システムソフトウェアと
オペレーティング・システム 85－6
（２０００．８．４）

分散オペレーティングシステムSolelcにおける負荷分散機構

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 †† 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，カーネルおよびプロセスは位置透過に動作可能である．これらを効率良く動作させるには，適切な負荷管理が必要となる．本稿では，Solelc における負荷管理の手法について述べる．本手法では，負荷分散と負荷均衡が協調動作することにより負荷管理を行う．これによって，カーネルやプロセスは，効率良く動作することが可能となる．

The Load Sharing Mechanism in Solelc Distributed Operating System

Koichi Nagamune † 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. The kernel and processes can work location transparency in Solelc. Work loading control is necessary to let OS work efficiently. In this paper, the mechanism of loading control in Solelc is described. Load sharing and load balancing cooperate for loading control in this mechanism. This mechanism makes it possible for kernel and processes to work efficiently.

1 はじめに

近年, 計算機の低価格化と高速ネットワークの普及に伴い, 複数の計算機をネットワークで相互に接続して利用する機会が多くなって来ている. しかし, 従来のオペレーティングシステム (以下 OS と記す) が管理する計算機資源は, その OS が動作する計算機上のものに限定されている. そのため, OS 上で動作するスレッドは, 他の計算機資源を有効に利用することが困難である. したがって, 分散環境において OS は十分な役割を果たしているとは言い難い. そこで, 我々は, 分散環境上で動作することを前提とした分散 OS Solelc[1] を構築している. Solelc は, ネットワーク上の複数の計算機を管理させることを前提として設計された OS である. 負荷管理は, 複数の計算機にまたがったシステムの応答性を向上させるために必要なものである. Solelc で管理される各計算機は必ずしも同じ性能ではないため, おのおのの計算機の性能と負荷を明確にしてスレッドを適切に配置する必要がある. Solelc では, このような負荷管理を, 負荷分散機構と負荷均衡機構によって実現している.

負荷分散は, スレッド生成時にスレッドを動作させる計算機を適切に決定することでスレッドの応答性を向上させるものである. 負荷均衡は, スレッドを移送することで負荷の偏りをなくし, システム全体の応答性を向上させるものである. しかし, 負荷分散では, スレッドの最適な配置は困難である. また, 負荷均衡では, 最適な配置はなしえるが, スレッドの実行を停止させて移送することから, 移送自体が効率を下げる.

Solelc では, 負荷分散と負荷均衡を協調させることによって負荷管理を行う. 負荷分散で行った配置の失敗を, 負荷均衡がフォローし, そして, 次の負荷分散で活かすことにより, スレッドをより最適な配置に近づける.

以下, 本稿では, 2 章で Solelc の概要を述べ, 3 章で負荷分散と負荷均衡の概要について述べ, 4 章では負荷分散機構の構成, 5 章では負荷均衡機構の構成について述べる. また, 6 章で関連研究について述べ, 最後に 7 章で本稿のまとめと今後の課題について述べる.

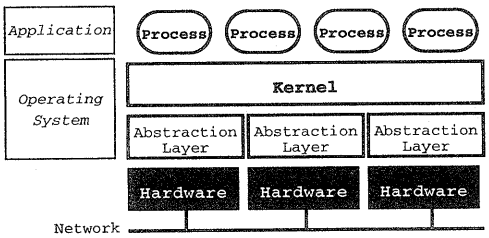

図 1 システムの全体構成

2 Solelc の概要

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 階層に階層化されている. 下位層の抽象化層は, 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を管理することができる. 同様に, アプリケーションも, 資源の位置を意識することなく動作可能である. Solelc では, アプリケーションを, プロセス・スレッドモデルを用いて実行する. プロセスとは, コードやデータから構成されるスレッドの実行環境である. 抽象化層によるメモリ資源の抽象化により, すべての計算機で 1 つの仮想アドレス空間が共有される. すなわち, プロセスやカーネルのメモリ領域では, すべての計算機上で同一の内容を読み書きすることが可能である. したがって, スレッドは任意の計算機上で動作可能であり, また, 同一のプロセスに属するおのおののスレッドがそれぞれ異なる計算機で動作することも可能である.

