システムソフトウェアとオペレーティング・システム 86−17
(2001. 3. 6)

分散オペレーティングシステムSolelcにおけるネットワーク機構の構築

藤本 堅太 † 芝 公仁 † 大久保 英嗣 ††

† 立命館大学大学院理工学研究科 †† 立命館大学理工学部

我々は, 現在, 分散オペレーティングシステム Solelc を開発している. Solelc では, 単一のオペレーティングシステムでネットワーク上の複数の計算機を管理する. Solelc の通信機構には, Solelc が管理している計算機が協調動作を行うための機構と, 管理外の計算機上で動作するプロセスと通信するための機構の 2 つが必要となる. 本稿では, これらを実現する Solelc のネットワーク機構の構成について述べる. また, 本機構の評価についても述べる.

キーワード: 分散オペレーティングシステム, ネットワーク

The Design and Implementation of Network Function on Solelc Distributed Operating System

Kenta Fujimoto † Masahito Shiba † Eiji Okubo ††

†Graduate School of Science and Engineering, Ritsumeikan University
††Faculty of Science and Engineering, Ritsumeikan University

We have been developing a distributed operating system Solelc. In Solelc, plural computers connected via a network are managed by a single operating system. With cooperative works of the abstraction mechanism on each computer, the location transparency of computer resources can be realized. Solelc has two mechanisms for communications. One is for cooperative works of the abstraction mechanism. The other is to communicate with machines which are not controlled by Solelc. In this paper, the structure and performance evaluation of network function are described.

Key words: Distributed operating system, Network

1 はじめに

我々は, 現在, 分散オペレーティングシステム (以下 OS と記す) Solelc の開発を行っている. Solelc では, 従来の分散 OS とは異なり, 単一の OS で同時に複数の計算機を管理することが可能である. 計算機毎にそれぞれ OS を動作させるのではなく, 1 つの OS ですべての計算機を管理することによって, システム全体を考慮した資源管理を行うことが可能となる [1].

Solelc における通信には, Solelc が管理している計算機間の通信と, Solelc が管理していない計算機との通信の 2 つがある. Solelc が管理している計算機間の通信は, 計算機間でシステム管理情報を受け渡しするときに行われる. この形態の通信は, システム管理情報を通信するため, 高信頼性を実現している. さらに, ネットワーク負荷を軽減するために, 効率化を図っている. Solelc が管理していない計算機との通信は, 当該計算機上で動作するプロセスと Solelc 上で動作するプロセスが通信するときに行われる. この形態の通信では, 他の OS と通信するため, 従来から広く用いられている TCP/IP をサポートしている. また, カーネルが通信を管理することにより, 位置透過にメッセージを受け渡しすることが可能となっている.

以下, 本稿では, 2 章で Solelc の概要について述べ, 3 章でネットワーク機構の概要, 4 章で抽象化層の間の通信機構, 5 章で他の OS との通信機構について述べる. 次に 6 章で評価を行い, 最後に 7 章で本稿のまとめと今後の予定について述べる.

2 Solelc の構成

Solelc の構成を図 1 に示す. Solelc では, OS が抽象化層とカーネルの 2 つの層に階層化されている. 下位の抽象化層は, すべての計算機に配置されており, 各計算機上の資源を抽象化する役割を持つ. 抽象化層は, 他の計算機上の抽象化層と協調動作し, 位置透過な資源管理を可能とする環境を実現する. 上位の層は 1 つのカーネルから構成され, システム全体の資源を管理する役割を持つ. カーネルは, 抽象化層が提供する環境上で動作するため, 任意の計算機上ですべての計算機の資源を管理することができる. また, カーネルは, プロセスの実行環境を構築する. カーネルは, すべての計算機上のプロセスにサービスを提供できるため, プロセスも

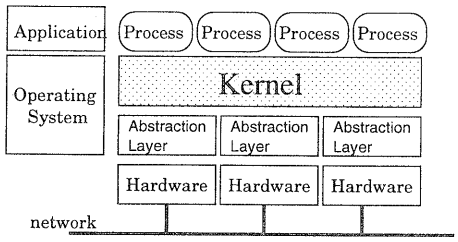

図 1 システム構成

位置透過に動作することが可能である. 1 つのカーネルが, すべての資源を管理し, すべてのプロセスにサービスを提供するため, プロセスは任意の計算機上ですべての資源を利用することができる.

