

# 分散共有メモリ SMS における分散割付 API 使用時の性能評価

片野 真吾・緑川 博子・飯塚 肇

成蹊大学大学院工学研究科情報処理専攻

## 1. はじめに

筆者らは、クラスタ型並列コンピュータシステム上に共有メモリプログラミング環境を実現する、ソフトウェア分散共有メモリ SMS の設計、実装を行ってきた<sup>[1]</sup>。このようなシステムでは、アプリケーションのアクセスパターンと一致するようにデータを配置することや、通信の局所性を避けるためにデータを分散して配置することが重要である。そこで、共有データを大域多次元配列宣言のような形式で宣言し、また並列計算で頻繁に使用するパターンに分散割付するための API を実装した<sup>[2]</sup>。

本報告では、分散割付 API を用いた時の SMS の性能評価について報告する。

## 2. 割付単位とページ境界

並列計算では、図 1 のような様々な割付が行われる。これらの割付はいずれも特定のサイズのブロックを各プロセスにサイクリックに割り付けることで実現できる。しかし、ページ単位の割付を行うシステムでは、ユーザの指定したブロックサイズがページサイズの倍数でなかった場合は、ユーザの希望とは違った割付がされてしまう。図2は、分散共有メモリシステム JIAJIA<sup>[3]</sup>での割付例である。ブロックサイズをページサイズに切り上げるのでずれが生じ、ユーザの希望と違った割付になる。図3は、本システムの API での割付例である。同一ページに複数のプロセスが指定された場合は、そのページをより多くアクセス可能性の高いプロセスを優先する。この機構により、ユーザがページ境界を意識せずに記述しても最善の割付がなされる。

## 3. 分散割付 API 使用時の性能評価

従来、ページを跨ぐブロックサイズを考慮した割付機構が存在しなかったため、常にページサイズを意識した割付を行い評価を行ってきた。そこで、今回はページサイズを跨ぐような割付に焦点を当て SMS の評価を行った。

### 3.1 評価環境

評価環境として表1に示す PC クラスタを用いて評価を行った。

### 3.2 ライン型割付

LU 分解プログラムを用いて評価を行った。LU 分解は処理が進むにしたがって行ごとの計算量の差が大きくなるので、各割付プロセス間の負荷バランスを良くするためにライン割付が適している。しかし、割付単位(ライン割付の場合は1行のサイズ)がページサイズと一致しない場合は、アクセスパターンと割付結果にずれが

生じるため、性能が著しく低下する。図4は、ページを跨ぐ時(1200x1200)の SMS と JIAJIA の実行時間比較である。JIAJIA では、割付単位が小さいほど割付のずれが蓄積するので速度が著しく低下する。このことから、ずれが最も少なくなるバンド割付を用いることになる。行列が小さければこれで十分である。しかし、サイズが大きくなった場合、負荷バランスが悪くなり効率が低下する。SMS では、ページを跨ぐ場合でもずれが生じることがないので、複数行を1ラインとして割り付けることにより、負荷バランスを良くすることが出来る。図5は、LU 分解に複数ライン割付を行ったときの実行時間である。サイズの大きいときの方が効果が大きく、バンド割付よりも効率が良くなる。

### 3.3 タイル型割付

タイル割付は多くの DSM で容易に記述できないので、代わりにバンド割付が用いられることが多い。そこで、メモリアクセスのローカルリティを人為的に操作するマイクロベンチマークプログラムを用いて、本 API で提供するタイル割付がデータアクセスローカルティに対してどの程度効果があるのかをタイル割付とバンド割付に関して比較した。

このプログラムはタイル型のアクセスパターンを持つ。例えばプロセス1の場合は、

write : 図6の斜線部分、つまりタイル割付時に割り付けられる領域(own area と呼ぶ)に対してのみ write を行う

read : 指定した確率で own area から read を行う。それ以外の場合は own area 以外から read を行う。例えば80を指定すると、80%は own area から 20%はその他の領域から read を行う。

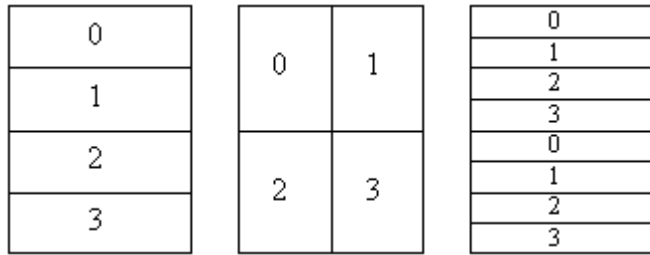
以上のようなアクセスパターンで、read した値を write する作業を10000回行う。図7は、2048x2048(ページ境界を跨がない)と2400x2400(ページ境界を跨ぐ)で実行したときの実行時間である。図8は、実行時の通信量である。この結果から、バンド割付を行った場合は割付とアクセスパターンが一致していないため、read 時のページ転送や write 時の DIFF 送信により性能がかなり低下することが分かる。

## 4. おわりに

本報告では、分散割付 API を使用したときの性能について評価した。分散割付 API を使用することで、ページ境界と一致しない場合でも比較的効率の良い割付が可能となり、自由度の高い並列プログラムの作成が容易になった。

