分散処理システムを利用した並列処理環境における通信処理の影響

2P−8

手塚 忠則 了戒 清 Bernady O. Apduhan 末吉 敏則 有田 五次郎
(九州工業大学 情報工学部)

1. はじめに

近年，大学や研究所では複数の高性能ワークステーションをローカルエリアネットワークによって接続した分散処理システムが広く利用されるようになってきている．この分散システムを活用して並列処理を行なわせることが可能になれば，アプリケーション実行速度の向上が期待できる．そこで我々は，分散システムの持つ計算機間の通信機能を利用して並列処理を行なうための基本システムを作成し，分散システム上で高速処理を行なう分散スーパーコンピューテイング環境(Distributed Supercomputing Environment, DSE)の構築に関する実験を行なっている．

DSEは，オペレーティングシステムに手を加えることなくUNIXのユーザレベルプロセスとして動作する移植性の高いものである．現在までにDSEを用いて様々な並列アプリケーションを実行し，DSEの問題点を明らかにするとともにその改良を行なってきた[1]．本論文では，DSEにおけるワークステーション間の通信処理の影響について述べるとともに，現在のDSEにおける問題点について述べる．

2. 並列処理環境の概要

DSEは，分散メモリ共有型並列計算機と等価な機能を分散システム上に実現するものである[2]．DSEは，複数のワークステーションによって構成される(図1)．

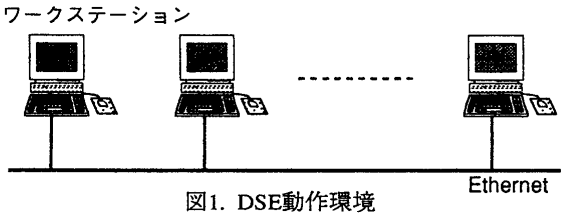

図1. DSE動作環境

図2に個々のワークステーション上で動作するDSEの構成要素を示す．DSEの構成要素は，DSE-kernelとDSE-processの2つのUNIXプロセスからなる．DSE-kernelは，2つの機能ブロックによって構成される．1つはメッセージ交換機構で他のワークステーションとの通信を行なうための機構である．他の1つは分散メモリ共有型並列計算機のプロセッサ要素の機能を実現するための機構で，DSE-processのスケジューリングおよび共有メモリの管理を行なう．現在，共有メモリは各プロセッサ要素毎に256Kbyte(4Kbyte×64page)を用意している．DSE-processは，DSEにおいて逐次実行される処理単位でありDSE-kernelにより先着順(First Come First Served, FCFS)でスケジューリングされる．

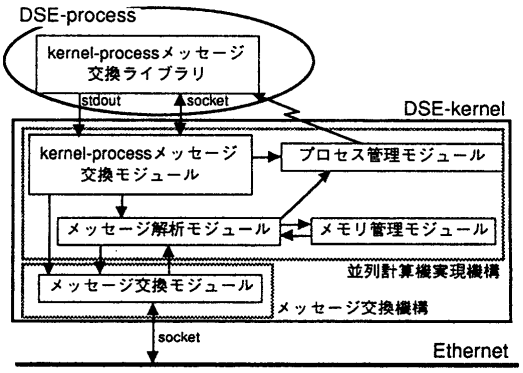

図2. システム構成

DSEは，UNIXオペレーティングシステムに手を加えずにUNIXユーザレベルプロセスによって実現されており，他のワークステーションとの通信およびDSE-kernelとDSE-process間の通信には4.3BSD UNIXが提供するストリーム型ソケット(socket)を利用している．また，各ワークステーション間の通信は，可変長のメッセージによって行なっている．このメッセージの内容は，並列処理のための基本操作であるシステムプリミティブである．現在，共有メモリアクセス，同期・排他制御および並行プロセス生成/終了のためのシステムプリミティブを15種類用意している．

ここで，DSEにおけるシステムプリミティブの実行の流れを共有メモリのREADを例にして説明を行なう．DSE-processからのREADリクエストはソケットを介してkernel-processメッセージ交換モジュールへ渡される．ここでDSE-processのREAD要求はメッセージ交換モジュールを介して適当なワークステーションにネットワークを通して送られる．送られたデータは相手のメッセージ交換モジュールで受取られ，メッセージ解析モジュールで解析され，共有メモリの内容が読み出される．読み出された内容は再び要求元のワークステーションへメッセージ交換もモジュールを通して送られる．このように，共有メモリのREADは，ネットワークを通した2度のメッセージ交換によって実現されている．

3. 通信処理影響の測定

ワークステーションでの通信処理の時間を調べるためにDSE-kernelへのタイマの組込み及びイーサネット

Effect of Communication Processing in Parallel Processing Environment using Workstation Clusters
T. Tezuka, K. Ryokai, Bernady O. Apduhan, T.Sueyoshi and I. Arita
Kyushu Institute of Technology

