⑫ Design of an Efficient Dataflow Architecture Without Data Flow

G.R.Gao(McGill Univ.,カナダ)

発表要旨

アーギュメント・フェッチによるデータ駆動方式に基づく効率的なスタティック・データフロー・アーキテクチャが最近DennisとGaoらにより提案された。このアーキテクチャは既存の高性能パイプラインアーキテクチャと並列処理におけるデータフローモデルの結合の可能性を開くものである。鍵となる特徴は新しいアーキテクチャにおいて命令のスケジューリングはデータ駆動であるにも拘わらず，データ自体は決して流れないことである。本論文では，このアーギュメント・フェッチング・データフロー・アーキテクチャの命令セットを概観する。ある重要な特徴，すなわち，配列のための構造メモリの付加，命令セットの設計におけるsplit transaction array参照と，プロセス間通信を支援することについて述べる。我々はまた，複数の関数呼び出しが，効率的に行われるためにダイナミック・アーギュメント・フェッチング・データフロー・アーキテクチャを拡張する点についても考察する。

質疑応答

質問：従来のデータフロー・マシンに比べ，argument-fetchのメモリ・バンド幅の要求量がどうかシュミレーションしたか？

回答：した。要求量は少ない。なぜならば，我々のアーキテクチャはデータのコピーをとらないからだ。

質問：メモリに対してダイナミックなオーバレイを行うのか？

回答：そうだ。

質問：その時，カラーのようなものとしてベーシック・アドレスを使うのか？

回答：そうだ。

質問：では，メモリをダイナミックにリアロケーションする時に，どうやってカラーが同じかどうかわかるのか？

回答：各オーバレイに対してベーシック・アドレスはユニークに決まる。したがって，関数の実体に対してベーシック・アドレスをユニークに識別することができる。

質問：リアロケーション，すなわちメモリを動かしたり，リプレイスしたりする時はどうするのか？

回答：関数が生成または消滅する時だけダイナミック・アロケーションが起きるので，ベーシック・アドレスはプールなどに貯めておいて持ってくればよい。