システムソフトウェアと
オペレーティング・システム　89－１８
シ　ス　テ　ム　評　価　2－１８
（２００２．２．１５）

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評価

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

現在, 我々は, 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. 本論文では, Solelc におけるネットワーク機構について述べる. 本機構では, パケット処理の負荷を分散させるために, すべての計算機でパケット処理を可能にしている. また, 適切な経路制御を行うことにより, 単一の Solelc を複数のネットワークセグメントで動作させることも可能としている. 本論文では, これらの機能を実現しているネットワーク機構の構成について述べる. また, 本機構の評価を行い, Solelc の有用性について議論する.

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

Evaluation of Network Function in 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 Solelc distributed operating system. In Solelc, plural computers connected via network are managed by a single operating system. In this paper, the network function in Solelc is described. In order to distribute the cost of packet processing, the network function is enabled to process packets on any machines. Furthermore, a single Solelc can run on several network segments by appropriately controlling packet routing. In this paper, the structure and performance evaluation of the network function in Solelc are described.

Key words: Distributed operating system, Network

1 はじめに

現在, 我々は, 分散 OS Solelc の開発を行っている. Solelc は, 従来の OS とは異なり, 1 つの 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, 自らが管理している計算機との通信と, 自らが管理していない計算機との通信の 2 つの形態がある. Solelc では, 自らが管理する計算機と通信を行うときには, 専用のプロトコルを使用する. 本プロトコルは, 計算機間で効率の良い協調処理を実現している. 自らが管理していない計算機と通信を行うときには, Solelc 専用のネットワーク機構を使用する. Solelc 上で動作するプロセスやカーネルは, ネットワーク機構を使用して, 他の OS との通信を行う. ネットワーク機構は, 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, UDP などをサポートしている.

Solelc では, カーネルやプロセスは, Solelc が管理するすべての計算機上で位置透過に動作可能である. したがって, カーネル内に通信を管理する機能を実現することにより, すべての計算機上のプロセスやカーネルは, 他の OS との通信が可能となる. さらに, プロセスやカーネルは, 位置透過にメッセージを受け渡しすることも可能である.

ネットワーク機構では, ネットワークインタフェース (以下 インタフェース) を管理している. Solelc では, 他の OS との通信時には, すべてのインタフェースを利用可能である. すなわち, プロセスは, 任意のインタフェースを利用して通信可能である. 複数のインタフェースでパケットを送受信することにより, 複数の計算機にパケット処理の負荷が分散され, システム全体の処理効率が向上する. また, カーネルにおいて, パケット処理を複数の計算機で分散して行うことにより, システム全体の処理効率の向上を図る.

本稿では, 以上の特徴を持つネットワーク機構の構成と評価について述べる. 以下, 2 章でシステムの構成, 3 章でネットワーク機構の概要と研究目的, 4 章で複数の計算機で IP を動作させる手法, 5 章で抽象化層間の経路制御について述べる. また, 6 章でシステムの評価について述べ, 7 章で本稿のまとめを述べる.

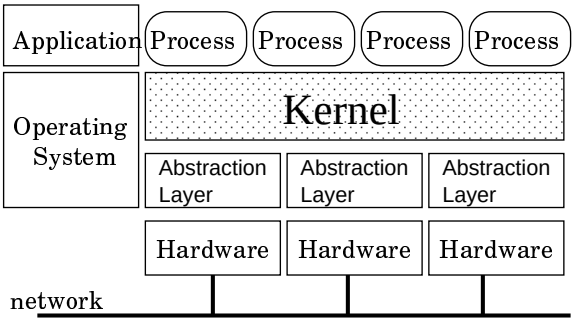

図 1 システム構成

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. 具体的には, 抽象化層は, 以下に示す機能を用いて資源の管理を行う.

- メモリ
  各計算機の物理メモリを, 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として抽象化し, 位置透過にメモリアクセスを行うことを可能とする.

- CPU
  CPU資源を, これを使用する単位となるスレッドとして抽象化し, スレッドを任意の計算機上で動作可能とする.

