

PDE-IIにおけるリソースのリザベーションに関する考察

1H-7

松尾聡^{1,2} 岡村耕二² 荒木啓二郎² 福田晃^{2,3}

¹ミノルタカメラ株式会社 西神情報センター

²奈良先端科学技術大学院大学 情報科学研究科

³九州大学工学部 情報工学科

1. はじめに

近年、ワークステーションやパーソナルコンピュータの高性能化、ネットワークの大容量化が進んできた。これらにより、ネットワークを介したマルチメディア処理が実用的になった。しかし従来の処理とは異なり、マルチメディア処理においては個々のメディアは連続的に処理にされ、複数の処理間の関連性がある。また、現在のOSではマルチメディア処理の制約に対する機能が備わっていない。そこで、我々はマルチメディア処理のためにOSにそれらの機能を組み込んだPDE(Parallel Distributed Environment)-IIの開発を行なっている^[1]。

本稿では、PDE-IIでの同期機構^[2]を行なう周期スレッド^[3]のリソースのリザベーション方式について述べる。

2. リソースサーバによるリソース管理

本稿では、マルチメディア処理のために扱うリソースとして、CPU、ネットワーク、メモリを考える。これらのリソースは性質が異なるが、抽象化することで同様に扱う。

また、リザベーションの定義は以下の通りである。

- リソース毎に設定するリソースサーバがそれぞれのリソースを定量化する。
- リソースサーバが要求された周期性を満たす割り当てを行なう。

個々のリソースでのリザベーションは、CPUの場合はCPUのプロセスへの割り当て、ネットワークの場合はネットワークバンド幅を確保すること、メモリの場合はメモリ領域を確保することである。

これらの処理のためにリソース毎にリソースサーバを作成する。リソースサーバは各リソースをそれぞれ管理する。そのため、リザベーションはリソースサーバを介して行なわれる。また、処理内容によっては複数リソースのリザベーションを行なう場合があり、その際にはリソースサーバ間で調停を行なう(図1)。

3. CPUのリザベーション

本稿では、2章の定義に基づきCPUのリザベーションについて具体的に述べる。

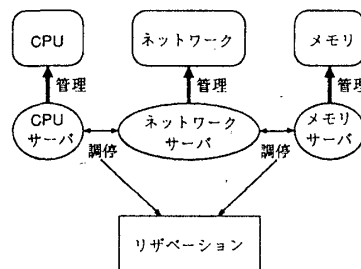


図1: リソースサーバ

3.1 CPUサーバ

CPUサーバは、CPUのリザベーションを行なう。CPUサーバの処理は以下の通りである。

- CPUの定量化
- CPUのリザベーション
- CPUの管理

CPUの定量化は、一定時間にコンテキストスイッチを起こすために行なう。そのためには、CPUの処理能力を数値化して表すことが必要である。実際には時間当たりの処理量で表し、この時間をCPUの周期時間と定義する。例えば、MIPSでは1秒当たりの処理数で数値化している。

コンテキストスイッチを起こす一定時間を単位時間とし、この単位時間でCPUを時分割し、時分割されたCPUの1つをリザベーションの単位(処理量/時間)とする(図2)。

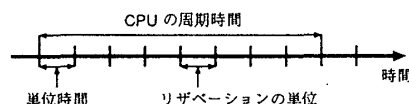


図2: 時分割したCPU

3.2 CPUサーバ実現における問題点

CPUサーバの実現における問題点は次の2つである。

- 周期性
- カーネルモード

"Study on Resource Reservation for PDE-II"

S.Matsuo^{1,2}, K.Okamura², K.Araki² and A.Fukuda^{2,3}

¹Minolta Camera Co.,Ltd. Advanced Science Center of Seishin

²Graduate School of Information Science, Nara Institute of Science and Technology

³Department of Computer Science and Communication Engineering, Kyushu University

周期性はメディアの連続的な処理を行なうためと、複数の処理間で同期をとるために必要である。しかし現在の OS のプロセス処理において、コンテキストスイッチが発生するのは、lock などにより WAIT 状態に遷移する場合 (図 3 (1)) や、タイムスライシングにより READY 状態にもどる場合 (図 3 (2)) である。このために、コンテキストスイッチが必ずしも一定間隔で起こることは保証できない。また、CPU は時間的制約のない処理とある処理を区別しないために、時間的制約のある処理を周期的に行なうことができない。

