スケーラブルな分散サーバ環境の研究
— *SSS–CORE* の機能拡張 —
A study of the scalable distributed-computing servers
— Functional extension of the *SSS–CORE* —

松本　尚 [*1*2]
tm@is.s.u-tokyo.ac.jp
渦原　茂 [*3]
uzu@axe-inc.co.jp
平木 敬 [*1]
hiraki@is.s.u-tokyo.ac.jp

[*1] 東京大学 大学院理学系研究科 情報科学専攻
[*2] 科学技術振興事業団 さきがけ研究21「情報と知」領域
[*3] 株式会社 アックス

近年、コンピュータネットワーク（インターネットや企業内ネットワーク）の規模が拡大し、ネットワーク利用者は急速に増大している。また、ネットワークの利用方法も年々高度化して、マルチメディアデータがネットワーク上において流通し始めている。これに伴い、ネットワークを流れるデータ量が大幅に増大している。このため、ネットワークを介した情報処理の中核となるコンピュータ（サーバマシン）への性能要求が増大している。企業内ネットワークを構築する企業やインターネットプロバイダはこの要求増大に、高価なサーバ用計算機（専用並列計算機や大型機）を導入することにより対処している。既に導入したサーバマシンの処理能力が不足した場合には、さらに高価な計算機とリプレースする必要に迫られ、リプレース時にはソフトウェアおよびデータを移し換えるために、多大な作業コストが発生する。本研究においては、これらの状況を大幅に改善するために、サーバマシンをワークステーションクラスタによって構築し、その能力を構成マシンの台数によってスケーラブルに変更可能にするための基本ソフトウェア群を研究開発する。

1 はじめに

1.1 研究の背景

近年、コンピュータネットワーク（インターネットや企業内ネットワーク）の規模は拡大し、ネットワーク利用者が急速に増大している。特に携帯電話によるインターネット利用者が急増している。また、ネットワークの利用方法も年々高度化して、マルチメディアデータがネットワーク上において流通し始めている。携帯電話によるデータ通信の容量もブロードバンド化によって、この一、二年のうちに大幅に増強されつつある。これらの現象に伴い、ネットワークを流れるデータ量が大幅に増大し、多くの人にとってネットワークの重要性が高まっている。このため、ネットワークを介した情報処理の中核となるコンピュータ（サーバマシン）への性能と信頼性への要求が増大している。しかし、現状ではその要求に応えることができていない。一例を挙げれば、携帯電話とインターネットを中継するサーバマシンが負荷に耐え切れずにダウンする事態がたびたび発生して、多くの人々に多大な迷惑を発生させている。負荷増大でマシンがダウンするようなシステムは論外であるが、負荷増大でサービスが著しく低下するようではネットワークを生活に組み込んでいる人達の生活を乱すことになる。企業内ネットワークを構築する企業やインターネットプロバイダはこの要求増大に、高価なサーバ用計算機（専用並列計算機や大型機）を導入することにより対処している。既に導入したサーバマシンの処理能力が不足した場合には、さらに高価な計算機とリプレースする必要に迫られ、リプレース時にはソフトウェアおよびデータを移し換えるために、多大な作業時間と作業費用が発生する。

本研究においては、これらの状況を大幅に改善するために、サーバマシンをワークステーションクラスタによって構築し、その能力を構成マシンの台数によってスケーラブルに変更可能にするための基本ソフトウェア群を研究開発する。

1.2 期待される効果、成果

本研究開発の大局的目的は前記研究背景で示されるように、サーバ計算機の大幅なコストパフォーマンスの改善と処理能力にスケーラビリティを与えることである。本目標を達成するために下記4項目の研究開発内容を含む基本ソフトウェア群の再構築を行っている：

1. マルチプラットホーム間メモリベース通信ファシリティ

† 本研究は情報処理振興事業協会「独創的情報技術育成事業に係る開発」の一環として行われたものである。

メモリベース通信ファシリティ (MBCF: Memory-Based Communication Facilities) [1] [2]は、独創的先進的情報技術に係わる研究開発事業のテーマの一つとして平成 6 年度から約 4 年間研究開発を行った「汎用超並列オペレーティングシステムカーネル *SSS–CORE* の研究」[3] [4]において、平成 8 年度に松本が考案開発した新しい通信方式である。MBCF は特殊なハードウェアをまったく必要とせず、通常の LAN に接続可能なコンピュータに保護され仮想化された高速な通信および同期手段を提供する。現在事実上標準となっている TCP/IP プロトコルと比べて、オーバヘッドコストを二桁、レイテンシ（通信遅延）を一桁改善することができる。この MBCF 方式をクライアントとして使われる可能性のある様々なマシンやオペレーティングシステムに移植する。MBCF プロトコルによって通信することにより、サーバに掛かる通信のためのオーバヘッドを大幅に削減することができる。

