1J-8

T-Kernel における分散共有メモリの研究

山原　亨†　　刈　一杰†　　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、複数の機器同士で協調動作を行うシステムが増えてきている。こういったシステムを効率的に開発する技術として分散共有メモリが考えられる。

我々は既に最新の組込み OS である T-Kernel 上で分散共有メモリミドルウェア[1][2]を開発している。今までは 1 対 1 を前提にしたメモリ共有しかできなかったので、3 機以上でメモリ共有をできるようにした。さらに分散共有メモリの補助機能としてメモリ同期機能を追加した。

2. T-Kernel 分散共有メモリ機能

2.1 今までの開発状況

分散共有メモリ(Distributed Shared Memory :DSM)は複数のシステムが仮想メモリ空間の一部を共有する技術である。一般的に DSM は大型サーバなどの並列コンピュータで使われている技術であり、プログラムの負荷分散による高速処理を目的に利用されている。しかし、この研究においては DSM の透過性に着目し、アプリケーション間の簡便な情報共有を目的としている。

ここでは機器間で共有しているメモリのことを共有仮想メモリと呼ぶ。共有仮想メモリのメモリ実体はそれぞれの機器に分散しており、他の機器が持つメモリにアクセスした場合は通信が起こる。通信は共有仮想メモリにアクセスした際カーネルレベルで自動的に行われるので、アプリケーションが通信を意識する必要はない。

2.2 問題点と解決案

現状ではまだ 1 対 1 の単純なメモリ共有しかすることができず、3 機以上でもメモリ共有をできるようにする必要がある。

またメモリ共有を補助する機能は何も無く、現状のままではアプリケーションの開発者が DSM を意識してプログラムを書く必要がある。それでは DSM の透過性の効果が薄れてしまう。そこでメモリ共有を補助する機能の 1 つとしてメモリ同期機能を実装することにした。

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

3. 3 機以上によるメモリ共有

3.1 メモリ実体の位置解決

3 機以上では共有仮想メモリのメモリ実体がどの機器にあるのか把握する必要がある。これはそれぞれの機器が位置解決用のテーブルで管理することにした。自分の持つメモリにアクセスする際は通常通りにアクセスできるが、他の機器が持つメモリにアクセスする場合は位置解決テーブルを見てメモリ実体を持つ機器に要求を送りメモリ転送を行う（図 1）。

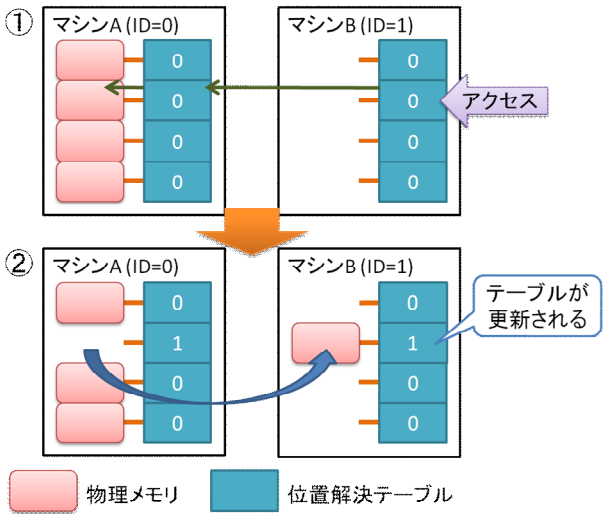

図 1. メモリ実体の位置解決

3.2 位置解決テーブルの更新

位置解決テーブルの更新には 2 種類の方法が考えられる。1 つはメモリ転送時に全ての機器にメモリ転送が行われたことを通知する方式である。この場合共有仮想メモリへのアクセス時間が短くなるがメモリ転送の度に通知のオーバヘッドが掛かる。もう 1 つはメモリ転送を行った機器だけがテーブルの更新をする方式である。この場合テーブルの更新処理が簡単になるが更新されない機器がある分メモリアクセスが複雑になる。前者においてはせっかく通知を行っても、通知を受け取った機器がそのメモリにアクセスするとは限らない。比較的遅いネットワークで繋がる組込みシステムでは無駄な通信はできるだけ避けたいので、前者は不採用とし本研究では後者の方式を選択することにした。