2.1 抽象化層

抽象化層は, 各計算機で 1 つずつ動作し, カーネルが動作するための環境を構築する. 抽象化層は, ハードウェアを直接操作する機能と, 資源の位置を管理する機能の 2 つの機能を持つ.

抽象化層は, CPU やメモリの管理を行う機能を持つ. CPU は, スレッドとして抽象化され管理される. カーネルやプロセスは, スレッドを用いることによって, 1 つの CPU で複数の処理を並行して実行できる. また, 抽象化層は, すべての計算機から共有される単一の仮想アドレス空間を構築し, これをカーネルやプロセスに提供する. 各計算機に接続された周辺デバイスも抽象化層によって操作される. 抽象化層は, デバイスドライバを持ちデバイスを直接操作する機能を持つ. また, 割込みハンドラも抽象化層内に存在し, 割込み発生時には抽象化層がこれを取得し割込みの種類に応じた処理を行う.

また, 抽象化層は, 他の計算機上で動作する抽象化層と協調することによって, 資源の位置を管理する. 抽象化層は, システムで使用するすべての資源にシステム全体で一意な識別子を付与する. 資源とは, Solelc で管理される計算機とそれらが持つ CPU やメモリ, NIC などの周辺デバイスである. 抽象化層は, 識別子を用いて, どの NIC がどの計算機に接続されているかといった資源の位置情報を管理する.

2.2 カーネル

カーネルは, システム全体で1つであり, 抽象化層が提供する機能を用いて, システム全体の資源を管理する. 抽象化層が提供する機能は位置透過に使用することができるため, カーネルは任意の計算機上で動作可能である. また, カーネル内のすべての機能が同一の計算機上で動作する必要はなく, カーネルを構成する各モジュールをそれぞれ異なる計算機上で動作させることも可能である. この場合も, 抽象化層が実現する位置透過性によって, 各モジュールは他のモジュールと同一の計算機上で動作しているかのように処理を行うことができる.

カーネルは, 従来のOSが実現しているものと同様の機能を実現する. すなわち, プロセスの実行環境を構築し, 各プロセスにシステムの資源を適切に分配する. また, プロセスに対して, プロセス間通信などのサービスを提供する. 計算機毎にカーネルを配置し, これらを協調動作させることによって資源管理やサービスを実現する場合, 複雑な処理が必要になりカーネルの構造が複雑になるといった問題がある. Solelcでは, 1つのカーネルですべての計算機を管理することによって, カーネルの構造を単純化している [1].

ユーザプロセス同士のデータ送受信は, 各計算機の抽象化層を介することで実現される. このシステム構成に従い, カーネル上に位置透過性を持つ通信機構を実現することで, すべてのユーザプロセスは, 位置透過に他のユーザプロセスと通信することが可能となる.

3 ネットワーク機構の概要

Solelcでは, 抽象化層間でシステム管理情報を通信するときは, 内部通信機構を使用し, Solelcが管理していない計算機上のプロセスと通信する時は, 外部通信機構を使用する. Solelcにおいて, 内部通信とは, 抽象化層間の通信を指し, 外部通信とは, 管理外の計算機との通信を指す. 図2に, それぞれの機構を示す.