2. スケーラブルサーバ内の仮想化された高性能メモリシステムおよびファイルシステムの実現
サーバマシンは大量のデータおよびプログラムをファイルシステムに保持管理する必要があり、ファイルシステムの性能がシステム全体の性能のボトルネックとなる可能性がある。このため、メモリ管理システムと統合された高性能ファイルシステムをサーバマシンに提供する。また、本研究において開発するサーバ用オペレーティングシステムが提供する共有メモリ機能と共有ファイルシステムを統合して、メモリ資源の有効活用とファイルシステムの高性能化を目指す。
3. 高速分散 Java 言語実行環境
本研究において、サーバ用スケーラブルオペレーティングシステムを独自開発するため、サーバ用アプリケーションプログラムを確保する必要がある。並列化が必要な負荷の重いアプリケーションは並列処理可能な形に変更する必要がある。しかし、負荷の軽いアプリケーションまですべて新しいオペレーティングシステムに移植したのでは、移植のコストが大きくなってしまう。このため、プラットフォーム依存性がなく、アプリケーション開発言語として注目されている Java 言語の実行環境を構築する。この Java 言語実行環境自体もシステムの高速通信機構やスケーラビリティの恩恵を享受できるように、分散並列拡張を行う。
4. スケーラブルサーバ用オペレーティングシステムの実現
独創的先進的情報技術に係わる研究開発事業のテーマ「汎用超並列オペレーティングシステムカーネル *SSS–CORE* の研究」として、平成 6 年度から約 4 年間研究開発を行った *SSS–CORE* オペレーティングシステムをスケーラブルサーバ用オペレーティングシステムとして機能強化と最新鋭ワークステーションへの移植を行う。機能強化の内容としては以下の項目が挙げられる。

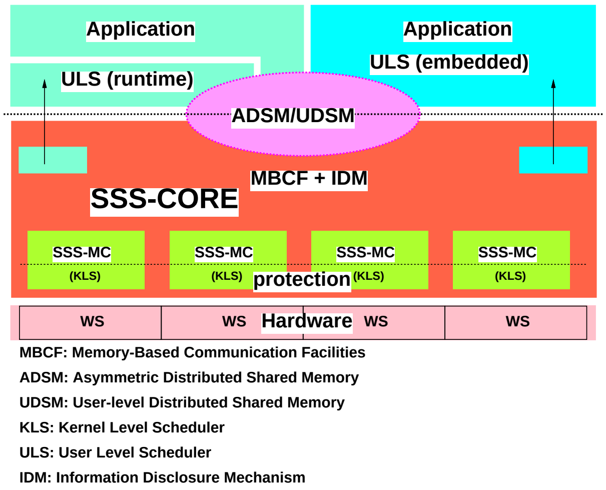

図 1 *SSS–CORE* の機能構成

- オペレータから単一イメージに見えるオペレーティングシステム
- 低オーバヘッドのメモリベース通信によって多数のクライアントに対応
- UDP/IP, TCP/IP, MBCF を高効率でサポート
- 分散並列拡張された Java をスケーラブルに処理
- サーバ OS としての安定性とセキュリティの確保
- 共有メモリ・共有ファイルシステムを活用した負荷分散

2 これまでの研究開発成果の概要

本研究開発は平成 10 年度から 3 年間の予定で始まり、平成 12 年度は最終年度の 3 年目である。本節ではこれまでの研究開発成果の概要について述べる。

2.1 Ultra 版 *SSS–CORE* の開発

独創的先進的情報技術に係わる研究開発事業のテーマ「汎用超並列オペレーティングシステムカーネル *SSS–CORE* の研究」において開発した *SSS–CORE* Ver.1.1 オペレーティングシステムは、Sun Microsystems 社の SPARCstation 20 またはこの互換機を Ethernet または Fast Ethernet で接続した環境で動作する。SPARCstation 20 は *SSS–CORE* の開発二年目である平成 7 年度に発売されたワークステーションであり、プロセッサは SuperSPARC-II である。現在、Sun Microsystems 社の最新鋭ワークステーションに搭載されているプロセッサは UltraSPARC-IIs であり、単体性能は SuperSPARC よりも数倍高速である。*SSS–CORE* をサーバ用オペレーティングシステムとして実用的な物とするためには、UltraSPARC プロセッサへの対応が必須である。UltraSPARC はユーザレベルのコードに関して、SuperSPARC と互換性があるものの、カーネルレベル（Supervisor mode）のアーキテクチャと命令体系は大幅に改良されている。このため、プロセッサ依存のコー

