

T-Kernel における分散共有メモリの研究

山原 亨[†] 片瀬 由貴[†] 寺島 悠貴[†] 大谷 真[†]湘南工科大学[†]

1. はじめに

組込み機器の高性能・高機能化に伴い、センサーネットワークシステムなどの協調動作する自律的なユビキタスシステムが増えている。このようなシステムを効率的に開発するためには、複数の機器のメモリを透過的に扱える分散共有メモリ (Distributed Shared Memory: DSM) が必要だと考えた。

既に、この考えに基づき組込み OS である T-Kernel に分散共有メモリの実装を行った。開発したソフトウェアは、分散共有メモリとしてのメモリ制御を行う DSMM[1]と機器間的高速メモリ転送を行う RMAD[2]から成る。分散共有メモリ機能をサポートした実用的な組込み OS はほとんど無いため、組込み OS 上での分散共有メモリ機能の実装可能性の検証を重視して開発を行った。しかし、性能的に不十分な所が残っている。

本論文では現在の T-Kernel 分散共有メモリ機能の状況と課題、そしてこれからの方針について述べる。

2. 開発したソフトウェア

組込み OS には最新のオープンソースリアルタイム OS である T-Kernel を用い、さらに拡張プログラムである T-Kernel/SE を使い MMU を使用できるようにした。開発したソフトウェアは DSMM と RMAD から成る (図 1)。

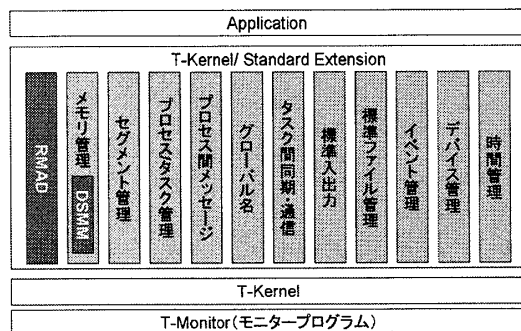


図 1. アーキテクチャ

Study on Distributed Shared Memory for T-Kernel
Toru Yamahara[†], Yuki Katase[†], Yuki Terashima[†], Makoto Oya[†]
Shonan Institute of Technology[†]

2.1 DSMM (Distributed Shared Memory Middleware)

DSMM は分散共有メモリを実現するためのミドルウェアであり、ローカルメモリと共有仮想メモリの管理を行う。共有仮想メモリとは複数の機器間で共有している仮想メモリのことである。DSMM は T-Kernel/SE のメモリ管理サブシステムの拡張として実装した。DSMM の詳細については「3. 設計」に記述する。

2.2 RMAD (Remote Memory Access Driver)

RMAD は DSMM をサポートするための通信デバイスドライバであり、他の機器との通信とメモリ転送を行う。共有仮想メモリに対してページフォールトが発生すると DSMM に処理が移る。その後 DSMM から RMAD が呼び出されて、共有仮想メモリの実メモリを持つ機器と通信を行い、メモリ転送をする。

ページフォールト中は他の処理をブロックするため、処理の高速化が必要である。高速メモリ転送を実現するために、このソフトウェア用に設計した通信プロトコルを使用した。

3. 設計

3.1 メモリ実体保証方式

組込み機器はリソースが少ないために、他の機器からのメモリ転送時に自分の物理メモリが確保できない可能性がある。これを防ぐために全ての機器があらかじめ共有仮想メモリと同じサイズの物理メモリを確保しておく方式にした。

3.2 メモリー貫性保証方式

複数の機器から共有仮想メモリに対して同時にアクセスが起こる可能性があり、この時にもメモリ内容の一貫性が保たれている必要がある。これに関しては厳密な一貫性を保持することで解決した。つまり、複数の機器間において同じ仮想ページの有効になっている数が一つになるようにすることで、仮想ページが有効になっている機器を通さないとそのメモリにアクセスできないようにするということである。

3.3 メモリ参照方式

ページフォールト時にその仮想ページの実メモリを参照するには、欲しい実メモリがどの機

器にあるのかを知っている必要がある。そこでそれぞれの機器が実メモリの位置のリストを持つことで解決した。

最初に共有仮想メモリを作成した段階では、全ての機器のリストの内容は作成を行った機器を指すようになっている。その後はメモリ転送が起こる度にリストが更新される。リストの更新はメモリ転送に関わった機器でのみ行われる。

