分散サーバ環境における情報の共有手法に関する一提案

4AC－9

井上 晃　阿部 徹治

NTTアクセス網研究所

1 はじめに

近年，インターネットの普及やPC/WSの低価格化にともない，従来のサーバ/クライアント型システムは，分散サーバ環境に移行してきている．分散サーバ環境では，処理性能や耐故障性の向上を低コストで実現することができる．更には，携帯型PCによるサーバやDialup接続の採用など，より自由度の高い分散サーバ環境の実現が可能である．しかし，この分散サーバ環境の実現には，いくつか克服しなければならない課題がある．その一つに各サーバ間の情報の共有という問題である．従来，様々な共有手法が提案されているが，サーバ構成やネットワーク環境が頻繁に変更される環境では，これらは必ずしも最適とは限らない．

本稿では，この動的に変更される分散サーバ環境における情報共有問題を解決する手法として，サーバを一つのAgentと見なし，複数Agentの協調により，自律的に，かつ，非同期に情報を共有する手法を提案する．

2 既存手法の問題点

既存の複数サーバ間の情報共有手法について，その特徴を示す(Table 1)．サーバ間に必要とされるネットワーク接続性，サーバ構成の可変性，サーバ間主従関係の有無，情報更新の同期性についてまとめた．これらの

Table 1: 既存技術の特徴比較

Method	NFS	ftp+mirror	DB 複製技術
接続性	密	不問	密
可変性	固定	固定	固定
主従関係	あり	あり	あり
同期性	同期	非同期	同期

手法では，以下の点で，動的に変更される分散サーバ環境には，適用することが難しい．

- サーバ間のネットワーク接続状況やサーバ構成の動的変化に対応出来る構造ではない．
- サーバ間の主従関係が固定的である．

PFS[1]のように，既存技術を基に動的な通信環境に対応する試みもあるが，サーバ間主従関係は固定的である．

3 Agentを利用した情報共有手法

Agentを利用した複数サーバ間の情報共有手法について述べる．各サーバ上には，１つのAgentが存在し，ローカルな情報を管理する．同時に，Agentの相互協調により，サーバの情報共有を実現する(Fig.1)．ここでAgentとは，「サーバ上のデーモンプロセスの一つであり，自律的に状況を認知し，動作するもの」と定義する．

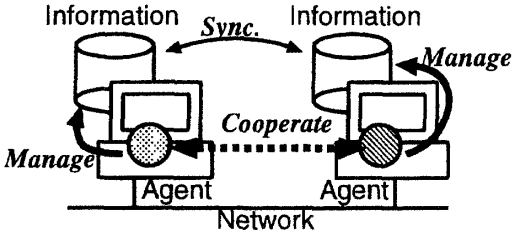

Fig.1: 概要

本手法は次の各項目に対応することを基本方針とする．

- サーバ間ネットワーク接続状況の動的な変化
- サーバ構成の動的な変更
- サーバ間の主従関係の非固定化

この方針を実現するため，Agentは動作状態として，次の４つのモードを選択可能とし，状況に応じて，自律的にこのモードを切り替える．

- 観察モード：ローカルな情報やネットワーク状態の変化を観察し，適当なモードに遷移する．
- 初期モード：ネットワークに接続された新規サーバの情報を，全体の情報に同期させる．
- 更新モード：情報の変化が検出されたときに，全てのサーバとの間で該当情報を最新に更新する．
- 削除モード：明示的な削除要求があった場合に，全てのサーバとの間で，情報の削除を行う．

また，各モード間の状態遷移関係をFig.2に示し，それに対応させながら，以下，各モードの動作の詳細について述べる．

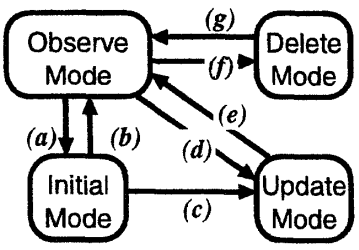

Fig.2: Agentの内部動作の状態遷移関係

●観察モード (Observe Mode)

このモードでは，常時，ローカルな情報の変化やネットワーク状況を監視し，適切なモードに遷移する．(1) 情報の更新があった場合，その情報について更新モードに遷移する (Fig.2.(d))．(2) 同様に，削除要求がきた場合には，削除モードに遷移する (Fig.2.(f))．(3) また，起動直後や周囲のネットワーク状況が変化し，自己以外のAgentをはじめて認知した場合には，初期モードに遷移する (Fig.2.(a))．