- 周辺デバイス
  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, 任意の計算機からデバイスドライバを使用可能にすることにより, すべてのプロセス, カーネルは, 位置透過にデバイスを操作することが可能となる.

- 割り込み
  割り込みを, 各計算機で発生した事象をカーネルに通知するイベントとして抽象化し, 任意の計算機上でイベントの取得を可能とする.

さらに, 抽象化層は, 抽象化層間で協調処理を行うために抽象化層間でシステム管理情報を通信する. 抽象化層間の通信には, 専用のプロトコルが使用され, 効率の良い協調処理を実現している. 各抽象化層は, 互いに協調処理をすることにより, 以下に示す機能を実現する,

- 計算機やシステムが使用する資源に対して, システム全体で一意な整数値である識別子を付与する. 計算機には計算機 ID が付与され, デバイスにはデバイス ID が付与される.

- カーネルからの資源操作の要求を, 対象となる資源をもつ計算機の抽象化層へ転送する.

抽象化層には, デバイス情報を管理するために, 各抽象化層がデバイスリストを持つ. デバイスリストは, デバイスタイプ, 最小処理サイズ, デバイス ID, デバイスが接続されている計算機 ID といったデバイスの情報を保持する. デバイスタイプは, イーサネットカードあるいはキーボードといったデバイスの種類を識別する識別子である. 最小処理サイズは, デバイスが処理可能な最小処理単位である. 例として, イーサネットカードは, 1 バイト, ディスクは, 512 バイト (1 ブロック) などがある.

カーネルは, 資源に設定された識別子を用いることにより, 資源に対する操作を位置透過に行うことが可能である. 資源操作の要求は抽象化層によって転送されるため, カーネルは, 自身が動作する計算機や資源の位置とは無関係に, 自身が動作する計算機上の抽象化層に対して要求を出せばよい.

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. Solelc では, IP, UDP, TCP などの通信プロトコルをカーネル内に実現し, 他のシステムとの通信を行う. IP アドレスとインタフェースの関係は, カーネルによって管理される. ユーザは, Solelc が管理するすべてのインタフェースに対して, IP アドレスを設定可能である. カーネルで IP を実現するため, すべてのプロセスは, 位置透過に他のプロセスと通信することが可能である.

3 ネットワーク機構

3.1 概要

Solelc では, 自らが管理していない計算機と通信を行うときには, ネットワーク機構を使用する. ネットワーク機構は, カーネル内に通信管理機能を実現し, 抽象化層内のデバイス管理機能を使用して動作する. 抽象化層が受信したパケットの内, 自らが管理していない計算機からのパケットは, カーネルによって処理される. カーネルは, すべてのインタフェースが受信したパケットを取得し, 一元的にパケットの処理を行っている. これにより, カーネル上で動作するプロセスは, どの計算機上で動作していても他のシステムとの通信を行うことが可能である [2].

3.2 目的

本研究の目的は, Web サーバのように他の計算機から多数のパケットを受信するシステムを想定し, パケット処理の負荷分散を行うことにより, システム全体の効率化を図ることである. 図 2 に, Web サーバを想定したシステムモデルを示す. 図 2 では, 計算機 1, 2, 3, 4, 5 上で単一の Solelc が動作し, この Solelc 上で Web サーバが動作している. また, 6, 7 上では, それぞれ他の OS が動作しているものとする. A から J は, イーサネットカードであり, A から H は, スイッチングハブに接続されており, 各スイッチングハブは, 異なるセグメントに構成している. I, J は, 別セグメントと接続されている. また, IP アドレスは, D, G, H, I, J に設定されている. I, J は, セグメント外の計算機と接続されている. 当該環境において Solelc を動作させたとき

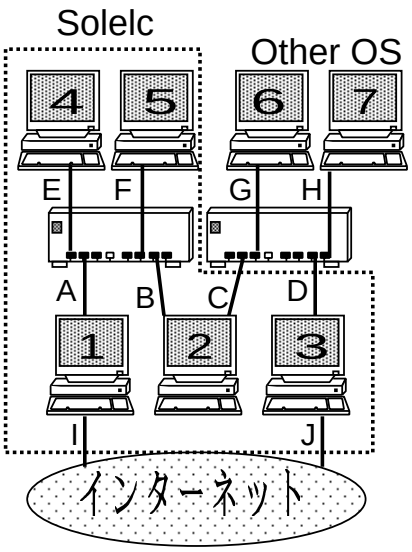

図 2　システムモデル

の特徴と利点を以下に示す.