スレッドは, カーネルのグローバルスケジューラと抽象化層のスレッド管理部によって管理される. スレッドは, レジスタ, 優先度, 状態, 計算機 ID などの情報を持つ. 計算機 ID とは, 計算機に対し一意に与えられる識別子である. 通

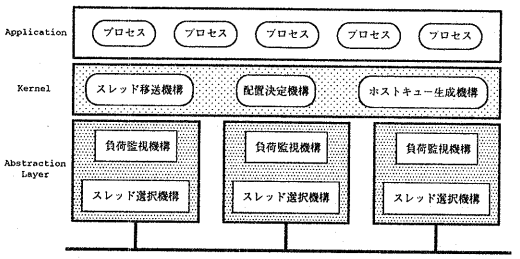

図 2　負荷管理機構の構成

常, スレッドの計算機 ID は, グローバルスケジューラによって決定され, スレッド管理部は, グローバルスケジューラによって指定された計算機上でスレッドを動作させる. Solelc における負荷管理は, 抽象化層が発生させるイベントを受け, グローバルスケジューラがスレッドを適切に配置することによって実現される.

3　負荷管理の概要

Solelc では, スレッドは, 位置透過なメモリアクセスが可能であるため, 任意の計算機上で動作することができる. しかし, 無秩序にスレッドの配置を行えば, 負荷に偏りが生じスループットが低下してしまう. Solelc では, 負荷分散と負荷均衡を用いて, スレッドの配置を決定することでシステム全体のスループットを向上させている. 負荷分散は, スレッド生成時にスレッドを配置する計算機を適切に決定することによって実現される. また, 負荷均衡は, システム全体の負荷に偏りが生じたときに, スレッドを再配置することによって負荷の均衡を図るものである. Solelc では, 負荷分散を主に行い, 負荷均衡をシステムとして最悪時の処理としている. 最悪時の状況を再び繰り返さないために, 最悪時の状況の原因となったスレッドに関する情報を記憶しておき, 以降の負荷分散に利用する. これによって, より良いスレッドの配置が実現される.

カーネルでは, 主にスレッドを動作させる計算機の決定を行い. 抽象化層では, 負荷の監視を行う. Solelc では, 図 2 に示すような機構によって負荷管理を行う.

- 負荷監視機構
  定期的に自計算機の CPU 負荷, メモリ負荷, 通信負荷を計測する. 負荷情報に変更があれば, ホストキュー生成機構にこれを通知する. また, 負荷が著しく高いときは, スレッド選択機構にスレッドを移送するように要求する.

- ホストキュー生成機構
  ホストキューとは, 残存処理能力の順番に計算機を並べたものである. ホストキュー生成機構は, 各計算機から集められた CPU 負荷, メモリ負荷, 通信負荷から判断し, ホストキューを生成する. ホストキューは, 配置決定機構, スレッド移送機構により参照される.

- 配置決定機構
  実際に負荷分散を行う機構である. スレッド生成時に呼び出され, スレッドを配置する計算機を決定する. このとき, 適切な計算機を選択するために, スレッドの負荷情報およびホストキューを参照する.

- スレッド選択機構
  計算機の負荷が著しく高いときに, 負荷監視機構により呼び出される. 問題のスレッドを特定し, 当該スレッドの移送をスレッド移送機構に要求する.

- スレッド移送機構
  実際に負荷均衡を行う機構である. スレッド選択機構の要求を受けスレッドの移送先を決定する. このとき, スレッド選択機構により提供される情報をもとに, ホストキューを参照し条件を満たす計算機を選択する. また, 移送したスレッドの負荷情報を共有領域に保持する.

4　負荷分散機構

スレッド生成時に新たなスレッドを配置する計算機を適切に決定することによって負荷分散が実現される. スレッド配置の決定には, 負荷監視機構, ホストキュー生成機構, 配置決定機構が使用される. 本章では, 以下, これら 3 つの

