分散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

手塚 忠則　　了戒 清　　末吉 敏則
(九州工業大学 情報工学部)

概要

我々は，既存のオペレーティングシステムに手を加えることなく，分散システム上で並列処理を可能にする，分散スーパーコンピューティング環境(Distributed Supercomputing Environment, DSE)の構築を行っている．本論文では，このDSEにおける通信処理に関する詳細な調査の結果について述べる．さらに，この調査結果に基づき，現在のDSEの通信処理の速度にもっとも影響を与えている要因を明らかにする．

1. はじめに

ネットワーク技術の発展に伴い，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(Local Area Network, LAN)等で接続し，分散システムを構築するようになった．このような分散システムでは，その通信機能を利用して，ワークステーション間で比較的に高速な通信を行うことが可能である．また，分散システムでは，この通信機能を利用することにより，複数のワークステーションで1つのまとまった仕事，いわゆる並列処理を行う能力を潜在的に持っている．しかし，現在最も広く利用されているUNIXオペレーティングシステムでは，ユーザが並列処理を行うための十分な機能を提供していない．また，並列処理機能を提供する分散オペレーティングシステムが幾つか研究されてきた[1]が，これらの多くは，既存のオペレーティングシステムの変更や，新しいカーネルの構築を必要とする．大学のように，ワークステーションを複数のユーザで共同利用を行っている環境では，オペレーティングシステムのようなシステムソフトウェアの変更を行うことは非常に難しい．したがって，このような環境下では，現在のシステム環境を変更することなく並列処理が行える環境が望まれる．

そこで，我々の研究では，既存のオペレーティングシステムを変更することなくユーザに分散共有メモリ型並列計算機の機能を提供する，分散スーパーコンピューティング環境(Distributed Supercomputing Environment, DSE)の構築を行っている[2][3].

DSEは，分散共有メモリ型並列計算機の機能を，ワークステーション間のメッセージ交換によって実現する．DSEのように，分散システム上で並列処理を行う場合は，ワークステーション間の通信能力が重要となる．そこで，本研究では，このDSEにおける通信処理について詳しく調べることにより，通信処理

The Effect of Communication Processing in Parallel Processing Environment on a Distributed System.
by Tadanori Tezuka, Kiyoshi Ryoukai, Toshinori Sueyoshi
(Kyushu Institute of Technology)

の速度にもっとも影響を与えている部分の調査を行った．本論文では，まず，DSEについて簡単に説明を行う．続いて，通信処理の影響について，共有メモリ読出し処理の調査結果を例にして述べる．さらに，DSE上で偏微分方程式の解を求めるアプリケーションを実行した結果を利用して，通信処理の影響について調査した結果について述べる．

2. DSEの概要

DSEは，分散共有メモリ型並列計算機と等価な機能をユーザに提供する．図1は，DSEが実現する並列計算機のモデルである．

図に示すように，各プロセッサ要素は，非共有のメモリであるローカルメモリ(LM)と，共有メモリであるグローバルメモリ(GM)を持っている．このモデルにおいては，プロセッサ間ではグローバルメモリのみが共有され，ローカルメモリは他のプロセッサからアクセスすることはできない．また，グローバルメモリへのアクセスは，相互結合網を介した通信によって行われる．DSEでは，1つのプロセッサ要素がワークステーション1台に対応する．グローバルメモリについては，DSE kernelと呼ぶソフトウェアによって管理されている．また，ローカルメモリは，UNIXがプロセス(DSE process)生成時に確保するメモリ空間に対応している．

