

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6083278号
(P6083278)

(45) 発行日 平成29年2月22日(2017.2.22)

(24) 登録日 平成29年2月3日(2017.2.3)

(51) Int.Cl. F I
G06F 9/50 (2006.01) G O 6 F 9/46 4 6 5 D
G06F 1/32 (2006.01) G O 6 F 1/32 Z

請求項の数 8 (全 20 頁)

(21) 出願番号	特願2013-60736 (P2013-60736)	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成25年3月22日 (2013.3.22)		神奈川県川崎市中原区上小田中4丁目1番1号
(65) 公開番号	特開2014-186522 (P2014-186522A)	(74) 代理人	100090273 弁理士 園分 孝悦
(43) 公開日	平成26年10月2日 (2014.10.2)	(72) 発明者	川邊 幸仁 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
審査請求日	平成27年11月6日 (2015.11.6)	(72) 発明者	岡野 廣 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	大塚 俊範

最終頁に続く

(54) 【発明の名称】 計算システム及びその電力管理方法

(57) 【特許請求の範囲】

【請求項1】

複数のプロセッシングエレメントと、

異なる複数の動作負荷を特定の基準周波数下で処理した際のプロセッシングエレメントのダイナミック電力と、前記動作負荷の最適周波数とを関連付けた最適周波数導出表を用いて、前記基準周波数で並列処理プログラムを前記複数のプロセッシングエレメントのうち少なくとも1個のプロセッシングエレメントに仮動作させたときのプロセッシングエレメントのダイナミック電力を基に、並列処理プログラムを実行するプロセッシングエレメントの性能最適周波数及び前記性能最適周波数に対応する電源電圧を導出する導出部と、

前記導出部により導出された前記性能最適周波数、前記電源電圧及び計算システムの制約電力を基に、前記プロセッシングエレメントの稼働数を設定する設定部とを有することを特徴とする計算システム。

【請求項2】

前記設定部は、前記複数のプロセッシングエレメントのすべてを稼働するよりも性能が高くなるように、前記プロセッシングエレメントの稼働数を設定することを特徴とする請求項1記載の計算システム。

【請求項3】

前記導出部は、性能が最大となるように前記性能最適周波数及び前記電源電圧を導出することを特徴とする請求項1又は2記載の計算システム。

【請求項 4】

前記導出部は、複数の周波数及び電源電圧の組みのうちから、性能が最大となる組みの前記性能最適周波数及び前記電源電圧を導出することを特徴とする請求項 1～3 のいずれか 1 項に記載の計算システム。

【請求項 5】

前記設定部は、前記導出部により導出された前記性能最適周波数及び前記電源電圧を基に前記プロセッシングエレメントの変更後電力を導出し、前記導出したプロセッシングエレメントの変更後電力及び前記計算システムの制約電力を基に、前記プロセッシングエレメントの稼働数を設定することを特徴とする請求項 1～4 のいずれか 1 項に記載の計算システム。

10

【請求項 6】

前記複数のプロセッシングエレメントのうちの少なくとも 1 個は、メモリと、前記メモリを制御するメモリ制御回路と、第 1 の検出回路とを有し、前記メモリ制御回路は、第 1 のバッファを有し、前記第 1 の検出回路は、前記第 1 のバッファのフル率を検出し、前記導出部は、前記第 1 のバッファのフル率を基に、前記性能最適周波数を導出することを特徴とする請求項 1～5 のいずれか 1 項に記載の計算システム。

【請求項 7】

さらに、前記複数のプロセッシングエレメントに接続されるネットワークを有し、前記複数のプロセッシングエレメントのうちの少なくとも 1 個は、前記ネットワークの入出力を制御する入出力制御回路と、第 2 の検出回路とを有し、前記入出力制御回路は、第 2 のバッファを有し、前記第 2 の検出回路は、前記第 2 のバッファのフル率を検出し、前記導出部は、前記第 2 のバッファのフル率を基に、前記性能最適周波数を導出することを特徴とする請求項 1～6 のいずれか 1 項に記載の計算システム。

20

【請求項 8】

複数のプロセッシングエレメントを有する計算システムの電力管理方法であって、導出部により、異なる複数の動作負荷を特定の基準周波数下で処理した際のプロセッシングエレメントのダイナミック電力と、前記動作負荷の最適周波数とを関連付けた最適周波数導出表を用いて、前記基準周波数で並列処理プログラムを前記複数のプロセッシングエレメントのうちの少なくとも 1 個のプロセッシングエレメントに仮動作させたときのプロセッシングエレメントのダイナミック電力を基に、並列処理プログラムを実行するプロセッシングエレメントの性能最適周波数及び前記性能最適周波数に対応する電源電圧を導出し、

30

設定部により、前記導出部により導出された前記性能最適周波数、前記電源電圧及び計算システムの制約電力を基に、前記プロセッシングエレメントの稼働数を設定することを特徴とする計算システムの電力管理方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、計算システム及びその電力管理方法に関する。

【背景技術】

【0002】

産業・基礎科学など様々な分野での大規模な並列シミュレーションの活用増大により超並列高性能計算システム(いわゆるスーパーコンピュータ)の性能は年々高くなっている。このような計算システムの大規模化に向けた様々な課題の中に、消費電力の削減がある。CPU (Central Processing Unit) の演算性能は上がっているのに、計算システムの消

50

費電力制約のために、計算システムは処理性能を出し切れないという状況になりつつある。計算システムの低消費電力化と同様に与えられた消費電力制約下で、ジョブの実行効率を最大化することも重要となってきた。

【0003】

サーバやデータセンタにおいては、多数のCPUで構成されるシステムの消費電力制約を超えない範囲でジョブを実行させるための技術として、CPU側の技術としては動的に周波数を調整するDFS (Dynamic Frequency Scaling) や周波数に合わせて電圧も制御するDVFS (Dynamic Voltage and Frequency Scaling) の技術が知られている。また、CPU内のレジスタに消費電力上限を設定して、その設定を自律的に守らせるPower Capping技術、又はPower Capping技術で電力が上限を超えた時の急速な消費電力低減に使用されるClock Throttling (クロックパルスのマスクによる周波数の強制低減) の技術が知られている。また、Pipeline Throttling (命令発行制限やメモリアクセス制限等による命令処理量の強制低減) の技術が知られている。

10

【0004】

システム側の電力管理技術は、システムの電力制約、電力管理ポリシー又は電力管理モードなどが別途設定されていれば、その設定に基づき稼働中の各CPUに対する消費電力割り当て値を決定する。各CPUは、その割り当て値を守るように前述のCPU電力管理技術を使用してCPU電力を制御することでシステム全体の消費電力制約を守らせる。

【0005】

また、少なくとも1台以上の演算器と制御部とを有する並列計算装置が知られている (例えば、特許文献1参照)。演算器は、その個数が処理タイミングによって変動し、かつ、それぞれの処理時間が一定のタスクを処理する。制御部は、演算器全体の処理能力を一定として、タスク処理に使用する演算器の台数と、使用する演算器の動作周波数を制御する。

20

【0006】

また、並列コンピュータにおける事前対応型電力管理の方法が知られている (例えば、特許文献2参照)。並列コンピュータは、サービス・ノードと複数の計算ノードとを含む。サービス・ノードは、帯域外サービス・ネットワークを介して計算ノードに接続されている。各計算ノードは、コンピュータ・プロセッサと、コンピュータ・プロセッサに動作可能に連結されているコンピュータ・メモリとを含む。

30

【先行技術文献】

【特許文献】

【0007】

【特許文献1】特開2006-190104号公報

【特許文献2】特表2010-537266号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

並列処理プログラムの中には、並列化可能な部分と並列化不可能な逐次処理部分が存在する。並列処理プログラムの処理時間Tは、おおよそ次式のように示すことができる。ここで、Cpは、並列処理部分ののベサイクル数である。Csは、逐次処理部分ののベサイクル数である。Fpは、並列処理部分のクロック周波数である。Fsは、逐次処理部分のクロック周波数である。PLは、並列数である。

