

モバイルコンピューティングシステムにおける メモリ管理方式

横山繁盛[†] 清水正貴[‡] 渡辺 尚[‡] 水野忠則[‡]

モバイル端末とサーバとで構成されるモバイルコンピューティングシステムは、無線通信の帯域幅の問題や通信コスト、モバイル端末のバッテリーの持続時間等の課題がある。一般的なアプリケーションも電子メールやWebのブラウザを除くと少ない。モバイル端末の一部のメモリ領域とサーバのメモリの一部の領域が共通メモリとなるように構成し、この共通メモリの内容の一貫性管理を行うことにより、モバイル端末やサーバから自由に共通メモリにアクセス可能となる。モバイルコンピューティング共通メモリ管理方式(MCCMS)を提案する。これによりモバイル端末やサーバでのアプリケーションプログラムは通信を意識する必要がなくなるためプログラムの構築が容易となり、また通信の効率を上げることができる。アプリケーションプログラムのモデルによるシミュレーションを行い、従来方式に比べ通信時間が減少し、実行時間が短縮されることを示す。

Memory Management System for Mobile Computing System

Shigemori Yokoyama,[†] Masaki Shimizu,[‡] Takashi Watanabe,[‡]
and Tadanori Mizuno[‡]

Mobile Computing Systems that consist of mobile hosts and servers has several issues that are narrow bandwidth wireless communications, the limited battery duration of mobile hosts, and others. There are a little number of general applications except electronic mail systems and web browsers. We propose the Mobile Computing Common Memory Management System (MCCMS) which controls a part of memory areas of a mobile host and a part of memory areas of a server to be shared, and maintains the consistency of the common memory areas. MCCMS makes application programs on the mobile host and on the server easier to be developed and increases the efficiency of communications. The proposed system was evaluated by the simulator using sample application programs. The results show that the communication time decreases and the execution speed increases compared to the former system.

1. はじめに

無線通信技術の発展による携帯電話やPHS等の急速な普及、携帯情報端末の高性能化、小型化に伴い、モバイルコンピューティング環境が実現されつつある。しかしながら現状は、無線通信の伝送帯域幅が狭く接続の継続性に難点があり、モバイル端末においてはバッテリーの動作時間が短く、

CPU性能やメモリ容量等に制約がある等の課題が多い。モバイル端末とサーバとが連携した一般的なアプリケーションは、電子メールや、Webブラウザを除くと少ない。これはモバイル端末とサーバとの間は通信が必要となり、通信も各種の媒体や通信方式があるため、媒体ごとに通信処理をアプリケーションに作り込む必要があることが理由の一つと考えられる。モバイル端末とサーバのメモリが共通メモリになるように構成し、通信を意識することなくメモリへのアクセスができればアプリケーションの構築が容易となる。

[†]三菱電機(株)情報システム製作所
Information Systems Engineering Center, Mitsubishi Electric Corp.
[‡]静岡大学情報学部
Faculty of Information, Shizuoka Univ.

従来より共通メモリ、または共有メモリに関し、多くの研究がおこなわれており、また実用化例も多い。本論文ではモバイルコンピューティングシステムに適した新しいメモリ管理方式、Mobile Computing Common Memory Management System(MCCMS)、についての提案をおこなう。

まず第2章でメモリ方式について概括し、続いて第3章でMCCMSのメモリ管理方式について述べ、第4章では有効性の確認のためのシミュレーションによる評価結果について述べ、第5章でまとめをおこなう。

2. メモリ方式

ストアプログラム方式の計算機の発明以来、計算処理の高速化のため処理装置の高速化とともに、メモリの高速化のための研究がおこなわれてきている。メモリの高速化にはメモリ素子自身の高速化とメモリの方式技術による高速化が行われている。

メモリの高速化のための方式技術の中での重要なメモリ技術はキャッシュメモリであろう。キャッシュメモリは比較的低速なメモリと処理装置の間に高速であるが小容量のメモリを置き、当面必要とする分を收容する方式であり、当初は大型の汎用計算機で採用されたが、その有用性や、半導体技術の進展によるメモリの高速化、大容量化、低価格化により、現在では比較的廉価なワンチップマイクロコンピュータにも採用されるようになってきている。またキャッシュメモリの方式についても各種の方式が提案され、また実用化されている。

一方処理速度の高速化のため複数処理装置で構成された各種の方式が提案、実用化されている。一つのメモリを複数の処理装置で共有したマルチプロセッサシステム、メモリを共有しないマルチコンピュータシステム、処理装置毎にメモリを持ち、それらのメモリが一つのメモリ空間となるように制御する、分散共有メモリシステム等の各種方式が研究および実用化されてきている³⁾。

