システムソフトウェアと
オペレーティング・システム 90－4
（２００２．６．２７）

分散オペレーティングシステムSolelcにおける スワップ機構の構成

中西 数樹 † 水口 孝夫 † 永宗 宏一 † 芝 公仁 †† 大久保 英嗣 †††

† 立命館大学大学院理工学研究科 †† 龍谷大学理工学部
††† 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. 本稿で述べるスワップ機構は, この特徴を利用することにより, ディスクを持たない計算機におけるスワップが可能となる. また, Solelcのメモリ管理を考慮することにより, 高速な空きページの作成が可能となる. 本稿では, Solelcにおけるスワップ機構の構成について述べた後, 本機構の評価を行い, その有用性について述べる.

The Structure of Swap System in Solelc Distributed Operating System

Kazuki Nakanishi † Takao Mizuguchi † Kouichi Nagamune †
Masahito Shiba †† Eiji Okubo †††

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

We have been developing Solelc distributed operating system. Plural computers which Solelc works on are managed by an operating system. The swap system described in this paper can make the machine without a disk do swapping. And, by considering memory manegement of Solelc, an operating system can rapidly make free pages. In this paper, the structure of swap system in Solelc and the evaluation of this system are described.

1　はじめに

近年, 計算機の低価格化とネットワーク機器の普及により, 複数の計算機をネットワークに接続して使用することが多くなってきている. このような環境において, カーネルが計算機ごとに存在するシステムでは, 計算機ごとに資源管理が行われるため, システム全体を考慮した資源管理が困難となっている. この問題の解決を目的として, 我々は, 単一のカーネルによって複数の計算機を管理する分散オペレーティングシステム Solelc[1, 2] の開発を行っている.

Solelc では, 各計算機上の計算機資源を抽象化し, カーネルが位置透過に資源管理を行う. これにより, すべての計算機において同一の環境が実現され, プロセスは, 任意の計算機上で動作することが可能となる.

従来の OS では, 物理メモリに空き領域が存在しない状況においてプロセスがページを要求した場合, スワップを行うことにより空きページを作成していた. 本稿では, これに加え, Solelc のメモリ管理の特徴を利用した空きページの作成手法を提案する. Solelc のスワップ機構の特徴を以下に挙げる.

- ディスクを持たない計算機におけるスワップ
- 高速な空きページの作成

Solelc では, カーネルが計算機資源を位置透過に管理しているため, ディスクを持たない計算機においてもスワップを行うことが可能である. また, Solelc では, 他の計算機にページの複製が存在するため, これを利用した高速な空きページの作成を行うことが可能である.

以下, 本稿では, 2 章で Solelc の構成ついて述べ, 3 章でスワップ機構の構成について述べる. 次に, 4 章で評価を行い, 最後に 5 章で本稿のまとめを行う.

2　Solelc の概要

Solelc は, 従来の OS とは異なり, 1 つのカーネルで複数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可能としている. 本章では, Solelc の概要について述べる.

2.1　システム構成

Solelc では, OS を抽象化層とカーネルの 2 層に階層化している. Solelc の構成を図 1 に示す. 抽象化層は, 計算機資源の抽象化を行い, カーネルが位置透過に資源管理を行うことを可能とするための環境を構築する. カーネルは, この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資源を管理することが可能となる. プロセスは, カーネルと同様に, 計算機の位置を意識することなく資源を利用することが可能である. 以下, 抽象化層およびカーネルについて述べる.

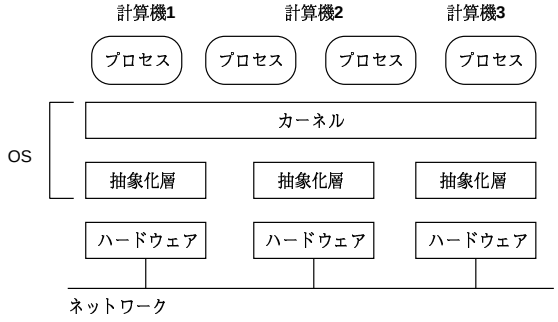

図 1　Solelc のシステム構成

- 抽象化層
  抽象化層は, すべての計算機上で 1 つずつ動作し, おのおのが自身の動作する計算機を管理する. 抽象化層の役割は, 各計算機上の資源を抽象化することである. 抽象化層が抽象化する資源は, メモリ, CPU, 割込み, 周辺デバイスである. メモリ管理において, 抽象化層は, 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アドレス空間として抽象化し, 一貫性制御を行うことにより位置透過にメモリアクセスを行うことを可能としている.

