

OZ: 対象指向開放型分散システムアーキテクチャ

6H-4

— 分散カーネルのインタフェースと実装 —

塚本享治 (電総研)

近藤貴士 (シャープ)

中込昌吾 (ABC)

田中伸明 (松下電器)

吉江信夫 (住友電工)

1. はじめに

高度情報化社会の基盤技術として、異機種間相互接続とその応用についての研究が行われている。われわれは、この一環としてオブジェクト指向のネットワークアーキテクチャOZシステムを提案し、現在実証評価システムを構築中である。本稿では、OZシステムにおける分散カーネル部について、設計方針およびUNIX上での実装について報告する。

2. 分散カーネルの構成と機能

OZでは、実マシン上にいくつかの仮想マシンを実現し、これを論理的な実行単位としている。これをドメインとよぶ。OZプログラムは、ドメインの一種である実行系内で、オブジェクトがメッセージパッシング、あるいは移動などを行い処理が進行する。分散カーネルは、オブジェクトにその存在が意識されることはないが、実行系が処理を進行させる上で必要な機能を提供する。

2.1 構成

分散カーネルは、転送サービスインタフェースを提供するOTM(Object Transfer and Management)およびステーション管理からなる。ステーション管理は、プロトコル上ドメインと同位の実行単位で、自マシン上にあるドメインの管理(生成、削除など)を行う。OTMはオブジェクト、データ構造の転送を行うもので、ASN.1エンコーディングにより相互運用性を実現している。

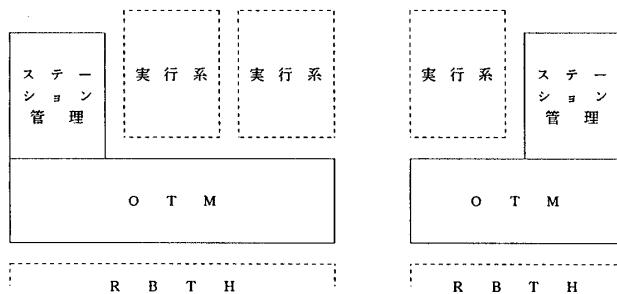


図1. 分散カーネルの構成

2.2 サービス

分散カーネルは、実行系ならびにサーバ系(以後これらを分散カーネルサービス利用者あるいは単に利用者と呼ぶ)に対しサービスを提供する。サービスは次のように分類される。

- (1)ドメイン管理サービス
- (2)オブジェクト転送サービス
- (3)DIBサービス
- (4)ファイルサーバサービス

(3),(4)は本来それぞれDIB、ファイルサーバの機能で、分散カーネルは操作インタフェースのみ提供する。

2.3 分散不可視性

分散カーネルは、ドメイン、オブジェクトをIDにより管理している。ドメインIDおよびオブジェクトIDはシステム全体で一意である。分散カーネルの利用者は、IDによってサービス対象を指定でき、基本的には所在情報を知る必要がなく、ドメイン、オブジェクトの分散不可視性を実現している。

また、分散カーネルは、OZシステムを構成するネットワーク上のマシンにそれぞれ存在するが、分散を特に意識したサービスを除き、利用者から見ると1つの統合したものとして見える。

3. 対実行系インタフェース

実行系においては、すべてのオブジェクト間メッセージパッシングはOZ言語によって次のように記述する。

```
dest_obj:message(arg1,arg2,...,argn);
```

ここで、dest_objはメッセージmessageのあて先オブジェクトであり、argは引数オブジェクトである。

実行系が分散カーネルのサービスによってメッセージを送る場合、次の処理が順に実行される。

①実行系のインタプリタはメッセージパッシングを検出したとき、オブジェクト表を調べ、dest_objが自ドメイン内のものであるかオブジェクト表を用いて判断する。外部にあると判断した場合、分散カーネルに対し転送要求が発生する。

②OTMでは、オブジェクトのストリーム化処理を行い、

OZ: Object Oriented Open Distributed System Architecture --Interface and Implementation of Distributed Kernel-- Michiharu TUKAMOTO⁽¹⁾, Takashi KONDO⁽²⁾, Shogo NAKAGOME⁽³⁾, Nobuaki TANAKA⁽⁴⁾, Nobuo YOSHIE⁽⁵⁾

(1)Electrotechnical Laboratory (2)SHARP Corporation (3)ABC Co.,Ltd.

(4)MATSUSHITA Electric Industrial Co.,Ltd. (5)SUMITOMO Electric Industries,Ltd.

