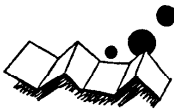


解説



電力系統問題における超高速演算技術 —マルチプロセッサの適用について—†

S. N. Talukdar†† M. Carey††† S. Pyo††

松本吉弘 訳††††

1. ま え が き

マルチプロセッサ技術の進歩が、ある種の問題の求解に役立ち始めている。しかし、どのような問題に対して、どのような型のマルチプロセッサが向いているか、という質問に答えることは難しい。本稿ではまず、電力系統を計画したり、運用したりするとき起こってくる問題の特性を簡単に考察し、つぎに技術的にもっとも本流にのっとったマルチプロセッサのひとつと考えられている Cm* を用いて問題を解いてみたいいくつかの経験、およびその結果にもとづいた実時間制御に対するマルチプロセッサの用法に関する今後の研究方向について述べている。この仕事の一部は Electric Power Research Institute の project RP 1764-3 によって援助された。

2. 電力系統における問題の性質

2.1 目 的

電力系統の目的は、電力という商品を多数の需要家に分配供給することにある。一般に発電設備は少数、大容量で、数箇所にかたまっているのに対して、需要家のほうは大変な数であり、分散し、消費量は小さく、規則性がない。最近、問題となっている負荷管理施策 (Morgan, Talukdar, 1979) の場合を除くと、電力系統は一般に、需要家の欲するときその要求量を完全に満たすことを前提として運用されている。配送

における遅れが許されないにかかわらず、貯蔵能力が小さいので、正味発電量が需要量に追従せねばならない。追従能力の検討に影響を与える需要現象の生起する時間軸上での時間幅は、短いもので数マイクロ秒 (過渡的な電力じょう乱)、長いもので数年 (電力の長期需要計画を考える場合) に及ぶ。このような時間幅の異なる現象のすべてを一緒に扱うわけにいかないもので、計画と運用の問題は階層になっているものとして扱われている。この階層のいくつかのレベルとそのおのおのに対する業務は表-1 に示されているとおりである。

2.2 問題のもついくつかの特徴

電力系統を他の商品の製造販売システムからかけ離れたものにしていく特徴のひとつに、電力網というもののもつ特性がある。電力系統の計画者、運用者の双方によってよく用いられる一連の代数方程式は、電力網を、それがいつも準定状態 (quasi-steady state) にあると仮定して扱っている。すなわち、

$$G(X)=0 \quad (1)$$

表-1 計画および運用業務の例

	おおよその時間幅	業 務
設 備 計 画	20年先	負 荷 予 測
	20年先	発電設備の追加計画
	20年先	燃 料 計 画
	5~15年先	負荷および発電設備の変化に合わせた電力系統拡張
	10~20年先	電力融通計画、租税支払計画
運 用 計 画	2~5年先	定期点検期を計画し、水力発電のように限界のある発電量の計画をなし、融通計画を行う。
	1~5年先	燃料入手および貯蔵計画
突 運 時 間 用	8時間~1週間先	起動停止計画
	常 時	各発電量や発電電圧などの変数値の計画

† High Speed Computation Technique in the Electric Power Systems Problem-Applications of Multi-processors by Sarosh N. Talukdar (Design Research Center, Carnegie-Mellon Univ.), Michael Carey (Berkeley, University of California), Sam Pyo (same with Talukdar) and Y. Matsumoto's translation (Toshiba Corp.).

†† カーネギーメロン大学設計術研究センター

††† カリフォルニア大学バークレー校

†††† 東京芝浦電気(株)

G はベクトル関数であり、 X はネットワーク変数のベクトルである。方程式の数は非常に多く (G は一般に1,000 から 10,000 の次元をもっている)、大変疎 (sparse) であり、並な非線形性をもっている。これに対する解法のほとんどは Newton 法を用いている。すなわち、繰り返しの各回で、つぎの線形方程式を解くことによって、前回の推定値 X_{old} から、解により近い推定値 X_{new} をもとめるのである。

$$J(X_{old})\Delta X = -G(X_{old}) \quad (2)$$

$$X_{new} = X_{old} + \Delta X \quad (3)$$

$J(X)$ は X に関する G のヤコビアンである。このヤコビアンは非常に疎であり、また各行の中に2ないし10個のゼロでない要素がランダムに散在しているような形状をもっている。何年にもわたって、疎マトリックスの処理という分野が大変な関心を集めてきた。その結果、汎用計算機上で実行される潮流計算 (Load Flows) とよばれる効率の良いプログラムが、数多く開発されている (Scott, 1974)。

