

# 可変プロセッサ速度に対応した実時間処理の考察<sup>†</sup>

緒方正暢<sup>1</sup> 西尾信彦<sup>2</sup> 河内谷清久仁<sup>3</sup> 徳田英幸<sup>4</sup>

3H-2

<sup>1</sup>日本アイ・ビー・エム(株)パワー・モバイル・システムズ

<sup>2</sup>慶應義塾大学 環境情報研究所

<sup>3</sup>日本アイ・ビー・エム(株)東京基礎研究所

<sup>4</sup>慶應義塾大学 環境情報学部

## 1 はじめに

バッテリーオペレーションを想定した携帯型計算機が急速に普及している。これらのシステムでは、低電圧動作の素子を採用したり、液晶パネルやハードディスク装置などは使用しない時に電源をオフにすることで消費電力の削減を行なっている。プロセッサは、その動作周波数に比例して消費電力が多くなる特性をもつため、初期のシステムでは性能の低いプロセッサしか採用できなかった。しかし、近年、デスクトップ機と同等の高性能プロセッサを採用し、動作周波数を動的に変更、設定できるようにしているシステムが登場している。つまり、処理能力を必要としない状態ではプロセッサの性能(動作周波数)を下げ消費電力を抑えようとしている。

携帯型計算機上でもシステム性能の向上につれて連続メディア処理[1]が行われるようになってきている。連続メディア処理の実現のためには、各メディアが持っている時間制約を満たすための実時間処理が必須である[1]。しかし、従来の実時間処理では、今日の携帯型計算機で行われているような動的なプロセッサ動作速度制御を考慮していない。プロセッサの動作速度が下がると実時間タスクの処理時間が長くなり、デッドラインを満たせなくなることがある。また、ユーザは、OSやアプリケーションの動作とは独立にプロセッサ速度を下げる操作を行うことができるので、システムが突然、過負荷状態になってしまうこともある。この場合は従来のようなアドミッション制御が行えないし、通常の実時間スケジューリングだけでは対処できない。

我々は、これらの問題を解決するための研究開発を行っている[2],[3],[4]。動的にメディアの品質を制御(動的QOS制御)し、システム性能に適應するためのソフトウェア技術が特に重要である[5],[6]。本稿では、プロセッサの動作速度が動的に変化することを想定した場合の実時間処理手法について検討する。

“Real-time Processing for Power Managed Computers”<sup>†</sup>  
Masanobu Ogata<sup>1</sup>, Nobuhiko Nishio<sup>2</sup>, Kiyokuni Kawachiya<sup>3</sup> and Hideyuki Tokuda<sup>4</sup>

<sup>1</sup>Power Mobile Systems, IBM Japan, Ltd.,  
1623-14, Shimotsuruma, Yamato, Kanagawa 242, Japan  
E-Mail: ogata@trl.ibm.co.jp

<sup>2</sup>Keio Institute of Environmental Information, Keio University,

<sup>3</sup>IBM Research, Tokyo Research Laboratory,

<sup>4</sup>Faculty of Environmental Information, Keio University

<sup>†</sup>この研究は、情報処理振興事業協会(IPA)が実施している独創的情報技術育成事業における「M3K: 超分散並列環境のためのオペレーティングシステムカーネル」プロジェクトのもとに行われた。

## 2 パワーマネージメント

### 2.1 マイクロ・パワーマネージメント

初期のシステムで用いられていたパワーマネージメント(PM)手法で、ハードウェアがシステムの状態を監視して電力を管理する手法である。システムのアイドル状態をいかに正確に検出できるかが重要であったが、その手法は対応するオペレーティングシステムやアプリケーションに依存しており、汎用的な方法はなかった。また、明示的にソフトウェアからの制御が行えないので節電効果は少なかった。

### 2.2 マクロ・パワーマネージメント

現在行われているPM管理手法である。OSがPM管理制御の中心になり、ハードウェアと協調して制御を行う。今日のおよそ90%のノート型計算機にはAdvanced Power Management (APM) BIOS[7]が実装されている。

PM管理はシステム内の資源(例えば、液晶パネル、ハードディスク、プロセッサ)毎に行われる。ただし、各資源のPM管理値を個別に設定してもシステム全体の性能バランスがよくないので、これらの設定を組み合わせた電力モードを用いる。PMモードの個々の値は、OSやアプリケーションプログラムからAPMインタフェースを通じて設定できる。

例えば、IBM ThinkPad<sup>1</sup> 750C (Intel 486SL-33MHz)の場合、以下の電力モードがサポートされている。

- Highモード: プロセッサ速度:高速(33MHz), 液晶バックライト:高輝度, など
- Mediumモード: プロセッサ速度:中速(約20MHz), 液晶バックライト:低輝度, など
- Lowモード: プロセッサ速度:低速(約10MHz), 液晶バックライト:低輝度, など

## 3 実時間処理に関する考察

### 3.1 要求と動的QOS制御

電力モードの設定に関しては以下のような要求がある。

- システム性能を使いきらない場合は、その分電力モードを下げことで消費電力を節約し、バッテリーオペレーションの時間を長くする[8]。アイドル時間がないように動作周波数を制御しタスクスケジューリングができるのが理想的である。
- 低電力モード時に、高いメディア品質が必要な処理を実行する時は高電力モードしてから実行する。

<sup>1</sup>IBM, ThinkPadはIBM Corp.の商標

- バッテリーでの動作時間を伸ばすことが重要な場合は、たとえシステム性能をフルに必要とする処理があっても電力モードを下げる。
- 優先度の高い処理は常に高電力モードで実行される。
- バッテリーの残量によって電力モードを選ぶ。バッテリー残量がある閾値を下回った場合は無条件に低電力モードで実行する。

