

動的なネットワーク環境に適した 適応型オンラインストレージシステムの提案

井上 知洋 中村 元紀 久保田 稔 †

将来実現されるユビキタスネットワークは、様々な性質のネットワークが混在した不均一なネットワーク環境になることが予想される。特に無線リンクなどによってアドホック的に作られるネットワークが今後普及するにつれ、ネットワーク単位の移動や分離など、ネットワークトポロジの動的な変化への対応が求められる。我々は、このような環境においてもシームレスなデータアクセス環境を提供する、環境適応型オンラインストレージシステムを提案する。このシステムでは、データの一貫性制御のための管理権限を任意の端末上に移動可能とすることによって、トポロジの変化に起因するデータの可用性低下を抑えることが出来る。

A Proposal of Adaptive Online Storage System for Dynamic Networks Environments

Tomohiro INOUE Motonori NAKAMURA Minoru KUBOTA †

The future ubiquitous network will be a heterogeneous complex of various types of networks such as hotspot networks or ad-hoc networks. Following to the spread of networks which are consisted of radio links, the network mobility (movement of a network it self) will cause dynamic changes of the network topology. We propose a adaptive online storage system in which the management token of replicated data can move to any nodes in the network, and discuss advantages of the proposed system about the availability of data access in dynamic networking environments.

1 はじめに

1.1 オンラインストレージ

遍在するコンピュータやネットワーク上のリソースをいつでも利用することが可能なユビキタスネットワーク社会の実現が期待されて久しいが、現在の一般的なユーザのネットワーク利用形態は依然として限定されたものにとどまっている。つまり作業用端末として個人のノートPCやPDAなど特定の端末を携帯し、情報参照やメール送信のために必要に応じてネットワークへ接続するという利用形態である。このような利用形態の場合、ユーザは特定の端末に縛られておりネットワークがユビキタス化する恩恵を十分に受けられていない。真にユビキタスなネットワークが普及した環境では、ユーザは周囲に存在するどのような端末からもネットワークを利用したサービスを自由に受けられることが望ましい。

現在、ユーザがこのように特定端末に依存した形でネットワークサービスやアプリケーションを利用している原因として、サービス利用に必要なユーザの環境情報（ユーザのプロファイルやアプリケーションの設定情報など）がユーザ端末に強く結びつけられていることが考えられる。例えば、任意の端末から普段使っている電子メールを送受信することを考える。このときユーザは、ユーザ

の名前やメールアドレス、メールサーバのアドレス、ポートなど様々な情報を利用する端末上のメールアプリケーションに対して新しく設定しなければならず、非常に手間がかかる。また、ユーザが普段利用しているメールボックスにはIMAPなどを利用しない限りアクセスできない。

一方Webブラウザを利用したネットワークサービスでは特定のURLのみをブラウザに指定すればサービスを利用することが可能であるため、Personal Portal [1]などのサービスと組み合わせることによってどのような端末からも同じような環境で作業することができる。しかし現状では情報の閲覧が主な利用方法であり、文書作成のように情報の更新を頻繁に行うような作業には適していない。また、移動中などネットワークとの接続を確保できないときにはサービスそのものを利用することができない。

もしURLのようなグローバルユニークな識別子のみを用いてアクセスできるネットワーク上の環境情報保存サービス（ストレージサービス）が存在すれば、ユーザは特定の端末へ縛られず、より自由にネットワークサービスを受けることが可能だと考えられる。そのためにはユビキタスネットワーク環境での有用なオンラインストレージサービスが必要となる。

1.2 将来のネットワーク環境

将来のネットワークは、Home NetworkやCellular Network, Hot Spotなど様々な形態のネットワークが固定ネットワークを介して連結され

† 日本電信電話（株）NTT未来ねっと研究所，
NTT Network Innovation Laboratories, NTT Corporation

た、不均一性の強い動的なネットワークになると考えられる。今後、Adhoc Network や PAN (Personal Area Network) など無線リンクで構成されるネットワークが普及するにつれ、ネットワークポロジが動的に変化するネットワーク環境が一般化すると考えられる。

また、今後普及するネットワーク形態を考えると、インターネットや様々な固定ネットワークへ常に接続可能とする方向性だけでなく、ある特定の場所に集まった端末間で一時的に構築される Adhoc Network のように、常にあらゆる場所でネットワークを動的に構築する方向の発展も期待されている。それらの一時的なネットワークは、固定ネットワークと接続されるかもしれないし孤立しているかもしれない。あるいは、移動している乗り物内の端末によって構成されるネットワークのように、接続点を逐次変更しながら固定ネットワークと接続と孤立を繰り返しているかもしれない (いわゆる Network Mobility [2])。

このような将来のネットワーク環境でオンラインストレージを利用することを考えると、各端末が常に固定ネットワークと接続しているとは限らない状態で、そのことをユーザやアプリケーションに意識させない技術が必要となってくる。そこで我々は、ユビキタスネットワーク環境で利用するオンラインストレージとして、環境適応型オンラインストレージシステム CAOSS (Circumstances Adaptive Online Storage System) を提案している [3]-[5]。