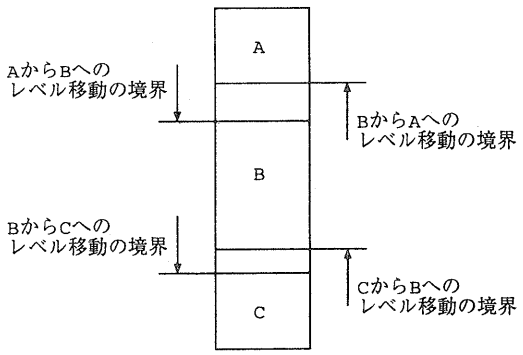

図 3　レベルの境界線

機構を使用した負荷分散の方法について述べ，最後に負荷分散の処理の流れについて述べる．

4.1　負荷監視機構

Solelcが管理する各計算機の処理能力は必ずしも同じではない．スレッドのより良い配置を決定するには，各計算機の残存処理能力に関する情報が必要となる．残存処理能力は，計算機の性能(静的な情報)と現在の負荷(動的な情報)から求める．

負荷監視機構は，各計算機の抽象化層で1つずつ動作する．負荷監視機構は，タイマ割り込みのハンドラ内で，定期的に自計算機の負荷情報を確認する．負荷情報に変更がある場合は，イベントを発生させ負荷情報をホストキュー生成機構に知らせる．また，負荷が著しく高い場合は，スレッド選択機構を呼び出し負荷均衡を行う．

負荷管理機構が，負荷を取得するときに問題となるのは，情報の精度である．厳密な負荷情報を集めることは，現実的ではない．なぜなら，厳密な値は刻一刻と変化するが，このような情報の維持は非常に負荷が大きいためである．したがって，取得した情報をレベルに分割して管理することで，頻繁に情報が変更されることを防ぎ，情報の維持の際にかかる負荷を軽減する．また，レベルの切り替わる境界線付近で負荷が変動する場合の，頻繁なレベルの変更を防ぐために，図3のようにレベルの境界に差をつける．AからBへ負荷が移行する場合と，BからAへ負荷が移行する場合では，レベルの切り替わる境界線が異なる．このようにすれば，負荷情報がBからAに上がった直後に僅かに下るような場合でも，すぐにレベルが切り替わることもなく，また情報の精度にも大きな問題はない．

負荷監視機構は，CPU，メモリ，通信の3種類の負荷を取得する．より最適なスレッドの配置を行うためには，より低負荷の計算機という情報だけでなく，計算機の性能も必要となる．

CPU　CPUの性能と実行可能状態のスレッドの数から，残存処理能力を求める．

- CPUの性能
  OSの起動時に測定．CPUが単位時間当たりに処理可能な命令数(MIPS)をCPUの性能とする．
- 実行待ちスレッドの数
  実行可能キューにスレッドがつながれるときと削除されるときにキューの数を調べ，その数を現在の負荷とする．

メモリ　メモリの確保および解放時に，空き物理メモリ量を調べ，これを残存処理能力とする．

通信　NIC(Network Interface Card)の性能と単位時間当たりの送受信回数から，残存処理能力を求める．

- NICの性能
  通信速度を性能とする．
- 単位時間当たりの送受信数
  単位時間あたりの送受信回数を通信負荷とする．

CPU残存処理能力は，スレッドのスループットを向上させるための配置に使われる．メモリ残存処理能力は，スレッドを実行するだけのメモリがあるか否かを調べるために使われる．グローバルスケジューラは，これらの情報を使用することによって適切な計算機を選択することが可能となる．同様に，通信残存処理能力も，グローバルスケジューラがスレッドを動作させる

計算機を決定する際に使用される．グローバルスケジューラが，スレッドの配置を適切に行うことによって通信負荷の軽減が可能である．例えば，頻繁に通信し合うスレッドを同一の計算機上で動作させることによって，通信負荷の軽減が可能となる．

4.2 ホストキュー生成機構

負荷分散と負荷均衡において問題になるのが，計算機の選択である．負荷が高くなったときに計算機を探す送信者起動分散発見的アルゴリズムでは，計算機を探すこと自体が大きな負担となる．受信者起動分散発見的アルゴリズムでは，負荷の低い計算機が仕事を探す [2]．Solelc では，カーネルの動作のために頻繁に通信が行われるため，無駄な通信は OS にとって大きな負荷となる．そこで，スレッドの生成または移送に直接関わらない第 3 者が，計算機を選択する処理を行うことにより，スレッド生成，移送を行う計算機に負担をかけないようにすることが考えられる．ホストキュー生成機構は，このような手法を実現するための機構である．