内部通信は, Solelcが管理する抽象化層間の通信にのみ使用する. 外部通信は, Solelc上で動作するプロセスとSolelcの管理外の計算機上で動作するプロセスの間での通信に使用する. 管理外の計算機上のプロセスは, Solelc上で動作するプロセスと通信するとき, Solelc上で動作するプロセスの位置

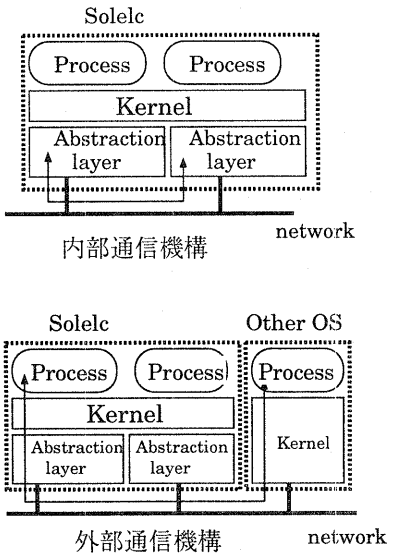

図2 Solelcにおけるネットワーク機構

を意識する必要はない. これは, 位置透過に動作するカーネル上に実現している外部通信機構が通信を管理していることによる.

4 内部通信機構

4.1 概要

Solelcでは, 抽象化層間でシステム管理情報を通信するとき, 内部通信機構を使用する. 内部通信機構は, システム管理情報を通信するため, 高い信頼性が必要となる. また, 従来のOSと異なり, 抽象化層間でシステム管理情報を通信するため, 通信オーバヘッドが大きい. このため, 内部通信機構では, 効率的な通信処理を行う必要がある.

以下, 内部通信機構が持つ機能, および信頼性を確保するために使用する再送機構, 効率化を行うために使用する処理選択機構について述べる.

4.2 機能

抽象化層間で通信するシステム管理情報には, スレッド情報, 割り込み情報, ドライバテーブルへのドライバ情報の登録と削除, ページの移送, ホストの登録と削除情報などがある. Solelcでは, システム管理情報の通信を抽象化層間で行うために以下の機能とデータを実現している.

- ハードウェアアドレスの管理
  Solelcでは, Solelc内に接続されているNICのハードウェアアドレスを管理するために, ハードウェアアドレステーブルを持つ. ハードウェアアドレステーブルは, 計算機IDとハードウェアアドレスから構成される.

- 計算機IDの管理
  Solelcでは, 各計算機毎に計算機IDを設定する. 計算機IDは, 抽象化層初期化時に設定される. 2台の計算機(A, B)で動作しているSolelcに新たに計算機(C)を1台追加するときの処理の流れを図3に示す.

  (1) 抽象化層Cは, ブロードキャストを用いて, Solelcに対して, Solelcへの参加を要求する.

  (2) 登録要求は, Solelcが管理する抽象化層の中で計算機IDが最も小さい抽象化層が受信する. それ以外の抽象化層は, パケットを破棄する. 要求を受信した抽象化層Aは, 計算機IDを設定し, Solelcが管理するすべての抽象化層へ新たな計算機IDテーブルを通知する.

  (3) 登録通知を受信したそれぞれの抽象化層は, 計算機IDとハードウェアアドレスをハードウェアアドレステーブルに登録し, 正常に登録が行われたことを抽象化層Aに通知する.

  (4) すべての抽象化層から, 正常登録通知を受信した抽象化層Aは, すべての抽象化層へ新たな抽象化層の参加完了を通知する.

- 信頼性の確保
  Solelcでは, 抽象化層間の通信を保証するために再送機構を実現している. 再送機構は, 受信側の抽象化層が, データ受信時に正確にデータが届いたことを, 送信側の抽象化層に通知する. 送信側では, 送信時に, 送信すべきデータのデータ番号, データへのポインタ, 再送時間を持つリストを作成する. 受信側から, 指定時間内に正常到着通知がなければ, データを再送する. これにより, 内部通信方式では, データが正常に届くことを保証している.

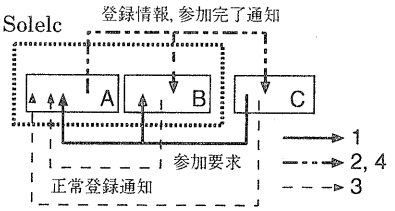

図3 計算機ID登録処理手順

4.3 処理手順

Solelcでは, 内部通信における効率化を図るため, 割込みコンテキスト処理方式とスレッド委託処理方式を実現している. 本節では, それぞれの方式の処理手順を述べる.

