RDTにおける分散共有メモリシステムの性能評価

2B－8

福嶋泰仁　好村公一　工藤知宏
東京工科大学情報工学科

1 はじめに

現在、国内7大学でJUMP-1[1]と呼ぶ超並列計算機を共同開発中である。JUMP-1は、分散共有メモリを持ち、RDT(Recursive Diagonal Torus)[2]と呼ぶクラスタ間結合網を用いる。JUMP-1では分散共有メモリのコヒーレンシ維持に階層マルチキャスト方式と呼ぶディレクトリ縮約方式が用いられる。本稿では、RDT上での階層マルチキャスト方式の性能評価について報告する。

2 キャッシュのコンシステンシ維持

JUMP-1ではページ単位でディレクトリを持つためメモリコヒーレンシ維持のためのメッセージの宛先数が多くなる。

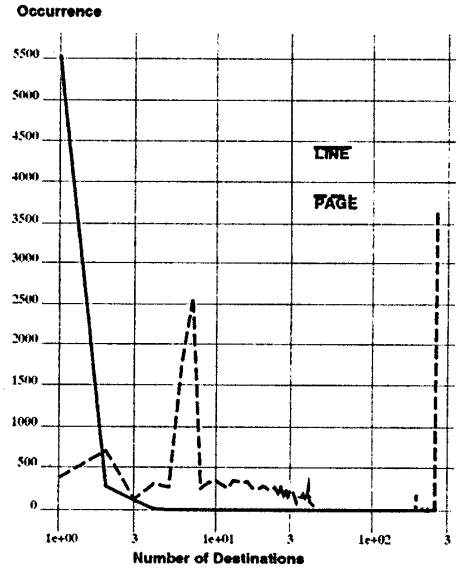

図1: invalidate型プロトコルを用いた場合の無効化メッセージ宛先数の分布

図1と図2は、並列ベンチマークプログラム集SPLASH[3]のMP3D問題を1024クラスタで1ステップ実行した際の、invalidate型とupdate型のそれぞれについて宛先プロセッサ数の分布を表したものである。これは、アドレストレースを基に分析したもので、キャッシュラインの大きさを32バイト、ページの大きさを4Kバイトとして、ライン単位とページ単位で管理した場合のそれぞれについて表している。

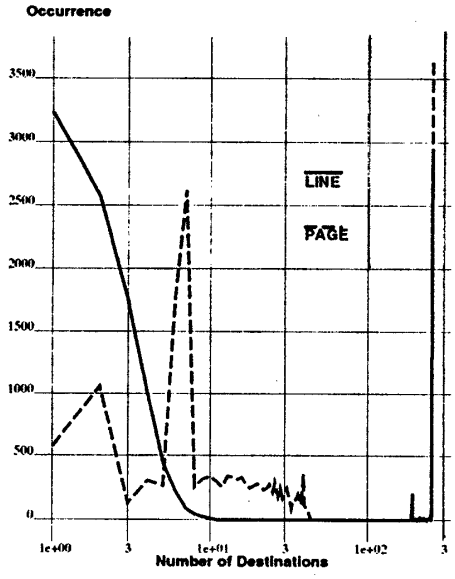

図2: update型プロトコルを用いた場合の更新メッセージ宛先数の分布

図1と図2より、ライン単位でディレクトリを管理すれば、特にinvalidate型で宛先プロセッサ数が3程度以下になる確率がかなり高いのに対し、ページ単位の管理では宛先数4にピークが見られ、それ以上の宛先を持つ場合もかなり多いことが分かる。このためJUMP-1では階層マルチキャスト方式と呼ばれるディレクトリ構成方式が用いられる。

3 階層マルチキャスト

超並列計算機を構成するネットワークを階層性を持つ木構造を構成するものととらえ、この階層性を利用してマルチキャストを行なう手法を階層マルチキャストと呼ぶ。RDTでは上位トーラスを構成する各ノードが下位トーラスのノードにマルチキャストすることにより疑似的に8進木を構成していると考えることができる。この各節にビットマップを対応させ、各節のビットマップを組み合わせて階層化マップを得る。本稿で扱う階層マルチキャスト方式では、

- ある節以下はブロードキャストとする
- 複数の節で同一のビットマップを用いる

のいずれか、もしくは両方の組み合わせによって節毎にビットマップを持つ。図3に3進木を用いた場合の3つの階層マルチキャスト方式を示す。

Evaluation of distributed shared memory on RDT
Yasuhito FUKUSHIMA, Koichi YOSHIMURA.
Tomohiro KUDOH
Tokyo Engineering University

