1L-6

プロセスを分散実行するためのアドレス空間管理

深尾 拓司† 三添 匠†† 芝 公仁† 岡田 至弘†
†龍谷大学理工学部 †龍谷大学大学院理工学研究科

1 はじめに

近年，複数のスレッドを使用して動作するアプリケーションが増加している．しかし，一般的なシステムでマルチスレッドのプロセスを実行しても単一の計算機上でしかそのプロセスを動作させることができない．我々プロセスを複数の計算機で共有し，プロセスが持つスレッドを分散実行することの実現とを目的にプロセス共有機構を構築している．これにより，計算機の資源を有効利用することが可能となる．

現在，我々が開発を行っているプロセス共有機構では，単一のプロセスをネットワークに接続されている複数の計算機上で共有し，動作させることが可能である．これは，プロセスの持つアドレス空間を他の計算機と共有することで実現される．複数の計算機で共有するプロセスを監視し，各計算機で通信を行いアドレス空間を拡張されたカーネルのメモリ管理機能により制御する．共有するプロセスのアドレス空間を他の計算機と共有し，プロセスがアドレス空間へ書き込みを行う際に，アドレス空間を読み出す他の計算機に矛盾が生じないように管理する．このようなメモリを分散共有するシステムにおいて効率化を計る点となるのは，書き込みに対する管理である．

本稿では，単一のプロセスを複数の計算機で実行するためのメモリ管理手法について，その機能の効率化について述べる．本手法により，読み込みや書き込みのアクセスが共有しているアドレス空間の一部に集中するようなプロセスであっても，スループットの低下を抑えた計算をすることが可能となる．

2 プロセス共有機構

現在，研究を行っているプロセスを共有し実行する機構は，プロセスのアドレス空間を制御するメモリ制御部と，システムコールを適切な計算機に転送するシステムコール制御部をもつ．今回はメモリ制御部について述べる．メモリ制御部の構成を図 1 に示す．また，プロセスを共有しているネットワーク上の計算機を，これより本稿ではノードと呼称する．

3 メモリ制御機能

メモリにデータが存在しない場合，カーネルにおいてページフォールトが発生する．拡張されたカーネルは，管理しているメモリ領域でページフォールトが発生すると，ページ内容を持つノードから適切なページ内容のデータを取得しページフォールトを解消させる．これにより，プロセスはメモリアクセスを位置透過に行うことが可能になる．

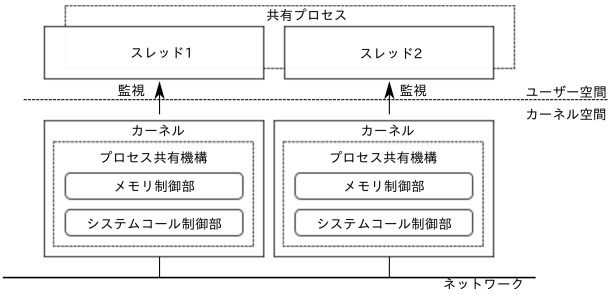

図 1 システム構成

3.1 ページの状態遷移

一貫性制御は順序一貫性モデルを参考にしており，ページの内容や複製されたページを取得するまでの流れをカーネルはページの状態として管理している．共有されるプロセスのアドレス空間はページ単位で制御される．各ページの状態は，必ずの以下の 2 つの状態をとる．1 つ目は，読み書きが行えるノードが 1 台存在し，他の全てのノードは読み書きが行えない状態である．2 つ目は，複数のノードが読み出し可能で，他の全てのノードが読み書き不可能な状態である．後者の方は，読み出し可能なノードのうち 1 台が通信においてページの内容を管理の中心となる役割を持っている．

- INVALID
  ページの内容を持たず，読み書きができない
- READ ONLY
  ページの内容を持ち，読み出しのみ可能
- REPLICA
  READ ONLY の複製を持ち，読み出しのみ可能
- READ WRITE
  ページの内容を持ち，読み書き可能
- REPLICATING
  READ ONLY の複製の取得中
- DELETING REPLICA
  すべての REPLICA を無効化中

INVALID 状態はページの内容を所持していない状態である．INVALID 状態は読み書きが禁止されており，アクセスが行われるとページフォールトが発生する．もしページにアクセスが行われると該当ページの内容を持ってるノードから通信でページを取得する．READ_ONLY 状態はページの内容をもっており，ページの複製を他のノードに渡す事ができる状態である．

Address Space Management for Processes Shared by Multi-nodes

Takumi Mizoe†, Shouta Kokaji††and Masahito Shiba†and Yoshihiro Okada†
†Faculty of Science and Technology, Ryukoku University ††Graduate School of Scientice snd Technology, Ryukoku University