4.3.1 割込みコンテキスト処理方式

割込みコンテキスト処理方式は, 受信したパケットの処理をすべて割込みコンテキスト内で行う. 割込みコンテキスト方式は, スレッド委託方式に比べ, 抽象化層で行う処理が完了する時間が小さい. これは, 受信したパケットの処理を割込み処理内で行うため, スレッドの切り替えが行われないためである. 本方式は, ページの移送やスレッドの移送など, 直ちに処理を行う必要があるものに対して有効である. 以下に, 割込みコンテキスト処理方式の処理手順を述べる.

(1) 送信側の抽象化層は, パケット処理を割込みコンテキスト内で処理することをヘッダに設定し, 受信側の抽象化層へパケットを送信する.

(2) パケットを受信した抽象化層は, 割込みコンテキスト処理方式で処理を行うパケットであることを判別し, その処理を行う.

(3) パケットの処理が終了すると, 割込みコンテキスト処理は終了する.

4.3.2 スレッド委託処理方式

スレッド委託処理方式は, 受信したパケットの処理をスレッドに委託して, パケット受信の割込みを終了させる. 次に, 処理を依頼されたスレッドは, パケットの処理を行う. 図4は, 本方式の処理の

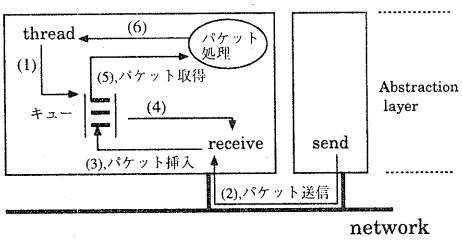

図 4　スレッド委託処理方式

流れを示したものである. 本方式は, スレッドが持つレジスタのセーブやロードが発生するため, 割込みコンテキスト処理方式に比べ, オーバヘッドが大きい. しかし, 本方式は, パケットの処理をスレッドに委託するため, 割り込み禁止時間は短い. したがって, 本方式は, パケットの処理に時間がかかるものや I/O 待ちを伴う処理など, 直ちに処理を行う必要がないものに対して有効である. 以下に, スレッド委託処理方式の処理手順を述べる.

(1) 抽象化層は, 初期化時に, パケットキュー監視スレッドを生成する.

(2) 送信側の抽象化層は, スレッド委託処理方式で処理することを設定し, 受信側の抽象化層へパケットを送信する.

(3) データを受信した抽象化層は, 受信したパケットがスレッド委託処理方式で処理すべきパケットであれば, 内部通信機構が持つパケットキューにパケットを挿入する.

(4) 抽象化層は, パケットキュー監視スレッドを実行可能状態に遷移させ, 割込み処理を終了する.

(5) 実行可能状態となったパケットキュー監視スレッドは, パケットの処理を行う.

(6) パケットキュー監視スレッドは, パケットの処理が終了すると, 次のパケットの到着を待つ.

5　外部通信機構

5.1　概要

Solelc では, Solelc が管理する計算機との通信の他に, Solelc が管理していない計算機上で動作するプロセスとの通信を行う必要がある. このため, Solelc では, 管理外の計算機上で動作するプロセス

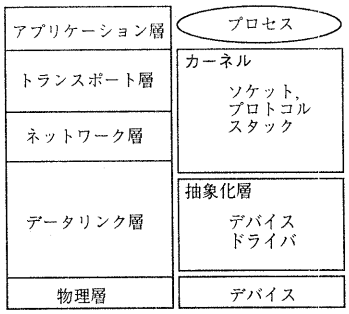

図 5　機構の位置づけ

と通信を行うために, 外部通信機構を実現している. 外部通信機構では, カーネル内に通信管理機能を実現し, 抽象化層内にデバイス管理機能を実現している. 図 5 に, 各機能の位置づけを示す. カーネルは, 通信を管理するために以下の機能を持つ.

- ソケットインタフェース
- TCP/IP 等のプロトコルスタック

一方, 抽象化層は, デバイスの管理を行うために, 以下の機能を持つ.

- デバイスが発生させた割込み処理を行うデバイス処理機構
- 受信したパケットの処理を行うパケット処理機構

以下, 外部通信機構が持つ機能, および送信処理, 受信処理, 読出し処理時の処理手順について述べる.