40

$$T = C_p / (F_p \times P_L) + C_s / F_s$$

【0009】

このように並列処理プログラムの処理時間Tは、並列処理部分と逐次処理部分で分離されているため、並列処理部分の高速化の効果と逐次処理部分の高速化の効果とを独立に考えることができる。逐次処理部分は、稼働CPU数が1個なので、DVFSを用いたクロック周波数及び電圧制御により高速化が可能である。しかし、並列処理プログラムの処理のうち並列処理部分の高速化には、改善の余地がある。

50

【 0 0 1 0 】

本発明の目的は、全プロセッシングエレメントを稼働させる場合より高い処理性能を達成することができる計算システム及びその電力管理方法を提供することである。

【課題を解決するための手段】

【 0 0 1 1 】

計算システムは、複数のプロセッシングエレメントと、異なる複数の動作負荷を特定の基準周波数下で処理した際のプロセッシングエレメントのダイナミック電力と、前記動作負荷の最適周波数とを関連付けた最適周波数導出表を用いて、前記基準周波数で並列処理プログラムを前記複数のプロセッシングエレメントのうちの少なくとも1個のプロセッシングエレメントに仮動作させたときのプロセッシングエレメントのダイナミック電力を基に、並列処理プログラムを実行するプロセッシングエレメントの性能最適周波数及び前記性能最適周波数に対応する電源電圧を導出する導出部と、前記導出部により導出された前記性能最適周波数、前記電源電圧及び計算システムの制約電力を基に、前記プロセッシングエレメントの稼働数を設定する設定部とを有する。

10

【発明の効果】

【 0 0 1 2 】

プロセッシングエレメントの周波数及び電源電圧の他に、プロセッシングエレメントの稼働数を設定することにより、全プロセッシングエレメントを稼働させる場合より高い処理性能を達成することができる。

【図面の簡単な説明】

20

【 0 0 1 3 】

【図1】図1は、本実施形態による計算システムの構成例を示す図である。

【図2】図2は、トランジスタのリーク電流を示すグラフである。

【図3】図3は、クロック周波数及び遅延時間の例を示すグラフである。

【図4】図4は、プロセッシングエレメントの稼働数及び計算システムの性能を示すグラフである。

【図5】図5は、計算システムの電力管理方法を示すフローチャートである。

【図6】図6(A)及び(B)は、図5の処理主体を説明するための図である。

【図7】図7は、図5の最適周波数導出表の構成例を示す図である。

【図8】図8は、プロセッシングエレメントの非稼働時電力に応じた最適動作点の変動を示すグラフである。

30

【図9】図9は、マルチコアプロセッサのCPUの構成例を示す図である。

【図10】図10は、バッファフル率検出回路を有するCPUの構成例を示す図である。

【図11】図11は、電力管理コントローラの処理例を示すフローチャートである。

【図12】図12は、本実施形態の効果の説明するためのグラフである。

【図13】図13は、図7の最適周波数導出表の例を示す図である。

【発明を実施するための形態】

【 0 0 1 4 】

図1は、本実施形態による計算システムの構成例を示す図である。計算システム100は、複数のプロセッシングエレメント(PE)101がネットワーク104に接続されている。各プロセッシングエレメント101は、処理を行う処理要素であり、演算処理用の中央処理装置(CPU)102及びCPU102に接続されたローカルメモリ103を有する。計算システム100は、単一のラックで構成されていてもよいし、複数のラックから構成されていてもよい。複数ラックの構成の場合は、プロセッシングエレメント101間のネットワーク104は、ラック内ネットワークとラック間ネットワークで構成される。

40

【 0 0 1 5 】

計算システム100では、処理対象とする並列処理プログラムの処理全体のうちの大部分を占める並列処理可能部分は並列プログラミングや並列化コンパイラ等により処理が類似した複数のサブタスクに分割される。これらのサブタスクは、プログラム実行時に各ブ

50

ロセッシングエレメント 101 (プロセッシングエレメント 101 がマルチコアプロセッサの場合はプロセッシングエレメント 101 内の各プロセッサコア) に割り当てられて、実行時に該当するプロセッシングエレメント 101 にロードされた上で並列に実行される。個々の並列処理プログラムは、タスクの規模により計算システム 100 全体で処理される場合もあれば、計算システム 100 全体を幾つかの部分に区画化したうちのひとつで実行される場合もある。後者の場合、各区画には計算システム 100 全体の消費電力制約値が何らかのポリシーに基づいてその区画の消費電力制約として分配される。各区画で処理される並列処理プログラムは、そのジョブを処理する区画に割り当てられた消費電力制約を満たすように実行されるため、この場合でも各区画を計算システム 100 の各区画の電力制約を計算システム 100 の電力制約とみなす。

10

【0016】

並列処理プログラムの並列処理部分は、プロセッシングエレメント 101 の数に対して十分大きい並列数を持つ。並列処理部分の分割数 (サブタスク数) は、並列処理プログラム作成時に予めパラメータ化されていたり、並列化コンパイラによる並列数を指定したコンパイル等により容易に変更可能である。また、並列実行数を調整することによるサブタスクの割り当てやサブタスク間の通信等のプロセッシングエレメント 101 間の通信時間の変動量は、各並列実行プロセッシングエレメント 101 の処理時間全体と比べて無視できるくらい小さいとする。

【0017】

プロセッシングエレメント 101 の 1 個当たり消費電力 P_a は、その CPU 102 内の全ゲートのクロック信号当たりの平均動作率を、配線及び素子のスイッチングにかかわる総容量を C 、CPU 102 の電源電圧を V 、CPU 102 のリーク電流を I_k とし、CPU 102 のクロック周波数 (動作周波数) を f とすると、次式 (1) のように表すことができる。ここで、 P_b は、CPU 102 の固定電力であり、 P_c は、メモリ関連電力である。

20

$$P_a = C \times V^2 \times f + I_k \times V + P_b + P_c \quad \dots (1)$$

【0018】

メモリ関連電力 P_c は、ローカルメモリ 103 自体の消費電力並びに CPU 102 及びローカルメモリ 103 間のデータ転送にかかる消費電力であり、メモリアクセスの発生頻度と相関がある。CPU 102 の固定電力 P_b は、CPU 102 内で別電源供給されたアナログマクロやシリアル I/O 等のように半導体チップ内のクロック周波数にかかわらずに、一定量消費される電力成分を表す。

30

【0019】

図 2 は、トランジスタのリーク電流 I_k を示すグラフである。横軸は、ドレイン及びソース間電圧 V_{ds} である。縦軸は、ドレイン及びソース間電流 I_{ds} である。DIBL (Drain Induced Barrier Lowering) 効果等により、電圧 V_{ds} が増加すると、閾値電圧 V_{th} が減少し、リーク電流 I_k が増加する。リーク電流 I_k の曲線は、トランジスタの動作時に変動させる程度の狭い範囲では、2 ~ 3 次式程度の正係数の多項式で近似することができる。例えば、リーク電流 I_k は、次式のように、2 次曲線 201 に近似することができる。ここで、電圧 V_{ds} は、例えば電源電圧 V であり、 a 及び b は定数である。

40

$$I_k = a \times V^2 + b$$

【0020】

図 3 は、クロック周波数 301 及び遅延時間 302 の例を示すグラフである。横軸は、電源電圧 V を示す。電源電圧 V が増加すると、素子のスイッチング性能が上がるため、遅延時間 301 が短くなり、クロック周波数 302 が上がる。逆に、電源電圧 V を下げていくと、素子のスイッチング性能が下がり、電源電圧 V が閾値電圧 V_{th} より下がると、素子が動作しにくくなる。電源電圧 V とクロック周波数 f の関係では、電源電圧 V は閾値電圧 V_{th} を切片としたクロック周波数 f の単調増加関数となり、実用的な電圧変動範囲では、1 次直線 303 に近似すると、次式になる。ここで、 a は定数である。

$$V = a \times f + V_{th}$$

50

【0021】

ローカルメモリ103は、それが接続されるCPU102のクロック信号の周波数fが増加すると、内部処理速度の上昇により、メモリアクセス頻度も増加するため、メモリ関連電力Pcは、次式により、ある固定値P0mを切片として、クロック周波数fが線形増加するとしてモデル化できる。

$$P_c = P_{0m} + f \times P_m$$

【0022】

このような関係を考慮して、上式(1)を、クロック周波数fについて整理すると、次式のように、正係数の多項式として近似することができる。ここで、C₀、C₁、C₂、C₃は、正の実数である。

