

並列演算環境における消費電力量低減化方式の検討

尾崎敦夫[†]

佐藤裕幸[†]

大規模な演算を行うための並列演算環境では、消費電力、発熱、そして設置スペースなどの問題があるが、特に長時間連続運用するシステムにおいては、消費電力および発熱の問題が深刻となる。本稿では、これらの問題を解決することを目的に、近年盛んに研究開発されている動作周波数および電圧が可変である省電力型のプロセッサを利用して、並列演算環境での消費電力量低減化のための実行方式選定手法を提案する。そして、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、本環境でタスクを処理する場合に、提案手法を適用した場合の効果を示す。

A Study of CPU Power Control and Scheduling Technique on Parallel Computing Environment

ATSUO OZAKI[†] and HIROYUKI SATO[†]

Large-scale parallel and distributed computing environment has several problems, e.g. a power consumption, a thermal ascent, a setting space, and so on. Especially the power consumption and thermal ascent problems are serious problem when the system is used for a long time. In this paper, we propose CPU power control and scheduling technique (PCST) on parallel and distributed computing environment, in which the processors can be changed these frequencies and voltages, to solve these problems. By introducing a practical processor, which can be changed its frequency and voltage, we estimate the effectiveness of PCST on parallel computing environment based on this processor.

1 はじめに

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、一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えるプロセッサ[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。

我々が対象としているアプリケーションの特徴は、システムに周期時間毎に入力されるデータをこの周期時間内に処理していくというものである。この周期時間とデータ数およびデータ量はその都度変化する。この処理を行うシステム（並列演算環境）では、該周期時間に最も多くのデータ数およびデータ量が入力される事態を考慮して、プロセッサ数を設定している。このような事態は予告無しに生じるもので、瞬時に対応することが求められているため、プロセッサの状態を、電力はほとんど消費しないが、起動に時間を要するような休止状態にしてしまうのは危険である。このため、通常はほとんどのプロセッサが遊休状態であり、無駄な多くの電力量を消費しているのが実情である。

本稿では、上記のような状況およびシステムを対象に、動作周波数および電圧可変型のプロセッサを利用した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。そして本手法の具体例を挙げ、この中から最適な実行方式を選定するための基準を示す。また、動作周波数および電圧が可変である実プロセッサの例を挙げ、本プロセッサによる並列演算環境を想定して、本環境で提案する上記実行方式選定手法の具体例に基づいてタスクを実行する場合の、実行方式選定基準およびその場合の効果について説明する。

[†]三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

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。このため、遊休状態のプロセッサは、動作周波数および電圧を、設定できる最も小さい値にすることで大幅な消費電力量の削減が実現できる。更に、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任意のアプリケーションを実行する場合に、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、動作周波数を下げて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処理データ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処理データの性質に基づいて、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、処理データの性質(プロセッサ数に応じた処理時間と並列処理を行った場合の通信処理時間)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、システム(並列演算環境)に入力されるデータを与えられた制限時間内に順次処理していくタイプのアプリケーションを対象としている。本手法は、処理データが与えられる度に、与えられた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。図1はプロセッサ数と動作周波数および電圧に対する消費電力と実行・制限時間との関係を概念的に示したものである。基本的に制限時間内に処理が完了するのであれば、一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(図1:状態C)。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、