図2は，DSEのシステム構成である．図中の各ワークステーションは，図1のプロセッサ要素に対応している．また，イーサネットは，モデルの相互結合網に対応している．

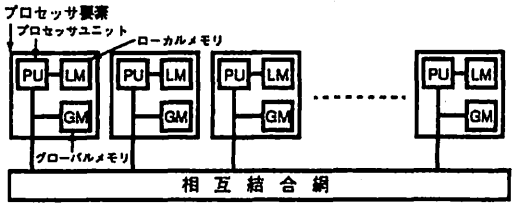

図1. 並列計算機のモデル

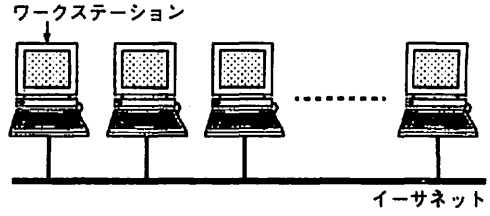

図2. DSEシステム構成

DSEは，イーサネット上に仮想的に様々な結合形態を構築可能である[4]ので，静的網であれば，モデルと同一の結合網によるワークステーション間の仮想的な接続が可能である．

各ワークステーション上では，DSE kernelと呼ぶソフトウェアがUNIXプロセスとして動作している．このDSE kernelは，並列計算機の機能を実現する基本ソフトウェアである．また，ユーザがDSE上でアプリケーションを実行している場合は，DSE kernelの他に，DSE processと呼ぶプロセスがUNIX上で動作する．

図3は，DSE kernel, DSE processおよびUNIXカーネルの関係を示したものである．図に示されるように，DSE kernelは5つのモジュールから構成される．また，DSE processには，kernel-processメッセージ交換ライブラリが組込まれている．kernel-processメッセージ交換ライブラリは，ユーザの作成したアプリケーションに組込まれるライブラリである．ユーザは，このライブラリを利用することにより，共有メモリのアクセスや他のプロセッサへのプロセスの起動の要求といった並列処理のための手続きを行うことができる．現在，DSEでは，共有メモリのアクセス／プロセスの起動・終了／同期・排他制御のための手続きをkernel-processメッセージ交換ライブラリにおいて，C言語の関数としてユーザに提供している．

ここで，ユーザのアプリケーションが，他のワークステーションの持つ共有メモリの

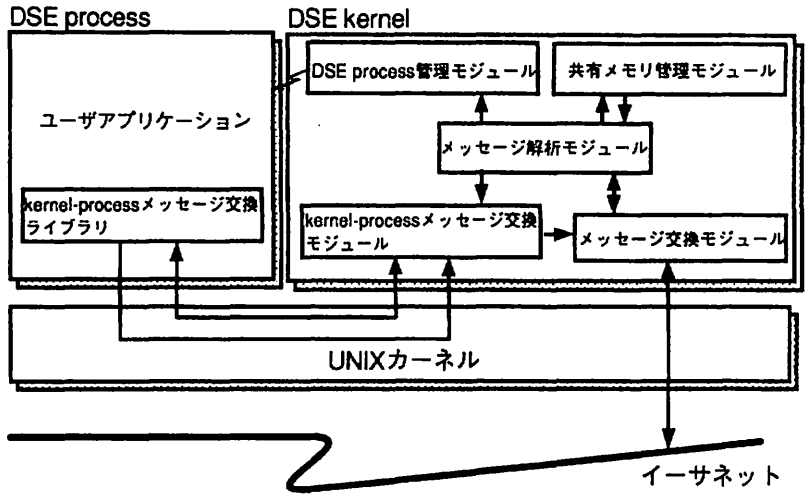

図3. ソフトウェア構成

読出し(READ)を行う場合の処理について説明を行う．共有メモリの読出しは，ワークステーション間の2回のメッセージ交換によって行われる．1つは，アプリケーションからの共有メモリの読出し要求メッセージであり，このメッセージには，読出しを行う共有メモリのアドレス，および読出すデータ長などが含まれている．もう1つは，要求に対する応答メッセージであり，このメッセージには，読出した共有メモリの内容が含まれている．以下に，この2つのメッセージの送受信処理について，メッセージの流れを用いて示す．ここでは，共有メモリアクセス要求を行ったワークステーションを要求元，共有メモリを持つワークステーションを要求先として示している．

(1)共有メモリ読出し要求

要求元: DSE process(kernel-processメッセージ交換モジュール)→ UNIXカーネル → DSE kernel(kernel-processメッセージ交換モジュール)→ DSE kernel(メッセージ交換モジュール)→ UNIXカーネル → イーサネット→