一つのメモリを共有するマルチプロセッサはメモリのボトルネックのため接続できる処理装置数に制約があり、より大規模な構成向きできない。一方マルチコンピュータは大規模な構成は可能であるが、メモリを共有していないため、プログラミングが個別対応となり、プログラムの構築や流用性に難点がある。プログラムの容易化や、汎用性のためには、単

一のメモリ空間を持つ方式が望ましく、大規模なシステム向けとしては、分散共有メモリ方式が適している。これは物理的に離れていたり異なる構成を持つメモリを一つのアドレス空間となるように構成する方式であり、このための各種の制御方式が研究されている³⁾⁴⁾。これらのシステムにおいても、プログラムの効率的な実行のためには、メモリの物理的構成を意識したプログラミングが必要である。なお共有メモリや分散共有メモリでは、処理装置とメモリまたはメモリ間は高速の専用バスや、高速の通信網で接続されているのが一般的である。

本論文で提案するメモリ管理方式、MCCMSの特徴および従来方式との違いは次の通りである。

- モバイル端末とサーバはそれぞれ異なるアドレス空間を持つが、一部のメモリ空間を共有する(モバイル端末とサーバが一つのアドレス空間を持つように構成することも可能である)。
- モバイル端末とサーバは空間的に離れており、通信は無線が基本である。
- 通信の帯域が狭く、通信が不可となる場合もある。
- 共通メモリ領域の内容の一貫性の制御は、ハードウェアと基本ソフトウェアにより管理される。

3. MCCMSのメモリ管理方式

(1) システムの概要

MCCMSは、複数のモバイル端末とサーバとで構成されるモバイルコンピューティングシステムにおいて、モバイル端末とサーバとを連携させるためのメモリ管理方式であり、モバイル端末のメモリ空間の一部とサーバのメモリ空間の一部とが共通領域と

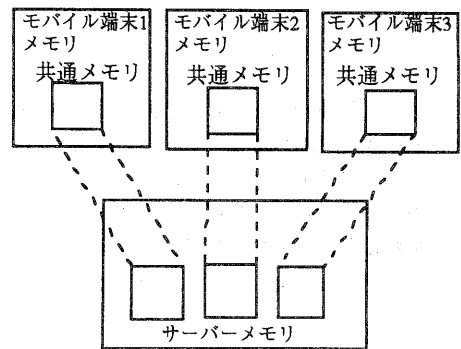


図1 MCCMSのメモリ管理方式

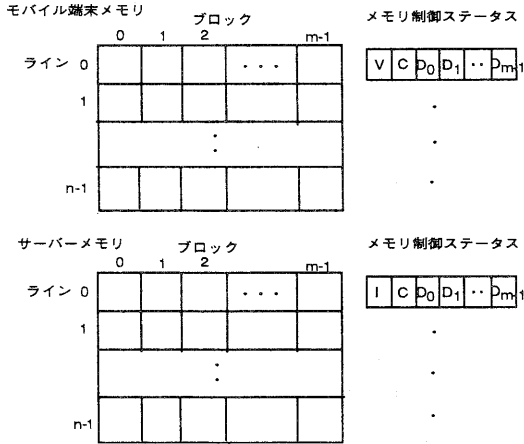


図2 メモリライン、ブロックとメモリ制御ステータス

ICD	サーバのメモリラインステータス
000	最新、アクセス権有り
001	最新、アクセス権有り、内容を変更した
01-	モバイル端末に書き移された、アクセス権なし
1--	初期状態、内容無効

VCD	端末のメモリラインステータス
100	最新、アクセス権有り
101	最新、アクセス権有り、内容を変更した
11-	サーバに書き移された、アクセス権なし
0--	初期状態、内容無効

表1 メモリラインのステータス

としてアクセス可能となり、またサーバからも、端末のメモリ空間の一部が自メモリ空間としてアクセス可能となる。

提案のメモリ管理方式を採用することにより、サーバ側にデータを持ち、モバイル端末からデータの参照や更新をおこなうグループスケジュール管理などのアプリケーションや、サーバ上のデータベースの検索や更新、サーバ経由で他のモバイル端末と通信する等のモバイル端末とサーバとの間で通信をおこなうアプリケーションにおいて、通信を直接意識する必要がなくなるため、アプリケーションの構築が容易となり、必要時のみ通信をおこなうため、通信の効率化を図ることができる。

