

Mach における仮想記憶のページサイズによる影響

5 G-1

数藤義明^o 宮本剛 岩本信一 柴山茂樹
キャノン(株) 情報システム研究所

1 はじめに

UNIXをはじめとして現在多くのOSが仮想記憶を用いており、現在までに数多くの仮想記憶の研究がなされている。CMUにおいて開発されたMachオペレーティング・システムにおいてはターゲットに依存しない仮想記憶管理インターフェースを持ち、仮想記憶の研究における良いプラットフォームを提供している。本論文では、Machの仮想記憶管理システムが提供する仮想ページの機構を用い、物理ページサイズと仮想ページサイズの影響を独立に考慮することによって、両ページサイズがシステム全体のメモリ効率と速度に与える影響についての考察を行なった。さらに実際に物理ページサイズの処理速度に与える影響を計測した結果を報告する。

2 Mach内のページ処理

Machでは機種に依存しない仮想記憶管理インターフェースが提供されており、汎用的に機種に独立して利用できる仮想記憶管理システムと、機種に依存した仮想記憶管理システムとに分離されている。vmシステムとよばれる機種に独立した仮想記憶管理システムでは、メモリの割り当てや解放などの基本的な機能の他に、詳細なメモリの保護機能や継承機能などが実装されている。また仮想記憶管理とページングのシステムが論理的に分離されているのも大きな特徴であり、ユーザレベルで作成されたページャ(外部ページャ)を使用することも可能となっている。仮想記憶管理システムの機種依存部はpmapと呼ばれるモジュールによって実装されており、実際のMMUハードウェアによってアクセスされるページテーブルを管理している。このpmapモジュールのインターフェースも様々な仮想記憶管理ハードウェアに対応することが可能なものとなっている。

このような柔軟な仮想記憶管理システムのもとで、Machではpmapより上位の仮想記憶管理部分での単位ページ(仮想ページ)のサイズとMMUハードウェアの単位ページ(物理ページ)のサイズをOSの初期化時に変えることが可能である。この機能を用いて物理ページサイズと仮想ページサイズの影響を独立に測定する

ことが可能となる。ここで仮想ページとはページフォールト時に外部デバイスより主記憶にローディングされマップされる単位となる。一方物理ページはハードウェアに依存するページテーブルの1エントリに対応したページである。現在のアーキテクチャでは、1回のページフォールトについて一般には1個または複数個の物理ページが割り当てられ、ページテーブルに登録される。実装上の理由から仮想ページサイズは物理ページサイズの2の累乗倍が一般的に用いられる。

3 物理ページサイズの考察

物理ページサイズの処理速度に与える影響は、TLBのヒット率とページテーブルの登録処理である。物理ページはTLBの1エントリに対応するので、限られたTLBエントリに対しては、物理ページが大きいほどヒット率に有利である。ページテーブルのサイズに関しても、物理ページが大きいほどテーブルのサイズが小さくできるのでメモリ効率もあがる。しかし、TLBにヒットしてアドレス変換を受けた仮想ページは、主記憶上に存在しなければならないという現在のアーキテクチャがもつ制約から、物理ページサイズは仮想ページサイズを越えられないという条件が存在する。

物理ページサイズの自由度はハードウェアに依存しており、かつ従来は物理ページサイズによって仮想ページサイズも決まっていたため、あるアプリケーションに対して適当なページサイズを持つアーキテクチャが、必ずしも別のアプリケーションに対して有効とは限らなかった。Machは、このハードウェアに依存した物理ページサイズと、アプリケーションに適切な仮想ページサイズの関係を、制約つきではあるが、柔軟にしたと捉えることができる。

4 仮想ページサイズの考察

デマンドページングによって実行ファイルがローディングされるのが仮想ページ単位であることから、仮想ページサイズの処理速度に与える影響は、主にページフォルトの処理時間とディスクからのローディング時のデータ転送に関係するものである。ページフォルト

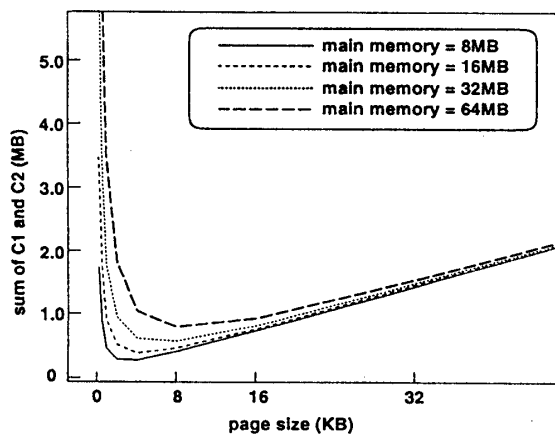


図 1: 仮想ページサイズと $C1 + C2$ との関係

の回数や頻度は実行しているアプリケーションやページ置換のアルゴリズムに依存したものである。ローディング時のデータ転送に関するものとして、直接的なディスクからのデータの転送時間に関するものと命令キャッシュのフラッシュに関するものなどが考えられる。以上のように、単純に処理速度に与える仮想ページサイズの影響を述べることは不可能である。

