分散ファイルシステムPLUSの設計と実装

亀井 仁志 (最所研究室)

あらまし

分散ファイルシステム PLUS は一つの論理ファイルを分割して複数のサーバ上にファイルとして配置することにより、巨大なストレージ空間をユーザへ提供する。本研究では分散ファイルシステム PLUS ついて提案し、設計と実装について述べる。

1 はじめに

ファイルのアクセス中にストレージ空間が一杯になり利用できなくなることがある。従来のファイルシステムでは図 1 の左図に示すように、ストレージ空間の不足に対してディスク装置を付加し、それをマウントすることにより空間を拡張していた。これに対して、本研究で提案する PLUS システムでは、図 1 の右図に示すようにディスクに空きのあるサーバがその領域を公開ストレージとして提供し、それをクライアントが動的に検索し利用することにより、ディスク空間を拡張する。このことによってクライアントは巨大なファイルを扱うことができるとともに、ディスクが一杯になることよる障害を防ぐことができる。

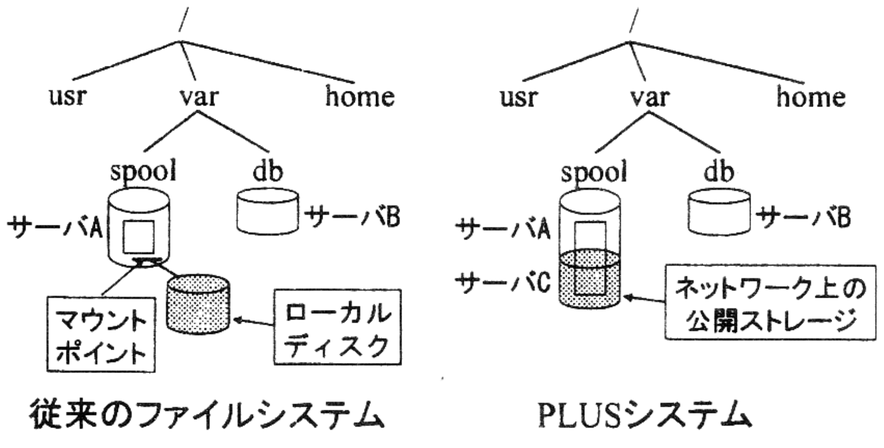

図 1: ストレージ空間の拡張

2 PLUS システムの概要

PLUS システムは 3 つのサーバとインタフェース関数で構成されており、アプリケーションから PLUS システムへの要求はインタフェース関数を通して行われる。

2.1 仮想ファイルと実ファイル

PLUS システムでは一つのファイルは図 2 に示すように、分割され複数のディスクサーバ上に配置される。それらを繋げて大きなファイルとしてクライアントへ提供する。

2.2 サーバ構成とストレージ空間管理

PLUS システムは図 3 で示される 3 つのサーバで構成される。それぞれのサーバは、1)PLUS システムとのインタフェースとなるサーバ (PLUS client agent)、2)PLUS システムを利用するクライアントを管理するサーバ (PLUS client manager)、3) 公開ストレージを提供し、管理するサーバ (PLUS server agent) である。

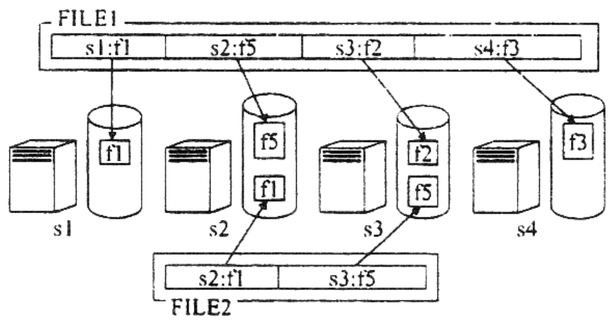

図 2: 仮想ファイルと実ファイル

PLUS システムでは、PLUS client agent から PLUS client manager へ仮想ファイルの操作が要求されるので、この組を仮想ファイル操作系と呼ぶ。また、PLUS client manager は仮想ファイルへの要求を実ファイルへ変換し、実ファイルを操作する要求を PLUS server agent へ行う。この組は実ファイルへの操作を行うので実ファイル操作系と呼ぶ。PLUS client manager は仮想ファイルへの操作を実ファイルへの操作へ変換する。具体的には仮想ファイルと実ファイルとの対応を管理する。

2.3 インタフェース関数

インタフェース関数は PLUS システムを利用するための関数で、現在、実装とデバッグの容易さからライブラリとして実装している。ファイルシステムを操作するシステムコールと 1 対 1 に対応させ、引数と返り値もシステムコールに合わせた。これにより、システムコールを対応するインタフェース関数に変更することで PLUS システムを利用することができる。

