

# 統合型並列化コンパイラ・システム — ネットワーク・シンセシス —

1 G-2

蒲池恒彦 森 眞一郎 村上和彰 福田 晃 富田眞治

(九州大学大学院総合理工学研究科)

## 1. はじめに

現在我々は、統合型並列化コンパイラ開発の一環として、相互結合網およびメモリ構成を可変構造化した『可変構造型並列計算機』<sup>[1]</sup>のためのネットワーク・シンセサイザを開発している。可変構造型並列計算機は、128台のPE (Processing Element) を128×128のクロスバー網で相互結合したマルチプロセッサ・システムである。本クロスバー網は動作モードの1つとして、予め指示された複数のPE間接続パターンを時分割により設定していく機能(プリセット・モード)を有する点に特長がある。

本稿では、まず、可変構造型相互結合網におけるネットワーク・シンセシス問題を定義したあと、そのネットワーク・シンセシス処理について述べる。

## 2. ネットワーク・シンセシス問題

一般に、ある応用問題がマルチプロセッサ上で効率よく実行できるか否かは、その通信形態が相互結合網の topology (幾何学的形状) にどの程度適しているかに強く依存する。そこで、可変構造型並列計算機では相互結合網を可変構造化し、解くべき問題の通信形態をPE間結合形態に直接写像することを可能としている。これにより、種々の応用に対してPE間結合形態を柔軟に適応させる“適応型並列計算機”として動作する。

このとき、解くべき問題の通信形態の相互結合網への反映の方法が問題となる。すなわち、応用問題の通信形態に最も適した相互結合網の topology を如何に構成するかである。これを“ネットワーク・シンセシス問題<sup>[2]</sup>”と呼ぶ。

可変構造型並列計算機(プリセット・モード動作時)における本問題は、次のように定式化される。

- ① 入力: タスク・スケジュールおよび分割プログラム・グラフ。タスク・スケジュールは、静的タスク・スケジューリングにより得られたものである。また、分割プログラム・グラフはPE単位に分割されたプログラム・グラフで、PE間通信(プロセス間メッセージ交換およびリモートメモリ・アクセス)に関する情報を有する。
- ② 出力: ネットワーク制御情報。これは、1個以上のPE間接続パターン、および、複数パターンの場合その時分割間隔とパターン遷移順序から成る。
- ③ 処理: 以下の条件を満足する接続パターン集合、および、接続パターン・シーケンスを生成する。  
(条件a) 送信側PEおよび受信側PEで競合がない。  
(条件b) 定められた接続パターン数を越えてはならない(終了条件x)。  
(条件c) プログラム実行時間を最小とする(終了条件y)。

## 3. ネットワーク・シンセサイザの構成

ネットワーク・シンセサイザは、以下の4つのフェーズから構

成される。なお、②③のフェーズは、終了条件に達するまで繰り返す。

- ① 通信ガント図作成: ネットワーク・シンセシスの前処理にあたり、入力されたタスク・スケジュールと分割プログラム・グラフとから、通信ガント図を作成する。これは、タスク・スケジュールであるガント図に、PE間通信に関する情報(各PEにおける通信先PE, 通信タイミング, 通信時間)を加えたものである。
- ② 接続パターン集合生成: 2章で述べた条件a, bを満足するように、接続パターン集合を生成する。まず、通信ガント図を適切に分割して、PE間の通信形態を表現する複数のPEグラフを作成する。PEグラフのノードはPEを、辺はPE間で通信が行われることを表わす。各々のPEグラフから接続パターン集合を生成し、それらをまとめる。
- ③ 接続パターン・シーケンス生成: 2章で述べた条件cを満足するように、接続パターン・シーケンス(遷移順序)を求める。まず、接続パターン・シーケンスは、PE間通信のスケジューリング、および、PE間通信と接続パターンの対応付けを同時に行うことで得る。そして、当該シーケンスにおけるプログラムの実行時間を算出し、条件cを満たしたかどうかを判定する。
- ④ ネットワーク制御情報生成: ネットワーク・シンセシスの後処理にあたり、出力となるネットワーク制御情報を生成する。ネットワーク制御情報の詳細については、参考文献[2]を参照されたい。