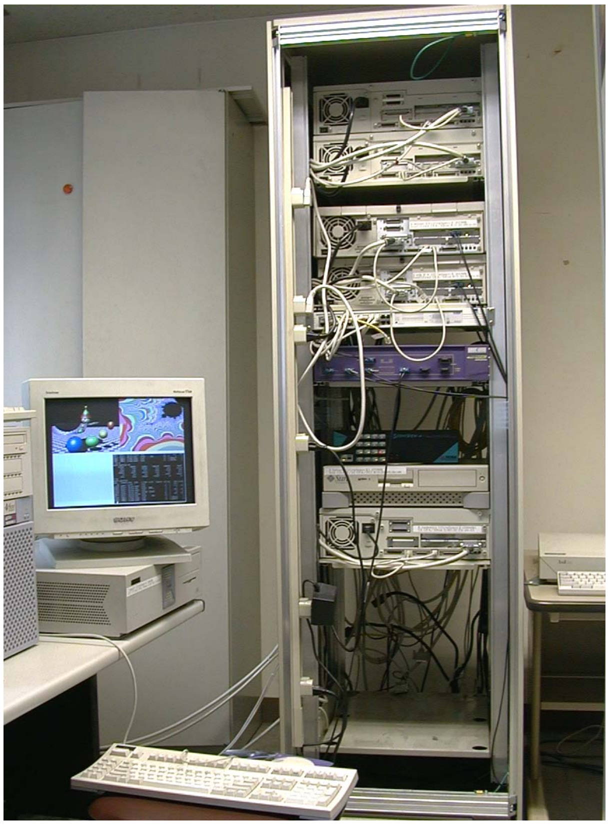

図 2　*SSS–CORE* **Ver.2.3 (Ultra 版)** の開発実行環境

ドを大幅に変更する必要があった。また、 UltraSPARC が使用された Ultra シリーズのワークステーションは Boot　ROM が Open　BOOT　ver.3.x になっており、 SPARCstation　20 の Open　BOOT　ver.2.x と大幅に異なっている。これらの事由により移植作業は非常に困難であったが、平成 10 年度に *SSS–CORE* のノード常駐核である SSS–MC　Ver.3.0 の作成に成功し、平成 11 年度には SBus 版 Ultra ワークステーションに *SSS–CORE* Ver.2.1 を移植し、平成 12 年度前半には PCI 版 Ultra ワークステーションに *SSS–CORE* への移植を完成させた[1]。 Ultra 版　*SSS–CORE*　は 64bit アドレスを採用した本格 64bit オペレーティングシステムであり、ユーザアプリケーションは従来の 32bit アドレス空間と広大な 64bit 空間を用途に応じて選択することができる。もちろん、従来の 32bit 空間のアプリケーションタスクを変更無しに動作させることも可能である。そして、現在は新規機能開発で先行していた SPARCstation　20 上の *SSS–CORE* Ver.1.2 と機能的に等価な *SSS–CORE* Ver.2.2 の開発を終了させ、負荷分散機能（マイグレーション機能）を持った *SSS–CORE* Ver.2.3 の開発を行っている。この負荷分散機能を含む *SSS–CORE* の新規追加機能に関しては次小節で述べる。

1　Sun Microsystems 社は I/O バスを SBus から PCI バスへ移行しようとしており、一部の製品を除いて最近のワークステーションには PCI バスを採用している。

図 3　*SSS–CORE* **Ver.1.2** の開発実行環境

2.2　*SSS–CORE* の機能強化

UltraSPARC プロセッサへの *SSS–CORE* 移植版の機能が SuperSPARC 版においついたため、従来は SuperSPARC 版をベースに機能強化していた *SSS–CORE* を統一版として機能強化可能になった。以下に機能強化の内容を列挙する。

1. IPsec プロトコル [5] [6]による高セキュリティ通信のサポート
2. プログラムからのシェル機能呼び出しを実装
3. タスクマイグレーション機能の実装
4. 情報開示機構の安定性強化
5. MBCF の No-flow-control 機能拡張
6. Gigabit Ethernet カードをサポート
7. Creator,Creator3D(ffb) タイプのフレームバッファをサポート
8. 外部委託ファイルシステムの高速化・機能強化
9. 実時間時計のサポート
10. Java 処理系の移植
11. X11 ウィンドウシステムの移植（現在進行中）
12. 最適化コンパイラ RCOP [7] [8]の機能強化

なお、マイグレーション機能および MBCF の　No-flow-control 機能拡張に伴い *SSS–CORE* のバージョンを Ultra 版 Ver.2.2（SuperSPARC 版 Ver.1.2）から Ver.2.3（Ver.1.3）に変更した。

第 1 項目に挙げた IPsec プロトコルによる高セキュリティ通信のサポートは *SSS–CORE* をサーバ用オペレーティングシステムとするのに必要不可欠な機能である。 IPSec はインターネット上の標準プロトコルである IP 上において暗号通信および通信内容の認証を行うための標準プロトコルであり、現在急速に広まりつつある。

サーバとしてクラスタシステムを利用してもらうためには、外部との通信においては速度や性能も重要であるが、セキュリティが最優先事項である。このため、当初の開発予定には入っていなかったが、IPsec 機能の開発を平成 11 年度に行った。Linux や FreeBSD 等の IPsec プログラムと通信試験を行い、暗号通信が行えることを確認した。

第 2 項目に挙げたプログラムからのシェル機能呼び出しの実装は UNIX の system() 関数と同様のプログラムからのシェル機能呼び出しを実現した。*SSS–CORE* の nikomon シェルはカーネル権限で動く特権タスクであり、システムの実装に関わる多くの機能を実現している。今回のシェル機能呼び出しの実装によって、これらの機能をユーザアプリケーションからも利用可能になった。

第 4 項目に挙げた情報開示機構の安定性強化は、ノードの突発的なダウンや再起動に *SSS–CORE* の情報開示機構 (IDM: Information Disclosure Mechanism) を対応可能にした。IDM は MBCF によって eager にお互いのノードの負荷情報や資源管理情報を交換することにより、ユーザタスクに低コストでこれらの情報を提供可能にしている。しかし、MBCF は通信保証を行うプロトコルであるため、MBCF を使用する場合には相手のノードが生きていることが保証されないと、無駄な再送等が起こってしまう。情報開示機構の情報はユーザタスクが負荷分散や負荷調整ためのヒントに使う情報であるため、情報開示に伴う通信は通信保証を必要としない。そこで、第 5 項目に挙げた MBCF の No-flow-control 機能拡張によって、通信に失敗しても再送が起こらないオプションを MBCF に導入して、無駄な再送が起こらないように変更した。また、クラスタの立ち上がっているノードの情報を、*SSS–CORE* のブートサーバからダイナミックに獲得可能に変更した。これによって、システムはクラスタの生きている構成ノードをダイナミックに検出可能になった。

