

## WebDAV を用いた遠隔地間相互データ同期保持システムの 構築と JGN 上での評価

渡邊 貴之† 湯瀬 裕昭† 鈴木 直義† 橋本 浩二‡ 柴田 義孝‡

† 静岡県立大学 経営情報学部  
‡ 岩手県立大学 ソフトウェア情報学部

大地震・火山噴火などの大規模な自然災害や人的災害から、重要な電子データを保護する方法としては、これらのデータを蓄積するファイルサーバを遠隔地に複数設置し、各拠点間でデータの同期・非同期ミラリングを行う方法が考えられる。本報告では、大規模災害等を想定し、遠隔地に設置したサーバ間でデータファイルを同期保持する汎用的なシステムを提案する。本システムでは、サーバ間のミラリングとクライアントとのファイル共有のためのプロトコルとして、ファイルの分散オーサリング及びバージョン管理プロトコルである WebDAV を用いる。これにより、ストレージブロック単位ではなく、ファイル・ディレクトリといったリソース単位での柔軟なミラリング環境を構築することができる。本研究では、WebDAV サーバを直線距離で約 600km (回線距離で約 800km) 離れた静岡県立大学と岩手県立大学に設置し、Japan Gigabit Network を通じて同期ミラリングを行う実験を行い、システム性能に関する評価を行った。

## Development of Data Mirroring System between Distributed Remote Servers Based on WebDAV, and Its Performance Evaluation on JGN

Takayuki Watanabe† Hiroaki Yuze† Naoyoshi Suzuki† Koji Hashimoto‡  
Yoshitaka Shibata‡

†School of Administration and Informatics, University of Shizuoka  
‡Faculty of Software and Information Science, Iwate Prefectural University

In this report, we propose the generalized system of the data mirroring between distributed remote servers. In this system, the WebDAV (Web-based Distributed Authoring and Versioning) is used as a protocol for the file sharing between a client and a server, and mirroring between servers. Therefore, the flexible mirroring environment based on resources-oriented access, such as file level and directory level, can be developed. In this research, we placed the WebDAV servers at University of Shizuoka and Iwate Prefectural University, which are with distance about 600km or network distance about 800km. We demonstrated the efficiency of our system using Japan Gigabit Network.

### 1 はじめに

現在、コンピュータの高性能化やディスクストレージの大容量化、ネットワーク技術の発達などにより、膨大な情報データがコンピュータに蓄積され、ネットワークを介した情報共有や電子商取引のために活用されている。一般的に、電子データをグループ内で共有したり、共同して管理・編集したりする場合には、データファイルを蓄積するためのファイルサーバや NAS (Network Attached Storage) を用意し、このサーバにネットワークを介して多くのクライアントコンピュータがアクセスすることによって情報共有環境を実現している。従って、データファ

イルを保持するサーバやネットワークに障害が発生すると、情報データの喪失などによって多大な損失を被る可能性がある。更に、将来予想される大地震・火山噴火などの大規模な自然災害や人的災害によって、ファイルサーバを設置するサイト全体がダウンする可能性もある。この場合、情報データを保持するファイルサーバやストレージを遠隔地に複数設置し、各拠点間でデータの同期・非同期ミラリングを行う方法が考えられる。

そこで本研究では、遠隔地に設置したサーバ間でデータを同期保持する汎用的なシステムを提案する。本システムでは、遠隔同期ミラリングのためのサーバ間プロトコルとして、ファイルの分散オーサリン

グ及びバージョン管理プロトコルである WebDAV (Web-based Distributed Authoring and Versioning) を用いる [1, 2]。これにより、ストレージブロック単位ではなく、ファイル・ディレクトリといったリソース単位での柔軟なミラリング環境を構築することができる。また、クライアント・サーバ間のファイル共有プロトコルとしても WebDAV を採用する。WebDAV は HTTP の拡張プロトコルであり、これまでに培われた暗号化技術・XML 技術・相互運用性などの Web 技術の恩恵を得ることができる。現在、多くのアプリケーションソフトウェアや OS が WebDAV に対応し始めており、一般ユーザからは単なる共有ディレクトリとして取り扱うことが可能であり、特定の OS やハードウェアに依存しないという利点がある。本研究では、まず、WebDAV サーバを理想的な環境下で運用した場合のミラリング性能に関する実験を行った。更に、WebDAV サーバを直線距離で約 600km (回線距離で約 800km) 離れた静岡県立大学と岩手県立大学に設置し、JGN (Japan Gigabit Network) を通じて同期ミラリングを行う実験を行い、システム性能に関する評価を行った。