以上のような要求を満たすためには、連続メディアの品質を動的に変化させることで対応できる。ところが、プロセッサの動作速度は電力モードによって大きく変化する。例えば、高電力モードに比べて低電力モードは約1/3の動作周波数になる。処理性能比は約3である。従って、実時間スレッドの実時間属性をフィードバック情報をもとに操作する Self-Stabilization 方式 [9] では対応が困難である。また、QOS 制御サーバ [10] 方式を用いて実行中の実時間スレッドの時間属性を制御する方法でも3倍の性能比を汎用的に制御するのは困難と思われる。

#### 4 プロセッサキャパシティ

図1に本稿で提案するプロセッサキャパシティの概念を示す。プロセッサキャパシティによるプロセッサ資源の制御方式として既に [11] が提案している。この方式は「Tミリ秒ごとにCミリ秒を予約する」といったキャパシティの概念である。本方式は動作周波数も考慮しようとしているところが異なる。横軸は時間、縦軸は単位時間あたりの処理能力（動作周波数に比例する）を表す。電力モード Low の時はA、Medium の時はA+B、High の時はA+B+Cのキャパシティがあると考えられる。単位時間あたりに実行できる命令数は動作周波数に比例するのでこのようにプロセッサのキャパシティを2次元的にとらえるのは自然であると考えられる。

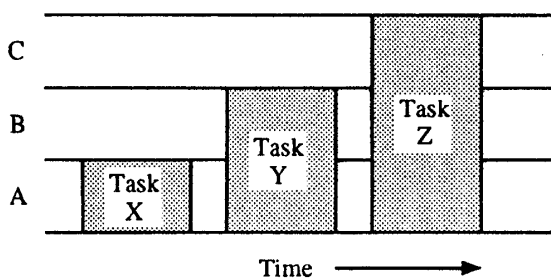


図1: プロセッサキャパシティの概念

#### 5 今後の課題

今後はプロセッサキャパシティを利用した制御方式の検討、設計を行ない、Real-Time Mach 3.0 [12] をもとにプロトタイプシステム (図2) を作成したい。

パワーマネジメント機能が実現されているシステムでは、必要ない資源の電源を切ったり、スタンバイモードにすることで消費電力を下げるができる。しかし、一旦、スタンバイ状態になった資源はすぐには利用できないといった欠点を持っている。実時間処理においてはこのような待ち時間は望ましくないが、常に資源を利用可能な状態にしておくのでは節電効果がない。必要な資源をできるだけ必要な時だけ利用可能にするための研究が重要である。

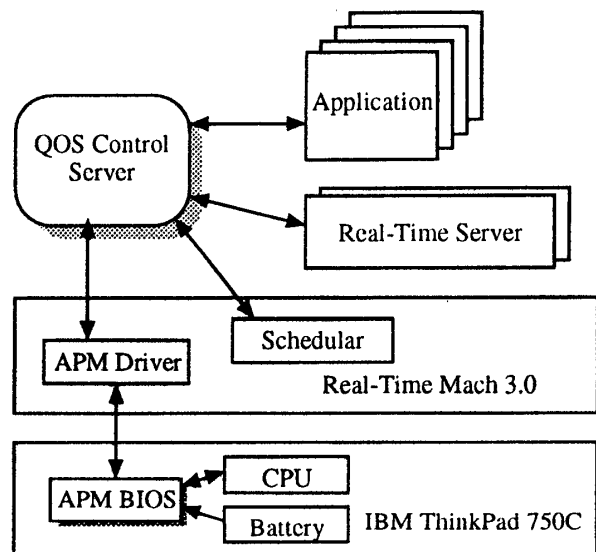


図2: プロトタイプシステムの構成図

#### 参考文献

- [1] 西尾, 他: “マイクロカーネルによる連続メディア処理の基盤技術,” 情報処理学会「第5回コンピュータシステム・シンポジウム」, pp.17-24 (1993).
- [2] 緒方, 他: “モバイルコンピュータを使った分散マルチメディア環境の実現,” 第49回情処全大, 6S-9 (1994).
- [3] 追川, 他: “M3K: 拡張可能なマイクロカーネル,” 第50回情処全大, 3H-1 (1995).
- [4] 西尾, 他: “モバイルコンピューティングのためのマイクロカーネルの改良,” 第50回情処全大, 3H-3 (1995).
- [5] 緒方, 他: “動画のフレーム間相関を利用した動的なQOS制御の実験,” 第47回情処全大, 4V-4 (1993).
- [6] 河内谷, 他: “連続メディアのQOS制御のためのOSサポート,” 第6回コンピュータシステム・シンポジウム, pp.119-126 (1994).
- [7] “Advanced Power Management (APM) BIOS Interface Specification, Revision 1.1,” Intel Corporation and Microsoft Corporation (1993).
- [8] M. Weiser, et al.: “Scheduling for Reduced CPU Energy,” USENIX 1st Symp. on Operating Systems Design and Implementation (OSDI), pp.13-23 (1994).
- [9] H. Tokuda, et al.: “A Real-Time Thread Model for Continuous Media Applications,” Technical Report, CMU (1993).
- [10] 河内谷, 他: “Real-Time Mach 上でのQOS制御サーバの実験,” 第47回情処全大, 4V-3 (1993).
- [11] C. W. Mercer, et al.: “Processor Capacity Reserves: Operating System Support for Multimedia Application,” Proc. Intl. Conf. on Multimedia Computing and Systems, pp.90-99 (1994).
- [12] H. Tokuda, et al.: “Real-Time Mach: Towards a Predictable Real-Time System,” USENIX Mach Workshop, pp.73-82 (1990).