第 5 項目に挙げた MBCF の No-flow-control 機能拡張は前記の情報開示機構の機構強化要求から行われた。アプリケーションの並列実行では通信保証（到着保証や順序保証）が非常に重要であるが、分散環境におけるヒントのような情報の情報交換には通信保証がかえって邪魔になる可能性がある。この No-flow-control 機能は TCP に対する UDP のようなものである。ただし、MBCF は非常に低オーバヘッドで実装されているため、No-flow-control にしても CPU オーバヘッドはほとんど減少しない。

第 7 項目に挙げた Creator, Creator3D(ffb) フレームバッファのサポートは現在 Sun Microsystems 社のワークステーションに標準で採用されている 24bit カラーのフレームバッファを *SSS–CORE* で使用可能にした[2]。

第 9 項目に挙げた実時間時計のサポートはワークステーションが持っている実時間時計を UNIX の gettimeofday 関数と同じスタイルで使用可能にした。*SSS–CORE* Ver.1.1/Ver.2.1 までは tick と固定周波数のタイマしかサポートしておらず、相対的な時間は認識可能だが、絶対的な時間はユーザが獲得できなかった[3]。ファイルシステムにはファイルの生成時刻等の時刻情報が不可欠であるため、ファイルシステム作成の下準備として実時間時計をサポートした。

第 10 項目に挙げた Java 処理系の移植の内容は Java 言語実行環境のベースとして PDS の Kaffe システムの移植である。既存の他の UNIX システムとの差異を解消するために Kaffe のコードを一部変更した。また、Kaffe の実装に依存した様々な問題に対応するために、いくつかの機能を *SSS–CORE* に実装した。シグナルについては、*SSS–CORE* で提供されていたシステムコールのみでは対応できなかったため新たに機能を追加した。これは、*SSS–CORE* の `MBCF_SIGNAL` を用いて実装した。Kaffe で使用されているシグナルのうち必須であるのは、SIGALRM、SIGIO、SIGCHLD であったが、SIGIO および SIGCHLD は Kaffe のコードを改変して対応し、残った SIGALRM にのみ対応することにした。*SSS–CORE* 用の Kaffe とは別に、シグナル用のデーモンが立ち上がっており、そのデーモンと `MBCF_SIGNAL` を用いた通信をすることによってシグナルを実現した。現在は SIGALRM のみの対応であるが、将来的には他のシグナルにもこのデーモンを利用して対応することが可能である。

第 11 項目に挙げた X11 ウィンドウシステムの移植は平成１０年度以前から少しずつ行われており、X サーバが *SSS–CORE* 上において立ち上がるところまで作業が進んでいる。平成１２年度には *SSS–CORE* 上のアプリケーションから X サーバを利用可能にして、X11 ウィンドウシステムによる GUI 環境を *SSS–CORE* に導入する予定である。

第 12 項目に挙げた最適化コンパイラ RCOP の機能強化は共有メモリ並列プログラムの最適化においてユーザの負担を大幅に軽減する。丹羽純平らが開発した最適化コンパイラ RCOP（Remote Communication OPtimizer）[7] [8]は松本が考案した ADSM (Asymmetric Distributed Shared Memory) [9] [10]と UDSM (User-level Distributed Shared Memory) [10]と呼ばれる二つの分散共有メモリ方式をサポートしている。これらの分散共有メモリの実現手法は共有メモリサポートハードウェアのない NUMA（Non-Uniform Memory Access）環境上であっても、最適化コンパイラのサポートによって効率良く分散共有メモリを実現する。平成１０年度までの RCOP は ADSM に関しては最適化を自動でサポートしていたが、UDSM 用コードの生成に関しては共有メモリの読み出しコードの最適化のためのユーザの編集作業が必要であった。平成１１年度の機能強化により、UDSM に関しても最適化を自動で行うことが可能になった。新しい RCOP を使用して、アプリケーションによる UDSM と ADSM の特性が調べられた。

2 Ver.1.1/Ver.2.1 以前の *SSS–CORE* は GX, TurboGX (cgsix) というタイプの 8bit カラーフレームバッファにしか対応していなかった。

3 SunOS エミュレータで外部 SunOS マシンに問い合わせれば獲得可能。

第 3 項目のタスクマイグレーション機能の実装、第 6 項目の Gigabit Ethernet カードのサポート、第 8 項目の外部委託ファイルシステムの高速化・機能強化は節を改めて後に詳述する。

2.3 Linux 版 MBCF プロトコルスタックの開発

平成 11 年度には、 MBCF プロトコルが使用可能なプラットフォームを拡大するために、 UltraSPARC プロセッサを搭載したワークステーションを対象に Linux 版 MBCF プロトコルスタックの開発を行った。 UltraSPARC プロセッサを搭載したワークステーション上の Linux を開発対象としたのは、ソースコードが入手可能であり、ネットワークハードウェアのデータシートがすでに手元にあったからである。開発作業の結果、 MBCF の基本プリミティブの中の遠隔書き込みおよび遠隔読み出しの実現に成功した。 100BASE-TX イーサネットで接続された 2 台の 300MHz の UltraSPARC プロセッサを搭載した Ultra 2 ワークステーションを使ってラウンドトリップタイムを測定した所、 40μsec 台の最小値を記録した。測定を行ったマシンが異なるため、単純な比較はできないが、この値は SS20 上の *SSS–CORE* の性能を上回るものである。また、 Linux 版 MBCF は Linux のメモリスワップアウトに対応するコードになっており、 MBCF のスワップアウト対応の最初のコードである[4]。今回の UltraSPARC 用 Linux 上の MBCF プロトコルスタックの開発のノウハウをベースにして、 Pentium プロセッサ等の x86 プロセッサを持つ IBM-PC 互換機を対象とする Linux への MBCF の移植を進めている。