5.2　機能

外部通信機構では, 位置透過に動作するカーネルが通信を管理する. このため, カーネル上で動作するプロセスは, カーネルが提供する通信機構を位置透過に利用することができる. カーネルは, TCP/IP 等のプロトコルスタックを実現し, メッセージを管理する. Solelc では, アプリケーションから見たデータセグメントをメッセージと呼ぶ. また, データリンク層の MTU 規格によりメッセージを分割したものをパケットと呼ぶ. また, ユーザに対し, ソケットインタフェースを提供し, これらの機能を実現可能としている. 以下に外部通信機構が持つ機能とデータを挙げる.

- read スレッド
  read スレッドは, 各々の抽象化層初期化時にカーネルにより作成され, 抽象化層の受信パケットキューを監視する. すなわち, read スレッドは, 受信パケットキューにパケットがなければ, 待ち状態となる. read スレッドは, 受信パケットキューにパケットがあれば実行可能状態となる. その後, カーネルへ処理が移行し, プロトコルスタックがパケットの処理を行う.

- 受信パケットキュー
  受信パケットキューは, 各々の抽象化層が1つずつ持ち, 受信したパケットをリスト構造で管理している. 受信パケットキューは, 抽象化層の初期化時に生成され, 当該抽象化層は, カーネルに受信パケットキューの生成を通知する. 受信パケットキューの生成通知を受信したカーネルは, 受信パケットキューが生成された抽象化層に対し, read スレッドを生成する.

- メッセージキュー
  メッセージキューは, カーネル内に存在し, メッセージをリスト構造で管理している. 本機構では, 各抽象化層から取得したパケットをプロトコルスタック内で再構築し, カーネルが管理するメモリ空間にメッセージとして格納する. メッセージキューは, ポートごとにリストを持つ. ユーザは, read() や resvfrom() を用いることで, メッセージキューに格納されているメッセージを読み出すことが可能である.

カーネルは, read スレッドを利用することで, すべての計算機が取得したパケットを処理することができる. これは, 受信処理を行うカーネルスレッドがどの計算機へ移動しても, すべての抽象化層に対して, read スレッドが, 受信パケットキューの監視を行っているためである.

5.3 処理方法

外部通信機構は, 送信処理部, 読出し処理部, 受信処理部から構成される. 送信処理部は, 送信時に使用され, 読出し処理部と受信処理部は, 受信時に使用される. 以下, これらの処理手順について述べる.

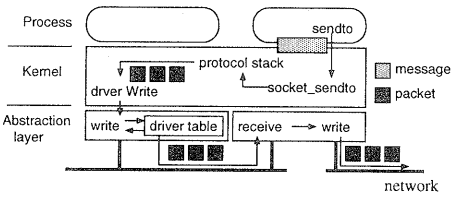

図 6 送信処理

5.3.1 送信処理部

送信処理部は, ユーザアプリケーションが管理外の計算機にメッセージを送信するときに使用する. 図6は, 管理外の計算機にメッセージを送信するときの処理の流れを示したものである. アプリケーションから write() または sendto() が呼び出されると, カーネルは, メッセージをユーザ領域からカーネル領域にコピーする. カーネルは, socket() により得たファイルディスクリプタにより, 下位のプロトコルを選択し, メッセージの処理をプロトコルスタックに依頼する. 次に, プロトコルスタックがメッセージに適切なヘッダを付加し, メッセージがデータリンクの MTU よりデータサイズが大きいときは, メッセージをフラグメント化する.

送信処理部は, パケットを送信すべきデバイスを選択し, 管理外の計算機へパケットを送信するために, 抽象化層に対してパケットの送信を要求する. 要求を受けた抽象化層は, パケットをカーネル領域から抽象化層が管理するメモリ領域へコピーする. 抽象化層は, カーネルが指定したデバイスが他の計算機に接続されている場合, カーネルが指定したデバイスを持つ計算機へパケットを送信する. カーネルが指定したデバイスを持つ計算機は, パケットを受信すると, 指定されたデバイスを使用し, パケットを他のセグメントへ送信する. 抽象化層は, カーネルが指定したデバイスが自ノードにあるとき, そのままパケットを送信する.

