広域情報を用いたルーティングアルゴリズムの汎用シミュレータへの実装

森 裕貴 † 横田 隆史 † 大津 金光 † 馬場 敬信 †
† 宇都宮大学工学部情報工学科

1 はじめに

近年では，数千，数万ノードの超並列計算機が登場しており，一般に普及している計算機でも複数のコアを搭載しているものもあることから，計算ノードを増やして性能向上を目指す傾向が見られる．特に大規模な並列計算機では，ノード間の通信性能がシステム全体の性能に影響を及ぼすため，通信性能の向上は大変重要な課題となっている．

このような計算機での情報伝達で用いられる手法として，経路が固定された非適応型ルーティングや，ネットワークの状況に応じて経路を選択する適応型ルーティングが存在する．しかし，適応型ルーティングであっても，その多くは接続されているリンクの状態などの局所的な情報しか用いないため，広範囲に渡る混雑を検出して対処することはできない．

我々はこの問題に対して，仮想チャネルの ready/busy 情報を輻輳情報として扱い，相互結合網内に伝搬させることで各ノードで混雑状況を把握し，適切な経路選択を行うルーティングアルゴリズム Cross-Line を提案した．これは，広範囲の輻輳情報を用いることでより適切なルーティングを行い，スループットの向上とレイテンシの低減を目指した．

また，Cross-Line の評価を行うためにネットワークシミュレータ “Chimera” を開発したが，これは Cross-Line の評価を目的として開発されたため，新たなアルゴリズムの実装には向いていない．そこで本論文では，この Cross-Line を汎用のネットワークシミュレータに実装することで，どの程度の機能追加で，どれほどの性能向上を見込めるかを検討し，さらに，発展型のアルゴリズムを開発できる環境を用意する．

2 広域情報を用いたルーティングアルゴリズム Cross-Line

このルーティングアルゴリズムは，処理を行うルータと同次元軸上に存在するルータの VC の ready/busy 情報 (VCinfo) を収集して各 VC が持つ InfoBuffer へ格納し，その情報を基により空いている方向へデータを転送する適応型ルーティングアルゴリズムである．送信方向の選択は以下のように行う．

1. まず InfoBuffer の LSB (自ノードに近い情報) から順に連続する送信可能ノード (Available) 数を比較し，これの長い方向へ送信を行う．
2. 連続する Available 数が等しかった場合は，さらに連続する送信不可能ノード (Not-Available) 数を比較し，これの短い方向へ送信を行う．
3. 送信方向が決定しなかった場合は 1，2 の手順を繰り返す．
4. それでも送信方向が決まらなければ，より多くの VCinfo を保持する方向へ送信する．
5. 保持する VCinfo 数も等しい場合は，次元の若い方向へ送信する．

また，デッドロックへの対処として，送信方向による VC の分割とターンの制限を行う．トーラストポロジにおいてはさらにラップアラウンドごとに VC 番号を増加させることによりパケットの循環を防ぐ．

3 相互結合網シミュレータ BookSim

これは文献 [1] にて利用される汎用ネットワークシミュレータで，ネットワークサイズや次元数，ルータ内のクロスバスイッチのアロケータ等が自由に変更可能なものである．さらに，モジュールごとに分けて記述してあるため，ルーティングアルゴリズムや通信パターン等が容易に追加できる構造となっている．

Cross-Line シミュレータとしてすでに Chimera[2] が存在するが，これは Cross-Line の性能評価用であるため，ルータ機能の構成の自由度はあまり大きいものではなく，機能追加も容易ではない．そこで本論文では，BookSim に Cross-Line ルーティングを実装することにより，追加で必要となるハードウェアや性能の評価を行い，さらに VCinfo を用いたルーティングの追加を容易にできるような環境を構築する．