3 タスクマイグレーション機能

SSS–CORE は構成マシンを追加することによってスケーラブルに性能が改善できることが最大の特徴である。しかし、一部のマシンに負荷が偏った状態を解消する能力がないと、システムのスケーラビリティが有効に活用できない。データベースサーバやウェブサーバにように一つ一つのタスクの処理量が小さく、タスクの発生消滅頻度が高いアプリケーションでは、タスク起動時のマシンへの振り分けのみで負荷分散が可能である。しかし、大規模シミュレーションのような応用では実行中のタスクを他のノードに移送 (マイグレーション) する能力がないと負荷の均衡を図ることができない。 *SSS–CORE* は汎用スケーラブルオペレーティングシステムを目指すため、タスクマイグレーション機能を実装した。

SSS-CORE の最大の特徴である MBCF はタスクがノードを含めて仮想化されており、 MBCF によって通信同期を行っているタスクはすべてマイグレーション可能である。ただし、 *SSS–CORE* のノード常駐部である SSS–MC が提供する同期機構 (queue, semaphore, event) は同一ノードに存在することを前提に提供しているため、これらの機能を利用するタスクはマイグレーションできない。しかし、これらの同期機構は MBCF の完成までに使用するために実装されたにすぎず、 MBCF よりも効率が悪く使い勝手もよくない。実際に、最近開発された遠隔実行サーバやマイグレーションサーバはすべて通信同期が MBCF で記述されている。このため、 SSS–MC の同期プリミティブをサポートできないことは将来的なデメリットになることはない。

現在、 *SSS–CORE* はファイルシステムとして外部委託ファイルシステムを使用している。これは外部にある UNIX (SunOS,Solaris) に TCP/IP 通信でシステムコールの実行を依頼し、ファイルアクセスを代わりに実現してもらう方式である。 *SSS–CORE* 内のタスクに対応したプロセスが UNIX 側に生成されて、同一タスクから発生した UNIX へのシステムコールは同一プロセスで処理される。 *SSS–CORE* 内のユーザタスクがマイグレーションされた場合には、対応する UNIX プロセスとタスクとの対応を移送先のタスクに切替えてマイグレーションに対処する。この方式により、外部委託ファイルシステムを使用中のタスクであっても、ノードを移動して実行を続けることが可能である。

マイグレーションされるタスクの新しいノードにおける再起動に平成 10 年度に実装した遠隔実行機構 [11]を利用することにより、移送されたタスクは標準入出力を継続的に使用可能である。マイグレーションされたタスクは遠隔実行機構の標準出力機構を使って、標準出力を起動された端末画面またはコンソール画面に出力する。標準入力も同様に起動された端末キーボードまたはコンソールキーボードから移送後も入力を続けることが可能である。

SSS–CORE は新しいスケジューリング方式として「自由市場原理に基づくスケジューリング方式 (FMM 方式) 」 [12] [11]を採用予定である。この方式の下ではアプリケーションタスクが自分の判断でマイグレーションのためのシステムコールを発行し、自分が指定したノードへの移送を要求する。今回実装したマイグレーション機能は FMM 方式に対応しており、マイグレーション要求用のシステムコールが実装された。ただし、今回実装したマイグレーション機構にはタスク中断に関して何も制約がないため、システム側からの強制的なマイグレーションにも対応できる。 FMM 方式のスケジューリングを完成させるため、前述のように情報開示機構を整備した。今後は、プロセッサ割り当てのエイジングアルゴリズムに飽和資源への使用要求に対するペナルティを加えて FMM 方式のスケジューリングが完成する。

4 Gigabit Ethernet による MBCF

4.1 Gigabit Ethernet のサポート

SuperSPARC 版の *SSS–CORE* は 1997 年より Fast Ethernet をサポートしており、 1997 年当時は非常に高いバンド幅と低いレイテンシを示していた。 1998 年から Gigabit Ethernet の普及が始まり、 Fast Ethernet による MBCF の性能は見劣りがするようになってきた。早い時期に Gigabit Ethernet のサポートを行おうとしたが、 Gigabit Ethernet カードに使われている LSI の資料を入手することが困難でサポートできずにいた。 1999

4 MBCF 方式はスワップアウトに元々対応可能であるが、現段階の SSS–CORE ではメモリスワップアウトを行っていない

年の暮れに Sun Microsystems 社から Sun GigabitEthernet 2.0 に使用されている PCI 用 LSI [13]と SBus 用 LSI [14]の資料をいただいて、サポートすることが可能になった。イーサネットコントローラとメインメモリの間のインタフェースは Fast Ethernet のものと大差がなかったが、初期設定用のレジスタの構成が大幅に異なっており、 LSI の資料の記述も不十分でデバイスドライバの実装にはかなりの労力(試行錯誤)を要した。