## 2 WebDAV の概要

### 2.1 分散ファイルシステムと分散オーサリング

情報共有や共同作業を目的としてファイルサーバを設置し、複数のクライアントからアクセスする場合、サーバ側のファイルシステムをクライアント側からネットワークを介して直接マウントし、ファイル操作を行う分散ファイルシステムが広く用いられている。分散ファイルシステムの実装例としては、Sun Microsystems 社の NFS (Network File System) や、Microsoft 社の CIFS (Common Internet File System)、Apple 社の AFP (AppleTalk Filing Protocol) など様々な方式やプロトコルがある。また、汎用的な分散コンピューティングを目指した方式として、AFS (Andrew File System) や OSF/DFS (Distributed File System)、Coda [3] がある。一方、これらの技術は、特定の OS に依存していたり、LAN 環境下での運用に限定されていたりする場合が多い。

このような状況の中、1998 年に IETF から RFC2291 が勧告され、WebDAV というアプリケーションレイヤのプロトコルが提案された。WebDAV は、Web 技術の標準プロトコルである HTTP の拡張であり、HTTP の普及と共に発展した様々なインターネット技術を継承したまま、新たに情報共有に欠かせないファイル書込みやロック仕様、属性操作、世代管理に関する機能などを追加したプロトコルである。実際の詳細な仕様に関しては、オーサリングについては RFC2518 において、バージョン管理に

表 1: WebDAV リクエストメソッド (RFC2518)

メソッド名	機能
PROPFIND	リソースプロパティの取得
PROPPATCH	リソースプロパティの設定・削除
MKCOL	コレクションの作成
DELETE	リソースの削除
PUT	リソースの書込
COPY	リソースの複写
MOVE	リソースの移動
LOCK	リソースの共有・排他ロック
UNLOCK	リソースのロック解除
GET, HEAD, POST	HTTP/1.1 と同様

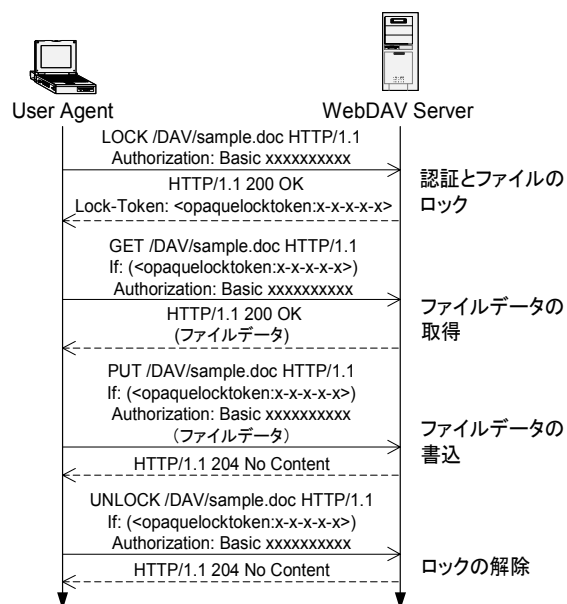


図 1: ファイル編集時における動作シーケンス例

については RFC3253 において規定されている。これらは特定のプログラムや OS の実装に依存した仕様ではないため、WebDAV に基づく多くのサーバ・クライアントプログラム間で高い相互運用性を確保できる。WebDAV で導入された XML に基づく属性操作や世代管理といった機能は、既存のファイル転送プロトコルや分散ファイルシステムの範疇に収まらない、分散オーサリング環境を実現するための機能といえる。

### 2.2 WebDAV の機能

WebDAV では、HTTP と同様に User Agent からのリクエストに対し、サーバがレスポンスを返すことによって、その機能を提供する。WebDAV で規定されているリクエストメソッドを表 1 に示す。ただし、RFC3253 で規定されたバージョン管理メソッドについては省略した。WebDAV では、通常のファイルシステムを抽象化したリソース、コレクション、