- 単一のホスト名に対して, I, J に設定された IP アドレスを設定する. 他の OS は, ネームサーバから任意の IP アドレスを取得し, Web サーバとの通信を行う. このとき, I, J のどちらのインタフェースからパケットを受信しても, 同様に Web サーバと通信することが可能である. これにより, 複数の計算機にパケット処理負荷が分散され, システム全体の処理効率が向上する.
- 抽象化層内で経路制御を行うことにより, 単一の Solelc が複数セグメントで動作可能である.

Solelc では, どの計算機上でパケットを受信しても一元的にパケット処理を行う機能を利用し, システム全体の効率化を図る. また, Solelc では, 抽象化層が経路制御を行うため, 複数のセグメントにまたがって動作することが可能である.

3.3　機能

ネットワーク機構では, カーネル内にインタフェースの管理情報を持つ. Solelc では, カーネルがインタフェースを操作するときには, 抽象化層に対して, 使用するインタフェースのデバイス ID を通知する. ネットワーク層では, すべてのデバイス情報を持つインタフェース管理テーブルを用いてインタフェースを管理する.

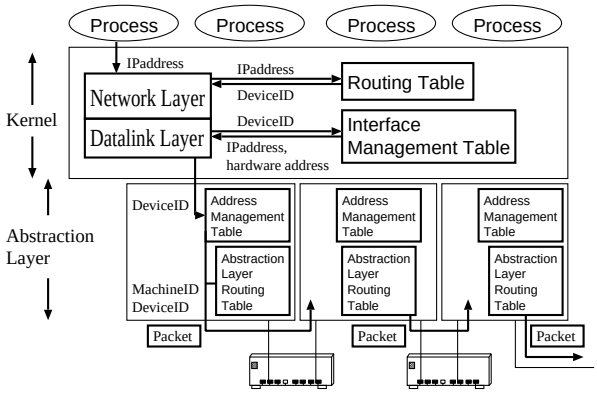

図 3　機能配置図

また, ネットワーク機構では, カーネルが使用するインタフェースを決定するとき, ネットワーク層に実現されたルーティングテーブルを参照する. カーネルは, ルーティングテーブルからパケットを送出すべきデバイス ID を取得する. カーネルは, ルーティングテーブルから取得したデバイス ID を基に, 送信元のデバイス情報をインタフェース管理テーブルから取得し, パケットのプロトコルヘッダに付与する. 抽象化層は, カーネルからインタフェース操作要求を取得し, 当該インタフェースが接続された計算機へパケットを転送する. 転送先の計算機では, 指定されたインタフェースを用いて, パケットを送信する.

抽象化層は, デバイスリストからカーネルに指定されたデバイスを持つ計算機の ID を得る. 抽象化層は, 当該計算機へパケットを送信するとき, パケットに当該計算機が持つインタフェースのハードウェアアドレスを付与する. 抽象化層では, 計算機 ID を基にアドレス管理テーブルからハードウェアアドレスを取得する.

また, 抽象化層は, 経路を管理する. 経路を管理することにより, 単一の Solelc が複数のセグメントで同時に動作することが可能となる. 経路を特定するときには, 送信先計算機 ID から送信元のデバイス ID と送信先のデバイス ID を特定し, 送信先の計算機へパケットを送信する. 図 3 に, カーネル, 抽象化層にそれぞれ配置されたテーブルを示す. 次章以降では, 各テーブルの詳細な情報や登録処理について述べる.

