに

近年一般的になってきたマルチプロセッサ型スーパーコンピュータを効率的に利用するために、並列化コンパイラの研究が重要である. 当初は, 並列化コンパイラにおける並列性の抽出は, ループ内部の並列化 [2, 11] に重点が置かれていた. しかし, 複雑なデータ依存やループ外への条件分岐が原因で並列化できないループが存在するため, これまでのループ内部の並列化手法 (中粒度並列処理) では並列性の抽出の限界にきている.

そこで, 従来は抽出できなかった基本ブロック内部におけるステートメントレベルの粒度のタスクの近細粒度並列処理手法 [3], および基本ブロック・ループ・サブルーチン間の粗粒度並列処理手法 (マクロデータフロー)[6, 7] が提案されている. マクロデータフロー処理手法は, プログラムを幾つかのマクロタスク (MT) に分割し, MT 間の制御フロー解析及びデータ依存解析を行うことによって MT 間の並列度を最大限にまで引き出すことを目的としている. このマクロデータフロー処理を階層的におこなう [8] ことによって並列性を高める. さらに, 近細粒度並列処理手法・粗粒度並列処理手法 (マクロデータフロー処理手法) と従来のループ並列化手法を階層的かつ効果的に組み合わせたマルチグレイン並列処理手法 [1, 8, 10] が提案, 実装されている.

このようなマルチグレイン並列処理を効率良く行なうためには, アーキテクチャの点でも様々な工夫が必要になる. しかしこのようなアーキテクチャは Illinois 大学の Cedar[5] を除いて, ほとんど開発された例がない. そこで我々は, マルチグレイン並列処理コンパイラ **OSCAR Fortran Compiler**[6, 8] を支援するマルチプロセッサ, **ASCA**(Advanced Scheduling oriented Computer Architecture) プロジェクトを開始した [12]. ASCA がマルチグレイン並列処理を効果的におこなうためには, プロセッサ間を接続する結合網もこれに適したものが必要である. 特に近細粒度並列処理における無同期実行 [3] においてはネットワークの通信遅延が確実に予想できなければならない. また粒度によって要求されるネットワークの性質が異なるためこれらをそれぞれ個別に独立してサポートする必要がある.

そこで本報告では, マルチグレイン並列処理にふさわしいネットワークの 1 つとして階層型多重バスモデルを定義し, これをエミュレート可能な多段結合網 *R-Clos* 網を提案する. *R-Clos* 網は, *Clos* 網 [4] を階層構造が可能な形に拡張したブロッキング網である. しかし, ブロッキング網でありながら本稿で述べる条件の付加のみで各 Processing Element(PE) に対して, それぞれ仮想バスを一本ずつ割り当て可能な高性能なネットワークを実現することができる.

2　階層型多重バスモデル

2.1　マルチグレイン並列処理に適合するネットワークの条件

マルチグレイン並列処理は, プログラムを粒度によってマクロタスクに分割し, 各粒度における並列度を最大限に引き出す. したがって, その通信機構も粒度によって異なったものが要求される. このため, 各粒度における通信は, それぞれ独立して設けられ, お互いに干渉しあってはならない. そこで本章では, マルチグレイン並列処理をサポートするネットワークの通信機構を粗粒度, 中粒度, 近細粒度の各粒度ごとに検討する.

粗粒度並列処理におけるデータ転送は, 主に先行マクロタスクから後続マクロタスクへのデータ (配列, 配列の一部, または単純変数群) 転送である. このため一度に転送する量が多いことが特徴である. このためネットワークとしてはピーク転送容量の大きいものが求められる. また, マクロタスクの配置は動的になされるため, 転送先の事前予測は困難である. したがって転送時間を一意に決定することは不可能で, ある時点までに転送が終了していればよい. 後続のマクロタスクは, 先行タスクからのデータ転送が終了していることを確認してから実行を開始する.

中粒度並列処理 (ループ並列化) におけるデータ転送は, データのローカライゼーション [9, 10] を行なうことによって最小限に抑えられる. このため転送回数は少なく, 転送量も粗粒度並列処理のデータ転送よりも少ない. とくに, Doall Loop のイタレーションでは, PE(Processing Element) 間のデータの送受はない. また, 転送先を事前に予測することが可能である.

近細粒度並列処理におけるデータ転送は, ステートメント間の単純変数の受渡しで, 一度に転送する量は少ないが, 転送回数は多い. また転送先の事前予測は, 中粒度並列処理の時と同様に可能である.

これらに加えて最も重要なことは, コンパイラがスケジューリングをおこなう対象として構造が単純であることである.

まとめると, ネットワークに要求される性能は以下のようになる.

条件 **1** (マルチグレイン並列処理適合ネットワーク)

- スケジューリングの対象として構造が簡単
- 転送時間の保証 (近細粒度, 中粒度)
- 高速転送 (近細粒度)
- ピーク転送容量が大きいこと (粗粒度)
- クローズドパーティショニングが可能

これよりマルチグレイン並列処理に適合したネットワークとしては粒度によって階層的な通信構造をもち、かつその構造が簡単である必要があることがわかる.

2.2　階層型多重バスモデル

条件 1 を満たすネットワークとして, 階層型多重バスモデルを考える.

