

個人向け管理サービスを提供するための ソフトウェア基盤の構築

宮澤 元

南山大学 数理情報学部

E-mail: miyazawa@it.nanzan-u.ac.jp

本稿では、広帯域ネットワークを前提に、個人向けに管理サービスを提供するためのソフトウェア基盤の概要を示す。計算機と計算機ネットワークが安価に提供されるようになり、個人ユーザが自宅で常時ネットワークに接続するような環境が普及しつつある。一般に個人ユーザは計算機や計算機ネットワークの管理に関する知識に乏しく、日常の計算機管理作業や、ネットワーク接続のための管理作業などを適切にこなすことは困難である。本研究の目的は、これらの個人ユーザを対象に、近年普及しつつあるケーブルテレビ(CATV)回線や Digital Subscriber Line (DSL) などの広域かつ広帯域なネットワークを介して計算機管理サービスを提供するシステムを開発することである。このようなシステムを開発するに当たって、分散ファイルシステムの技術を応用することにより、ファイルのバックアップやアプリケーションの提供を行うことができる。一方、ネットワークを介して全般的な計算機管理サービスを提供するためには解決しなければならない課題も多い。

1 はじめに

計算機と計算機ネットワークが安価に提供されるようになり、個人ユーザが自宅でネットワークに常時接続できる環境が普及しつつある。例えば、CATV回線や、Asynchronous DSL (ADSL) に代表される xDSL を利用したインターネット常時接続サービスの利用者は近年急増している。この影響で、従来の Integrated Services Digital Network (ISDN) 回線を利用した常時接続サービスも開始されたのに加え、CATV や ADSL よりもさらに広帯域の通信が可能となる光ファイ

バーを利用した個人向けの常時接続サービスも開始予定である。

インターネットに常時接続できる環境では、各計算機が適切に管理・運用されている必要がある。このような管理・運用の作業は、計算機と計算機ネットワークに関する専門的な知識を必要とするので、例えば、従来からインターネットに常時接続されていた大学や企業など、ある程度大規模な組織では計算機と計算機ネットワークの管理・運用のために専門の管理者を置くのが普通である。

一方、一般の個人ユーザには常時接続回線を

利用する上での管理・運用の技術を期待できない。このようなユーザが利用する計算機は、ネットワークからの侵入などのセキュリティ的な危険にさらされる他、逆にソフトウェア設定の不備でネットワークに接続されている他の計算機に被害を与える可能性もある。また、アプリケーションソフトウェアのインストールや保守、ファイルのバックアップといった、計算機を利用する上で必要な基本的な管理作業を行うことができなかつたり、おろそかにしていることが多い。

本研究では、個人向けの常時接続回線を介して、個人ユーザの計算機に対して管理サービスを提供するためのソフトウェア基盤を開発することを旨とする。本研究で想定する管理サービスには以下のようなものがある。

- ユーザが作成したファイルのバックアップ
- アプリケーションソフトウェアの自動インストール/保守
- 計算機のソフトウェア設定の自動化

本稿では、このシステムの概要を示す。ユーザが作成したファイルのバックアップやアプリケーションの自動インストールに関しては、分散ファイルシステム技術を応用して実現する方針であるが、現在のところまだ実装は行っていない。また、ソフトウェアの設定など、これ以外の管理サービスをネットワークを介して提供するためには、分散ファイルシステムの技術を単純に応用するだけでは解決しない問題も多く、解決の目処はついていない。

以下、2節では関連研究について述べる。本研究で開発するシステムとその概要に関しては3節で述べる。4節では本システムの実現方式、有用性、課題などについて検討を行い、5節を本稿のまとめとする。

2 関連研究

本節では、広域分散ファイルシステムをはじめとして、広域ネットワークにおける関連研究について紹介する。本研究で対象とするネットワークはローカルエリアネットワーク(LAN)に比べて通信帯域が狭く、LAN上の技術を直接適用するのは困難であると考えられるからである。

2.1 AFS

広域ネットワークを対象とした分散ファイルシステムには、Carnegie-Mellon 大学で開発された Andrew File System (AFS)[8, 3, 9]がある。AFSは、広域分散環境においてもスケラブルなシステムを構築することを主眼に設計されているファイルシステムであり、サーバやネットワークの負荷を低減することを目的とした設計が行われている。AFSでは、サーバの負荷を低減するために、ファイルの全体をクライアントのローカルディスク上にキャッシュし、全てのファイル操作をキャッシュに対して行なう。ファイルアクセスのセマンティクスにセッションセマンティクスを採用しており、ファイルが更新された場合のみサーバにキャッシュを書き戻し、サーバは更新されたファイルをキャッシュしている別のクライアントに対してキャッシュを破棄するよう通知する(callback)。これによりファイルアクセスのセマンティクスをUNIXセマンティクスより弱めることになるが、サーバの負荷だけでなく、ネットワークトラフィックも大幅に低減できる。