要求先: UNIXカーネル → DSE kernel(メッセージ交換モジュール)→ DSE kernel(メッセージ解析モジュール)

(2)共有メモリ読出し要求に対する応答

要求先: DSE kernel(メッセージ解析モジュール)→ DSE kernel(メッセージ交換モジュール)→ UNIXカーネル → イーサネット →

要求元: UNIXカーネル → DSE kernel(メッセージ交換モジュール)→ DSE kernel(メッセージ解析モジュール)→ DSE kernel(kernel-processメッセージ交換モジュール)→ UNIXカーネル → DSE process(kernel-processメッセージ交換ライブラリ)

以上のように，共有メモリの読出しには，計6回のカーネルを通した通信が必要である．このUNIXカーネルを通した通信には，ワークステーション間の通信には，UNIX 4.3BSDの提供するソケット[5](TCP/IP)を，DSE kernel-proces間の通信にはソケット(UNIX domain)を，それぞれ利用している．

2. 通信処理影響の評価

並列計算機においては,プロセッサ要素間の通信能力が,並列処理の速度に大きく影響を与えることはよく知られている．特に，DSEのようなLANベースでの並列処理を行う場合には,通信処理速度がアプリケーションの実行速度に与える影響は大きい．そこで，ここでは,DSEの通信処理に関して詳しく調査を行った結果について述べる．

2.1. 共有メモリアクセス性能

通信処理について詳しい調査結果を示す前に，まず,共有メモリのアクセス性能について示しておく．共有メモリのアクセスは，DSEにおいて通信処理を必要とするシステムがユーザに提供する機能の代表的なものであり,共有メモリアクセス速度が並列アプリケーションの実行速度に与える影響は大きい．

今回は，イーサネット(10Mbps)で接続された Sun Microsystems社 の Sparc Station 2を2台用いて,プロセッサ自身の共有メモリのread/write速度(ローカル)と，隣接したプロセッサからの共有メモリアクセス速度(リモート)を測定した．

測定の結果を表1に示す．結果は，16byteのメモリアクセスを1000回行った場合の1回あたりのメモリアクセスに要する時間である．

表1. 共有メモリアクセス速度

	ローカル	リモート
Read	1.011	2.684
Write	0.790	1.335

単位 ：ミリ秒

共有メモリへのwriteは，書込み確認を行わないため,1回のメッセージ交換しか必要としない．したがって，プロセスは,連続して書込み要求をDSE kernelに送ることが可能である．このため,共有メモリのwriteは，2回のメッセージ交換が必要なreadに比べて非常に高速である．

また，結果から,他のプロセッサからの共有メモリのreadには，約3ミリ秒要することが分った．これは，ローカルな共有メモリreadに比べて3倍程度かかっている．この処理時間の違いは,イーサネットを通した通信処理に要する時間の差である．また,共有メモリの読出しサイズが大きくなっても,数百バイトまでは,ほとんど処理時間が変らないことも他の実験によって確認している[6].

2.2. 共有メモリ読出しによる評価

DSEにおける通信処理の影響について詳しく調べるために，DSEの基本的操作の1つである,共有メモリの読出しについて詳しく調査を行った[7]．DSEにおける共有メモリの読出し処理については先に示した．図4は,この共有メモリ読出し処理を図式化したものである．

今回測定を行ったのは，図中のT, T_a, T_b, T_cおよび T_dに相当する処理の時間である．以下に，それぞれの処理について説明を行う．

T : ユーザのアプリケーションが共有メモリのREAD要求を送ってから，共有メモリの内容が返されるまでの時間.

T_a: DSE kernelがDSE processからのREAD要求を受取ってから，プロセッサBに送出するまでの時間.

T_b: プロセッサBでREAD要求を受取ってから，プロセッサAに，READされた共有メモリの内容を送るまでの時間.

T_c: プロセッサAのDSE kernelが共有メモリの内容を受取ってから,DSE processに渡すまでの時間.