ホストキュー生成機構は，カーネル内で動作し，抽象化層の負荷管理機構が発行したイベントを処理する．ホストキュー生成機構は，負荷管理機構が提供する CPU，メモリ，通信の負荷のレベルから，CPU 残存処理能力，メモリ残存処理能力，通信残存処理能力の順番の優先度で計算機のキューを生成する．このキューを負荷分散と負荷均衡に用いることで，計算機の選択の手間を軽減し，より良い配置を行うことが可能となる．

ホストキューの構成を図 4 に示す．ホストキューは，まず CPU 残存処理能力のレベルによって計算機を分類する．CPU 残存処理能力で分類した後に，メモリ残存処理能力の高い順番にキューを生成する．同じメモリ残存処理能力の場合は，通信残存処理能力の高い順番で並べる．各残存処理能力のレベルが全て同じであれば，同じレベルの計算機の最後につなぐ．

4.3 配置決定機構

配置決定機構は，スレッドの生成時に，これを配置する計算機を決定する機構である．スレッドを適切な位置に配置することによって，

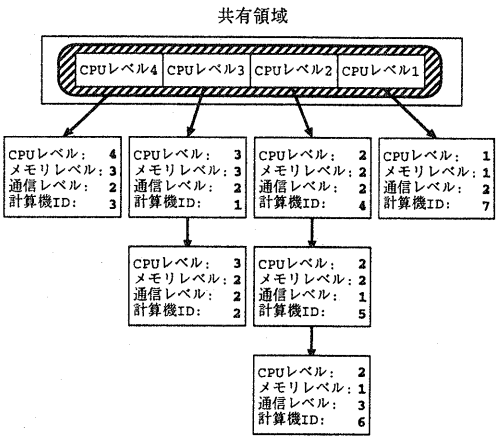

図 4 ホストキュー

負荷分散が可能となる．

配置決定機構は，スレッドを動作させる計算機を決定する際にスレッドの負荷情報を調べる．スレッドの負荷情報とは，スレッド移送機構により生成される，スレッドが計算機にかけた負荷の情報である．負荷の情報として，CPU 負荷と通信負荷がある．スレッドの負荷情報をもとに，生成するスレッドの条件に合致する計算機をホストキューから First Fit で取り出す．スレッドの情報がない場合は，キューの先頭の計算機を取り出す．処理の最後に，計算機を同じ CPU 残存処理能力レベルのキューの最後につなぐ．

4.4 負荷分散の処理の流れ

図 5 は，以下に示すような，スレッド生成時の処理の流れを示している．

1. スレッド生成システムコールが発行される．
2. グローバルスケジューラが，システムコールを取得する．
3. グローバルスケジューラは，抽象化層のスレッド管理部を用いてスレッドを生成する．
4. グローバルスケジューラは，配置決定機構に計算機 ID を要求する．

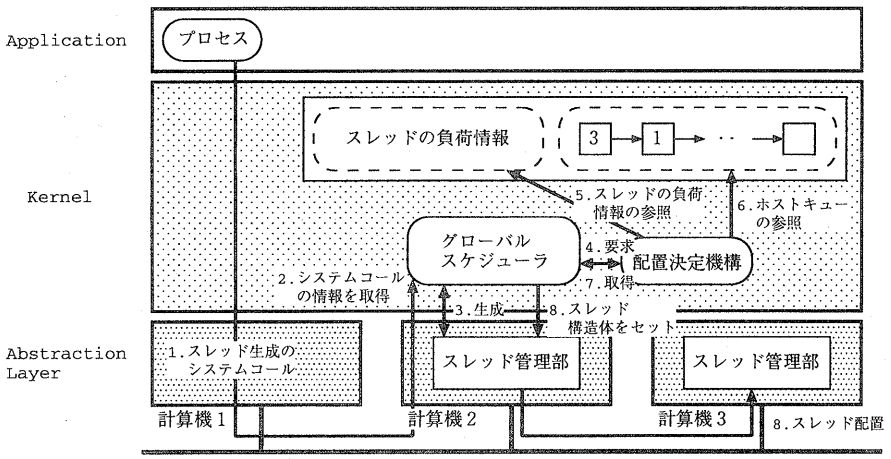

図 5　負荷分散の処理の流れ

5. 配置決定機構は, スレッドの負荷情報を参照する.

