消費電力制御のためのジョブタイプの判別方法

1S－4

相原 達　古市 実裕
日本アイ・ビー・エム (株)
東京基礎研究所

1 はじめに

パソコンの省電力機能は，ユーザが事前に設定したパラメータに基づいて制御されることが多い．

ユーザはバッテリ駆動時間を延ばすために，CPU速度，LCD輝度，LCDオフタイマ，HDDのモータ停止タイマ，内蔵モデムや通信ポートやPCカードスロットなどの電源，などの設定を変更する．長いバッテリ駆動時間と，高いパフォーマンス・高機能を同時に実現することは困難である．さらに，これらの設定値の変更には，特別な操作が必要であり，変更しないユーザも少なくない．

そこで，学習機能付きの省電力機能も開発されているが，過去の使用実績に基づくため，制御が後手に回ったり，ユーザの意思に反したりしがちである．

本論文では，CPUやキーボードなどの複数のデバイスの使用状況を監視し，パソコンの動作状況とユーザの作業内容（ジョブタイプ）を判別することで，タイプに応じた省電力設定を動的かつ自動的に選択する手法を提案する．この方法は，バッテリ駆動時間を延ばすことだけでなく，システムのパフォーマンスを向上させることも期待できる．

2 ジョブタイプによる省電力制御

典型的な作業内容をジョブタイプとして分類し，それぞれのジョブタイプごとに最適な省電力制御をすることで，パフォーマンスを犠牲にせずに，効率的に省電力を実現する．

ジョブタイプを判別するには，それぞれのタイプに特徴的なデバイスの稼働状況を利用する．ジョブタイプ識別の監視対象となるデバイスは，CPU，キーボード・マウス，ハードディスク，CD-ROMドライブ，通信ポートなどである．

ここでは，ユーザ操作とCPU負荷を基準に５つのジョブタイプに分類してみた．なお，これらの作業は，通常独立に行なわれ，特にモバイル環境では，組み合わせてバックグラウンド処理することは少ないと考えられる．

(1) 高負荷インタラクティブ (Heavy)…ユーザ操作が多く，CPU負荷も高い．図形の編集や文字フォントの変更など，プレゼンテーション作成ツールやワードプロセッサを使ってグラフィカルな編集をしている場合が該当する．

(2) 軽負荷インタラクティブ (Light)…ユーザ操作が多いものの，CPU負荷は低い．表計算ソフトでのデータ入力やエディタでの文章編集が該当する．

(3) バッチ…ユーザ操作がなく，CPU負荷は高い．データ圧縮やコンパイラが該当する．高速処理が望ましい．

(4) アイドル…有意味な処理をしていない場合．

(5) 中間…上記各ジョブの境界に位置する場合．

次に，それぞれのジョブタイプの性質を考慮した省電力方針を検討する．

LCD輝度は，ユーザが画面を見ている場合にのみオンにし，見ていない時にはオフにするように制御する．ユーザが画面に注視して操作している高負荷および軽負荷インタラクティブについては輝度を確保する．逆にユーザが画面に興味を持たないと考えられるバッチおよびアイドルについては輝度を抑制またはオフにする．

CPUに関しては，単位時間あたりにすべき処理が多い場合には，CPU速度を上げて処理にかかる時間を短縮することが，システム全体としての省電力につながる．逆に，単位時間あたりの処理が少ない場合には，CPU速度を下げることで，CPUの消費電力そのものを削減する．バッチでは，ユーザの操作を待たずに処理が進められるので，CPU速度を最大にする．高負荷インタラクティブでは，CPU速度を高く保つことが重要ではあるが，ユーザビリティを