プロパティという概念を用いる。

- ・リソース: URI で指定可能なファイルやコレクション
- ・コレクション: 複数のリソースの集合 (ディレクトリ・フォルダに相当)
- ・プロパティ: リソースに関連付けられた属性 (名前・値のペア)

また、実際のファイル編集時の動作シーケンス例を図 1 に示す。WebDAV の通信は HTTP と同様に TCP80 番ポートを用いる。従って、HTTP の普及によって開発された認証機構や SSL などの暗号化方式をそのまま利用できるため、セキュリティの確保が可能である。また、ロック時にはリソースのロック識別子として UUID (Universal Unique Identifier) がサーバ側から通知される。以後、ロックされたリソースを操作する際には、この UUID を用いる。

### 3 WebDAV を用いた同期ミラリング方式

#### 3.1 同期ミラリング方式の検討

ファイルサーバや NAS の可用性を高める方法としては、ディスクストレージを冗長化する RAID (Redundant Array of Independent Disks) 方式がある。また、複数のサーバが Fibre Channel 接続された 1 台のストレージを共有することによって、サーバ自体の故障によるサービスの停止を避ける方式もある。特にデータセンタ等の大規模システムでは、複数のサーバ・ストレージを専用の SAN (Storage Area Network) によって接続する例が一般的である。一方、一般に高価な SAN を構築せずに可用性を高める方法としては、ローカルストレージを持つ複数のサーバを、LAN を介してソフトウェア的にミラリング・ストライピングするネットワーク分散型 RAID 方式が提案されている [4, 5]。

上記に示した手法では、その適用環境として、サーバやストレージが同一のサイト内に設置されると仮定している場合が多い。しかし、大規模災害によって、サイト全体がダウンする可能性がある。この問題に対応するためには、サーバやストレージを遠隔地に複数設置し、各拠点間でデータの同期ミラリングを行う必要がある。遠隔地同期ミラリングを実現する手段としては、まず、遠隔地に構築した SAN 同士を長距離接続して、ストレージ間で直接ミラリングを行う方法が考えられる。ところが、ストレージ間の接続距離は、Fibre Channel の規格上 10km に制限されるため、大規模災害を想定した 100km オーダのミラリングは不可能であった。一方、近年、FCIP (Fibre Channel over IP) や iSCSI (SCSI over IP) など、既存のストレージプロトコルを IP でカプセル化した IP ベースのストレージ技術が提案されており、既設の LAN/WAN 回線を利用した遠隔 SAN

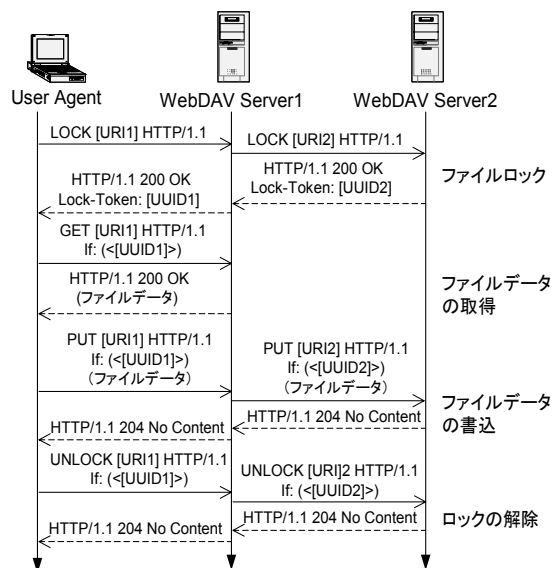


図 2: ミラリングを考慮したファイル編集時における動作シーケンス例

の構築も可能となった。但し、FCIP や iSCSI などでカプセル化される既存の SCSI コマンドは、元々今日のインターネット環境を考慮したプロトコルではないため、IP ネットワーク環境下でのセキュリティ面の不安が懸念されている。また、ストレージブロック単位の転送に特化しているため、IP パケットへのカプセル化の際にオーバーヘッドが大きく、専用 NIC の利用が推奨されている。