- カーネル
  カーネルは, システム全体で 1 つであり, すべての計算機資源を管理する. 計算機ごとにカーネルを動作させている従来のシステムでは, 他の計算機の資源を使用するためには協調動作のための処理が必要であった. 一方, Solelc では, 単一のカーネルがすべての計算機を管理するため, 協調処理が不要である. カーネルは, 抽象化層の上で動作することにより, 位置透過な資源管理を行うことができる. カーネルの役割は, 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ことである. メモリ管理において, カーネルは, アドレス空間を複数の領域に分割してプロセスを

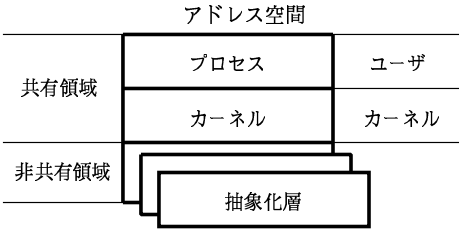

図 2　Solelc のアドレス空間

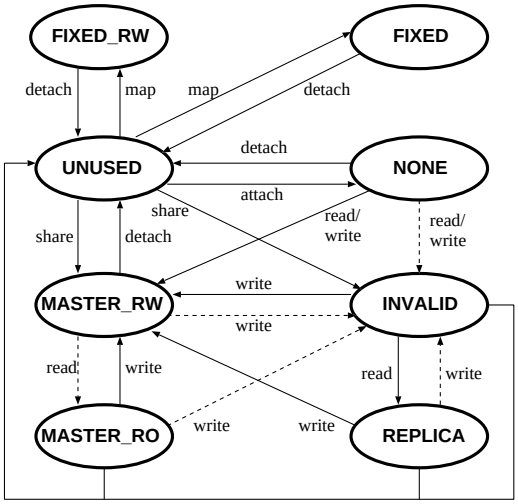

図 3　ページの状態遷移

配置し, プロセス間でのメモリの共有や保護の設定を行う.

2.2　アドレス空間

Solelc では, アドレス空間を共有領域と非共有領域に分割している. Solelc におけるアドレス空間を図 2 に示す.　Solelc のアドレス空間は, 抽象化層が配置される非共有領域と, カーネルおよびプロセスが配置される共有領域から構成される. 非共有領域は, 計算機ごとに固有の内容を持つ. 一方, 共有領域は, すべての計算機が同一のメモリ内容と保護属性を持つ. プロセスのコードおよびデータは, 共有領域上に存在するために任意の計算機上でコードやデータにアクセス可能であり, プロセスは位置透過に動作する.

2.3　メモリ管理

抽象化層は, 共有領域の各ページの所有者と状態を管理する. ページの所有者とは, ページを管理する計算機である. ページの所有者となった計算機は, ページの複製の管理を行う. 他の計算機がページを必要とする場合は, ページを所有する計算機へ要求することにより, ページの内容を取得することができる.

表 1　ページの状態

ページの状態	説明
UNUSED	使用されていない
NONE	使用可能であるが, どの計算機の物理メモリも割り付けられていない
MASTER_RW	ページの内容を持ち, 他の計算機は複製を持たない
MASTER_RO	ページの内容を持ち, 他の計算機が複製を持つ
REPLICA	MASTER_RO 状態のページの複製
INVALID	無効な状態
FIXED_RW	物理メモリが割り付けられている移送できないページ
FIXED	物理メモリが割り付けられていない移送できないページ
SWAPPED	ページアウトした状態