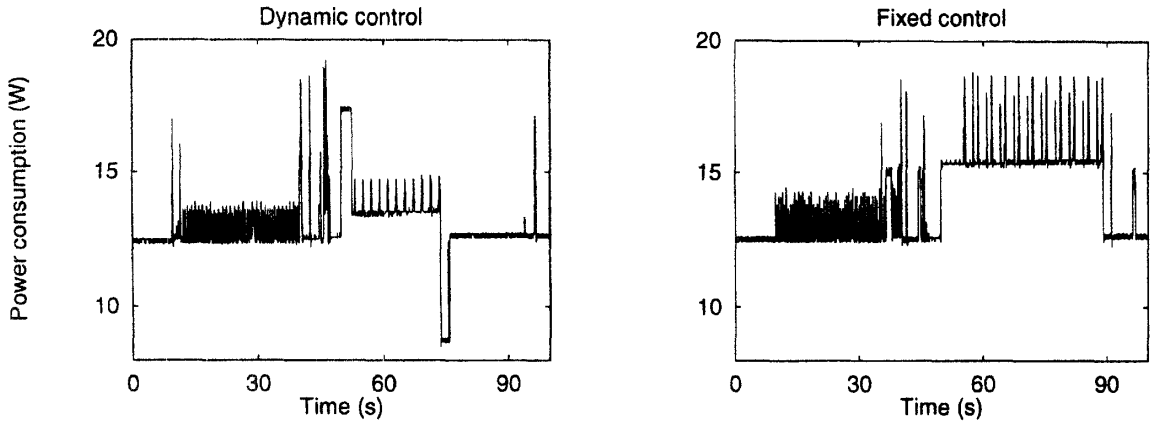

図 1: 消費電力の比較．動的設定と固定的設定では，省電力効果に大きな違いが出る．

損なわない程度にCPU 速度を抑制する．低負荷インタラクティブでは，CPU の負荷は極めて軽いので，CPU 速度を大きく下げる．アイドルでは，行なう処理がないので，CPU を停止する．

3 省電力制御の例

3.1 ジョブタイプの分類

ジョブタイプの判別には，CPU の平均負荷 $I_{D_1}(t)$ とキー入力頻度 $I_{D_2}(t)$ とを用いる．

	CPU 負荷	キー入力頻度
Heavy	$I_{D_1} \geq 50$ (%)	$I_{D_2} \geq 50$ (回/min)
Light	$I_{D_1} < 50$ (%)	$I_{D_2} \geq 50$ (回/min)
バッチ	$I_{D_1} \geq 75$ (%)	$I_{D_2} = 0$ (回/min)
アイドル	$I_{D_1} < 10$ (%)	$I_{D_2} = 0$ (回/min)
中間	上記以外	

ジョブタイプの分析に基づいて，制御対象のデバイス CPU，LCD，HDD を以下のように制御する．

	CPU 速度	LCD 輝度	HDD タイマ
Heavy	50%	75%	1 分
Light	12.5%	75%	0(即時停止)
バッチ	100%	0	0(即時停止)
アイドル	0	0	0(即時停止)
中間	50%	75%	1 分

3.2 省電力効果

ここでは，表計算ソフトでの作業を想定する．まず，スプレッドシートに大量を数値データを入力し，一定の入力作業終了後，データに対して長大な計算プログラムを実行する．図 1のように，本論文の動的省電力設定は通常の固定的省電力設定（ここでは，クロック周波数が 50%，LCD 輝度が 75%）に比べて電力消費が少なくなる．

通常のデータ入力時，動的設定では軽負荷インタラクティブと識別され，CPU 速度が低下する．一方，固定的設定では必要以上の CPU 速度となる．動的設定では CPU の消費電力分の差だけ省電力となる．

計算プログラムの実行が始まると，動的設定ではバッチと識別され，CPU 速度は最高速に，LCD はオフになる．最高のパフォーマンスが得られ，また，LCD を節電することで，電力消費のピーク値を抑制する．一方，固定的設定では，CPU 速度が遅いので，計算時間が長くなる．CPU の消費電力は低いが，LCD が点灯したままなので，全体としての電力消費は同程度である．動的設定では計算が早く終了する分だけ省電力となる．

4 まとめ

システムのパフォーマンスを確保しつつ，バッテリ駆動時間を伸ばすためには，ジョブタイプによる省電力制御が有効であることを，CPU の負荷とユーザの操作を監視することで，CPU，HDD，LCD の消費電力を制御する方法で示した．

無難な判別のために中間タイプを用いたが，さらにジョブタイプの判別精度を向上させる必要がある．そのために，監視するデバイスの数を増やすと制御が複雑化する．監視の負荷が大きくならないように配慮する必要がある．また，LCD をオフにするなど大胆な制御を行なう場合は，本当にユーザビリティに影響はないか，実際にユーザに受け入れられるか，官能評価する必要性がある．