4. 課題

今まではコードをより小さく、より単純にして、実装可能性に注力できるようにしていた。しかしそれでは性能的に不十分な所があるので改良していく必要がある。

4.1 DSMM の改良

■メモリ実体保証方式の課題と改良案

現在の方式では常に共有仮想メモリと同じサイズの物理メモリを確保している。これでは無駄なメモリも確保していて、メモリ消費量も大きくなる可能性がある。

この解決案としては必要になる度に物理メモリを確保する方式が考えられる。最初に共有仮想メモリを作製した機器に実メモリを配置しておき、それ以外の機器は共有仮想メモリにアクセスした時に物理メモリを確保する。この方式の利点はシステム全体のメモリ消費量を最小限に抑えられることだが、メモリ転送の度にメモリを確保する時間が掛かるということと物理メモリ不足時の処理を考慮しなければいけないという欠点もある。

■メモリー貫性保証方式の課題と改良案

現在の方式ではある仮想ページが有効になっている機器は一つのみである。それ以外の機器がそのページにアクセスするには、有効なページを持つ機器からメモリ内容を受け取らなければならない。つまり無効な仮想ページにアクセスする度に必ず通信が発生してしまう。

これはキャッシング機能を実装することで通信回数を大幅に減らすことができる可能性がある。他の機器に移動したページが再び必要になった際はまずページテーブルの持つダーティフラグを確認して、更新されていなければ保持しておいたページを利用する。キャッシングを実装するためにはキャッシュを保存しておくためのバッファやページの更新を知らせるために発生する通信を考慮する必要がある。

■メモリ参照方式の課題と改良案

リストの更新はメモリ転送に関わった機器でのみ行われるため更新処理そのものは軽い実装になっている。しかしメモリ転送に関わらな

かった機器はリストの情報古いままである。古いリストをもとにメモリ参照を行うと要求をたらい回しにされる可能性がある(図 2)。

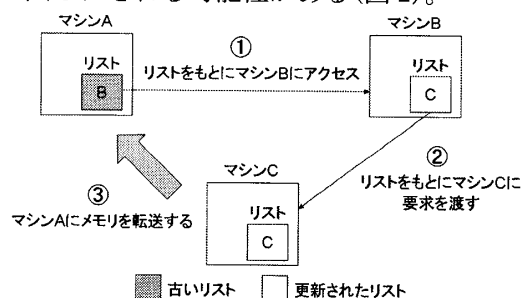


図 2. 要求のたらい回し

図 2 の場合は一回要求が回されるだけだが、これがたくさんの機器で行われるとメモリ参照のオーバーヘッドが大きくなってしまふ。

最も単純な解決方法はリストの更新の度にブロードキャストを使って全機器に更新を知らせることである。しかしこの方法はあまり共有仮想メモリにアクセスしない機器にも更新が頻繁に起こってしまう。なので、よくメモリ転送が起こる機器同士でグループを作りマルチキャストで更新を知らせる方法なども考えられる。

4.2 RMAD の改良

現在の RMAD は DSMM と同様に単純な実装になっている。あくまでも高速なメモリ転送を実現するためのもので、排他制御やメモリ保護などの機能はついていない。そういった仕組みはアプリケーション側が OS の機能を利用するなどしなければならない。しかしアプリケーション開発の負担を減らすためにも、排他制御やメモリ保護機能の実装を検討する必要がある。

5. まとめ

現在の T-Kernel 分散共有メモリ機能の課題について考察した。今後は本ソフトウェアの性能向上を行い、より実用的なものに仕上げる。

また本ソフトウェアは韓国で開かれた組込みソフトウェアコンテスト[3]での発表や T-Kernel の配布を行っている T-Engine Forum が主催の TRONSHOW2010[4]で発表を行った。

参考文献

1. 松原、山原他、組込み OS における分散共有メモリの研究、情報処理学会第 71 回全国大会、2009
2. 山原、松原他、T-Kernel 分散共有メモリ機能のためのメモリ高速転送の実現、情報処理学会第 71 回全国大会、2009
3. The World Embedded Software Contest 2009, <http://eswcontest.com/>
4. TRONSHOW2010, <http://www.tronshow.org/>