4.2 Gigabit Ethernet による MBCF の性能

今回新たに実装した Gigabit Ethernet ドライバを使用した MBCF の MBCF_WRITE の性能測定結果を本小節で示す。測定条件は Sun Microsystems 社 Ultra 60 ワークステーション (UltraSPARC-IIs 450MHz) に Sun GigabitEthernet 2.0 Adapter (1000BASE-SX) を PCI 64bit 66MHz スロットに装着したマシン 2 台を光ファイバケーブルによって直結したもので測定を行った。 MBCF の測定に使用したオペレーティングシステムは *SSS–CORE* Ver.2.3 である。なお、参考データとして同一ハードウェア条件で Solaris2.6 による TCP/IP の性能測定を行った。 TCP/IP は標準のソケットインタフェースを使用し、ソケットバッファを 64Kbyte に設定し、遅延測定時には TCP_NODELAY オプションを付加し、ピークバンド測定時には同オプションを付加しなかった。さらに、参考のために Fast Ethernet (100BASE-TX) を使った SPARCstation 20(SuperSPARC 85MHz)における MBCF の性能測定値を掲げておく。

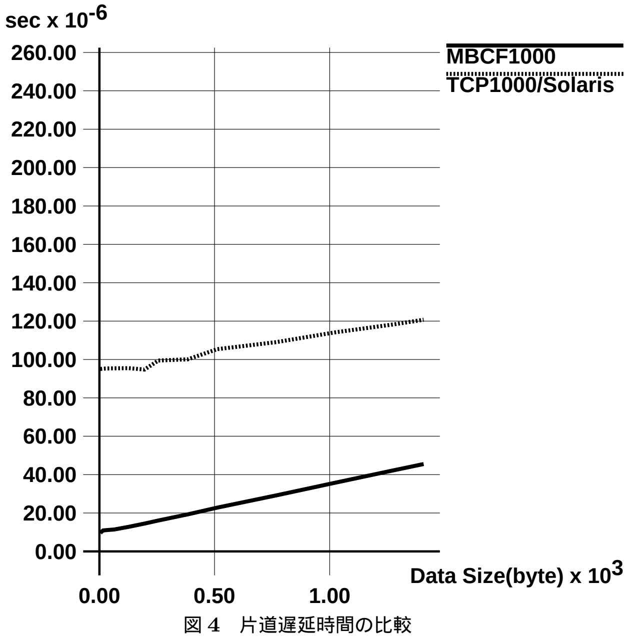

図 4　片道遅延時間の比較

表 1に片道遅延時間の測定値と図 4に片道遅延時間の比較を示す。測定は往復遅延時間を測定し、その値を半分にしたものである。 4byte のデータ転送時に Gigabit Ethernet を使用した MBCF では **9.6**μsec で異なるノードのアプリケーション間で通信が可能である。この値は Solaris2.6 上の TCP/IP の約 10 分の 1 である。

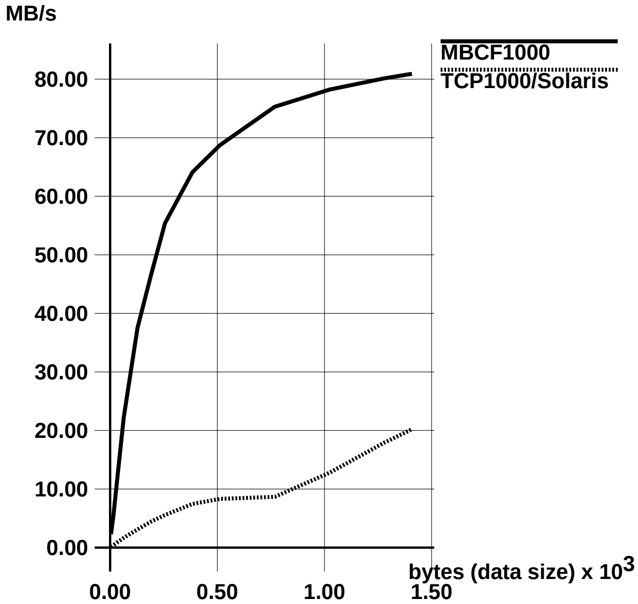

図 5　ピークバンド幅の比較

表 2にピークバンド幅の測定値と図 5にピークバンド幅の比較を示す。なお、 TCP/IP は転送データサイズ(send() 関数で一度に渡すデータサイズ)をイーサネットのパケットサイズを越えて大きくした場合に、より大きなピークバンド幅を示し、データサイズを 64Kbyte にした場合に 53.06Mbyte/sec の性能を示した。しかし、 MBCF の 1408 バイトデータで示した **80.92**Mbyte/sec には及ばない。片道遅延時間の結果からも判るように、送信と受信のオーバヘッドタイムの和は 9.6μsec(HW によるパケット転送時間を含む)以下である。また、別の測定から 1408byte 転送時に転送用兼再送用バッファへのコピー時間を含む送信オーバヘッドは 3.2μsec であり、送信のデータコピーなしのオーバヘッドは 1.6μsec である。受信時のオーバヘッドも高々この倍程度と考えられる(上限は $8.0 = 9.6 - 1.6$)。 Gigabit Ethernet では 1408byte パケット時に転送路において物理的に 11.26μsec 以上の時間がかかる。よって MBCF はソフトウェア的なオーバヘッドは十分に小さく、他にボトルネックがなければ Gigabit Ethernet の理論的転送限界(125Mbyte/sec)付近まで性能が出せるはずである。しかし、今回の測定では 80Mbyte/sec 強の値で頭打ちとなっている。図 5からも判るように、何らかのボトルネックが 80Mbyte/sec 強のところに存在していると考えられる。以上の測定と考察からのそのボトルネックは Ultra 60 のハードウェア側にあるものと推定される。