2.2 Coda

AFSにおけるクライアント側のキャッシュシステムを発展させることにより、移動体計算機に対しても分散ファイルシステムを提供するのがCodaファイルシステム[4, 7]である。Coda

では、クライアントがアクセスしたファイルを単純にキャッシュするのに加え、あらかじめ指定しておいたファイルに関してはアクセスされなくてもクライアント側に転送してクライアント計算機のディスク上にキャッシュする hoarding と呼ばれる動作を行う。Hoarding を行っておけば、計算機がネットワークから切断されている場合にも、キャッシュにアクセスしてファイルに対する操作を継続することができる。また、各ファイルに hoarding の優先度を設定し、ファイル転送に十分なネットワーク帯域を使用できない場合、優先度に応じたファイル転送を行う。ネットワークから切断されている時にファイルに対して行われた更新操作は、ネットワークに再度接続されたときにサーバに対して書き戻すようになっている。このとき、クライアント計算機がネットワークから切断されている間に、同じファイルが他のクライアントに更新されていたとすると、ファイルの内容の不整合が生じる。このような場合、システムはある程度までは自動的に整合性を取ろうとする [5] が、これができない場合、内容の不整合が生じたことをユーザに知らせ、ユーザにファイル内容の整合性の解決をまかせる。

2.3 WWW

World-Wide Web (WWW) は、広域ネットワークにおける情報提供システムとして広く使われているクライアントサーバ方式のシステムである [2]。データ転送だけを見ると、本質的には分散ファイルシステムと良く似たシステムであるが、不特定多数のユーザに対してデータを提供するため、基本的には読み出し専用アクセスしか提供しない¹点が異なる。

¹WWW のデータ転送プロトコルである Hyper-Text Transfer Protocol (HTTP) 自体には、書き込みを行うためのプロトコルも定義されているが、ほとんど利用されていない。

3 管理サービス提供システム

本システムでは、管理・運営サーバが広帯域回線を介して接続されたクライアント群に対して管理サービスを提供する。クライアント計算機としては個人所有の計算機を想定している。一方のサーバ計算機は、典型的にはインターネットサービスプロバイダ (ISP) 等、個人ユーザにネットワーク接続を直接提供する組織に置くことを想定している (図 1)。

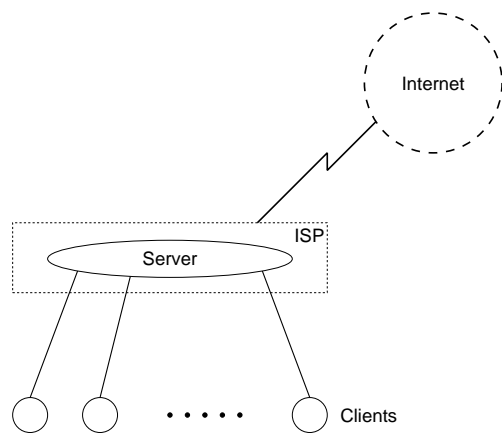


図 1: システムの概念図

3.1 サーバ計算機の概要

サーバ計算機は、自分が持つディスク領域を、クライアント計算機に対して以下の 2 種類の方法で参照可能とすることができる。

1. 書き込み専用参照

サーバ計算機に各クライアント計算機ごとの専用ディスク領域を用意して、この領域を分散ファイルシステムのプロトコルを利用して各クライアント計算機に提供することにより、クライアント計算機が書込むファイルのバックアップを自動的に行う。書込

まれたファイルは、クライアント計算機側のディスクにもキャッシュされており、通常クライアント計算機はこのキャッシュのみを参照して作業を行う。サーバ側に書き込まれたファイルは、通常の読出し操作によってクライアント計算機から参照することはできない。このようにすることにより、各クライアント計算機が別のクライアント計算機に割当てられた領域を相互に参照することも防ぐことができる。なお、クライアント計算機の障害等によりクライアント計算機上にキャッシュされたキャッシュが失われた場合、専用のプロトコルを用いてバックアップファイルからキャッシュ内容を復旧する必要がある。