定義 **1** (階層型多重バスモデル)

[1]　各階層で物理クラスタを構成

[2]　クラスタ内でクラスタ構成 $PE(unit)$ 数と同数のバス

[3]　クラスタ内のバスのうち 1 本を, 上位階層への連絡バス ($JointBus$) として使用することができる

以上の定義により, クロスバ同様に 1 PE に対して 1 本のバスを割り当てることが可能になり, 階層構造によって粒度による独立した通信機構を確保することができる. 階層型多重バスモデルの概念図は, 図 2に示されている. 物理クラスタはある数を単位に階層的に Level-0,1,... と構成され, クラスタ内にはクラスタ構成単位数 (unit(ユニット))

本のバスが用意される．各 PE は unit 本の Bus から 1 本を Bus Interface によって獲得する (図 1)．クラスタ外にデータを転送する場合などは, unit 本のバスの 1 本を用いて, 上位階層に移り, 上位階層のバスを用いて他のクラスタに移る．これらの上位階層に移る際に用いられるバスを Joint Bus とよび, 定義 1の [3] に示したようにあるタイムスライスでは, クラスタ内で 1 本の Bus のみが Joint Bus として使用することが許される．

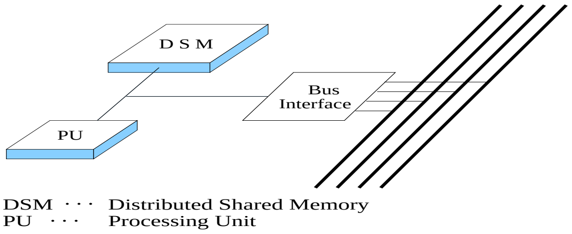


図 1: 階層型多重バスモデル -PE-

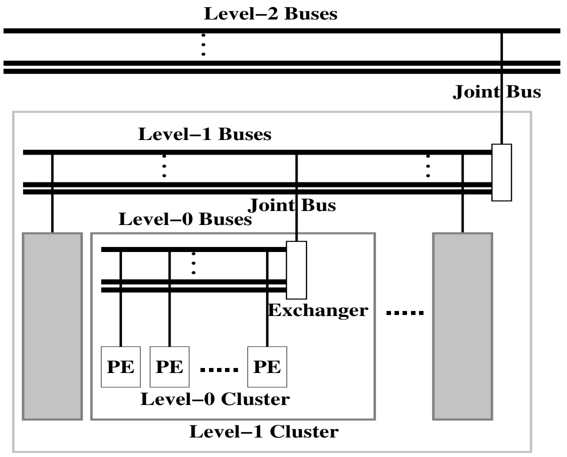


図 2: 階層型多重バスモデル -全体図-

3 R-Clos 網

前章で述べた階層型多重バスモデルは, 階層的な通信機構とクラスタ内に PE 数と同数のバスの提供から, マルチグレイン並列処理に適したネットワークである．しかしながら実際に, この多重バスをハードウェアで実装することは困難で, スケーラビリティがない．そこで, 多段結合網 (MIN)Recursive-Clos(R-Clos) によってこの階層型多重バスモデルをエミュレートすることによって高いスケーラビリティを保証する．本章では, *R-Clos* 網について詳しく述べる．

3.1 構成

R-Clos 網は, *Clos* 網を基本構成単位として階層的に拡張した構成をとる．

Clos 網は, 3 ステージ (Distributer, Exchanger, Concentrater) の MIN で, その接続は, スイッチの出力を次のステージの各スイッチに入力させる．Distributer 及び Concentrater の入力数, 出力数と Exchanger の数によって, 網の能力がノンブロッキング, リアレンジブル, ブロッキングと変わる．

R-Clos 網では, この *Clos* 網を構成する 3 ステージ (基本ステージ) に, 拡張ステージと呼ばれるさらに幾つかのステージを付加する．基本ステージは, Distriburer と Concentrater の入力数及び出力数と Exchanger のスイッチングエレメント数を同じにした *Clos* 網で構成される．拡張ステージは, 基本ステージの Exchanger にさらに *Clos* 網を階層化するためのリンクを設け, このリンクを拡張ステージ (rank 1, . . . , Exchanger) に繋ぐ．上位階層へのリンクのため, Exchanger の出力端子 1 つを用意する．

クラスタの構成は, ユニット数を第 1 ステージのスイッチングエレメント (Distributer) 1 つの入力数と同数に定め, 階層的かつ再帰的に Level-0 , 1 , . . . , と構成される．つまり, Level-N クラスタは, Level-(N−1) クラスタをユニット数集めて構成されることになる (したがって, Level-N クラスタ内の PE 数は $unit^{(N+1)}$ 個)．ユニット数 4, 全 PE 数が 64 である場合の *R-Clos* 網は, 図 3に示されている．図 3のように Level-1 クラスタまでは, *Clos* 網の中におさまる．