T_d: READ要求メッセージがイーサネット上

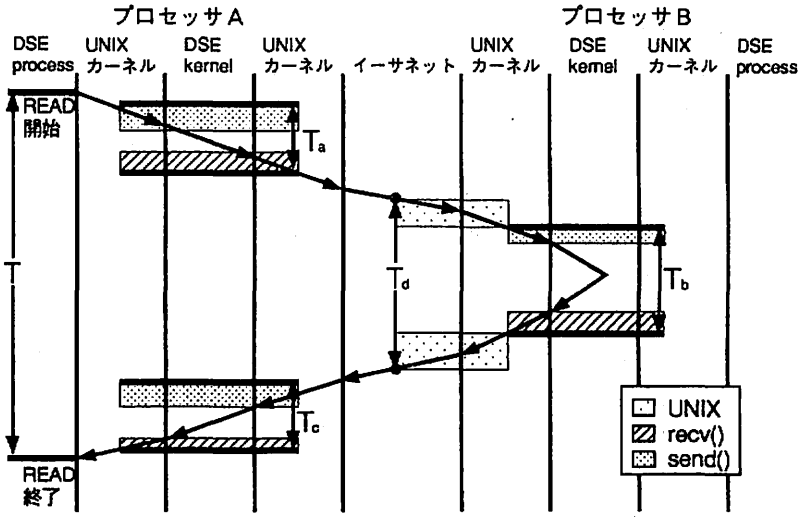

図4. 共有メモリreadの処理モデル

を流れてから,共有メモリの内容を含むメッセージがイーサネット上を流れるまでの時間.

以上のうち, T_a, T_b, T_cについては,DSE kernelにタイマを組込むことによって測定を行った. T_dについては, イーサネット上を流れるパケット情報をリアルタイムで獲得することが可能な,パケットモニタを利用して測定を行った. また, T_a, T_b, T_cに関しては,処理中に含まれるUNIXシステムコールsend()およびrecv()に要する時間について, 更に詳しく測定を行っている.UNIXのsend() および recv()は, 図中では,網掛および斜線で示される部分である.この2つのシステムコールは, DSE kernel, DSE processとUNIXカーネルのインタフェースの役目をしている. したがって,このシステムコール実行に要する時間は,UNIXカーネルとDSE kernel双方の処理時間に含まれると考えられる.

ここでは, 結果を示す前に,先に測定を行った環境を示しておく.

・ワークステーション
Sun Microsystems社 Sparc Station 2
・ネットワーク
イーサネット (10Mbps)
・読出し(read)サイズ
16bytes

実験は,ワークステーションを他のユーザのアクセスから完全に切離した状態で行っている. 表2に, 測定を行った結果を示す.

表2. 処理時間の測定結果

	処理時間	send()	recv()
T	2684	1507	638
T_a	649	368	202
T_b	628	351	210
T_c	1009	788	226
T_d	1010	$T_d - T_b = 382$	

単位 : マイクロ秒

結果から, 全処理時間に対するsend() /recv()に要する時間が非常に大きいことが

分る．send()/recv()は，ソケットを介した通信を行うためのUNIXシステムコールである．つまり，共有メモリの読出し処理では，このsend()/recv()に要する処理時間が非常に大きいことが分る．また，$T_d - T_b$は図4でUNIXと示される領域の処理時間である．物理媒体であるイーサネットの転送時間は無視することが可能なほど小さいので，この時間は，send()/recv()の処理時間を除いたUNIXカーネルでの処理時間である考えることができる．また，プロセッサAとプロセッサBのUNIXカーネルでは，メッセージの受取りと送出を各1度づつ行っているので，プロセッサBでの処理時間もほぼ同じであると考えられる．したがって，send()/recv()の処理を除いたUNIXカーネルでの処理時間は，$2\times(T_d - T_b) = 764$マイクロ秒となる．

