

## オンラインストレージを用いた分散仮想ディスクの開発

野尻 祐亮<sup>†</sup> 金井 遵<sup>†</sup> 並木 美太郎<sup>†</sup>

本研究では、個人ユーザの利用を意識した、新たな分散ディスクシステムの開発を行った。ストレージサーバには、インターネット上の HTTP サーバ、FTP サーバのほか、ISP 提供のディスクスペースやオンラインストレージサービスなどを利用することを想定した。クライアント側では、本仮想ディスクシステムはディスクデバイスとしての透過性の高いインタフェースを提供する。ネットワーク上のディスクを仮想化するだけでなく、ソフトウェア RAID によるディスクアレイ化も可能とした。評価では満足いくパフォーマンスが得られた。今後の課題は KNOPIX への組み込みやセキュリティの向上である。

### Development of Distributed Virtual Disk System Using Online Storage

YUSUKE NOJIRI<sup>†</sup>, JUN KANAI<sup>†</sup> and MITARO NAMIKI<sup>†</sup>

In this study, we developed new virtual disk system suitable for individual users. It is supposed to use HTTP servers, FTP servers, disk space of ISP or online storage service as storage server. On client side, the virtual disk system provides highly transparent interface as a disk device, which enables not only virtualization of disks on network but also composing disk array by using software RAID. The evaluation showed sufficient performance. The problems are to merge the virtual disk system into KNOPIX and to improve security.

#### 1. はじめに

ドキュメントファイル、メール、あるいはユーザの使用ソフトウェアや設定情報など、ユーザそれぞれの個人データを、どの端末からも共通して利用したいというニーズがある。そのため、ドキュメントファイルやメールは端末間でコピーする必要があり、ソフトウェアや設定情報はそれぞれの端末でセットアップしなければならない、作業は煩雑である。

一般に提供されているオンラインストレージサービスにより、インターネット上のサーバ内にファイルを保管することが可能である。オンラインストレージや ISP の提供する HTTP サーバを利用することで、どの端末からでも共通の個人データにアクセスできるようになることができる。しかし、操作が煩雑で統一されておらず、利便性が高いとはいえない。本研究では、インターネット上の HTTP、FTP サーバやオンラインストレージサービスなどを、仮想的なディスクデバイスとしてアクセスできるようにする。本システムの

開発により、インターネットを通じた個人データの利用において、高い利便性や機能性を提供する。

#### 2. 目 標

本研究では、個人ユーザの利用に適した仮想ディスクの開発を行う。本研究の目標は、次の通りとする。

- 透過性の高いシステムとする。すなわち、ユーザには単なるローカルディスクであるかのように見せかけて、ローカルディスクと変わらない扱いやすさを提供する。
- 複数の種類の外部ストレージを利用可能とし、ユーザの環境に対する柔軟性をもたせることで、導入や管理運用が容易となるようにする。
- 新たな種類の外部ストレージへの拡張が容易とする。
- ソフトウェア RAID<sup>1)</sup>を用いたディスクアレイの構築にも応用できるようにし、並列化によるパフォーマンス向上や耐障害性の向上を可能とする。

#### 3. 概 要

本研究で提案する仮想ディスクシステムの概念図を図 1 に示す。ユーザは、インターネット上に仮想的に

<sup>†</sup> 東京農工大学  
Tokyo University of Agriculture and Technology

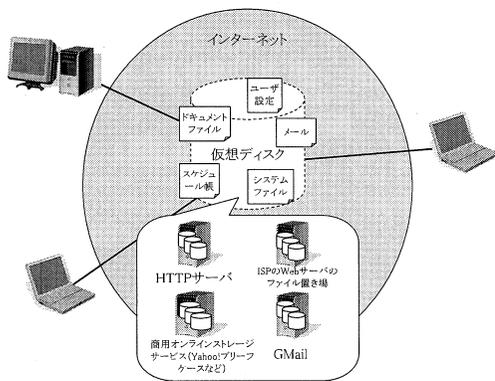


図 1 概念図  
Fig. 1 Conceptual Diagram

存在するディスクに、ドキュメントファイルやメールなどの個人データを保存し、取り出すことができる。物理的なシステム構成としては、インターネット上に存在するストレージサーバ（外部ストレージ）に、クライアントマシンのストレージドライバが入出力を行う方式を採用する。