Solelc は, ページの状態を複数用意することにより, 共有メモリの管理を行っている. Solelc の共有領域におけるページの状態を表 1 に示す. 抽象化層は, カーネルからの要求やプロセスが行うメモリアクセスに応じてページの状態を遷移させる. ページの状態遷移の様子を図 3 に示す. 図中の点線は，他の計算機で行われたメモリアクセスによって，ページの状態が変更されることを表す. 各ページの初期状態は UNUSED であり, この状態のページは使用不可である. NONE は, UNUSED の状態から使用可能な状態へと移行したが, どの計算機の物理メモリも割り付けられていない状態である. NONE 状態のページへアクセスが発生すると, ページの状態が MASTER_RW に遷移し, ページの状態の変更をすべての計算機に通知する. 通知を取得した抽象化層は, 当該ページの状態を INVALID に変更する. これらの処理は, すべて抽象化層によって行われるため, カーネルおよびプロセスは意識する必要がない. INVALID 状態のページへの読出しアクセスが発生すると, MASTER_RW のページを持つ計算機からページの内容を取得し, 状態を REPLICA へ

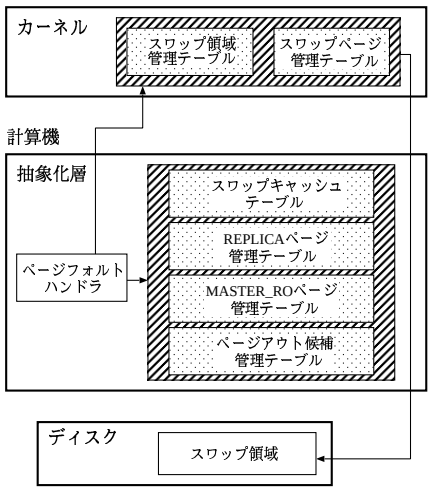

図 4　スワップ機構の内部構成

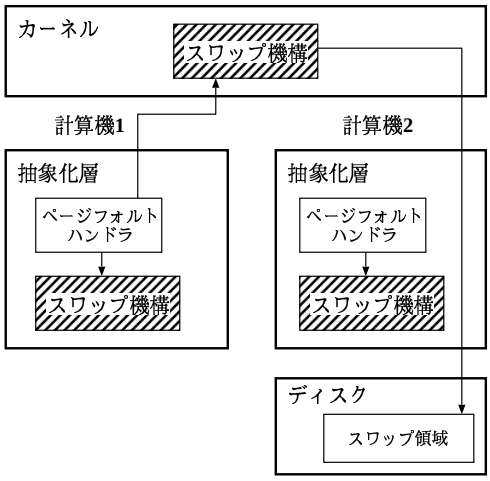

図 5　スワップ機構の協調動作

遷移させる．このとき，コピー元の MASTER_RW は，MASTER_RO に遷移する．MASTER_RW および MASTER_RO は，ページを所有している状態である．FIXED_RW および FIXED は，所有者の変更や複製の作成が行われない状態である．これは，VRAM などの I/O がマップされたメモリを共有領域に割り付けるとき，すなわち特定の計算機の物理メモリを固定的に割り付ける必要がある場合に使用される．ページアウトされた場合は，SWAPPED に遷移する．この状態のページは，物理メモリ上には内容が存在せず，スワップ領域上に退避されている．

3　スワップ機構の構成

Solelc におけるスワップ機構では，抽象化層とカーネルが協調して動作する．抽象化層では，自計算機の物理メモリを管理し，空きページの作成を行う．カーネルでは，計算機全体のスワップ領域の管理を行う．図 4 にスワップ機構の内部構成を示す．抽象化層は，スワップキャッシュテーブル，REPLICA ページ管理テーブル，MASTER_RO ページ管理テーブルを用いて空きページを作成する．また，抽象化層は，アクセス頻度の低いページを検出するためにページアウト候補管理テーブルを保持する．カーネルは，スワップ領域管理テーブルを用いて，ページアウトしたページの内容を格納しておくファイルであるスワップ領域を管理し，スワップページ管理テーブルを用いてページアウトしたページの情報を管理する．

カーネルと抽象化層におけるスワップ機構の協調動作の例を図 5 に示す．図 5 において，計算機 1 はディスクを持っていない．そのため，ページアウトをする場合は，カーネルへ要求を通知し，計算機 2 のスワップ領域へページアウトしている．このような動作を行うことにより，Solelc におけるスワップ機構は，ディスクを持たない計算機におけるスワップを可能としている．