以上の結果から，共有メモリ読出し時間に対する各処理時間の占める割合をグラフ化すると図5になる．表2の結果から分るように，図4のモデルから計算した共有メモリ読出に要する全処理時間$(T_a+T_b+T_c+(T_d-T_b)\times2)$と，Tの値は一致しない．この主な原因は，測定のためにDSE kernelに組込んだタイマ処理に要する時間である．タイマの処理は，(1)タイマの開始,(2)測定を行う処理の実行,(3)タイマの停止,(4)タイマの値のファイルへの書込み，という手順で行われている．この内，(1)から(3)の間におけるタイマの使用による処理時間の増加は，数マイクロ秒であることを測定により確認している．したがって，処理時間の測定結果に含まれるタイマ処理時間は数マイクロ秒であり，タイマ組込による処理時間への影響は極めて少ない．しかし，(4)の処理については，数百マイクロ秒の処理時間が必要であり，これがタイマを組込んだ場合に全体の処理時間を増加させる要因となっている．そこで，グラフでは，共有メモリ読出し処理時間，

$$T = (T_a+T_b+T_c+(T_d-T_b)\times2)$$

として計算を行った．

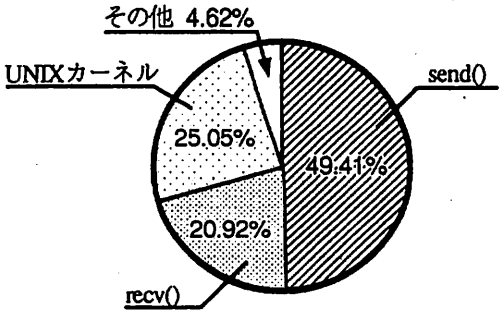

図5. 共有メモリアクセスにおける各処理の割合

グラフより，共有メモリ読出しに要する時間の約70%がUNIXシステムコール send()およびrecv()に要した時間であることが分る．また，send()/recv()にUNIXカーネルの処理時間を加えた時間が，処理時間全体の約95%を占めていることが分る．つまり，DSE kernelにおける通信処理時間の大部分がUNIXカーネル内の処理と，DSE kenelとUNIXカーネルとのインターフェースであるシステムコール send()/recv()に要している時間であることが分る．

2.3. 並列アプリケーション実行による評価

先に示した実験により，共有メモリの読出し処理のうち，もっとも処理時間を要する部分は，UNIXシステムコールsend()/recv()であることが分った．ここでは，実際のアプリケーションをDSE上で実行させた場合の通信処理の影響について調査を行った．ここで利用したのは，並列アプリケーションとして一般的な，偏微分方程式の解を求めるアプリケーションである[2].

以下に測定を行った環境を示す．

・ワークステーション

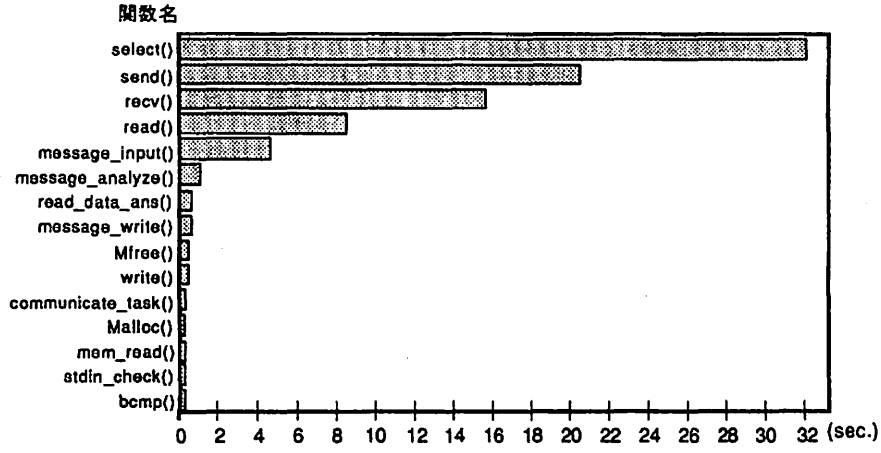

図6. DSE kernel内の手続きの処理時間

Sparc Station 2 × 16台

・ネットワーク

イーサネット(10Mbps)