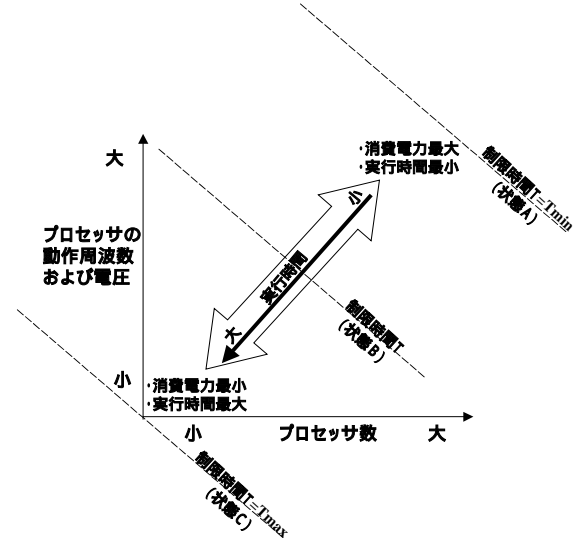


図1. プロセッサ数とその動作周波数・電圧に対する消費電力と実行・制限時間の関係

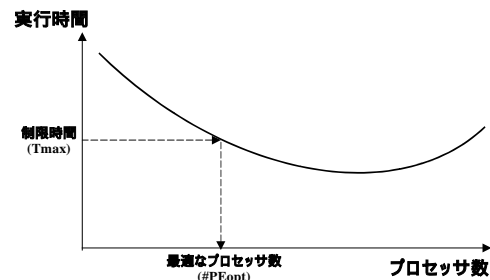


図2. 同一動作周波数の場合のプロセッサ数と実行時間の関係の例

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(図1:状態A)。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(図1:状態B)。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(図2参照)、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。なお、図3は、本手法のフローチャート例を示したものである。ここで、「許容時間」とは、マージンを持たせるため

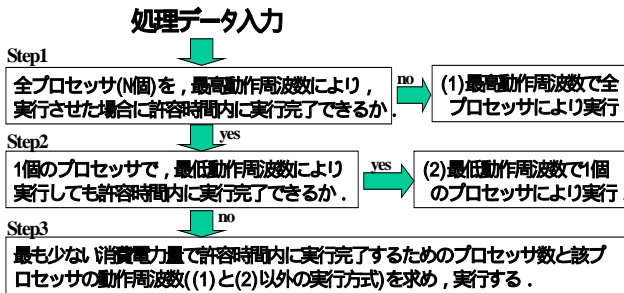


図3. 提案手法のフローチャート例

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。

3 問題を限定した場合の提案手法の例

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。

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、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 ≤ n ≤ 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。

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4は、処理制限

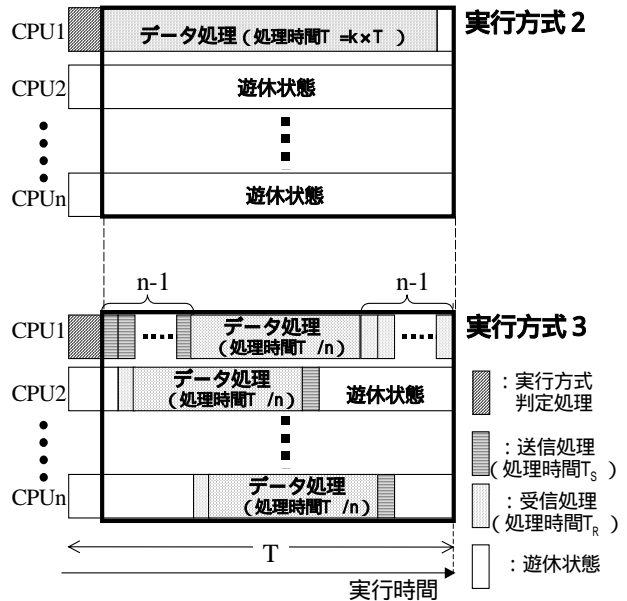


図4. 実行方式2と3のタイムチャート例

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、この部分に関する消費電力量を比較すれば良い。図4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。

$$T = (n-1) \cdot T_c + T / n \tag{2}$$

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4:CPU2~CPU_n)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4:CPU1)の消費電力量を示したものである。また、 $k = \dots$ である。

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

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。

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

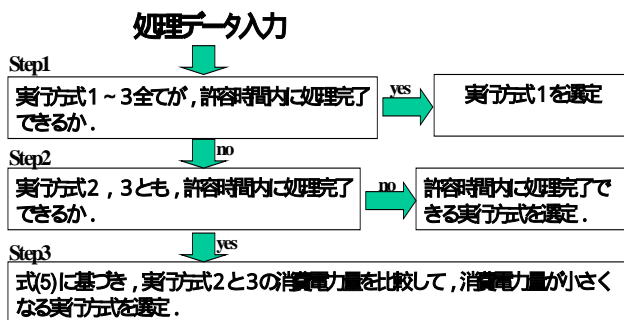


図5 . 本例の場合のフローチャート

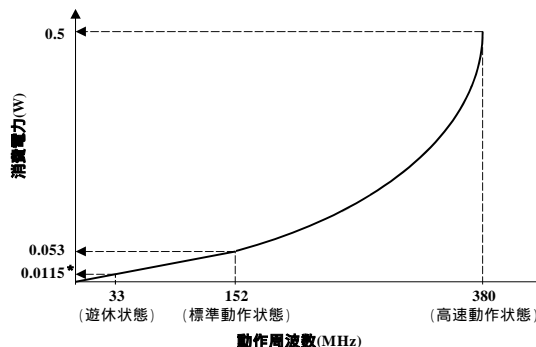


図6 . PowerPC405LP の特性

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、 T_c 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。