2. 読出し専用参照

サーバ計算機にアプリケーションをインストールしたディスク領域を用意し、これを分散ファイルシステムのプロトコルを利用してクライアント計算機に提供する。ディスクのパーティション単位ではなく、ディレクトリ(あるいはファイル)単位で参照可能情報を設定することができる。クライアント計算機のライセンス情報を使って、アプリケーションごとの参照可能情報を適切に設定すれば、各クライアントごとに利用できるアプリケーションを制限することができる。

3.2 クライアント計算機の概要

クライアント計算機は、サーバ計算機が参照可能としたディスク領域をマウントすることができる。マウントした領域に含まれるファイルはローカルディスク上にキャッシュされ、これらのファイルに対する操作は全てキャッシュに対して行われる。

マウントした領域をサーバが書込み専用と読

出し専用のどちらで参照可能としているかによって、ファイルを読み書きする際の動作が異なる。

- サーバが書込み専用で参照可能としている場合
 - クライアント側で読み書き可能でマウントすれば、この領域に対して読出しと書込みを行うことができる。
 - － 読出し
 - ローカルディスクのキャッシュ上に必要なファイルがあればそのファイルを読出す。ファイルがなければエラーとして処理し、サーバ計算機にはそのファイルに対するアクセス要求を出さない。
 - － 書込み
 - ローカルディスクのキャッシュ上に書込みが行われ、close 時にサーバに転送される。この場合、ファイルはローカルディスク上にはそのまま記録されるが、サーバには暗号化された上で転送される。これは、個人情報を保護するためである。暗号化の詳細については未定だが、パスワード情報などをクライアントのみに格納し、この情報をキーとして暗号化すればよいだろう。
- サーバが読み出し専用で参照可能としている場合
 - クライアント側で読み書き可能でマウントすれば、この領域に対して読出しと書込みを行うことができる。
 - － 読出し
 - ローカルディスクのキャッシュ上に必要なファイルがあればそのファイルを読出す。ファイルがなければサーバ計算機にそのファイルに対するアクセス要求を出す。サーバから転送されたファ

イルはローカルディスクにキャッシュされる。

- 書込み
ローカルディスクのキャッシュ上に書込みが行われる。サーバ計算機には書込み要求を出さない。

3.3 運用例

本システムを利用して、管理サービスを運用する例を示す。

- ファイルのバックアップ
Unix のホームディレクトリ (例えば /home) や Windows の \My Documents のように、ユーザが個別に作成したファイルを格納するための領域は、サーバ計算機が書込み専用で参照可能とした各クライアント専用のディスク領域をマウントし、そこに作成する。この領域に作成/書込されるファイルの内容は、close 時にサーバ計算機にも転送され、バックアップされる。このとき、ファイル内容は暗号化されるので、ユーザの情報は保護され、サーバ側で参照されることはない。
- アプリケーションの利用
サーバ計算機にアプリケーションをインストールし、これを読み出し専用で利用可能とする。クライアント計算機は、これを /usr/local や \Program Files にマウントして利用する。アプリケーション単位で参照可能情報を設定できるので、ライセンス情報などを利用して、特定のクライアントには特定のアプリケーションを利用させないことも可能である。例えば図 2 では、クライアント A は MS-Office2000 や Photoshop のライセンスを持っていて利用できるのに対し、クライアント B はこれらのラ

イセンスを持っていないので、サーバはこれらのアプリケーションを参照可能とせず、利用できない。クライアント A,B がともにライセンスを持っている AcrobatReader は両方で利用できる。

なお、この領域に対してクライアントで行われた書込みはサーバには反映されないで、各クライアントが独自にアプリケーションをインストールすることも可能であろう。

4 議論

この節では、本システムの有用性や限界、性能などに関して議論を行う。

4.1 アプリケーションソフトウェアの提供

本システムでサーバ計算機がクライアント計算機に対してアプリケーションを提供する方式は、従来の分散ファイルシステムでサーバ計算機上のアプリケーションを共有していたのと同様と同じ方式である。本システムの方式が従来と異なるのは、エクスポートするアプリケーションを制御する機能を備え、クライアント計算機ごとに異なったアプリケーション群を提供する点である。

