

# 異種OS相互運用方式の検討 — 実験システムと評価 —

3T-2

末広 亮太 本田 邦夫  
(松下電器産業(株) 東京研究所)

## 1. はじめに

ネットワークに接続された異機種 of 計算機が管理する資源の相互運用を行うため、各種OSのシステムコールを特定の機種、特定のOSに依存しない共通性の高い「システムコールホット」という概念を基本にした方式について、その概要を発表した<sup>[1]</sup>。

本稿では、この基本処理方式を基に、資源をファイルシステムにしたクライアント/サーバモデルの実験システムを構築し、基本処理方式の評価を行ったので、その結果について報告する。

## 2. 実験システムの構成

システムコールホットによる異種OS資源の相互運用方式を評価するために構築したクライアント/サーバモデルによる実験システムを図1に示す。この実験システムの各部の処理について説明する。

### 2.1 クライアントシステム

- (1) リクエスト ... コマンドや、アプリケーションプログラムより発行されたシステムコールを解析し、対象がローカルシステムか、リモートシステムのファイル名かをファイル名とそのファイルの所在情報から判断し、ローカルOSもしくは、トランスレータへシステムコールを渡す。
- (2) トランスレータ ... リクエストより受け取ったシステムコールを共通表現形式であるシステムコールホットに変換し、CNIFに渡す。
- (3) CNIF ... クライアントシステム内の通信機能を利用して、サーバシステムとの間におけるビットフレームの送信及び受信処理を行う。

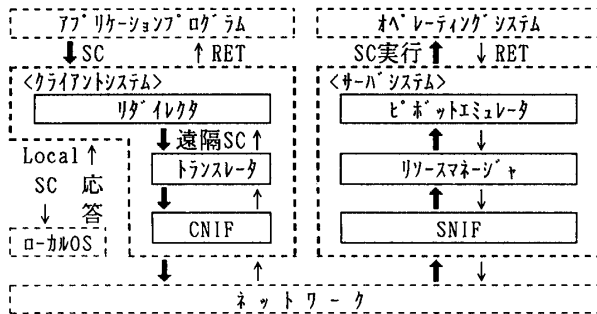


図1 クライアント/サーバモデルによる実験システム

### 2.2 サーバシステム

- (1) SNIF ... サーバシステム内の通信機能を利用して、クライアントシステムとの間におけるビットフレームの送信及び受信処理を行う。
- (2) リソースマネージャ ... サーバシステム内における資源管理を実現するため、資源に対応したクラス/インスタンスオブジェクトを管理する処理及び受信した資源アクセス要求処理をエミュレータに依頼する処理を行う。
- (3) エミュレータ ... リソースマネージャから受け取った資源アクセス要求である受信ビットホットを、サーバ側のOS上で実行可能なシステムコールとマッピングをとりながら依頼された処理を実行し、その応答を応答ビットホットとしてクライアントシステムに返す処理を行う。

## 3. 性能評価実験

実験システムにおける基本処理方式の性能を評価するため、

以下の実験及び評価を行った。

### 3.1 リモートアクセス時及びローカルアクセス時の性能測定、比較評価

リモートアクセス時及びローカルアクセス時の性能測定結果を表1に示す。1番のケースは、ローカルアクセスに対してリモートアクセスの性能は同程度であるが、2番のケースでは、リモートのファイルへのアクセスの方が処理時間は短いという結果が出た。また、read/write系システムコールでは、ローカルアクセスに対してのリモートアクセスの性能は約50%セト向上するという結果が出た。これらの原因は、クライアント側のマシソに対するサーバ側のマシソのもつ性能の差によるものと考えられる。

表1 システムコール処理時間比較表1

システムコール	リモートアクセス時の処理時間 ローカルアクセス時の処理時間
1 open -> close	0.93
2 create -> close	0.75
3 read(100byte)	0.40
4 write(100byte)	0.17

### 3.2 システム稼働時及びシステム非稼働時の性能測定、比較評価

実験システム稼働時及び実験システム非稼働時の性能比較評価結果を、表2に示す。これは、実験システムを稼働することによるオーバーヘッドの割合を示したもので、この結果から、実験システムの稼働によるオーバーヘッドは1番のケースで最大7%セト程度であるという結果が出た。このオーバーヘッドは、システムコール発行時に、そのシステムコールが、ファイルアクセスのものかどうかを判断し、ファイルアクセスの場合、パス名をもとにリモート/ローカルのリクエスト処理を行うためである。また、read/write系システムコールでは、実験システムの稼働によるオーバーヘッドはほとんどないという結果が出た。

表2 システムコール処理時間比較表2

システムコール	システム非稼働時の処理時間 システム稼働時の処理時間
1 open -> close	1.07
2 create -> close	1.03
3 read(100byte)	0.97
4 write(100byte)	0.99

## 4. まとめ

本実験システムを用いての性能評価実験を行った結果より、本方式の機能面及び性能面での有効性が確認された。今後は、アプリケーションインタフェースに着目し、リモートプロシージャコールを用いた分散処理に関する手続きのインタフェースプログラムを自動生成する技術の開発を行う予定である。なお、この研究は新エネルギー・産業技術総合開発機構(NEDO)が出資を受けて実施した、通商産業省工業技術院の大型プロジェクト「電子計算機相互運用データベースシステム」の研究開発の成果である。

## 5. 参考文献

- [1]末広 本田: "異種OS相互運用方式の検討-基本処理方式-", 情報処理学会第39回全国大会講演論文集