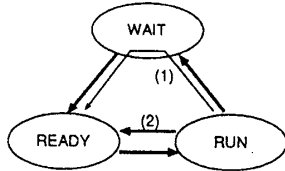


図 3: プロセスの状態遷移図

また、カーネルモードになる要因としては、

- システムコールを使用する
- 割り込みを受ける

場合があげられる。このとき、タイマーがコンテキストスイッチを起こすことができず、周期性の保証の妨げになる。

3.3 CPU サーバによる CPU の管理

周期性を保証するために、タイマーによるコンテキストスイッチと時間的制約のある処理の管理のために差分リストを用いる。タイマーは、定量化の時の単位時間ごとにコンテキストスイッチを強制的に起こす。また、単位時間内に処理が終了した場合にも、コンテキストスイッチを起こすことで、CPU を効率的に使用する。

時間的制約のある処理の管理に差分リスト^[4]を用いることで、単一処理の周期性の保証と複数の処理の同期機構を実現した。しかし、差分リストでは処理にかかる時間を考慮していないため、複数の処理の周期が重なる場合に問題が生じる。そこで複数処理の時間を考慮した差分リストの拡張を行なった。具体的には、縦の処理数分だけ単位時間をそれ以後の横軸に付加している (図 4)。

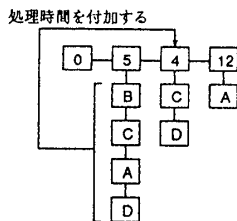


図 4: 差分リストの拡張

また、リザベーションされていない部分には、時間的制約のない処理を割り当てることで、時間的制約のない処理も CPU に割り当てる。これは、時間的制約の

ある処理とない処理を別のキューに配置することで行なう (図 5)。

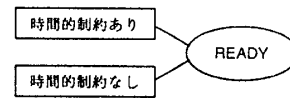


図 5: プロセスの管理

3.4 カーネルモードへの対策

周期性を保証するためには、カーネルモードの状態の時間を短くしなければならない。そのためにデバイスをユーザレベルで使用できるようにするためのユーザレベルデバイスドライバ^[5]の使用を検討している。

4. 考察

QOS を考慮したリザベーションを考える。これは、リザベーションを唯一の値で指定するのではなく、許容範囲の幅を付加した形にすることで、指定値でのリザベーションが不可能な場合、許容範囲を加えて値でリザベーションを行なうことである。この場合、要求の QOS は下がるが、リザベーションは行なわれる。例えば、CPU のリザベーションの指定を周期時間に許容範囲の幅を付加することで、指定の周期時間でのリザベーションができない場合に許容範囲を加え周期時間を大きくすることでリザベーションを行なう。この場合、周期時間が大きくなるために要求の QOS は下がることになる。

5. まとめ

時間的制約を考慮した CPU のリザベーションについて述べた。本方式には複数の周期性のある処理を行なうことが可能である。今後は実装に関する問題点を解決し、CPU だけでなく他のリソースについて研究する予定である。

終りに本稿の執筆に際して御助言を頂いた奈良先端科学技術大学院大学の荒木、福田研究室の研究生に感謝致します。

参考文献

- [1] 岡村, 荒木, “PDE-II の概要 ~QOS に基づいたマルチメディア処理モデル~”, 情報処理学会全国大会, 1H-5, March 1994.
- [2] 稲垣, 岡村, 荒木, “PDE-II におけるメディア間同期機構の実現に対する考察”, 情報処理学会全国大会, 1H-6, March 1994.
- [3] 吉川, 岡村, 荒木, “PDE-II における実時間同期のための周期スレッドの提案”, 情報処理学会全国大会, 1H-8, March 1994.
- [4] Douglas Comer 著, “OPERATING SYSTEM DESIGN THE XINU APPROACH”, Scholars Book Co., Ltd.
- [5] 加藤, 乾, 人見, “マイクロ・カーネル・アーキテクチャにおける I/O driver 実装方法に関する一考察”, 情報処理学会 システムソフトウェア・オペレーティングシステム研究会, OS-58-5, 1993.