従来の方式は、LAN などの閉じた環境で利用された方式で、クライアント計算機ごとにアプリケーションの利用資格が異なるといった事態は想定されていない。これに対し、本システムが想定している環境では、それぞれのクライアント計算機が個人ユーザに所有されており、どのアプリケーションを利用したいか、あるいは利用できるか(=ライセンスを持っているか)はクライアント計算機ごとに異なる。

本システムを用いることにより、ユーザはソフトウェアのインストール作業やバージョンアップ作業などのアプリケーションの管理を外部の

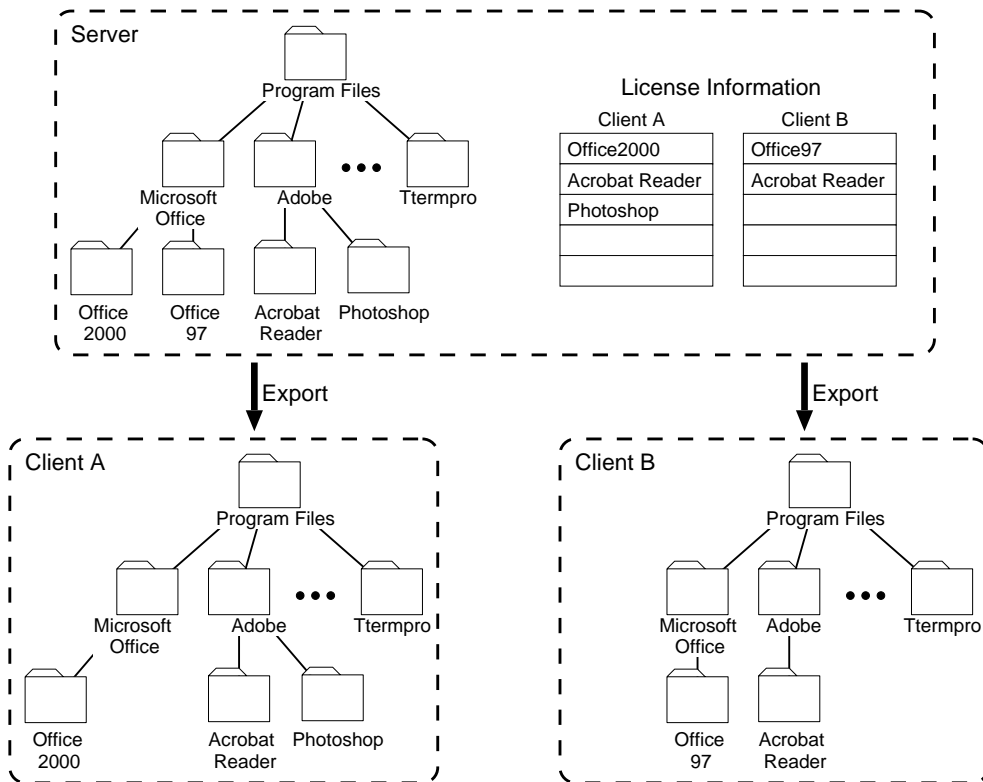


図 2: 参照可能情報の制御

専門業者に任せることが可能となる²。例えば、ISP がインターネット接続サービスに加え、このようなアプリケーション管理サービスを提供するようになれば、ユーザはソフトウェアの利用ライセンスを購入するだけでインストール作業やバージョンアップ作業などを自分で行う必要がなくなり、ソフトウェアの導入コストを下げるができる。

²現在の商用ソフトウェアのライセンス規定ではこのようなことはできないことが多い。

4.2 OS コンポーネントの提供

本システムを、OS のバージョンアップなど、システムソフトウェアのバージョン管理に応用できないか考察してみたい。

現在実用的に利用されている OS の多くはモノリシックに構成されているので、OS の各機能は不可分であり、これをネットワーク経由でバージョンアップすることは困難であると予想される。しかし、近年では Mach[1] などで開発されたマイクロカーネルの技術を利用した OS が実用に供されるようになってきている。このようなアプローチで OS を適切に構成できれ

ば、動作中のOS機能の一部を停止してバージョンアップするようなことも可能となるだろう。

また、従来ディスクレスワークステーション[6]で利用されたネットワークブートの技術を用いれば、サーバ計算機に置かれたOSをバージョンアップしておき、クライアント計算機の起動時に新しいバージョンのOSを起動する形でバージョンアップできる。本システムのようなものを考えなくても比較的簡単にOSのバージョンアップが可能かもしれない。もちろん、想定している環境では、各クライアント計算機ごとにハードウェア構成などが大きく異なるので、従来のネットワークブートの技術がそのまま利用できるかどうかは不明である。