あて先オブジェクトが存在するドメインを調べ、オブジェクト転送要求インタフェースを呼び出す。

③OTMはPCIの付加なエンコード処理を行い、プロトコルデータをRBTH部のバルクデータ転送サービスを用いて転送する。

④プロトコルデータを受信したOTMはデコード処理を行い、オブジェクト転送通知インタフェースを呼ぶ。

⑤ここでストリーム化されたオブジェクトを再構成し、messageをあて先オブジェクトへ渡す。

OZでは、オブジェクトが動的に移動できるため、あて先オブジェクトが移動してしまうことがあるが、RBTHのヒンティング処理によりあて先オブジェクトのあるドメインへ転送される。

分散カーネルが提供するサービスは、以上のようにはOTMのみで実現されるものと、OTMを利用しステーション管理との相互作用により実現されるものがある。

分散不可視性実現のため、マシン名→ステーション管理IDなどの対応関係をOTMが利用する。この情報はディレクトリに格納されている。

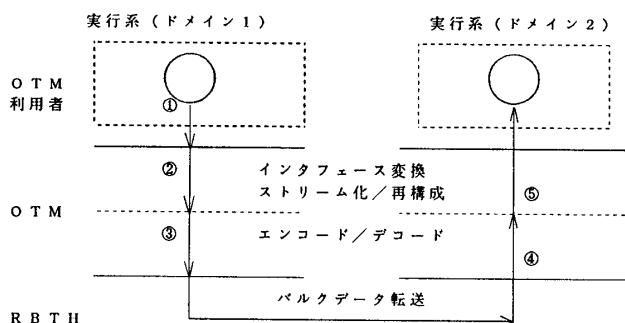


図2. オブジェクトの転送

4. 対LANインタフェース

RBTHは、コネクションレス型ACSEに基づく高速高信頼性バルクデータ転送機能を提供する。分散カーネルはこのバルクデータ転送機能サービスを利用する。RBTHは、送信要求を受けると、あて先が自マシン上にあるか判定する。自マシンの場合マシン内でデータをそのまま渡し、他マシンの場合下位層のデータ転送サービスを用いる。

5. 実装

現在、UNIX SystemV Release 2上にOZシステムを構築中である。ドメインはUNIXプロセスに対応させている。

ステーション管理は、C言語で記述されている。OTMは、C言語のユーザプログラムとして実装されており、実行系に対しC言語インタフェースを見ている。RBTHはドライバとして実装されている。

ドメイン、特に実行系が停止する時間がないようwait(sleep)は極力回避する。

5. 1 送信処理

OZでは、コネクションレスプロトコルを用いており、実行系が通信期間中停止するのを防ぐため、送信要求後できる限りすみやかに呼び出し元へ復帰する。再送が必要な場合は受信デーモンが代行する。

5. 2 受信処理

受信処理は受信デーモンによって行われる。データを受信すると、受信デーモンが実行系などに対しトラップワードなるフラグを用いて通知を行う。実行系に対し非同期な通知は行わない。受信データは共有メモリにおかれる。

5. 3 メモリ管理

UNIXカーネルからは直接仮想メモリを利用するのが困難である。そこで、共有メモリが一杯になると受信処理中のUNIXプロセスは、ユーザメモリへの吸い上げを行う。

5. 4 優先通知

実行系がガーベジコレクション中などの場合、ある特定の通知のみを優先的に処理し他の通知は後に回す必要がある。OTMにこのための通知キューイング機構を設ける。

6. 評価・検討

OTMサービスインタフェースで、ストリーム化されたオブジェクトのドメイン間転送時間を対向ループバックテストによって測定した。

条件：ストリームサイズ 80bytes, 往復, 実時間
U-station E20 および E30

- ・E20→E20間転送(同一マシン) 15 mS
- ・E30→E30間転送(同一マシン) 10 mS
- ・E20→E30間転送(異なるマシン) 82 mS

別途測定した下位層のみの転送処理時間が、70mSほどかかっていることから、下位層の通信がボトルネックになっているといえる。ただし、ストリームサイズが大きい場合、下位層プロトコルの性質上、上記の値よりスループットは良くなる。

なお、実用上必要なスループットは、実行系においてOZ言語のプログラムを実行し、評価せねばならない。

7. まとめ

分散カーネルが提供するサービス、インタフェースと実装法について記述した。

[参考文献]

- 1) 塚本 他, "OZ:対象指向開放型分散システム7-キタチ", 第36回情報大全, pp. 633-642 (1988)
- 2) M. Tsukamoto et al, "The Architecture of OZ: Object-Oriented Open Distributed System", Proc. of 2nd ISIIS, pp. 153-166 (1988)