4　複数の計算機上で動作する IP

カーネルは, 各計算機のインタフェースの情報や経路の管理を行う. Solelc では, インタフェース管理テーブルを用いて, インタフェース情報を管理する. また, カーネル内で経路管理するために, ルーティングテーブルを用いる. 以下, 本章では, 各テーブルが持つ情報や登録手順, 使用法について述べる.

4.1　ルーティングテーブル

ルーティングテーブルは, プロセスがカーネルに対して, 経路情報を通知することによって作成される. テーブルへの登録は, 従来の Linux などと同様の方法で行う. したがって, ここでは説明を省略する.

ネットワーク層では, パケットの送信処理を行うとき, ルーティングテーブルを参照し, 送信先 IP アドレスから当該パケットを送出すべきインタフェースのデバイス ID を特定する. 送出するインタフェースは, 以下に示す情報から決定される.

- IP アドレス
- ネットマスク
- デバイス ID

ネットワーク層では, デバイス ID を特定するとき, IP アドレスとネットマスクの論理積を計算する. カーネルは, 計算結果と一致するレコードのデバイス ID を取得する. これにより, カーネルは, 送信するインタフェースを決定することが可能である.

4.2　インタフェース管理テーブル

インタフェース管理テーブルには, システムが持つすべてのインタフェースが登録されている. インタフェース管理テーブルは, カーネルがプロセスからメッセージの送信要求を取得したときや, 受信したパケットを他のセグメントへフォワーディングするときに参照される. インタフェース管理テーブルは, 以下の情報を保持する.

- IP アドレス
- ハードウェアアドレス
- デバイス ID

インタフェース管理テーブルのレコードは, システムにインタフェースが追加される毎に追加される. カーネルは, デバイス ID とハードウェアアドレスを, 抽象化層から取得し, 登録する. カーネルは, インタフェース情報と IP アドレスをプロセスから取得し, インタフェースに対して, プロセスが指定した IP アドレスを付与する.

カーネルは, パケットを送信するときには, ルーティングテーブルから送信元となるインタフェースのデバイス ID を取得する. カーネルは, デバイス ID を基にインタフェース管理テーブルから, IP アドレスとハードウェアアドレスを取得する. この IP アドレスとハードウェアアドレスは, Solelc が他の OS にパケットを送信するときに最終的に送出するインタフェースに対応したものである. すなわち, IP アドレスは, ネットワーク層で送信元 IP アドレスとしてパケットのヘッダに付与される. ハードウェアアドレスは, データリンク層で送信元ハードウェアアドレスとしてパケットのヘッダに付与される. パケットが完成すると, カーネルは, パケットを送信するために, デバイス ID とパケットを抽象化層へ通知する.

カーネルは, 送信先の計算機情報を得るときに, 従来から使用されている ARP テーブルを使用する. ARP テーブルは, IP アドレスとハードウェアアドレスから構成される. カーネルは, 経路が決定すると ARP テーブルから, 次に中継する計算機のハードウェアアドレスを取得し, パケットに付与する.

このように, カーネルが持つ機能は, 従来のシステムにおけるそれと同様である. 従来のシステムとの違いは, 抽象化層が実現する環境上で動作することによって, 同時に複数の計算機が持つインタフェースを管理することである.

5　抽象化層間の経路制御

抽象化層は, 計算機やデバイスの情報を管理することにより, カーネルが位置透過に動作可能な環境を実現している. 抽象化層は, カーネルがすべてのデバイスを使用可能とするために, すべての計算機の情報や計算機へ到達するための経路を管理する. これらの情報は, アドレス管理テーブルや経路制御テーブルによって管理される. 以下, 本章では, これらのテーブルが持つ情報や情報の登録手順, 使用法について述べる.

5.1 アドレス管理テーブル

アドレス管理テーブルは, 抽象化層内に実現され, すべての抽象化層で同一の情報を持つ. 本テーブルは, 他の抽象化層間での通信に使用される. アドレス管理テーブルは, 以下に示す情報を保持する.