本システムでは、ユーザの視点から、通信に用いるプロトコルの種類やハードウェアの物理的な構成を隠蔽し、単なるストレージデバイスのように見せかける。ユーザにとっては、あたかもインターネット上に存在するストレージデバイスを直接読み書きしているかのような利用形態を提供する。

## 4. 設 計

### 4.1 ファイルシステムレベル対ブロックデバイスレベル

本開発では、ファイルシステムのドライバではなく、ブロックデバイスのドライバを作成する。<sup>2)</sup> ファイルシステムレベルの実装では、ext3 や NFS といったファイルシステムドライバのように、ファイル単位で create, link, read, write などのファイルシステム操作を実行する。ブロックデバイスレベルの実装では、IDE ドライバや SCSI ドライバのように、セクタ単位のブロック I/O を実行する。ファイルシステムレベル・ブロックデバイスレベルの各レベルでの設計・実装においては、表 1 に示すような長所と短所がある。それぞれ長所・短所があるが、本開発ではブロックデバイスレベルの実装を採用する。ブロックデバイスとすることにより、設計・実装を比較的単純に行うことができる。また、既存の ext3 のようなファイルシステムや、ソフトウェア RAID といった、ストレージデバイスのための既存の枠組みをそのまま適用するこ

表 1 ファイルシステム・ブロックデバイス各レベルの設計・実装における長所・短所

Table 1 Advantages and disadvantages in design and implementation on file system level or block device level

	ファイルシステムレベル	ブロックデバイスレベル
長所	一度に読み書きするサイズを自由に決められたり、ファイル単位で先読みすることができるという意味で、保存するデバイスの特性に合わせた設計が可能。	単なるデータのかたまりを処理するだけでいいため、設計・実装が単純である。一貫性の保証や各種ファイルシステム操作などの仕事を既存のファイルシステムに任せられる。ext3 や FAT のような、既存の枯れたファイルシステムの安定性の恩恵が受けられる。
短所	各種ファイルシステム操作を、一貫性制御も考慮しつつ実装するため、実装が難しい。	ブロックレベルで一貫性をどのように保証するかが問題となる。ファイルシステムは、「1 台のデバイスに対して複数の端末が読み書きする」という構成を想定していない。パフォーマンスの最適化が、セクタサイズやファイルシステムといった枠に制限されてしまう。

とができ、パフォーマンスや信頼性の向上を行える。

### 4.2 仮想ディスクドライバの内部構成

クライアントマシン上で動作させる仮想ディスクドライバの構成を図 2 に示す。ドライバは、カーネルモード部分とユーザモード部分の 2 つの部分からなる。実際のネットワーク入出力処理などをユーザモードで行うことで、次のような利点が生じる。

- カーネルモードでのプログラミングは、メモリの取り扱い方が独特であったり、デバッグが難しかったりなど、カーネルモード特有のスタイルに縛られる。ユーザモードではそのような制約がなく、容易に開発が可能である。
- ユーザモードでは各種ライブラリをリンクできるため、本システムにおいて必要なネットワーク入出力のような凝った処理は、ライブラリを利用できる。
- プロトコル依存処理をユーザモードに分離するため、新たなプロトコルへの拡張が容易となる。