(2)メモリ管理方式

a. メモリ制御ステータスメモリ

サーバとモバイル端末のメモリの共通領域を、

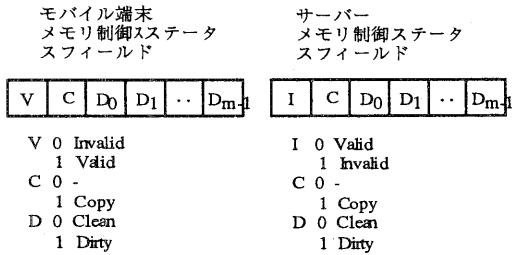


図3 メモリ制御ステータスフィールド

なるように構成し、その内容の一貫性をする方式である。図1にその概念図を示す。モバイル端末から、サーバのメモリ空間の一部が自メモリ空間

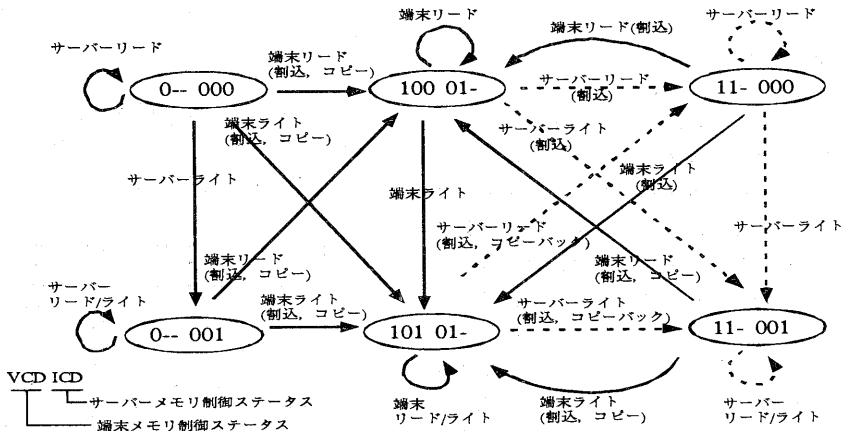


図4 メモリ制御ステータス遷移図

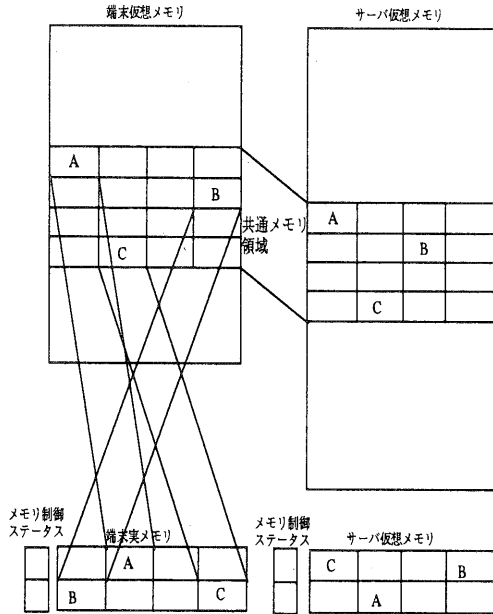


図5 仮想メモリ方式による端末メモリ空間の拡大

ラインと呼ぶ固定長の領域に分割し、ライン単位にエントリを持つメモリ制御ステータスメモリを設け共通領域のメモリ管理をおこなう。図2にメモリラインとメモリ制御ステータスの対応を示す。ラインをさらにブロックに分割し、メモリの書き込みや、書き戻しをブロックの単位で管理する。ラインの大きさは16-4096 バイト程度、ブロックの大きさは16-1024 バイト程度を想定する。メモリ制御ステータスメモリのエントリのフィールドを図3に、メモリラインのステータスを表1に示す。

b. メモリアクセス制御

アクセス権のあるラインにアクセスが行われた場合には、メモリの内容は有効であり正常にメモリリードまたはメモリライトがおこなわれ、ライトの場合には変更ビット(D)を1にセットする。アクセス権のないメモリにアクセスが行われた場合には、メモリ例外の割り込みを発生させ、相手側のメモリラインのステータスを調べ、変更ビットが1になっていれば、メモリラインの内容を書き移してアクセス権を取得し、変更ビットが0であればメモリの内容は最新のため書き移しはせず、アクセス権のみの取得を行う。図4にメモリ制御

ステータスの遷移図を示す。

c. 通信が接続不可の場合の制御

メモリ例外の割り込みが発生したが通信が不可となった場合には、接続不可をプログラムに対して通知するとともに、メモリ制御ステータスフィールドの内容によりメモリのステータスを通知する。これによりアプリケーションによっては、条件付きでデータを利用することが可能となる。