3.1　抽象化層における機構

物理メモリに空き領域が存在しない状況においてページ割当ての要求が発生した場合は，ページアウトする前にあらかじめ用意しておいたテーブルを使用して空きページを作成する．このとき，抽象化層は，スワップキャッシュテーブル，REPLICA ページ管理テーブル，MASTER_RO ページ管理テーブルを順に使用する．空きページが作成できない場合は，ページアウト候補管理テーブルを用いてページアウトする．各テーブルのデータを以下に示す．

- スワップキャッシュテーブル
  ページインされたページのアドレスを管理するテーブル
- REPLICA ページ管理テーブル
  REPLICA 状態のページのアドレスを管理するテーブル
- MASTER_RO ページ管理テーブル
  MASTER_RO 状態のアドレスを管理するテーブル

- ページアウト候補管理テーブル
  ページアウト候補となるページのアドレスを管理するテーブル
- スワップ領域管理テーブル
  スワップ領域を管理するテーブル
- スワップページ管理テーブル
  ページアウトしたページを管理するテーブル

以下, これらのテーブルを使用した動作について述べる.

3.1.1 スワップキャッシュの利用

スワップキャッシュテーブルは, スワップ領域上のデータを再利用するためのものである. スワップキャッシュテーブルは, スワップ領域上に同一のデータが存在する, メモリ上のページの更新を監視するために使用される. 抽象化層がスワップ領域上のページをページインした場合, メモリ上とスワップ領域上の当該ページは, 同一の内容を持つ. そのため, カーネルは, スワップ領域上の当該ページを有効にし, スワップキャッシュに存在するページを無効化することにより, ディスクに書き出すことなく空きページを作成することができる.

3.1.2 REPLICA ページの無効化

REPLICA ページは, 同一の内容が他の計算機に存在するため, 破棄することができる. 抽象化層は, あらかじめ REPRICA ページ管理テーブルを作成し, 空きページの要求が発生すると, このテーブルから REPLICA ページを取得する. 抽象化層は, テーブルから取得した REPLICA ページを無効化することにより空きページの作成を行う. REPLICA ページが存在する限り, そのうちの 1 つを無効化することにより, 当該物理ページを空き領域にすることができる.

3.1.3 MASTER 権限の譲渡

メモリ不足の解消は, REPLICA 状態のページを持つ計算機を当該ページの所有者へ変更することによっても実現可能である. これを行うため, 抽象化層は MASTER_RO ページ管理テーブルを作成する. MASTER 権限の譲渡の動作を図 6 に示す. 抽象化層は, 計算機 1 の MASTER_RO ページの MASTER 権限を, 計算機 2 の REPLICA ページを持つ計算機へ移送する. これにより, 計算機 1 の MASTER_RO ページは INVALID へ遷移し, 計算機 2 の REPLICA ページは MASTER_RO へ, 他に当該ページの複製が存在しない場合は MASTER_RW へ遷移する. この手法により, 抽象化層は MASTER 権限の譲渡を行うのみで空きページを作成することができる.

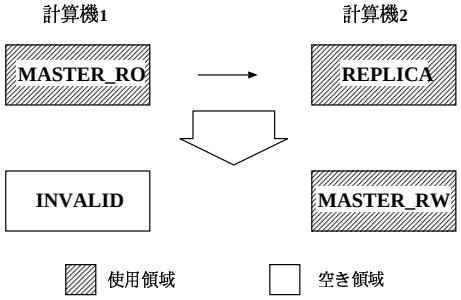

図 6 MASTER 権限の移動

3.1.4 ページアウトの実行

これまで述べた手法による空きページの作成を行うことができない場合は, ページアウトを実行する. UNUSED, NONE, INVALID, FIXED, FIXED_RW, SWAPPED 状態のページは, 物理メモリが割り付けられていない. また, MASTER_RO および REPLICA 状態のページは, 同一の内容が他の計算機に存在するため, 先に述べた手法を用いて破棄する. そのため, ページアウトの候補となるページは, MASTER_RW 状態のページである. 抽象化層は, MASTER_RW 状態のページが作成されるたびにページアウト候補管理テーブルへ追加する. また, スワッパは, ページアクセスの頻度を調べ, ページアウト候補管理テーブルを管理する. これにより, 抽象化層は, 使用頻度の低いページをページアウトすることができる.

