

PIEにおけるアクティビティ制御

6F-4

～ コマンドトラヒックのシミュレーション評価

丸山 勉 , 相田 仁 , 後藤 厚宏 , 田中 英彦 , 元岡 達

(東京大学 工学部)

1 はじめに

PIEでは、並列処理される多数のゴールフレーム(GF)を関係木によって管理する。関係木は複数台のアクティビティ・コントローラ(AC)によって保持される。ACはAC間でコマンドを送受しながら関係木を操作し種々のアクティビティ制御を実現する。今回は、実際の例題プログラムにおいてどの程度のコマンドトラヒックが生じるかについてシミュレーションを行なったので報告する。

2 PIEにおけるコマンド

今回、シミュレーションを行なったコマンドは次の四種類に大別される。[1][2]

① <son >

新GFの導出に伴い関係木を伸長するとき用いられる。<son >コマンドの数は導出されたGFの数と同じである。

② <success >, <failure >

GFの決着が付いた時に、関係木を末端から刈り込みながら、親ノードに実行結果を知らせる。

③ <stop >, <activate >

メタ述語の実行等において、処理待ち/処理中のGFの強制終了や、停止中のGFの活性化を行なう。

④ <son-change >, <parent-change >, <eliminate >

子ノードをただひとつしか持たない中継ノードの多くはコマンドの中継を行なうだけなので、不要ノードとして削除することができる。これにより、関係木上での高速なコマンドの伝搬が可能となり、また資源の節約もできる。ただし、一つの中継ノードの削除には4回のコマンドの転送が必要であるため、ノードの削除は他のコマンドのトラヒックが小さく、ACが比較的暇なときに行なわれる。

3 シミュレーション結果

今回のシミュレーションでは、コマンドの処理時間、コマンドネットワークによる遅延、およびコマンドの宛先(同一AC内/他のAC)による違いは考慮せずに、コマンドトラヒックの総数について調べた。シミュレーション結果を図1~5に示す。シミュレータについては[3]を参照されたい。シミュレーションに用いたプログラムは論理回路の検証プログラム[LC]と覆面算[LL2P]である。

[LC]ではnot が使われている。not を含むGFは2つにAND分割され、それぞれのGFは対のGFの実行結果を待たずに実行される。

各図の横軸は、UP台数が16台([MC])、256台([LL2P])のときのシミュレーションクロックである。縦軸は各2000クロック間に発生したコマンド数、またはそのシミュレーションクロックにおいて存在した関係木上のノード数である。

図1、図4は不要中継ノードを削除しないときのコマンド数、また図2、図5は中継ノードを削除したときのコマンド数である。図3は[LC]における関係木のノード数である。

4 検討

まず、コマンドトラヒックについて考える。[LC]では図1より、不要中継ノードの削除をしないと処理の終盤で多くの<success >, <failure >コマンドが発生する。これに対し、削除を行なうと、図2から解るように総コマンド数は増えるが、コマンドの発生は定常的となり、ある時点で非常に多くのコマンドが発生することはない。さらに、図3に示すようにACが保持するノード数も大幅に減少する。この結果からは、不要中継ノードの削除はコマンドトラヒックの平均化、および資源の節約に有効である。ただし、[LL2P]では図4、図5に示すように、不要中継ノードの削除によって、<success >, <failure >コマンドの数は大きく減少するが、かえって総トラヒックは大きくなる。しかし、中継ノードの削除は、本来、コマンド伝搬の効率化のための操作であるため、他のコマンドの障害にならない範囲で行なえばよい。コマンドに優先権を設け、中継ノード削除用のノードに低い優先権を与えることにより、<success >等の本質的なコマンドの伝送速度を上げることが可能であると考えられる。

次に、ACの処理能力について考える。UPと同数のACがあるとすると、ACに要求される1コマンド当りの処理速度は、コマンドトラヒックのピーク時において、UPのGFの導出速度(平均約800~1000クロック)の4~6(約130~250クロック)倍である。この結果からは、ACはUPの処理を妨げることなく、コマンドを処理することができると考えられる。