カーネルモード・ユーザモードの各部分について述べる。

### 4.3 カーネルモジュール vdisk.ko

カーネルモジュール vdisk.ko は、カーネル側に向けたインタフェースとしてブロックデバイス /dev/vdiskn (n はマイナー番号)、ユーザモード側

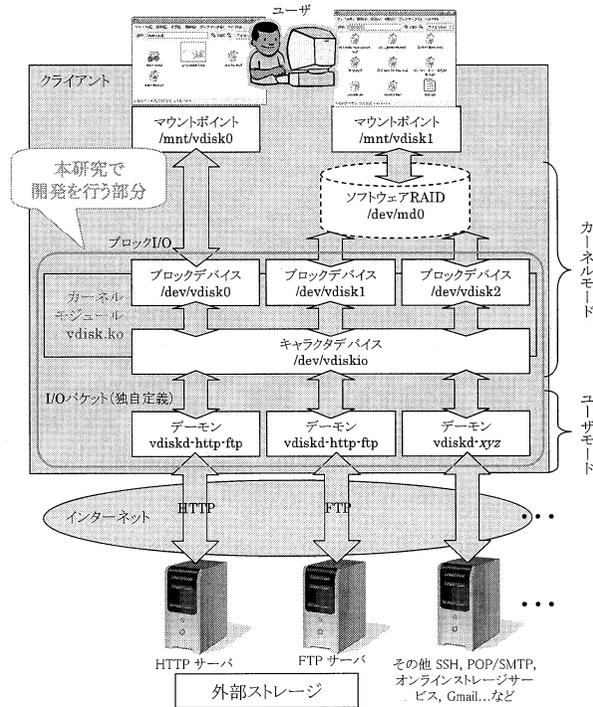


図 2 仮想ディスクシステムの設計  
Fig. 2 Design of virtual disk system

に向けたインタフェースとしてキャラクタデバイス `/dev/vdiskio` を持つ。カーネルモジュール `vdisk.ko` は次のような役割を持つ。

- ブロックデバイスインタフェースを提供することで、ユーザに対して仮想的なディスクと見せかける。
- カーネルから受け取ったブロック I/O リクエストを、独自定義の I/O リクエストパケットに変換して、ユーザ空間へ転送する。
- ユーザ空間から独自定義のレスポンスパケットを受け取り、カーネルに対してブロック I/O の終了を通知する。
- カーネルからブロック I/O リクエストを受け取ってから、指定した時間内（デフォルトでは 120 秒）にユーザ空間から I/O レスポンスパケットが返ってこなかった場合、処理が完了しなかったものとして、リクエストを破棄する（タイムアウト）。

#### 4.4 デーモン `vdiskd-xyz`

ユーザモード部分は、デーモンプログラム `vdiskd-xyz` (`xyz` はプロトコル名など) として実装する。`vdiskd-xyz` は、カーネルモジュール `vdisk.ko` から

送られてきた I/O パケットを解釈し、実際にインターネット上の外部ストレージに対してダウンロード・アップロードを行う。デーモンは、外部ストレージの種類（プロトコル）ごとに特化したプログラムとして実装する。仮想ディスクシステムの構成にあたっては、そのシステムで用いる外部ストレージの種類に対応したデーモンを使用する。

デーモンプログラムの構成としては、さらに 2 つの部分に分けられる（図 3）。これは FUSE<sup>3)</sup> と同様の構成となっている。

**共有ライブラリ `libvdisk`** 外部ストレージの種類に依存しない処理は、共有ライブラリ `libvdisk` にまとめている。`libvdisk` 内では、ブロックデバイスノードの作成と初期化、キャラクタデバイスノード `/dev/vdiskio` との各種パケットのやり取りと解釈、細かいセクタ単位データのブロック単位への取りまとめ、ブロックデータのキャッシュを行う。外部ストレージ依存処理は、次に述べる実行形式ファイル部分にコールバックして依頼する。

