超並列計算機 SR2201 の並列ファイル転送プログラム HFTP

4Z－7

山崎 康雄† 森 利明† 鍵政 豊彦† 藤田 不二男‡

†(株)日立製作所 中央研究所 ‡(株)日立製作所 ソフトウェア開発本部

1.はじめに

近年大規模データを計算機間で高速転送する必要性が高まっている. 特に超並列計算機やスーパコンピュータ等のサーバ機の間で膨大な数値計算結果をやりとりする上でこの問題は深刻になっている.

本稿では, 超並列計算機SR2201における並列ファイル転送プログラム「HFTP」の高速化方式について報告する.

2.HFTP の概要

図 1 に HFTP を用いて SR2201 からファイルサーバへファイル転送を行う例を示す. 高速チャネル HIPPI(High Performance Parallel Interface) で SR2201 とファイルサーバを結び, ディスク入出力ネックを解消するために, ファイルサーバは複数のディスクを, SR2201 は複数のディスクノードを持つ. ディスクおよびディスクノードに対応するファイル入出力プロセスが並列にファイル入出力を行い, 外部通信プロセスどうしが計算機間の通信を行う. SR2201 の内部通信プロセス間では, データコピーを行わない高速なノード間通信を行う.

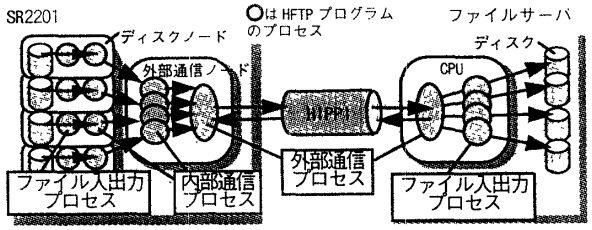

図1 HFTP を用いたファイル転送

計算機間の通信プロトコルとして, OS でのバッファリングを行わない大規模データ転送向け通信プロトコル(HFTP プロトコル)を開発した.

3.HFTP プロトコルにおける高速化方式

一般に, OS でのバッファリングを行わない通信プロトコルでは, データコピー処理の替わりに制御通信が発生する. HFTP プロトコルでは, 制御通信の回数を削減する一括転送方式と, 制御通信をデータの通信と同時に行うパイプライン転送方式という2つの高速化を行って, 制御通信によるオーバヘッドを回避した.

A Parallel File Transfer Program HFTP on SR2201
Yasuo YAMASAKI, Toshiaki MORI, Toyohiko KAGIMASA, Fujio FUJITA, Hitachi, Ltd.

3.1一括転送方式

複数のファイル入出力プロセスから送られるデータを逐次に処理する転送を逐次転送, 一括して処理する転送を一括転送と呼ぶ. 図2は4つのデータを, それぞれ逐次転送する例と一括転送する例を示す. 要求送信から完了受信までの 4 つの送受信からなる転送手順でひとつのデータ転送を行う. 逐次転送では転送手順を逐次に繰り返すが一括転送では複数の転送手順を一括するので, 一括転送は逐次転送に比べ高速である.

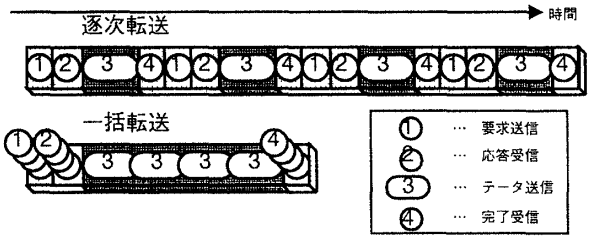

図2 一括転送

しかし, この一括転送を単純に実装すると, 多くのデータを一括すると全体の応答受信が遅れてしまう. 外部通信プロセスは OS の通信バッファを介さずに, 相手計算機の外部通信プロセスとの間で直接データを転送する. あるプロセスは, 他のプロセスが受信バッファ内のデータ処理を終えるまでは次のデータを受信できないので, 受信準備にかかる時間は一定ではない. 一括転送では全ての受信準備が整ってから応答受信を行うため, 一括するデータ転送の数が多くなるほど受信準備の時間がばらつき, 応答受信が遅れる.

そこで, 一括された転送手順を動的に再構成して受信準備の遅れを小さくする方式を開発した. 全ての受信準備が整うまで待たずに, 受信準備が整った要求を先に処理する方式である. 受信側の外部通信プロセスは一括された要求送信を受け取ると, その時点で受信準備が整っている要求に対しては要求の「応答」を, 整っていない要求に対しては要求の「拒絶」を, 一括して返信する. 送信側の外部通信プロセスは応答を得た要求については転送手順を続行し, 拒絶を得た要求については転送手順をはじめから再実行する.

図 3 に一括転送における動的再構成の例を示す. 4つのデータ A～D の転送を一括して処理している. 要求送信が行われた時点ではデータ D の受信準備が整っておらず, 通常の一括転送ではデータ D の受信準備が整うまでデータ A～D の転送は行われない. 一方動的再構成を行う一括転送では, 要求送信時に受信準

備の整っているデータA～Cを直ちに転送している.

