全対全通信の応用

2X-6

武 理一郎, 野口 泰生
(株) 富士通研究所

1．はじめに

並列処理に於いては，その対象問題に応じて様々な通信パターンが現われる．境界値問題をシミュレートする時にはメッシュパターンが現われ，FFTやバイトニックソートを処理する時にはバタフライパターンが現われる．又，ハッシュジョインを行う時には全対全通信パターンが現われ，パイプラインソートを行う時には点対点のストリーム通信が要求される．

ある並列計算機がどんな問題を処理するために使えるかは，その並列計算機の備えている結合網の性格によって強く影響される．安く簡単な結合網は，コスト／パフォーマンスが良いが適用範囲が狭い．一方，どんな問題も処理できるという場合には，結合網が贅沢過ぎてコストが上がり，誰も買わないマシンとなるかも知れない．

我々は，問題が含む通信パターンをそのままの形でなく，別の通信パターンに変換して処理することを考えている．様々なパターンが1種類のパターンに変換できるものならば，その1種類のパターンを処理する結合網を作ればよい．その様なネットワークは安く高速に作れるのではないだろうか．

ここでは，変換先の通信パターンとして全対全通信を考える．全対全通信は大変重い通信パターンであるが，既に報告した通り超立方体のトポロジーの上で輻輳なく処理することができる［1］．以下では，点対点のストリーム通信，2次元のメッシュ及びトーラスパターンの通信を全対全通信パターンに化かす方法について極く簡単に説明する．

2．全対全通信

まず通信パターンの化かし先である全対全通信について説明する．ここで全対全通信と言っているのは，下図の様に各点が全点と通信するものである．（ブロードキャストとは異なる．）

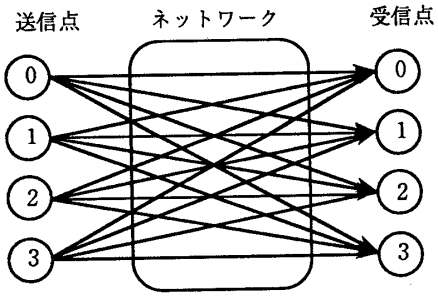

図1 全対全通信

各点が送信するメッセージの数はnであり（ここでnは点の数），従って各点でのメッセージ処理が逐次的に行われるとすると，1回の全対全通信が処理されるためにかかる時間は少なくともO(n)である．この最適時間は，log n段の多段結合網を同期的に制御することで達成することができる．を示す．この様な結合網としては，我々が［1］で提案したdragon netやマルチポートページメモリのポートとメモリバンクの間を結ぶ結合網がある［2］．

dragon netは，全対全通信を時分割で処理する．フェーズsの時，点vは点v⊕sと結合される．フェーズが0からn－1までをスイープすると各点は自分を含む全ての点と1回づつ結合され，全対全通信が処理される．この間輻輳が起きないため，最適時間で全対全通信を処理することができる．但し，各点からのメッセージの送信順は予めスケジュールされてしまい，各点の自由にはならない．

3．点対点通信

全対全通信とは対称的に局所性の高い点対点通信について考える．各ペア間での通信量はある程度大きいと仮定する．この様な結合網はBenes網などの置換網として長く研究されて来た［3］．ここでは，点対点通信を全対全通信に分解しreroutingの要らない置換網を実現する方法を述べる．

あるペア間の通信は下図の様に，全対全通信を2回使って処理される．この2回の全対全通信の間をダブルバッファリングで繋ぐことで，切れ目のないストリーム通信が可能となる．全対全通信が輻輳を起こさないため，複数のストリーム間で輻輳を起こすことがない．従って，reroutingをせずに置換網を実現することができる．

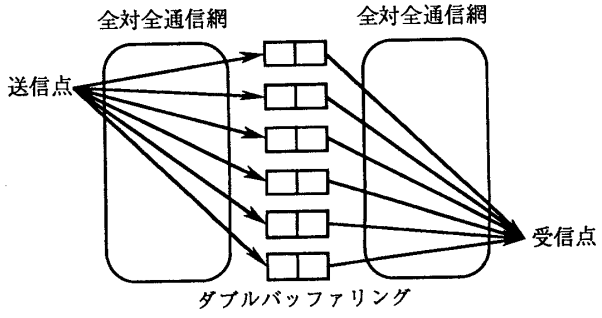

図2 点対点通信の全対全通信への分解

全対全通信を実現するのにdragon netを用いた場合を下図に示す．到着点に於いてデータの到着順が乱れていることに注意されたい．この乱れを直すために到着点に於いてもバッファリングが必要である．従って，各ストリームは，到着までに2つのバッファリングを経ることになる．これによる遅延が，reroutingが不要であることのコストである．

Application of All-to-all Interprocessor Communication

Riichirou Take, Yasuo Noguchi
Fujitsu Laboratories Ltd.