●初期モード (Initial Mode)

このモードでは，新規サーバの情報を周囲と同期させる．(1) 任意のAgentとの間で双方の全情報を比較する．(2) 異なる情報がある場合には，それについて更新モードに遷移する (Fig.2.(c))．(3) そうでない場合や，他にAgentがいない場合には，観察モードに遷移する (Fig.2.(b))．

●更新モード (Update Mode)

このモードでは，該当する情報について，接続されている全Agentとの間で，契約ネットプロトコルを用いて，最新の情報に更新する (Fig.3)．(1) Agent αはネットワーク上の全Agent $\beta_1 \sim \beta_n$ に対して，該当情報を提示し，入札を求める．(2) 提示されたAgentはその情報について，応札する．(3) Agent αは全ての入札の中から，最新の情報を選びだし，全Agentに入札結果を通知する．(4) 通知されたAgentは，最新情報を持つサーバから該当情報を取得し，更新する．(5) 更新が終了すると，観察モードに遷移する (Fig.2.(e))．

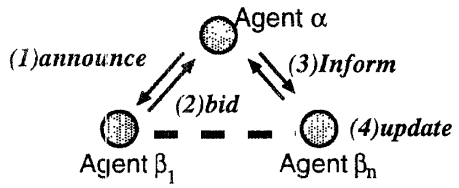

Fig.3: 更新モードにおける協調手順

●削除モード (Delete Mode)

このモードでは，情報の削除を行う．(1) 削除を要求された情報について，全Agentに削除承諾の確認を行う．(2) 全てのAgentから承諾が得られた場合，該当情報の削除を実行し，全Agentに削除を命令する．(3) 承諾が得られない場合，処理を中断する．(4) 処理が終了後，観察モードに遷移する (Fig.2.(g))．

4 実装と評価

実装は，10Mbps Ethernetに接続されたPC/AT互換機 (Intel Pentium 200MHz, 64MB) で動くOPENSTEP 4.1J上に行った．Agentは，ユーザレベルで動くデーモンプロセスとした．共有の対象なる情報として，特定ディレクトリ以下のファイル構造 (ファイルとディレクトリ) を用いた．即ち，全てのサーバ間で，同じファイル構造を共有することが目的となる．同様に，情報の変化は，ファイル構造の変化に相当する．

評価は，同じファイル構造の共有方法のひとつであるNFSと比較し，応答性と情報更新時間について行った．

・応答性

情報更新に対する応答性の評価として，10MBのファイルの書込み，読込み時のローカルな終了までの時間を比較した (Table 2)．値は10回の計測の平均である．

Table 2: 提案手法とNFSの応答時間の比較

Method	Agent	NFS
Write[sec]	2.8	47.0
Read[sec]	3.0	2.7

・情報更新時間

サーバ全体への情報の更新は，Agentの協調により非同期に行われる．ここではサーバ全体の更新終了までの時間を評価する．2台のサーバ間の更新に限定し，10MBのファイルを追加した場合と，削除した場合の処理時間を計測した (Table 3)．値は10回の平均である．

Table 3: 提案手法とNFSによる総処理時間の比較

Method	Agent	NFS
Copy[sec]	23.7	47.0
Delete[sec]	8.3	2.7

5 考察

分散サーバ環境における情報の共有方法として，Agentを利用した情報共有手法を提案し，評価した．応答性について，書込みがNFSより優れるのは，本手法の構造上，情報の変更はローカルなファイルシステムの変更にほぼ等しい為である．一方，読込みがNFSより若干劣るのは，NFSのキャッシュの効果だと思われる．全体の情報更新時間については，追加の場合はNFSの倍程度の性能が出ている．これは，本方式が情報の転送に圧縮を用いていることとNFSにおける書込み保証処理のオーバヘッドの影響だと思われる．一方，削除の場合には，協調プロトコルによる処理のオーバヘッドにより，若干劣る．今後は，サーバ台数が多数になった場合や多様な通信状況での処理性能を評価していきたい．

参考文献

[1]楯岡, 植原, 砂原, 寺岡,"PFS:通信環境に動的に適応するファイルシステム", インターネットカンファレンス'96 論文集,p.81-88,1996