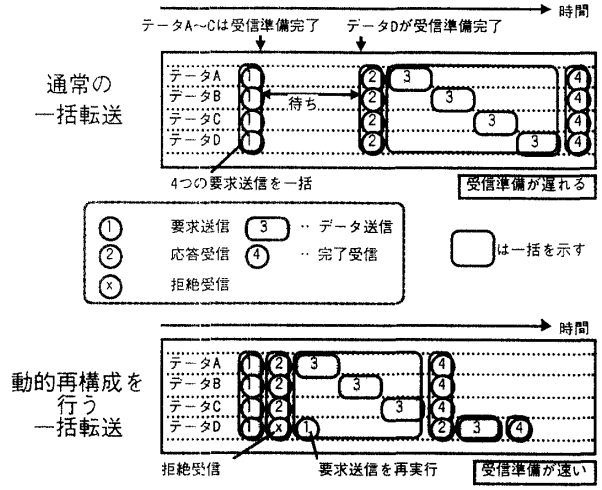

図3 一括転送における動的再構成

3.2パイプライン転送方式

複数のファイル入出力プロセスから送られるデータを複数のパイプラインに順次割り当て，複数のパイプラインを並行に処理する転送がパイプライン転送である．図4は4つのデータをそれぞれ逐次転送する例とパイプライン転送する例を示す．パイプライン転送では，要求送信とデータ送信をまとめた送信および，応答受信と完了受信をまとめた受信，を同時に行うので，単純転送の場合に比べてデータの転送が高速である．

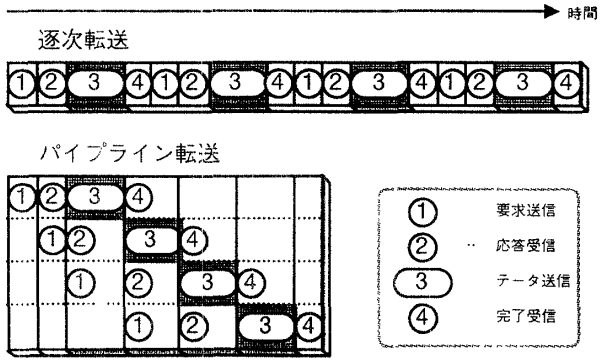

図4 パイプライン転送

しかし，このパイプライン転送を単純に実装すると，受信準備が少しでも遅れた場合に転送手順が大きく遅れていまう．パイプライン転送では，他のパイプライン処理を待たせないために拒絶受信と転送手順の再実行を行うが，パイプライン中の転送手順の順序は固定であるので再実行はすぐには行えない．

そこで，転送手順を別のパイプラインに動的に変更して転送手順の遅れを小さくする方式を開発した．転送手順の再実行を早く行えるパイプラインを選択し，そのパイプラインへ再実行処理を移す方式である．

図5はパイプライン転送における動的変更を行うパイプライン転送の例を示す．4つのデータをパイプライン転送している．パイプラインCにおいて拒絶された転送手順の再実行は，通常のパイプライン転送では2回の送受信の後に行うが，動的変更を行うパイプライン転送では即座に行う．

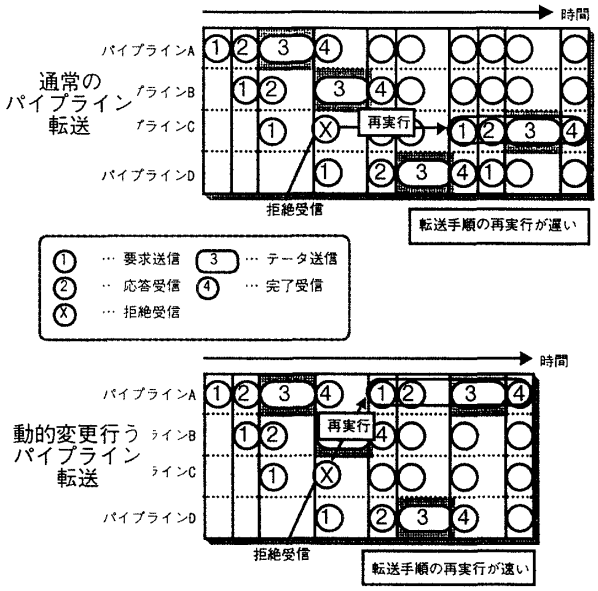

図5 パイプライン転送における動的変更

4.評価

HFTPによるSR2201からベクトルプロセッサS3600へ24並列のファイル転送を行った性能の実測値を図6に示す．逐次転送に比べ，2つの高速化手法を適用することで約2倍の性能向上を得た．

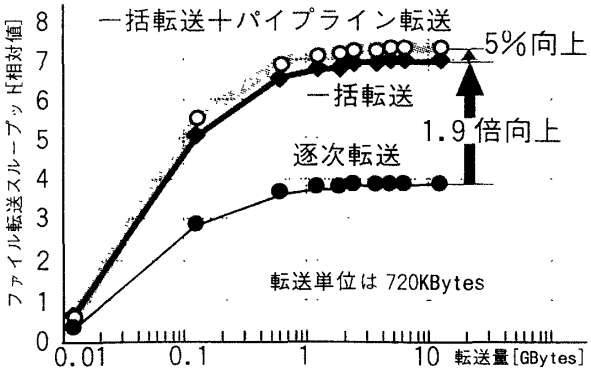

図6 ファイル転送性能

5.むすび

高速な並列ファイル転送プログラムHFTPを開発した．OSのバッファリングを行わない通信プロトコルを開発し，更に一括転送，パイプライン転送という2つの高速化を適用することで，大規模ファイルの並列転送においてよい台数効果が得られた．

6.謝辞

日頃ご協力を頂いている，筑波大学CP-PACSプロジェクトの関係者に深謝致します．