5.3.2 読出し処理部

読出し処理部は, カーネル領域からユーザ領域へのメッセージのコピー, 抽象化層からカーネル領域へのパケットのコピーの2つの機能から構成される. 図7は, 読出し処理部と受信処理部の処理の流

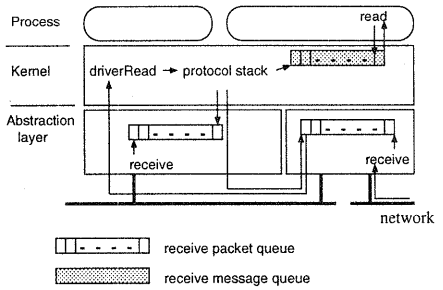

図 7　受信と読出し処理

れを示したものである．この 2 つの処理は，非同期に実行される．

ユーザは，受信ポート番号を設定し，read() または resvfrom() を呼び出すことで，メッセージを受信することができる．カーネルは，read() または resvfrom() が呼び出されると，受信ポート番号をメッセージキューに登録する．この登録処理により，メッセージキューには，登録したポート番号宛に到着したメッセージが挿入される．次に，カーネルは，各プロトコルスタックが持つ read() を呼び出す．read() は，登録したポート番号宛のメッセージがあるとき，ユーザが指定したメモリ領域へメッセージをコピーする．ユーザスレッドは，read システムコールを用いることで，メッセージを取得可能となる．

5.3.3　受信処理部

受信処理部は，カーネルが，管理外の計算機からメッセージを受信するときに使用する．パケットを受信した抽象化層は，パケットの種類が管理外の計算機から到着したパケットか他の抽象化層から到着したパケットかを判別する．パケットが管理外の計算機から到着したパケットであれば，受信パケットキューに挿入する．read スレッドは，受信パケットキューからパケットを取得し，当該パケットの処理を行う．パケットは，パケットが分割されていないとき，メッセージとして受信メッセージキューに挿入される．分割されているときは，プロトコルスタック内でパケットを再構築し，メッセージキューへ挿入される．

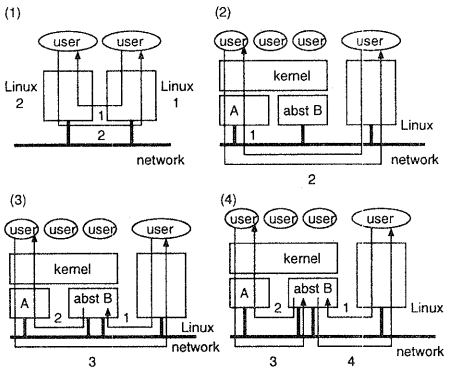

図 8　実験環境

6　評価

本章では，Solelc における外部通信時の処理時間について述べる．外部通信機構の評価を，UDP エコープログラムを用いて行った．UDP エコープログラムとは，クライアントが UDP を用いて送信したメッセージをサーバが受信し，サーバがデータをそのままクライアントに返送するプログラムである．評価環境は，Celeron 500MHz, Intel EtherExpress 100+ Ethernet Card を搭載した PC/AT 互換機を，100Mbps のイーサネットで接続したものである．なお，評価に使用した Linux は，Vine Linux 2.0, kernel 2.2.14 である．上記の環境を用いて，以下の 4 パターンについて評価を行った．図 8 は，4 種類の実験環境と処理手順について示したものである．本実験では，抽象化層 A が動作する計算機上で，メッセージを処理するカーネルスレッドが動作するものとする．

(1) Linux 1 がデータを送信し，Linux 2 がデータを受信し，データを Linux 1 へ返送する．

(2) Linux がデータを送信し，抽象化層 A がデータを受信する．カーネルは，抽象化層 A から，データを受信する．ユーザプロセスは，データを受信し，返送する．データを受信したカーネルは，抽象化層 A が管理するデバイスを用いて，Linux へデータを返送する．