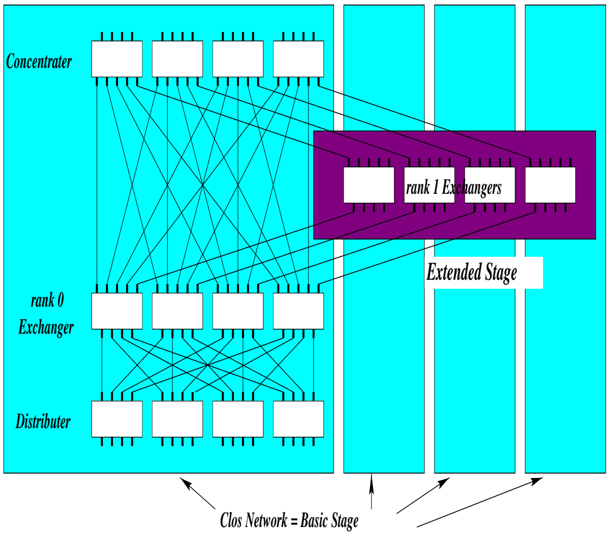


図 3: *R-Clos* 網 の構成 - unit 4, 64 PE

Level-1 クラスタ (*Clos* 網) を基点として, クラスタを何階層重ねるかは, システムサイズ (PE 数) に依存し, 重ねる階層を R, PE 数を P とすると,

$$R = \log_{unit} P - 1$$

と表わせる．

なお, 階層が 2 以上のとき中間ステージは以下の 2 通りの構成方法がある．

- Circulate Linking Model(CLM)
  このモデルでは, rank n と rank n-1 が循環している構造になっている．これは rank n-1 Exchanger に 2 つの機能を併用しているためで, その 2 つの機能とは, 1 つはパケットを上位階層に送り出すという Exchanger としての機能と, もう 1 つは上位階層から下位階層に降りてきたパケットを下位クラスタに配送する rank n-1 Concentrater としての機能である．unit 数 4 で構成した場合の CLM のモデルを図 4に示す．

  このモデルの特徴は, 使用するスイッチの個数が後述の Non-Circulate Linking Model(NCLM) よりも少な

くてすむことである．ただし，循環したネットワークとなるため，rank n-1 と rank n のパケットの衝突が発生する．

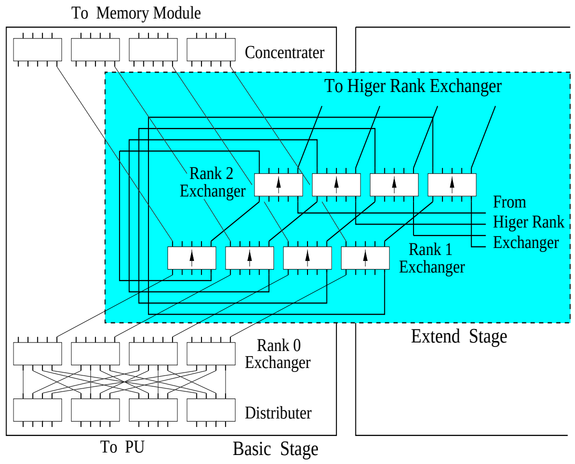


図 4: *R-Clos* 網 - CLM unit 4

- Non Circulate Linking Model(NCLM)
  CLM とは異なり,Exchanger とは別に rank 毎に Concentrater を設けている．このため,CLM のように rank n-1 と rank n が循環するという構成はとらない．図 5 に unit = 4 の場合のモデルが示されている． このモデルでは,CLM と異なり,rank n Exchanger に到達したパケットは rank n-1 Exchanger に循環することはなく,rank n-1 Concentrater に向かうことになる．このため CLM のように循環による衝突は起こらない．ただし，入力 2 つに対し出力が多数あるため，使われない出力が多く使用効率が悪い．また，循環構造でないため，階層を増やせば増やす程中間ステージの段数が増える．

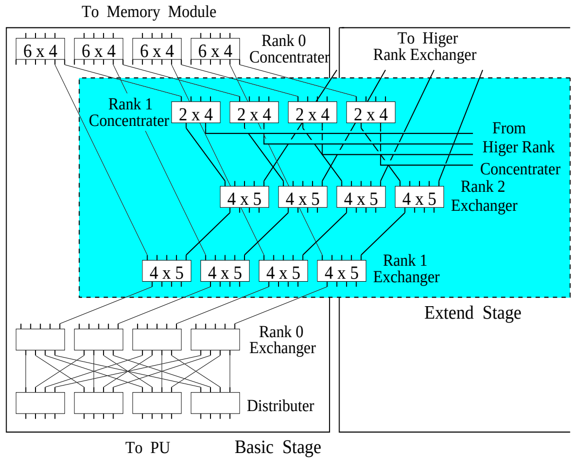


図 5: *R-Clos* 網 - NCLM unit 4

  このとき R-Clos 網の拡張ステージの段数 S は,

$$S = 2 \times R - 3$$

  となる (CLM では,S=(R-1))．したがって CLM に比べ，スケーラビィリティはない．

いずれの場合も，図中の実線部のみを抽出するとただの Clos 網となる．

3.2 *R-Clos* 網の接続定義

第一ステージのスイッチを Distributer として、中間ステージを Exchanger(NCLM では、Concentrater も含む)、最終ステージを Concentrater(NCLM では、rank 0 Concentrater) とする．スイッチの入力・出力端子の位置を

ステージ名 (スイッチ番号, スイッチ内でのユニット番号)

で表す．Distributer のスイッチ番号は，Level-0 クラスタの番号と同じであるので，スイッチ内のユニット番号は Level-0 クラスタ内の何番目のノードであるかも示している．クラスタ構成単位 (unit) 数を u で示すと，2 つのモデルの接続は，以下のように定義される．

