

# 組み込み制御システム向け分散共有メモリ機構を持つリアルタイムOS A Real-Time Operating System with Distributed Shared Memory for Embedded Control Systems

田村宥士<sup>†</sup>  
Yuji Tamura

知場貴洋<sup>‡</sup>  
Takahiro Chiba

兪明連<sup>†</sup>  
Myungryun Yoo

横山孝典<sup>†</sup>  
Takanori Yokoyama

## 1. はじめに

自動車等の組み込み制御アプリケーションはマルチタスク構成で実装され、タスク間でメモリ上のデータを共有することが多い。分散型の組み込み制御システムにおいても、ノード間でデータを共有可能とする分散共有メモリを利用できれば、アプリケーションプログラムの書き換えを行うことなく分散化が可能となる。しかしこれまで提案されてきた分散共有メモリの多くは仮想記憶機構を利用したページベースのもので [1], リアルタイムシステムに適していない。また、自動車等の組み込み制御システムで使用されているプロセッサはMMU(Memory Management Unit)をもたないのが普通である。このため、MMUを必要とせず、リアルタイム性を保証できる組み込み制御システム向けの分散共有メモリが求められている。

そこで我々は、それらの要求を満たす分散リアルタイムOS (Real-Time Operating System, RTOS) に関する研究に着手した [2]。本分散RTOSは、自動車制御分野の標準RTOS仕様であるOSEK OS[3]をベースとし、ネットワークにTDMA(Time Division Multiple Access)プロトコルに基づくFlexRay[4]を用いる。

本論文では、本分散共有メモリ機構で採用したFlexRay上の転送順序に基づいてデータの一貫性を保証する手法を提案するとともに、その実装法について述べる。

## 2. 分散共有メモリの一貫性

### 2.1. 一貫性モデル

分散共有メモリでは、複数のアプリケーションが同時に共有データに読み書きする可能性があるため、各ノードで読み出す共有データの値は、特定の一貫性に従わなければならない。全ノードで同一順序でデータの書き込みを観測できる順序一貫性 [1][5] が望ましいが、効率的に実装することは難しい。そこで我々は、一定の通信サイクルで同期して通信を行うFlexRayの特徴を利用して、順序一貫性と等価な一貫性を保証できる分散共有メモリ機構を提案する。この一貫性を本論文では等価的順序一貫性と呼ぶ。また、より弱い一貫性として単一変数について全ノードで同順序で観測できる一貫性を提案する。これを本論文では部分的順序一貫性と呼ぶ。そして、これら2つの一貫性のうち、アプリケーションに適したものをユーザが選択できるようにする。

等価的順序一貫性は、FlexRay上での転送順序に基づいて一貫性を保証する。しかしFlexRayでは、コン

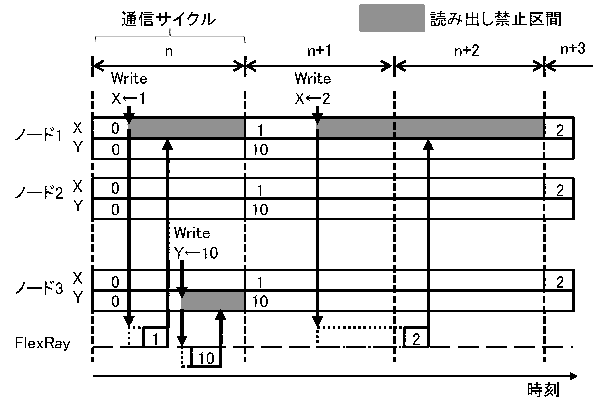


図 1: 等価的順序一貫性における動作例

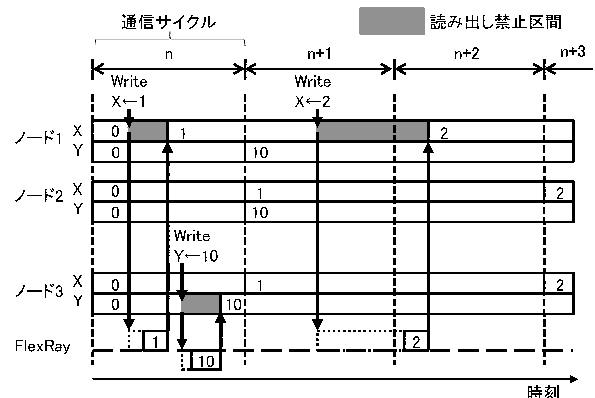


図 2: 部分的順序一貫性における動作例

ローラに送信要求を出した順序と実際にデータ転送が行われる順序は必ずしも一致しない。そこで、書き込み処理を行ったノードでは、書き込まれた値を即座に分散共有メモリに反映するのではなく、他ノードへのデータ転送が完了した時点で反映するとともに、一貫性を保証できるまで分散共有メモリの読み出しを禁止する。また、受信したノードでは、通信サイクル開始時に同期して受信した値を分散共有メモリに反映させる。以上により一貫性を実現する

### 2.2. 等価的順序一貫性

等価的順序一貫性の場合には、書き込みノードにおいて、分散共有メモリへの書き込みからデータ転送後の最初の通信サイクル開始時まで分散共有メモリからの読み出しを禁止する。複数のノードで分散共有メモリに値が書き込まれてから全ノードに値が反映されるま

<sup>†</sup>東京都市大学  
<sup>‡</sup>現在 株式会社セック

```

/*分散共有メモリ用リソース取得*/
GetResource(distmem_res);

/*分散共有メモリの値を取得*/
value = d_mem;

/*分散共有メモリに値を書き込み*/
d_mem = 10;

/*分散共有メモリ用リソース解放*/
ReleaseResource(distmem_res);