【0023】

$$\begin{aligned} P_a &= \alpha C \times V^2 \times f + I_k \times V + P_b + P_c \\ &= \{ \alpha C \times V^2 \times f + I_k \times V \} + \{ P_{0m} + f \times P_m \} + P_b \\ &= \alpha C \times (\alpha f + V_{th})^2 \times f + (a \times V^2 + b) \times V + P_{0m} + f \times P_m \\ &+ P_b \\ &= \alpha C \times (\alpha f + V_{th})^2 \times f + (a \times (\alpha f + V_{th})^2 + b) \times (\alpha f + V_{th}) + P_{0m} + f \times P_m + P_b \\ &= (\alpha C \times \alpha^2 + a \times \alpha^3) \times f^3 + (2 \times \alpha C \times \alpha V_{th} + 3 \times a \times \alpha^2 \times V_{th}) \times f^2 + \dots \\ &= C_0 \times f^3 + C_1 \times f^2 + C_2 \times f + C_3 \end{aligned}$$

【0024】

プロセッシングエレメント101単体の性能は、クロック周波数fに比例するため、並列処理時の計算システム100全体の性能は並列処理を行っているプロセッシングエレメント101の数Nとそのクロック周波数fの積N×fに比例するとみなすことができる。ここで、計算システム100の電力をPs、プロセッシングエレメント101の電力をPpとすると、計算システム100の電力制約下での性能最大化を目指すためには、余剰電力が出ないように、Ps = N×Ppとなるように、プロセッシングエレメント数N、電源電圧V及びクロック周波数fを調整する。この時の計算システム100の性能Gsは、次式(2)のようになる。

【0025】

$$\begin{aligned} G_s \quad N \times f &= f \times P_s / P_p \\ &= f \times P_s / (C_0 \times f^3 + C_1 \times f^2 + C_2 \times f + C_3) \\ &= P_s / (C_0 \times f^2 + C_1 \times f + C_2 + C_3 / f) \quad \dots (2) \end{aligned}$$

【0026】

図4は、式(2)のプロセッシングエレメント101の稼働数N及び計算システム100の処理性能Gsを示すグラフである。横軸は、CPU102のクロック周波数fを示す。式(2)のように、計算システム100の性能Gsは、クロック周波数fの関数として表され、f>0で極大点401を一つ持つ曲線となる。ここまでの説明は、計算システム100内で非稼働となったプロセッシングエレメント101の消費電力をほぼ0とみなしたモデルの説明である。次に、非稼働時のプロセッシングエレメント101の電力が0ではない場合を考える。プロセッシングエレメント101の非稼働時電力をC₄とし、計算システム100内の全プロセッシングエレメント101の数MのうちN個が稼働中とすると、上式(2)の性能Msは、次式(3)のようになる。

$$G_s \quad N \times f = (P_s - C_4 \times M) / \{ C_0 \times f^2 + C_1 \times f + C_2 + (C_3 - C_4) / f \} \quad \dots (3)$$

【0027】

また、稼働時の最適のプロセッシングエレメント101の電力Pp1が導出された後の最適のプロセッシングエレメント101の数Nは次式で求められる。

$$N = (P_s - M \times C_4) / (P_{p1} - C_4) \quad \dots (4)$$

【0028】

10

20

30

40

50

この場合も、式(4)の分子は固定値であり、式(4)の分母は式(2)と同様の形となるので、式(2)と同様に、計算システム性能 G_s は、図4のような、極大点401を一つ持つものとなる。ここで、非稼働時にクロック信号が停止すると考えれば、 C_3 、 C_4 とみなすことができるため、 $1/f$ の係数も正と考えることができる。

【0029】

このことから、計算システム100の消費電力制約の下で、計算システム100の性能 G_s を最大化するクロック周波数 f とそれに対応する電源電圧 V 及びプロセッシングエレメント101の数 N の最適値が存在し、使用可能なプロセッシングエレメント101をすべて使用することが、必ずしも計算システム100の性能 G_s の最大化につながらないということがわかる。

10

【0030】

高性能の超並列計算システム100では、プロセッシングエレメント101間通信による処理性能の低下を抑えるために、プロセッシングエレメント101間通信として消費電力の大きな高速入出力(I/O)制御回路を用いたシリアル転送を用いることができる。その場合には、入出力制御回路のアクセスの有無にかかわらず、常時大電力を消費し続けることになり、上のプロセッシングエレメント101の消費電力式の固定電力成分が非常に大きくなる。式(2)の性能 G_s の最適点は、式の性質上、図4において、プロセッシングエレメント101の固定電力成分 C_3 が大きくなると、クロック周波数 f が大きい側(プロセッシングエレメント101数 N が少ない側)に移る。そのため、このような固定電力成分 C_3 の大きいプロセッシングエレメント101は全プロセッシングエレメント101を稼働させるより、稼働させるプロセッシングエレメント101の数 N を絞って、その分、電源電圧 V やクロック周波数 f を上げた方が計算システム100全体の性能 G_s は高くなるケースが多い。

20

【0031】

また、式(2)からわかるように、図4の特性グラフの性能最大点401は、 C_0 及び C_1 が大きい(実行するプログラムの消費電力が大きい)と、クロック周波数 f が低い側(プロセッシングエレメント101数 N が多い側)に移る。このように、最適動作点401は、実行するプログラムの消費電力によっても変動することがわかる。

【0032】

本実施形態は、動作点401を基に、稼働するプロセッシングエレメント101の数 N 、クロック周波数 f 及び電源電圧 V を調整することにより、計算システム100の性能 G_s の最大化を図る。

30

【0033】

図5は、計算システム100の電力管理方法を示すフローチャートである。図5の処理は、ステップS511のプロセッシングエレメント101の特性情報取得処理と、ステップS512の性能最大化プロセッシングエレメント数及び動作条件算出処理と、ステップS505の動作条件設定処理と、ステップS506の並列処理プログラム実行処理とを有する。ステップS511は、ステップS501及びS502を有する。ステップS512は、ステップS503及びS504を有する。

【0034】

図6(A)は、図5の処理主体を説明するための図である。計算システム100は、複数のCPU102を有する。各CPU102は、複数の演算コア601を有するマルチコアプロセッサである。1個のCPU102内の1個の演算コア601は、プログラムを実行することにより図5の処理を行い、各CPU102から情報603を入力し、各CPU102に対して電源電圧 V 及びクロック周波数 f などを設定するための制御信号602を出力する。また、1個のプロセッシングエレメント101が図5の処理を行ってもよい。この場合、図5の処理を行うプロセッシングエレメント101は、図1のネットワーク104を介して、他のプロセッシングエレメント101に対して通信を行う。

40

【0035】

図6(B)は、他の図5の処理主体を説明するための図である。計算システム100は

50

、複数のCPU102の他に、電力管理コントローラ611を有する。電力管理コントローラ611は、プログラムを実行することにより図5の処理を行い、各CPU102から情報603を入力し、各CPU102に対して電源電圧V及びクロック周波数fなどを設定するための制御信号602を出力する。この場合、電力管理コントローラ611は、図1のネットワーク104とは別のネットワークを介して、プロセッシングエレメント101に対して通信を行う。以下、電力管理コントローラ611が図5の処理を行う例を説明する。

【0036】

稼働させるすべてのプロセッシングエレメント101は、各並列タスクの処理時間を揃えるために、同一のクロック周波数fに設定される。電源電圧Vは、全プロセッシングエレメント101で同一でもよいし、プロセッシングエレメント101毎のプロセスばらつきに応じて調整してもよい。