潮流計算は電力網の定態状態での振舞をシミュレートするものである。これに対して、潮流計算式とこれに並んだ他のいくつかの条件式を満たしながら、たとえば運用コストのような目的関数を最小にするような問題がある。このような問題を最適潮流問題 (Optimum Power Flow—OPF と略称する) とよんでいる。ここで用いる制約式は予期しないで発生する機器の事故に対して、電力系統を安定に保つため、すなわち需要家に電力供給を継続しつつ、なお他の機器に過度な負担をかけないための条件を表わしたものである。このような事故は突然発生し、電力系統構成の変更を招くので、電力系統を安定に保つためには、予想される構成のそれぞれにおいて、系統が安定に作動するように考えておかねばならない。

$$(OPF): \text{Min } f(X^0)$$

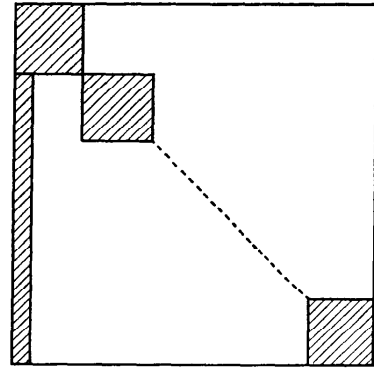
$$\text{s. t. } G^k(X^k) = 0, \forall k \quad (4)$$

$$H^k(X^k) = 0, \forall k \quad (5)$$

$$L^k(X^0, X^k) = 0, \forall k \quad (6)$$

ここで f は目的関数で、添字 k は考える電力系統構成のそれぞれに対応している ($k=0$ は初期構成に対応)。式(4)は k 番目の系統構成における潮流方程式の集まり、式(5)は機器の定格やそれ以外の限界関係を規定する式の集まり、式(6)は k 番目の構成における各変数が、どのように初めの構成における変数値の影響を受けるかを定義する式の集まりである。

式(6)があるために、任意の2つの電力系統構成の



注: $L=[L^0, L^1, \dots, L^k]$, かつ $X=[X^0, X^1, \dots, X^k]^T$
もし、時間に対する微係数を考えると、左側の垂直帯の幅が、左上の箱の幅まで広がる。

図-1 時間に対する微係数を省略した場合のヤコビアン行列の形

各変数の値の間に関係が生じる。もし、動的変化を無視すると、 X (複数) に対する L (複数) のヤコビアンは図-1 に示すような形状となる。この形状は結合の弱さを示している。もし、並列処理の必要があるとすれば、このマトリックスは並列に扱える。実用されている (OPF) プログラムの大部分は、制約式の数を減らすために、初めの電力系統構成のみを考え、さらに他の部分についても簡略化をはかっている。もっとも簡略化された (OPF) が古典的経済負荷配分問題 (Classical Economic Dispatch) とよばれるものである。これは、全発電量が全需要量と電力網内の損失の二次近似の和に等しい、という制約のもとで発電コストを最小にする問題である。

2.3 アルゴリズム

ネットワークモデルから作られたマトリックスを扱う大量の計算を処理するアルゴリズムを LFI (Load Flow Intensive) アルゴリズムとよぶ。前項でも述べたように、このアルゴリズムによる標準的な計算は、大きくて非常に疎で、しかも規則性のない形をした線形代数方程式を対象としている。このような計算を高速に行うためには、高速の計算装置を手に入れるだけでは駄目で、規則性の少ないマバラさを逆に利用した特殊なアルゴリズムを必要とする。LFI アルゴリズムによる計算量は、実時間運用計算では全計算量の50%のオーダを占めるが、実時間に無関係なオフラインの計算では図-2 に示すように、その占める比率は大きくない。

図-2 によると、原子力発電をもつ電力会社では、

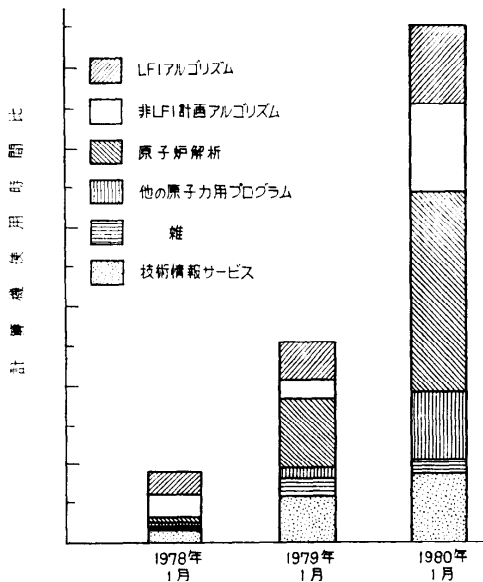


図-2 原子力発電をもつ電力会社における実時間用でない計算のための計算機使用時間集計 (Peczynski, Hyatt, Silva, 1980)