**実行形式ファイル** この中で行うことは、HTTP、FTP などのプロトコルに依存する処理、例えば

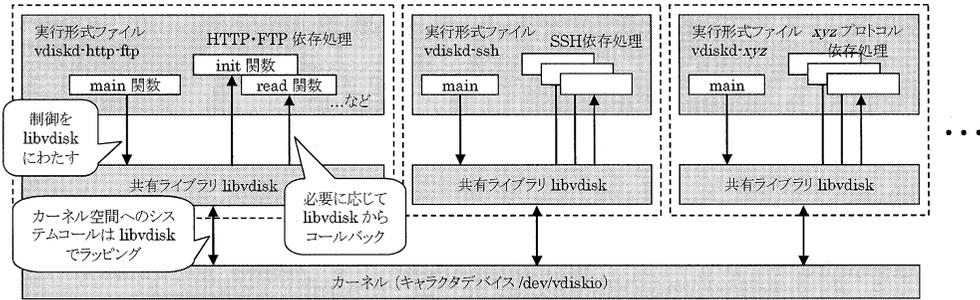


図 3 デーモンプログラムの内部構成  
Fig. 3 Configuration of the daemon programs

接続の確立・切断, エンコード・デコード, データの送受信, セッション管理などを, コールバック関数として宣言・定義し, その関数テーブルを libvdisk に通知することである。

#### 4.5 処理の流れ

カーネルモジュール vdisk.ko と, ユーザモードで実行するデーモン vdiskd-xyz は, 本デバイスドライバ独自定義の packets を互いにやり取りしながら協調動作する。ここでは, カーネル空間とユーザ空間を交えた処理の流れを説明する。

入出力時のシーケンス図を図 4 に示す。仮想ディスクに対して入出力を行うリクエストプロセスが, システムコールを発行してから, ネットワーク入出力が行われるまでの呼出し関係は図 4 のとおりとなっている。

- (1) デーモン vdiskd-xyz はあらかじめ起動しておく。vdiskd-xyz は, read システムコールを呼び出す。キャラクタデバイス /dev/vdiskio 内で制御をブロックし, リクエストが到着するまでスリープ状態に入る。
- (2) リクエストプロセスが, read や write システムコールなどを仮想ディスクに対して呼び出す。
- (3) カーネルはブロック I/O リクエストを, ブロックデバイス /dev/vdiskn の make\_request 関数にわたす。
- (4) スリープしている vdiskio を覚醒させる。
- (5) vdiskd-xyz へ制御を戻す。同時に, I/O リクエスト packets をわたす。
- (6) vdiskd-xyz は I/O リクエスト packets を解釈し, それにしたがってネットワーク入出力処理などを行う。
- (7) I/O レスポンス packets を write システムコールを使って vdiskio にわたす。
- (8) Linux カーネル API の bio\_endio で, ブロッ

表 2 開発・実行環境

CPU	Intel Pentium4 2.80EGHz
メモリ	512MB
OS	KNOPPIX 5.0.1 日本語版 ハードディスクインストール
Linux カーネル	2.6.17
ネットワーク	1000BASE-T で構成した LAN

- ク I/O の終了をカーネルに通知する。
- (9) リクエストプロセスに制御が戻る。最初へ戻る。

## 5. 実 装

### 5.1 開発・実行環境

Linux 用デバイスドライバを x86 PC 上で実装した。開発・実行環境を表 2 に示す。

### 5.2 デーモン vdiskd-xyz の実装

本研究において実装した, 特定のプロトコルに特化した 2 種のデーモンについて説明する。

**vdiskd-http-ftp** vdiskd-http-ftp は, HTTP と FTP に特化したデーモンである。デーモンプログラムの引数には http:// や ftp:// などで始まる URL を与える。デーモンはその場所に対して, ブロックファイルをダウンロード・アップロードする。SSL 接続・ユーザ認証に対応している。

