

共有メモリ結合マルチプロセッサにおける KL1 向きガーベジコレクション

- MRB を用いた最適化と並列実行 -

今井 明 川合 英夫 中川 貴之 中島 克人 後藤 厚宏

新世代コンピュータ技術開発機構 (ICOT)

はじめに

我々は、並列論理型言語 KL1 を高速に実行する並列推論マシン PIM を開発中である。KL1 は、副作用を持たない言語であるため、システム全体での処理能力向上のためには、効率の良いガーベジコレクション (塵集め:GC) の実装は不可欠である。このため我々は、既に MRB 方式 [1] という実時間 GC 方式を提案しているが、MRB 方式では、すべての塵を実時間で回収出来ないため、一括型 GC と併用する必要がある。

PIM はクラスタを用いた階層構成で、クラスタ内の要素プロセッサ (PE) は共有メモリ / 共有バスで接続される。各 PE は、コピーバック方式の一貫性キャッシュを持ち、メモリアクセスの局所性が高い場合には、バス競合を抑えることができるように設計されている。

一括 GC を高速に実行するためには、クラスタ内の全てのプロセッサで GC 処理を並列実行すればよい。しかし、一括 GC は基本的にメモリアクセスに局所性がないため、キャッシュミスによる共有バス獲得待ちが頻発し、思いうる台数効果が得られないことが予想される。そこで、一括 GC の並列実行で台数効果を稼ぐためには共有バスを出来るだけ使わないことが特に重要である。本稿は、この方針に基づいて設計された一括 GC の並列実行に関して述べたものである。

1 MRB を利用した最適化

一括 GC の基本的なアルゴリズムには、一括移動法を採用する。これは、ヒープ空間を 2 分割し、片方の半空間がなくなった時点で、生きているセルをもう片方の半空間にコピーする方式で、一括 GC 時に「生きているセル」のみにアクセスするだけでよいからである。

移動法では、同一のセルを多重にコピーしないように

するために、コピー済みのセルには、コピー先の新領域のアドレスを書き込む必要がある。MRB 方式では、ポインタに付与した 1 ビットのタグにより、参照先が単一参照か多重参照かを判定できる。そこで、単一参照であることが明かなセルには、コピー先のアドレスを書き込む必要がない。MRB 方式の評価 [1] では、ベンチマークによって異なるが、半数以上のセルが単一参照であることが判明しており、書き込みによるバス使用を大幅に削減することができる。

2 ローカルページへのコピー

2.1 ヒープトップの更新頻度低下

単純に並列化すると、セルをコピーする毎に共有ヒープトップを更新することになる。共有ヒープトップの値は共有資源であり、これを排他制御しながら更新するとバストラフィックが増大するため、この更新頻度を下げることは重要である。そのために、共有ヒープトップを論理ページ単位 (256 ワード) で更新し、ヒープトップの更新頻度を抑える [2]。なお、コピー先のページは PE 個別の領域となるため、書き込みに関して排他制御は不要である。また、この書き込みには Direct Write という書き込みで先立って該当ブロックをキャッシュメモリにフュッチしないメモリ操作コマンド [3] が使える。通常の Write コマンドを使った場合に比べて、共有バス使用を大幅に削減することができる。

2.2 セルの大きさ毎のページ獲得

論理ページ単位でヒープを伸ばすと、現在確保しているページ残部以上の大きさの構造体をコピーできなくなる問題が起こる。そこで、構造体のサイズ毎にページを確保し、1つのページには同じ大きさの構造体しかコピーしないとすることによって、この問題を解決する。

すなわち、構造体の割り付けは、ページサイズ以下の構造体は 2 の冪乗、ページサイズ以上の構造体はページサイズの整数倍で丸めて割り付けることによって、ペー

Garbage Collection for KL1 on Shared Memory Multi-Processor
- Optimization by MRB and Parallel Execution -
Akira Imai, Hideo Kawai, Takayuki Nakagawa, Katsuto Nakajima
Atsuhiko Goto (ICOT)

