並列処理シンポジウム JSPP '89

並列処理への期待と課題

司会　田中 英彦

パネラー　亀田 壽夫　斎藤 信男
名取 亮　野崎 昭弘
村岡 洋一

田中　このシンポジウムは理論からプログラミング，オペレーティング・システム，アーキテクチャ，アプリケーションを縦断的に見，縦断的なディスカッションをやろうという目的で企画されました．したがいまして，研究会もいろいろな研究会の合同ということで進めております．それをまずご紹介したいと思います．パネラーとしておすわりですが，一番左側から順に，アルゴリズム研究会主査の野崎先生(国際基督教大学)，オペレーティング・システム研究会主査の亀田先生(電気通信大学)，電子情報通信学会のコンピュータ・システム研究会委員長の村岡先生(早稲田大学)，数値解析研究会主査の名取先生(筑波大学)，プログラミング言語研究会主査の斎藤先生（慶應義塾大学）です．それから私は計算機アーキテクチャ研究会の主査をしております東京大学の田中です．ということで，この六つの研究会が集まってやっているわけです．五つが情報処理学会で，コンピュータ・システム研究会だけが電子情報通信学会です．

それでは先ほどご紹介した順にお話いただきたいと思います．テーマは，実はオープニングということで自由となっております．

それでは野崎先生，お願いいたします．

よい対象の選択と上手な設計

野崎　並列処理シンポジウムということで，これだけ大勢の方がお集まりになったのでびっくりしました．並列処理そのものは随分昔から関心が持たれていました．私も昔たとえばセルラー・オートマトンであるとか，いまでいうシストリック・アレイみたいなものをいろいろ考えたのですが，何しろ日本が貧しかったので，そんな物を実際に作るということはできなかったのです．ですから非常に理論的な，実用からはかけ離れたことをやっていたのをいま懐しく思い出しています．いまは本当に具体的な物に結びつけられて，役に立つ研究ができるようになりましたから，それでこれだけ関心のある方々にお集まりいただけるようになったのだろうと思います．

ところで私の場合，人間の頭脳は明らかに並列処理をやっていろいろな効率を上げている，ということが関心の発端でした．ただ，その並列性は必ずしも効率向上のためだけではなくて，たとえば安全性の向上とかの役割も果たしていますし，人間の脳の効率が高いことが必ずしも並列処理のお蔭だけではなくて，それ以外のファクターもいろいろ入っているということに，気をつけなければいけないと思っていました．

たとえば図形認識などですと，明らかに並列処理が大きな役割を果たしているけれども，機能的な処理，言語のかかわる処理になりますと，かなりシーケンシャルな部分も入ってきます．それと同時に人間の場合には「答えを必ずしも100パーセント絶対的に正確に出さなくてもよろしい」という許容度があるのです．そういうところが非常に役に立っていると思います．それからデータの，リオーガニゼーション（という言葉が適当かどうかはわかりませんけれども，）がどうも脳の中で非常に上手に行なわれているらしい．人間の記憶のシステムは，

このパネル討論会は，1989年2月2日～4日に国民宿舎熱海ビレッヂで催された「並列処理シンポジウム JSPP (Joint Symposium on Parallel Processing) '89」の1日目に行なわれたものである．

田中英彦（東京大学）

野崎昭弘（国際基督教大学）

まだよくわかっていないわけですが，記憶内容の組織化や再組織化が情報処理の能率向上に役立っているのではないか，と思います．

これからの研究でも，並列処理を応用すれば必ず有効であるとは限らないと思います．ですから非常に並列処理がぴったりするよい対象を選ぶことが，一つの大きな面白いポイントになろうかと思います．それから並列処理を含む全システムをどのように上手に設計するか，ということも非常に大事な問題だと思います．

ついでに個人的に興味を持っていることを申します．データ処理のほうで何か並列処理が役に立つことがいろいろあるのではないか．人間のやっている，さっきちょっと申しましたデータのリオーガニゼーションなどということを何かうまくできないか．ガーベッジ・コレクションみたいなものを含めてですけれども，いままでは割合バックグラウンド・ジョブみたいな感じで処理していたと思うのですが，それをパラレルでうまく処理できたら非常に能率アップするのではないか，ということをぼんやり考えております．ただデータのリオーガニゼーションというと，言葉はいいのですけれども，抽象的で，そこをどう具体的にすればいいのかということは，私もまだアイディアを持っておりません．