**vdiskd-local** vdiskd-local は, ローカルディスクに対してブロックファイルの入出力を行うテスト用デーモンである。

## 6. 評 価

### 6.1 評価条件

本システムの性能を評価するため, スループットを計測するベンチマークテストを行った。ローカルディ

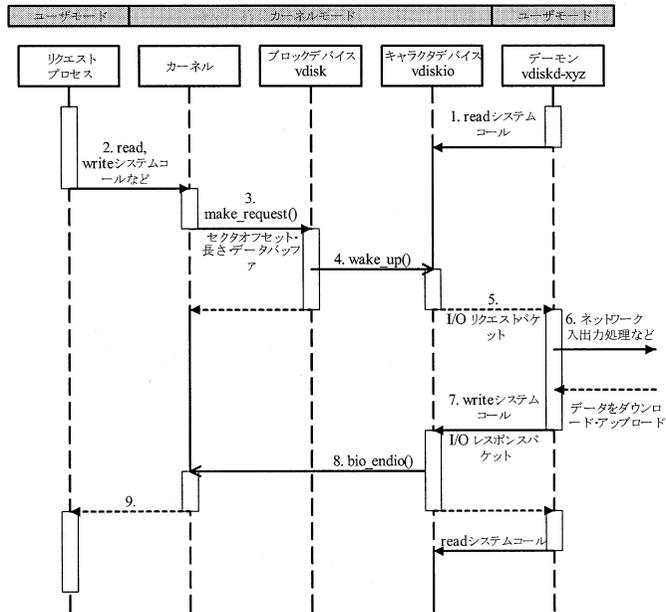


図 4 入出力時のシーケンス

Fig.4 Sequence at input/output

白い縦長の箱の部分は制御フォーカスがある部分、実線の部分はスリープ状態にある部分であることを示している。

スク, LAN 内の HTTP サーバ, LAN 内の FTP サーバを本デバイスドライバにより仮想ディスクとする。それぞれについて、仮想ディスクに直接入出力を行った場合と、同種の 2 台の仮想ディスクからなる RAID-0 ディスクアレイとして入出力を行う場合について、スループットを計測する。比較のため、ローカルファイルシステムに対して、本デバイスドライバを通さずに直接入出力を行う場合も計測する。

### 6.2 評価結果

ベンチマークの結果を図 5 に示す。ローカルディスクに対する入出力で、本ドライバを介する場合と介さない場合では、スループットが読み込み時で約 36%、書き出し時で約 59%低下している。この値は本ドライバのオーバーヘッドであるといえる。書き出し時のスループット低下がより大きいのは、本デバイスドライバではセクタよりも大きいブロック単位で入出力を行うために、書き出し操作に先立って読み込みを行うのが理由である。

RAID-0 (ストライピング) のディスクアレイ化した場合では、むしろソフトウェア RAID のオーバーヘッドが災いしたためか、多くの場合でスループットが低下している。

ネットワークを介した場合のローカル環境に対する

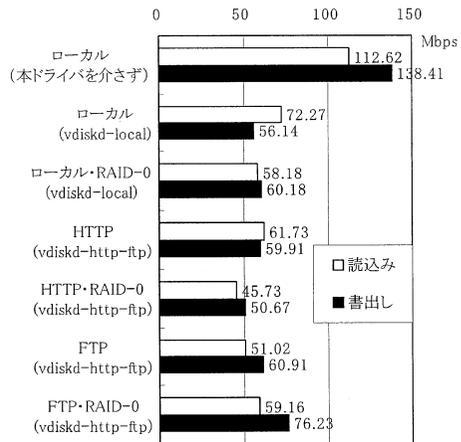


図 5 仮想ディスクデバイスのスループット

Fig.5 Throughput of the virtual disk devices

スループットの低下は、最悪でも約 21% (「ローカル・RAID-0」/「HTTP・RAID-0」の読み込み時) にとどまっている。ローカル環境と比較して低下率は小さく、いいパフォーマンスを得ることができた。

FTP の一部のケースでは、ネットワークを介してスループットが低下するのではなく、ローカルよりもスループットが高いケースがあった。本デバイスドライバは CPU の負担が大きいドライバで、ローカルディスクに対する読書きのパフォーマンスにも悪い影響を与える。一方、プロトコルのオーバーヘッドが少ない FTP では、むしろローカルよりもデータをより効率的に処理できたという可能性が考えられる。

## 7. 考 察

本研究では、ディスクインタフェースをブロックデバイスドライバとしたことで、高い透過性を提供できた。OS においてシステムコールレベルでの互換性があるため、単にユーザにとって物理ディスクデバイスと変わらない使い勝手を提供するだけにとどまらない。カーネルやドライバ、アプリケーションなどから見ても、ディスクデバイスとして振舞うため、既存のソフトウェアを利用した応用を考えることができる。応用としては、ホームディレクトリや OS そのものを本仮想ディスクシステムに載せる応用が考えられる。現在、HTTP-FUSE-KNOPPIX に本仮想ディスクシステムを組み込み、個人データを永続的に保存することのできる環境の実現に向けて試行している。