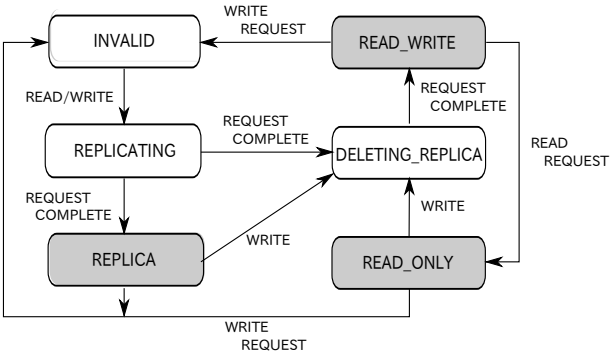

図 2　ページの状態遷移図

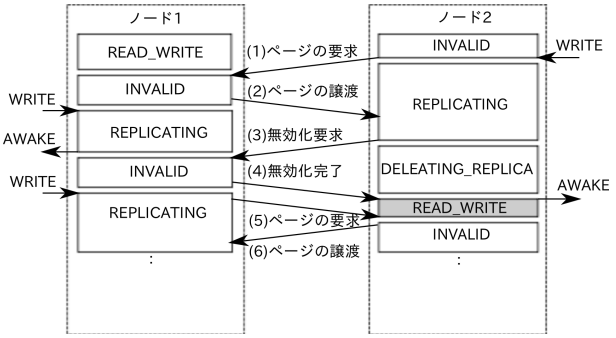

図 3　ページの競合状態

REPLICA 状態はページの複製を持っている状態である．READ_ONLY 状態，REPLICA 状態では書き込みは禁止されており，書き込む際には無効化要求を全ノードを送信する必要がある．READ_WRITE 状態はページの複製を持ち，他の全てのノードは INVALID の状態にある．READ_WRITE は読み書きが可能である．REPLICATING は INVALID 状態から遷移した場合，そのアクセスの要求をページ内容を所持しているノードに送信する状態である．DELETING_REPLICA は書き込み禁止のページに書き込もうとしたとき，全ノードに無効化要求のを通信行う状態である．

4　ページ内容の取得の競合

単一のページに複数の計算機から書き込みが集中した場合，READ_WRITE のページを持つノードが連続して変更され，メモリの一貫性制御の処理の負荷が高くなる．極端な場合には，一貫性制御の処理ばかりが行われ，共有プロセスが事項されない．READ_WRITE 状態に遷移させた直後に，他ノードからページ内容の要求を受け取り，共有プロセスが実行される間に INVALID 状態などに遷移させてしまうことがある．このような遷移が複数のノードで起こり競合状態が続くと，共有プロセスの実行がほとんど行われないようになる．このような乗号を検知し，競合が共有プロセスの実行を妨げられることがないようにすることによって，システム全体の効率化を実現することができる．

5　競合の検出と効率化

検出の手法として，回数から求める手法，時間から求める手法，ページの再送要求から求める手法の 3 点挙げられる．始めに，単位時間あたりのページの状態の遷移回数から求める手法がある．この競合が発生したときに，ページの遷移回数が極端に増加する特徴がある．したがって，遷移回数が一定の閾値を超えた場合，ページ内容の取得における競合が発生していると判断する．次に，平常時のページ内容を取得してからページ内容の送信が発生までの時間から求める手法がある．ノードがページ内容を取得してから送信するまでの平均時間を求めておき，平均時間から一定数の割合より短い時間でページ内容を送信した場合，競合が発生していると判断する．最後に，ページの再送要求から求める手法がある．この競合は，複数のノードが絶え間なくページ要求を行っている場合に発生する．ページ要求に対して再送が行われる場合，READ_WRITE に遷移したとき競合が発生することを予想することができる．

次に効率化の手法として，ページ内容の送信を遅らせる手法と，ページ内容の要求を遅らせる手法の 2 点挙げられる．始めに，ページ内容の要求に対する返信を遅らせる方法がある．返信を遅くすることで，ノードはページに対して書き込みを行うまでの時間が確保する．次に，再取得の要求を遅らせる手法がある．再取得の送信を遅くすることで，送信先のノードはページに対して書き込みを行うまでの時間が確保する．ただし，この手法をとった場合，本機構の構造上，一定時間ユーザプロセスがスリープすることになるため本機構以外の処理のパフォーマンスを低下させる．従って，前者のページ内容を送信する時間を遅らせる手法を採用する．

以上より，検出は再送要求から求める手法を用い，効率化はページの送信を遅らせる手法を用いる．ただし，検出についてはページ要求が受信できなかった場合の対策として以下の時間から求める手法を合わせた手法をとる．まず効率化の手法において，最も理想的なページ要求の受信からページ内容の送信までの遅延時間を求める．その遅延時間を閾値とし，ページ状態の受信時，ページ状態を取得した状態からの時間が遅延時間を下回った場合は，必ず競合が発生していると見なす．

6　おわりに

本稿では，共有するプロセスのアドレス空間を管理するメモリ管理手法と効率化の手法について述べた．本機構により，単一のページに対して書き込みが集中するプロセスにおいても，スループットやパフォーマンスの低下を抑えることができると考えられる．

参考文献

[1] 三添　匠，小鍛治　翔太，芝　公仁，“ プロセスを分散実行するためのシステムコール制御手法，” 情報処理学会研究報告．[ハイパフォーマンスコンピューティング] 2011-HPC-132(33), 1-7, 2011.