

# 並列演算環境における消費電力量低減化手法

## CPU Power Control Technique for energy efficiency on Parallel Computing Environment

尾崎敦夫†  
Atsuo Ozaki

佐藤裕幸†  
Hiroyuki Sato

### 1. まえがき

更なる性能向上のためのプロセッサ開発は止むことなく、高性能な製品が世の中に登場し続けている。一方、高性能な演算環境を必要とするアプリケーションも次々に開発される状態であり、プロセッサの高性能化に対する要望は留まることを知らない。一般的に、1つのプロセッサで扱えないような演算負荷の高い処理で、複数に分割可能であるものは、複数のプロセッサを備えた並列演算環境で実行するのが適している。しかし、プロセッサ数が増えると消費電力、発熱、そして設置スペースの問題などが新たに出てくる。特に、長時間連続運用するシステムでは、消費電力および熱の問題が深刻となる。

他方、モバイルコンピューティングの分野では、長時間バッテリー駆動によりシステムを利用できる技術が必須となる。近年、消費電力量削減のために必要に応じて動作周波数および電圧を切り替え可能なプロセッサ[1]が登場し、高性能な製品群にまで展開され始めており、マルチプロセッサ化も計画されている。この省電力技術は熱問題解消にも繋がる。我々は、このような省電力技術を導入したプロセッサを備える並列演算環境での消費電力量低減化手法に関する検討を行っている。

本稿では、並列演算環境上での消費電力量低減を目的に検討した実行方式選定手法について説明する。また、動作周波数および電圧が可変であるプロセッサの具体例を挙げ、本プロセッサによる並列演算環境を想定し、本環境で提案手法に基づきタスクを処理する場合の効果を示す。

### 2. 消費電力量低減化のための実行方式

#### 2.1 基本概念

現在主流の CMOS 回路では、基本的に動作周波数を上げた場合にはプロセッサを安定動作させるために電源電圧も上げる必要がある。逆に、動作周波数を下げた時には電圧も下げることが可能な場合がある。一般的に、消費電力  $P[W]$  と動作周波数  $F$  および電源電圧  $V$  との関係は、リーク電力を無視した場合、式(1)を満足する[2]。ここで  $t$  は信号遷移率、 $C$  は静電容量を示す。

$$P = t \cdot C \cdot F \cdot V^2 \quad (1)$$

消費電力  $P$  は式(1)に示すように動作周波数  $F$  に比例し、かつ電圧  $V$  の二乗に比例するという両要素が支配的である。すなわち、消費電力  $P$  は動作周波数  $F$  の変化率に対して二乗のオーダーで推移する。我々は、この特性から動作周波数が可変であるプロセッサを備える並列演算環境においてタスクを実行する場合に、例えば、少ない数のプロセッサで動作周波数を上げて実行する場合と、動作周波数を下げた多くのプロセッサで実行する場合とで消費電力量  $[W \cdot s]$  の大小関係が、プロセッサの特性やタスクの性質に応じて異なる場合があることに着目した。そして、プロセッサの特性やタスクの性質に基づいて、最も消費電力量が少なくなる実行方式(使用するプロセッサ数および該プロセッサの動作周波数)を選定するための指標が示せるものと考えた。ただし、処理タスクの性質(プロセッサ数に応じたタ

スク処理時間と通信処理時間)が事前に分かっていることが前提となる。

#### 2.2 実行方式選定手法

我々が提案する実行方式選定手法は、タスクが与えられる度に、そのタスクの処理制限時間内に最も少ない消費電力量で処理することができるプロセッサ数とそれらのプロセッサの動作周波数および電圧を選定するものである。なお、使用しないプロセッサおよび処理していないプロセッサはその際、最も電力を消費しない動作周波数および電圧に設定される。基本点に制限時間内に処理が完了するのであれば、1つのプロセッサで低い動作周波数および電圧により実行するのが最も少ない消費電力量で実行できる。他方、処理時間に余裕がない場合はシステムの全プロセッサを、設定できる最も高い動作周波数で実行させる必要があり、この実行方式が最も消費電力量を要する。これらはタスクの負荷状態および処理制限時間が両極端な場合であるため、比較的容易に上記実行方式を選定することができるが、実行方式の選定が困難になるのは、これらの中間状態にある場合である。前述したように、本手法は対象とする処理の性質が事前に分かっていることを前提としている。従って、その処理を何個のプロセッサでどの動作周波数を用いればどれだけの実行時間で処理が完了するかを見積もることができるため、処理制限時間が分かれば動作周波数毎の必要プロセッサ数が限定できる。また、同じ動作周波数であれば使用するプロセッサが少ない方が省電力量で実行できるため、動作周波数毎の最適な実行方式間で消費電力量を比較すれば最も消費電力量が少ない実行方式が選出できることになる。