ベンチマークテストの結果から、本研究において実装したデバイスドライバのオーバーヘッドは、スループットの低下にして読込み時で約 36%、書出し時で約 59%であった。データの入出力においてユーザ空間を経由するため、ある程度のパフォーマンス低下は避けられない。しかし、ユーザ空間を経由する設計にしたことにより、外部ストレージとの入出力において複雑な処理を容易に行うことが可能となり、さまざまな種類の外部ストレージに仮想ディスクシステムを適用することができる。その利点を考慮すれば、この程度のパフォーマンス低下は妥当な結果である。

ネットワークを介した場合の仮想ディスクのスループットは、HTTP での読込み時に 61.73Mbps、書出し時に 59.91Mbps であった。FTP では読込み時に 51.02Mbps、書出し時に 60.91Mbps であった。これらの値は、一般家庭に提供されている光ファイバの実効速度程度であることを考えると、満足のいく値である。この程度の速度が得られれば、パフォーマンス最適化においては、スループットがネックになることはない。

今回はサーバを LAN に設置するという評価条件であった。インターネットを介した場合、レイテンシの影響によりいいパフォーマンス得られないことも考え

られる。ただ、この問題は先読みの実装やブロックサイズの調整である程度改善できると考えている。

本デバイスドライバは、ソフトウェア RAID の構築にも応用できる利点があった。ストライピングによる入出力の並列化でパフォーマンス向上を狙う RAID-0 のディスクアレイを、ベンチマークにおいて試した。しかし、FTP を除く多くの場合で、狙いとは逆にスループットが低下するという結果が出た。理由としては、本デバイスドライバの CPU 負荷が大きく、ソフトウェア RAID の動作に影響を与えていることが考えられる。デバイスドライバの設計・実装を改良して、高負荷時の動作を改善する余地がある。

## 8. 関連する既存システム・研究

本研究に関連する研究や、既存のシステムなどについて考察し、問題点を示す。また、それらに対しての本研究で開発したシステムの優位な点を述べる。

### 8.1 既存方式の点

個人データの利用においては、どの端末から、どの場所からということ意識せずに、またデータの存在している場所やデータの持ち運びについて意識せずに利用可能であることが理想である。しかし、そのようなユビキタスな利用には、現状では次に示す問題がある。

#### 8.1.1 持運び可能メディアによる方式

フロッピーディスク、CD-R、DVD-R、USB フラッシュメモリ、各種メモ리카ードなどの持運び可能な物理メディアに、個人データを保存して持ち運ぶことは古くからよく行われてきた。しかし、持運び可能メディアには、次のような欠点がある。

- 持ち運ぶ手間やコストがかかる。
- 紛失・盗難のリスクがある。
- 置き忘れると使えない。
- 遠隔地間でデータを同時に共有することができない。

持運び可能メディアの利用法は単純であるが、上に示すように扱いが煩雑で不便な点が多い。持運びやすさや利便性の観点からは、物理メディアの取り扱わずに済むのが望ましい。

#### 8.1.2 ネットワークによる方式

ここでは、ネットワークを介する各方式を分類し、それぞれの長所や短所を比較する。表 3 は、各方式を比較してまとめたものである。各方式の特徴について次に述べていく。

