

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

山原 亨† 松原 裕人† 酒井 隆行† 大谷 真†

湘南工科大学†

1. はじめに

組み込み機器がユビキタス環境で使われるようになるに伴い、複数のユビキタス機器が連携動作するシステムが一般化すると予測される。これに対応すべく T-Kernel での透過的分散共有メモリ機能の研究開発を行っている[1]。本研究ではその一環として、LAN で接続された複数の T-Engine 間でページフォールト発生時に高速メモリ転送を行うデバイスドライバ RMAD (Remote Memory Access Driver) を開発した。この論文では RMAD の仕様と実現方式および検証結果について述べる。

2. 開発の前提と方針

RMAD の開発には次期組み込み OS として期待されている T-Kernel を用いた。ハードウェアは T-Kernel の標準開発プラットフォームである T-Engine ボードを用いる。LAN インターフェースは T-Engine の拡張ボード (TTZ101-002) を用いた。TTZ101-102 には ASIX 社製 LAN コントローラの AX88796 が使用されている。16kByte のバッファ RAM を送受信バッファとして使用できる。受信バッファはリング構造になっており、レジスタを用いてハードウェアで制御される。

RMAD はページフォールト発生中に分散共有メモリ制御部 (カーネルモード) から呼び出される。ページフォールト中は他の処理をブロックする必要がある。処理の高速化が必要なため、物理層とデータリンク層に関しては Ethernet 標準規格を用いるがその上位には専用のプロトコルを設けることにした。また、基本的な処理の流れをできるだけ単純にし、かつ要求が重なった特殊な状況に対応した処理を準備するようにした。

3. 外部仕様

RMAD の主要な機能は read() に実装した。read() を使用する際はクライアント側マシン (read() を呼び出した機器) とサーバ側マシン (読みたいメモリがある機器) に分かれて動作

する。read() に渡す引数を表 1 に示す。また read() の動作を図 1 に示す。

表 1. read() の引数

macID	サーバ側マシン ID
src	サーバ側メモリアドレス
dest	クライアント側メモリアドレス
size	転送サイズ

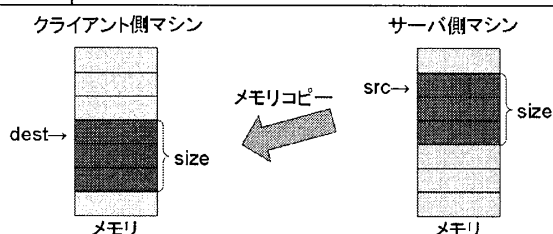


図 1. read() の動作

read() は macID でサーバ側マシンとなる機器を識別し、サーバ側マシン内の src で指定されたアドレスから size 分のメモリ内容を読み込み、クライアント側マシンの dest で指定されたアドレスに書き込む。

4. 制御方式

4.1 通信プロトコル

送受信時のデータフレーム形式を表 2 に示す。データフレームの最大長は 1500Byte のため、リード応答の data 領域は最大 1499Byte までである。

表 2. 送受信データフレーム

リード要求		
type	リード要求識別子	1Byte
macID	クライアント側マシン ID	1Byte
addr	サーバ側メモリアドレス	4Byte
size	転送サイズ	4Byte
リード応答		
type	リード応答識別子	1Byte
data	リード要求で指定したメモリ内容	1499Byte

4.2 基本的な処理の流れ

RMAD の基本的な処理の流れを図 2 に示す。

①クライアント側マシンが read() を呼び出すことでデータのやり取りが始まる。read() は引数の情報を元にリード要求をサーバ側マシンに送り、サーバ側マシンからリード応答が送られてくるまで割り込み禁止のままビジー状態で受信バッファを監視する。

Implementation of High-speed Memory Transfer for T-Kernel Distributed Shared Memory Feature

†Toru Yamahara, Hiroto Matsubara, Takayuki Sakai, Makoto Oya - Shonan Institute of Technology

