

統合ファイル転送システムの構築法

1 P-1

小川 恒治

(株) 富士通神戸エンジニアリング

1. はじめに

最近のコンピュータ業界には、ダウンサイジング、オープン化の波が押し寄せている。このため、アプリケーションのマルチプラットフォーム展開は必須である。

しかし、OSの標準化に比べ、ネットワークはOSI, TCP/IP, 独自プロトコル等が並存する状況であるため、分散処理における必須機能の一つであるファイル転送においても、FTAM (OSI), FTP (TCP/IP), 独自プロトコル等が存在している。このため、マルチプラットフォーム展開における、分散処理アプリケーション向けのネットワーク無依存の標準ファイル転送基盤が必要とされている。

本稿では、分散処理アプリケーションのマルチプラットフォーム展開におけるファイル転送の問題点と転送基盤の構築について紹介する。

2. マルチプラットフォーム展開におけるファイル転送の問題点

分散処理に不可欠である機能の一つに、ファイル転送機能がある。しかし、現在、OSIにおけるFTAM, TCP/IPにおけるFTP等の転送プロトコルが統一されておらず、また、転送プロトコルによりサポートしている機能が違っている。このため、ファイル転送を利用する分散処理アプリケーションは、ネットワークに応じた転送プロトコルを使い分け、また、機能レベルに差異があるため補完する必要がある。さらに、各プラットフォームごとにファイル転送システムのインタフェースが異なるため、それぞれのシステム用に分散処理アプリケーションを作成する必要がある。

これらのファイル転送における問題が、分散処理アプリケーションのマルチプラットフォーム展開の妨げとなっている。

3. ファイル転送システムの標準化

3.1 問題点

図1に現状のファイル転送システムの形態を示す。そして、分散アプリケーションを構築し、動作させる場合の問題点を示す。

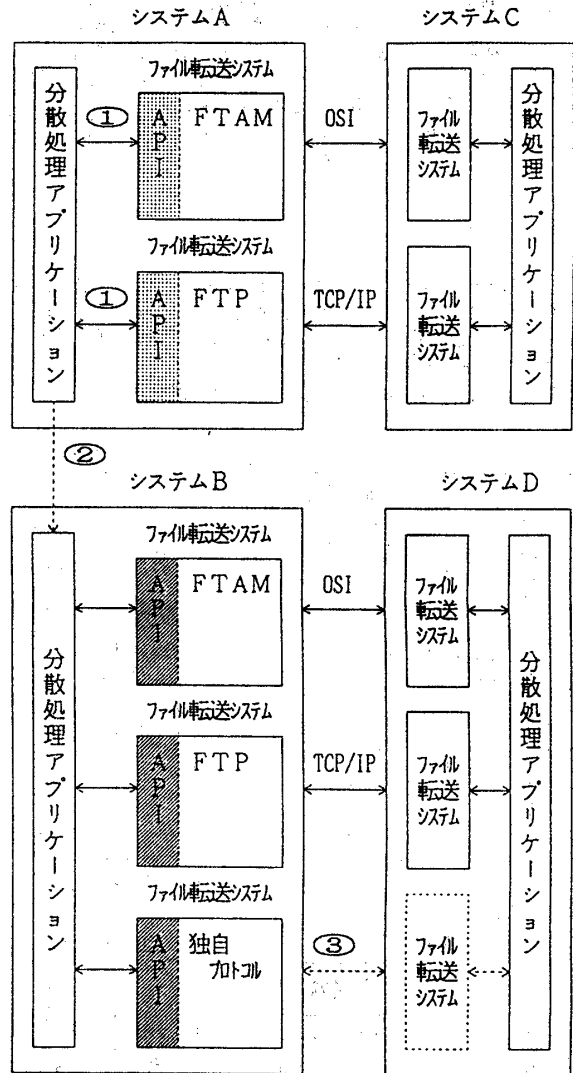


図1 現状のファイル転送システムの形態

- ① 分散処理アプリケーションがOSI系ネットワークサポート機能を実現させるために、FTAMのファイル転送システムを使用する。また、TCP/IP系ネットワークサポート機能を実現させるために、FTPのファイル転送システムを使用する。

このように、複数のネットワークをサポートしていることから、分散処理アプリケーションを構築するために、複数の転送プロトコルを使い分ける場合がある。また、それぞれの転送プロトコルに機能差が存在するため、補完する必要がある。