10

【0037】

プロセッシングエレメント101の消費電力、特にCPU102の電力については、ダイナミック電力とリーク電力がある。リーク電力については、計算システム100が水冷等により低温で温度管理され、リーク電力の変動が小さい領域で動作するという前提で、既知の固定値として扱う。プロセッシングエレメント101は、動作時に内部の各部の稼働情報を記録して、それを基に自身の消費電力(ダイナミック電力)を演算する機能を持つ。

【0038】

電力管理コントローラ611は、同一のクロック周波数fの下でプロセッシングエレメント101の消費電力の異なる複数の動作負荷において、下記の3個のパラメータを求める。

20

【0039】

(1) その負荷を特定の基準周波数下で処理した際のプロセッシングエレメント101のダイナミック電力

(2) その負荷での最適周波数(図4の最適動作点401)

(3) その負荷で周波数を(2)の最適周波数に変更した場合のプロセッシングエレメント101の電力

【0040】

図7は、図5の最適周波数導出表521の構成例を示す図である。最適周波数導出表521は、例えば、電力管理コントローラ611内に記憶される。最適周波数導出表521は、規定周波数で実行した時のプロセッシングエレメント101のダイナミック電力値 $P_1 \sim P_m$ と、その時の性能最適周波数 $F_1 \sim F_m$ と、その性能最適周波数 $F_1 \sim F_m$ 及びそれに対応する電源電圧 $V_1 \sim V_m$ を使用した時の変更後電力 $P_{1a} \sim P_{ma}$ を有する。最適周波数導出表521は、予め作成される。

30

【0041】

上の(1)～(3)の情報は、プロセッシングエレメント101の電力及び周波数特性がわかっているならば求めることができるため、例えば、設計時の設計データを用いた消費電力解析や、製造後、プロセッシングエレメント101単体での実機測定等により、計算システム100を組む前に、最適周波数導出表521を求めることができる。

40

【0042】

最適周波数導出表521は、後述する検索処理のために、予めダイナミック電力 $P_1 \sim P_m$ の大きさで、エントリをソートしておく。最適周波数導出表521は、式(2)の $G_s = f \times P_s / P_p$ のうちの f / P_p の部分で決まる値で作成されるため、並列処理プログラムを計算システム100全体で処理する場合でも、区画化した一部分で処理する場合でも、同じ最適周波数導出表521を使用することができる。

【0043】

実行される並列処理プログラムは、図5の処理開始前に並列実行数Nを実行時に指定できるように、Nをパラメータ化して作成(コンパイル)されているか、このプログラムの

50

処理に割り当てられる全プロセッシングエレメント 101 の数 M と同じかそれより小さい並列数で、並列化コンパイラによって予めコンパイルされている。

【0044】

まず、ステップ S501 では、電力管理コントローラ 611 は、電力情報取得用の並列処理プログラムを各プロセッシングエレメント 101 にテスト実行させる。この処理は、消費電力が計算システム 100 の制約電力を超えないような十分低い周波数で実行し、定常特性を取得するのに十分な時間実行させる。

【0045】

次に、ステップ S502 では、電力管理コントローラ 611 は、1 個又は複数のプロセッシングエレメント 101 の並列処理部分の定常的なダイナミック電力 P_d を取得する。並列処理部分について、定常的なダイナミック電力の取得に十分な長さだけ並列処理プログラムを実行した上で、その時のプロセッシングエレメント 101 のダイナミック電力 P_d を取得する。

【0046】

ダイナミック電力 P_d を取得するプロセッシングエレメント 101 は 1 個でもいいし、ダイナミック電力 P_d のばらつきを考慮する場合は、複数のプロセッシングエレメント 101 からダイナミック電力 P_d を取得した上で、平均処理等を行い、1 個のダイナミック電力 P_d を求めてもよい。

【0047】

また、プロセッシングエレメント 101 単体においても、異なる期間の定常状態のダイナミック電力 P_d を保持する複数レジスタを用意するなどして、複数のダイナミック電力 P_d を取得してもよい。電力管理コントローラ 611 は、得られた複数のダイナミック電力 P_d を基に、平均ダイナミック電力 P_d を用いる。

【0048】

また、電力管理コントローラ 611 は、統計処理により分散を考慮して例えば「平均 + 1」相当のダイナミック電力 P_d を求めてもよい。

【0049】

また、複数のプロセッシングエレメント 101 が同時動作することにより、ばらつきが重なり合って緩和される効果を加味する為の係数 α を導入し、以下の式で算出したダイナミック電力 P_d を用いてもよい。

$$P_d = \text{平均電力値} + (\text{最大電力} - \text{平均電力}) \times \alpha$$

【0050】

次に、ステップ S512 の性能最大化プロセッシングエレメント数及び動作条件算出処理を説明する。

【0051】

ステップ S503 では、電力管理コントローラ 611 は、得られたダイナミック電力 P_d を基に最適周波数導出表 521 を検索し、その並列処理プログラムに最適な周波数 f を求める。例えば、図 7 の最適周波数導出表 521 を用いて、ダイナミック電力が P_1 の場合には、最適な周波数が F_1 になる。また、得られたダイナミック電力 P_d が、図 7 の最適周波数導出表 521 の i 番目のエントリのダイナミック電力 $P(i)$ と $i+1$ 番目のエントリのダイナミック電力 $P(i+1)$ の間の場合、電力管理コントローラ 611 は、 i 番目のエントリの最適周波数 $F(i)$ と $i+1$ 番目のエントリの最適周波数 $F(i+1)$ を用いて、次式の線形補間により、最適周波数 f を導出する。

$$f = F(i) + \{ F(i+1) - F(i) \} \times \{ P_d - P(i) \} / \{ P(i+1) - P(i) \}$$

【0052】

また、電力管理コントローラ 611 は、上記の最適周波数 f に対応する最適電源電圧 V を求める。最適電源電圧 V は、最適周波数 f で CPU 102 を正常動作させることができる電源電圧である。

【0053】

10

20

30

40

50

次に、ステップS504では、電力管理コントローラ611は、図7の最適周波数導出表521を用いて、求めた最適周波数 f 及び最適電源電圧 V の動作時のプロセッシングエレメント101の電力を変更後電力 $P_{1a} \sim P_{ma}$ として求める。この場合も、電力管理コントローラ611は、上記の最適周波数 f の導出と同様に、線形補間により、変更後電力 $P_{1a} \sim P_{ma}$ を求める。

【0054】

次に、電力管理コントローラ611は、計算システム100の制約電力 P_s 、最適動作時の変更後電力 P_p ($P_{1a} \sim P_{ma}$)、全プロセッシングエレメント数 M 及びプロセッシングエレメント101の非稼働時電力 C_4 を基に、式(4)を用いて、次式により、最適稼働プロセッシングエレメント数 N を導出する。

$$N = (P_s - C_4 \times M) / (P_p - C_4)$$

【0055】

ここで、最適稼働プロセッシングエレメント数 N が全プロセッシングエレメント数 M より多い場合は、最適稼働プロセッシングエレメント数 N を M とする。その場合、各プロセッシングエレメント101は、 P_s / M の電力で動作させられる。この制御は、ステップS511の処理と同様のテスト実行を行い、最適周波数 f 及び最適電源電圧 V の組みを変化させながら、取得するダイナミック電力 P_d が P_s / M となる最適周波数 f 及び最適電源電圧 V の組みを見つけてもいいし、Power Capping手法のように各プロセッシングエレメント101に制約電力 P_s / M を設定して、各プロセッシングエレメント101が設定電力以下で動作するように自律的に制御を行わせてもよい。

【0056】

次に、ステップS505では、電力管理コントローラ611は、上記の最適稼働プロセッシングエレメント数 N に基づき稼働させるプロセッシングエレメント101を決定し、稼働させるプロセッシングエレメント101の上記の最適周波数 f 及び最適電源電圧 V を設定する。また、電力管理コントローラ611は、オペレーティングシステム(OS)等の並列タスクをスケジューリングするプロセスに最適稼働プロセッシングエレメント数 N を通知する。電力管理コントローラ611は、実行するタスクが並列実行数固定でコンパイルされている場合は、本処理内で対象タスクを並列化コンパイラを用いて、並列数を最適稼働プロセッシングエレメント数 N として、対象タスクを再コンパイルする。タスクの並列数がパラメータ化されて作成されているプログラムの場合は、電力管理コントローラ611は、実行時の並列数の設定を最適稼働プロセッシングエレメント数 N に設定する。