3.3 メモリ転送方式

メモリ転送を行った機器だけがテーブルの更新をする方式では前述したようにテーブルの更新がされてない機器がある。そこでメモリ転送の実装を図2のようにした。

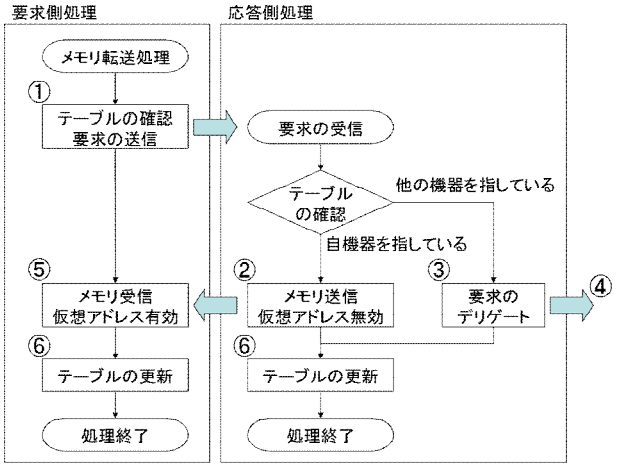

図2.メモリ転送フローチャート

図のように応答側はテーブルの値によって処理を変える。自機器を指している場合(図 2.②)通常通りメモリを送り返す。他の機器を指している場合(図 2.③)はその機器に対して要求を転送(デリゲート)する。デリゲートされた要求を受け取った機器(図 2.④)は同じように応答側処理を行う。このようにデリゲートを繰り返すことで、テーブルの更新がされていない機器でもメモリ実体を持つ機器に辿り着けるようにしている。それぞれの機器は最後にテーブルの更新(図 2.⑥)をしてからメモリ転送処理を終了する。

4. メモリ同期機能

4.1 概要

通常のプロセス間の同期ではメッセージ通信による同期が一般的である。その場合はアプリケーションが明示的にメッセージの送信と受信をする必要がある。つまり非透過なのである。この研究の DSM は透過性を基本方針として実装しているため、メッセージ通信による同期では不十分と考えた。よって T-Kernel 分散共有メモリ機能には専用の同期機能を実装した。

4.2 機能説明

メモリ同期機能は特定の機器が共有仮想メモリの特定の位置に書き込みをするのを待つ機能である。読み込み側機器が読み込みを行おうとした時、書き込み側機器が書き込みをしていなければ、書き込まれるまで読み込みが待たされる。書き込み側機器による書き込みが既に行われていれば、通常通り読み込める。

メモリ同期機能を実装するにあたり DSMsync()という API を追加した。DSMsync()を呼び出すと指定した機器の書き込みの監視を始め、これ以降の書き込みに対して同期を行う(図 3)。

DSMsync()はあくまでもメモリ同期機能の補助的な API であり、メモリアクセスの際の透過性は保持されている。特に書き込み側に関してはメモリ同期機能のためにプログラムを書き換える必要がまったくない。

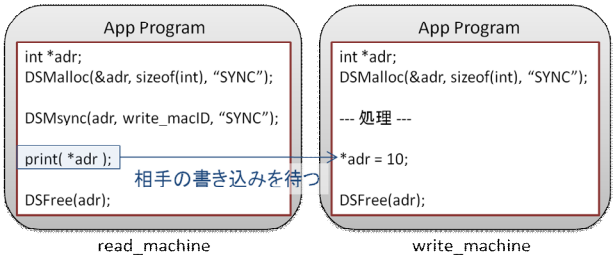

図3.メモリ同期のプログラム例

4.3 実装方式

同期を取る際もメモリアクセスの透過性を維持する方針のため、仮想アドレスのアクセス権を制御することで実装した。

①DSMsync()を呼び出した側の機器は指定した機器に対して同期開始を通知する。その後指定したメモリをアクセス禁止にする。このメモリを読み込もうとするとページフォールトが起きるのでその中で書き込み通知が来るまで処理を止めておく。

②同期開始の通知を受け取った機器は指定されたメモリを書き込み禁止にする。このメモリに書き込もうとするとページフォールトが起きるのでその中で書き込みが起きたことを元の機器に通知する。その後書き込みを許可する。

③書き込み通知を受け取った機器は禁止していたメモリへのアクセスを許可する。止められていた読み込み処理は再開され、相手の書き込んだ値を読むことが出来る。

5. まとめ

T-Kernel 分散共有メモリ機能への機能追加とメモリ転送の測定評価を行った。

本研究については T-Engine フォーラムが主催の TRONSHOW にて発表を行っている。また韓国で行われた組込みソフトウェアコンテストの国際部門にも出場した。

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