VLSI アルゴリズムについて触れますと，私の所属するアルゴリズム研究会でもいろいろな発表が出るのですが，行列の乗算であるとか，コンプレックスの低いところでの能率向上というのがまだ多いようです．それをコンプレックスのもう少し高い問題に使えないだろうか．たとえば偏微分方程式の境界値問題，ネットワークの問題，たとえば平面グラフ上の最短経路とかには本当に並列アルゴリズムが有効に働くのではないか，という気がしております．そんなところを取り上げてやっていければと個人的には思っております．

並列処理のむずかしさは多くの研究問題の源

亀田　OHP を使いながらお話させていただきたいと思います．

田中先生のほうから最初に「並列処理の期待と課題」という題をいただいていますが，私のほうは野崎先生のお話と違いまして，高水準でないお話になるかと思います．私は先ほどのご紹介にありましたように情報処理学会のオペレーティング・システム研究会を世話させていただいております．

まずオペレーティング・システム研究会について少し紹介させていただきたいと思います．15年ぐらい前になりますが1974年に実際の提案の示唆はここにいらっしゃいます萩原宏先生からいただきまして，その活動が始まりました．最初は「システム性能評価」という研究会名で，主査は大野豊先生，その次が石田晴久先生でした．それから「計算システムの解析と制御」となり高橋延匡先生，益田隆司先生，その後「オペレーティング・システム」になりまして私が主査を務めています．

取り扱う研究分野は，学会の年度の初めの究研会登録案内のところにも出ていますが，「オペレーティング・システムの基礎・構造論」「人間工学的側面」「プログラミング環境」，そして「分散・並列処理」です．これは，このシンポジウムにも直接関係があると思います．それから「システム性能評価」，これは最初「性能評価研究

会」から発足したということで，一つの大きな軸になっております．それから「信頼性」と「関連するコンピュータ・システムの諸問題」というように非常に広く扱うようにしています．

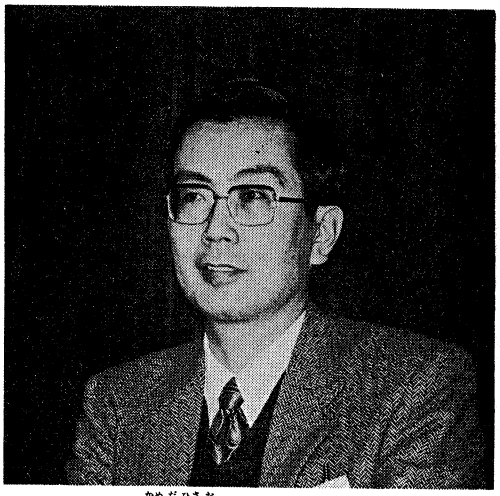
コンピュータ・システムにおいてオペレーティング・システムが重要な役割を果たしております．大ざっぱにいうとプログラムという論理的な手続きをハードウェアにのせて一つの意味のある仕事をさせる，というのがコンピュータ・システムというものの考え方だと思います．その場合に，プログラムをロードし，実行する．そういうサイクルが必ず繰り返されていろいろな仕事がなされると考えられます．そのサイクルの管理をするのがオペレーティング・システムの仕事であると思われます．

これまでのオペレーティング・システムは，フォン・ノイマン型のプログラム記憶式の計算機というものに即して発展してきたわけです．しかし，外から見た働きが変わらなければ，ハードウェア自体が変わっても差し支えないということで，並列計算機が使われるようになる可能性があるわけです．その場合にも，今までのオペレーティング・システムのいろいろな概念のうち，当然並列計算機という新しい方式にも移行しうるものがいろいろとあるはずです．また並列処理の目的は高速化にもあるはずですので性能評価も重要だと思われます．その辺の話題が今度のシンポジウムでもいろいろあると期待しております．

並列処理に並べて比較される言葉に逐次処理があります．もう一つ分散処理という言葉もありますが，それに対して集中処理という言葉が対比させられています．しかし，並列処理とか分散処理などを実現するのは逐次処理や集中処理に比べてはるかに難しいことは明らかです．

