細粒度並列処理向け相互結合網 HTN の適応ルーティング

†的山　和也，†三浦　康之
湘南工科大学　工学部情報工学科

● 1. はじめに

近年 LSI の大集積化、三次元 LSI の登場により、LSI 上に多数の PE を配置し、PE 間をネットワーク結合する技術が注目され、さまざまな三次元 LSI 向けの相互結合網が提案されている[3]。これらのルーティング アルゴリズムとして、一般的には固定ルーティングが用いられているが 固定ルーティング は適応ルーティングと比べて、チャネルの 利用率が劣り、大 きなデータ通信の際に遅延を起こしやすい 。我々は、低遅延な適応ルーティングの実装法を研究中である。 更なる適応ルーティングアルゴリズム の充実により、HTN の大幅な性能向上が見込める。

本研究では、細粒度並列処理向けの相互結合網 HTN の適応ルーティングを高 性能なシミュレーションプログラムを用いて、性能評価、検証を行う。これまでに HTN の固定ルーティングの実装と評価が完了したので、本稿ではそれらの結果を示すと同時に適応ルーティングアルゴリズムの提案を行う。

● 2. HTN(Hierarchical Torus Network)

2.1 構造

HTN[1]は基本モジュール(Basic Module: BM)が(m×m×m)の 三次元トーラス網の構造をしており、BM を複数個用いて(n×n)の二次元トーラスと相互結合させた構造をしている。 本研究では m=n=4 としている。HTN を作成するにあたっては、上位レベルのネットワークとして BM の一番上のレベルに位置する PE のみを使用する。実際の HTN の構造を図 1 に示す。

2.2 HTN の固定ルーティング

HTN の固定ルーティングアルゴリズムを図 2、図 3、図 4 に示す。

固定ルーティング は基本的に最上位レベルから最下位レベルまで、レベル順に転送が行われる。HTN の固定ルーティングでは、始めに上位レベルの二次元トーラスのルーティングを行って目的の PE のある BM まで転送を行い、その後 BM 内の転送が行う。

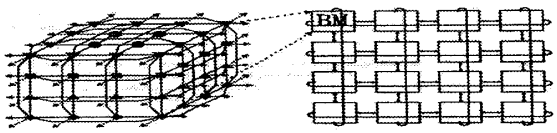

図 1 HTN 構造

```
BM_ROUTING(src, dest){

if(src-z != dest-z){
  tag = (dest-z - src-z);
  if(((tag <= -2)||((tag > 0)&&(tag <= 2)))
        return z+;
  else
        return z-;
}
else if( src-x != dest-x){
  tag =(dest-x - src-x);
  if(((tag <= -2)||((tag > 0)&&(tag <= 2)))
        return x-;
  else
        return x+;
}
else if(src-y != dest-y){
  tag = (dest-y - src-y);
  if((tag <= -2)||((tag > 0)&&(tag <= 2)))
        return y-;
  else
        return y+;
}
```

図 2 BM ルーティング

```
nextroute(src,dest){

if(src == dest)
   return 0;
if(src-z != dest-z){
    tag = (dest-z1 - src-z1);
    if(((tag < -2)||((tag > 0)&&(tag < 2))
        ||((tag == 2)&&((src-x1 >= 2)
        ||((tag == -2)&&(src-x1 >= 2)){
        if(src-z1 >= 12)&&(src-z1 <= 15){
            return z1+;          }
        else{
            BM_MESHROUTING(src, (src % 4) + 12 );   }
 else if(src-z1 <= 3){
            return z1-;    }
        else{
            BM_MESHROUTING(src,src % 4);     }     }
else if(src-x1 != dest-x1){
    tag = (dest-x1 - src-x1);
    if(((tag < -2)||((tag > 0)&&(tag < 2))
        ||((tag == 2)&&((src-x1 >= 2)
        ||((tag == -2)&&(src-x1 >= 2)){
        if((src-x1 == 3)||(src-x1 == 7)
            ||(src-x1 == 11)||(src-x1 == 15)){
              return x1-;                 }
        else{
            BM_MESHROUTING(src-x1);      }     }
  else if((src-x1 == 0)||(src-x1 == 4)
        ||(src-x1 == 8)||(src-x1 == 12)){
          return x1+;                 }
        else{
            BM_MESHROUTING(src-x1);   }     }
    else BM_ROUTING(src,dest);
}
```

図 3 上位レベルでのルーティング

```
BM_MESHROUTING(src,dest){

 if( (src-z != dest-z){
 tag = (dest-z - src-z);
 if(tag > 0)
        return z+;
    else
        return z-;                    }
 else if(src-x != dest-x){
 tag =(dest-x - src-x);
 if(tag > 0)
        return x+;
    else
        return x+;
 else if(src-y != dest-y){
 tag = (dest-y - src-y);
 if(tag < 0)
        return y-;
    else
        return y+;              }   }
```

図 4 BM MESH ルーティング

図 2 は BM 内での転送を行うルーティングである。src-z、src-x、src-y はそれぞれ現在地の PE 番号を示す。dest-z、dest-x、dest-y は行き先の PE 番号を示す。'z+'と書かれたものは z 軸の上方向に進むと解釈する。'z-'と書かれたものは z 軸の下方向に進むと解釈する。他も同様である 。