3.2 カーネルにおける機構

カーネルにおけるスワップ機構は, システム内のすべてのスワップ領域およびページアウトしたページを管理する. これにより, カーネルは, 各計算機における最適なスワップ先を決定する. 以下, カーネルにおけるスワップ機構について述べる.

3.2.1 スワップ領域管理テーブル

スワップ領域管理テーブルは, スワップ領域の情報を保持するテーブルであり, 各計算機上に存在するすべてのスワップ領域を一括して管理する. スワップ領域管理テーブルの情報を以下に示す.

- デバイス ID

- i ノード
- 計算機 ID
- スワップ領域の ID

デバイス ID は, スワップを行うデバイスの識別子である. i ノードは, デバイス内に存在するスワップ領域の位置を示す. 計算機 ID は, 当該デバイスを接続している計算機に設定された ID である. スワップ領域の ID は, おのおののスワップ領域を識別するための正の整数値である.

3.2.2 スワップページ管理テーブル

スワップページ管理テーブルは, ページアウトしたページを管理するテーブルである. スワップページ管理テーブルの情報を以下に示す.

- オフセット
- 仮想アドレス
- スワップ領域の ID

オフセットは, スワップ領域内の位置を示す. このデータが指し示す位置にページアウトされたページの内容が存在する. 仮想アドレスは, ページアウトしたページの仮想アドレスである. スワップ領域の ID は, ページが保存されているスワップ領域の ID である. カーネルは, 以上のデータを使用することにより, すべてのスワップ領域およびページアウトしたページを管理する.

3.2.3 ページアウト先の決定

カーネルは, 各計算機ごとに最適なページアウト先を決定し, その情報を保持する. 以下にページアウト先の決定における処理を示す.

- 自計算機上にスワップ領域が存在する場合は, 当該スワップ領域をページアウト先とする.
- 自計算機上にスワップ領域が存在しない場合は, 最も近い計算機上のスワップ領域をページアウト先とする.

計算機間の距離に関しては, 応答時間の最も短い計算機を最も近い計算機とする. カーネルは, パケットを送信してから返答されるまでの時間をもとに最適の計算機を決定する.

3.2.4 スワップ領域の解放

カーネルがスワップ領域を解放する場合, スワップ領域内に存在するページを退避する必要がある. 退避先は, 以下の 4 通りが考えられる.

(1) 自計算機のメモリ

(2) 自計算機の他のスワップ領域

(3) 他の計算機のメモリ

(4) 他の計算機のスワップ領域

1 および 2 が使用できる場合は, 通信を発生させないことを目的として, これらを使用する. また, ページをディスクへ書き込む場合は, 先にメモリへ書き込む必要がある. この場合, スワップ領域へ書き込む処理が不要である. よって, メモリに空き領域がある場合は, 2 ではなく 1 を優先する. 1 および 2 がどちらも使用できないときは, 3 および 4 を利用する. この場合においても, 上記と同じ理由により 3 を優先する.

カーネルは, ページを移動させるときにスワップページ管理テーブルの情報を変更する. これにより, 次にページイン要求が発生したときは, 移動先のスワップ領域を参照することが可能となる. また, すべてのページの退避が終了すると, スワップ領域管理テーブルより当該領域を削除する.

3.3 スワップ機構の動作

ページアウトおよびページインの要求が発生した場合, 抽象化層とカーネルが協調して処理を行う. 以下, おのおのの動作について述べる.

3.3.1 ページアウト時の動作

ページアウト時の動作を以下に示す.

(1) 抽象化層は, ページアウト候補検索テーブルよりページアウトの対象となるページを取得する.

(2) 抽象化層は, 当該ページの状態を SWAPPED へ遷移させる.

(3) 抽象化層は, カーネルへページアウトの対象となるページの仮想アドレスおよび計算機 ID を通知し, ページアウトを要求する.

(4) カーネルは, あらかじめ決定しておいたスワップ領域内から, 使用されていない領域を検索する.

(5) カーネルは, 使用されていない領域へページを書き込む.

(6) カーネルは, スワップページ管理テーブルへ当該ページの情報を追加する.

(7) 抽象化層は, ページアウトしたページの物理メモリへの割付けを解除する.

以上の動作を行うことにより, 新たにページを割り付けることが可能となる.

3.3.2 ページイン時の動作

ページイン時の動作を以下に示す.