SIGPLAN/SIGOPSのワークショップに「パラレル・アンド・ディストリビューテッド・デバッキング」というのがありましたが，その言葉からも，大変であることが示唆されます．ところが難しいということは逆に，チャレンジするような問題をいろいろ提供してくれる，ということでもあります．

今回のシンポジウムはいくつかの研究会の共催ということになっておりますが，実質はコンピュータ・アーキテクチャ研究会の主査の田中先生とか，電総研の弓場敏嗣さんなどをはじめとする方々のご努力によるものであるわけです．その方々の熱意というものが，いろいろと接触したところで非常に強く感じられました．実行委員会の熱意と積極性というのはすごいものでした．田中先生からお話がありましたように論文投稿がほかのシンポジウムに比べて多く，また参加者が，これだけ自発的に集まられるということで，並列処理に対する関心の高さを感じました．



亀田壽夫（電気通信大学）

トータルシステム・パフォーマンスの向上を目ざして

村岡　コンピュータ・システム研究会（電子情報通信学会）の村岡です．いま亀田先生のほうから研究会の宣伝がありましたので，私も少しだけ宣伝させていただきます．コンピュータ・システム研究会というのは非常に惨めな立場でして，面白いトピックがありますと，次から次へと新しい研究会（電子情報通信学会では第2種研究会などと呼んでいるのですが）ができる仕組みになっています．たとえば田中先生が「データフロー・コンピュータ」研究会をやられましたし，今年は，甘利俊一先生が「ニューロ・コンピューティング」研究会をお作りになります．面白そうだから，こういうトピックをこの研究会でもやろうかなと思うと，新しい研究会ができて，そちらに吸いとられる宿命になっております．

そういうことで，何か新しい，これは誰にも取られそうもないというトピックがありましたら，大いに歓迎いたしますので，是非教えていただければありがたいと思います．

並列処理ということでいろいろ話題がありますが，明るい話題，日の当たるほうの話題は，これから3日間いろいろ議論されると思いますし，先ほどからパネラーの方もいろいろお話されていますので，私はコンピュー

タ・システム研究会にふさわしく，地味な話を少ししてみたいと思います．

ご存じの方もいらっしゃると思いますが，アメリカのアーゴン研究所のドンガラがいろいろな並列計算機の性能を LINPACK の中の代表的なプログラムをのせて測っています．個々の計算機の性能をいろいろとご存じの方はわかると思いますが，いわゆるメーカがいっているトップスピードというのはなかなか出ないものです．それからA社の計算機とB社の計算機で，カタログ性能で見るとA社のほうがいいけれども，実際に同じプログラムを掛けてみるとB社のほうがよくなってしまったというようなことがあります．そのような観点から，並列処理をコンピュータ・システムとして見た場合にどういうことを考えなければいけないかを少しお話してみたいと思います．

一番の話題である並列処理のソフトウェアの話は米澤先生のパネル（p. 44の記事）で話さなければなりませんので，その分は明日にとっておかせていただきます．並列処理のコンピュータで次に考えなければいけない話に，メモリ・マネージメントがございます．スーパーコンピュータはたいがいの場合，レジスタ・ファイル，私用メモリ，そして共用メモリの階層になっています．その間のデータのステージングをどういう契機で行なうのか，もしくは私用メモリと共用メモリの間でのデータの処理をどうやるのかを考えてやりませんと，途端に性能が悪くなるというのはよく知られていることです．

昨年の **bit**（Vol. 20, No. 12, pp. 48-50）に，日本IBM の鈴木則久さんが「研究者のテイスト」の最後のほうで「並列処理ではメモリのステージングが大事である」という話を書いていらっしゃいました．私もまったく同感なのですけれども，実は並列処理システムにおけるメモリ・マネジメントの大切さは，20年前に Illiac IV ができたときにすぐに問題になった話です．Illiac IV はメモリがプロセッサ当り 2KB で，いまから見ると本当に馬鹿みたいな小ささだったのです．メモリにステージングするのにだいたい 40m/s かかります．ということは 2KB×プロセッサ数，たとえば 256 としますと，それだけのデータを処理するのに 40m/s 以上の時間を使わないと，必ず I/O 待ちになるというわけです．これを避けるために，データ（たとえば行列）をどの方向にスイープ（sweep）するのがよいのかなどが盛んに議論されました．

