分散ファイル群高度管理のためのミドルウェアの開発*

三森祐一郎 (学籍番号 200721560)
研究指導教員: 森嶋厚行
副研究指導教員: 杉本重雄

1. はじめに

近年，計算機による情報処理が日常のものになり，計算機が格納するデータ量は飛躍的に増大している．また，コンピュータネットワークや各種デバイスの発達により，これらのデータは複数の機械に分散して格納されていることが一般的である．例えば，大学の小規模な研究室でも，ファイルサーバに格納されているファイル数が数十万を越える事は珍しくない．また，それらのファイルのコピーや関連ファイルはファイルサーバ内に留まらず，研究室の各構成員のノート PC をはじめとして，様々な機器に分散して格納されている．その結果，例えば，あるプロジェクトに関連するファイルがどこに散らばっているのか，あるファイルの最新バージョンはどれか，など，計算機に格納されているファイル群の管理はますます困難になっている．

このようなファイル群の管理において大きな手掛かりとなるのが，ファイル間の関連である．ファイル空間には，ファイルのコピーや参照関係などといったファイル間の関連が多数存在している．これらの関連は，１つの計算機に格納されているファイル間にとどまらず，ネットワーク内の他の計算機上のファイルとの間にも存在する．しかし，このような関連に関する情報は既存のファイルシステムでは明示的に扱われていないため，利用することができない．そのため，ファイル間の関連を利用した既存のアプリケーション [1][2] では，特定の関連を計算・利用するための機能を個別に実現しているのが現状である．関連を利用したこのようなアプリケーションの開発コストは一般的に大きく，既存の関連を利用したアプリケーションの上に構築するという方法も，それら既存のアプリケーションの多くが関連発見とその応用を一体としているため困難である．

そこで本論文では，このようなファイル間の関連を容易に利用するための基盤となるミドルウェアの開発を行う．本ミドルウェアは，分散ファイル環境において，異なる関連を統一的に発見・利用可能とすることで，関連を利用したアプリケーションの関連発見部分の開発コストを低減し，開発を容易にすることを目的としている．

2. コミュニティ情報空間ガバナンスプロジェクト

本研究で開発するミドルウェアは，**コミュニティ情報空間ガバナンスプロジェクト**の一部として推進したものである．本プロジェクトでは，クライアント PC やファイルサーバに格納されている多量のファイル群の管理を行うフレームワークである **InfoSpace Governor** の構築を目指している．

本フレームワークでは，個々のファイルシステムではなく，**コミュニティ情報空間** (Community Information Spaces, CIS) と呼ぶ複数の構成員とファイルシステムを含む空間を管理の範囲としており，既存のファイルシステムでは明示的に扱われていない，ファイル間の関連といったメタデータを保持することにより，情報空間の管理を行う．具体的には，各クライアント PC 上で InfoSpace Plug と呼ばれるソフトウェアモジュールを動作させ，情報空間に含まれる計算機やファイル間の関連を，メタデータ DB と呼ばれるグラフデータベースに明示的に保持する。そして，このデータベースを利用することで，コミュニティ情報空間の管理を行う (図 1).

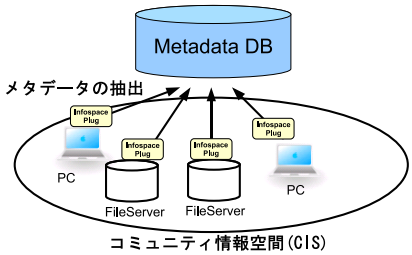

図 1　InfoSpace Governor のアーキテクチャ

3. 分散ファイル群高度管理のためのミドルウェア

本研究では，InfoSapce Governor において，ファイル間の関連発見およびそれを利用した高度ファイル管理機能を提供するミドルウェアの開発を行う．本ミドルウェアの特徴は次の 3 つである．(1) 単一

* “Development of Middleware for Advanced Management of Distributed Files” by Yuichiro MITSUMORI

の PC ではなく，複数の PC に分散したファイル群を対象としている．(2) 既存のフレームワークと互換性があり、できるだけその存在を意識させない．(3) ファイル間の関連など，既存のファイルシステムより豊富なメタデータを利用可能．