4 シミュレータの Cross-Line への対応

BookSim はノード間の広域情報 (VCinfo) の受渡しが実装されていないため，まずこの機能を追加する必要がある．しかし，この実装はルーティングアルゴリズムや通信パターン等の代替モジュールの追加とは異なり，新規の機能を追加することとなるため，ルータ自体の動作を書き換えることとなる．ここでは，実装を容易にするためにできるだけルータが持っている機能を用いて VCinfo の伝搬処理を実装していく．

4.1 ルータの基本動作

ルータの一連の動作を簡単に示す．

1. 各チャネルのリンクにフリットがあれば，それをチャネルの入力バッファに格納する．
2. 各チャネルの入力バッファの先頭にパケットの先頭フリットが格納されていれば，それに格納されている情報を基にルーティングを行う．
3. ルーティングの結果から，送信を行う VC の選択やクロスバスイッチの確保を行う．
4. 各入力バッファに格納されているフリットは，クロスバにより指定された方向へ送信される．

ルータは 1 サイクルごとにこの動作を行い，パケットを目的のノードまで運搬する．

4.2 広域情報の受渡し

まず，VCinfo の送信方法を考える．専用線を用いればデータパケットには非干渉で輻輳情報の伝搬を行えるが，データパケット送信とは独立した動作になることから，1 つのルータで 2 つのネットワークの動作を行わせる規模となることが予想される．そこで，ここでは VCinfo をデータパケットと同じよう送ることで，低コストでの実装を目指す．

VCinfo をパケットとして送るために，まずこのパケットを生成する領域を用意する．VCinfo はあくま

Implementation of a Routing Algorithm that uses Global Information on a General-purpose Simulator

†Hiroki Mori, Takashi Yokota, Kanemitsu Ootsu and Takanobu Baba
Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

でデータパケットのルーティングの目安に用いられるものであるため，極力データパケットの送信を妨げないのが望ましい．もし，データパケットが格納される入力バッファを生成領域として用いるとすると，このバッファはキュー構造であるため，既にデータパケットが存在するとVCinfoの後に格納されたデータパケットはVCinfoの転送にかかる時間分遅れてしまう．そこで，VCinfoパケット用にさらにVCを1つ増やし，これの入力バッファをVCinfo生成領域として用いることによりこの問題を解決する．

次にVCinfoパケットの生成タイミングを考える．VCinfoパケットは，InfoBufferの更新があったときに，この更新を隣接したノードへ伝えるために生成される．この更新は，隣接ノードに接続されているVC状態の変化を検知したときと，隣接ノードからのVCinfoの転送により発生する．データパケットのルーティング後はデータパケットによるチャネルのリクエストが決まるので，ここでリクエスト情報を基にすぐに利用できるチャネルを判別し，そのチャネルに対応したVCinfoパケットを生成し，チャネルのリクエストを行う．この動作により，最新のVCinfoをデータパケットのブロッキング無しに行うことができる．

また，最大でネットワークの1辺分のVCinfoを送受信するが，フリット幅によっては1フリットで全ての情報を送信できないことが考えられる．その場合VCinfoを分割して送信することになるが，VCinfoパケットを2フリット以上の長さとしてしまうとVCinfoパケットの送信が1クロックで終わらず，その間にデータパケットのリクエストが発生してしまうとデータパケットのブロッキングとなってしまう．そこで，VCinfoパケットを複数の1クロックで送れる小さいパケットに分割する．これにより，2パケット目以降のVCinfoパケットはデータパケットによるブロッキングで情報が古くなってしまう可能性があるが，データパケットのブロッキングを回避することができる．

最後にVCinfoパケットの受信処理である．VCinfoの目的地はノード番号でなく送信方向によって決められるためVCinfoパケットは目的地の情報を持たない．また，このパケットを受信したノードはVCinfoが持つ情報を，受信したVCに対して適用する必要がある．そこで，VCinfoパケットにVCinfoパケットであることを示すフラグを設け，手順1の段階でパケットを振り分け，VCが持つInfoBuferを更新する．