上記②③のフェーズが、ネットワーク・シンセシスの本処理にあたる。

## 4. 接続パターン集合生成

### 4.1 PEグラフ作成

与えられた通信ガント図を時間軸に関して、2等分に分割する。このとき、あるPE間通信が分割境界にまたがる可能性がある。これについては、

- ① 当該PE間通信をそのまま前あるいは後の通信ガント図に含める。
- ② 当該PE間通信を2分割して、両方の通信ガント図に含める。

のいずれかの方法で、通信ガント図を修正する。分割した各々の通信ガント図から、PEグラフを以下の手順で作成する。

- ① 送信側PEを左側に位置するノード、受信側PEを右側に位置するノードとする。
- ② PE間通信を行う送信側PEと受信側PEとを無向辺で結ぶ。得られるPEグラフは、2部グラフとなる。

## 4.2 PE グラフ辺彩色

1個のPEグラフから、接続パターン（厳密には“接続パターンの候補”）の集合を生成する。このとき、2章で述べた条件aを満たす必要がある。すなわち、1つの接続パターンにおいては、送信側PEおよび受信側PEで競合があってはならない。これを保証するため、PEグラフの辺彩色を行う。

グラフの辺彩色とは、隣接する辺が必ず異なる色となるようにグラフの辺を彩色することである。2部グラフの辺彩色問題に関しては、グラフの最大次数をDとしたとき、D色で彩色する最適アルゴリズムがいくつか提案されている<sup>[9]</sup>

PEグラフ辺彩色の結果、1つの色で塗られた辺の集合が1つの接続パターンとなる。よって、最大次数DのPEグラフに対しては、D個の接続パターンが生成される。

## 4.3 接続パターン数の制限

個々のPEグラフの辺彩色で得た接続パターンの総数を求める。この総数は、2章で述べた条件bを満たす必要がある。すなわち、定められた接続パターン数（現仕様では1024）を越えてはいけぬ。これは、同時にネットワーク・シンセシス処理の終了条件x（5.3節参照）でもある（このとき、1つ前のイタレーションで生成した接続パターン集合が出力となる）。

ただし、4.2節で得た接続パターンはあくまでも“接続パターン候補”であり、以下の手法により接続パターンの数を減らすことが可能である。

- ① 2個以上の接続パターンが条件aを満たす場合、1個の接続パターンにマージする。
- ② あるいは、PEグラフの最大次数Dが減るように、辺の削除を行う（ただし、削除された辺に対応するPE間通信は直接通信でなく、ルーティングによる間接通信となる）<sup>[4]</sup>

## 5. 接続パターン・シーケンス生成

### 5.1 シーケンス生成

接続パターン・シーケンスは、PE間通信のスケジューリング、および、PE間通信と接続パターンの対応付けを同時に行いながら生成する。これには、PE間通信の先行制約を示すDAG（CDDAG：Communication Dependency Directed Acyclic Graph）を用いる。

CDDAGは、4.1節で述べた分割通信ガント図に含まれるPE間通信に関するもので、個々のPE間通信をノードとする。各ノードには、次の情報が付加される。

- ・CST：通信開始タイミング
- ・CFT：通信終了タイミング
- ・CQ：通信時間量（=CFT-CST）
- ・CR：後続のPE間通信に与える影響度  
（=自ノードのCQ+ $\sum$ {後続ノードのCQ}）