似たような宿命が，現在のスーパーコンピュータにもあります．プロセッサの性能のほうがメモリに比べて非常に速いものですから，メモリのマネージメントをどうするか，ということが大きな問題になります．最近，ご存じのように FORTRAN などの逐次型プログラムを並列処理向きに解析する技術が随分進んでいます．そういった技術を援用すると，プログラムの実行順序が解析できますので，データのプレステージングやプリフェッチも随分楽になると思います．ぼちぼち研究されているようですが，そういう話が大事ではないかと思います．

むらおかよういち
村岡洋一（早稲田大学）

並列処理技術の課題

（1） システム性能
- 並列性
- メモリ（レジスタ）
  ステージング，同期，内容一致

（2） オンライン・モデリング

（3） 図形表示

（4） 計算物理学

上図の（2）から（4）は，人間まで入れたシステム性能と考えたいのです．

よく並列処理の大切さを説明するときに例に出される話として，QCD というプログラムは，DEC-11/750 でやると3万年，IBM の GF11 というスーパーコンピュータでも1年かかるということがあります．スーパーコンピュータで並列処理で計算するというのは，宿命的に大きなプログラムを計算するわけですが1週間も計算してから，「あっバグってた」ということではとてもしんどいわけです．ここでバグというのは，単なるプログラムのバグのみならず，数値解析アルゴリズムのバグまで含みます．トータルのピープル・パフォーマンスの向上という点からは，これでは困ります．

そうしますと高いレベルでアルゴリズムを記述することができるようになることが必要になってくると思います．高いレベルで記述するというと，すぐに for 文の代わりに行列文を書こうとかということになるのですが，そういう意味ではありません．数値的なアルゴリズムを高いレベルで扱えるような方法を考えて，気軽にアルゴリズムを直せるように考えませんと，スーパーコンピュータでこれからどんどん大きい問題が扱われていくときに，システムとしてのトータル・パフォーマンス向上の点から問題ではないかということです．オンライン・モデリングという言葉を使いましたが，何かオンラインでモデルをチョッチョッと簡単に変えるとすぐ計算ができる，というようなことを考えていただけるといいのではないかと思います．

（2）が入力としますと，出力としては（3）のように（いまでもだいぶ盛んにやっていますが，）図形とか，動画で簡単に結果を表示できるような話が必要になると思います．スーパーコンピュータの中でも CRAY がアメリカなどでは性能上のみならずシステムとして評判がいいのは，チャネルのスループットが 100MB 以上と高速で，これに簡単に端末を接続して結果を得ることができるからです．ですからオンラインで画像の入出力ができます．そういう意味でシステム・トータルとしてのスループットを考えていく必要があるだろうと思います．

最後に，最近コンピュテーショナル・ニューロ・サイエンスとか，コンピュテーショナル何とかとつけるのがはやっているようです．コンピュテーショナル・フィジックスという言葉があるかどうか知りませんけれども，実験物理学ではなくて計算機を使った，計算物理学という意味で，先ほど申し上げたようなオンライン・モデリングの話も含めて，応用まで踏み込まないと，並列処理は160人† の話で終わってしまうのではないか．1,600人，16,000人には広がらないのではないかということを心配しております．

ユーザからの勝手な注文

名取　「並列処理への期待と課題」ということで，今日は数値解析研究会の立場からお話したいと思います．われわれは計算機を使わせていただくという立場ですので，ユーザからの勝手な注文というのをここでお話してみたいと思います．

まずわれわれの問題である数値計算というのはいくらでも大型化するということを指摘したいと思います．一つの例として数値予報というのがあります．天気予報をするのに大気の状態を表わす偏微分方程式を解くということをやるわけです．たとえば，1辺の長さが100km，東京から熱海というような間隔でメッシュを切って計算します．それでは精度が悪いというときには，間隔を半分にして藤沢でもう1回切って50kmにするわけです．そういうことで縦，横，高さともに半分にすると，節点の数は8倍になるわけです．そうするとどういうことになるかというと，メモリはその2乗で64倍，そして演算量は3乗で512倍というわけで，すぐ512倍速い計算機が必要になります．そんな具合に際限なく大型化が進むことになります．いま現在天気予報が当たらないのは，計算機の能力が足りないからだ，といえるかもしれません．最近ではベクトル計算機がだいぶ実用化されるようになってきましたが，これから先それがどのぐらい高速化できるか，というのがまず問題です．そういう場合にはだいたい1桁といっておけば間違いないらしいのですが，10倍にはなるだろうということなのか，100倍に近いのか，その辺はよくわかりません．それでもさっきの話のような状況ですから，10倍などというのでは到底間に合わないわけです．そうしますと，ベクトル計算機も単体では無理だからということで，いわゆるマルチプロセッサ・システムでスピードを稼ぐことになります．もう一つ高並列，いわゆるパラレルプロセッサでどれだけ需要を満たすことができるか，ということが問題であろうと思います．