以上を踏まえ，広域情報を扱うルータの動作を示す．

1. 各チャネルのリンクにVCinfoパケットがあれば対応したVCへ適用し，データパケットがあればチャネルの入力バッファに格納する．
2. 各チャネルの入力バッファの先頭にパケットの先頭フリットが格納されていれば，それに格納されている情報を基にルーティングを行う．
3. ルーティングの結果から，利用可能なリンクを用いて送信するVCinfoパケットを生成する．
4. データパケット，VCinfoパケットのリクエストを基に，送信を行うVCの選択やクロスバスイッチの確保を行う．
5. 各入力バッファに格納されているフリットは，クロスバにより指定された方向へ送信される．

追加するハードウェアとしては，広域情報を保存するInfoBuffer領域を各VCに，VCinfoを転送する際に用いるVCを各送信方向につき1つ用意すればよい．

表1: ネットワークパラメータ

トポロジ	2次元メッシュ，トーラス
フロー制御	Virtual Cut-through
ルーティング アルゴリズム	Dimension-Order Cross-Line
VC数	Dimension-Order (メッシュ): 1 Dimension-Order (トーラス): 2 Cross-Line (メッシュ): 2(+1) Cross-Line (トーラス): 6(+1)
ネットワークサイズ	32×32 (1024ノード)
フリットサイズ	32 (フリット)
パケット入力量	0.016~0.320(フリット/クロック)
通信パターン	ユニフォーム通信

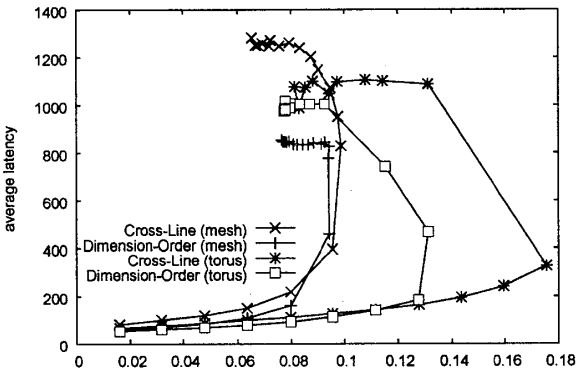

図1: Cross-LineとDimension-Orderの比較

5 評価

ここでは，実装したCross-Lineの評価を行う．比較対象として，最短経路ルーティングとして有名なDimension-Orderを選択した．これは次元の若い順にパケットを転送するもので，リンクの状態に依らず固定的に経路が選択されるものである．

図1に表1でのネットワークの動作結果を示す．メッシュ，トーラストポロジの両方においてDimension-Orderよりも高いスループットが確認できる．また，パケット入力量が多い点においてはDimension-Orderの方がレイテンシが小さくなっているように見えるが，これはネットワークの混雑によりパケット入力がそれ以上できなくなり頭打ちのようになったと考えられる．

6 おわりに

この実装により，Cross-Lineルーティングの機能の多くは既存のルータ機能を転用することで実装可能であることが確認でき，実装による性能向上も確認できた．BookSimでのCross-Lineの環境構築を行ったことで，今後容易に機能追加を行っていくことができる．

今後はこのシミュレータを用いてCross-Lineの改良や，VCinfoを用いた新たなルーティングアルゴリズムの提案等を行っていきたい．

謝辞 本研究は，一部日本学術振興会科学研究費補助金（基盤研究(B)18300014，同(C)19500037，若手研究(B)17700047）および宇都宮大学重点推進研究プロジェクトの援助による．

参考文献

[1] William James Dally, Brain Towels, "Principles and Practices of Interconnection Networks," Morgan KaufmannPublishers, 2004.

[2] 横田隆史，西谷雅史，大津金光，古川文人，馬場敬信，"大域的な情報を用いる相互結合網方式 Cross-Line"，情報処理学会論文誌コンピューティングシステム，Vol.46, No.SIG 16 (ACS-12), pp.28-42, 2005年12月．