

⑦ Estimating the Inherent Parallelism in Prolog Programs

D.C.Sehr*, L.V.Kalé(Univ. of Illinois at Urbana-Champaign,米国)

発表要旨

本論文では、Prologプログラムのソースコードを変換することによってPrologプログラムの実行時間を計測する手法について述べる。本手法によって得られるプログラムは、逐次実行することにより、逐次実行による実行時間と最大限のOR並列実行を行ったときの実行時間をそれぞれ評価する。本手法は最大限のAND/OR並列実行を行ったときの実行時間評価を行えるように拡張可能である。このときのプログラムの変換効率はそう悪化しない。いくつかのプログラムによってその結果を示す。

我々のAND並列評価法はFortranプログラムの本質的並列性の評価におけるKumar (1988)の研究に基づいている。この手法では各データ項目に対応するタイムスタンプをソースプログラムに追加し、データ項目の書き込みが行われるたびにタイムスタンプの更新を行う。追加されたプログラムによって計算される最大のタイムスタンプが、もとのプログラムの最適実行時間である。この時間をもちいて、インプリメンテーションではどれだけの並列性が引き出されているかを評価できる。

質疑応答

質問：chat_parserプログラムでAND並列性の評価をしているが、私の予想よりかなり多いので驚いた。そんなに多くのAND並列性があるとは思いが、...

回答：我々は非常に多くのAND並列性をchat_parserで見つけた。ご存知のように、AND並列の追加によりおよそ6倍の性能向上があったことより、あなたの驚きも理解できる。しかし、非常に多くのリテラルを含む節が幾つかあり、このような結果はしばしば存在する。

質問：あなた方は10個のWAMインストラクションの実行時間を計算することにより計測を行っているが...

回答：しかし、allocateやindexingインストラクションの類は省いていると断ったはずだ。このような完全なWAMコンパイラを用いるのは、私の行っている研究の重要な部分ではない。

質問：実行効率がとても良過ぎると思う。この様な、簡単な仮定はしないほうが良いのではないか。つまり、より完全なモデルでは、手続き呼び出しの方が、計測しようとしているインストラクションの実行単位時間よりむしろ時間がかかるだろう。

回答：それには別の問題がある。つまり、もし手続きの呼び出しといったレベルを考えると、計測のための変換は、実際には複雑さのためにうまくいかない。このモデルを選択したのは殆んどそういった理由による。しかし、このような仮定でも、例えばバックトラックのコストについては、この種のシステムの拡張として実現可能だと思う。そのような機構も考えてみるつもりだ。

質問：データサイズに対する速度向上については調べているか？

回答：いや、ベンチマークでの入力データについてはかなり制約を与えた。しかし、非常に簡単に調

べられると思う。そのことについて特に気になることがあるのか？

質問：わたしは、あなたの並列による速度向上の幾つかに疑いを持って。プログラムによっては、データ依存があると思われる。その依存性は予見できないことから、AND並列の速度向上はデータサイズに依存するのでは？

回答：その様な大きなデータベースでは実際に、あなたの言うようにかなりのデータ依存性があると思う。今後、この点についても調べるつもりだ。