さて，われわれの注文の第一は従来あるプログラムは変えたくないということです．従来あるプログラムというのは，だいたい FORTRAN で書かれているわけです．計算機屋さんにいわせると，FORTRAN のような言語はもう過去のものであって，こんなものは使わないほうがよろしい，ということになるのかもしれません．しかし FORTRAN で書かれたプログラムはたくさん存在していて，それを人間の手で書き換えることはしたくない．自動的にやってほしいというのが，こちらの勝手な注文なわけです．現在すでに自動ベクトル化ということで，FORTRAN のコンパイラがやってくださっていますし，次第にいろいろなことができるようになっているわけですが，さらにそれを推し進めていただきたいという注文なわけです．

その次の「これからも FORTRAN を使いたい」ということですけれども，これから作るのだったら何かほかの言語だっていいじゃないかといわれるのではないか

† 「並列処理シンポジウム JSPP '89」の参加者数．

今後の課題

- 自動並列化
- 高並列機は専用機とする
  問題自身に内在する並列性を素直に表現するシステムを作ること
- コンパイラ
  並列性の検出をどうしたらよいか
- アーキテクチャ
  問題に内在する並列性をどう実現するか
- オペレーティング・システム
  マルチプロセッサ・高並列機の効率化と大容量メモリの管理をどうするか

という気がします．実際私もそう思わないこともないわけです．ですから，もしFORTRANを「もうやめなさい」とおっしゃるのであれば，それに代わる非常にいいものを考えていただければ，それを使うようになるだろうと思います．

自動ベクトル化については，すでに申し上げましたが，その次の自動並列化というのは何のつもりで書いたかといいますと，ベクトル計算機が複数台になりますと，プログラムの中に含まれる並列性を検出して，その複数台のプロセッサにうまく割り当て，並列処理をうまくやることが必要になります．それはもっと難しいものだと思います．それを自動並列化といってみたのですが，それがどの程度まで進んでいるのかよくわかりません．

それから最後は高並列計算機，いわゆる並列プロセッサというのがありますが，ベクトル計算機は汎用機として作られて使われているわけですが，高並列機は専用機とするのがいいのではないかと思っています．専用機というのは何かというと，問題向きに作るということです．それはどうしてかというと，問題ごとに内在する並列性というのはいろいろ異なっていると思うのです．その並列性を素直に表現するようなパラレルプロセッサを作る，ということが重要だと思うのです．それがうまくゆけば，その問題に関しては非常に高速に処理できる計算機を作ることができるわけです．それがほかの問題に使えなくてもかまわないわけで，その問題に対してだけ使えればそれでよいと思うのです．

それとは逆に，あるパラレルプロセッサを作ってみてから，その計算機に合うような問題はないだろうかなどと探す．これは本末転倒も甚だしいと思うわけです．

それから，それぞれのご専門の方々に対する課題を私なりに考えたのですが，コンパイラ屋さんに対しての課題としてはさっきもいいました並列性の検出，という問題が非常に難しいだろうというわけです．これは人間のほうで指定してくれというのが一番安易な方法です．そうすれば簡単なわけです．ところがそうではなくて，それを自動的に検出してほしい．たとえばさきほどもいいましたが，言語としてはFORTRANを使うんだといい張るわけです．その中で検出するのはあなたの役目だと，そういいたいわけです．並列処理言語のいいものを考えて，それで記述してもらって処理する，というのも一つのやり方だと思います．非常に記述しやすい言語ができれば，それはそれでもよいのかもしれないのですが，それよりはいまあるものから並列性を検出する，ということを研究していただきたいわけです．

名取 亮（筑波大学）

それからアーキテクチャ屋さんに対しての課題としては，たとえばさっきもいいましたけれども，それぞれの問題に内在する並列性をどういうふうに実現するか．それも多分いろいろな方式があるのだと思いますが，そういうことが問題なのではないだろうかと思います．