原子炉のシミュレーションにかなりの計算機時間を使っていることが分る。このような計算ではたくさんの偏微分方程式の解を求め。 (Barry, 1963; Fowler, 1971; Rowe, 1970; Moore, Retting, 1973)。ここで用いられている数値計算では、疎ではあるが規則性をもった (たとえば Varga, 1962) 多元の連立線形代数方程式を用いて偏微分方程式を近似する方法が用いられている。規則性がある場合には、CRAY-1 のようなパイプライン計算機または、ベクトル計算機の能力を十分に生かすことができる。また、超 LSI チップをこの目的のために開発することすら計画できる。

電力系統計算で用いられているアルゴリズムを、その計算法の特徴によって分類するという作業はまだ十分に組織化されていない。

2.4 計算機設備と給電指令所

電力系統に関係した計算アルゴリズムは2種類の計算装置の上で実行されている。すなわち、資源、運用計画のような実時間性を必要としない計算は汎用計算機上で実行され、実時間で実行せねばならない計算は給電指令用計算機と称する複合システムの上で実行されている。後者はマルチプロセッサのもつ本来の特徴を生かし、その恩恵を引き出しやすいもののように思われるので、これについて少し考察を加えてみよう。

給電指令用計算機は2つの機能をもっている。すなわち、SCADA と運用計画 (scheduling) である。SCADA (supervisory control and data acquisition) はデータを収集し、操作員がデータと接触をもったり、システムに指令を与えるときに必要とされるマンマシン交信を管理する。現存のシステムでは、遠隔して散在する100以上の局を2~10秒おきに走査している。これらのデータはテレメータで集められ、測定中に起こる大きな誤りや不連続を除き、測定できない電力系統部分の変数を推定するために、汜波、状態推定、汜流計算などのプログラムによって処理される。

操作員は電力系統の種々の部分の現状を知るために、7色のグラフィックスで描かれた1,000枚前後の単線結線図のいずれかを呼び出すことができる。また、操作員はライトペンか、扱いの容易なもので統括指令 (たとえばしゃ断器の開閉指令) を発することができる。

給電指令用計算機の第2の機能である運用計画は、さらに起動停止 (unit commitment) と、負荷配分 (dispatching) とにわかれる。前者は発電機の起動停止を決定し、後者は発電機出力、電圧値、変圧器のタップ位置のような連続、または連続に近い変数の値を決定する。

すべての負荷に供給が行われ、機器の定格や運転方法に重大な変化が起きていないような正常条件下の運用のみを対象にし、古典的経済負荷配分法によって自動的に各発電機の発電計画をもとめることは現実に行われている。いくつかの給電指令システムの場合には、起動停止も自動化されている。しかし、それ以上の運用計画に関する決定は、とくに異常状態において、操作員に任されている。操作員を助けるために、“what if” question に答えるべく多数のシミュレーションおよび最適化のためのプログラムが提供されている。(what if question というのは、もし〜が起きたら、どうするかという質問のこと) これらの給電指令用プログラムは図-3に示すようなマルチプロセッサ構成の上で実行されている。

現状を一口で表わすと、SCADA機能は高度に自動化されますます高度化を続けているが、運用計画のほうはそれほどではない。後者についてはより良い手法や、さらに自動化するための手法 (Talukdar, Wu, 1981) が強く望まれている。すなわち、古典的経済負荷配分法を越えてより一層検討された OPF や、計算機に対してかかる重負荷を処理する強力な解決策が必

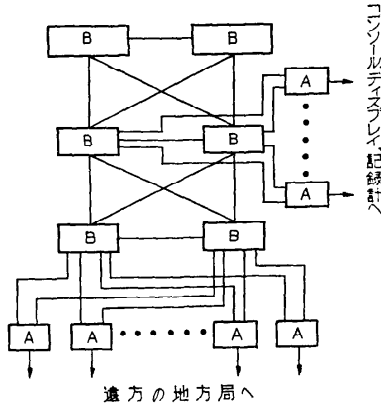


図-3 給電指令に適用したマルチプロセッサの典型的構成

B は CDC CYBER, DEC VAX 11/780, SEL 32 のような計算機で, A はそれに対するフロントエンドミニコンである。B は 2 台を組にし片方をバックアップ用として信頼性を上げる。プロセッサ間はデータリンクで結ぶ。共有メモリで結ぶ場合もある。異プロセッサにまたがる並行プロセスはない。