### 3. 提案手法の具体例

本章では、対象とする問題と並列演算環境のプロセッサの特性および実行方式の種類を限定した場合の提案手法に基づく実行方式の選定基準について検討する。具体的には、1つのタスク(処理データ)が与えられた場合に、1つのプロセッサで処理する場合と、複数のプロセッサにより処理する場合との省電力実行のための選定基準を求める。まず、並列演算装置の各プロセッサは表1に示す特性を備えているものとする。

表1. プロセッサの特性

動作周波数	消費電力	状態
	P	高速動作状態
	P	標準動作状態
	P	遊休状態

また、本並列演算環境は  $N$  個の上記プロセッサを備えているものとし、該処理データが与えられたとき、以下の3つの実行方式から、最も少ない消費電力量で実行できる方式を選定するものとする。

**実行方式1:** 動作周波数を標準状態のとして1つのプロセッサにより実行する。他のプロセッサは遊休状態に設定する。

**実行方式2:** 動作周波数を高速状態のとして1つのプロセッサにより実行する。他のプロセッサは遊休状態に設定する。

**実行方式3:** 動作周波数をとして  $n$  個 ( $2 \leq n \leq N$ ) のプロセッサにより実行する。他のプロセッサは遊休状態に設定する。

†三菱電機株式会社情報技術総合研究所

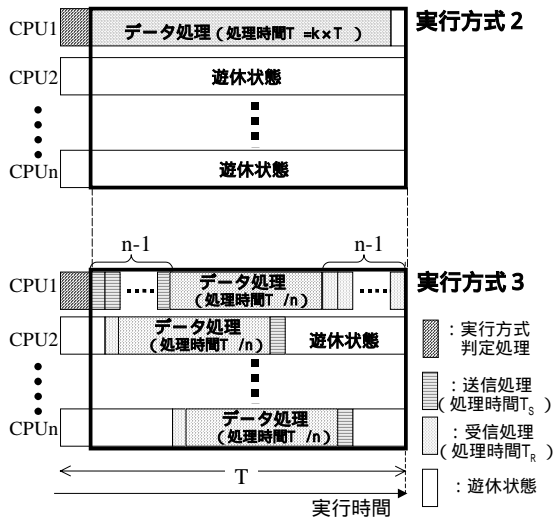


図1. 実行方式2と3のタイムチャート例

なお、処理データを割り付けられたプロセッサにおいても、実行していない状態の時は、遊休状態の動作周波数に設定されるものとする。まずは、1つのプロセッサで動作周波数を上げて処理する場合と、動作周波数は標準状態にして複数のプロセッサで処理する場合の消費電力量の比較を行うため、実行方式2と3の比較を行う。図1は、処理制限時間(T)内の実行方式2と3のタイムチャート例を示したものである。両者の違いは太線枠内部分であるため、この部分についての消費電力量を比較すれば良い。図1の場合では、処理制限時間(T)は、実行方式2より実行方式3の処理時間の方が大きいため、式(2)のように示すことができる。ここで、 $T_c(=T_s+T_r)$ は1回の通信処理に要する時間であり、送信処理時間  $T_s$  と受信処理時間  $T_r$  を加えたものである。また、T は、1つの処理データを1つのプロセッサで動作周波数により処理した場合の実行時間である。また、nはプロセッサ数を示す。

$$T = (n-1) \cdot T_c + T/n \quad (2)$$

この場合の実行方式2による消費電力量  $C_2[W \cdot s]$  を示したものが式(3)である。ここで、式(3)の第1項は動作周波数でデータ処理を行うのに要する消費電力量であり、残りの第2項は、遊休状態であるプロセッサ(図1: CPU2~CPU<sub>n</sub>)とデータ処理が終わり遊休状態となった期間のプロセッサ(図1: CPU1)の消費電力量を示したものである。また、 $k = \frac{T_c}{T}$  である。

$$C_2 = P \cdot k \cdot T + P \cdot [(1-k) \cdot T + n \cdot (n-1) \cdot T_c] \quad (3)$$

