KL1の並列処理

2Q－2　－密結合マルチプロセッサでの並列処理系の評価－

佐藤正俊、清水肇、後藤厚宏
((財)新世代コンピュータ技術開発機構)

1．はじめに

我々は，ＩＣＯＴで開発中の並列推論マシン（ＰＩＭ）のクラスタを構成する密結合マルチプロセッサ上の並列論理型言語（ＫＬ１）の処理方式の検討として，擬似並列シミュレータを用い，負荷分散方式の検討を行ってきた［１］．本稿では，実際の密結合マルチプロセッサ（Sequent社のBalance 21000）上に開発したＫＬ１並列処理系をもとに負荷分散方式について評価を行ったので，その結果を報告する．

2．ベンチマーク・プログラム

負荷分散の戦略を比較するために使用したベンチマーク・プログラムは以下のものである．

・８クイーン問題
－ＯＲ並列探索をストリームを用いてＡＮＤ並列に単純に merge 述語を用いて書き直したもの（8q-m）
－サスペンドの可能性を排除する等の最適化を行ったもの（8q-K）
－階層化した再帰的構造（レイヤードストリーム［２］）によるもの（8q-l）
・構文解析問題（ＢＵＰ）
・並び換え問題（qsort；512要素）
・素数生成問題（primes；1000以下の素数）

これらの問題の持つ性質を明らかにするために，以下の項目について整理する．

① プログラムの大きさ

各プログラムの大きさを知るためにプロセッサ１台で実行したときの実行時間（単位は秒），リダクション数，１リダクション当たりのＫＬ１－Ｂコード数（KL1-B/reduc.）を表１に示す．

表１．プログラムの大きさ

	8q-m	8q-K	8q-l	BUP	qsort	primes
実行時間(秒)	125	54	59	52	10	21
リダクション	108K	39K	19K	36K	8K	17K
KL1-B/reduc.	10.5	14.4	32.6	15.0	14.0	14.0

② プログラムの持つ並列性

プログラムの持つ並列性を，ここではプロセッサ１台で実行した時の計算木の正規化深さと正規化広がりで表わすことにする．計算木の正規化深さとは，全リダクション中にどれだけのリダクションがdepth-first に実行可能であったかを示し，execute 命令実行数／リダクション数で表わす．また，計算木の正規化広がりとは，全リダクション中のフォークゴールとdepth-first 実行ゴールの比を示し，enqueue 命令実行数／execute命令実行数で表わす．ここでフォークゴールとは，depth-first 実行対象外のゴールでボディゴールが１つ以上ある場合，１つをdepth-first 実行対象とすると，それ以外のゴールを意味する．

計算木の正規化深さと正規化広がりを各プログラムについて表２に示す．

表２．計算木の正規化深さと正規化広がり

	8q-m	8q-K	8q-l	BUP	qsort	primes
正規化深さ	0.57	0.72	0.61	0.58	0.87	0.99
正規化広がり	0.76	0.40	0.64	0.74	0.16	0.01

③ サスペンド

各プログラムは逐次実行ではサスペンドを起こさない．

3．負荷分散方式

ＫＬ１の並列処理においてゴールキューを集中管理するとキューの管理がボトルネックとなる．さらにゴールが相互の依存性に関わりなくプロセッサに分割され，ゴール間の同期待ちによって計算量も増大することが予想される．そこで以下のような要求駆動に基づく負荷分散方式を提案した［１］．その概要は以下のとおりである．

①各プロセッサにゴールキューを持たせ，スケジューリングを個別に行なう．
②ゴールの分配は，他のプロセッサがアイドルとなった時を契機とする．

ここでは，実際の密結合マルチプロセッサ上で，①システムに１つのキューを置き，各プロセッサがゴールを取り合う戦略，②各プロセッサに個別のキューを置き，各プロセッサがゴールを他のプロセッサにランダムに分配する戦略の２つと比較することにより，要求駆動に基づく負荷分散戦略③の評価を行なう．以下では①をcommon，②をrandom，③を requestと呼ぶ．

プログラムの並列性が充分あるＢＵＰを例に，プロセッサ台数を１～９台変化させたときの各戦略の台数効果の様子を図１に示す．また，各プロセッサの稼働率，サスペンド数，処理に要したＫＬ１－Ｂ命令数の比（プロセッサ１台と８台）のデータを表３に示す．

Parallel Processing of KL1
- Evaluation on shared memory multiprocessor.-
Masatoshi SATO, Hajime SHIMIZU, Atsuhiro GOTO
Institute for New Generation Computer Technology (ICOT)

表３より，各戦略とも稼働率を高く保っているにもかかわらず，図１で示すようにrequest 戦略は，他の戦略に比べて高い台数効果を得ている．この原因は，①②の方式では，不必要にゴールが分配されたためサスペンドが増加し，結果としてＫＬ１－Ｂ命令の実行数が増えたためと考えられる．

表３．戦略による稼動率とサスペンド数

	common	random	request
稼働率	99.8%	94.1%	96.8%
サスペンド数	9.3K	6.2K	1.0K
KL1-B数比	1.27	1.15	0.98