6. 配置決定機構は, ホストキューから適切な計算機を選択し, 計算機 ID を返す.

7. グローバルスケジューラは, 生成されたスレッドの, レジスタ, 優先度, 状態, 計算機 ID を決定し, これをスレッド管理部に渡す.

8. スレッド管理部は, 指定された計算機上でスレッド動作させる.

このように, ホストキューを使うことで, 計算機を探す手間を大幅に省くことが可能である.

5　負荷均衡機構

負荷均衡では, いつ, なにを, どこへ移送するかが問題になる. Solelc では負荷均衡を最悪時の処理と位置付けており, 「いつ」は最も負荷が高いときである. 「どこへ」移送するかは, ホストキューを使用し計算機を選択することで解決する. 問題として残るものは, 「なにを」であるが, 本機構はスレッドを対象にしているため, どのスレッドを移送するかが問題になる. Solelc における負荷分散と負荷均衡は, 協調し動作する. 協調動作させる目的は, 最悪時の処理で負荷均衡を用い, 最悪の状態の原因を作ったスレッドの特徴を記録しておき, 同じような配置を再度行わないようにすることである. 以上の理由から, 本負荷均衡では問題を起こしているスレッドを移送することにする. これによって負荷均衡が実現される. 以下, 本章では, スレッド選択機構およびスレッド移送機構を使用した負荷均衡の方法について述べ, 最後に負荷均衡の処理の流れについて述べる.

5.1　スレッド選択機構

スレッド選択機構は, 各計算機の抽象化層で動作し, 負荷監視機構から呼び出される. 負荷監視機構は, 残存処理能力が最も低いレベルに遷移したとき, スレッド選択機構を呼び出す. スレッド選択機構は, 負荷監視機構から渡された負荷の種類 (CPU, メモリ, 通信) をもとに, 問題となっているスレッドを選択する. また, 選択したスレッドの情報とスレッドの負荷情報をスレッド移送機構に通知し, スレッドの移送を要求する.

CPU 負荷が高い状態にある場合, スレッド選択機構は, CPU 使用率を確認し, アイドルスレッドを除く最も CPU を使用しているスレッドを選択する.

メモリ負荷が高い状態にある場合は, メモリ不足によって実行できなくなったスレッドを選択する. 実行に必要なメモリは, ローカルの計算機に転送を行う. ローカルの計算機の物理メ

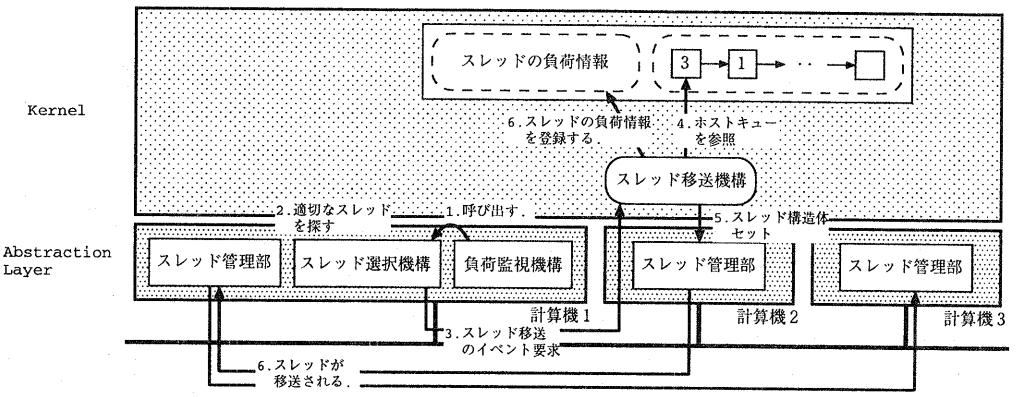

図 6　負荷均衡の処理の流れ

モリ量がない場合は, 実行できないからである.

通信負荷が高い状態にある場合は, 単位時間当たりのページフォルト回数の最も多いスレッドを選択する. スレッド同士の通信は, 共有領域を用いたメモリの読み書きで行われる. 共有領域は, アクセスしたページがなかった場合は, 通信を用いてページ転送を行うからである.

