分散OS DM-1における障害対策のためのシステム停止機能

4F－7

浜田 彰夫 †　岡部 寿男 †　大久保 英嗣 ††　津田 孝夫 †
† 京都大学工学部情報工学科　†† 立命館大学理工学部情報工学科

1　はじめに

分散OSでは、OSに必要な情報が複数のノードに分散されていることが多いので、システムの動作中に安易に一つのノードの電源を停止すると、他のノードにおいてOSとしての機能を続行させることが不可能となる場合がある。また、停電などによりすべてのノードの電源が一斉に停止すると、再び電源を投入しても以前の状態を回復することは通常不可能である。本研究では電源異常等によってシステムに障害が発生することを防ぐことを目的としている。

電源の供給が予告無しに停止すると仮定すると、いつ電源が停止してもその時のシステムの状態情報を復活できる必要があり、したがって任意の時点におけるシステムの状態情報をすべて二次記憶等に保存しなければならないが、これは非常に効率が悪い。そのため、ここでは電源が停止するまでにある程度の時間的余裕があると仮定する。たとえば、突然に停電が起こった場合でも、自動的に予備電源が入り、残り数分間だけは電源が供給されるという場合などである。このような場合にシステムの実行状態を凍結したり、一部のノードを切り離したりすることによって、システムを安全な状態にとどめる方式に関して論ずる。

我々は、現在開発中である分散OS DM-1においてこの機能を実現する予定である。

2　分散OS DM-1

我々は分散仮想記憶を用いたOSとしてDM-1を開発中であり、i386及びi486を搭載したパーソナルコンピュータ上で実装を行なっている。

分散仮想記憶とは、システム内に存在する唯一の仮想記憶空間であり、その実体はシステムに接続されたすべてのノードの主記憶と二次記憶に分散している。システムのどのノードからもこの仮想記憶空間が見えており、たとえばあるノードであるアドレスに書き込みを行なった後、別のノードでそのアドレスを読み込めば、書き込まれた値が読める。この様に、分散仮想記憶を利用することによって、位置透過性をはじめ様々な透過性が提供される。また分散仮想記憶はマイクロカーネルで実現されているので、それらの透過性はユーザのみならずOS自身にも提供される。

分散仮想記憶におけるメモリの実体はページ単位で管理され、一つのページの複製が複数のノードに作られる時は、write-invalidate 方式を用いて整合性の制御を行なう。

またDM-1ではタスク・スレッドモデルを用いており、一つのタスク内で複数のスレッドが走ることが可能である。システムタスクの一つであるスレッド分配機構によって、スレッドは自動的に負荷が分散されるように、またサイト間の通信量ができるだけ少なくなるように、適当なノードに配置され、動的に移送される。

3　ローカルシャットダウン機能

システムにノートパソコンを接続して使用しているときにそのバッテリーの容量が残り僅かになった場合など、ある一定の時間後にあるノードの電源が停止するということがあらかじめ分かっている場合、そのノードを切り離してシステムの動作を継続することが考えられる。また、ノードを切り離す機能は保守性や可用性を向上させるためにも必要である。

システムの動作中に、あるノードを切り離すことをローカルシャットダウン[1]と呼んでいる。

あるノードのバッテリーの残り時間がある一定時間以下になるなど、システムがこのノードを切り離す必

System-shutdown for Fault-tolerance in the Distributed Operating System DM-1
Akio Hamada, Yasuo Okabe, Eiji Okubo, Takao Tsuda
Kyoto University
Yoshida honmachi, Sakyo-ku, Kyoto, 606-01, Japan

[1] シャットダウンとは通常システム全体を停止する場合に用いるが、ここではシステム全体としては実行を継続させ、そのノードだけを停止させることを意味する。

要があると判断した場合や、直接ユーザから指示があった場合などにローカルシャットダウンが行なわれる。

ローカルシャットダウンの手順は以下の通りである。

1. スレッド分配機構を用いて切り離すノード上のスレッドを別のノードへ移送する。
2. システム内で切り離すノード上にしか存在しないページを他のノードへ移送する。
3. 切り離されるノード以外のカーネルの持っている整合性制御のための情報などを書き変える。

この時点で切り離そうとしているノードはシステムから論理的に切り離されており、電源を落としても何ら問題のない状態になっている。

ノードを切り離す際、そのノードに接続されているデバイスのドライバなどのスレッドは他のノードに移送する必要がない。また他のスレッドと通信をしているスレッドなどは、単に他のノードへ移送するだけでなく、その振舞いを変える必要がある場合がある。この様に、切り離すノード上のスレッドは単純に他のノードへ移送すればよいとは限らない。そこで、ローカルシャットダウン時には、システムタスクの一つであるシャットダウンサーバが、切り離すノード上のスレッドにシグナルを送り、シグナルを受けとったスレッドは自分で判断して終了したり振舞いを変えたりする。

今のところ、各ノードの二次記憶の容量、あるいはシステム全体の容量に関しては考慮していない。すなわち、切り離すノード以外のすべてのノードの二次記憶が満杯で、どこにもページを移送できない、といった場合は考慮に入れていない。このような場合に関しては、一部のメモリオブジェクトを切り離すなどの処理が必要になってくると思われるが、これは今後の課題である。

4 グローバルフリーズ機能

停電時に自動的に予備電源が作動した場合など、システムのすべてのノードの電源がある一定時間の後に停止することがあらかじめ分かっている場合、システムの状態を凍結させて、一旦電源を停止した後に凍結前の状態を再現することは、電源異常による障害の対策として有効である。DM-1では、システムの状態を凍結させ、電源を停止することをグローバルフリーズと呼んでいる。

システムの状態を決定するのは、仮想記憶空間のメモリオブジェクトとスレッドの現在の状態である。グローバルフリーズ時にはこれらの情報をすべて二次記憶に退避させることが必要である。しかしスレッドの実行状態もメモリオブジェクトとして管理されているので、凍結時にはメモリオブジェクトのみを二次記憶に退避すればよい。ここで、分散仮想記憶ではメモリの実体はすべてページ毎に管理されており、それらのページはどこかのノードの主記憶か二次記憶、またはその両方に存在している。新たに二次記憶に書き戻す必要があるのは主記憶のみに存在するページである。すなわち明示的にそれらのページをページアウトする。そしてそのページテーブルを含め、カーネルが持っている情報 (これらは分散仮想記憶とは別に管理されている) を二次記憶に保存する。

状態の回復時にはそれらの情報をまず読み込み、カーネルの状態とページテーブルを復活させる。これが凍結時に接続されていたすべてのノードで行なわれると、スケジューラが起動され、凍結していたスレッドを起動する。

5 おわりに

本稿では、分散OS DM-1におけるノードの切り離し機能とシステムの状態の凍結という二つの手法を障害対策に応用する方向でまとめているが、当然これらの手法は障害対策のみならず、可用性や保守性を高めることにもつながる。

従来のOSレベルにおける障害対策の研究では、障害が発生した後の資源や実行状態などの回復などには重点を置いているが、障害の発生を防止するという根本的な部分に関わるものが比較的少ない。本稿では障害の原因として最も単純であると考えられる電源異常を仮定して、その対策に関して論じた。

参考文献 前川守、所真理雄、清水謙多郎:分散オペレーティングシステム UNIX の次にくるもの, 共立出版.