5 外部委託ファイルシステムの高速化・強化

効率の高いネットワークファイルシステム(NFS)を作るために本研究開発において「スケーラブルサーバ内の仮想化された高性能メモリシステムおよびファイル

表 1　**MBCF/1000BASE-SX** の片道遅延時間 (**μsec**)

data size (byte)	4	16	64	256	1024
MBCF 1000BASE-SX	**9.6**	11.0	11.5	16.2	35.9
Solaris TCP/IP(1000BASE-SX)	**95.08**	95.22	95.39	99.45	114.15
MBCF 100BASE-TX	**24.5**	27.5	34	60.5	172

表 2　**MBCF/1000BASE-SX** のピークバンド幅 (**Mbyte/sec**)

data size (byte)	4	16	64	256	1024	1408
MBCF/1000BASE-SX	2.29	5.67	22.30	55.41	78.22	**80.92**
Solaris TCP/IP(1000BASE-SX)	0.09	0.43	1.67	5.56	12.79	**20.21**
MBCF/100BASE-TX	0.34	1.27	4.82	9.63	11.64	**11.93**

システムの実現」が四つの研究開発項目の一つとなっている。しかし、実用的なネットワークファイルシステムを作るためには、その前段階として準備すべきものが多い。前述の実時間時計のサポートもその一つであり、現在作業を行っている標準 NFS との接続機能の開発もその一つであり、ノードローカルな二次記憶装置のサポートもその一つである。しかし、二次記憶装置、周辺装置、ファイルシステムといったものが使用不可能な状況では、意味のあるアプリケーションを動作させることができない。この窮状を救うために、低開発コストで他のオペレーティングシステムが有する入出力機能を利用する手段として外部委託ファイルシステム（つまり TCP/IP を介した外部委託による SunOS エミュレーション機能）が松本によって考案された。この機能によって SunOS のプログラムやライブラリを *SSS–CORE* 上で使用可能になるという大きなメリットがある。けれども、*SSS–CORE* のファイルシステムとしては一種の「継なぎ」的な機能である。そこで、ファイルシステムとしての性能に関するチューニングがなされていなかった。自前のネットワークファイルシステムの完成にはまだ時間がかかるため、平成１１年度にはネットワークファイルシステム作成の下準備を進めると同時に、外部委託ファイルシステムを大幅に高速化した。

表 3に改良前と改良後のファイル転送性能を示す。実験条件は以下の通りである。*SSS–CORE* 側ワークステーションとして SPARCstation 20 (85 MHz SuperSPARC × 1) を使用しファイルサーバ側は SPARCstation 20 (60 MHz SuperSPARC × 1) を SunOS4.1.4 で使用した。これらのマシンを 3Com SuperStack 1000 (10BASE-T switching HUB) によって 10BASE-T イーサネット接続を行った。

表の結果から判るように、ファイル読み出しおよびファイル書き込みの性能が数十倍以上に改善した。改善前の値が異常に悪いことが大幅性能改善の理由であり、これは外部委託ファイルシステムの実装が悪かったというよりも、*SSS–CORE* のオリジナル TCP/IP が TCP/IP 本来の機能の一部（ウィンドウによる送信パケットの先出し）を未実装であったことが主な原因である。このため、外部委託ファイルシステムの性能チューニングとともに、*SSS–CORE* のオリジナル TCP/IP の改良作業も行った。

今回の改良によってファイル転送性能が同一条件の UNIX マシンの NFS 並に向上した。さらに、ファイルシステムとしての使い勝手を向上させるために、*SSS–CORE* 側からファイルシステムを提供している SunOS マシンのシェル機能を呼び出す system() 関数呼び出しシステムコールを開発した。UNIX の popen() と同様に実行結果である標準出力を system() 関数呼び出しを実行した *SSS–CORE* 上のプログラムで受け取ることができる。

6　おわりに

分散サーバ環境の核となるスケーラブルオペレーティングシステムの開発は「汎用超並列オペレーティングシステムカーネル *SSS–CORE* の研究」に引き続いて行われており、マイグレーション機能、IPsec 通信機能、情報開示機構の整備、MBCF の性能強化と機能増強、外部委託ファイルシステムの性能強化といった順調な進展を見せた。マイグレーション機能の実現により、情報開示機構と組み合わせて *SSS–CORE* 上で自動負荷分散を実現可能になった。IPsec 通信機能の実装によってサーバオペレーティングシステムとして通信におけるセキュリティ強化が可能になった。*SSS–CORE* を搭載したクラスタシステムは外部システムと暗号通信および通信内容の認証が可能になる。最新鋭プロセッサを使ったワークステーションへ *SSS–CORE* を移植する作業は、プロセッサのカーネルアーキテクチャが大幅に変更されているため、当初からの予想通り困難な作業であったが、汎用スケーラブルオペレーティングシステムとして移植が完了した。現在では、Ultra 版 *SSS–CORE* Ver.2.3 が Sun Microsystems 社の Ultra 2 ワークステーションおよび Ultra 60 ワークステーションで動作している。Ultra 版 *SSS–CORE* Ver.2.3 はカーネルが 64bit アドレスを使用し、ユーザアプリケーションも 64bit アドレス空間が使用可能な真の 64bit オペレーティングシステムの一つである。Gigabit Ethernet カードをサポートして、MBCF 通信の片道レイテンシが **9.6**μsec、スループット **80**Mbyte/sec を Ultra 60 ワークステーションにおいて記録した。MBCF の CPU オーバヘッドが極めて小さいことが改めて確認され、I/O バス周りの転送能