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

この式(5)に基づいて求められた T_c と、実行方式3により選定された省電力実行のための T_c とを比較すれば実行方式2と3の優劣が判定でき、 $T_c < T_{c3}$ であれば実行方式2を、 $T_c > T_{c3}$ 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4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4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 T_c 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() T_c 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。

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

以上の比較結果から、本章で取り上げた例の場合は、図5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。

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。

図6および表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 辺りまでは最低の電圧1.0V で動作可能である。従ってこの範囲の消費電力の増減は図6に示すように一次関数($P = a \cdot F$: a は定数)となる。他方、動作周波数が152MHz 辺りを超えると消費電力の増減は式(1)に示す二次曲線となる。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。

表2 . PowerPC 405LP のスペック

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7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 T_c の値を示したものである。なお、ここでは、 $T_c = 152$, $T_c = 380$, $P = 0.053$, $P = 0.5$, $P = 0.0115$,

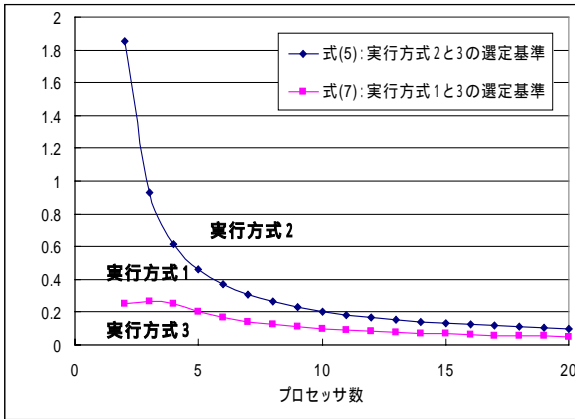


図7．実行方式選定基準

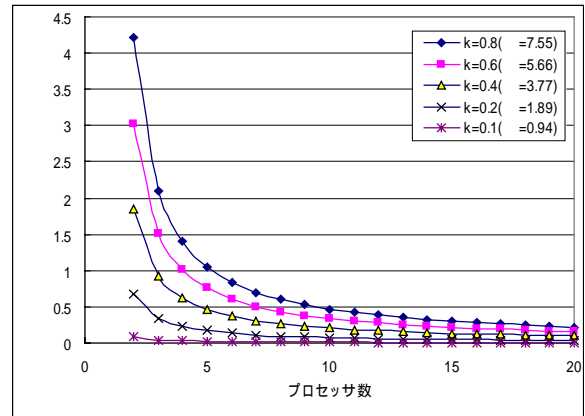


図9．kが に与える影響度合い

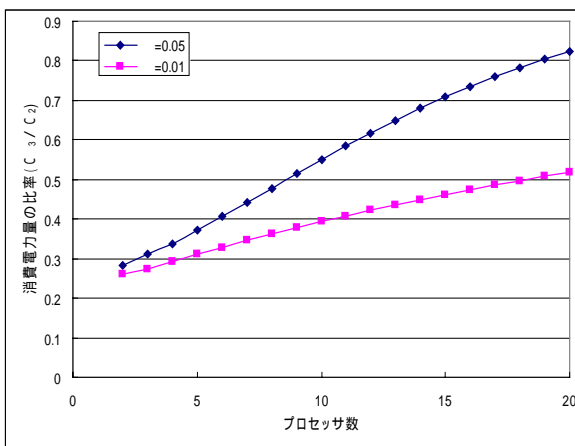


図8．実行方式2と3の消費電力量の比率
(実行方式3が勝る場合)

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 α の値が求めれば，図7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， α の値が図7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を選定した方がよい場合となる．

また，図8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 α (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， $\alpha=0.05$ 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8)より， α 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 α が小さくなればこの比率は小さくなることを確認できる．従って，プロセッサ数が増えるにつれ α 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8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(2.2節：図2参照)．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 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 考察

本章では，4章での結果を基に，主要なパラメータが実行方式選定基準に与える影響について考察する．

5.1 k (= α / β) の影響

表2に示すように，PowerPC 405LP では，標準動作周波数から高速動作周波数への上げ率に対する消費電力の増加率(式(8)参照)は，例えば他のPowerPC製品に比べて比較的大きいものである．本プロセッサでは，動作周波数を f_0 から $2.5f_0$ 倍上げると，9倍以上もの消費電力を要する．