このミドルウェア実現のために，次の 4 つの要素技術の開発を行った．(1) 各クライアント PC におけるファイル操作のロギング機能．(2) 拡張可能な汎用のファイル関連発見エンジン．(3) ファイル群管理のためのルール処理機構．(4) 発見した関連を利用するためのライブラリ．本論文ではこの中でも特に，ファイル関連発見エンジンを中心に説明する．

4. ファイル関連発見エンジンの開発

ファイル関連発見エンジンは，ミドルウェアにおけるファイル間の関連の発見・利用を汎用的に行うための基盤として構築されている．本エンジンは，InfoSapce Plug により収集した情報を，関連計算モジュールと呼ばれるコアモジュールによって処理し，メタデータ DB に格納して利用することで実現される．本エンジンの特徴は，(1) プロセスを分けることによる効率的な関連の発見，(2) 拡張可能なプログラムインタフェースの提供，(3) 分散ファイル環境への対応，である．

4.1 効率的な関連の発見

関連計算モジュールでは，関連の発見を効率的に行うために，プロセスを関連の候補を生成を行う Cadidate Generation プロセスと候補の検証を行う Inspection プロセスの 2 つに分解する（図 2）．

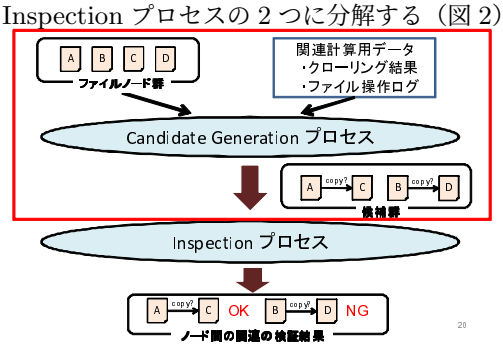

図 2　関連発見モジュールのアーキテクチャ

前者のプロセスでは簡単なチェックだけを行い，厳密な検証は後者のプロセスで行うことにより，実際に検証を行うファイル群を絞り込みを行うことができる．このように関連の検証を効率的に処理することで，関連発見の計算コストの削減を図っている．

4.2 拡張可能なプログラムインタフェース

本エンジンでは，それぞれのプロセスに対応する抽象クラスである，CandidateGenerator クラスと Inspector クラスが用意されており，このクラスを関連の種類ごとに実装してエンジンに登録することで，新たな関連が利用できるようになっている．このクラスの実装には，InfoSpace Plug が収集した情報などを提供するライブラリが利用可能である．

4.3 分散ファイル環境への対応

実際にファイルが分散する環境では，クライアントは常にサーバと接続しているわけではないため，関連の発見を行うタイミングが異なってしまう．また，関連発見処理をすべてクライアント側で行うとレスポンスに支障がでてしまう．これらの問題を解決するために，関連発見モジュールをクライアントとサーバに分散配置し，適切に動作するように設計した．

4.4 評価

本ミドルウェアを利用した場合に，関連の発見機能の実装に必要とするコード量と，利用しなかった場合のコード量を比較した．利用しなかった場合のコード量は，ミドルウェアの関連発見に関するコード行数を加算したものである．図 3 は，分散ファイル環境でコピーの関連を発見するのに必要なコード量の比較である．必要なコード量が大きく減っており，関連を利用したアプリケーションの開発コストを大幅に削減できると考えられる．

本ミドルウェアの利用	行数（行）
なし	4257
あり	282

図 3　必要なコード行数の比較

5. まとめ

本研究では，分散ファイル群高度管理のためのミドルウェアの開発に取り組んだ．具体的には，既存のファイルシステムでは明示的に扱われない，ファイル間の関連といったメタデータを利用可能にする基盤を構築した．対象とするのは，複数の計算機に分散して格納されたファイル群である．本ミドルウェア利用により，分散ファイル群管理のためのアプリケーションの開発が容易となると考えられる．

文献

[1] Soules, C. A. N., Ganger, G. R. : Connections: using context to enhance file search. In Proceedings of the 20th ACM Symposium on Operating Systems Principles, pp.119-132. 2005.

[2] 渡部徹太郎, 小林隆志, 横田治夫：キーワード非含有ファイルを検索可能とするファイル間関連度を用いた検索手法の評価，電子情報通信学会 第 19 回データ工学ワークショップ (DEWS2008) 論文集 E10-6, Mar 9-11, 2008.