(1) ページアウトしたページへのアクセスが発生した場合, 抽象化層は, 当該ページの仮想アドレスを取得する.

(2) 抽象化層は, 当該ページの状態を SWAPPED から MASTER_RW へ遷移させる.

(3) 抽象化層は, カーネルへページインするページの仮想アドレスおよび計算機 ID を通知し, ページインを要求する.

(4) カーネルは, 抽象化層からページインの対象となるページの仮想アドレスおよび計算機 ID を取得する.

(5) カーネルは, 仮想アドレスをもとにスワップページ管理テーブルを検索する.

(6) カーネルは, スワップページ管理テーブルで得られたスワップ領域の ID をもとにスワップ領域管理テーブルを検索する.

(7) カーネルは, デバイス ID, i ノード, オフセットをもとにスワップ領域からページを読み出す.

(8) 抽象化層は, 読み出したページを物理メモリへ割り付ける.

以上の動作を行うことにより, ページアウトしたページを使用することが可能となる. 以上のように抽象化層とカーネル間で協調することにより, Solelc におけるスワップ機構が実現されている.

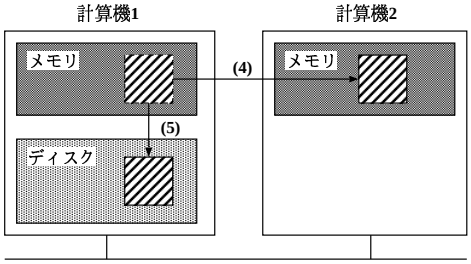

図 7 実験環境

表 2 ページアウトの処理時間

処理	時間
スワップキャッシュの利用	0.1ms
REPLICA ページの無効化	0.1ms
MASTER 権限の移動	0.3ms
他の計算機のメモリ上へのページアウト	2.7ms
自計算機のディスク上へのページアウト	9.1ms

4 評価

本章では, Solelc における空きページ作成の処理時間について述べる. なお, 性能評価には, Celeron 667MHz を搭載した PC/AT 互換機を 100Mbps のイーサネットで接続した環境を用いている. 評価対象の処理を以下に示す.

(1) スワップキャッシュの利用

(2) REPLICA ページの無効化

(3) MASTER 権限の移動

(4) 他の計算機のメモリ上へのページアウト

(5) 自計算機のディスク上へのページアウト

また, (4) および (5) の実験環境を図 7 に示す. (4) では, カーネルが計算機 1 のディスク上へページアウトしている. (5) では, カーネルが計算機 1 のメモリ上のページを計算機 2 のメモリ上へページアウトしている. 以上の動作により, 要求が発生してから空きページの作成が完了するまでに要する処理時間を表 2 に示す. (1) の処理に必要な時間は, ページイン後のメモリの更新の確認およびカーネルへの通知である. (2) の処理に必要な時間は, テーブル内

のリストの削除のみである．(3) の処理に必要な時間は，他の計算機への通信である．(4) の処理に必要な時間は，ACK 待ちに要する時間が処理時間の大半を占めている．(5) の処理に必要な時間は，主にディスクへの書込みに要する時間である．

表 2 において，上の 3 つの手法は，自計算機のディスク上へのページアウトと比較して処理時間が短い．そのため，これらの手法を用いることにより，ディスクへのページアウトにかかるオーバヘッドを軽減することが可能となる．このように，本スワップ機構は，ディスクを持たない計算機においても十分に実用的な速度でスワップを行うことができ，さらに複製ページが存在する場合は，空きページの作成を高速に行うことが可能であるといえる．

5 おわりに

本稿では，Solelc におけるスワップ機構について述べた．本機構を用いることにより，ディスクを持たない計算機においてもスワップを行うことが可能となった．また，空きページの作成を行うことにより，高速にメモリを確保することが可能となった．

今後の予定としては，MASTER 権限の譲渡による空きページの作成に関して，さらに検討する必要がある．カーネルが MASTER 権限を譲渡した後，直ちにアクセスが発生する場合も考えられる．そのため，今後，MASTER 権限を譲渡する条件について検討を行う予定である．

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] 芝公仁，大久保英嗣：“分散オペレーティングシステム Solelc におけるメモリ管理手法，” 情報処理学会論文誌 Vol.42, No.6, pp.1460-1471(2001).