- ② システムAで動作している分散処理アプリケーションをシステムBで動作させようとする。このとき、システムBにシステムAと同機能を持つファイル転送システムが存在していても、システムAとシステムBのそれぞれのファイル転送システムのインタフェースが違うため、分散処理アプリケーションはそのままでは動作しない。
このように、システムBのインタフェースに合った形式で、分散処理アプリケーションを構築する必要がある。
- ③ 分散処理アプリケーションが機能を実現させるために、独自プロトコルのファイル転送システムを使用する。しかし、相手側となるシステムDに独自プロトコルをサポートしているファイル転送システムが存在しないと分散処理アプリケーションの機能が実現できない。
このように、接続する相手システムごとに転送プロトコルを使い分ける場合がある。

3. 2 解決方法

現状の転送プロトコルの機能差異の吸収、ファイル転送システムのインタフェースの標準化を実現したネットワーク無依存なファイル転送機能(統合ファイル転送システム)を、各プラットフォームに構築する。

図2に、統合ファイル転送システムの形態を示す。

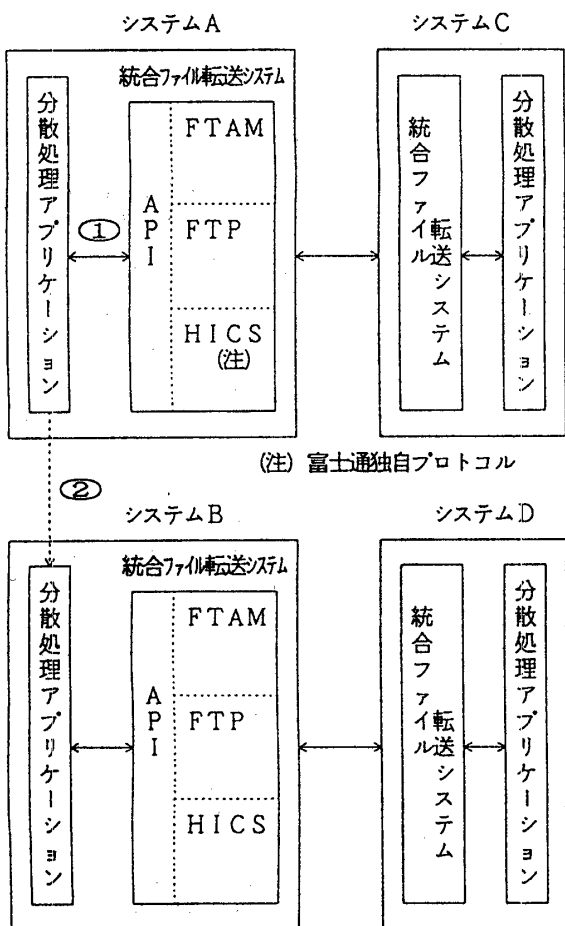


図2 統合ファイル転送システムの形態

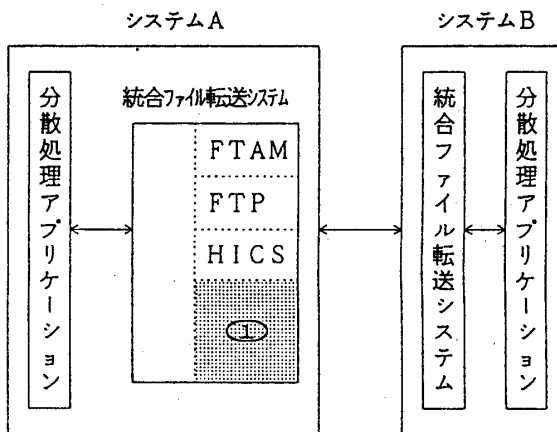
- ① 転送プロトコルの機能差異を吸収しているため、FTAM、FTP等の転送プロトコルを使用した分散処理アプリケーションの機能を実現できるようになる。
このように、ネットワークごとに転送プロトコルを使い分ける必要がなくなる。

- ② インタフェースを標準化しているため、システムAで動作した分散処理アプリケーションは、システムBでも動作することになる。
- ③ 転送プロトコルの機能差異を吸収しているため、FTPで実現しようとした分散アプリケーションの機能を、相手側となるシステムDにFTPをサポートしているデータ転送システムが存在しなくても実現できる。

4. まとめ

転送プロトコルごとに差異がある機能を統一し、かつ、インタフェースを規定することで、分散アプリケーションはネットワーク無依存に作成できる。この統合ファイル転送システムを各プラットフォームに展開することにより、分散処理アプリケーションもマルチプラットフォーム展開が容易になる。

今後の課題としては、現在のサポート外の転送プロトコルを任意に追加できる機構を検討し、利用者が転送プロトコルの追加を容易に行えるようにする方法を検討する。



① ユーザ追加プロトコル