同様に、この場合の実行方式3による消費電力量  $C_3[W \cdot s]$  を示したものが式(4)である。ここで、式(4)の第1項は通信処理に要する消費電力量と全部の遊休状態の消費電力量を足したものであり、第2項はデータ処理に要する消費電力量を示したものである。

$$C_3 = (n-1) \cdot [2 \cdot P \cdot T_c + (n-2) \cdot P \cdot T] + P \cdot T \quad (4)$$

式(3)と(4)より、 $C_2 = C_3$  とすると式(5)を導出することができる。ここで、 $\alpha$  はデータ処理に対する通信処理の処理時間の比率( $T_c/T$ )を表す。

$$\alpha = \{k \cdot P - P + P \cdot (1-k)\} / \{2 \cdot (n-1) \cdot (P - P)\} \quad (5)$$

この式(5)に基づいて求められた  $\alpha$  と、実行方式3により選定された省電力実行のための  $\alpha_3$  とを比較すれば実行方式

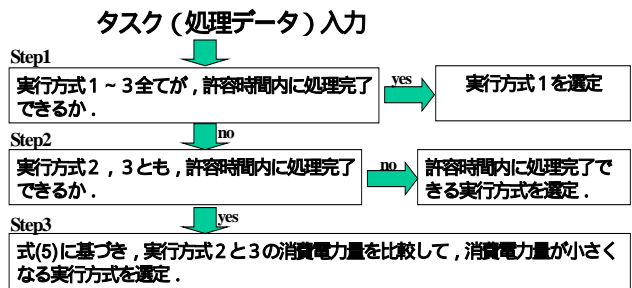


図2. 提案手法の具体例におけるフローチャート

2と3の優劣が判定でき、 $\alpha < \alpha_3$  であれば実行方式2を、 $\alpha > \alpha_3$  であれば実行方式3を適用すれば良いことになる。なお、ここまでの議論は、図1に基づいて実行方式2よりも実行方式3の処理時間の方が大きい場合に関するものであるが、逆の場合でも、式(3)と(4)は異なるものになるが、同じ式(5)が導出される。但し、 $n=2,3$  の場合は、送信処理時間  $T_s$  と受信処理時間  $T_r$  の大小関係で、例えば、図1で示した実行方式3のCPU1にも遊休状態が発生してしまう場合がある。しかし、 $T_s=T_r$  と仮定すると、 $n=2,3$  の場合でも  $\alpha$  は式(5)となる。

次に、実行方式1と2の比較を行う。まず、実行方式1と2とも制限時間内に処理が完了する場合を考える。この場合、実行方式1と2の消費電力量の差は、式(6)となる。ここで、 $C_1$  は実行方式1での消費電力量を示す。

$$C_2 - C_1 = T \cdot (P \cdot k - P) + P \cdot T \cdot (1-k) \quad (6)$$

基本的に、動作周波数の向上率以上に消費電力は向上するため式(1)参照、式(6)の第1項の()内は正となり、また第2項も正となるため、実行方式1と2とも制限時間に納まるのであれば、実行方式1が選定されることになる。

更に、実行方式1と3の比較では動作周波数が同じであるため、実行方式3の処理時間よりも実行方式1の処理時間の方が小さくなる(並列処理の効果が無い)場合は、消費電力量の観点からも実行方式3を選定するメリットは何も無い。両方式とも、制限時間内に処理を完了できるのであれば、実行方式1が選定される。なお、並列処理の効果がある場合は、式(7)を満足する場合である。

$$\left. \begin{aligned} < 1/n \quad (n=4 \text{ の場合}) \\ < 2 \cdot (n-1) / \{n \cdot (n+2)\} \quad (n=2,3 \text{ の場合: } T_s=T_r \text{ と仮定}) \end{aligned} \right\} (7)$$