- NFS<sup>4)</sup>、SMB

NFS (Network File System)、SMB (Server Mes-

表 3 各方式の比較  
Table 3 Comparison of variety of methods

方式	透過的アクセス	インターネット利用	セットアップ
NFS, SMB	○	× LAN 内での利用を目的とした設計である。	× 専用のファイルサーバを用意する必要がある。
HTTP (WebDAV), FTP, SSH	△ ファイルシステムとしてマウントできない場合が多い。	○ ファイアウォール内部であってもほとんどの場合使用可能である。	△ ありふれたプロトコルのため導入のハードルは低い。ISP 提供のディスクスペースを利用することもできる。
オンラインストレージサービス	△ Web ブラウザや専用のアップロード・ダウンロードソフトを使用することが多いため透過性は低い。	○	○ 利用手続きさえすればいい。
iSCSI	○ SCSI デバイスとして扱うことが可能である。	△ ファイアウォールの内側からは利用できないこともある。	× 大規模なシステムが対象である。
HTTP-FUSE-cloop (読込み専用)	○	○	○ 通常の HTTP サーバで機能を果たす。
Petal	○	△ WAN での利用は視野にあるが、インターネットでの利用は想定していない。	× 専用の Petal ストレージサーバを用意する必要がある。

比較項目は次の通りである。

- 透過的アクセス** ユーザがローカルファイルシステムとの違いを意識せずにアクセス可能であるかどうか、あるいは入出力 API やシステムコールの互換性が高いかどうか。
- インターネット利用** その方式がインターネットでの利用に適しているかどうか。
- セットアップ** 個人ユーザが利用するという観点で、サーバ資源を調達しやすいかどうか、サーバやクライアントの設定が簡単であるかどうか、各項目について、優劣を次の記号で表す。○: 優れている / △: 一部制限がある / ×: 劣っている

sage Block) は、それぞれ UNIX, Windows で用いているファイル共有プロトコルである。ファイルシステムとして利用でき、透過性が高いという利点があるが、インターネットでの利用には適していない。

- **HTTP (WebDAV)<sup>5)</sup>, FTP<sup>6)</sup>, SSH<sup>7)</sup>**  
これらのプロトコルは、インターネット上でファイルを送受信するためによく用いられる。特に HTTP は利用頻度が高いことから、ファイアウォール内部であってもプロキシサーバ経由で利用できることが多い。
- **オンラインストレージサービス**  
インターネットのサーバにファイルを保管し、外出先などで取り出すことのできるオンラインストレージサービスを、さまざまな業者が提供している。サービス名としては「Yahoo! プリーフケース」や「ファイルバンク」が挙げられる。ユーザインタフェースの操作性がよくない、統一されていないという問題点がある。
- **iSCSI<sup>8)</sup>**  
iSCSI (Internet Small Computer System Interface) とは、通常 SCSI インタフェースを通じて送受信する SCSI コマンドを、インターネット経由で送受信するためのプロトコルである。iSCSI に対応したホストバスアダプタ (ホスト側のデバイス) やターゲット (ストレージ側のデバイス)

が市販されているが、個人ユーザが導入するにはコストが高い。

- **HTTP-FUSE-cloop<sup>9)</sup>**  
HTTP-FUSE-cloop は、HTTP サーバ上の分割ブロックファイルを再構成する仮想ブロックデバイスである。Linux ディストリビューションの一つである KNOPPIX の派生版の HTTP-FUSE-KNOPPIX において、インターネットを通じてシステムをブートするために用いている。インターネットを通じて仮想的なブロックデバイスを復元するという点では、本研究の目的に沿うものであるが、しかしこのシステムはブートのためのシステムファイル読み出しに特化しており、ユーザファイルをオンラインで保存することについては考えられていない。
- **Petal<sup>10)</sup>**  
分散ストレージシステム Petal は、冗長性確保、動的な再構成やバックアップなどをサポートし、高機能でありながらサーバの管理が容易なシステムである。しかし、インターネット経由での利用が想定されていない。また、専用のストレージサーバを用意する必要があるため、個人ユーザが利用するにはコスト、管理運用の点で不利である。ここで挙げてきたように、既存の方式にはそれぞれ何らかの問題点、例えばストレージサーバの準備にかかる手間やコストが大きい・個人ユーザの利用に適し

ていない・インターネット経由の利用を想定していない・透過性が低い、などが存在する。

## 8.2 本研究の優位性

本研究において開発したシステムでは、外部ストレージとの通信に用いるプロトコルとして HTTP (WebDAV), FTP を用いた。本デバイスドライバを用いることにより次の利点がある。