要とされている。給電指令用計算機の詳細については文献 (Talukdar, Wu, 1981), (IEEE, 1977) を参照されたい。

3. 技術の現状——短評

電力系統に適用される電子計算機の用法に関する研究には 2 つの流れがある。

第 1 の流れは既存のベクトル計算機やアレイプロセッサが果して LFI アルゴリズムの高速処理に、はっきり分るほど有効かという問題を扱ったものである (Calahan, 1977; Dembart, 1977; EPRI, 1978; Orem 1978; Podmore, Liveright, Virmani, 1978)。少なくとも現時点以前には注目に値する結果は生れていない。CRAY-1 のような大型のベクトル計算機では、その能力全部を生かしきれない。電力系統で使われるマトリックス計算で現われる程度のベクトル長では、パイプライン機能の効果はあまり有効ではない。まったく役に立たないわけではないが、AP-120 B のようなもっと小型で安い外付け型アレイプロセッサの効果に比してあまり変わりばえない (表-2 を参照されたい)。

一方、アレイプロセッサはスカラ演算を単に高速化するだけで上の目的を果さんとし、小型、安価を特徴とするために、汎用科学用計算機 (たとえば CRAY-1) に較べると、ソフトウェア支援が一般に弱い。そのために電力系統のマトリックスを扱うのに必要な特殊な疎マトリックス解法をプログラムで記述するとき

表-2 計算装置の比較

	バースト能力 (MFLOPS)*	潮流計算標準問題の計算能力 (MFLOPS)*	概算価格 (百万ドル)
AP-120 B	12	4.0	0.05-0.15
VAX 11/780	?	0.4	0.2
IBM 370/168	?	1.0	3
CDC STAR-100	25	0.7	4-8
CRAY-1	160	4.0	5-8

* MFLOPS: Millions of Floating Point Operations Per Second.

には、プログラム作成が難しくならざるを得ない。

第 2 の流れはマルチプロセッサが電力系統アルゴリズムを高速で処理するのに使えるか? というテーマに関するものである (Carey, 1981; Durham and colleagues, 1978; Van Ness and colleagues, 1981; Wallach, 1977)。研究の焦点は疎マトリックスのレベルで分割された LFI アルゴリズムに割り当てられた多数の同形式のプロセッサをいかに配列するかという点におかれている。換言すれば、同形式のプロセッサを多数用いることによって疎マトリックス処理が高速化されるのか? という問題が集中的に扱われている。今までの研究結果からは、はかばかしい答が得られていない。次の章ではこの原因を考察し、マルチプロセッサがより有益に利用できる方向の指摘を試みよう。

4. Cm* 形マルチプロセッサ

マルチプロセッサの一般形を図-4 に示す。しかし、話を具体的なものにするために、Cm* を用いて説明を進める。

Cm* はカーネギメロン大学 (Fuller and colleagues,

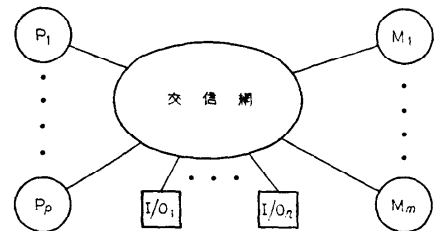


図-4 マルチプロセッサの一般形

プロセッサやメモリは同一である必要はない。各プロセッサはそれぞれ独立にプログラムされ、ひとつ以上のメモリへアクセスしてもよい。

P_i = プロセッサ i

M_i = メモリ i

I/O_i = I/O ユニット i

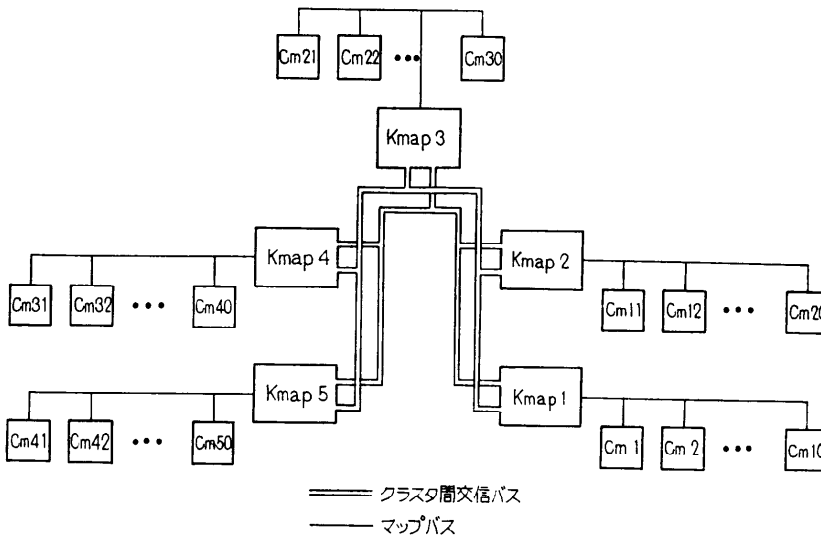


図-5 Cm* の構造 (Ousterhout, 1980)