【0057】

次に、ステップS506では、電力管理コントローラ611は、ステップS505の動作条件設定に基づいて、対象の並列処理プログラムを各プロセッシングエレメント101に実行させる。

【0058】

図8は、プロセッシングエレメント101の非稼働時電力 C_4 に応じた最適動作点の変動を示すグラフである。計算システム100の性能801は、プロセッシングエレメント101の非稼働時電力 C_4 が大きい場合の性能であり、最適動作点802を有する。計算システム100の性能803は、プロセッシングエレメント101の非稼働時電力 C_4 が

【0059】

式(3)から分かるように、プロセッシングエレメント101の非稼働時電力 C_4 の値が変わると、分母の $1/f$ の係数 $C_4 - C_3$ が変化するため、計算システム100の性能が最適となる周波数も変動する。つまり、プロセッシングエレメント101が休止時の消費電力が異なる複数の低電力モードを持っていた場合、休止させたプロセッシングエレメント101をどの低電力モードで待機させるかによって、最適な動作点802又は804が変わる。よって、プロセッシングエレメント101が休止時電力の異なる複数の低電力モードを持っていた場合、その低電力モードごとに図7の最適周波数導出表521を用意し、計算システム100の低電力モード設定によって、使用する最適周波数導出表521を

10

20

30

40

50

切り替えることにより、休止時電力が異なるモードを持つ場合にも対応することができる。

【0060】

図9は、マルチコアプロセッサのCPU102の構成例を示す図である。CPU102は、マルチコアプロセッサであり、複数の演算コア601を有する。また、CPU102は、共有キャッシュメモリ901、メモリ制御回路902及び入出力(I/O)制御回路903を有する。各演算コア601は、共有キャッシュメモリ901を介して、メモリ制御回路902に対して、コマンドCMDを出力し、データDTを入出力する。また、各演算コア601は、共有キャッシュメモリ901を介して、入出力制御回路903に対して、コマンドCMDを出力し、データDTを入出力する。メモリ制御回路902は、内部バッファを有し、ローカルメモリ103に対してデータを読み書き制御し、内部バッファがフルになるとビジー信号MBSYを出力する。入出力制御回路903は、内部バッファを有し、ネットワーク104に対してデータを入出力制御し、内部バッファがフルになるとビジー信号IBSYを出力する。

【0061】

これまでの説明における計算システム100をマルチコアCPU102と考え、これまでの説明におけるプロセッシングエレメント101を演算コア601と考え、規模を縮小してこれまでの式を当てはめて考える。その場合、演算コア601は、式(1)の固定電力に相当する周波数 f 及び電源電圧 V に依存しない電力成分がほとんどないため、式(2)及び(3)の性能 G_s で固定電力が0であるとみなして考えることにより、プロセッシングエレメント101内の全演算コア601を使用した場合が一番電力当たりの性能が高くなることが分かる。すなわち、周波数 f を低くし、その分、大量のプロセッシングエレメント101を動作させた方が全体の性能が上がる。このことから、プロセッシングエレメント101がマルチコアプロセッサの場合は、プロセッシングエレメント101内の演算コア601は全て動作させた上で、プロセッシングエレメント101の数 N を本実施形態で決定することにより、性能を最大化できる。これにより、プロセッシングエレメント101がマルチコアプロセッサの場合でも、性能を最大化することが可能となる。

【0062】

これまでの説明では、性能 G_s は $N \times f$ に比例するという前提で行ってきた。しかし、例えばメモリアクセスが多い処理の場合は、周波数 f を上げても、メモリアクセスの処理待ちがボトルネックとなって、いくら周波数 f を上げても、実際の処理性能は上がらず、 $G_s = N \times f$ という前提が崩れる場合がある。この代表的なものがCPU102のメモリ制御回路902や入出力制御回路903である。これらの制御回路902及び903は、仕様上、最大バンド幅や通信レイテンシが決まっているため、いくらCPU102内部の周波数 f を上げても、速くはならない。本実施形態により求めた最適周波数 f が、このようなメモリアクセスや入出力アクセスによる性能劣化が大きい領域に入っていた場合は、最高性能は達成できずに、性能向上に結び付かない無駄な電力が消費されることになる。このようなケースの対策として、以下の様に、メモリアクセス又は入出力アクセスによる性能劣化を検知して、周波数が無駄に高くない機構を入れて、消費電力を抑制する。無駄な電力消費を抑えることにより、電気代の削減や故障発生率の抑制などにつながる。計算システム100を区画化して、複数のジョブを実行している場合には、上位のシステム制御において、この機構により生じた消費電力の余剰を別の区画で処理されているジョブの電力制約値に加えることにより、その別の区画のジョブの性能をさらに上げるといったことも可能となる。

【0063】

図10は、バッファフル率検出回路を有するCPU102の構成例を示す図である。CPU102は、バッファフル率検出回路を有する。バッファフル率検出回路は、クロックカウンタ1001、カウンタ1002、除算器1003及びバッファフル率レジスタ1004を有し、メモリ制御回路902のバッファのバッファフル率を検出する。クロックカウンタ1001及びカウンタ1002は、リセット信号RSによりカウント値がリセット

される。クロックカウンタ1001は、クロック信号をカウントする。メモリ制御回路902は、内部バッファがフルになるとビジー信号MBSYを出力する。カウンタ1002は、メモリ制御回路902のビジー信号MBSYをカウントする。除算器1003は、カウンタ1002のカウント値とクロックカウンタ1001のカウント値との除算を行い、除算結果をバッファフル率として出力する。バッファフル率レジスタ1004は、保持信号HLDを入力すると、除算器1003が出力するバッファフル率を保持する。メモリ制御回路902のバッファフル率を検出する方法を説明したが、同様に、図9の入出力制御回路903のバッファフル率の検出回路を設けることができる。

【0064】

メモリアクセス又は入出力アクセスによる性能劣化を検出する回路として、CPU102内のメモリ制御回路902及び入出力制御回路903のバッファにおいて、カウンタ1002は、バッファがフルになったクロック数をカウントする。バッファフル率レジスタ1004は、バッファフル率を保持する。バッファフル率は、全実行クロック数のうちのバッファフルとなったクロック数の割合である。

10

【0065】

図11は、電力管理コントローラ611の図10のCPU102に対する処理例を示すフローチャートであり、図5のステップS511及びS512において最適周波数を求めた後かつステップS506の前に行われる処理である。この処理は、サンプル取得として十分な数のプロセッシングエレメント101で実行すればよく、全プロセッシングエレメント101で実行する必要はない。

20

【0066】

ステップS1001では、電力管理コントローラ611は、リセット信号RSの印加及び解除を行い、クロックカウンタ1001及びカウンタ1002のカウント値をリセットする。

【0067】

次に、ステップS1002では、電力管理コントローラ611は、一定期間（定常状態として十分な期間）プロセッシングエレメント101を動作させる。

【0068】

次に、ステップS1003では、電力管理コントローラ611は、保持信号HLDにより、バッファフル率をレジスタ1004に格納する。

30

【0069】

次に、ステップS1003では、電力管理コントローラ611は、バッファフル率レジスタ1004に格納されているバッファフル率を読み出す。

【0070】

次に、電力管理コントローラ611は、バッファフル率が閾値より小さい場合は、メモリボトルネックは無視できるとみなして、設定された最適周波数で図5のステップS506の処理を各プロセッシングエレメント101に実行させる。

【0071】

また、電力管理コントローラ611は、バッファフル率が閾値より大きい場合は、最適周波数から周波数を段階的に下げながら、上記と同様に、バッファフル率を取得する。そして、電力管理コントローラ611は、バッファフル率が閾値より小さくなった時点の周波数を設定し、図5のステップS506の処理を行う。