- ハードウェアアドレス
- デバイス ID

アドレス管理テーブルのレコードは, 計算機が追加されたときに追加される. 当該計算機に接続されたインタフェースのハードウェアアドレスとデバイス ID を取得するときには, デバイスリストを利用する. Solelc では, 計算機追加時にすべての抽象化層が持つデバイスリストが更新される. アドレス管理テーブルのデバイス ID を追加するときには, デバイスリストから, ドライバタイプがネットワークカードかつ計算機 ID が新しく追加された計算機 ID であるデバイス ID を取得する. 複数のインタフェースが接続されているときには, 複数のデバイス ID を取得する. このとき, 抽象化層では, デバイス ID とハードウェアアドレスを対応づけすることにより, インタフェースの識別を行う. 抽象化層では, ネットワークカード管理構造体を用いて, ネットワークカードの管理を行っている. ネットワーク管理構造体は, ネットワークカードを操作する関数へのポインタ, IRQ, ハードウェアアドレスから構成される. ネットワークカード管理構造体は, インタフェース初期化時に作成し, デバイス検出順に追加される. Solelc では, デバイス ID もデバイス検出順に整数値が設定される. これにより, デバイス ID とハードウェアアドレスの対応づけが可能である.

抽象化層は, パケット転送時には, デバイスリストを参照し, デバイス ID から計算機 ID を取得し, パケットのヘッダに付与する. 次に, 抽象化層は, 計算機 ID を基に, 経路制御テーブルから送信先のデバイス ID と送信元のデバイス ID を取得する. これらのデバイス ID を取得すると, 当該デバイス ID を基にアドレス管理テーブルからハードウェアアドレスを取得し, パケットのヘッダに付与し, 当該計算機へパケットを転送する. アドレス管理テーブルを使用することにより, 抽象化層は, パケット転送先の計算機情報を特定し, すべての計算機へパケットを転送することが可能となる.

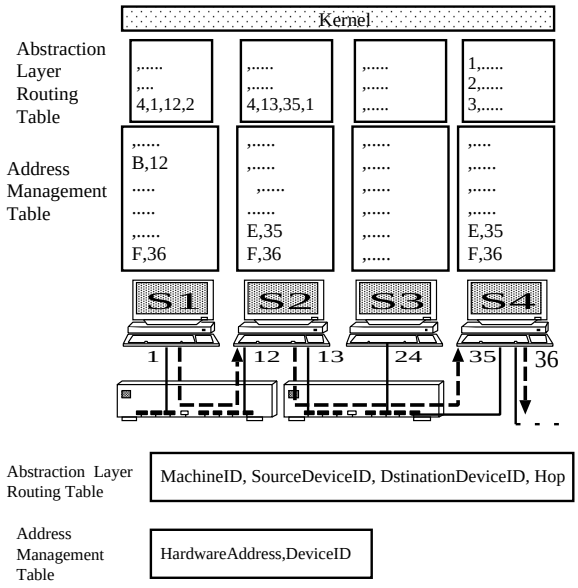

図 4 経路制御の手順

5.2 経路制御テーブル

経路制御テーブルは, 計算機毎に異なる情報を持つ. 本テーブルは, 他の計算機へパケットを転送するときに, 当該計算機へ到達するために必要な経路情報を持つ. 経路の制御を行うときには, 以下に示す情報を使用する.

- 送信先計算機 ID
- 送出先デバイス ID
- 送出元デバイス ID
- ホップ数

本テーブルのレコードは, 送信先の計算機 ID が本テーブル内に存在しないときに作成される. 当該計算機 ID が本テーブルに存在しないときには, ブロードキャストで他の計算機に経路取得要求を送信する. 要求を受信した計算機が経路を持つときは, デバイス ID とホップ数を返答する. 返答通知を受信した抽象化層は, 最小のホップ数を持つ計算機のデバイス ID とホップ数を経路として登録する. ホップ数は, 目的の計算機へ到達するまでに中継する計算機数である.