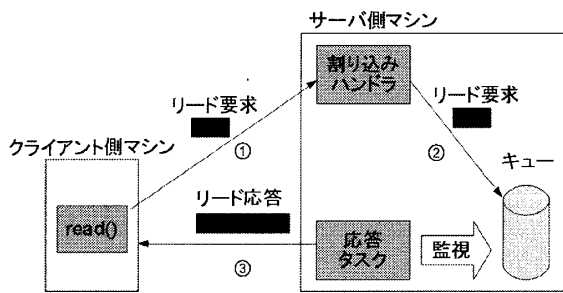


図 2. 基本的な処理の流れ

②サーバ側マシンはリード要求が送られてくると割り込みハンドラが起動する。割り込みハンドラは送られてきたリード要求をそのままキューに入れるだけで終了する。

③キューにデータが入るとキューを監視している応答タスクが起動する。このタスクはリード要求が指し示すメモリ内容をリード応答として送信する。リード要求で指定されたサイズが1500Byte 以上だった場合はメモリ内容を分割して複数のリード応答に分けて送る。リード応答を受け取ったクライアント側マシンは自分のメモリにリード応答内のデータを書き込んでいく。

高速化のためサーバ側マシンからのリード応答に対する受信の通知は行わず、size 分のメモリ転送が終了次第クライアント側もサーバ側もデータのやり取りを終了する。

read() 内の一連の動作は割り込み禁止状態でいき、read() の重複呼び出しを防ぐようにしている。重複して呼び出されると一度に多数のデータを受け取ることになり受信バッファがオーバーフローする可能性が大きくなる。それを防ぐための処置である。

4.3 特殊な状況

RMAD は複数機器間での通信を想定しているので、データのやり取り中に他の機器からリード要求が起きることがある。他の機器からリード要求が起きた場合の処理を次に示す。

(1) 応答中のサーバ側マシンに他の機器からリード要求が起きた場合

リード応答の送信を行う応答タスクは割り込み許可状態で動作している。応答処理中に他の機器からリード要求があった場合は、割り込みハンドラが起動しリード要求をキューに入れる。応答タスクは現在のデータのやり取りが終了した後にキューからリード要求を取り出してその応答を行う。

(2) read() 中のクライアント側マシンに他の機器からリード要求が起きた場合

クライアント側マシンは割り込み禁止状態でデータのやり取りを行うので、他の機器からリ

ード要求に対して割り込みは起こらない。しかし、割り込み禁止状態でも受信バッファにリード要求は入ってくるので、それを処理する必要がある。クライアント側マシンはリード要求を送信した後リングバッファを監視してリード応答を待つが、ここでリード要求が入ってくるとそのリード要求をキューに入れておく (図 3)。read() が終了した後、キューにデータが入っていると応答タスクを起動しキューからリード要求を取り出してその応答を行う。

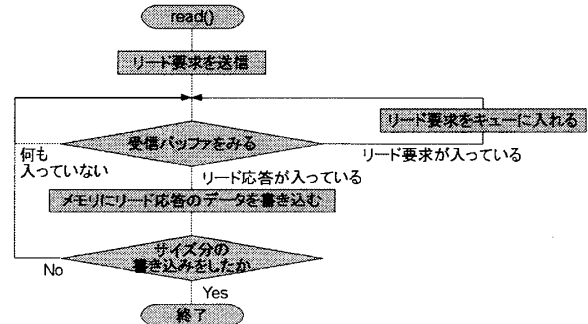


図 3. read() 内での処理

5. まとめ

read() で指定した転送サイズ (バイト) と read() に要す時間 (m 秒) の測定結果を図 4 に示す。グラフから 1 回の read() の時間 t (m 秒) は、 $t = 0.216 + 0.000294d$ となる。なお、 d は転送サイズである。

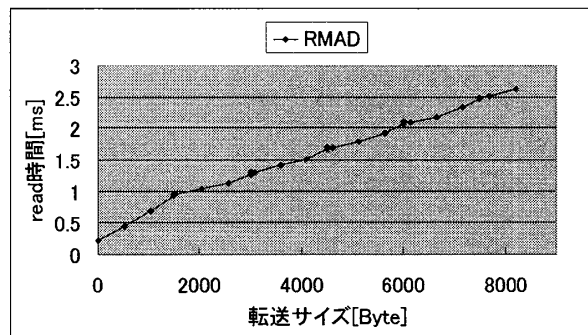


図 4. 処理時間の比較結果

初期オーバーヘッド (0.216) を除くと、転送速度は約 27Mbps であり、実験環境の LAN の最高速 100Mbps に対し 1/3 程度の性能は出ている。しかし初期オーバーヘッドを含めると、1 ページ (4k バイト) の転送に約 1.5m 秒掛かっており、ページフォールトの処理としては十分な性能とはいえない。今後、さらに性能向上が必要である。

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