定義 2 *CLM(Circulate Linking Model)*

$$D_{out}(j,k) \rightarrow E_{0in}(k + j - j \ \mathbf{mod}\ u, j \ \mathbf{mod}\ u) \tag{1}$$

$$E_{0out}(j,k) \rightarrow \begin{cases} E_{1in}(u \times (\frac{j}{u^2}) + j \ \mathbf{mod}\ u,\ (\frac{j}{u}) \ \mathbf{mod}\ u) & (k = u) \\ C_{in}(k + j - j \ \mathbf{mod}\ u, j \ \mathbf{mod}\ u) & (k < u) \end{cases} \tag{2}$$

$$E_{Iout}(j,k) \rightarrow \begin{cases} E_{(I+1)in}(u \times (\frac{j}{u^2}) + j \ \mathbf{mod}\ u,\ (\frac{j}{u}) \ \mathbf{mod}\ u) & (k = u) \\ E_{(I-1)in}(u^2 \times (\frac{j}{u}) + u \times k + j \ \mathbf{mod}\ u,\ u) & (k < u) \end{cases} \tag{3}$$

ただし，$1 \leq I \leq R - 1$ である．

NCLM と CLM では接続は基本的に同じであるが，NCLM では循環構造になっていないため $E_I \rightarrow E_{I-1}$ のかわりに Concentrater が各階層にあり $E_I \rightarrow C_{I-1}$ となっている．

また，$C_I \rightarrow C_{I-1}$ というリンクが加わる (R≥ 3 の場合)．

定義 3 *NCLM(Non-Circulate Linking Model)*

$$D_{out}(j,k) \rightarrow E_{0in}(k + j - j \ \ \mathbf{mod}\ u,\ j \ \mathbf{mod}\ u) \tag{4}$$

$$E_{0out}(j,k) \rightarrow \begin{cases} E_{1in}(u \times (\frac{j}{u^2}) + j \ \mathbf{mod}\ u,\ (\frac{j}{u}) \ \mathbf{mod}\ u) & (k = u) \\ C_{0in}(k + j - j \ \mathbf{mod}\ u, j \ \mathbf{mod}\ u) & (k < u) \end{cases} \tag{5}$$

$$E_{Iout}(j,k) \rightarrow \begin{cases} E_{(I+1)in}(u \times (\frac{j}{u^2}) + j \ \mathbf{mod}\ u,\ (\frac{j}{u}) \ \mathbf{mod}\ n) & (k = u) \\ C_{(I-1)in}(u^2 \times (\frac{j}{u}) + u \times k + j \ \mathbf{mod}\ u,\ 0) & (k < u)(I > 1) \\ C_{0in}(u^2 \times (\frac{j}{u}) + u \times k + j \ \mathbf{mod}\ u,\ unit) & (k < u)(I = 1) \end{cases} \tag{6}$$

$$C_{Iout}(j,k) \rightarrow \begin{cases} C_{(I-1)in}(u^2 \times (\frac{j}{u}) + u \times k + j \ \mathbf{mod}\ u,\ 1) & (I \geq 2) \\ C_{0in}(u^2 \times (\frac{j}{u}) + u \times k + j \ \mathbf{mod}\ u,\ u + 1) & (I = 1) \end{cases} \tag{7}$$

ただし,$1 \leq I \leq R - 1$ で，$E_{0out} \rightarrow C_{0in}$ の接続は CLM と同様のため省略した．Concentrater と Exchanger が分離しているため中間ステージにおける上位階層に出るパケットと下位階層に戻ってくるパケット同士の衝突がない．

いずれのモデルでも Concentrater では上位階層から送られてきたパケットと下位階層から送られてきたパケットの衝突が起こる．しかしそれ以外は，簡単なルーティングアルゴリズムで、階層バスを衝突なしで実現することができる。

3.3 *R-Clos* 網のルーティング

$D_{in}(j,k) \rightarrow C_{out}(p,q)$ の通信を行なう場合のルーティングについて考える．p が，

$$\begin{aligned} p &= a_U \times unit^U + a_{U-1} \times unit^{U-1} + \ldots + a_0 \times unit^0 \\ &= \sum_{k=0}^{U} (a_k \times unit^k) \end{aligned} \tag{8}$$

と表せるとき，$a_U \ldots a_0$ は，p (スイッチ番号) を unit 進数であらわしたものである．*R-Clos* 網におけるルーティングは，その構造 (Level-1 クラスタ内は *Clos* 網) 上，通信が Level-1 クラスタの内か外かによって異なるため，Level-1 クラスタ内通信と Level-1 クラスタ外への通信の場合にわけて述べる．

1. Level-1 クラスタ内通信 (下位通信)
   このとき，routing_Tag は，式 (1), (2),(4), (5) から導くと以下のように求まる．

   $$T(\ *\ ,\ a0\ ,\ q\ ) \tag{9}$$

   となる．ただし＊は，0 ~ unit−1 の任意の数字である．

   routingTag の 1 番目の数字は，Distributer の出力端子を意味し，2 番目の数字は rank 0 Exchanger, 3 番目の数字は Concentrater の出力端子をそれぞれ表している．

   最大の利点は、Level-1 内通信において Distributer では任意の出力端子を用いても目的地にルーティングできるという点である．

2. Level-1 クラスタ外通信 (上位通信)
   CLM の場合，中間ステージが循環しているため冗長なルーティングも存在するが，本論文ではマルチグレイン並列処理に適したネットワークを指向しているため，転送時間の予測は重要であり，また階層を重ねる程ネットワークが細くなる構造をしている *R-Clos* 網に用いれば却ってネットワークの混雑をまねくため，これら冗長なルーティングを一切排除する．