Mach の仮想記憶管理システムでは仮想ページに対応したページ管理用の多くのデータを持っている。そのために仮想ページサイズがシステム全体のメモリ効率に大きく影響する。まず 1 ページの仮想ページに対応するページ管理データのサイズを c とする。主記憶の容量を A 、仮想ページサイズを s としたとき、全ページ数は A/s となり、ページ管理に必要なデータの総量 $C1$ は cA/s で見積もることが可能である。また p 個のプロセスが動作している時に、フラグメンテーションによってセグメントの最後のページで無駄になるメモリのサイズ $C2$ は、簡単に $3ps/2$ で見積もることが可能である。これらから $C1 + C2$ を最小にするようなページサイズ s_0 は $\sqrt{2cA/3p}$ で算出できる。Mach の場合、仮想ページ管理に必要なデータは $c = 54$ バイト (モトローラ MC68040 と gcc を使用) であり、主記憶の容量 $A = 30 * 1024^2$ バイト、 $p = 30$ 個としたときには最適ページサイズ $s_0 = 6 * 1024$ バイトとなる。図 1 に仮想ページサイズと、仮想ページ管理に必要なデータ総量 $C1$ とフラグメンテーションによって無駄になるメモリのサイズ $C2$ の和との関係を示す。

5 実験

以上のような考察を元に、仮想ページサイズを一定にして物理ページサイズの処理速度への影響を計測した。実験は MC68040 上で行なった。仮想ページサイズを 8KB で固定し、物理ページサイズを 4KB と 8KB に

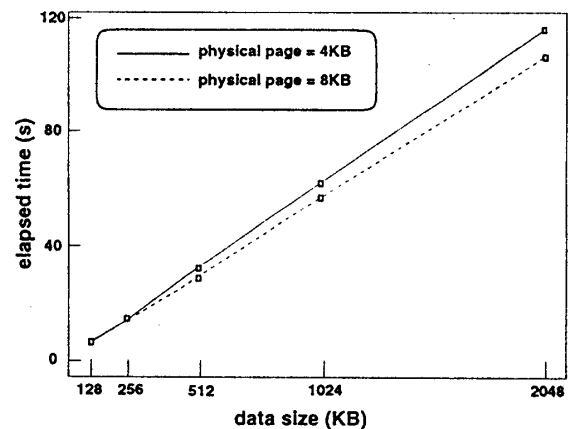


図 2: 物理ページサイズを変化させた時のデータサイズと処理速度の関係

して、様々な大きさの配列をアクセスするプログラムの実行時間を計測した (図 2)。この図からデータサイズが小さい時には、物理ページサイズはほとんど影響しないことがわかる。しかしデータサイズが 256KB を超えるところで、物理ページが 4KB のシステムでは 8KB のシステムと比較して実行時間が増加していることがわかる。これは MC68040 内の TLB は 64 エントリであり、大きなデータを一度に扱うものでは物理ページが 4KB のシステムでは TLB のヒット率が低下して処理速度を低下させるからである。

6 おわりに

物理ページサイズの影響と仮想ページサイズの影響を、処理速度とメモリ効率の両方について独立に考察した。それによると物理ページサイズは仮想ページサイズ以下という制約下では大きい方が良く、実際に Mach 上のアプリケーションを用いて処理速度を計測して確認した。また、メモリ効率に関して言えば仮想ページサイズの近似的な最適値が求まる。しかし仮想ページサイズの処理速度に対する影響は、アプリケーションに依存したものであり、またページ置換アルゴリズムにも依存しているために、一概には述べるできない。

今後の課題としては、仮想ページサイズの処理速度に対する影響の更に詳細な考察と実験が必要であると考えている。さらにページ置換アルゴリズムやマルチプロセッサ実行環境に対する考察も行なうつもりである。

参考文献

- 1) Tevanian, A., Jr.: Architecture-Independent Virtual Memory Management for Parallel and Distributed Environments: The Mach Approach, PhD thesis, CMU (1987).