

1S-4 制御用ソフトウェア開発実行環境 Objectpowerにおける分散オブジェクト管理方式

大柴正清、石川健一、安藤博文、金子淳一

富士ファコム制御株式会社

1. はじめに

分散オブジェクト実行環境をP A、F Aなどの制御向けで使用する場合には、実時間性と信頼性の問題を解決しなければならない。われわれは特別なハードウェアを使用せずにこの問題を解決する必要があった。しかし、既存のオブジェクト指向技術では、上述の問題を満足できなかったため、Objectpowerでその機能を開発した。本紙ではこの開発で採用した技術を紹介する。

2. 分散環境モデル

Objectpowerの分散オブジェクト環境を図1に示す。オブジェクト間の通信は、一般的な形式で実装し、①NISを基本としたネーミングサービスによりオブジェクトの存在するホストを探し出し、②探し出したオブジェクトに対して専用の通信デーモンを中継してメッセージを送っている（図1）。

3. 実時間性の向上

実時間性の向上を目的として、ホスト間の通信量を削減させるために採用した方法を以下に

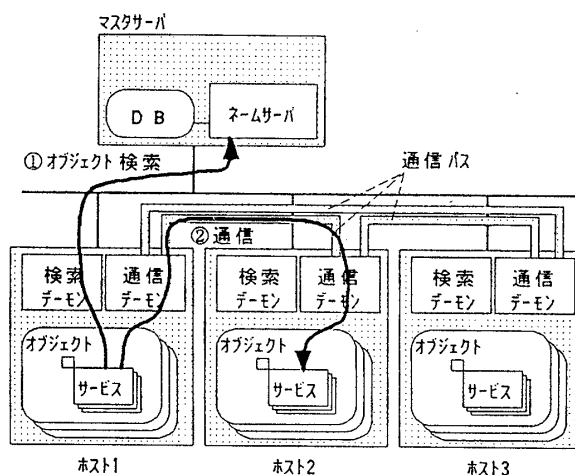


図1 分散環境モデル

説明する。

3. 1 ネームバッファによる高速化

一般的なネームサービスでは、問い合わせのたびにネームサーバへのアクセスが発生する。

この実装では1回アクセスしてネームサーバから得た情報は各ホストのバッファメモリに保存し、2回目以降の検索ではネームサーバへの問い合わせを行わないようにしている。これによりネットワークアクセス回数の削減による性能向上を実現している。

また、オブジェクトの登録・削除時には、ネームサーバから各ホストにバッファメモリの更新を依頼し、バッファメモリとネームサーバのDBの一貫性を維持できるようにしている。

3. 2 通信バスの固定化

T C P / I Pのような一般的なプロトコルを使用する場合、オブジェクト間のメッセージ通

Distributed object management method of Objectpower: Control software exective environment

Masakiyo Ohshiba, Kenichi Ishikawa, Hirofumi Ando, Jyunichi Kaneko

Fujifacom Corporation

1-2-2, Manpukuji, Asao-ku, Kawasaki-shi, Kanagawa 215, Japan

信のためにクライアントオブジェクトとサーバオブジェクト間に専用の通信パスを作成しなければならない。しかし、オブジェクトが数百個もあると膨大な数の通信パスが必要となる。また、通信のたびに通信パスを作成、クローズすると性能が問題になる。

この実装では、異なるホストのオブジェクト間のメッセージ通信はかならず通信デーモン経由にしている。通信する可能性のあるホストの通信デーモン同志には事前に双方向の通信パスを作成しておくことにより、メッセージ通信時の性能の確保と通信パス資源の有効利用の両方が実現できる。

3.3 非同期での通信

制御用ソフトウェアでは、クライアント・サーバ間で同期をとったRPCを実行するのではなく、ひとつの処理のターンアラウンドタイムを最小にするためにプログラムは可能な限り並列に動作させ、同期が必要な場合はアプリケーションプログラムの設計者がピア・トゥ・ピア通信で同期をとるという考え方が一般的である。この実装では、制御用ソフトウェアのそうした従来からの設計手法を利用できるようにするために、非同期でメソッド（Objectframeではサービスと呼んでいる）を実行できる機能を付加している。これはまた、分散されたオブジェクト間のメッセージ通信においても、①伝送時間を半減できる②サーバホストの高負荷時にも処理時間の遅延などの影響を受けにくいという副次効果を得ることができる。

4 信頼性の向上

信頼性（可用性）の向上を目的として、ホスト間の通信異常をすみやかに検出するために採用した方法を以下に説明する。

4.1 通信パスの監視

ホスト間のメッセージ通信を管理する通信デーモンは、通信パスの状態の変化により通信相手のホストまたはネットワークの異常を検出することができる。しかし、実際にメッセージ通信を行っていない場合には、状態の変化が発生してから異常を検出するまでに数分のオーダのタイムラグを生ずるのが通常である。

このタイムラグを最小にするために、本実装では利用者の指定した時間間隔で相手先のホストとの通信の可否を確認するメッセージを送ることができる。異常の検出時には、利用者のオブジェクトにこれを通知している。

5. おわりに

Objectframeの実システムでの運用による評価を参考にして今後の開発を方向づける予定である。現時点では以下の点が今後の検討課題であると考えている。

①ネームサービスの共用DBへの変更

制御用として考えると、ネームサーバの多重化、ホスト毎の自立性の向上に対応する必要がある。オブジェクト情報はネームサービスに登録するのではなく、（レプリカ付きの）DBに登録することを検討する予定である。

②異種プロトコルとの共存

Objectframeに閉じたオブジェクト間の通信のみでなく、異種ネームサービス、ORBとの相互接続の方式を検討してゆく予定である。

6. 参考文献

- [1] 清水謙多郎：分散・並列OSの機能と実現技術，トリケップス，1992