４．ベンチマークによる台数効果

各ベンチマークに対して，要求ベースのゴール分配におけるプロセッサ１～９台での台数効果の変化を図２に示す．図２より，プログラムに並列性が充分存在している場合（クイーン問題や構文解析問題）は，台数効果が確認できる．また，充分に並列性の無い場合（並び換えや素数生成）は，台数効果は飽和の傾向にある．

4.1. 台数効果の現れ方の比較

プログラムに並列性が充分存在している場合でも，そのプログラムの台数効果の現れ方（傾き）が異なっている．ここでは，高い台数効果の8q-Kと低い台数効果の8q-lについて，稼働率，サスペンド数，KL1-B数比を表４に示す．

表４．ベンチマークの比較

	稼動率	サスペンド数	KL1-B数比
8q-K	98.6%	0	1.00
8q-l	97.4%	5.9K	1.12

表４より，8q-lはサスペンドの増加によりKL1-B数が増加していることが判る．このようにKL1-B数は，負荷分散の戦略のみならず，プログラム・スタイルによっても変化する．また本処理系は，サスペンドのコストが比較的大きいためサスペンドを減らす工夫（処理系，プログラム共に）が高い台数効果を得るポイントと考えられる．

4.2. 問題の並列性と処理系の抽出度

問題の特性として定義した並列性（正規化深さと正規化広がり）と処理系による台数効果の現れ方について考察する．処理系にとって，与えられた問題の並列性抽出の度合は，実際に分配したゴール数／分配可能数で表わすことができる．この値を，処理系の並列性抽出度と呼ぶ．この値は１に近いほど，処理系にとって問題の並列性を取り出しにくい（要求してもすぐにゴールをもらえない状況でありアイドル率が増す）状態にあると考えられる．プロセッサ台数を変化させた場合の各プログラムの並列性抽出度を表５に示す．

表５より明らかなように，primesやqsort の正規化広がりが小さい問題では，処理系の並列性抽出度は既に高い値（並列性を取り出しにくい状態）を示している．例えば，primesではプロセッサ２台の時点ですでに，分配可能なゴールのほとんどを分配していることがわかる．事実，図２においてもprimesはプロセッサ２台で台数効果は飽和している．またqsort では，並列抽出度はプロセッサ４台で0.2，プロセッサ８台で0.5で，これに伴ってアイドル率が増し台数効果が飽和の傾向にある（図２）．

以上より，正規化広がりの小さなプログラムでは，ここで採用した要求ベースの負荷分散方式は，そのプログラムの持っている並列性を充分引き出せないことが判る．

また，正規化深さは，台数効果が飽和していない状態では，プロセッサ間のゴール分配を低く抑えられることを表わすと考えられる．つまり，正規化深さの大きい8q-Kは，BUP に比べ並列性抽出度は小さい（表５）．

表５．並列性抽出度の変化

	8q-K	BUP	qsort	primes
8台	0.051	0.071	0.532	1.000
4台	0.019	0.028	0.214	0.970
2台	0.014	0.013	0.028	0.970

５．おわりに

負荷分散方式として要求駆動に基づく負荷分散方式は，プログラムの持つ並列性が高い場合に有効であることを示した．また今後は，さらに台数効果を引き出すために，サスペンドを減らす工夫をしていく予定である．

<参考文献>

［１］佐藤他，" 並列推論マシンＰＩＭ－クラスタ内実－　験処理系－"，情報処理第34回全国大会予稿2P-4

［２］奥村他，" レイヤードストリームを用いた並列プログラミング"，LPC'87 (ICOT), June 1987

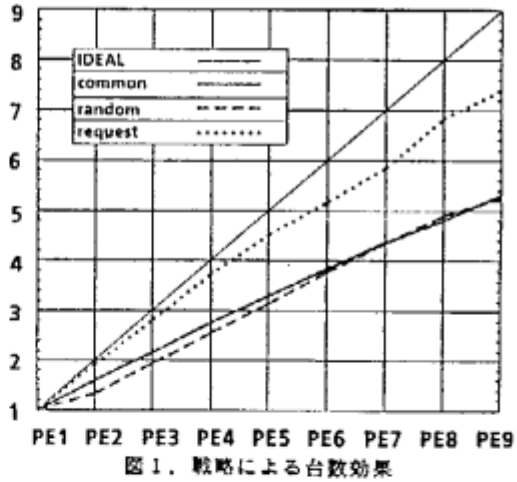

図１．戦略による台数効果

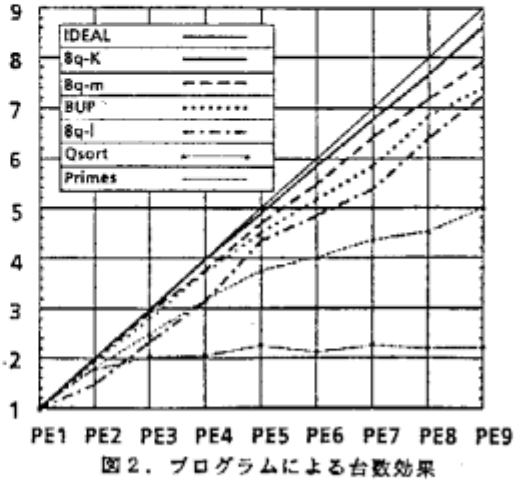

図２．プログラムによる台数効果