(3) アドレス空間の拡大

端末の共通メモリ領域を仮想メモリ方式で管理することにより、端末側のメモリ空間の拡大が可能となる。図5に仮想メモリ方式による端末のメモリ空間の拡大の図を示す。これは端末の仮想メモリ空間の一部を共通メモリ領域に割り当て、サーバ側の共通メモリ領域と同一となるように管理することで実現する。サーバ側の共通メモリ空間は、実メモリでも仮想メモリでもよい。端末の仮想メモリ空間の共通領域に対して、その一部のみが実メモリに割り当てられる。このためサーバから端末のメモリラインへの転送が必要となり、かつ実メモリが不足する場合にはページの入れ替えの制御がおこなわれる。仮想メモリ方式のメモリ管理は、ページを単位とするため、MCCMS方式のラインのサイズの整数倍がページのサイズとなるように、ラインのサイズを選択する必要がある。

4. シミュレーションによる評価

アプリケーションプログラムの例としてスケジューリング管理プログラムの簡易モデルを作成し、ラインサイズと通信時間、ラインサイズと実行時間についてシミュレーションによる評価をおこなった。表2にスケジューリング管理プログラムで用いたデータのパラメータを示す。端末でのスケジューリングデータは1画面で1日分の表示とし、日の指定の順序、参照、変更を組み合わせたアクセスパターンを例を作成してシミュレーションを

総データ量	60Kバイト
スケジューリングデータ量	50日分
1日のデータ量	1152バイト
1項目のデータ量	48バイト
通信速度	9600Bps

表2 スケジューリング管理データのパラメータ

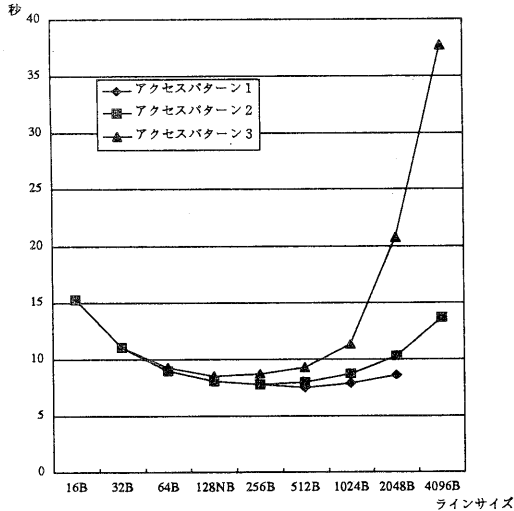


図6 ラインサイズと通信時間

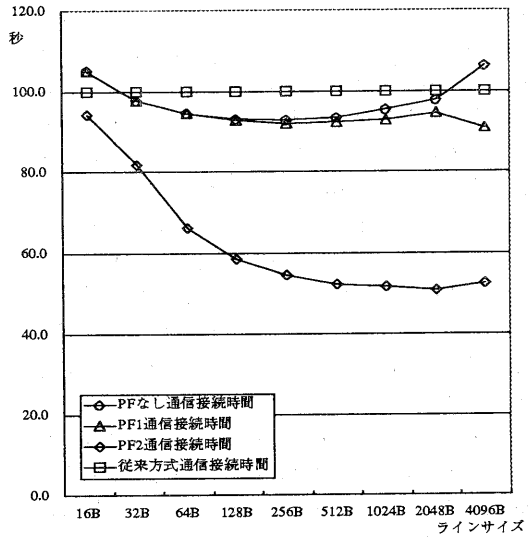


図8 ラインサイズと通信接続時間

行った。図6に連続した順番のアクセス、逆順のアクセス、7日飛びのアクセスの3種のアクセスパターンについてのラインサイズと通信時間の関係を示す。ラインサイズが128Bから512B近辺が最も通信時間が短くなるのがわかる。ラインサイズが小さくなると実行時間が増大するのは、通信プロトコルのデータ部分の割合が減少し通信のオーバーヘッドが増大するためである。ラインサイズが大きくなったときに実行時間が増大するのは使用しないデータを転送する割合が増加するた