4.3 計算機設定作業の提供

本システムを利用することにより、ファイルのバックアップやアプリケーション管理など、ファイルを計算機間で共有することで解決するような管理サービスをクライアント計算機に対して提供することができる。しかし、計算機の管理という観点からもっとも厄介なものは、計算機のハードウェアおよびソフトウェアの設定である。

アプリケーションの設定など、サーバ計算機がクライアント計算機ごとに用意した設定ファイル等を提供することで解決するようなものもあると思われるが、各ユーザの設定に関する希望をどのように取り入れるべきかという問題がある。また、例えばネットワーク接続そのものに関わる設定作業など、OSを初めとするシステムソフトウェアの設定作業や本システム自体の設定作業を外部からネットワークを介して提供することは困難である。

計算機設定に関しては、ハードウェアなどでは自動設定を行い、それ以外は専門知識の乏しいユーザでも設定可能なようなインターフェー

スを設計するべきなのかもしれない。

4.4 性能面の検討

本システムを、ISPを中心として運用することを考えると、数千から数万台規模のクライアント計算機に対してサービスを提供する必要がある。このような大規模なネットワークに管理サービスを提供するには、サーバ計算機の負荷やネットワークの負荷が問題となる。

サーバ計算機の負荷を低減するためにはサーバを並列化して、1台あたりの負荷を軽減する[10]などの手法を利用することもできるだろう。一方、多数のクライアントのデータを全てバックアップするためには、近年のディスク装置の大容量化・低価格化を考慮したとしても非常に多数のディスク装置が必要となる。必要なディスク装置を減らすために効率的な圧縮技術等も開発しなければならないかもしれない。

動画や音声などのマルチメディアデータを効率的に扱うには、現在のCATV回線やxDSLでは帯域が不十分である。そのため、データ転送のプロトコルを工夫し、できるだけネットワークに負荷をかけないかたちでデータ転送を行う必要がある。また、将来的には光ファイバーを用いた回線の利用も検討する必要があるだろう。

5 まとめ

本稿では、個人向けの常時接続回線を介して、個人ユーザの計算機に対して管理サービスを提供するシステムに関して、その概要を述べた。本システムでは、分散ファイルシステムの技術を応用することにより、以下のような管理サービスを提供することができる。

- ユーザが作成したファイルのバックアップ
- アプリケーション管理

また、本システムの特徴や、上記以外の管理作業を提供する方法などについても議論を行った。

謝辞

この研究は、南山大学パツへ研究奨励金 (Pache Research Subsidy)I-A の助成を受けています。

参考文献

- [1] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young. Mach: A new kernel foundation for UNIX development. In *USENIX Summer Conference Proceedings*, pages 93–112. USENIX Association, 1986.
- [2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext transfer protocol – http/1.1. Technical report, Network Working Group RFC 2068, 1997.
- [3] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems*, 6(1):51–81, February 1988.
- [4] James J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. In *Proceedings of 13th ACM Symposium on Operating Systems Principles*, pages 213–225, October 1991.
- [5] Puneet Kumar and M. Satyanarayanan. Flexible and Safe Resolution of File Conflicts. In *Proceedings of the Usenix Winter 1995 Technical Conference on Unix and Advanced Computing Systems*, January 1995. available via ftp (CMU-CS-94-214).
- [6] Edward D. Lazowska, John Zahorjan, David R. Cheriton, and Willy Zwaenepoel. File Access Performance of Diskless Workstations. *tocs*, 4(3):238–268, August 1986.
- [7] Lily B. Mummert, Maria R. Ebling, and M. Satyanarayanan. Exploiting Weak Connectivity for Mobile File Access. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, pages 143–155, December 1995.
- [8] M. Satyanarayanan, John H. Howard, David A. Nichols, Robert N. Sidebotham, Alfred Z. Spector, and Michael J. West. The ITC Distributed File System: Principles and Design. In *Proceedings of the 10th ACM Symposium on Operating Systems Principles*, pages 35–50, December 1985.
- [9] Mirjana Spasojevic and M. Satyanarayanan. An Empirical Study of a Wide-Area Distributed File System. *ACM Transactions on Computer Systems*, 14(2):200–222, May 1996.
- [10] 宮澤 元. 大規模分散ファイルシステムの性能測定. 南山経営研究, 第14巻(3号):487–501, 3月 2000年.