40

【0072】

このような制御により、メモリボトルネックがある場合に無駄に周波数を上げることなく、適切な消費電力で並列処理を実行することができる。

【0073】

これまでの説明からわかるように、計算システム100の性能特性から求められる最適プロセッシングエレメント数Nが計算システム100の全プロセッシングエレメント数Mより小さいような動作状況の場合、全プロセッシングエレメント101を動作させた時には、図4の特性グラフ上で最適点401ではない動作点で動作していることになる。その

50

ため、その動作点の性能と最適動作点 401 の性能の差分に当たる分だけ、本実施形態を用いた方が性能が向上することがわかる。

【0074】

図12は、本実施形態の効果を説明するためのグラフであり、計算システム100の性能1201及び稼働プロセッシングエレメント数1208を示す。例えば、計算システム100で図12のような特性を持った並列処理プログラムを実行したとする。計算システム100の全プロセッシングエレメント数1206が17000個だった場合、17000個のすべてのプロセッシングエレメント101を動作させた場合は、動作点1202で動作して、計算システム性能1201は8.0 [a.u.] となる。これに対し、本実施形態の最適動作点1203は、稼働プロセッシングエレメント数1205が12500個であり、周波数1204が1.6 [a.u.] である。本実施形態の最適動作点1203で動作させた場合には、計算システム性能1207は9.5 [a.u.] となり、全プロセッシングエレメント稼働の動作点1202に対して、約19%の性能1209の向上となる。

10

【0075】

図13は、図7の最適周波数導出表521の例を示す図である。以下、図13の最適周波数導出表521を用いた処理例を、図5を参照しながら説明する。例えば、全プロセッシングエレメント数Mは17000個であり、計算システム制約電力は950kWであり、非稼働プロセッシングエレメント電力C₄は11Wである。

【0076】

ステップS501では、電力管理コントローラ611は、まず最適周波数導出表521が前提としている基準動作モードの周波数f及び電源電圧Vで並列処理プログラムを各プロセッシングエレメント101に仮動作させる。

20

【0077】

次に、ステップS502では、電力管理コントローラ611は、並列処理を受け持つプロセッシングエレメント101の定常的なダイナミック電力を取得する。このダイナミック電力が38Wだったとする。これは、前に述べたように複数の電力平均でもよいし、電力値のばらつきを考慮した補正值でもよい。

【0078】

次に、ステップS503では、電力管理コントローラ611は、図13の最適周波数導出表521を基に、ダイナミック電力が38Wの時の最適周波数を求める。図13の最適周波数導出表521は、2行目が36Wであり、3行目が40Wであるので、最適周波数fは、次式の線形補間により、求められる。

30

$$f = 1.7 + (1.5 - 1.7) \times (38 - 36) / (40 - 36) \\ = 1.6 \text{ [a.u.]}$$

その後、電力管理コントローラ611は、最適周波数fに対応する電源電圧Vを導出する。

【0079】

次に、ステップS504では、電力管理コントローラ611は、図13の最適周波数導出表521を基に、次式の線形補間により、プロセッシングエレメント101の変更後電力P_pを導出する。

40

$$P_p = 73 + (71 - 73) \times (1.5 - 1.6) / (1.5 - 1.7) \\ = 72 \text{ [W]}$$

【0080】

次に、電力管理コントローラ611は、式(4)を用いて、最適稼働プロセッシング数Nを導出する。

$$N = (950000 - 17000 \times 11) / (72 - 11) \\ 12500 \text{ 個}$$

【0081】

次に、ステップS505では、電力管理コントローラ611は、上記で求めた最適周波数(=1.6 [a.u.])、それに対応する電源電圧、及び稼働プロセッシングエレメン

50

ト数 (1 2 5 0 0 個) を計算システム 1 0 0 の設定に反映させる。さらに、電力管理コントローラ 6 1 1 は、プログラム並列数設定が 1 2 5 0 0 でない場合は、パラメータ設定及び再コンパイル等の方法で並列数を 1 2 5 0 0 にする。

【 0 0 8 2 】

次に、ステップ S 5 0 6 では、電力管理コントローラ 6 1 1 は、上記で行った計算システム 1 0 0 の設定及び並列数設定に基づいて、各プロセッシングエレメント 1 0 1 に並列処理プログラムを実行させる。

【 0 0 8 3 】

次に、プロセッシングエレメント 1 0 1 が図 7 に示す m 個の動作モードを有する例を説明する。m 個の動作モードは、それぞれ、周波数及び電源電圧の組みが設定されている。例えば、第 1 の動作モードは、周波数 F 1 及びそれに対応する電源電圧 V 1 が設定されている。すなわち、プロセッシングエレメント 1 0 1 は、異なる複数の周波数 F i 及び各周波数 F i で CPU 1 0 2 を正常動作させる電源電圧 V i の組 (F i , V i) として規定される複数の動作モード M i を持つ。ここで、i は自然数である。動作モードは、図 5 の処理を実行するプログラムから設定可能である。この場合、図 5 のステップ S 5 0 3 では、複数の動作モード M i の中から 1 個の動作モードを選択する。

【 0 0 8 4 】

ステップ S 5 0 3 では、電力管理コントローラ 6 1 1 は、上記と同様に、図 7 の最適周波数導出表 5 2 1 を参照し、得られたダイナミック電力 P d を用いて、次式の線形補間により、最適周波数 f を導出する。

$$f = F (i) + \{ F (i + 1) - F (i) \} \times \{ P d - P (i) \} / \{ P (i + 1) - P (i) \}$$

【 0 0 8 5 】

次に、電力管理コントローラ 6 1 1 は、動作モードとして最適周波数 f より周波数が低く最も最適周波数 f に近い周波数 F i の動作モード M i を選択し、選択した動作モード M i の周波数 F i 及び電源電圧 V i を設定する。

【 0 0 8 6 】

次に、ステップ S 5 0 4 では、電力管理コントローラ 6 1 1 は、図 7 の最適周波数導出表 5 2 1 を参照し、選択した動作モード M i の周波数 F i 及び電源電圧 V i におけるプロセッシングエレメント 1 0 1 の変更後電力 P p を読み出す。そして、電力管理コントローラ 6 1 1 は、その変更後電力 P p を基に、上記と同様に、最適稼働プロセッシングエレメント数 N を求める。

【 0 0 8 7 】

仮に、周波数 f が動作モード M i の周波数 f i と動作モード M i + 1 の周波数 f i + 1 の間の周波数だった場合、動作モード M i と動作モード M i + 1 のどちらで動作させた方が性能が出るかは特性によってどちらの場合も考えられるため、事前には決められない。本実施形態では、周波数が低い側の動作モードを選択し、選択した動作モードにおける変更後電力 P p を読み出し、総電力が計算システム 1 0 0 の制約電力を超えないことを保証している。なお、周波数 f より周波数が低い動作モードで動作させるので、変更後の実際の電力は変更後電力 P p より小さくなるため、本来の最適稼働プロセッシングエレメント数 N は変更後電力 P p を用いて求めた稼働プロセッシングエレメント数 N より大きくなる。最適稼働プロセッシングエレメント数よりやや少ない稼働プロセッシングエレメント数で動作することになるため、計算システム 1 0 0 の電力が制約電力を超えることはない。このように、変更後電力 P p を基に稼働プロセッシングエレメント数 N を求めることにより、簡便に動作モード M i 及び稼働プロセッシングエレメント数 N を決定することができる。動作モード M 1 ~ M m の周波数 F 1 ~ F m のステップが十分細かければ、処理を簡略化することによる性能劣化も小さくなる。

【 0 0 8 8 】

次に、他の方法を説明する。例えば、図 7 の最適周波数導出表 5 2 1 の最後列の変更後電力をダイナミック電力とスタティック電力に分けて記録しておく。ここで、上の方法で

10

20

30

40

50

最適周波数を求めるために用いた各動作モードの変更後電力を $P_{p o 1}$ とし、変更後のダイナミック電力を $P_{p d 1}$ とし、変更後のスタティック電力を $P_{p s 1}$ とする。また、目的とする動作モードでのダイナミック電力を $P_{p o 2}$ とし、変更後のダイナミック電力を $P_{p d 2}$ とし、変更後のスタティック電力を $P_{p s 2}$ とする。