Adaptive Routing of the Interconnection Network HTN for Finegraded Parallel Processing
†Kazuya Matoyama,Yasuyuki †Miura-Shonan Institute of Technology

図 3 では上位レベルのネットワークである二次元トーラスでのルーティングである。 src-z1、src-x1 はそれぞれ上位レベルの二次元トーラスの現在地の PE 番号を示す。dest-z1、dest-x1 は行き先の PE 番号を示す。図 4 は上位レベルでの移動を終えて、目的 BM にたどりついた後に行うルーティングである。
このように BM 内だけでの通信と上位レベルでの移動を行う時の２つに分けて行っている。

● 3. 実験環境

3.1. シミュレーションプログラムの構成

本研究で用いた PE の構成はノードプロセッサ、送信用ネットワークインターフェイス、受信用ネットワークインター フェイス、そしてルータとなっている。ルータの中身は FIFO、制御回路、デマルチプレクサ、マルチプレクサ、クロスバスイッチとなっている。 これら全部を１つの PE として扱う。図５は PE の構成図となっている。

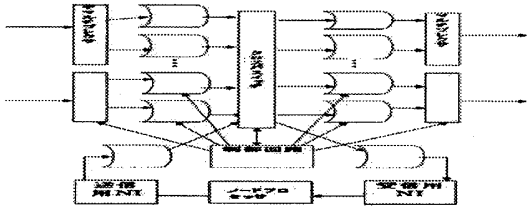

図５ PE の構成

3.2. シミュレーションプログラム環境設定

本プログラムでの PE 数は 1024 となっている。HTN では BM(4×4×4)が 16 個並んでいる。MESH では BM(32×32)となっている。

● 4. 実行結果

本実験ではパケットの送信はランダムで行っている。HTN の固定ルーティングでシミュレーションした結果を図６に示す。

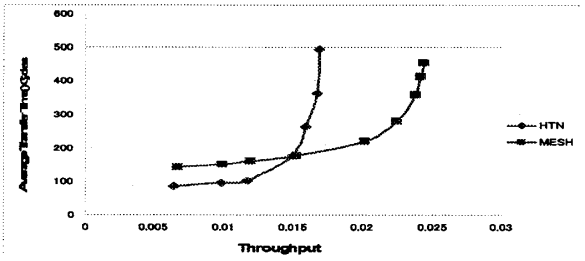

図６ 実行結果グラフ

図６の結果より MESH の固定ルーティングと HTN の固定ルーティングの最大スループットが MESH の方が約 1.4 倍の結果となっている。HTN のような階層型相互結合網では、上位レベルリンクが混み合うことから、上位レベルリンクに対して適応ルーティングを用いることにより、通信性能の大幅な向上が期待できる。

● 5. HTN の適応ルーティング

5.1 Turn モデル

デッドロックは結合網内のバッファが論理的に循環構造を作るため起こる。
Turn モデル[2]ではパケットがルーティング中に方向を変えるパターンに制限を加え、循環させないようにするものである。 このモデルは論理的な循環構造に着目し、結合網に依存しないのが特徴である。 これにより、故障地点や混雑地点を迂回する適応ルーティングが可能となる。

5.1.1 NF(North Fast)法

本研究では HTN の適応ルーティングとして NF 法の HTN への実装を検討中である。HTN でもっとも混雑が予想されるのが上位レベルの転送部分である。そこで、上位レベル転送について NF 法の実装を考える。図 7 に NF 法の Turn モデルを示す。NF 法では図中の上に行く動作に制 限を加えることでデッドロックを回避する手法であ る。NF 法の動作は目的 PE が、自分よりも下か同じ高さにいる時経路を自由に選べ、目的 PE が自分よりも上にいる時は固定ルーティングを行う。
現在 NF 法は BM 内リンクのうち上位レベル転送に使う部分についてのみ実装を始めている。そのため、実際に NF 法を適用しているのは上位レベルでの MESH 網だけである。今後上位レベル部分の実装を終えた後は下位レベルに対しての実装も考えている。

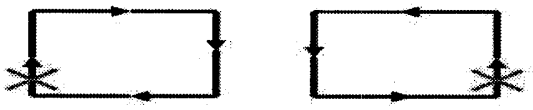

図７ Turn モデル NF 法

● 6. 今後の予定

今回の実験により、HTN の固定ルーティングの実装と評価は完了した。今 後、NF 法用いたルーティングを作成し、動作確認後、性能評価および検証を行う。また、他の適応ルーティングアルゴリズムの実装を考えており、三次元 LSI 向けの階層型相互結合網 DDR(Dynamic Dimension Reversal)法[3]の性能評価、検証を行う予定である。

● 参考文献

[1] M.M. Hafizur Rahman,
Hierarchical Interconnection Networks for Massively Parallel Computers
[2] G.J.Glass and L.M.Ni, “Maximally Fully Adaptive Routing in 2D Meshes,” ISCA92. pp. 278-287.
[3]三浦康之, 堀口進, 福士将, 細粒度並列処理向け相互結合網 TESH における適応型ルーティングアルゴリズム