5.2　スレッド移送機構

スレッド移送機構は, スレッド選択機構からの要求に従いスレッドの配置を決定する負荷均衡のための機構である.

スレッド移送機構は, スレッド選択機構から渡されたスレッドの計算機 ID を他計算機に変更し移送を行う. CPU 負荷や通信負荷による移送要求に関しては, スレッド選択機構から提供されたスレッドの負荷情報の要求を満たす計算機を First Fit でホストキューから選択する. 該当するものがない場合は, 最高レベルの CPU 残存処理能力のキューの先頭の計算機を取り出す. 最高レベルのものがない場合は, 移送を行わない. メモリ負荷による移送要求の場合は, 最もメモリ残存処理能力のある計算機を選択する. 最低レベルのものしかない場合は, 移送を行わない. 最後に, 選択した計算機を同じ CPU 残存処理能力レベルのキューの最後につなぐ. また, スレッド移送機構は, 移送されたスレッドの負荷情報を共有領域に保持する. 負荷の情報として以下のものがあげられる.

- CPU 負荷
  当該スレッドの単位時間当たりの実行した命令数
- 通信負荷
  単位時間当たりのページフォルト数

以上の情報を記録しておき, 以降の配置決定に利用することで, 最適に近い負荷分散を行うことが可能である.

5.3　負荷均衡の処理の流れ

図 6 は, 以下に示すような, スレッド移送時の処理の流れを示している.

1. 負荷管理機構が, スレッド選択機構を呼び出す.
2. スレッド選択機構が, 適切なスレッドを選択する.
3. スレッド選択機構が, スレッド移送要求のイベントをスレッド移送機構に発行する.
4. スレッド移送機構は, スレッド選択機構から受け取ったスレッドの負荷情報の要求を満たす計算機をホストキューから探し, スレッドの計算機 ID を変更する.
5. スレッド移送機構は, スレッドを抽象化層のスレッド管理部に渡す.
6. - スレッドの負荷情報を共有領域に記録する.

- スレッド管理部が, 指定された計算機 ID の計算機にスレッドを配置し, 実行する.

このように, ホストキューを利用することで計算機を探す手間を大幅に省くことが可能である. スレッドの負荷情報を記録することで, 次回の負荷分散の配置をより良いものに変えていくことが可能となる. すなわち, Solelc が動作していく中で徐々にスレッドの配置が最適化されていく.

6 関連研究

ホストキュー生成機構のように, 残存処理能力順に計算機を並べキューとする手法として, 並列ジョブのスケジューリング手法 [3] がある. これは, 並列実行を目的としたスケジューリングであるが, アイドルプロセッサをキューとして管理し, ジョブを配置するときにこのキューを使用する. 並列実行を目的としているため, ジョブの情報として使用プロセッサ数があり, キューからジョブが要求するだけプロセッサを与える. しかし, このリストはアイドルプロセッサをまとめたもので, Solelc のように計算機毎に性能や負荷の状況が異なる環境で, この手法を適用することは難しい. 本論文では並列処理には触れていないが, この手法を Solelc の負荷管理に取り入れることで, さらに応答性を良くすることが可能である.

7 おわりに

本論文では, 分散オペレーティングシステム Solelc における負荷分散と負荷均衡機構について述べた. Solelc における負荷管理は, 負荷分散を主に行い, 最悪事に負荷均衡を用いている. このように, 負荷分散と負荷均衡を協調させることによって効率的なスレッドの実行が可能となる.

現在, 負荷を CPU, メモリ, 通信に関するもののみとしているが, 今後はディスク等の周辺デバイスも考慮した負荷管理についての検討を行う必要がある.

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).

[2] Andrew S. Tanenbaum 著, 水野忠則, 鈴木健二, 宮西洋太郎, 佐藤文明 訳 : “分散オペレーティングシステム,” 株式会社プレンティスホール出版 (1996).

[3] 合田憲人, 笠原博徳, 成田誠之助: “マルチプロセッサシステム上での並列ジョブのスケジューリング手法の評価,” 情報処理学会研究報告 1996-OS-73, pp. 73–78 (1996).