表 **3**　改良前と改良後の外部委託ファイルシステムの性能

file size (Kbyte)	16	32	64	128	256	512	1024
改良前 read (Kbyte/s)	4.5	4.7	4.9	5.0	5.0	5.1	5.1
改良後 read (Kbyte/s)	163.1	204.1	233.3	250.6	261.2	266.6	271.0
改良前 write (Kbyte/s)	1.7	1.7	1.7	1.7	1.7	1.7	1.7
改良後 write (Kbyte/s)	267.7	355.8	434.3	498.4	535.0	550.7	562.5
改良前 read/write(Kbyte/s)	2.4	2.4	2.5	2.5	2.5	2.6	2.6
改良後 read/write(Kbyte/s)	199.7	255.5	303.5	332.7	345.0	357.5	365.5

力が高いプラットフォームでは十分に Gigabit Ethernet の能力を使い切れることが示された。

ネットワークファイルシステムの研究開発は下準備となる開発作業を進行中である。自前のネットワークファイルシステム完成までの「継なぎ」的な意味で外部委託ファイルシステムの性能を大幅向上し実用可能なレベルにした。

Java 言語実行環境の研究開発では PDS の Kaffe システムを *SSS–CORE* 上に移植し、 *SSS–CORE* 上で簡単な Java プログラムが実行可能となった。 X11 ウィンドウシステムの *SSS–CORE* への移植があと一歩で完成するところまでこぎつけているので、平成１２年度中には GUI を伴う Java アプリケーションが実行できる予定である。

MBCF の *SSS–CORE* 以外のシステムへの移植は、Linux の UltraSPARC 版を対象に開発作業を行い MBCF の基本機能である遠隔書き込みと遠隔読み出しの実現に成功した。現在は、 PC 互換機上の Linux を対象にして MBCF の移植を行っている。

謝　　辞

ワークステーションに使用されている LSI のデータシートならびにユーザマニュアルの手配には、サンマイクロシステムズ株式会社の方々に協力していただいた。また、 *SSS–CORE* の開発環境の整備には多くの平木研究室メンバの支援を受けた。

References

[1] 松本 尚, 平木 敬: 汎用超並列オペレーティングシステム SSS–CORE のメモリベース通信機能. 第 53 回情報処理学会全国大会講演論文集, 第 1 分冊, pp.37–38 (September 1996).

[2] Matsumoto, T. and Hiraki, K.: MBCF:A Protected and Virtualized High-Speed User-Level Memory-Based Communication Facility. In *Proc. of the 1998 ACM Int. Conf. on Supercomputing*, pp.259–266 (July 1998).

[3] 松本 尚, 平木 敬: 汎用並列オペレーティングシステム SSS–CORE の資源管理方式. 日本ソフトウェア科学会第 11 回大会論文集, pp.13–16 (October 1994).

[4] 松本 尚, 渦原 茂, 竹岡 尚三, 平木 敬: 汎用超並列オペレーティングシステムカーネル *SSS–CORE*. 第 17 回技術発表会論文集, 情報処理振興事業協会, pp.175–188 (October 1998).

[5] S. Kent and R. Atkinson: IP Authentication Header. RFC 2402, (November 1998).

[6] S. Kent and R. Atkinson: IP Encapsulating Security Payload (ESP). RFC 2406, (November 1998).

[7] 丹羽 純平, 稲垣 達氏, 松本 尚, 平木 敬: 非対称分散共有メモリ上におけるコンパイル技法. ハイパフォーマンスコンピューティング研究会報告 No.67–21, 情報処理学会. pp.121–126 (August 1997).

[8] 丹羽 純平, 稲垣 達氏, 松本 尚, 平木 敬: 汎用超並列オペレーティングシステム SSS–CORE 上の非対称分散共有メモリにおける最適化コンパイル技法. コンピュータソフトウェア, Vol.15, No.3, pp.54–58 (May 1998).

[9] 松本, 駒嵐, 渦原, 平木: メモリベース通信による非対称分散共有メモリ. コンピュータシステムシンポジウム論文集, 情報処理学会 pp.37–44 (November 1996).

[10] Matsumoto, T., Niwa, J., and Hiraki, K.: Compiler-Assisted Distributed Shared Memory Schemes Using Memory-Based Communication Facilities. In *Proc. of The International Confference on Parallel and Distributed Processing Techniques and Applications* (PDPTA-98), Vol.2, pp.875–882 (July 1998).

[11] 松本 尚, 渦原 茂, 竹岡 尚三, 平木 敬: スケーラブルな分散サーバ環境の研究 — *SSS–CORE* の実用化に向けて —. 第 18 回技術発表会論文集, 情報処理振興事業協会, pp.217–225 (October 1999).

[12] 松本 尚, 平木 敬: 自由市場原理に基づくスケジューリング方式. 信技報, Vol.99, No.251, CPSY 99–55, pp.63–70 (August 1999).

[13] Sun Microsystems,Inc.: Specification for GEM Gigabit Ethernet ASIC. Spec Number:950-3239-01, Sun Microsystems,Inc. (August 1998).

[14] Sun Microsystems,Inc.: Specification for Sbus GEM Gigabit Ethernet ASIC. Spec Number:950-3284-01, Sun Microsystems,Inc. (July 1998).

図 **6**　*SSS–CORE* マスコットキャラクタ「マネキッコ」