一方、遠隔地に設置した WAN 接続のファイルサーバ同士で、同期ミラリングを行う方法も考えられる。この場合、ストレージブロック単位でなく論理的なファイル・ディレクトリ単位のミラリングであれば、ファイルサーバ間の通信プロトコルとして、様々な既存のファイル共有プロトコルを利用することができる。通常のファイルサーバでは、クライアントからのアクセスはリソース単位に行われる。また、リソース単位のミラリングに着目すると、情報共有という観点からはより柔軟性のあるシステムを構築できる。例えば、リソース毎にミラリングサーバや論理的な保存ディレクトリを変更したり、ミラリングサーバ毎にリソースの属性を付け替えたりすることができる。

#### 3.2 処理概要

ファイル・コレクションといったリソース単位でのファイル共有・分散オーサリング環境と、データ同期ミラリングを合わせて実現するために、本提案システムではクライアント・サーバ間、サーバ・サーバ間のプロトコルとして WebDAV を用いる。クライアントからのリクエストに対し、以下の手順に従って、サーバ及びミラーサーバは動作する。

```

DAVLockDB          /usr/local/apache/var/DAVLock
DAVMirrorLockDB   /usr/local/apache/var/DAVMLock
<Location /dav>
  DAV On
  DAVMirrorHost    hoge1
  DAVMirrorHost    hoge2
  AuthUserFile     /usr/local/apache/var/.htpasswd
  AuthGroupFile    /dev/null
  AuthName         DAVhome
  AuthType         Basic
  <Limit GET PUT DELETE PROPFIND PROPPATCH MKCOL
COPY MOVE LOCK UNLOCK>
    Require user watanabe
  </Limit>
</Location>

```

図 3: ミラリングに対応した DAV サーバの設定

- 1) クライアントからサーバ 1 へ PROPPATCH, MKCOL, DELETE, PUT, COPY, MOVE, LOCK, UNLOCK リクエストが発生した場合、リクエストを受けたサーバ 1 は対応する処理を行う。また、このリクエストをあらかじめ登録されたサーバ  $i$  ( $= 2, \dots, N$ ) へ転送する。その際、URI 中のホスト名は、適宜、サーバ  $i$  に置き換えられる。
- 2) LOCK, UNLOCK 要求は、全てのサーバ上でリソースのロックが成功したときのみ、クライアントへ成功レスポンスを返す。その際、クライアントへはサーバ 1 の UUID が通知される。サーバ 1 はサーバ  $i$  から通知された全ての UUID を、ミラー用のデータベースに、  
 [ミラーホスト名] : [リソース URI] : [UUID]  
 の形式で登録しておく。以後、サーバ 1 がサーバ  $i$  に対して代理アクセスを行う場合、このデータベースを参照して必要な UUID を得る。
- 3) パフォーマンス確保のため、PROPFIND, GET, HEAD リクエストの場合には、要求を直接受け取ったサーバ 1 がそのままレスポンスを返す。
- 4) サーバ 1 においてミラー先をサーバ 2 に、サーバ 2 においてミラー先をサーバ 1 に設定することで、相互ミラリングを行うことが可能である。但し、ループを避けるため、ミラーサーバとして登録されたホストからの DAV アクセス時には、そのリクエストを他のサーバには転送しない。