(3) Linux がデータを送信し，抽象化層 B がデータを受信する．データは，抽象化層 A へ移送され，カーネルがデータを受信する．ユーザプ

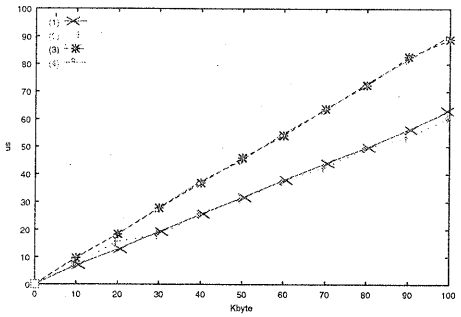

図 9　評価

ロセスは, データを受信し, 返送する. データを受信したカーネルは, 抽象化層 A のデバイスを操作し, Linux へデータを送信する.

(4) Linux がデータを送信し, 抽象化層 B がデータを受信する. データは, 抽象化層 A へ移送される. カーネルは, 抽象化層 B のデバイスを操作し, Linux へデータを返送する. このとき, データは, 抽象化層 A から抽象化層 B へ転送され, 抽象化層 B が持つデバイスがデータを送信している.

(2) は, 受信処理カーネルスレッドがパケットを受信した計算機上にあり, 送信すべきデバイスが, パケットを送信する計算機にあるため, もっとも通信回数が少ない. (4) は, 受信処理カーネルスレッドが, パケットを受信した計算機上になく, 送信すべきデバイスが, パケットを送信する計算機にないため, 最も通信回数が多い.

図 9 と表 1 は, 4 種類の実験結果を示したものである. (1) と (2) の差が非常に小さいことから, 同一環境下において, 1 対 1 の計算機間で通信を行う場合, 抽象化層とカーネルの 2 層構造でネットワーク機構を実現している Solelc と, カーネル上にネットワーク機構を実現している Linux が, ほぼ同等の性能を持つと言える.

Solelc では, カーネルが動作している計算機と異なる抽象化層が管理するデバイスを操作し, パケットを送受信するとき, 通信オーバヘッドは増大する. (3) の場合は (1) や (2) よりも 1 回通信回数が多く, (4) の場合は 2 回通信回数が多いため, オーバヘッドが大きい. このオーバヘッドは, 従来の OS では使用しない内部通信を抽象化層間で行うために発生する.

表 1　メッセージの処理時間 単位 μ(sec)

	(1)	(2)	(3)	(4)
0.001 KB	0.324	0.413	0.529	0.466
0.01 KB	0.267	0.386	0.478	0.469
0.1 KB	0.324	0.400	0.529	0.536
1 KB	0.735	0.708	1.051	1.092
10 KB	7.016	7.580	9.582	9.842
100 KB	63.102	60.466	88.868	90.298

しかし, 各抽象化層が自身の取得したパケットをカーネルに通知するため, カーネルは位置透過にすべての計算機が取得したパケットが処理可能である.　また, 位置透過に任意の計算機からパケットを送信可能である.

7　おわりに

本稿では, 分散 OS Solelc におけるネットワーク機構の構成について述べた. Solelc の内部通信機構は, 効率的かつ高信頼な通信を実現している. さらに, Solelc は, カーネル上に通信管理機構を実現し, 抽象化層にデバイス管理機構を実現している. Solelc 上で動作するユーザプロセスは, 外部通信機構を利用することで, Solelc が管理していない計算機上で動作するユーザプロセスと通信を行うことが可能となる. Solelc が Solelc 以外の OS との通信が可能となると, Solelc 上に, Web Server, DNS, NFS 等のアプリケーションを実現し, サーバを動作させることが可能となる.

OS がこのような構造を持つ場合, 従来の OS で用いられているルーティングの方式をそのまま適応することが困難となる. 今後は, Solelc に適したルーティング方式を明らかにする予定である.

参考文献

[1] 芝公仁, 大久保英嗣: “分散オペレーティングシステム Solelc の構成,” 情報処理学会研究報告 2000-OS-84, pp.237-244 (2000).