【 0 0 8 9 】

この場合、目的とした動作モードでの電力 P_p は、次式により導出する。

$$P_p = P_{p o 1} \times P_{p d 2} / P_{p o 2} + P_{p s 2}$$

【 0 0 9 0 】

電力 P_p は、目的の動作モードにおける変更後のダイナミック電力の変動比 $P_{p d 2} / P_{p o 2}$ を用いて、元のダイナミック電力 $P_{p o 1}$ を補正することにより、前述の簡便な
10
変更後電力の導出よりも、電力の精度を上げて、より無駄の小さい最適稼働プロセッシング
エレメント数 N を求めることができる。また、ダイナミック電力の変換係数 $P_{p d 2} / P_{p o 2}$ の部分を予め求めて表に格納して利用してもよい。

【 0 0 9 1 】

また、前述の簡便な方法では、動作モードは周波数が小さい側の動作モードを選択して
いたが、上記のようにプロセッシングエレメント電力の精度を向上させるため、得られた
周波数 f を挟む両側の動作モード M_i の周波数 F_i を用いて、次式により性能 G_s を計算
し、性能が高い側の動作モードを選択してもよい。

$$G_s = N \times F_i$$

【 0 0 9 2 】

次に、本実施形態の効果の説明をする。計算システム 1 0 0 が並列処理プログラムを実行
した時の並列演算部分の性能特性が図 1 2 のようになっていたとする。稼働プロセッシング
エレメント数 1 2 0 8 は、計算システム 1 0 0 全体の消費電力が規定の制約電力を超え
ない範囲で最大となるように調整する。横軸は、CPU 1 0 2 のクロック周波数を表す。
計算システム性能 1 2 0 1 は、右側の縦軸に表す。稼働プロセッシングエレメント数 1 2
0 8 は、左側の縦軸に表す。計算システム性能 1 2 0 1 は、式 (3) で表されるように、
周波数 f に対して極大点 2 0 3 を一つ持つ曲線になる。稼働プロセッシングエレメント数
1 2 0 8 は、周波数 f が高くなるにつれて、プロセッシングエレメント 1 0 1 単体の消費
電力が上昇するため、減少する特性となる。本実施形態では、計算システム 1 0 0 の全
プロセッシングエレメント数 1 2 0 6 を 1 7 0 0 0 個としているため、稼働プロセッシング
30
エレメント数 1 2 0 8 は、全プロセッシングエレメント数 1 2 0 6 を超えるグラフ左側
では 1 7 0 0 0 個で横ばいとなっており、対応する計算システム性能 1 2 0 1 は、稼働
プロセッシングエレメント数 1 2 0 8 が増やせないため、周波数 f を低くした時の性能が急激
に減っている。

【 0 0 9 3 】

計算システム 1 0 0 の全プロセッシングエレメント 1 0 1 を動作させた場合は、周波数
が 1 . 0 [a . u .] 近辺のところ計算システム制約電力下で周波数最大となる動作点
1 2 0 2 となる。これに対し、本実施形態では、計算システム性能 1 2 0 1 から得られる
最適動作点 1 2 0 3 を予め求めて最適周波数導出表 5 2 1 にテーブル化して利用するため
、計算システム性能 1 2 0 1 は周波数 1 2 0 4 が 1 . 6 [a . u .] 付近の動作点 1 2 0 3
40
となる。本実施形態の動作点 1 2 0 3 の計算システム性能 1 2 0 7 は、全プロセッシング
エレメント動作時の動作点 1 2 0 2 の計算システム性能に対して、8 . 0 [a . u .] から
9 . 5 [a . u .] への約 1 7 % の改善となる。この時、稼働プロセッシングエレメント数
1 2 0 8 は、1 7 0 0 0 個から約 1 2 0 0 0 個程度に減少する。これより、本実施形態に
より計算システム性能が改善されることが分かる。

【 0 0 9 4 】

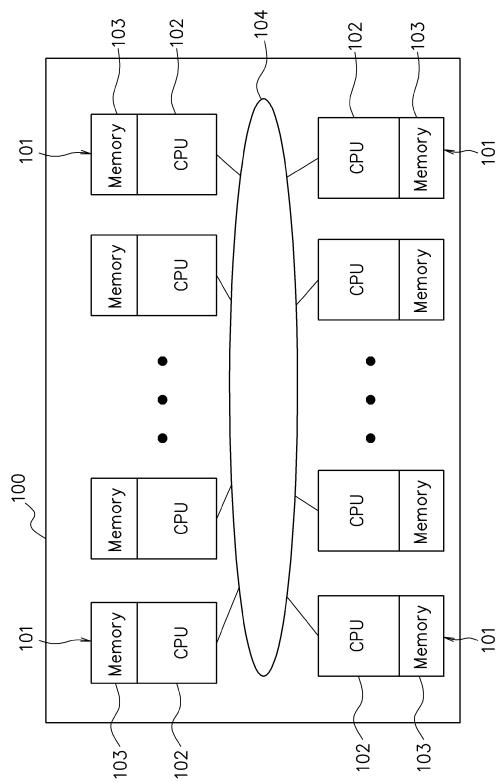
なお、上記実施形態は、何れも本発明を実施するにあたっての具体化の例を示したもの
に過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものであ
る。すなわち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々
50
な形で実施することができる。