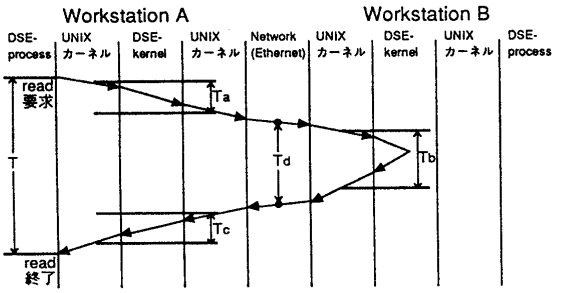

図3. 共有メモリreadにおける通信

上のパケットの流れを調べることができるパケットモニタを利用して，4byteの共有メモリのREADについてSun Microsystems社のSparcStation2を2台用いて測定を行なった．図3は測定を行なった共有メモリREADの概略である．TはDSE-processがREAD要求を行なってからREAD操作が終了するまでの時間，TaはDSE-kernelがDSE-processから要求を受けてワークステーションBに要求を送るまでの時間，Tbはワークステーション BのDSE-kernelがその要求を受けてから共有メモリのREADを行ない共有メモリの内容をワークステーションAに送るまでの時間，TcはワークステーションAのDSE-kernelが共有メモリの内容を受けてからDSE-processへその内容を送るまでの時間である．これらの値はDSE-kernelにタイマを組み込んで調べたものであるが，この値にはUNIXカーネルとDSE-kernelの間でデータのやり取りを行なう時間，つまりUNIXのread()/write()システムコールの時間が含まれている．従って，これらの値にはUNIXカーネルでの処理時間の一部が含まれていると考えられる．

また，Tdはパケットモニタを用いてネットワークに流れたパケットの時間を計測することによって時間測定を行なったものである．表1に測定結果を示す．

表1. 測定結果

	処理時間(μsec.)
T	2860~2878
Ta	540~545
Tb	513~518
Tc	961~970
Td	837~845

TdからTbを引いた値，つまり(Td-Tb)は，UNIXのread()/write()システムコールに含まれるUNIXカーネル処理時間を除いたワークステーションBでのUNIXカーネルの処理時間である．これは，共有メモリのREADに要する処理時間全体の約10%を占めている．ワークステーションAでのUNIXカーネルの処理時間もこれとほぼ同じであると考えると，イーサネットを通した通信におけるUNIXカーネル内の処理時間はREAD処理全体の約20%を占めることになる．さらに，Ta,Tb,Tcの処理時間のうち，UNIXカーネルとのインタフェースであるUNIXのread()/write()システムコールに要する時間を調べると図4になる．これから，UNIXのread()/write()システムコールがREAD処理全体の約60%を占めていることが分かる．つまり，現在のDSEではUNIXのread()/write()システムコールに要する時間およびUNIXカーネルでの処理がREAD処理全体の大部分を占めていることが分かる．

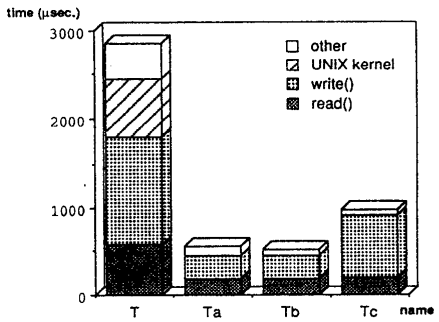

図4. read()/write()システムコールの処理時間

4. おわりに

解析の結果，DSE-kernelとUNIXカーネルのインターフェース，つまりUNIXシステムコールread()/write()に要する時間がDSEでの通信処理時間の多くの部分を占めていることが分かった．

今後，DSEの処理速度を向上するためには，このread()/write() UNIXシステムコールを削減する事がもっとも効果的であると考えられる．しかしながら，現在のDSEは既にできる限りのread()/write()の削減を行なっているため，現在の実装法でこれ以上の速度向上を図ることは難しい．そこで，速度改善を図る1つの方法としてDSE-processとDSE-kernelを1つのUNIXプロセスにまとめる実装法を検討した．この実装法であれば，DSE-kernelとprocess間でのread()/write()システムコールを削除することができるので，大幅な処理速度の向上が期待できる．

現在，上記の設計のDSEをSun OSの提供するlight-weight process[3]を利用して実装を行なうと共に，DSEにおける通信処理の影響についてさらに詳しい調査を進めている．

参考文献

[1] B. O. Apduhan, T. Sueyoshi, Y. Namiuchi, T. Tezuka, T. Fujiki and I. Arita, "Experiments and Analysis Toward Distributed Supercomputing on a Distributed Workstation Environment", *in Proc. of 1991 International Symposium on Supercomputing*, pp. 182-190, Japan, Nov. 1991; or *SUPERCOMPUTER Special Issue for ISS'91(Revised Version)*, Stichting Academisch Rekencentrum Amsterdam (SARA), volume VIII, number 6, pp. 90-100, November 1991.

[2] B. Apduhan, T. Sueyoshi, T. Tezuka, Y. Ohnishi and I. Arita, "Reconfigurable Multiprocessor Simulation Environment on a Distributed Processing System", *in Proc. of 6th International Joint Workshop on Computer Communications*, pp. 283-290, Fukuoka, Japan, July 1991.

[3] *Sun OS Reference Manual Vol.II-3L*, Sun Microsystems, Inc., 1991.