   この条件をもとに，routing_Tag を 定義 2, 3及び式 (8) より求める．このとき，図 6のアルゴリズムによって決定される通信使用ランク R’ を用いて

   $$T(\ a_0,\ \underbrace{unit, \ldots,\ unit}_{R'},\ a_U,\ a_{U-1},\ \ldots,\ a_{U-R'+1},\ q) \tag{10}$$

   と表せる．

   Level-1 クラスタ内通信との違いは，Distributer における自由度がまるっきり無い点である．したがって，Level-1 外通信では，ルーティングは静的にただ一通りに定まる．

パケットのルーティングに必要な情報は，そのパケットが Level-1 クラスタ内通信かどうかのビット，行き先を unit 進数にしたもの，および通信使用ランク R’ のみである．

```
j’ = j;
p’ = p;
c  = 0;
while( j’ != p’ ){
        j’ = j’ / unit;
        p’ = p’ / unit;
        c ++;
}
R’ = c --;
```

図 6: 通信使用ランク決定アルゴリズム - C 言語 -

3.4 *R-Clos* 網の階層型多重バスのエミュレーションの証明

以上までに述べられた *R-Clos* 網の接続，及びルーティングによって *R-Clos* 網に対して以下のような定理が導ける．

定理 1 *R-Clos* 網は，階層型多重バスモデルをエミュレート可能で，そのモデル下における衝突は，バスの出線競合に相当するもの以外は起こらない．

この定理を証明するためには，任意のレベルにおいて階層型多重バスモデルをノンブロッキングでエミュレート可能であることと，各レベルで異なるレベル同士の通信がお互いを妨げないことを示せばよい．これより，Level-0, 1, N ($N > 1$) について検討する．

Level-0

Level-0 通信は，Level-0 クラスタ内の通信であるから，Distributer と Concentrater は，同じスイッチ番号でなければならない．また，階層型多重バスでは，各 Level においてクラスタ構成ユニット数と同数のバスを割り当てられることを保証している．したがって，ユニットが 4 で Level-0 クラスタを構成する場合は、Level-0 Bus は，4 本割り当てられなければならない．

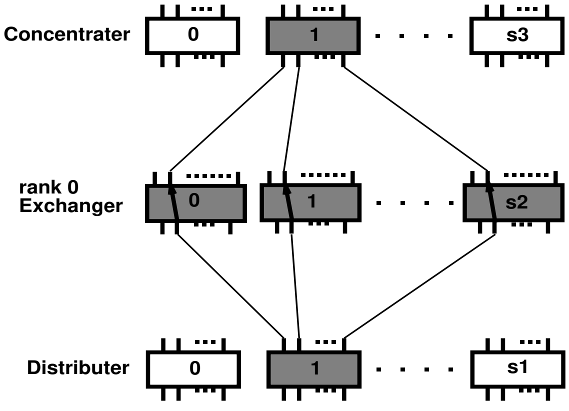


図 7: *R-Clos* 網における Level-0 Bus

(証明) *R-Clos* 網は, *3.1*節で述べたように図 *7*において

$$s_1 = s_2 = s_3$$

また, クラスタ構成ユニット数を $unit$ とすると, $Distributer$ は $unit \times unit$ のクロスバで構成され,

$$s_2 = unit$$

である. $D_{in}(j,k) \rightarrow C_{out}(p,q)$ のルーティングについて考えると, *Level-0* 通信では

$$j = p$$

である. このときルーティング T_0 は式 (9) より

$$T_0 = (*, j, q)$$

である. ここで, $0 \leq * \leq unit - 1$ の任意の数である. $D_{out}(s,t) \rightarrow E_{0in}(t,u)$ のリンクを $L_0(s,t)$, また最終ステージの $Concentrater$ に接続されているリンク $E_{0out}(v,w) \rightarrow C_{in}(w,z)$ を $L_{last}(v,w)$ と表す. この時, クラスタ S における *Level-0* 通信は, ルーティングタグより $L_0(S,x), L_{last}(x,S) (0 \leq x \leq s_2)$ によって実現される. 従って、このリンクが *Level-0 Bus* に対応している. 階層型多重バスでは, 各階層に $unit$ 本の Bus を用意しなければならないが, L_0,L_{last} で, $s2 = unit$ 本 それぞれあるためこの条件を満たしている. また, これら *Level-0 Bus* は, それぞれ異なる $Exchanger$ で異なる出力を選ぶため, 最終ステージにおいて同じユニットを選ぶ (出線競合) 以外では同じ *Level-0* 通信同士は衝突しない. *Level-1* 通信とは, 以降に述べる条件 *2*によって, 衝突が回避可能である. その他, 上位通信と最終ステージで時間差衝突が発生する可能性があるが, これはレベルごとに通信遅延がことなるためで, 実際の階層型バスでも発生するため問題としない.

Level-1

Level-1 通信は, Level-1 クラスタ内で行なわれ, Level-0 クラスタ間の通信である. 階層型多重バスモデルでは, Level-1 通信は, 各 Level-0 クラスタでそれぞれ 1 つのみ出すことが許され, Level-1 クラスタ内で unit 数の Level-1 通信が許容されることになる. Level-1 通信において, 階層型多重バスモデルでは Level-0 Bus を Joint Bus として用いて Level-1 に行き Level-1 Bus で, 目的の Level-0 クラスタを選択し, Level-0 Bus で Level-0 クラスタ内の目的のユニットに到着する. したがって, Level-1 Bus は, 各 Level-0 クラスタから 1 本ずつ用意されれば仕様を満たす. この Level-1 Bus は, 実際の MIN 上では, Level-0 Bus と 同じ段数を経て目的地に到達するが, rank 0 Exchanger において Level-0 Bus が自分のクラスタ番号を出力に選ぶのに対し, Level-1 Bus では自分以外のクラスタ番号を出力に選ぶ点が異なる. しかし無条件に, Level-1 通信と Level-0 通信を出すと衝突が発生する (図 8). そこで階層型多重バスモデルを満たすために以下 のように Level-1 通信をおこなう際のルールを定める.