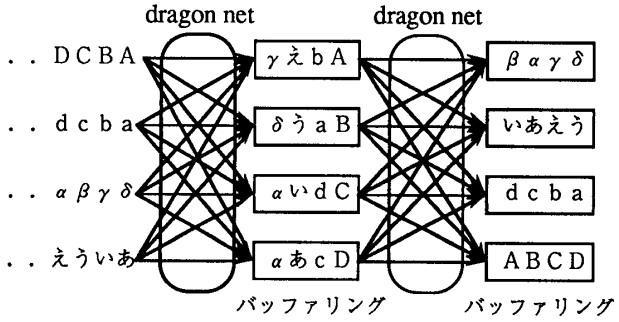

図3 置換網の dragon net による実現

4．メッシュ及びトーラス

メッシュやトーラスパターンを全対全通信を使って処理する方法を説明する．この方法はメッシュとトーラスでほとんど同じなので，ここでは主にトーラスについて述べる．この方法は，n × n のメッシュやトーラスを n 台のプロセッサで処理するものである．従って，問題の含む並列性を完全に抽出している訳ではない．

下図に 8 × 8 のトーラスを示す．各点に割り振られた番号は，その点の処理を行うプロセッサの番号である．ここではプロセッサは 0 から 7 までの番号を持つとしている．

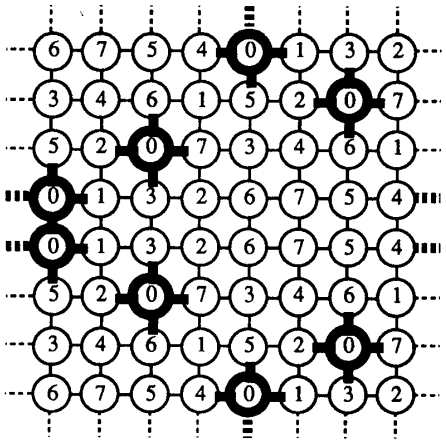

図4 トーラスへのプロセッサ番号の割り付け

この割り付けは図5の様なやり方で行われている．1つの×が割り付けの単位である．1つの×には，2つの数字を交互に並べてある．この様にすると，プロセッサ間で全対全通信を引き起こすことができる．例えば，プロセッサ 0 に割り当てられた点（上図中で強調してある部分）の周囲にある点（プロセッサ 0 が通信を行う相手）は，プロセッサ 0 から 7 までに均等に割り当てられている．

例えばプロセッサ 0 に於ける処理は次の様に行われる．まず，周囲に 0,1,4,5 のある 4 点の処理を行い，プロセッサ 0,1,4,5 に送信するデータを用意する．次に，それらのデータを相手プロセッサが用意していた自分宛のデータと交換しつつ，周囲に 2,3,6,7 のある 4 点の処理を行い，プロセッサ 2,3,6,7 に送信するデータを用意する．次に，先程交換しておいたデータを用いて周囲に 0,1,4,5 のある 4 点の処理を行いつつ，プロセッサ 2,3,6,7 とのデータ交換を行う．ここでは，トーラスへのプロセッサの割り当てと，dragon net のフェーズのシーケンスに反転グレイ符号を用いている．これにより，プロセッサ内処理と通信処理とをオーバーラップさせて進めることができる．

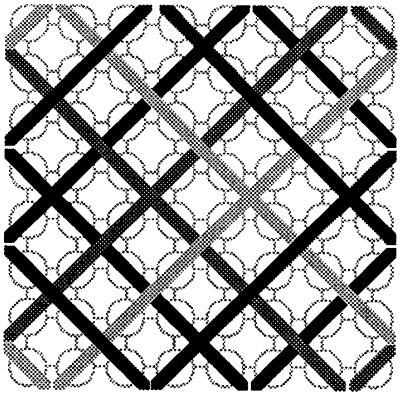

図5 トーラスへのプロセッサ番号割り付け方式

メッシュを処理する場合には，メッシュの各点に下図の様にしてプロセッサ番号を割り振ればよい．ここで割り付けの単位はループであり，トーラスの場合と同様，各ループに2つの数を交互に並べる．

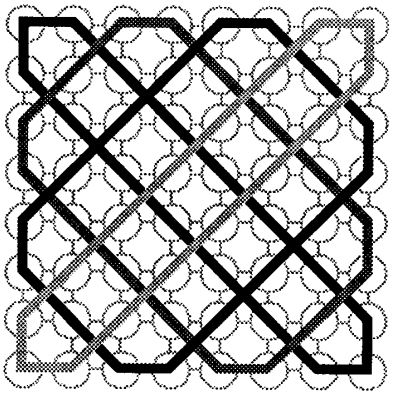

図6 メッシュへのプロセッサ番号割り付け方式

5．まとめ

ストリーム通信（置換網）とメッシュ／トーラスパターン通信を全対全通信に変換して処理することができることを示した．全対全通信は dragon net などの結合網により効率良く処理することができる．従って，例えば dragon net を備えた並列計算機は，置換網或いはトーラス形ネットワークを備えた並列計算機よりも守備範囲が広いことが期待される．

参考文献

［1］武，超立方体形ネットワークに於ける全点対全点通信の最適ルーティング法，情報処理学会第35回全国大会，1987

［2］田中，A Multi Page-Memory Architecture and A Multiport Disk-Chache System，New Generation Computing，1984

［3］Lev et al.，A Fast Parallel Algorithm for Routing in Permutation Networks，IEEE Trans. Comput.，Vol. C-30，No.2，1981