1978) で開発された複数マイクロコンピュータシステムである。Cm* の設計、製作は超 LSI 技術をどのようにしてアーキテクチャの中で生かすかを研究するために着手されたものである。Cm* の設計が初めに目標としたものは、複数の密結合プロセス (Jones and colleagues, 1978) から成る大きな仕事に役立てること、モジュール化され、拡張可能なアーキテクチャを提供すること、およびプロセッサ間通信の種々の形式の実験に役立てることであった。数百のプロセッサ要素を組み込めるようになっており、市販のハードウェアが可能な限り利用された。

図-5 に示す Cm* のアーキテクチャは 50 の計算モジュール (Cm) からできている。ひとつの Cm は DEC LSI-11 マイクロコンピュータで、図-6 のように 64 か 128K の主メモリと、Slocal と称するメモリスイッチをもっている。Slocal の働きによって、Cm* 内のすべての Cm は他のすべての Cm のメモリへアクセスすることができる。ただし、自分のメモリと他のメ

モリとの区別については自分で責任をもたねばならない、この目的のために、16 個の記憶場所から成るページマッピングテーブルがあり、アドレッシング上位 4 ビットでこの記憶場所のひとつを選定する。他のメモリが指定されたときは、Slocal は Kmap と称するマイクロプログラムで制御されるスイッチ用プロセッサを働かせる (マッピングテーブルのひとつの記憶場所に入っているデータは、目的のプロセッサおよびその主メモリ内の目的とする 4k バイトページの先頭アドレスを示し、上に述べた 4 ビット以外の残り 12 ビットはこのページ内のアドレスを与える)。Slocal は他のプロセッサから Kmap, map バスを經由して到着する要求をも処理する。

Cm* は 5 組のクラスタから構成され、それぞれのクラスタは Kmap でくくられている。Kmap は自分のクラスタ内の Slocal からの要求を受付けるが、この要求には同一クラスタ内の Cm のメモリに対するものと、他のクラスタ内の Cm メモリに対するものの 2 通りがある。Cm が自分のメモリを参照するのに要する時間は約 3 マイクロ秒、クラスタ内の他のメモリを参照するに要する時間は約 9 マイクロ秒、他のクラスタ内のメモリを参照するに要する時間は約 26 マイクロ秒である (Raskin, 1978)。

Cm* は Slocal と Kmap のおかげでアーキテクチャの自由度が広がり、非常に多数のプロセッサを組み込むことができる。Kmap はマイクロプログラムで制

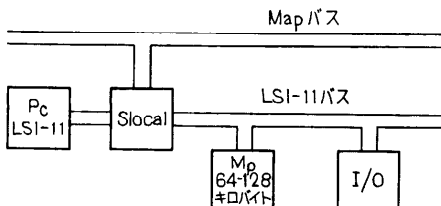


図-6 Cm の詳細 (Ousterhout, 1980)

御されるので、セマフォ、メッセージ交換式のプロセッサ間通信、その他頻繁に用いられる主要機能（たとえば、容量をベースにしたアドレッシング方式やメモリ保護）をマイクロプログラムで記述することによって、オペレーティングシステムの効率を上げることができる。

並行して走るプログラムを記述しやすくするため、 Cm^* はプロセッサ要素を多数もっていること、プロセッサ間交信命令をファームウェアで実現していること、メモリの動作を階層化していること、などの際立った特長をもっている。初めの2つは独立性の高い並行プログラムの設計、実現にとって有効である。最後のひとつについて次章でやや詳しく述べる。

5. プログラムの並列化について

はじめにプログラムの並列化にからむ問題を一般的に論じ、つぎに電力系統に特有の問題について述べる。さらに Cm^* について行った実測について述べる。

5.1 アルゴリズム上の問題

マルチプロセッサの効果を上げるためには、プロセッサとメモリ間の接続をより複雑にせねばならないようなデータ流れを相互間にもたない並行タスクの存在比率を、個々の手続き単位内で大きくせねばならない。

もし、並行タスクがそのままでは存在しない場合には、分割によってこれを作り出さねばならない。たとえば電力網について考えるならば、ある節点は切り離すとか、ある支線は省略するとかして、初めの電力網を独立したいくつかの部分に細分する必要がある。個々の部分はそれぞれ独立に処理し、結果を相互に関連させる。電力網全体の振舞は、個々の部分の結果を編成して推定される。

このような手順を分割/再編成 (decomposition/reconstitution) とよび、数多くの方法が考えられている。しかし、どの方法でもそのまま計算するのに較べて余分な処理が必要となるので、プロセッサの台数に比例して計算速度を上げることは不可能だとされている。詳細は文献 (Talukdar, 1979) を参照されたい。この余分な処理のことを分割のためのペナルティ (Decomposition Penalty) とよんでいる。