・アプリケーション

偏微分方程式の解を求める(SOR法)

メッシュサイズ 48×48

DSE kernel内の処理時間の測定には，UNIXの *prof* コマンドを利用している．今回の実験では，profコマンドによって得られる情報のうち，各手続の総処理時間の統計情報を利用した．図6は，測定を行った結果である．この結果では，処理時間の合計が非常に小さいものや，DSE kenrelの初期化処理(ワークステーション間の接続など)など，評価に重要でないと考えられる部分の削除を行っている．

測定結果から，DSE kernelの処理時間のうち，select()，send()，recv()および read()の処理時間が非常に大きな割合を占めていることが分る．この内，select()は，多重I/Oイベント待ちのためのUNIXシステムコールであり，DSE kernelでは，他のプロセッサからのメッセージ，DSE processからの要求待ちを行うために用いている．したがって，select()に要する時間が大きいことは，DSE kernelがイベントの発生を待っている時間，つまりアイドルな時間が大きいことを示している．

また，先に述べたように，send()/recv()はプロセッサ間の通信とDSE processとの通信に用いている．したがって，この処理時間が大きいことは，DSE kernelの処理時間全体に占める通信処理時間の割合が大きいことを示している．また，read()は，DSE processの標準出力(stdout)から内容を受取るために用いている．このread()の処理時間が大きいのは，使用した並列アプリケーションが，その途中の計算結果を標準出力へ書出しているためである．

以上のように，並列アプリケーションの実行におけるDSE kernel内の処理時間の測定から見ても，UNIX システムコール send()/recv()に要する時間が非常に大きいことが分った．

3. まとめ

本論文では，DSEの通信処理に関して調査を行った結果を示した．この結果，現在のDSEでは，UNIXカーネルとのインタフェース部分がもっとも重いことが分った．つまり，DSEにおける通信オーバヘッドのほとん

どが,UNIXに依存した部分であることが分った．したがって，現在の構成のままDSE kernel内の処理を高速化しても，DSEの速度向上にほとんど効果が無いことが分る.DSEがUNIX上で動作する環境である限り,ワークステーション間の通信には,UNIXシステムコールsend()/recv()，または等価なUNIXシステムコールの使用を行う必要がある.このため,この部分の処理時間の大幅な短縮は期待できない．そこで，send()/recv()を用いているもう1つの部分である,DSE kernel-process間の通信処理に着目し，この部分についてsend()/recv()を削減する方法を考えている．これは，現在2つのUNIXプロセスとして動作しているDSE kernelとDSE processを1つのUNIXプロセスにまとめることによって実現可能である．そこで，現在，Sun OSの提供するLightWeight Processライブラリを用いて，この2つのプロセスを1つのプロセスにまとめた構成のDSEの構築に関する研究を行っている．

参考文献

[1]前川 守,所 真理雄,清水 謙太郎,分散オペレーティングシステム UNIXの次にくるもの,共立出版株式会社,1991.

[2]B. 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. 183-190, Nov. 1991; or *SUPERCOMPUTER Special Issue for ISS'91*, SARA, vol. Ⅷ, No. 6, pp.90-100, 1991.

[3]T. Tezuka, K. Ryokai, B. Apduhan and T. Sueyoshi, "Implementation and Evaluation of a Distributed Supercomputing Environment on a Cluster of Workstations", *in Proc. of 1992 International Conference on Parallel And Distributed System*, pp. 58-65, Dec. 1992.

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

[5]W. Richard Stevens, *UNIX® Network programming*, Prentice-Hall, Inc., 1990.

[6]B. Apduhan, T. Sueyoshi, T. Tezuka and I. Arita, "The Effect of Communication Processing in Network Supercomputing Environment", *in Proc. of 7th Internaitonal Joint Workshop on Computer Communications*, pp. 373-380, July 1992.

[7]手塚 忠則,了戒 清,B. Apduhan,末吉 敏則,有田 五次郎,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,情報処理学会第45回全国大会論文集(4),2P-8, Oct. 1992.