それからオペレーティング・システムを専門にされる方に対しては，マルチプロセッサになったとき，それから高並列機になったとき，それをどのように効率化するか．それから，大容量メモリがどうしても大型計算の場合には必要になるわけですが，それをどう管理するかということが重要なのではないかと思います．

あまり専門でもないのに，いい加減なことをいっている，というお叱りを受けるかもしれませんが，今日はせっかくいろいろな専門の方がお集まりになっていますので，これからはそういう方向で研究を進めていただければありがたいという勝手な注文を申し上げました．

並列処理プログラミング —パラダイムと言語—

斎藤　最近は情報処理学会の研究会なども次から次へとたくさんできています．基本的なテーマで研究会をやろうとすると，合同でやらなければそんなにテーマもないよ，というような話にもなっているわけです．並列処理というのは大変複雑な問題であるので一つの側面からだけやるのは非常に難しいわけです．このシンポジウムは六つの研究会が一緒にやっていますけれども，これからはインテグレーションといいますか，総合化してやらないとなかなか本質的な解決が出てこないのではないかということで，大変タイムリな企画ではないかと思います．

通信学会のほうはわかりませんが，情報処理学会の研究会はレベルが低いのではないか，という意見もあります．ACM とか IEEE といったシンポジウムを見ていますと，大変レベルの高いこともやっていますので，是非このシンポジウムがずっと続いて，そういうレベルの高いシンポジウムになってくれれば，大変よろしいと思います．ここでは日ごろ考えている並列処理に関するコメントを，少しだけ私なりにお話させていただきます．

一つはプログラミングをするということで，やはり並列に対するプログラミングというもの，あるいは，方法論だとかパラダイムとかいうものがあるはずなのです．パラダイムというのは大変はやっておりまして，手続き型から始まって，オブジェクト指向とかが最近よくいわれます．そういうシーケンシャルの世界の中で考えられたパラダイムと，並列プログラミングというものがどう結びつくかということを，割合本質的なところから議論するとよいのではないかと思います．データフローのように，並列処理に合ったパラダイムもありますし，論理プログラミングというのは実は本質的に結びついているとかがありますから，そういうところを少し本質的に見て，並列処理のパラダイムとしてはこういうものがいいんだ，というようなものが少しでも増えてくると，大変よろしいのではないかと思うわけです．

それから名取先生，あるいは村岡先生なども少しおっしゃいましたが，並列計算機あるいは並列プログラムというものが本当に大事だということが実感として出てきたのは，多分 Illiac IV の段階ではないかと思うのです．あのころは，FORTRAN，あるいは TRANQUIL という言語があり for 文とか do 文を並行的に実行するようなものからスタートしたわけですけれども，先ほどいったように並列性というものを，数値計算としては何も考えないでプログラムを書いて，あとはコンパイラが並列性の自動検出を行なうというのが一つの方法としてあるわけです．

それに対して，やはりそれではまずいから，陽に並列性を書いていこうという方式があり，TRANQUIL などでも同時に実行することを指示するという機能があったわけです．どっちがいいかということは何ともいえないわけで，機能がどんどん高くなれば，あるいは問題がどんどん大きくなれば，並列性を直接書いても，何をやっているのかよくわからないことになります．やはり，人間の頭はシーケンシャルにしか考えられない，というのが本質ではないかと思うのです．野崎先生の意見にはちょっと反するかもしれませんけれども，少し頭の悪い人にとってはシーケンシャルにしか考えられませんから，並列性を直接書けということは，実は結構難しいわけです．

そういう意味で，自動検出というのは非常にいいわけですが，それだけでは何となく物足りない．やはり陽に並列性を書きたいということがあるわけでありまして，並列性を陽に含んだ言語，たとえば Occam といったようなものがあるわけです．これについては議論は尽きないかと思いますが，その辺のところをしっかりと議論していただければよろしいのではないかと思うわけです．

われわれは，FORTRAN のような手続き型で，しかもシーケンシャルな性格を持った言語からスタートしています．このとき，本質的にシーケンシャルでやらなければいけないんだということは，必ずしも考えないで書いている可能性があるわけです．実際に DO 100 I=1,N という形で順番に書いていますけれども，これはたまたまN個を指定するために書くわけでありまして，その中に1からNまで，順番はついていなくてもいいというものがたくさんあるのではないかと思うのです．