```

図 3: 分散共有メモリへのアクセス処理の記述例

での流れを図 1 に示す。X と Y は共有変数である。サイクル n の間にノード 1, ノード 3 で書き込まれた値はそのサイクルで転送されている。この場合サイクル n の間, ノード 1 では X の値が, ノード 3 では Y の値が読み出せないため, サイクル n+1 開始時に全ノードで同時に変数 X と Y の値が遷移したように観測される。また, サイクル n+1 の間にノード 1 で X に書き込まれた値は, 次のサイクル n+2 で転送されるため, サイクル n+3 開始時に全ノードで値が書き込まれたように観測される。通信量を考慮して FlexRay のコンフィギュレーションを行った場合, 送信の持ち越しは最悪でも次のサイクルまでなので, 読み出し禁止区間は最悪の場合で通信サイクル 2 周期分である。

### 2.3. 部分的順序一貫性

部分的順序一貫性の場合, 書き込みノードにおける読み出し禁止区間は, データ転送完了までとする。これにより, 等価的順序一貫性より待ち時間を短くできる。複数のノードで分散共有メモリに値が書き込まれてから全ノードに値が反映されるまでの流れを図 2 に示す。値の書き込みタイミングは全ノードで同一には観測されないが, X と Y それぞれの値の書き込み順序は各ノードで同一に観測される。

### 3. API と動作

OSEK OS では, 共有リソースの排他制御を行うための API として GetResource() と ReleaseResource() を提供している。そこで, これらの API を拡張して分散共有メモリへのアクセスに用いる。アプリケーションにおける分散共有メモリへのアクセス処理の記述例を図 3 に示す。distmem\_res は分散共有メモリ上の共有データを表すリソース, d\_mem は共有変数の名前である。アプリケーションは GetResource() を発行して distmem\_res を取得することで, 前述の一貫性に基づいた d\_mem の読み書きが可能となる。アプリケーションが ReleaseResource() を発行して distmem\_res を解放すると, 分散 RTOS の分散共有メモリ機構は d\_mem の値が GetResource() 発行前と異なっているか確認し, 異なっていた場合は, 書き込まれた値を全ノードに送信する。

### 4. 実装と評価

OSEK OS 仕様に基づく TOPPERS/OSEK カーネル [6] を拡張し, 本論文で提案した分散共有メモ

表 1: 拡張したシステムコールの処理時間

		GetResource()	ReleaseResource()
分散 RTOS	1Byte	3.1	10.5
	16Byte	4.3	11.7
オリジナル OSEK		2.4	2.7

単位 [μsec]

リ機構を有する分散 RTOS を実装した。ハードウェアは V850/PHO3 プロセッサを搭載した評価ボード GT201N10 を用いた。CPU のクロック周波数は 128MHz である。開発した分散共有メモリの性能を評価するため, 拡張した GetResource() と ReleaseResource() の処理時間を測定した。計測には 32MHz, すなわち 1 カウント 31.25nsec のハードウェアカウンタを用いた。データサイズが 1Byte 及び 16Byte の場合の処理時間を 100 回計測した平均値を表 1 に示す。一般に自動車制御アプリケーションタスクの周期は 10~100msec 程度であるため, 同一共有変数へのアプリケーションのアクセス間隔も同程度と想定できる。したがって, 処理時間の増加量は共有メモリへのアクセス頻度に対して十分小さい。また, FlexRay の典型的な通信サイクルは 1msec 程度であるため, 読み出し禁止区間も最悪の場合で 2msec 程度となり, タスクの周期に比べて十分小さく, 実用上問題ないと考えている。

### 5. おわりに

FlexRay ネットワークの特徴を利用し, 書き込みノードに分散共有メモリの読み出し禁止区間を設けることで, 実用的な性能で一貫性を保証することができる分散共有メモリを提案し, 分散 RTOS への実装及び評価を行い, その有用性を確認した。

今後, OIL(OSEK Implement Language)[7] を拡張して分散共有メモリに関する設定を記述可能とするとともに, 分散共有メモリ用のコンフィギュレーションデータを生成する SG(System Generator) を開発する予定である。

### 謝辞

本研究で使用した TOPPERS/OSEK カーネルの開発者に感謝する。本研究の一部は JSPS 科研費 24500046 の助成を受けたものである。

### 参考文献

- [1] Protic, J., Tomasevic, M. and Multinovic, V.: Distributed Shared Memory: Concepts and Systems, IEEE Parallel and Distributed Technology, Vol.4, No2, pp.63-71(1996)
- [2] 分散共有メモリ機構を持つ組み込み制御システム向け分散リアルタイム OS, 知場貴洋, 兪明連, 横山孝典, 情報処理学会研究報告, Vol.2013-EMB-28 No.3(2013)
- [3] OSEK/VDX Operating System Version 2.2.3(2005)
- [4] Makowitz, R. and Temple, C., FlexRay - A Communication Network for Automotive Control Systems, Proceedings of 2006 IEEE International Workshop on Factory Communication Systems, pp.207-212(2006)
- [5] Tanenbaum, A.S. and Van Steen, M.: Distributed Systems: Principles and Paradigms, Pearson Education(2002)
- [6] TOPPERS プロジェクト: TOPPERS/OSEK カーネル, <http://www.toppers.jp/osek-os.html>
- [7] OSEK/VDX: System Generation OIL: OSEK Implementation Language Version 2.5 (2004)