以上を考慮して、図 1 に対してミラリング用のシーケンスを加えたものが図 2 である。但し、認証処理に関しては図内における記述を省略した。

### 3.3 mod\_dav へのミラリング機能の実装

現在、WebDAV の機能を実装したサーバプログラムが多数存在する。例えば、オープンソースの Web

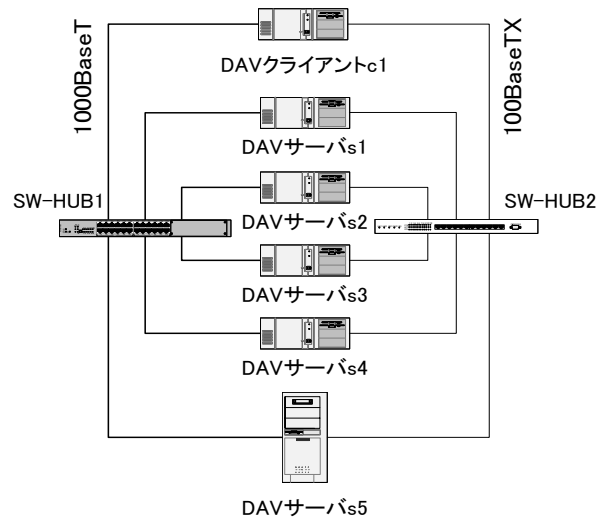


図 4: 理想的実験環境でのネットワーク構成

サーバである Apache に対して、外部モジュールとして DAV の機能を追加する mod\_dav がリリースされている。Apache は異なる OS 上で高い互換性を有しており、DAV ミラリング機能を実装するのに最適なプラットフォームであると言える。今回、この mod\_dav の version 1.0.3 に対して、本研究で提案したミラリング機能を追加する拡張を行った。コーディングは C 言語により行い、Linux 上においては GCC2.96、MS-Windows 上においては VisualC++6.0 でのコンパイルと動作を確認した。

DAV サーバの設定は、Apache の設定ファイルである httpd.conf 内で行う。ミラリング機能に関する記述を含めた mod\_dav の設定部分を図 3 に示す。太字で示した箇所が、今回拡張を行ったミラリング機能に関する設定である。DAVMirrorLockDB ディレクティブによって、ミラーサーバ毎の UUID を保持するデータベースを指定する。また、DAVMirrorHost ディレクティブによって、ミラー先サーバを複数指定することができる。

## 4 実験環境と性能評価

### 4.1 理想的環境下での実験

本研究で実装したシステムが、実用可能かどうか判断するため、図 4 に示すハードウェア及びネットワーク構成においてファイルの書き込みに関する性能測定を行った。

#### 4.1.1 実験環境

ハードウェア環境としては、クライアント c1 及びサーバ s1 から s4 としては、Intel Pentium4

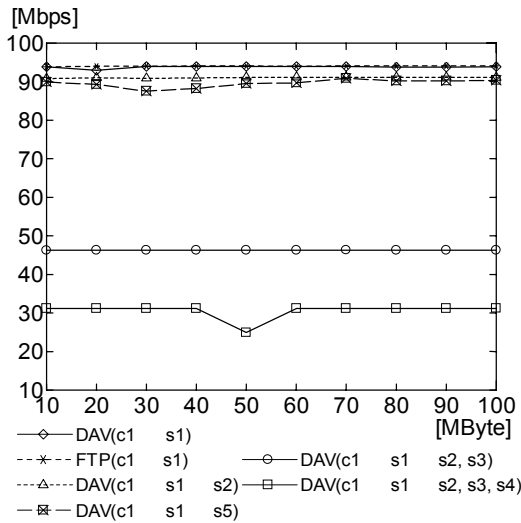


図 5: Fast Ethernet 環境におけるスループット

2.0GHz, Linux2.4 ベースのシステムを用いた。また、サーバ s5 については、Intel Dual Xeon 2.2GHz, Windows XP Pro ベースのシステムを用いた。また、100BaseTX SW-HUB として BUFFALO LSM10/100-24 を、1000BaseT SW-HUB として 3COM SuperStack3 Switch4900 を用いた。ソフトウェア環境としては、サーバ上では Apache 1.3.23 及びミラリング拡張を施した mod\_dav を用いた。クライアント側では、DAV クライアントとして cadaver 0.19.1 を、比較用の FTP クライアントとして NcFTP 3.1.3 を用いた。尚、netperf プログラムを用いたネットワークスループットの測定結果としては、100BaseTX 側が 94.12Mbps、1000BaseT 側が 421.90Mbps であった。

#### 4.1.2 Fast Ethernet を用いた実験

クライアント c1 からサーバ s1 にファイル書き込みを行い、s1 のミラーサーバとして s2, s3, s4, s5 を構成を変えながら指定し、書き込みに要した時間からスループットを算出した。その際、ファイルサイズを 10MByte から 100MByte まで 10MByte 毎に測定を行った。Fast Ethernet を用いたスループットの実測結果を図 5 に示す。結果は全て 5 回測定を行った平均値である。クライアント及びサーバが 1 対 1 の構成では、ネットワーク自体の最大スループットと同等の書き込み速度を示している。これは FTP を用いた場合と同等であった。一方、ミラーサーバとして s2 (Linux) もしくは s5 (Windows XP) を追加した場合、速度低下が若干見られるが、ほぼ 1 対 1 の構成と変わらない性能を示した。これは全二重通信によって、各々専用の送信路を用いているためと考えられる。また、Linux と Windows XP という異なる OS 間においても、問題無く同期ミラリングが行えることを確認した。