この衝突は, Level-1 通信がステージ 2 でどのスイッチを使用するかが事前に定まっていないため Level-1 通信がステージ 2 で拡散するために発生する. そこで, Level-1 通信が, Distributer において任意の出力端子が選べるという利点を用いて, Level-1 通信が使用するスイッチをステージ 2 に一つ設け, Level-1 通信を 1 つのスイッチに集めてしまう. このスイッチを, **Level-1 優先スイッチ**とする. Level-1 優先スイッチは, 以下のように使用される.

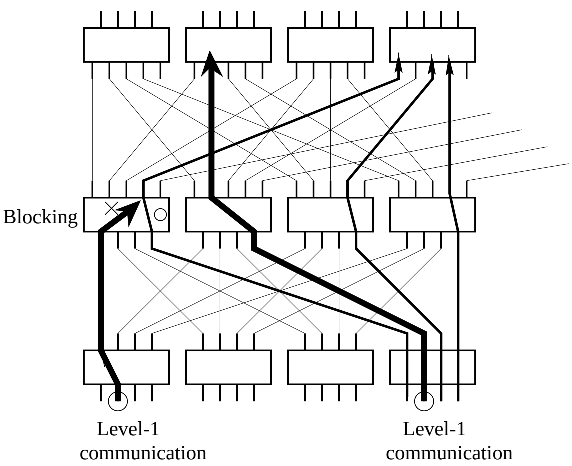


図 8: Level-0 通信と Level-1 通信の衝突

条件 **2 (Level-1** 優先スイッチ 使用条件)

- *Level-1* 通信は, 必ず *Level-1* 優先スイッチを用いる
- *Level-0* 通信は, ステージ *2* の他のスイッチが全て使用されているときのみ *Level-1* 優先スイッチを使用することができる
- *Level-N* ($N > 1$) 通信は, *Level-1* 優先スイッチを使用可能

Level-1 優先スイッチは, *R-Clos* 網上の表現であるが, 階層バスとしては, Level-0 Bus のうち 1 本を Level-1 優先バスとして使用することになる.

2 番目の条件は, クラスタ内の全ての通信が Level-0 通信である場合は他のクラスタからの通信よりも優先されることを示している. これは, 近細粒度通信が厳密に転送時間が保証されなければならないことに依る.

3 番目の条件は, Level-N ($N > 1$) 通信の Distributer においてのルーティングが静的にただ 1 通りに定まるために必要となる.

これらをまとめると以下のようになる.

(証明) *Level-1* 通信に対して, 条件 *2*の優先スイッチを設けることによって, *Level-1* 通信が *1* つの $Exchanger$ にまとめられる. したがって, *Level-0* 通信とは独立したパスを設立できるため *Level-0* とは衝突しない. また, *Level-1* 通信はこのスイッチを通るため, スイッチの入力数 (= 出力数) だけ *Level-1* 通信がエミュレート可能である. したがって *Level-1* の階層でユニット本の *Level-1 Bus* が確保できることから *Level-1* では階層型多重バスモデルを満たす. *Level-N* ($N > 1$) 通信とは, $Distributer$ においては, *Level-1* 通信が任意の出力でルーティング可能であるため, $Distributer$ では衝突は発生しない. *rank 0 Exchanger* では, *Level-N* 通信と *Level-1* 通信では, 異なる出力を選択する. このため, *Level-N* 通信とは衝突が起こらない.

補足として, 同じクラスタを要求した際に起こる衝突はバスであってもおこる出線競合であるため問題ではない.

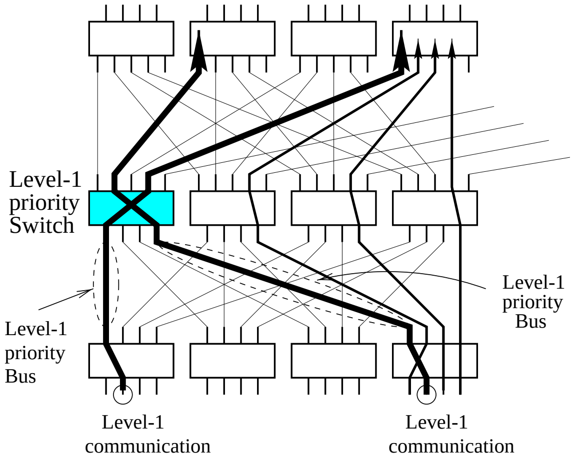


図 9: Level-1 優先バス及び 優先スイッチの設定

Level-N$(N > 1)$

(証明) *rank n-1 Exchanger* から *rank n Exchanger* へのリンクが *Joint Bus* に相当する．*rank n Exchanger* には，*unit* 数の *rank n-1 Exchanger* (もしくは *Concentrater*) へのリンクが存在し，このリンクを選択することが *Level-N Bus* の選択に相当する．したがって *rank n Exchanger* においても，*unit* 本の *Level N Bus* が用意されていることになるため階層型多重バスモデルを満たす (図 10)．*Level-N* 通信は，*rank 0 Exchanger* において *Level-0*，および *Level-1* 通信とは異なる出力を選ぶ (式 (9), (10)) ためこれらとは衝突しない．*Level-N* ($N > 1$) 同士の通信では *rank n* ($n > 0$)*Exchanger* において，同じ出力端子を選択すると衝突が起こる．しかしこれは，*Level-N Bus* の出線競合であるため実際の階層バスでも起こる衝突である．したがって *Level-N* 通信でも出線競合以外の衝突は起こらない．