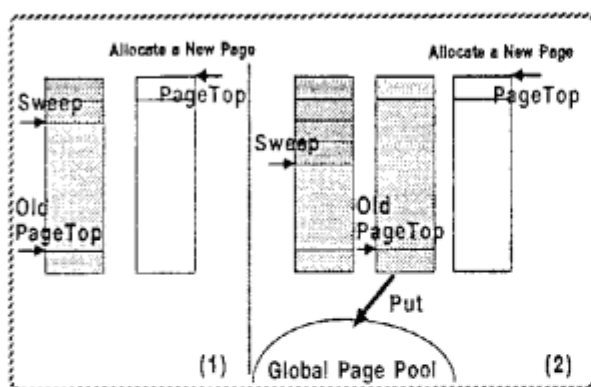


図 1: 獲得ページにコピーし尽くした時の処理

ジの最後にコピーできない構造体はなくなる。なお、このような割り付け方法は、MRB による実時間回収のためのフリーリストの本数を抑える意味も持つ。

2.3 スイープ方法と負荷分散

従来方式では、新領域を順にスイープして未コピーセルを捜してコピーを行うが、本方式では、ページ単位でヒープを更新した結果、新領域が連続でない。そのため、次のページを覚えておく仕組みが必要となる。獲得したページにコピーし尽くし、次のページを確保する時に、スイープポインタがこのページを指しているかどうかによって処理は異なる (図 1)。

- (1) このページを指していた場合
単に次のページを確保する。
- (2) このページを指していない場合
グローバルなスイープページプールにこのページを入れて、次のページを確保する。

すなわち、スイープポインタとページトップポインタが 2 ページ以上離れる場合にスイープページプールに入れることになる。このページプールのための領域には、新領域の最後部が使える。

一方、スイープポインタ側は、

- ページ最後に到達すると、ページトップの指すページの先頭に移る。
- ページトップと一致した場合、スイープするページを他のサイズのページに切り替える。

すべてのサイズのページで、ページトップとスイープのポインタが一致した場合、ページプールよりページを取り出してそのページをスイープする。これによって、各

PE 間の負荷の偏りを均等化することができる。ページプールが空になり、全ての PE の全てのサイズのページトップとスイープのポインタが一致した時点で、GC 処理は終了する。なお、一度両ポインタが一致したサイズも、他のサイズのページをスイープ中にページトップが伸びることもあるので、注意が必要である。

3 GC 後処理の高速化

トータルな性能向上のためには、GC 後のリダクション処理を高速化する処理を GC 処理に組み込むことも重要である。そのためには、既に提案しているデレファレンスや、MRB のメンテナンスを行う方式 [4] に加え、構造体のキャッシュブロック (4 ワード) 境界合わせを行うことで、ヒット率向上 / ブロック共有率低下を計る。これは、前節の方式で、自動的に境界合わせが出来るようになっている。

GC 後の処理を高速化する処理の是非は、GC 時にその処理のために余分に掛かる時間と、GC 後にその処理の効果で減少する時間とのトレードオフである。しかし、GC 時にはバス使用率が高いので、PE 内部での計算が多少増える傾向にあっても、バス獲得待ちによるアイドルよりも有益であると考えている。

おわりに

本処理方式は、現在 VPIM (Virtual-PIM) と呼ぶ PIM 実機命令への変換を意識した KL1 処理系に実装中である。VPIM エミュレータでの本方式の評価に関しては、別の機会に報告したい。

貴重など助言、ご討論を頂く ICOT および関連メーカーの PIM/Multi-PSI 研究開発スタッフに御礼申し上げます。

参考文献

- [1] 木村他 “KL1 の多重参照ビットによる GC 方式について”, データフローワークショップ '87 予稿集, pp 215-222, 1987
- [2] 佐藤他 “KL1 並列処理系の評価 - メモリ消費特性と GC -”, JSPP'89 予稿集, pp 195-202, 1989
- [3] 松本他 “KL1 のメモリ参照特性に適した並列キャッシュ機構”, データフローワークショップ '87 予稿集, pp 223-230, 1987
- [4] 宮内他 “MRB による多重参照管理方式 - KL1 処理系における一括型 GC の特性評価” 第 35 回情処全大 2Q-5, 1987