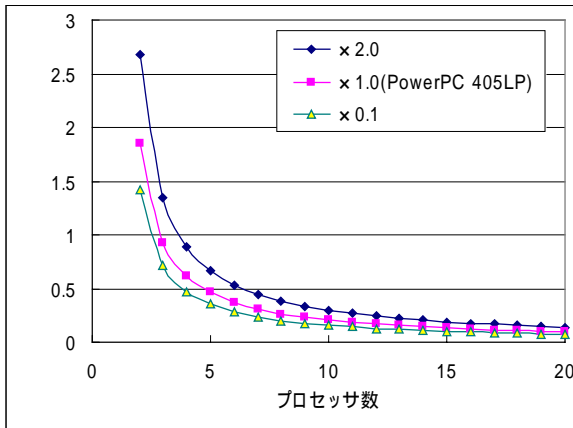


図 10 . P が に与える影響度合い

$$=(P / P) / (/) = (P / P) \cdot k \quad (8)$$

この割合()が実行方式 2 と 3 の選定基準にどの程度影響を与えるものなのか、 の値を変えた場合の例を示したものが図 9 である。ここでは、消費電力 P と P は一定にして k を変化させることにより、この割合()を変化させた。従って、式(8)より は k の値に比例する。なお、k=0.4(=3.77)とした場合が PowerPC 405LP の値である。この結果から、 および k が小さくなる度合いに比例して、省電力のための実行方式として実行方式 2 が有利になる度合いが大きくなる事が分かる。

5.2 P の影響

本節では、プロセッサが遊休状態時の消費電力値 P が、実行方式 2 と 3 の選定基準にどの程度影響を及ぼすかを考察する。図10は、PowerPC 405LPの遊休状態時の消費電力値 P =0.0115を基準に、この値を2倍にした場合(図10中「x 2.0」)と10分の1にした場合(図10中「x 0.1」)の、実行方式 2 と 3 の選定基準への影響を示したものである。この結果から、遊休状態時の消費電力値 P が10分の1に激減したとしても、実行方式 2 と 3 の選定基準値 はプロセッサ数が 2 の場合、約0.4ポイントしか下がらない。逆に、消費電力値 P が 2 倍になると、同様にプロセッサ数が 2 の場合、倍の約0.8ポイントも上昇してしまう。この結果から、基本的に P が小さくなれば省電力実行方式として実行方式 2 が有利になるが、P の選定基準への影響は P が小さくなるより大きくなる方が遥かに影響度が大きいことが分かる。

6 まとめと今後の課題

本稿では、近年盛んに研究開発されている動作周波数および電圧が可変できる省電力タイプのプロセッサにおいて、動作周波数に対する消費電力の変化率が線形でない点に着目し、並列演算環境上で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、本手法の具体例

を挙げ、その場合の選定基準を示した。本手法は、与えられた処理データの性質とプロセッサの性能および特徴に基づいて、省電力実行のためのプロセッサ数とその動作周波数および電圧を選定するというものである。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、本環境に提案手法を適用した場合の効果を示した。

今後の課題は、ここで検討した内容をまずはシミュレーションを用いて定量的に評価することである。また、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。更に、精度向上を目的にモデルを詳細化することも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[2]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。その他、動作周波数の切り替え時間に関しては、異なるプロセッサではあるが数ミリ秒で実現できるとの報告[4]もあり、制限時間と比べて充分小さければ問題ないと考えるが、回数が多い場合や、その場合のプロセッサの寿命への影響を調べることは今後の課題の一つである。特に、我々が対象としているシステムは連続的に運用する形態のものであるため、プロセッサの寿命短縮は深刻な問題となる。

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と考える。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 and K.Skadron, "Power-Aware Computing," Computer, IEEE COMPUTER SOCIETY, vol.36, no.12, pp.35-38, 2003.
- [2] 堀田義彦, 佐藤三久, 朴 泰祐, 高橋大介, 高橋睦史, 中村 宏, "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.
- [3] 十山圭介, 三坂 智, 相坂一夫, 在塚俊之, 内山邦男, 石橋孝一郎, 川口 博, 桜井貴康, "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.
- [4] 西山博仁, 志賀 稔, "センサーノードにおける動的電力制御方式," 2004 年 電子情報通信学会総合大会, p.B-9-7, 2004.