以上の比較結果から、本章で取り上げた例の場合は、図2のフローチャートに基づいて実行方式を選定すれば良いことになる。ここで、「許容時間」とは、マージンを持たせるために上記制限時間よりも短めに設定された時間のことである。

#### 4. 実プロセッサによる並列演算環境への適用検討

本章では、動作周波数および電圧が可変であるプロセッサの代表例として、PowerPC 405LP を例に上げ、本プロセッサを複数備える並列演算環境を想定して、前章で述べた実行方式選定基準を適用した場合の検討を行う。

表2はPowerPC 405LPの動作周波数Fと消費電力Pの関係を示したものである。本プロセッサは、動作周波数が152MHz 切りまでは最低の電圧1.0Vで動作可能である。従ってこの範囲までの消費電力の増減は一次関数( $P = a \cdot F$ ; aは定数)となる。なお、PowerPC 405LPの遊休状態の消費電力( $P_{idle}$ )は、表2の各値から算出したものである。

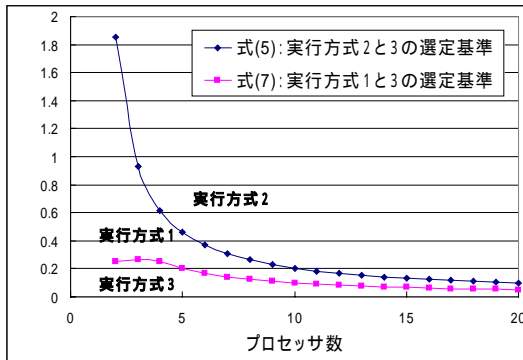


図3．実行方式選定基準

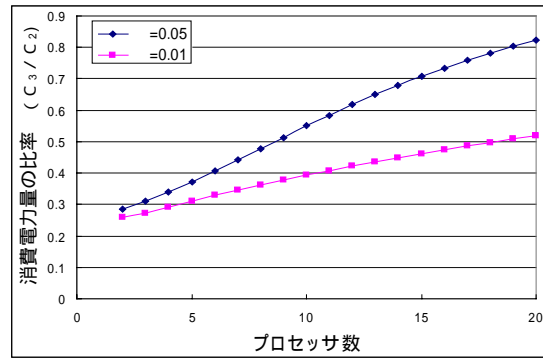
図4．実行方式2と3の消費電力量の比率  
(実行方式3が勝る場合)

表2．PowerPC 405LP の特性

動作周波数(MHz)	電圧(V)	消費電力(W)	状態
380	1.8	0.5	高速動作状態
152	1.0	0.053	標準動作状態
33	1.0	0.0115*	遊休状態

次に、この PowerPC 405LP を複数備える並列演算環境を想定して、本環境上で3章で示した提案手法の具体例を実現させた場合の検討を行う。図3は、3章で示した各パラメータに、表2の各値を与えた場合のプロセッサ数 ( $n > 2$ ) に対する、式(5)と式(7)の値を示したものである。なお、ここでは、 $f = 152$ 、 $f_0 = 380$ 、 $P = 0.053$ 、 $P_0 = 0.5$ 、 $P_1 = 0.0115$ 、そして  $k = 0.4(152/380)$  とした。そして、実行方式3で対象とする処理データを処理する場合の、省電力実行のための最適なプロセッサ数とその場合の  $C_3$  の値が求まれば、図3の式(5)の傾向から、実行方式2で実行すべきか実行方式3で実行すべきかが判定できる。ただし、実行方式3が選定されたとしても、 $C_3$  の値が図3の式(7)の値以上である場合は、前章で述べたように実行方式1を選定した方が良い場合となる。