5.2 実現および構成上の問題

マルチプロセッサのアーキテクチャに起因した固有のオーバーヘッドやプロセッサにタスクを割り付けた

めに発生するオーバーヘッドによるペナルティがさらに加わる。これを実現するためのペナルティ (Implementation Penalties) とよぶ。これに対してつぎの2つの要素が重要な役割を果たす。第1の要素はアルゴリズム上での並行プロセス間の結合度 (degree of coupling) である。結合度は、実行中に共通記憶域、メッセージなどを介してプロセス間で行われる交信量に相当する。プロセス間が密に結合されればされるほど、リソース取合いの機会がふえ、性能が低下する。第2の要素はコードとデータの計算機リソースに対する割り付けである。これら両要素はどちらもリソース取合いに関係するが、前者はアルゴリズムによって先天的に与えられているのに対して、後者はそうでないという基本的な差異がある。もちろん、両要素のそれぞれが性能に与える影響度となると、個々のハードウェアやシステムソフトウェアが関係してくる。実現のためのペナルティについては文献 (Carey, 1981) に詳述されている。

5.3 例題

1端に電圧が加えられ、他端に非線形抵抗が接続されているような1本の送電線を考えてみよう。この送電線に対する偏微分方程式はいくつかのマクロモデルに分割でき、それぞれを別々に解くことができる (Talukdar, 1976)。個々の解を編成 (reconstitute) して、この送電線に対する解を求める。この方法は図-7に示すような2段アルゴリズム (two phase algorithm) のひとつと見てよい。

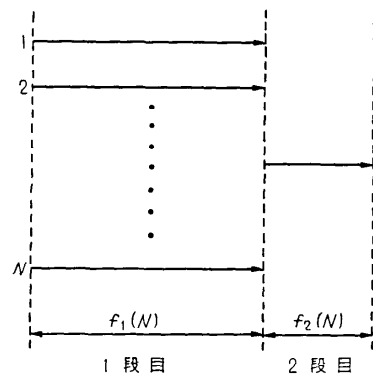


図-7 2段アルゴリズム

1段目は N 個の並行タスクに分けることができる。各タスクの実行時間を $f_1(N)$ とする。2段目は $f_2(N)$ の実行時間を必要とする1タスクより成る。 N が大きくなると $f_1(N)$ が減少し、 $f_2(N)$ は増加する。分割のためのペナルティは $[Nf_1(N) + f_2(N)] - [f_1(1) + f_2(1)]$ である。

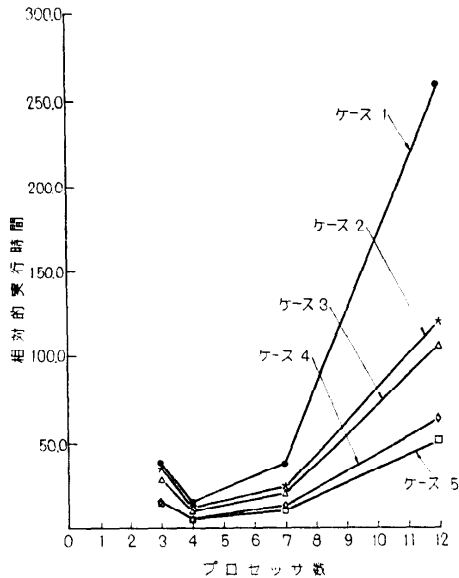


図-8 Cm* で例題を実行した結果

このアルゴリズムを Cm* 上で走らせるためにいくつかの実験が試みられた (Carey, 1981)。結果の一部を図-8 に示す。第1のケースではすべてのプロセッサに対して共通の場所におかれたただひとつのプロセスコードを、すべてのプロセッサが実行している。第2のケースでは、プロセスコードの原形は共通の場所におかれているが、各プロセッサがその写しをそれぞれもって実行に用いている。第3のケースではデータだけが共通の場所に置かれている。第4のケースではデータ、手続きコードとも各プロセッサが独自にもち、数メモリサイクルごとに同期をとりながら実行している。この同期間隔を30メモリサイクルにしたのが、第5のケースである。

分割によるペナルティはたちまち (プロセッサ数4の点で) プロセッサ数を増加することによる恩恵を帳消しにし、その後も急激に増加していることが分るだろう。

各ケースの曲線間の垂直方向の隔りは実現のためのペナルティを意味している。Cm* のメモリ階層のもとでは、コードやデータを共通しておくことは大変不利である。同期のための操作間隔を少し広くとったほうが、結果的に計算速度を上げられるというのは興味のある事実である。これはプロセス間の結合、ひいてはメモリ競合の大きさを減らすことがいかに有利であるかを示している。

6. むすび

大規模で念の入ったマルチプロセッサを作り上げることはいくらかでも可能である。問題はそれが電力系統問題に役立つかどうかである。この問題はいくつかの副問題を含んでいる。

- (1) 電力系統機能のうちのいずれがもっともマルチプロセッシングの恩恵に浴するか?
- (2) どのようなアルゴリズムを用いるべきか?
- (3) どのようなアーキテクチャがよいか?
- (4) アルゴリズムをどのようにアーキテクチャと組み合わせるべきか?