1個のCDDAGに関する接続パターン・シーケンスは、以下の手順をCDDAGが空になるまで繰り返すことで生成する。

- (1) 与えられたCDDAGにおいて、先行ノードを持たないものをすべて抽出し、候補ノード集合に加える。
- (2) 候補ノード集合の中で、以下の条件を満たす1次候補ノードを選択して選択ノード集合に入れる。
  - Ⓐ 最も早い通信開始タイミング（CST）を持つ。
  - Ⓑ CSTの同一のものが複数存在する場合は、通信時間量（CQ）が最も大きい。
  - Ⓒ CQの同一のものが複数存在する場合は、後続ノードに与える影響（CR）が最も大きい。
- (3) 1次候補ノードのPE間接続を満たす接続パターンを候補パターン集合に入れる。
- (4) 候補ノード集合が空になるか、あるいは、候補パターン集合の中の接続パターンが1個になるまで、以下の処理を繰り返す。

- ① 候補ノード集合の中から、以下の条件を満たすノードで、最も小さいCSTを持つノードを選択する。複数存在する場合は、後続ノードに与える影響（CR）が最も大きいノードを選択する。これを2次候補ノードと呼ぶ。

$$CST_p \leq CST, \text{ かつ } CFT \leq CFT_p$$

ここで、 $CST_p$ と $CFT_p$ は、1次候補ノードのCSTとCFTである。

- ② 候補パターン集合の中に、2次候補ノードのPE間接続を満たす接続パターンがあるか否か調べる。存在すれば、2次候補ノードを選択ノード集合に加え、候補パターン集合から当該接続パターン以外の接続パターンを削除する。存在しなければ、2次候補ノードを捨てる。

- (5) 選択ノード集合の中のノードを、CDDAGから除く。次に、残っているノードの中で、 $CST_q < CFT_p$ であるノードを抽出し、当該ノードのCSTおよびCFTを次のように変更する。

$$CST = CFT_p, CFT = CFT + (CFT_p - CST_q)$$

また、当該ノードのすべての後続ノードのCSTおよびCFTを次のように変更する。

$$CST = CST + \max \{ \text{すべての } (CFT_p - CST_q) \},$$

$$CFT = CFT + \max \{ \text{すべての } (CFT_p - CST_q) \}$$

## 5.2 実行時間の算出

すべてのCDDAGに関する接続パターン・シーケンスを生成した後、プログラムの実行時間を求める。これは、各CDDAGのシーケンスにおける最終接続パターンを割り付けた1次候補ノードの $CFT_p$ の総和により近似する。

そして、前回のイタレーションで得られた実行時間に対する、当該実行時間の向上率を計算する。これが、定められたしきい値以上であれば、再びネットワーク・シンセシス処理を繰り返す。そうでない場合（後述の終了条件y）は処理を打ち切り、前回と今回のいずれか良い方を選ぶ。

## 5.3 終了判定

ネットワーク・シンセシス処理は、以下の終了条件のいずれかが成立するまで繰り返される。

- （終了条件x）生成した接続パターン数が定められた接続パターン数を越えた。
- （終了条件y）プログラム実行時間の向上率が、定められたしきい値以下であった。
- （終了条件z）定められたイタレーション回数を越えた。

## 6. おわりに

以上、『可変構造型並列計算機』を1対象としたネットワーク・シンセシス処理について述べた。今後は、アルゴリズムの細部の詳細化、および、評価を行う予定である。

## 参考文献

- [1] K.Murakami et al.: The Kyushu University Reconfigurable Parallel Processor—Design of Memory and Intercommunication Architectures—, Proc. 1989 Int'l. Conf. Supercomputing, pp.351-360, June 1989.
- [2] 蒲池ほか：可変構造型並列計算機のネットワーク制御方式、信学技法、CPSY89-16（1989年8月）。
- [3] H.N.Gabow and O.Kariv: Algorithms for Edge Coloring Bipartite Graphs and Multigraphs, SIAM J. Comput., Vol.11, No.1, pp.117-129, February 1982.
- [4] I.Lee and D.Smitley: A Synthesis Algorithm for Reconfigurable Interconnection Networks, IEEE Trans. on Comput., Vol.C-37, No.6, pp.691-699, June 1988.