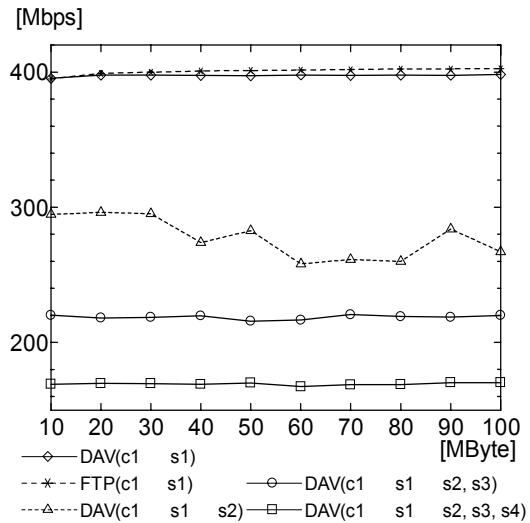


図 6: Gigabit Ethernet 環境におけるスループット

一方、ミラーサーバとして s2, s3 の 2 台を追加した場合、スループットが 1 対 1 構成時に比較して約 50%低下していることが分かる。また、ミラーサーバを 3 台を追加した場合は約 30%程度に低下する。これは、ミラーサーバへの送信路を 2 台もしくは 3 台で共有しているためと考えられる。

#### 4.1.3 Gigabit Ethernet を用いた実験

Fast Ethernet 環境での実験と同様に、クライアント c1 からサーバ s1 に書き込みを行い、ミラーサーバとして s2, s3, s4, s5 を指定して、書き込みに要した時間からスループットを算出した。Gigabit Ethernet を用いたスループットの実測結果を図 6 に示す。クライアント及びサーバが 1 対 1 の構成では、先の結果と同様に、DAV 及び FTP においてネットワーク自体の最大スループットとほぼ同等の書き込み速度を示している。実行速度で 50MByte/sec の書き込み性能を示しており、UltraWide SCSI の理論最高性能である 40MByte/sec を上回る性能を示している。一方、ミラーサーバとして s2 (Linux) もしくは s5 (Windows XP) を追加した場合、s2 との組み合わせでは正常動作したが、s5 との組み合わせでは、書き込みの初期に数秒間の応答停止状態が発生し、書き込み速度に大きなばらつきを生じた。ばらつきに規則性が無く、場合によっては十秒程度停止することから、図 6 には不記載とした。また、s2 との組み合わせにおいても、ファイルサイズ毎にスループットの変動が見られた。

