

## 83 Massively Parallel Implementation of Flat GHC on the Connection Machine

M.Nilsson,H.Tanaka(東京大学,日本)

### 発表要旨

本発表はFlatGHCのコネクション・マシンへの実装法に関するものである。実装が容易になるように、FlatGHCをベクトル化可能な中間言語にコンパイルした後、ベクトル化を行う。中間言語は、GHCにおいてほとんどのクローズはガードを持たないということ意識して設計されたものである。シュミレーションによれば速度は400KLIPS程度で、従来の逐次処理方式と比して速いわけではない。しかしハイパーキューブ・マシンにFlatGHCを実装することの利点は、プロセッサ数の増加にしたがって速度も向上することである。プロセッサ数を $n$ とすると速度は $O(n/\log^2 n)$ になるという。速度の上昇は、共有データの同時アクセスにより押さえられている。

MIMD言語でプログラムを書くと、ベクトルなどの問題に含まれるSIMD性を失うのではないかという質問があったが、今はまだ行っていないが最適化は可能だということである。またコネクション・マシン以外のハイパー・キューブ・マシンに本当に対応するかという質問が出されたが、これは簡単に実装できるという回答であった。

## 84 KL1 in Condition Graphs on a Connection Machine

J.Barklund,N.Hagner,M.Wafin(Uppsala Univ.,スウェーデン)

### 発表要旨

本発表は、KL1のプログラムからコンディション・グラフへのマッピングについて述べるものである。コミット付きのコンディション・グラフを実行する抽象マシンを定義し、この抽象マシンをコネクション・マシンに実装する。本方式による実装では、タームと、test,unification,linkなどの制御要素がそれぞれ1つのプロセッサに割り当てられる。このような統一は、実装を単純にするのに役立っている。コネクション・マシンへのKL1の実装で問題となる点は、負荷分散、荘園などのメタ・コール、ガード部に記述されるotherwiseなどである。

シュミレーション結果またはベンチマークの結果を知りたいという質問が出されたが、現在設計中であり、まだ実測したものはないということであった。

また、KL1のような言語では多数のプロセスを生成するプログラミング・スタイルを有するが、アクティブなゴールはそのうちの一部に過ぎないという性質があるので、コネクション・マシンへの実装ではこのことについての検討を進めるべきとのコメントがあった。