**参考文献**

- [1] 緑川, 飯塚: "ユーザーレベルソフトウェア分散共有メモリ SMS の設計と実装", 情報処理学会論文誌ハイパフォーマンスコンピューティングシステム Vol.42, No. SIG9 (HPS 3), pp.170-190 (2001,8)
- [2] 緑川, 片野, 飯塚: "分散共有メモリ SMS における共有変数の分散割付API", 第1回FIT全国大会, B-42, 2002-9
- [3] M.R.Eskicioglu et. al., : Evaluation of the JIAJIA Software DSM System on High Performance Computer Architectures, Procs.of the 32nd Hawaii International Conf. on System Sciences., 1999



(a) バンド割付 (b) タイル割付 (c) ライン割付

図1 割付方式(4プロセスの例)

表1 測定環境

CPU	INTEL PENTIUM III-S 1.13GHZ
メモリ	512MB
ネットワーク	Intel PRO/1000T
OS	Redhat Linux7.2 kernel2.4
台数	8

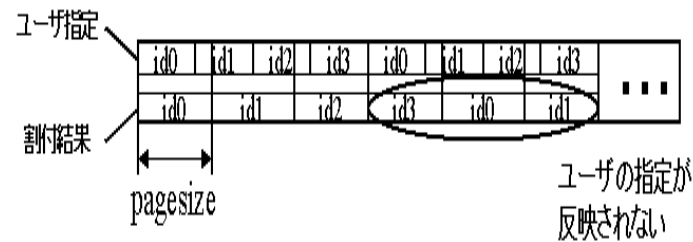


図2 JIAJIA の割付方式

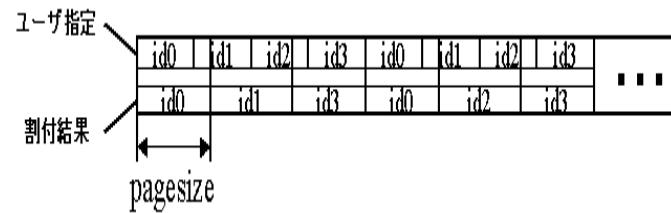


図3 SMS の割付方式

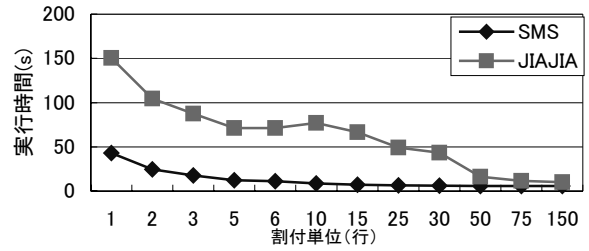


図4 SMSとJIAJIAの比較(1200x1200)

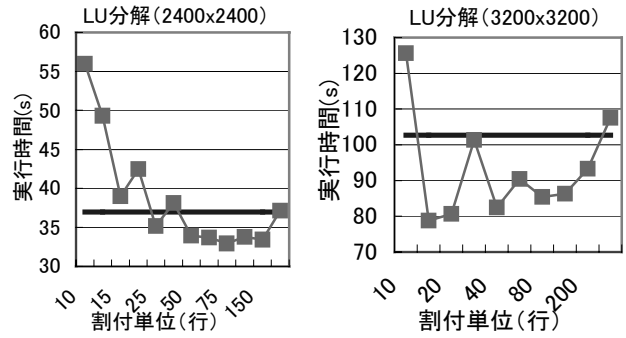


図5 ライン型割付の性能評価

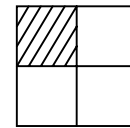


図6 own area

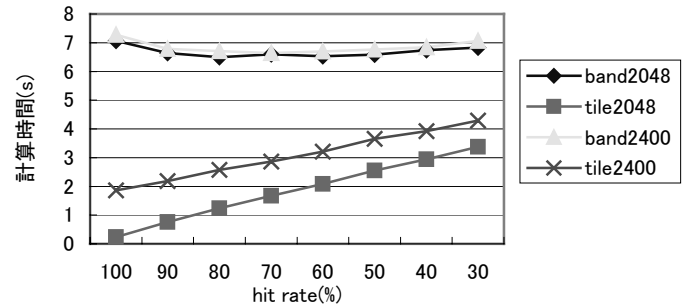


図7 タイル割付とバンド割付の比較

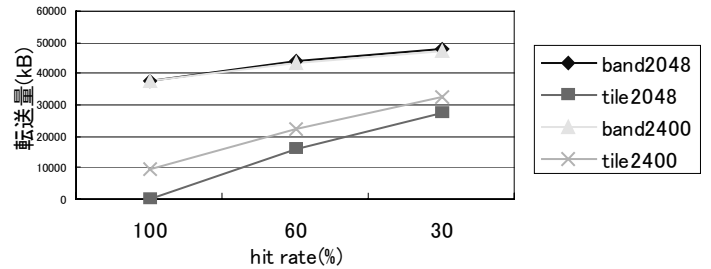


図8 計算時の転送量比較