実時間運用システムはマルチプロセッサに適合していると筆者らは考えている。運用中の意志決定を人間が行わず、計算機に委ねたいという根強い要求がある。最適潮流問題 OPF は自動意志決定の基礎的推進力であり続けるだろう (知識ベースシステム Knowledge Based Systems (Pao, Schultz, Oh, Fischer, 1981), (Michie, 1979) などはその挑戦例であるが、まだまだ先が長い)。現在用いられている古典的経済負荷配分システムを格上げしてより多くの決定を扱うようにし、今以上に安定度問題を含め、異常状態の処理も含めようとする、大変な計算量の増加を必要とする。実時間の環境下では、運用計画の決定は長くとも5分以内の間隔で行わなければならない。したがって高速はなににもまして重要である。計算を複雑にする原因の大半は、故障の発生の可能性と、それによって生ずる種々の系統構成を考慮に入れることにある。これに対するモデルは図-1 に示すように、本質的に並列な形をしているのだから、高速化のためにマルチプロセッサは適していると思える。

さて、(2)~(4)の問題について考えてみると、多数の案があるのに気づく。ひとつの機能をとり上げてみても、これにはたくさんのアルゴリズムが使えるし、そのアルゴリズムを実行できるアーキテクチャも多く考えられ、ひとつのアルゴリズムを実現する方法もたくさんある。これらにとり組むときはつきに示す順序に従うことを提言する。

- (a) アルゴリズムを選ぶこと: 分割された最小要素を高速に処理し (最小数のプロセッサを必要とする要素のこと——最小要素は必ずしも1プロセッサのみで走るものと考えなくてよい)、最小の分割のためのペナルティをもち、最小の実現のためのペナルティをもつと思われるアルゴリズムを選ぶ必要がある。今ま

で考えられたアルゴリズムはほとんど同期型であった。このようなアルゴリズムでは分割、実現ペナルティは両方とも大きい、最小要素に対する処理速度の向上が得られる。これに対して非同期型アルゴリズム (Chazen, Miranker, 1969), (Baudet, 1976) では上と全く逆となる。このへんは今後の研究の余地がある。

(b) アーキテクチャを形成するプロセッサ, メモリ, 接続装置の型を選ぶこと: (いままでの研究の多くはプロセッサの型を固定して行われてきたが) ひとつの型のプロセッサに固執してはならない。電力系統用アルゴリズムのためのタスクの型の多様性は大きい。ひとつの型に固執すると, 例外なくボトルネックが起き, 速度向上があまり得られない (Dugan, Durham, Talukdar, 1979), (Van Ness, 1981)。たとえば, アレイプロセッサをプロセッサの一員に加え, 疎マトリックス処理だけはこれに任せるようにするのが良い。

(c) 分割のレベル, すなわちタスクの大きさを適正に選ぶこと: たとえばマトリックスの要素分け (factorization) ルーチンくらいの規模をひとつのタスクにすることが望ましい。

(d) アーキテクチャを予備的に設計して, 各タスクにプロセッサ, メモリなどのリソースを割り付け, これでアルゴリズムを実現する方法について検討すること: これは整数計画法の問題のひとつである。ブランチアンバウンド法を用いて解くのは高価につき過ぎる。簡易な発見的手法 (Mehrotra, 1981) を直接的な動作モデル (Grosser, Talukdar, to appear) と組み合わせたとような方法が的を得ていると考える。

訳者注: 筆者らは電力系統解析, とくに実時間制御問題に長くとり組んでいる中堅の研究者である。

本特集の編集方針に従って, 訳者からはとくに“電力系統問題へのスーパーコンピュータの適用”として執筆を依頼したのに対して, 論述内容は“マルチプロセッサの適用”となっている。その理由は本文の中に述べられているとおりである。

