

モバイルコンピューティングシステムの

2Q-2

メモリ管理方式の一考察*

清水正貴[‡] 横山繁盛[†] 渡辺 尚[‡] 水野忠則[‡]
 静岡大学情報学部 三菱電機（株） 静岡大学情報学部

1. はじめに

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。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)についての提案と、その評価について報告する。

2. MCCMS の概要

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、共通領域となるように構成する。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ことで、通信の効率化を図ることができる。

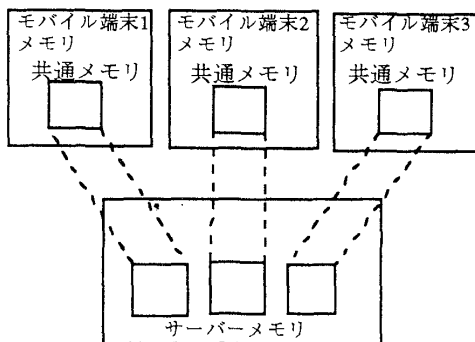


図1 MCCMS のメモリ管理方式

3. MCCMS のメモリ管理方式

(1)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、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により共通領域のメモリ管理をおこなう。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2に、メモリラインのステータスを図3に示す。

(2)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せず、アクセス権のみの取得を行う。図4に端末側でのメモリアクセス時のメモリステータスの遷移とメモリ動作を示す。サーバ側も同様である。

(3)通信が接続不可の場合の制御

メモリ例外の割込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

モバイル端末
メモリ制御ステータス
フィールド

サーバ
メモリ制御ステータス
フィールド

V	C	D0	D1	...	Dm-1	I	C	D0	D1	...	Dm-1
V	0 Invalid 1 Valid					I	0 Valid 1 Invalid				
C	0 - 1 Copy					C	0 - 1 Copy				
D	0 Clean 1 Dirty					D	0 Clean 1 Dirty				

図2 メモリ制御ステータスフィールド

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

図3 メモリラインのステータス

* A Study of Memory management system for mobile computing system

† Shigemori Yokoyama, Information Systems Engineering Center, Mitsubishi Electric Corp.

‡ Takashi Watanabe and Tadanori Mizuno, Faculty of Information, Shizuoka Univ.

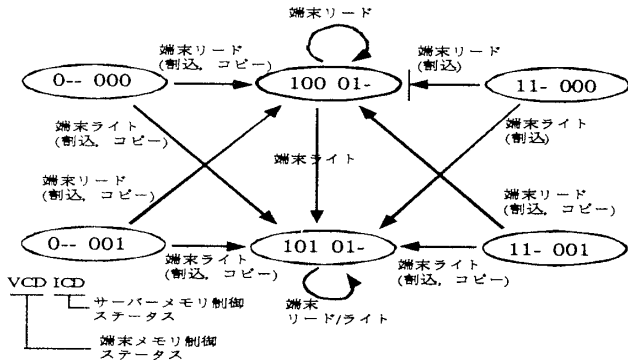


図4 端末側のメモリアクセスによるメモリ制御ステータス遷移図

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

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ル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/通信時間についてシミュレーションにより評価をおこなった。表1にスケジュール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ルデータは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の例を作成してシミュレーションを行った。図5に3種のアクセスパターンを仮定し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。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めである。図6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

総データ量	60Kバイト
スケジュールデータ量	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1 スケジュール管理データのパラメータ

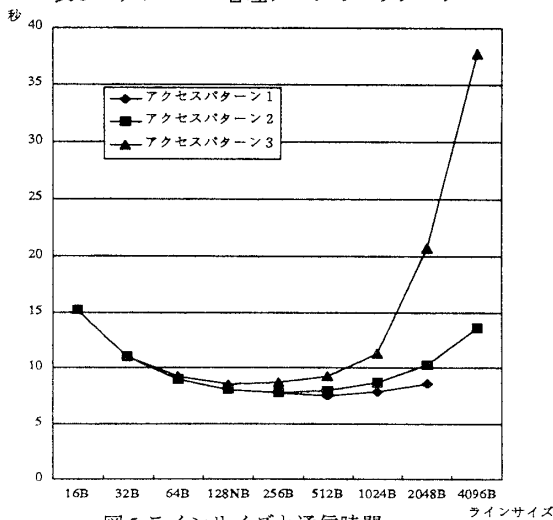


図5 ラインサイズと通信時間

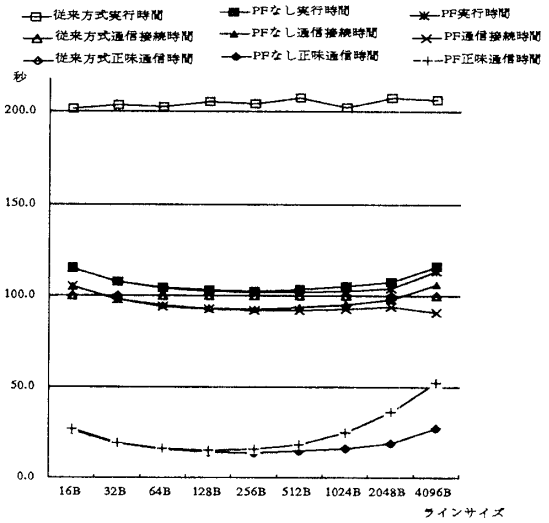


図6 スケジュール管理, ラインサイズと実行時間/通信時間

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1ラインを転送する動作である。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] 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
 [2] 田頭茂明, 福田文武, 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)
 [3] 有賀健一, 丹羽裕史, 菊地庸之, 南沢岳明, 松本英博, “モバイルコンピューティングにおける有線-無線間シームレス通信のためのミドルウェア,” DICO M O '98, July 1998