抽象化層は, パケット転送時に送信先計算機 ID を基に, 経路制御テーブルから, 送信元デバイス ID

と送信先デバイス ID を取得する．これにより，他の計算機への経路を特定することが可能となる．

経路制御の処理例を図 4 に示す．図中の計算機 S1 が計算機 S4 のデバイス ID36 のインタフェースを使用してパケットを送信するときの処理手順を以下に示す．

(1) S1 は，転送先の計算機である S4 の計算機 ID を基に経路制御テーブルから，送信元デバイス ID1，送信先デバイス ID12，ホップ数 2 を取得する．

(2) 送信先デバイス ID12 を基に，アドレス管理テーブルから，ハードウェアアドレス B を取得する．

(3) ハードウェアアドレス B をパケットのヘッダに付与し，計算機 S2 にパケットを転送する．

(4) 転送されたパケットを受信した S2 は，パケットを解析し，パケットのヘッダから，転送先の計算機が S4 であることを知る．

(5) S2 は，経路制御テーブルを参照し，送信元デバイス ID13，送信先デバイス ID35，ホップ数 1 を取得する．

(6) 送信先デバイス ID35 を基に，アドレス管理テーブルから，ハードウェアアドレス E を取得する．

(7) ハードウェアアドレス E をパケットのヘッダに付与し，計算機 S4 にパケットを転送する．

(8) 転送されたパケットを受信した S4 は，パケットを解析し，自計算機が持つデバイス ID36 を操作してパケットを送信する．

本機能を利用することにより，別セグメントの計算機へパケットを転送することが可能となる．さらに，抽象化層内で経路制御を行うことにより，すべての計算機にパケットを転送することも可能となる．したがって，単一の Solelc が複数のセグメントで動作することが可能となる．

6 評価

本章では，Solelc におけるネットワーク機構の評価を行い，その結果に基づいて Solelc の有用性について述べる．評価環境には，すべて Celeron 500MHz を搭載した PC/AT 互換機を用いた．ネットワークインタフェースカードには，Intel EtherExpress100+を用い，カード間を 100Mbps のケーブルで接続した．評価では，最初に Solelc と Linux のプロトコル処理の性能を比較する．Solelc では，抽象化層とカーネルの 2 つでパケットを処理するため，従来の OS に比べ，メモリコピーの回数，コンテキストスイッチの回数がそれぞれ 1 回ずつ増加する．これらのオーバヘッドを計測し，従来の OS と比較することにより，ネットワーク機構全体の性能を示す．

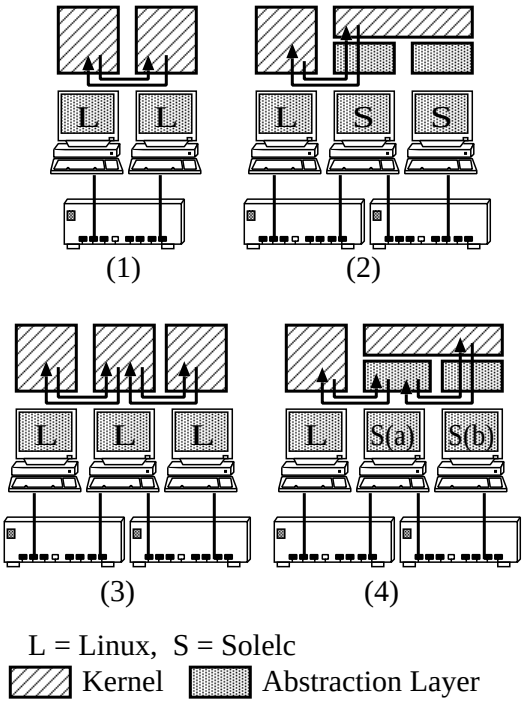

図 5 評価環境

6.1 プロトコルスタックの評価

Linux と Solelc の性能比較をプロトコル処理を用いて行う．評価には，図 5 に示す環境を用いた．通信プロトコルには，TCP/IP を用いた．クライアントには，メッセージをサーバに送信し，サーバからメッセージを受信するアプリケーションを用いた．また，メッセージ送信時とメッセージ受信時にそれぞれ時刻を取得し，その差分から通信時間を計測した．サーバには，受信したデータをそのままクライアントへ返送するアプリケーションを用いた．クライアントには，すべて Linux 上で動作さ

