4ZB-2

通信粒度予測機構を実装したソフトウェア分散共有メモリ

坂口朋也†，鈴木祥†，多忠行‡，吉瀬謙二‡，弓場敏嗣‡
電気通信大学情報工学科†　電気通信大学 大学院情報システム学研究科‡

1　はじめに

従来 MPI 等のメッセージ通信ライブラリを用いた並列プログラミングでは，プログラマの明示的なデータ転送の記述により，データの授受が行われる．データの転送をプログラマが考慮しなくてはならず，並列プログラミングを困難とする要因となっている．

ソフトウェア分散共有メモリ (S-DSM) は，分散メモリ環境においてソフトウェアにより仮想的な共有メモリを構築する．S-DSM はプログラマに仮想的な共有メモリを提供するため，プログラマはデータの転送を考慮せずにすむ．そのため，共有メモリモデルに基づく並列プログラミングを用いて，分散メモリ環境の並列処理を行うことができる．

S-DSM は分散メモリ間でデータの一貫性を保証することにより，仮想的な共有メモリを構築している．一貫性保証のためには分散メモリ間のデータ授受を頻繁に行う．そのため通信回数の増大により，期待した性能が得られない場合がある．また，ノード数の増加に伴い通信回数は増加するため，性能向上が得られにくくなる傾向がある．そのため，通信回数を削減することができれば，速度向上が期待できる．

本稿では，S-DSM にデータ転送回数を削減する機構を組み込むことで速度向上を得た．

2　通信粒度予測機構の提案

2.1　通信粒度予測

本研究では，S-DSM の 1 つである JIAJIA[1] にデータ転送回数を削減する機構を組み込んだ．JIAJIA では必要なデータの転送をページ単位で行っており，それぞれのページには固有のページ番号が付けられている．

今回実装した機構では，ページ要求の際に，過去に要求されたページ番号のパターンを元に将来要求されるであろうページ番号を予測する．要求のあったページと共に，その予測したページ番号のページを 1 回にまとめて，すなわち通信粒度を大きくして転送する．この要求のあったページと共にまとめて転送したページが将来使われた場合，転送回数を削減したこととなり，その分速度向上が図れる．しかし，予測が外れた場合には転送したページが無駄になり，転送時間を増加させてしまうので高い予測精度が必要となる．

2.2　次ページ予測機構

本研究で使用した次ページを予測する方法は以下の通りである．

- **ストライド値予測** 前回要求があったページ番号 L と今回要求があったページ番号 N との差を取り，その差と N との和である式 $P = (N - L) + N$ で、次に要求されるページ番号 P であると予測する方法 [2]．この差をストライドと言う．例えば，3 に続いて 4 とページ要求があった場合は、$(4 - 3) + 4$ という計算から次ページは 5 と予測する．ストライド値予測はページ要求が等差数列の場合に適した予測方法であるが，等差数列以外では予測が当たらなくなる．

- **コンテキストベース値予測** コンテキストベース値予測では，過去の連続した有限個のストライド値を保存し，そのパターンから次のストライド値を予測する．この予測したストライド値を利用することにより次ページ番号を予測する．この方法の場合，等差数列以外のストライドに一定のパターンがある場合にも有効であることが知られている．例えば，1，101，2，102，3，103 のようにページ要求するような場合でも，ストライドが 100，-99，100，-99，100，-99 となるので予測可能である．当然，等差数列は常に一定のストライドをとるという 1 つのパターンとしてみることができるので，コンテキストベース値予測でも等差数列を予測することが可能である．

3　評価

3.1　評価方法

以下に示す 3 つの次ページ予測手法の性能を比べるため実験プログラムを実行した．実験環境を表 1 に示す．

・予測機構を実装していない JIAJIA(normal)
・ストライド値予測をする JIAJIA(stride)
・コンテキストベース値予測をする JIAJIA(context)

表 1: 実験に用いた PC クラスタの構成

CPU	Intel Pentium4 Xeon 2.8GHz
メモリ	512MB
ネットワーク	ギガビットスイッチ
OS	RedHat Linux 9

Software DSM with a communication granularity tuning mechanism
Tomoya SAKAGUCHI, Sho SUZUKI, Tadayuki OHNO, Kenji KISE, and Toshitsugu YUBA
† Department of Computer Science, The University of Electro-Communications
‡ Graduate School of Information Systems, The University of Electro-Communications.