【符号の説明】

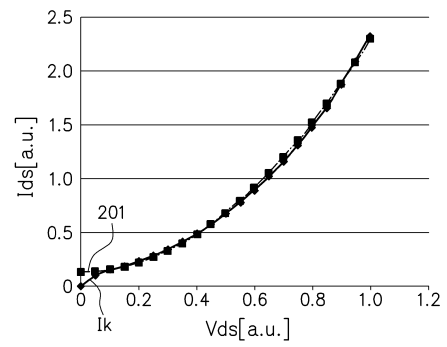
【0095】

- 100 計算システム
- 101 プロセッシングエレメント
- 102 CPU
- 103 ローカルメモリ
- 104 ネットワーク

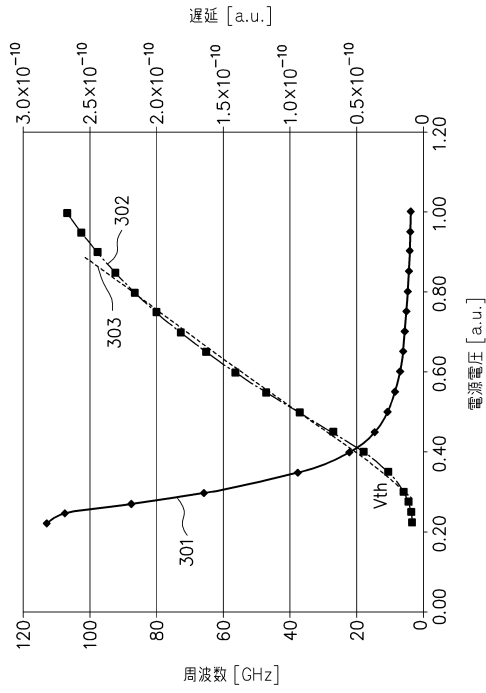
【図1】



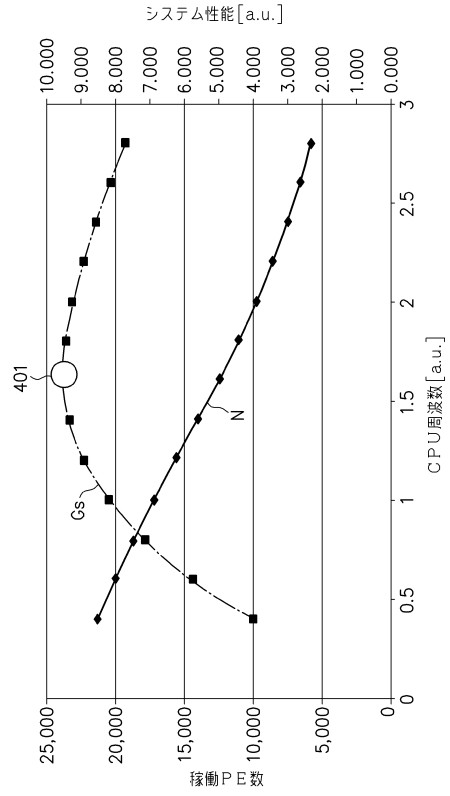
【図2】



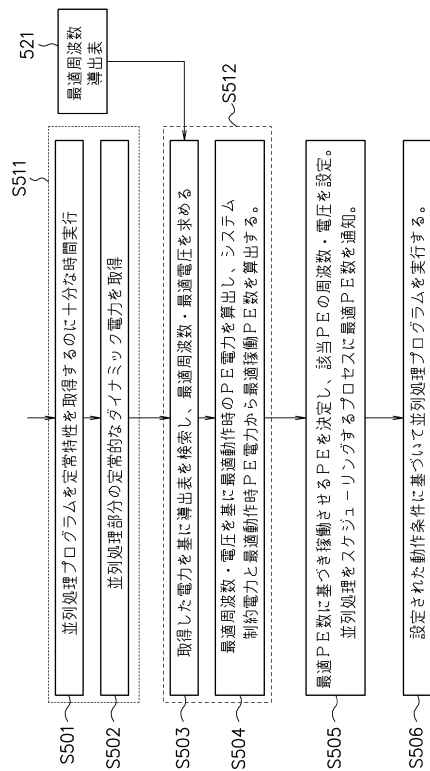
【図3】



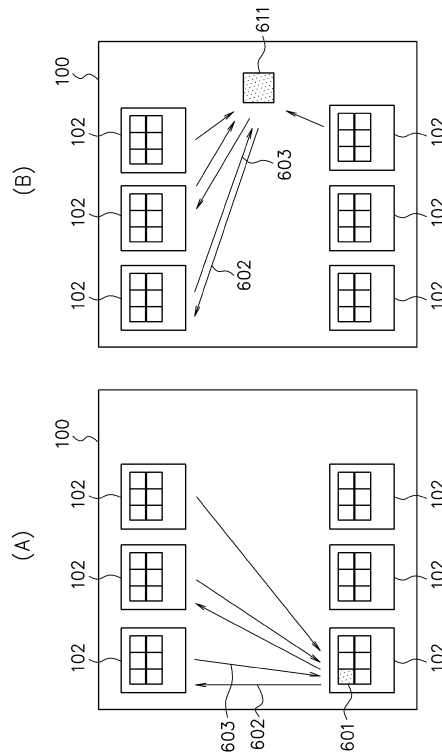
【図4】



【図5】



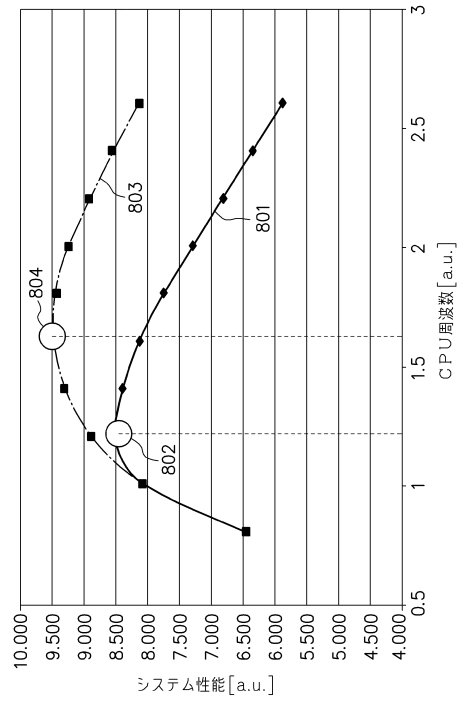
【図6】



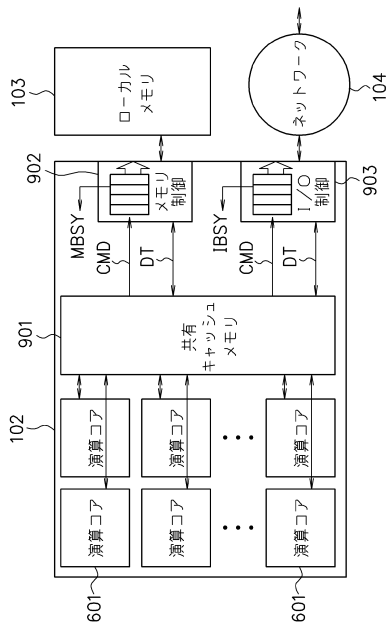
【図7】

	PEダイナミック電力	性能最適周波数	変更後電力
1	P1	F1	P1a
2	P2	F2	P2a
3	P3	F3	P3a
	⋮	⋮	⋮
m	Pm	Fm	Pma

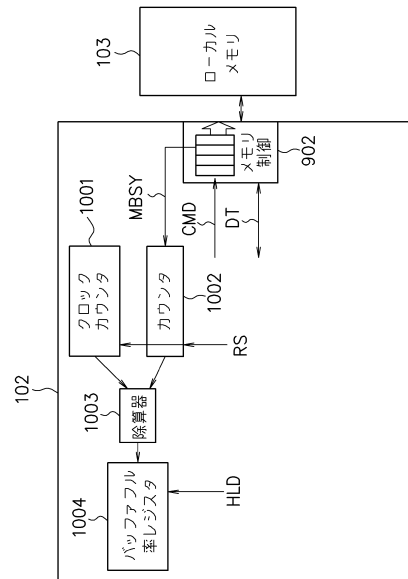
【図8】



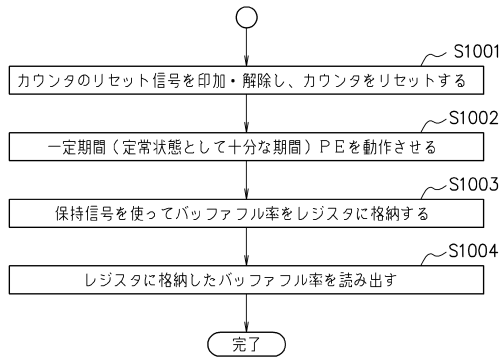
【図9】



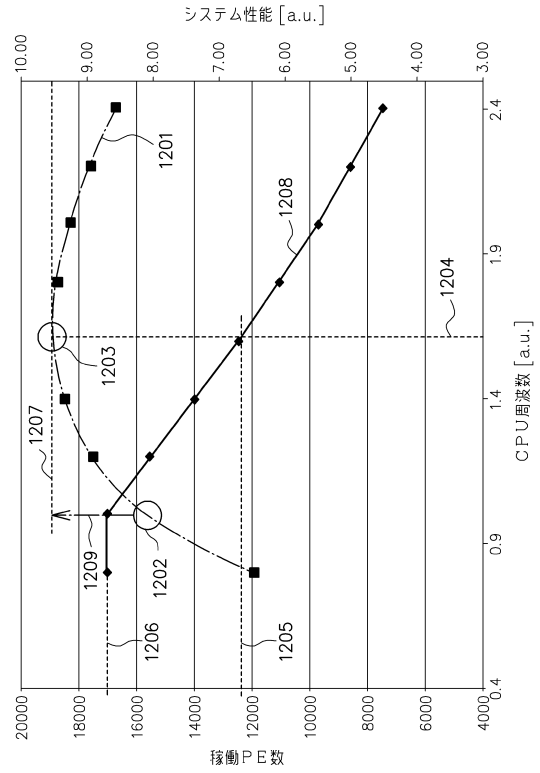
【図10】



【図 1 1】



【図 1 2】



【図 1 3】

	PEダイナミック電力	性能最適周波数 (a.u.)	変更後電力
1
2	36W	1.7	73W
3	40W	1.5	71W
	∷	∷	∷
m-1
m	70W	0.9	...

フロントページの続き

- (56)参考文献 特開2005-085164(JP,A)
国際公開第2012/108058(WO,A1)
国際公開第2012/001776(WO,A1)
特開2006-344162(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/50
G06F 1/32