参 考 文 献

- 1) Barry, R. F.: LEOPARD, a spectrum dependent non-spatial depletion code for the IBM-7094, WCAP-3741 (1964).
- 2) Baudet, G. M.: Asynchronous iterative methods for multiprocessors, Technical Report, Carnegie-Mellon University (1976).
- 3) Calahan, D. A.: Vectorized solution of load flow problems in exploring applications of parallel processing to power system analysis problems, EPRI EL-566-SR, pp. 113-127 (1977).
- 4) Carey, M.: Parallel processing for power system transient simulation, M. S. Thesis, Carnegie-Mellon University (1981).
- 5) Chazan, D., Miranker, W.: Chaotic relaxation, Linear Algebra and Its Applications, Vol. 2 (1969).
- 6) Dembart, B. and Neves, K. W.: Sparse triangular factorization on vector computers, EPRI EL-566-SR, pp. 57-101 (1977).
- 7) Dugan, R. C., Durham, I. and Talukdar, S. N.: An Algorithm for power system simulation by parallel processing, IEEE PES Summer Meeting (1979).
- 8) Durham, I., Dugan, R. C., Jones, A. K. and Talukdar, S. N.: Power system simulation on a multiprocessor, IEEE PES Summer Meeting (1979).
- 9) EPRI: Technology assessment study of near term computer capabilities and their impact on power flow and stability simulation programs, EPRI EL-946 (1978).
- 10) EPRI: Parallel processing for power system planning and operations, EPRI Workshop, Dallas, Texas (Oct. 21-23, 1980).
- 11) Fowler, T. B. and colleagues: Nuclear reactor core analysis code, CITATION, ORNL-TM-2496, Rev. 2 (1971).
- 12) Fuller, S. H., Ousterhout, J. K., Raskin, L., Rubinfeld, P., Sindhu, P. S. and Swan, R. J.: Multi-microprocessors: an overview and working example, Proceedings of the IEEE, 66, pp. 216-228 (1978).
- 13) Grosser, J. and Talukdar, S. N. (to appear): Models for MIMD machines, IEEE Trans. on PAS.
- 14) IEEE Tutorial Course: Energy control center design, 77 TV 0010-9-PWR (1977).
- 15) Jones, A. K., Chansler, R. J. Jr., Durham, I., Feiler, P., Scelza, D. A., Schwans, K. and Vegdahl, S. R.: Programming issues raised by a multiprocessor, Proceedings of the IEEE, 66, pp. 229-237 (1978).
- 16) Mehrotra, R.: A one-matching heuristic for processor assignments, Technical Report, Carnegie-Mellon University (1981).
- 17) Michie, D. (Ed.): Expert Systems in the Microelectronic Age, University of Edinburgh (1979).
- 18) Moore, K. V. and Retting, W. H.: RELAP 4-a computer program for transient thermal hydraulic analysis, Aerojet General Co. Report ANCR-1127 (1973).

- 19) Morgan, M. G. and Talukdar, S. N.: Electric load management: some technical, economic, regulatory and social issues, Proceedings of the IEEE, pp. 241-312 (1979).
- 20) Orem, F. M. and Tinney, W. F.: Evaluation of an array processor for power system applications, PICA-79 Conference Proceedings, pp. 345-350 (1979).
- 21) Ousterhout, J. K.: Partitioning and cooperation in a distributed multiprocessor operating system: medusa. Ph. D. Thesis, Carnegie-Mellon University (1980).
- 22) Pao, Y. H., Schultz, W. L., Oh, S. Y. and Fischer, R. G.: knowledge base engineering approach to power systems monitoring and control, IEEE Trans. on PAS (1981).
- 23) Peczynski, E. F., Hyatt, L. A. and Silva, R. F.: Application of array processors: one utility's view, Parallel Processing for Power System Planning and Operations, EPRI Workshop (Oct. 21-23, 1980).
- 24) Podmore, R., Liveright, M. and Virmani, S.: Application of an array processor for power system network computations, PICA-79 Conference Proceedings, pp. 325-331 (1979).
- 25) Raskin, L.: Performance evaluation of multiple processor systems, Ph. D. Thesis, Carnegie-Mellon University (1978).
- 26) Rowe, D. S.: COBRA-II, a digital computer program for thermal-hydraulic subchannel analysis of rod-bundle nuclear fuel elements, BNWL-1229 (1970).
- 27) Scott, B.: Review of load flow calculation methods, Proceedings of the IEEE, Vol. 62, pp. 16-29 (1974).
- 28) Talukdar, S. N.: METAP: a modular and expandable program for simulating power system transients, IEEE Trans. on PAS, 95, p. 6 (1976).
- 29) Talukdar, S. N.: On using MIMD-type multiprocessors-some performance bounds, metrics and algorithmic issues, Proceedings of the 10th Pittsburgh Modeling and Simulation Conference (1979).
- 30) Talukdar, S. N. and Wu, F.: Computer aided dispatch for power systems, Proceedings of the IEEE (1981).
- 31) Van Ness, J. E., Brasch, F. M. Jr. and Kang, S. C.: Design of multiprocessor structures for simulation of power system dynamics, EPRI EL-1756 Final Report of Project RP 1335-1 (1981).
- 32) Varga, R. S.: Matrix Iterative Analysis, Prentice-Hall (1962).
- 33) Wallach, Y.: MIMD-type parallel processing for electric power control, EPRI EL-566-SR, pp. 287-308 (1977).

(昭和56年7月14日受付)