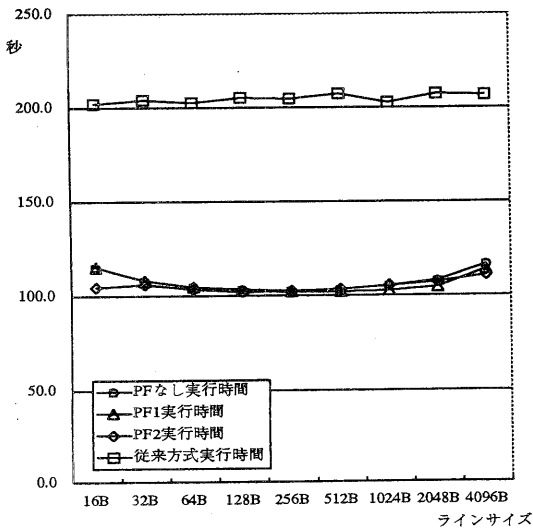


図7 ラインサイズと実行時間

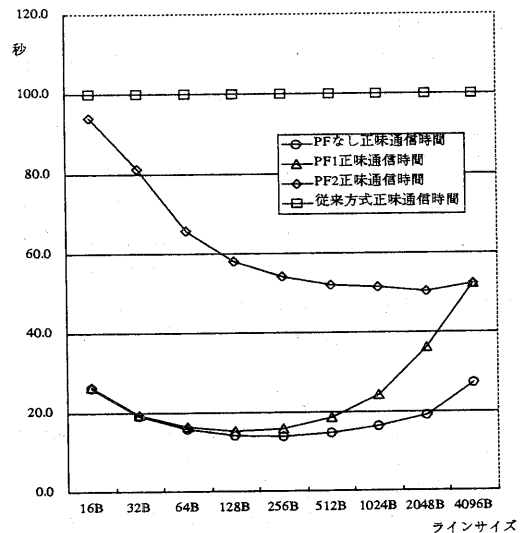


図9 ラインサイズと通信正味時間

めである。図7、図8、図9に従来方式と本方式の実行時間、通信接続時間および正味通信時間の関係を示す。本方式の場合にはさらにプリフェッチする場合とプリフェッチしない場合を示す。プリフェッチとは通信接続時、データ転送が行われていない時に、直前に転送がおこなわれたラインに続くラインを転送する動作であり、1ラインプリフェッチする場合と(図の中でPF1で示す)、通信の

空き時間のあるかぎりプリフェッチをおこなう場合の(図7の中でPF2で示す), 2とおりを示す. 従来方式の場合には最初に全データを書き移し, 最後に全データを書き戻すものとする. 本方式の場合には必要になったときに必要なラインのみ転送し, 最後に書き込みのおこなわれたブロックのみ書き戻す. 通信接続は, 従来方式の場合には最初のデータの書き移しの終了後一旦切断し, 最後の書き戻し時に再度接続するものとした. 本方式の通信接続は, 実行中に全データが書き移されれば通信を切断し, 最後の書き戻し時に再接続し, 実行中に全データの書き移しが行われなければ, 最後の書き戻しまで通信の切断はおこなわないものとした. 通信接続時間は時間による従量課金方式の通信費用に対応し, 通信正味時間はデータ量による従量課金方式の通信費用に対応する. 本方式では従来方式に較べ, 通信時間が減少し, 実行時間が高速化されることがわかる.

5. おわりに

以上モバイルコンピューティングシステムにおけるアプリケーションの構築とその実行に適したメモリ管理方式の提案とその評価について述べ, 本方式が従来方式に比較し, 通信時間の効率化が図られ, 実行時間の高速化が可能になることを示した. なお今回は9600Bpsの通信速度と1つのアプリケーションプログラムの例での評価を行ったが, 今後異なる通信速度および他のアプリケーションプログラムの例による評価をおこなう.

参考文献

- 1) Alan Jay Smith "Cache Memories", Computing Surveys, Vol. 14, No. 13, Sept. 1982
- 2) A.S.タネンバウム,水野他訳,"分散オペレーティングシステム," プレンティスホール,1996
- 3) Lenoski, D., Laudon, J., Gharachorloo, K., Gupta, A. and Hennessy, J. "The Directory-Based Cache Coherence Protocol for DASH Multiprocessor," 17th ISCA, 1990
- 4) Kuskin, J., et al "The Stanford FLASH Multiprocessor" Proc. 21st ISCA, 1994
- 5) Lovet, T. and Clapp, R. "STiNG: A CC-NUMA Computer System for the Commercial Marketplace," Proc. 23rd ISCA, 1986
- 6) 田頭茂明, 稲田文武, 最所圭三, 福田晃, "移動計算機情報発信環境におけるキャッシュの更新方式について," 信学技報, CPSY98-39 (1998-05)