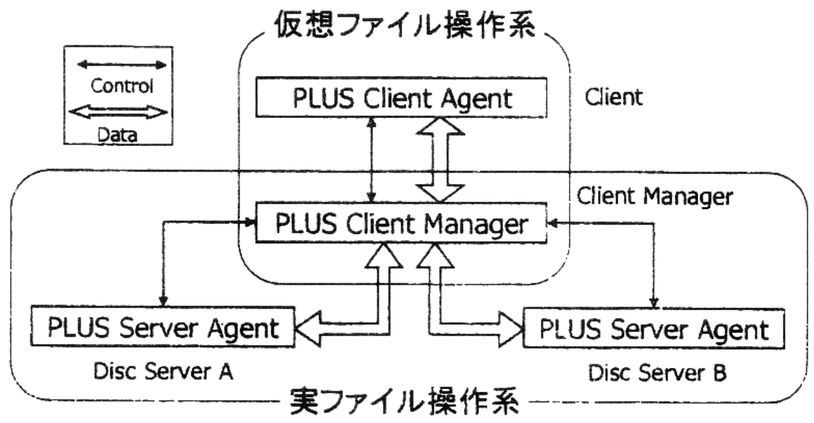

図 3: PLUS システムのサーバ構成

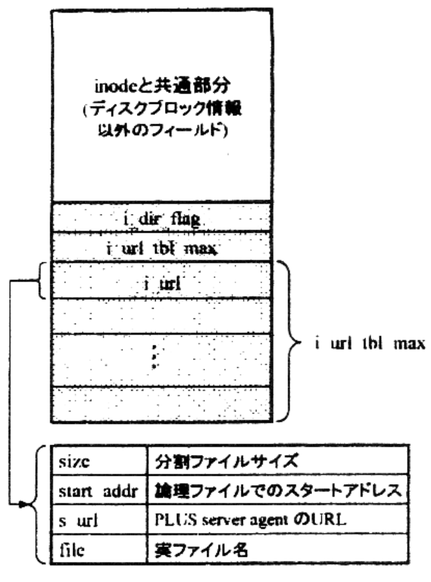

図 4: pinode の構造

3 PLUS システムの設計と実装

3.1 仮想ファイルとディレクトリ空間の管理

2.1 節で述べたように、PLUS システムでは仮想ファイルを分割ファイルで構成し 1 つの大きなファイルとして管理している。PLUS システムでは仮想ファイルを管理するための情報を pinode と称するファイルに保存している。pinode は PLUS client manager が管理している。

3.2 pinode

pinode を構成するレコードは UNIX 系のファイルシステム管理の中心である inode を拡張している。pinode のレコードは inode と共通の部分と PLUS システム用に拡張された部分で構成されている。図 4 に pinode の構造を示す。

3.3 pinode の管理

クライアントマシンは一意の IP アドレスが割り当てられているので PLUS システムでは IP アドレスを元にクライアントを管理している。pinode は PLUS client manager へ接続してきたクライアントの IP アドレスで管理されている。PLUS client manager は IP アドレスでディレクトリを作成し、その下に pinode の管理用ディレクトリを作成する。クライアントが PLUS client manager へファイル要求を行った場合、IP アドレスから管理ディレクトリを検索し対象ファイルの pinode を検索する。図 5 にファイル管理のディレクトリ構造を示す。

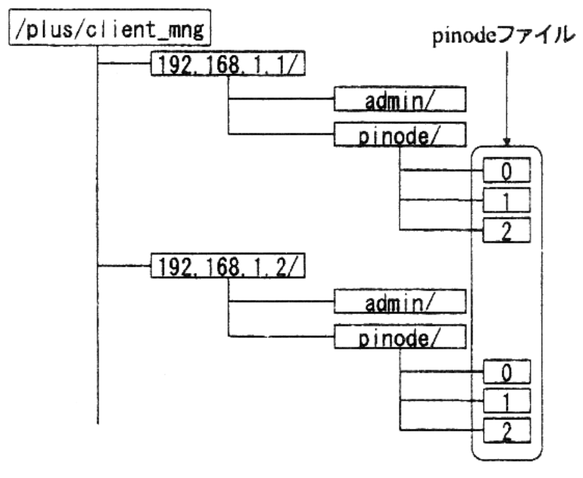

図 5: pinode の管理構造

3.4 オープン情報の管理

PLUS システムのインタフェース関数はライブラリで実装されているため、本来のファイルシステムで管理するオープン情報をシステムで管理しなければならない。また、PLUS システムはファイルディスクリプタをクライアントへ発行する際に、OS が返すファイルディスクリプタと重ならないように割り当てる。

3.5 仮想ファイルへの書き込み

仮想ファイルへの書き込みは PLUS システムの主要な部分であり、追記は書き込み判定や新規サーバ検索などの PLUS システムのもっとも重要な機能である。仮想ファイルは複数のディスクサーバ上へ分割されたファイルの集合であり、仮想ファイルへの書き込みは上書きと追記でシステムの動作が異なる。

3.5.1 上書き

上書きはすでに書かれている領域への書き込みである。クライアントが指定した書き込み位置から、対応する実ファイルを pinode から検索し書き込みを行う。

3.5.2 追記

追記の場合、書き込みサーバは pinode の URL レコードの最後のサーバとなる。このため、書き込み要求を行ったときにディスク領域が一杯である可能性もある。一杯の場合は新規サーバを検索する。

新規ファイル作成は追記書き込みと同様に考えることができるが、書き込みサーバが特定できない。このため、PLUS client manager は起動時に設定ファイルから最初に要求を出すマスタサーバを取得する。そして、マスタサーバを新規作成時の初期サーバとする。

3.6 仮想ファイルの読み込み

仮想ファイルの読み込みはクライアントが指定した位置から読みこむため、pinode の URL レコードを検索して読み込みサーバを特定する。読み込みサイズが大きく、読み込みサーバが複数にまたがる場合は、サーバごとに読みこんでそれらをまとめてクライアントへ渡す。この動作は上書きとほぼ同じである。

4 まとめ

本研究で設計し実装を行った PLUS システムの実装は初期段階である。現在のところ、PLUS システムを使った分散書き込みと読み込みは可能である。しかし、ファイルシステムとして必要な機能が実装されていない部分が多くある。また、実装の試験は簡単なコピーコマンドを用いて行った程度で不十分である。その際、ファイル操作のオーバーヘッドが大きいことが分かった。今後は、1) 詳細な性能評価を行いオーバーヘッドの原因を調査し性能向上を計る、2) サーバダウンへの対策等の設計を行い、信頼性の向上させる。