また、図4は実行方式3が選定および実行された場合のプロセッサ数に応じた実行方式2に対する消費電力量の比率 ( $C_3/C_2$ ) を、適当な ( $P = 0.05$ ) に関して示したものである。この比率が1以下であれば実行方式3で実行した方がその値分、実行方式2より少ない消費電力量で実行できることを示す。なお、 $P = 0.05$  であればプロセッサ数が2~20の範囲内では常に並列処理による効果が得られる。この結果(図4)より、 $P$  の値が一定の場合、プロセッサ数が多いほどこの比率が1に近づいてしまうが、逆に  $P$  が小さくなればこの比率は小さくなることが確認できる。処理データ量が一定であればプロセッサが増えるにつれ、任意のプロセッサ間で送受信されるデータ量は小さくなる。従って、プロセッサ数が増えるにつれ  $T_C$  が小さくなり、これによって  $C_3$  が小さくなるとすると、その状態の間は、プロセッサ数増加に対するこの比率の上げ率はより小さくなることになる。また、図4より、実行方式3ではプロセッサ数が少ないほど少ない消費電力量で実行できることが確認できる。

なお、実行方式3での最低限必要となるプロセッサ数は、与えられた制限時間から導出することができる。しかし、処理データが順次入力されるような状況では、その時点で利用できるプロセッサ数に限りがある場合もあり、実行方式3が省電力実行方式として選定されたとしても、必要なプロセッサ数を確保できない場合もあり得る。このような場合は実行方式2が選定されるかもしれないが、それでも制限時間を満足できない時は、動作周波数を  $f$  とした複数プロセッサにより並列処理する実行方式が選定されることになる。本実行方式は、実行方式3よりも少ないプロセッサで制限時間を満足できるものであり、実際のシステムで

は、少なくとも本実行方式で必要となるプロセッサ数は確保できるよう設計されている筈である。しかし、プロセッサの故障や異常などにより、必要プロセッサを確保できない場合は、処理データ量を間引く縮退処理などの適用が必要となる。また、このような場合、フォルトトレランスの仕組みとして、残りのプロセッサで動作周波数を上げて、システム全体としてスループットを維持させることなども考えられる。

## 5. むすび

本稿では、近年盛んに研究開発されている省電力タイプのプロセッサの動作周波数と消費電力の関係に着目し、該プロセッサを備える並列演算環境において、処理の性質に応じて少ない消費電力量で実行するための実行方式選定手法を提案した。そして、その実行方式の具体例を挙げ、その選定基準を示した。また、動作周波数可変型プロセッサの代表例として、PowerPC 405LP を取り上げ、本プロセッサによる並列演算環境を想定して、本環境に提案手法の該具体例を適用した場合の効果を示した。

今後の課題は、問題を限定しない場合のシステム全体としての解析を行い、本提案の実行方式選定基準を体系化することである。また、精度向上を目的にプロセッサ以外の部位に関する消費電力をも考慮したモデルの詳細化を検討する必要がある。但し、クラスタ環境における研究例では、ネットワークカードの消費電力はプロセッサの消費電力に比べて小さく、クラスタ環境全体としては無視できるレベルであったとの報告[3]もある。モデルを細かくし過ぎると逆に精度が落ちる場合もあるため、この辺を見極めてモデルの精度を検討していく予定である。

なお、本方式はまだ多くの課題があるが、マルチプロセッサ環境を備えるシステムであれば基幹系からモバイル系に至るまで、省電力実行のための有効な方式となる得る。また、本環境を備えるシステムでは、省電力という観点だけでなく、動作周波数が上げられる状態のプロセッサを利用して、フォルトトレランスを実現することも考えられ、今後は様々な角度からの検討を行っていく予定である。

## 参考文献

- [1] M.R.Stan, et al., "Power-Aware Computing," Computer, IEEE COMPUTER SOCIETY, vol.36, no.12, pp.35-38, 2003.
- [2] 十山, 他, "CPU 消費電力削減のための周波数-電力協調型電力制御方式の設計ルールとフィードバック予測方式による適用," 電子情報通信学会論文誌, vol.J87-D-I, no.4, pp.452-461, 2004.
- [3] 堀田, 他, "低消費電力プロセッサによるクラスタの検討," 情報処理学会研究報告, 2003-ARC-154(16), pp.91-96, 2003.