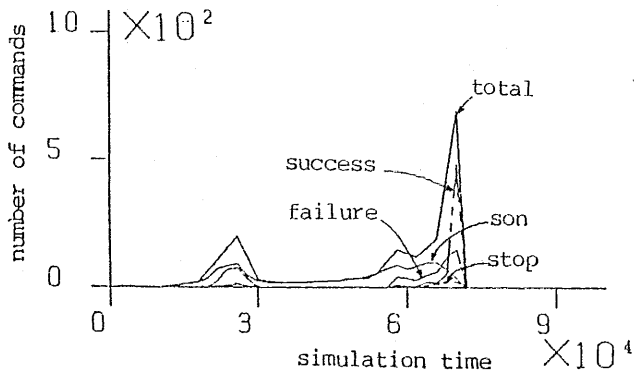


Fig. 1
Command Traffic without Relay Node Elimination
[LC] Up = 16

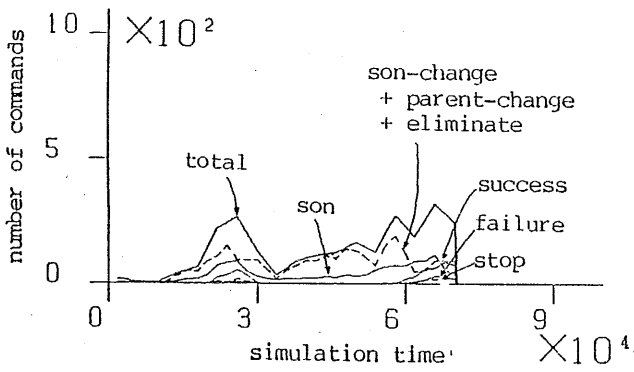


Fig. 2
Command Traffic with Relay Node Elimination
[LC] Up = 16

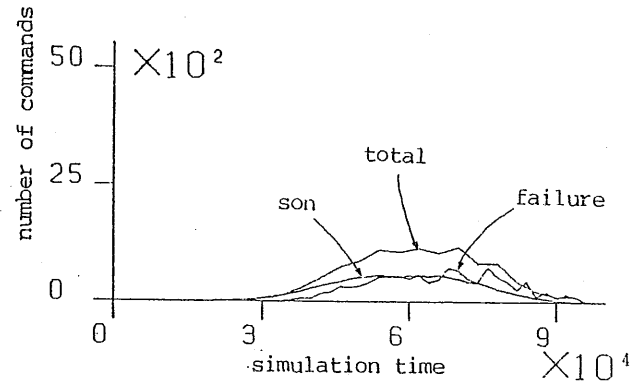


Fig. 4
Command Traffic without Relay Node Elimination
[LL2P] Up = 256

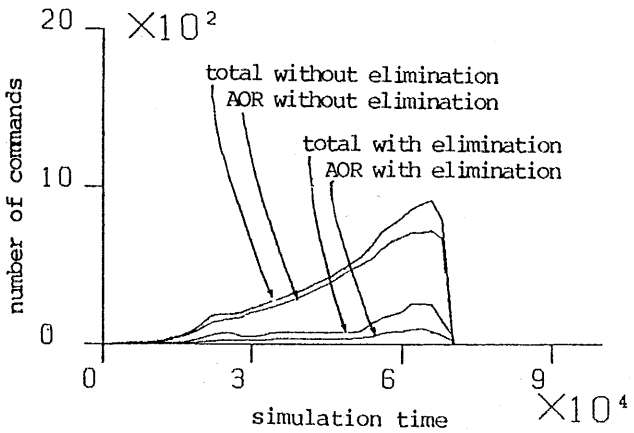


Fig. 3
Number of Tree Node
[LC] Up = 16

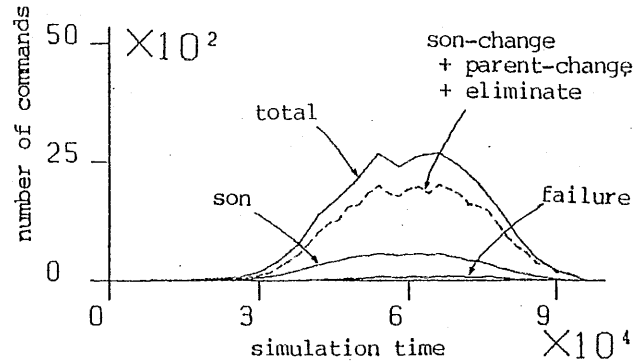


Fig. 5
Command Traffic with Relay Node Elimination
[LL2P] Up = 256

5 おわりに

PIEにおけるコマンドトラフィックの概要について述べた。ネットワークを考慮した場合、一般にAC間トラフィックが問題となりやすい。今回のシミュレーションでは、コマンドの宛先による区別をしなかったが、同一AC内のノード間ではコマンドによらずに関係木のノードを直接操作することができるので、関係木をAC間にうまく配置することによってコマンドトラフィックを減少させることができる。ただし、これはGFの分配方法と密接に関係する問題であり、今後シミュレーションを行なう予定である。同時にACの詳細設計も進める予定である。

[参考文献]

- [1] 後藤他 “高並列推論エンジンPIEについて” Logic Prof. Conf. '83 ICOT
- [2] 後藤他 “PIEにおけるアクティビティ制御～メタ述語の実現方式” 本大会 6F-3
- [3] 丸山他 “高並列推論エンジンPIE～並列度のシミュレーションとその評価” EC83-39