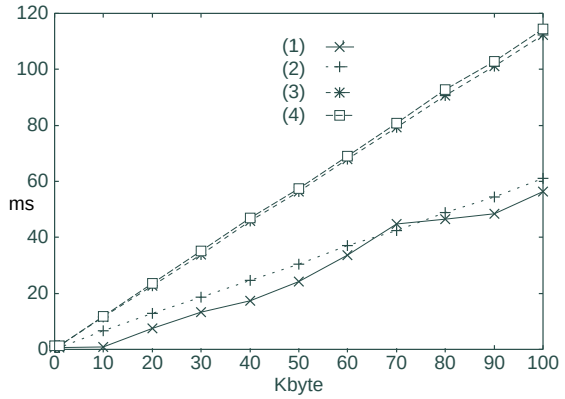

図 6　プロトコルスタックの性能比較

せた．(1) では，サーバに Linux を用い，(2) では，サーバに Solelc を用いた．(2) の Solelc は，プロセス，カーネルがすべて同一の計算機上で動作しているため，他の計算機とのシステム管理情報の通信は行われない．図 6 の結果より，(1), (2) は，ほぼ同等の結果が得られたため，Solelc と Linux のプロトコルスタックの性能は，ほぼ同等であるといえる．

6.2　複数の計算機を用いた評価

従来の OS では，異なるセグメントに存在する計算機との通信を行うときは，図 5(3) に示すモデルを用いて通信を行っている．図 5(4) は，計算機 S(a), S(b) 上で単一の Solelc が動作している．(4) では，計算機 S(b) 上でプロトコル処理を行うものとする．(4) では，計算機 S(b) 上でプロトコル処理を行うカーネルスレッドが動作しているため，計算機 S(a) が受信した IP パケットは，計算機 S(b) に転送され，処理される．本モデルは，パケットを受信した計算機とは，別の計算機上で IP パケットを処理するため，Solelc おいて，もっとも通信回数が多い場合となる．

図 6 の結果より，(3), (4) は，ほぼ同等の結果が得られた．Solelc では，プロセスが位置透過に動作可能なため，ネットワークを頻繁に使用するプロセスを，図 5(4) の計算機 S(a) 上に配置することができる．これにより，転送するパケット数が減少し，システム全体の処理効率が向上させることができる．

7　おわりに

本稿では，分散オペレーティングシステム Solelc 上に実現されたネットワーク機構の構成とその評価について述べた．複数の計算機上でプロトコル処理を行うネットワーク機構を利用し，Solelc 上で Web サーバなどを動作させたときには，ネットワーク負荷が減少し，システム全体の処理効率が向上した．さらに，抽象化層間で経路を制御することにより，Solelc は，複数のセグメントに存在する複数の計算機を同時に管理することが可能となった．また，実験により，ネットワーク機構の使用時に，もっとも通信回数が多い場合でも従来の OS が実現する手法と同等の結果を得た．その他の場合には，通信回数を削減できるようなプロセスの配置が可能であり，従来の OS が実現する手法に比べて，最大約 50%削減された．

本研究では，パケットを受信した計算機と別の計算機でプロトコル処理を行った場合と，パケットを受信した計算機と同じ計算機でプロトコル処理を行った場合の評価を行った．今後は，パケットを受信した計算機にプロトコル処理を行うカーネルスレッドを移送し，パケットを受信した計算機上でパケットを処理する手法の検討および，Solelc におけるスケーラビリティについて検討する予定である．

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 の設計と実装，” 電子情報通信学会論文誌 D-I, Vol.J-84-D-I, No.6, pp.617-626 (2001).

[2] 藤本堅太，芝公仁，大久保英嗣：“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，” 情報処理学会研究報告 2000-OS-86, pp.107-114 (2001).