- 特定のプロトコルに依存せず、複数の異なる種類の外部ストレージを同時に統合的に扱える。
- Web ブラウザや FTP クライアントソフトウェアを使用するよりも、透過性が高い。ローカルディスクと変わらない扱いやすさを実現できる。

HTTP (WebDAV), FTP を本デバイスドライバを通さずに、Web ブラウザや WebDAV クライアント, FTP クライアントのようなアプリケーションを通して利用する場合は、このようなストレージデバイスとしての利便性は得られない。

その他の外部ストレージへの拡張においては、さまざまな種類の外部ストレージを考慮することができる。たとえば、各種オンラインストレージサービスや、ISP の提供するサーバのディスクスペースが考えられる。この対応により、ユーザのサーバ資源調達是非常に容易となることが期待できる。GmailFS<sup>11)</sup> のように、Gmail を外部ストレージとすることもできる。

## 9. おわりに

本研究では、個人ユーザの利用を意識した新たな仮想ディスクシステムを提案し、実装した。

- 本仮想ディスクシステムは、特定のプロトコルに依存せず、複数の異なる種類の外部ストレージを扱える設計とした。
- さまざまな種類の外部ストレージを利用可能とすることにより、ユーザが多様な構成を選択することができる。そのため、ユーザのサーバ準備・管理運用の手間やコストを最小化することができる。
- デバイスドライバとして実装することにより、ユーザや既存のソフトウェアに対して高い透過性を提供でき、ネットワーク経由であることを意識させない利用を可能とした。
- ファイルシステムレベルではなく、ブロックデバイスレベルの実装とした。それにより、設計・実装がファイルシステムドライバ開発に比べ単純にでき、またファイルシステムやソフトウェア RAID といった、ストレージデバイスのための既存の枠組みをそのまま適用することが可能となった。
- パフォーマンスの測定においては、仮想ディスク

ドライバのオーバーヘッドは、読み込み時のスループット低下にして約 36% と小さい。ネットワークを介した場合も、最悪でもローカル環境の 78% 以上のスループットが得られた。満足のいくパフォーマンスを得ることができた。

## 参考文献

- 1) Østergaard, J.: The Software-RAID HOWTO, <http://www.linux.or.jp/JF/JFdocs/The-Software-RAID-HOWTO.html>.
- 2) Corbet, J., Rubini, A., Kroah-Hartman, G., 山崎康宏, 山崎邦子, 長原宏治, 長原陽子: Linux デバイスドライバ, オライリー・ジャパン, 3rd edition (2005).
- 3) FUSE: Filesystem in Userspace, <http://fuse.sourceforge.net/>.
- 4) Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M. and Noveck, D.: *RFC3530: Network File System (NFS) version 4 Protocol* (2003).
- 5) Goland, Y., Whitehead, E., Faizi, A., Carter, S. and Jensen, D.: *RFC2518: HTTP Extensions for Distributed Authoring - WEBDAV* (1999).
- 6) Postel, J. and Reynolds, J.: *RFC959: FILE TRANSFER PROTOCOL (FTP)* (1985).
- 7) Ylonen, T.: SSH - Secure Login Connections over the Internet, *Proceedings of Sixth USENIX Security Symposium, Focusing on Applications of Cryptography* (1996).
- 8) Meth, K. Z. and Satran, J.: Features of the iSCSI protocol, *IEEE Communications Magazine*, Vol. 41, No. 8, pp. 72-75 (2003).
- 9) 金井遵, 須崎有康, 八木豊志樹, 並木美太郎: HTTP-FUSE-cloop のソフトウェア RAID による高速化と耐故障性の実現, インターネットコンファレンス (2006).
- 10) Lee, E. K. and Thekkath, C. A.: Petal: Distributed Virtual Disks, *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, pp. 84-92 (1996).
- 11) Jones, R.: Gmail Filesystem, <http://richard.jones.name/google-hacks/gmail-filesystem/gmail-filesystem.html>.