3.2 実行プログラムの説明

使用した実験プログラムの主要部分を図 1 に示す.

```
{ int i,j;
  int start, end;
  start = (N/jiahosts)*jiapid;
  end   = start+(N/jiahosts);
  if(jiapid==(jiahosts-1)) end = N;

  for(j=0; j<N; j++){
    for(i=start; i<end; i++){
      C[i][j] = A[i][j] + B[i][j];
      wait_t();  /*演算 2000 回分待つ*/
}}}
```

図 1: 実験プログラムの主要部分

このプログラムは，2 つの行列 A，B の和を求める．しかし，それでは計算時間が極端に短くなってしまうため，加算の後に一定時間待つことで計算時間を増やしている．その一定時間待つ動作を行っているのが wait_t 関数である．

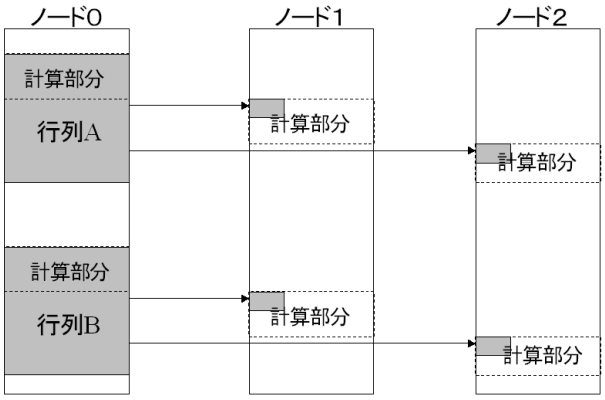

図 2: 実験プログラムの実行過程

図 2 は実験プログラムのメモリ配置を簡単に表している．2 つの行列 A，B はノード 0 に配置してある．その集中配置してあるデータを，ノード 0 以外の各ノードが自分の必要になったときに，ページを要求していく．プログラムの計算部分のループでは，A の行列のデータを要求した後に，B の行列のデータを要求している．このループを start から end までの間繰り返すので，ノード 0 以外のノードは A と B を交互に要求することになる．このページの要求パターンはコンテキストベース値予測で示した具体例と同様のパターンになる．

3.3 評価結果

normal，stride，context の各場合について，実験プログラムでのデータ転送回数を図 3 に示し，台数効果を図 4 に示す.

図 3 より，normal と stride の転送回数が一致していることがわかる．これは，このプログラムではストライド値予測が有効でないことを示している．context は

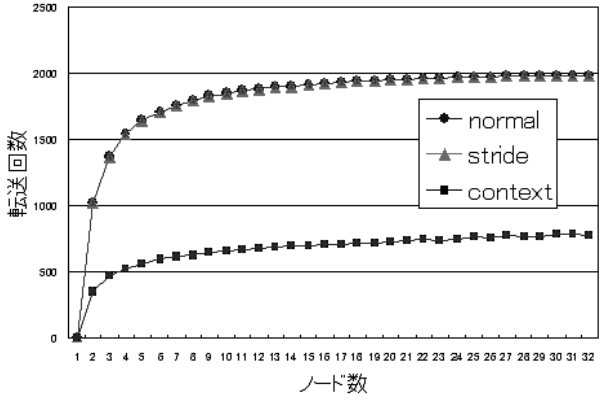

図 3: 転送回数の比較

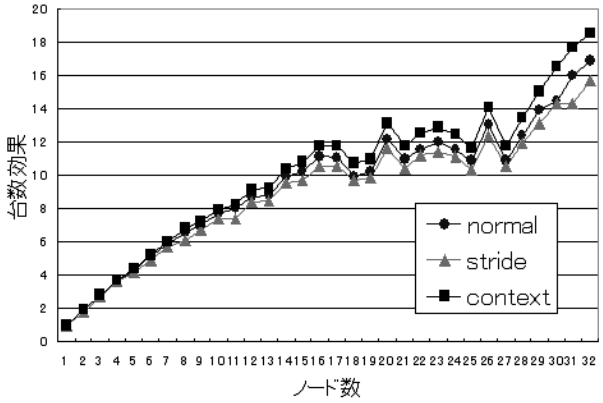

図 4: 台数効果の比較

normal と比べて転送回数を約 61%削減することに成功している．

図 4 より，stride は normal より台数効果が出ていない．これは，予測オーバヘッド分の実行時間が増加してしまったからである．context は normal と比べ，32 ノードで約 8.3%の性能向上が見られた.

4 まとめ

実験プログラムの場合により，コンテキストベース値予測の方がストライド値予測より優れていることを示すことができた．配列同士の演算では，同様なページ要求パターンが起こるので，そのような場合でもコンテキストベース値予測の方が有効であると言える．

参考文献

[1] M.Rasit Eskicioglu, T.Anthony Marsland, Weiwu Hu and Weisong Shi: Evaluation of the JIAJIA Software DSM System on High Performance Computer architectures, *HICSS-32* (1999).

[2] 田邊浩志, 吉瀬謙二, 本多弘樹, 弓場敏嗣: 通信粒度を動的に変更するソフトウェア分散共有メモリ, 電子情報通信学会総合大会, No. D-6-5, p. 56 (2002).