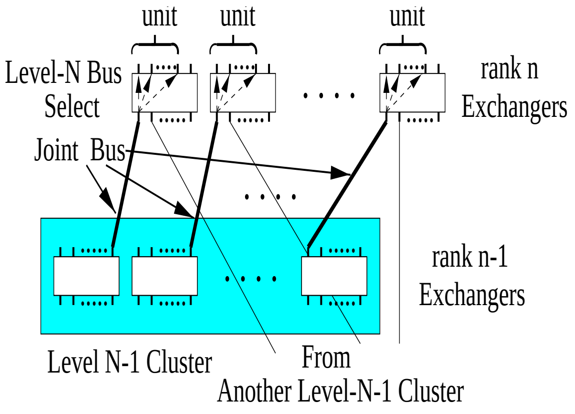


図 10: *R-Clos* 網上での Level-N Bus のエミュレート

この他に避けられない衝突として，Concentrater における下位 Level 通信と上位 Level 通信の時間差衝突[1] が発生する．しかし、これは実際の階層バスでも生じる出線競合による衝突である．

証明 1 各レベルごとの (証明) から *R-Clos* 網は，任意の *Level* において階層型多重バスモデル を エミュレート可能で，そのモデル下では，ノンブロッキングであることが証明できた．

補足として，時間差衝突に対する対処であるが，最終ステージにおけるスイッチングエレメントの，上位階層からのリンクに入力に QUEUE を設ける．これにより粗粒度通信における通信の遅延が読めなくなるが，粗粒度並列処理は MT の割り当てなどをダイナミックに行うため必ずを同期をとらなければならないため，粗粒度通信では，目的地にいずれ到着すればよい．

これに対し，近細粒度の通信は，クラスタ内ではコンパイラによってスタティックに処理されている場合が多いため，通信遅延はきちんと読めなければならない．このため Level-0 通信は最高の優先度が与えられる．

4 *R-Clos* 網のランダム転送シミュレーションの評価

R-Clos 網の基本性能を調べるため，ランダムアクセスによる転送シミュレーションを行った．ランダムアクセスによる転送能力は階層型多重バスモデル下による転送能力と異なるものであるが，ネットワークの基本的な性質や能力を評価するためには重要である．

4.1 シミュレーションの内容

本シミュレーションでは，ネットワークのランダム転送能力を評価するため PE からメモリへアクセスパケットを送り，そのパケットの通過率によって転送能力を評価した．シミュレータは C 言語によりかかれている．パケットには，優先度が通信レベルごとにつけられ、下位通信ほど優先度が高い．これは，近い場所へのアクセスが遠い場所へのアクセスよりも優先されることを示している．このように優先度をつけられたパケットは，衝突すると優先度によって調停がおこなわれる．同じ優先度同士では，ラウンドロビンによって調停される．ただし，Distributer では *R-Clos* 網の性質上，Level-1 以下の通信はどの出力でもルーティング可能であるため，下位通信よりも上位通信が優先される．衝突に負けたパケットは，PE によって再送される．

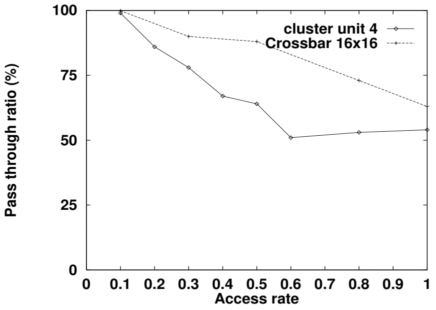
4.2 シミュレーションの結果

図 11(a),(b),(c) には，それぞれ PE 数が，16, 64, 256 の場合のシミュレーションの結果で，それぞれ階層を 1, 2, 3 と重ねたシステムサイズとなる (階層 1 では，ただの *Clos* 網)．縦軸は通過率，横軸はアクセス発行確率を示している．比較のため同 PE 数の場合のクロスバでの評価も載せている．いずれもユニット数が 4 である．

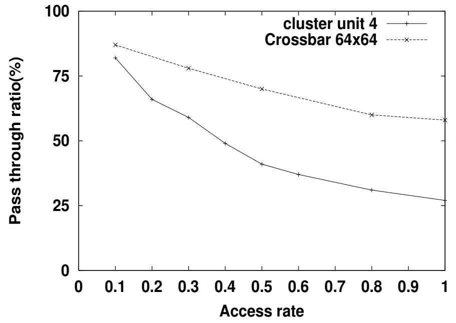
16 PE の場合は，階層が 1 で，これはただの *Clos* 網であることは先に述べた．このときの性能は，*R-Clos* 網が下位通信に対してどの程度のバンド幅を有しているかを示している．図 11(a) から 16 × 16 のクロスバ (ノンブロッキング) に対して 10% ~ 15% 程低い通過率であることがわかるが，クロスバがノンブロッキングネットワークであることを考慮すると，これは高い通過率であるといえる．