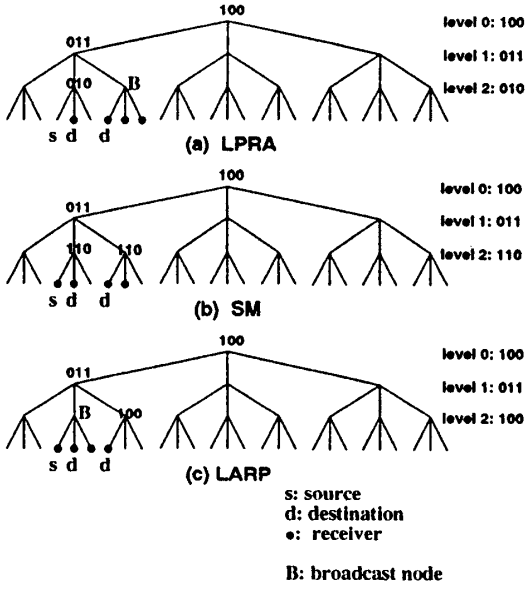

図 3: 階層マルチキャスト方式

4 階層マルチキャストにおける受けとられるパケット数の評価

LPRA、SM、LARP のそれぞれについて $8^4 = 4096$ のクラスタから構成される RDT 上でパケットの宛先の数と分布を変化させた時に、一回のマルチキャストあたり実際にいくつのクラスタにパケットが受けとられるのかを調べた。今回は、宛先を定められた分布に従う乱数により決定し、それぞれの場合について Sun OS4.1.3 の標準の乱数発生ライブラリを用いて 10000 回の試行を行ない、平均値をとった。

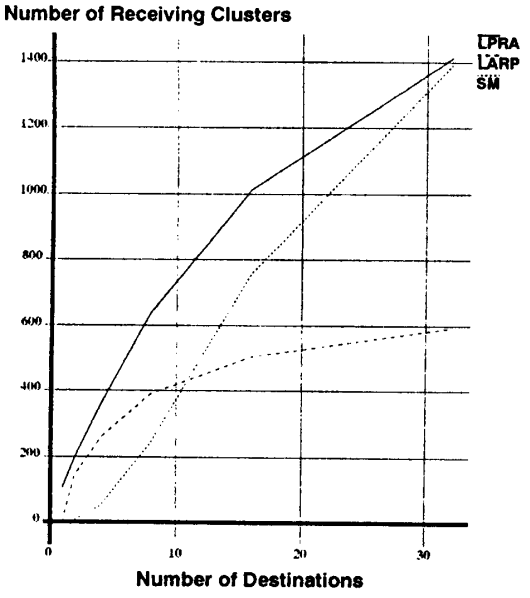

図 4: 分散 5 で分布している場合

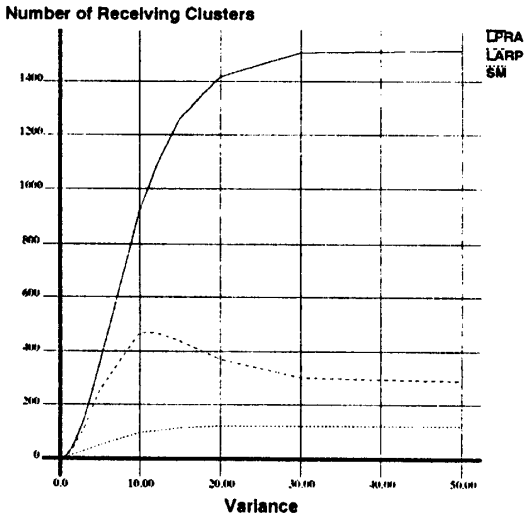

図 5: 宛先数 4 で分散を変化させた場合

RDT のランク 0 トーラス上で宛先の送信元からの X 方向、Y 方向の相対位置の分布がそれぞれ独立な (ランク 0 トーラス上の 1 リンクを単位として) 分散 5 の正規分布に従っている場合のパケットを受けとるクラスタ数のグラフを図 4に示す。この図より宛先数が 10 程度以下では SM が、それより多い時には LARP が有利であることが分かる。

前述のように、ページ単位でディレクトリを管理した場合、宛先数 4 程度にピークがある。宛先数を 4 とし、分散を変化させてパケットを受けとるクラスタ数を調べた結果を図 5に示す。常に SM がもっともパケットが送られるクラスタ数が少なく、分散が大きくなっても 120 程度に抑えることができることがわかる。

5 おわりに

今回の評価結果から、宛先数が 4 程度または分散が 5 程度であれば、階層マルチキャストの方式によっては宛先数があまり多くならないことが分かる。今後、実際のプログラムの振舞いを想定したシミュレーションを行なって、より精密な RDT の評価を行なっていく予定である。

参考文献

[1] 文部省重点領域研究「超並列原理に基づく情報処理基本体系」第 2 回シンポジウム予稿集.1993.

[2] Y.Yang, H.Amano. H.Shibamura, and T.Sueyoshi.Recursive diagonal torus:An interconnection network for massively parallel computers.In *Proc. of 1993 IEEE Symposium on Parallel and Distributed Processing*,1993.

[3] J.P. Singh, W. Weber, and A. Gupta. Splash:Stanford parallel applications for shared-memory. In *Tech. Report. Computer System Laboratory,Stanford University*.1992.