## 4.2 JGN を用いた遠隔地間実証実験

実装したシステムを遠隔地に設置し、JGN (Japan Gigabit Network) を通じて同期ミラリングを行う

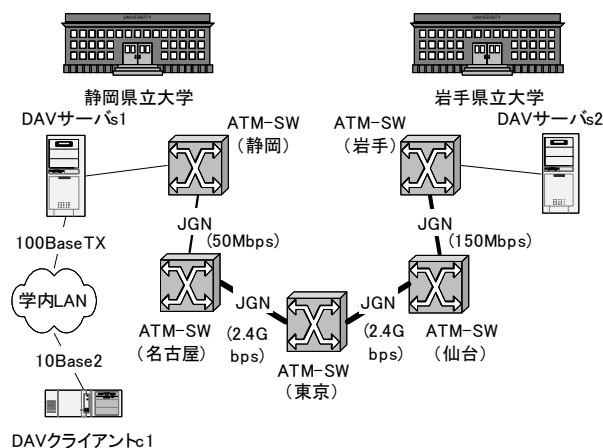


図 7: 遠隔地間実験におけるネットワーク構成

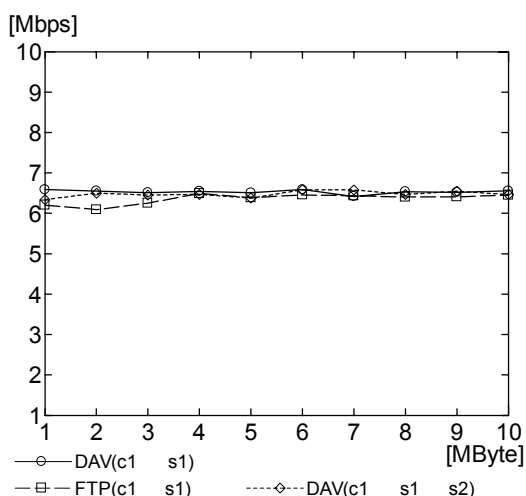


図 8: 遠隔地間通信環境におけるスループット

実証実験を行った。

#### 4.2.1 実験環境

実験におけるネットワーク構成を図 7 に示す。サーバを設置した静岡県立大学と岩手県立大学間は、直線距離で約 600km、JGN 回線距離で約 800km 離れている。JGN で用いた ATM 回線は、最も帯域が細い静岡・名古屋間で 50Mbps である。また、s1 は静岡県立大学の学内 LAN (ATM155Mbps → 10Base5 → 10Base2) を経由して c1 と接続されている。

ハードウェア環境としては、クライアント c1 に関しては前節と同様である。また、サーバ s1, s2 は Intel Celeron 800MHz, Linux2.4 ベースのシステムを用いた。各サーバは ForeRunnerLE 155M ATM-NIC によって JGN の ATM-SW と接続されている。但し、netperf プログラムを用いた実際のネットワークスループットの測定では、c1 から s1 まで 6.58Mbps、s1 から s2 まで 10.22Mbps であった。

#### 4.2.2 実験結果

クライアント c1 からサーバ s1 にファイル書き込みを行い、s1 のミラーサーバとして s2 を設定し、書き込みに要した時間からスループットを算出した。その際、ファイルサイズを 1MByte から 10MByte まで 1MByte 毎に測定を行った。スループットの実測結果を図 8 に示す。結果は全て 5 回の平均値である。DAV 及び FTP においてクライアント及びサーバが 1 対 1 の構成、ミラーサーバを岩手側に設定した場合の両者において、ネットワーク自体の最大スループットと同等の書き込み速度を示している。前節での実験結果と合わせて考察すると、より高速な回線を用いることで、内部ストレージと比較しても遜色のない書き込み性能を実現できると推測される。

## 5 まとめ

本報告では、WebDAV を用いた遠隔地同期保持システムの構築と、本システムを用いて行った実証実験について述べた。理想的環境下における実験と、JGN 環境下における遠隔地間実験の結果から考察すると、本研究において提案したシステムは、実用可能な性能を有すると判断できる。ただし、Gigabit Ethernet 環境下で見られた不安定性については、将来的な課題としたい。

今後の展望としては、Apache2.0 への対応と IPv6 環境下での実証実験、より高速な WAN 環境下での遠隔地間実験を行う予定である。更に、WebDAV ベースの分散オーサリング環境では、実際に書き込まれるリソースとは別に、任意の外部プロパティ値を XML 形式で設定可能である。従って、ミラリング時にサーバ毎に異なる情報をプロパティ値として設定することで、位置情報等を埋め込むことが可能であり、これを用いて新たな応用が可能ではないかと期待している。

## 参考文献

- [1] 田中 真一「WebDAV 入門」、ジャストシステム出版部、2001 年 8 月。
- [2] 一條 博「WebDAV 分散ファイル共有環境の作成」、テクノプレス、2002 年 3 月。
- [3] 稲村 浩、「Coda プロジェクトの概要」、情報誌学会研究報告 マルチメディア通信と分散処理 DPS-95, No.073-010, Dec. 1995.
- [4] 中村 俊一郎, 峯村 治実, 山口 智久, 清水 洋, 渡辺 尚, 水野 忠則, “分散 RAID 方式ビデオサーバ”, 情報誌学会研究報告 マルチメディア通信と分散処理 DPS-95, No.073-022, Dec. 1995.
- [5] K. Hwang, H. Jin, and R.S.C. Ho, “Orthogonal Striping and Mirroring in Distributed RAID for I/O-Centric Cluster Computing,” *IEEE Trans. on Parallel and Distributed Systems*, Vol. 13, No. 1, Jan. 2002.