そういう意味では，並列性を書くというよりも，むしろシーケンシャルな部分をどんどん削っていくと考え，余分な制限というものを書かないようにしていくと，割合と自然な形で並列性が検出しやすくなるのではないかと思うわけです．人間の頭はシーケンシャルに考えるほうが向いているわけですが，それを超えた形で現在のシーケンシャルな手続き型の言語というのはできているわけですから，Prolog，あるいは制約プログラミング(constraint programming) のようにミニマムな制約を与えるだけでアルゴリズムを書ける，あるいはプログラムを書けるということであれば，並列性の検出が割合素直に

いくのではないか，という気がいたします．

Illiac IV は1970年代初頭の研究ですから，15年以上もたっておりますが，あのころは制御の流れをちゃんと考えそれを複数個発生するといった方式であり，手続き型言語，あるいはシーケンシャルなアルゴリズムの書き方というところを素直に拡張していったわけです．Hoare の CSP だとか，分散処理というような方向から出てきたものが，メッセージ通信という形です．現在のオブジェクト指向における並列性は，そのメッセージ通信という形の書き方がいいのではないか，あるいはそのほうがずっとモダンであって，いつまでも DO SIMULTANEOUSLY なんてやっているのは時代遅れじゃないかというような話があるわけです．

確かに FORTRAN の拡張というような並列処理の言語よりも，オブジェクトというような一つの実行単位，あるいはアクティブな単位というものを管理しながら，メッセージの交換によって並列処理を進めていくほうが素直に書けるんだ，ということがよくわかればよろしいわけです．あるいは，もっと新しい方法もあるのではないかということも考えられるわけです．そんなところにも踏み込んで議論していただければ，大変面白いのではないかと思います．

超並列アーキテクチャ上の制約プログラミング

田中　私は大変大雑把な話を少しだけさせていただきたいと思います．私の考える並列処理の目的は二つあって，プログラミングは並列処理のほうが簡単で，記述力は並列処理のほうが高い，だからいいんだということを実証したい，というのが一つです．もう一つは，処理性能に関して10のN乗の高性能を実現したいということです．

研究のポイントのなかで重要ではないかと思いますのは二つあり，一つはプログラミングで，斎藤先生がおっしゃったような並列プログラミングのやり方を確立すること，もう一つは，計算機アーキテクチャ研究会をやっておりますので並列マシンの構成法です．これは制御を工夫して，十分な並列性を引き出すことが問題だろうというわけです．

そこで一つに絞って，何か並列処理の研究をやれといわれたら，次のようなことをやってみたいと思います．超並列処理システム研究，これを別の言葉でいえばニューロ・コンピューティングですが，ニューロ・コンピューティングでは，いかにも神経というのが明示的に出てくるものですから，そうしないで「超並列」と書いたわけです．けれどもここで強調したいのは次の点です．すなわち，超並列マシンをベースに用いて，その上に制約プログラミングを実現したらいいのではないかということです．この制約という意味はコンストレイント・ロジックではなく，あれは一つの例にしかすぎません．ここで申し上げたいのはもっと一般的に解を求めるときのいろいろな条件，Xは10より小さくなければいけないとか，$X+Y$は何とかより大きくなければいけないとか，そういった制約だけ式に書くと，あとは回路が自動的に動き，平衡に達して解を見つける，というニューロ・コンピューティングで，アナログコンピュータの現代版です．こういうことは，やはり並列処理のいいところだろうと思います．

斎藤信男（慶應義塾大学）

従来，並列処理がいろいろなされてきました．それはこのプロセスはこちらでやりなさい，これはこちらでやりなさいなどと，あらゆることをきちんと規定してやっていた並列処理の歴史だろうと思うのです．それが重要でないとは決して申しませんが，それはこれからどんどん研究が進むにつれてかなり進展するであろうと思いますし，ある程度の成果は得られると思います．それはやらなければいけないことですけれども，面白そうだなという興味だけからでいえば，先ほど申し上げましたような超並列処理というのはやはりやりたいな，と思います．

実はここで皆様から質問を受けて，というのが普通でございますけれども，ちょうど時間ですので，明日はまたパネルもあることですし，一応このパネルは終りにさせていただきます．どうもありがとうございました．