---

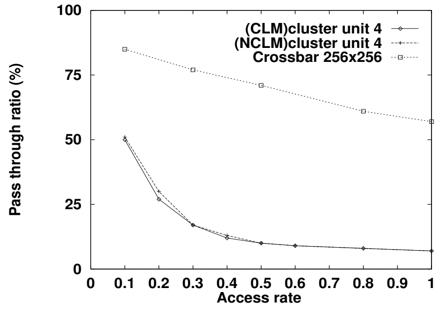
[1] CLM では，この他に rank n($n > 0$) におけるパケットの巡回による衝突が発生する



(a) PE 16(階層 1) のとき



(b) PE 64(階層 2) のとき



(c) PE 256(階層 3) のとき

図 11: アクセス発行確率に対する通過率

64 PE の場合は, ユニット数が 4 の場合は階層が 2 である. 階層が重なったことによって通過率が 16 PE の場合からさらに 10 % ~ 25 % 低下する. この原因は明白で, ランダム転送では, 通信のほとんどが上位通信となるため, アクセスが上位階層へのリンクに集中するためである. したがって, *R-Clos* 網は, 上位階層の転送能力は低いことがわかる.

256 PE の場合は, 階層が 3 となり階層が 2 である場合よりもさらに顕著に階層化による影響があらわれ, ネットワークはすぐに飽和する.

以上から, *R-Clos* 網は, 下位通信に対しては十分なバンド幅が得られるが, 上位通信に対しては十分なバンド幅が得られないことがわかる. しかしながら, 現実的に本シミュレーションのように上位通信の割合が極めて高い場合はまれで, データのローカライゼーションを行なえばこの弱点は補える. したがって, 階層型多重バスモデルのように, 上位階層へのアクセスを制限するモデルに対しては, 効果的な転送能力を保持しているといえる.

5 結論

本稿では, マルチグレイン並列処理に適合するネットワークの条件について考察した. そしてこれらの条件を満たすネットワークとして階層型多重バスモデルを定義した. これによって階層的な通信機構を必要とするマルチグレイン並列処理を効果的にサポートすることが可能となる. さらに実際に多重バスを実装することが困難なことと, MIN がバスよりも転送ポテンシャルの高いことから多段結合網 *R-Clos* でこの階層型多重バスモデルをエミュレートするアプローチをおこなった. そして, 階層型多重バスモデルにおいてノンブロッキングなネットワークの構成が行なえることを証明した.

また, *R-Clos* 網のランダム転送シミュレーションによる評価をおこない, *R-Clos* 網が, 上位階層のネットワークのバンド幅が少なく, 下位階層では十分なバンド幅が得られることを確認した. したがって, *R-Clos* 網は, 階層型多重バスモデルのような上位階層へのアクセスが制限されるモデルにおいては有効に機能することが予測できる.

今後は, マルチグレイン並列処理用マルチプロセッサ ASCA のアーキテクチャを詳細に検討し, R-Clos 網をチップに実装する.

参考文献

[1] 岡本雅巳, 合田憲人, 吉田明正, 笠原博徳, 成田 誠之介 "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 Oct. 1996

[2] Banerjee, U.: *"Loop Parallelization"*. Kluwer Academic Pub. (1994).

[3] 尾形航, 吉田明正, 合田憲人, 岡本雅巳, 笠原博徳: "スタティックスケジューリングを用いたマルチプロセッサシステム上の無同期細粒度並列処理". 情報処理学会論文誌, Vol. 35, No. 4,pp. 522-531 (1994)

[4] Clos,C.: *"A study of non-blocking switching networks"*. Bell system Tech.J. 32, 2 Mar.(1953).pp.406-424.

[5] *"The Organization of the Cedar System"*. Proc. of International Conference on Parallel Processing, vol.1 pp. I49-I56.(1991)

[6] 本多弘樹, 岩田雅彦, 笠原博徳: "*Fortran* プログラム粗粒度タスク間の並列性検出手法". 電子情報通信学会論文誌, Vol. J73-D-I, No. 12, pp. 951-960 (1990).

[7] 本多弘樹, 合田憲人, 岡田雅巳, 笠原博徳: "*Fortran* プログラムの粗粒度タスクの *OSCAR* における並列実行方式" 電子情報通信学会論文誌, Vol. J75-D-I, No. 8, pp. 511-525 (1992).

[8] 岡本雅巳, 合田憲人, 宮沢稔, 本多弘樹, 笠原博徳: "*OSCAR* マルチグレインコンパイラにおける階層型マクロデータフロー処理". 情報処理学会論文誌, Vol. 35, No. 4, pp. 513-521 (1994).

[9] Yoshida, A. and Kasahara, H.: *"Data Localization Using Loop Aligned Decomposition for Macro-Dataflow Processing"*. Proc. of 9th Workshop on Languages and Compilers for Parallel Computers. (1996).

[10] 吉田明正, 前田誠司, 尾形航, 笠原博徳: "*Fortran* マルチグレイン並列処理におけるデータローカライゼーション手法" 電子情報通信学会技術研究報告.CPSY 96-66, Aug. 1996

[11] 笠原博徳: "並列処理技術". コロナ社 (1991).

[12] 岩井啓輔, 藤原崇, 森村知弘, 天野英晴, 木村啓二, 尾形航, 笠原博徳 "マルチグレイン並列処理用マルチプロセッサシステム" 電子情報通信学会. CPSY 97, Aug. 1997