本稿では CAOSS の構成とデータ管理方式の概要について説明している。以下、2節では前述のようなユビキタスネットワーク環境における既存ストレージシステムの課題について説明する。そのような課題を解決するために提案している CAOSS のシステム構成とその効果について3節および4節で説明する。5節で関連研究について触れ、最後に6節でまとめと今後の方針を示す。

2 既存ストレージの課題

2.1 キャッシュデータの同期

ネットワーク経由でのファイルアクセスや環境共有のために、古くから NFS などのファイルサーバが用いられてきた。しかし NFS などは比較的小規模、あるいは十分に高速なネットワーク上での運用を前提としているため、インターネットのような広域ネットワーク環境では利用することができない。

このため、広域ネットワーク環境でのファイルアクセスのパフォーマンス向上を目標に AFS や Coda [6] などのキャッシュを用いる分散ファイルシステムが開発されてきた。これらのシステムではクライアント端末上にファイルのキャッシュを配置し高速なファイルアクセスを可能にしている。

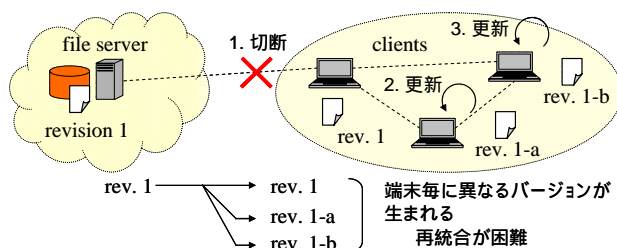


図1 分散ファイルシステムの切断時操作

特に Coda ではクライアント端末がネットワークから切断された場合にもキャッシュファイルへの更新を行うことが可能な切断時操作をサポートしている。この場合、切断時にキャッシュファイル上で更新されたデータはファイルサーバとの接続復帰時にオリジナルファイルへと同期される。Coda では更新が衝突した場合には人手で更新の整合性をとるという楽観的な同期のポリシーを採用している。

このようなキャッシュを用いる分散ファイルシステムを前節で述べたような動的なネットワーク環境に適用することを考える。このシステムではファイルサーバが全てのクライアント端末のキャッシュデータを管理し、キャッシュ間の整合性を維持する責任を負っている。このファイルサーバはあくまで固定の端末であるため、ストレージを利用している複数のクライアントを含むネットワークがファイルサーバとの接続を失った場合、キャッシュデータ間の一貫性を管理することができない。

Coda のように切断時操作を行うことは可能であるが、この操作はあくまでも各々のクライアント端末単位で行われる。そのため、図1で示すようにそれぞれの端末上のキャッシュデータがお互いに独立に更新されてゆき、バージョンの異なるデータが端末の数だけ生まれることになる。結果として、ファイルサーバとの接続が復帰してキャッシュデータの同期が行われたときに更新内容の衝突が発生する可能性が高くなってしまふ。更新の衝突が発生した場合には人手によって整合性を回復する作業が必要となるため、衝突回数の増加は衝突回復コストの増大を招き、ストレージとしての可用性を大幅に低下させる。

2.2 複製同期の柔軟性

キャッシュを用いる既存の分散ファイルシステムでは、生成されるキャッシュはシステム規定のポリシーで同期され、また全ての端末について平等に扱われる場合がほとんどである。例えば Coda では、いずれかの端末からファイルサーバ上のファイルが更新された場合、サーバが他の端末上の全てのキャッシュに通知を送り、キャッシュを全て無効化するという動作を行う。しかし将来は、PDA やハードウェアリソースのより小さなデバイスなど、様々な種類の端末がネットワークに接続

されるようになると予想される。各端末も、ミッションクリティカルなミラーリングから優先度の低いアーカイブまで、その役割に応じて様々なレベルの同期ポリシーを要求すると考えられる。このため、ストレージシステムを構成する端末の要求は、確実なデータ同期の保証から、キャッシュデータを用いないデータアクセスまで、様々なレベルに渡ると考えられる。既存の分散ファイルシステムなどではこのような要求を満たすことが出来ない。

2.3 ファイル単位の同期

既存の分散ファイルシステムにおけるキャッシュの生成と同期は、ユーザがファイルサーバ上のファイルにアクセスした時点で稼働し、キャッシュされる情報は基本的にアクセスしたファイルのみを対象とする。しかし多くのアプリケーションでは、意味的な依存関係を持つ複数のファイルを利用しているため、このうちの一部のファイルだけがキャッシュされている、あるいは最新の状態である、というだけではアプリケーションから利用する情報に不整合が生じる。一方 CVS [6] などのバージョン管理システムでは、ディレクトリツリーなどのファイル集合をレポジトリとして登録し、レポジトリの単位で同期を行うため、ユーザはレポジトリにファイルを登録していれば一度の操作によって必要なデータを全て同期することが出来る。しかし、CVS のようなシステムではユーザが指示したタイミングでのみ同期を行うため、トポロジの変化する動的なネットワーク環境で必要な自動的な同期機能を備えていない。

3 CAOSS

筆者らは、前述のような問題を解決し、ネットワークトポロジの変化による更新操作の可用性低下を抑えることのできるストレージシステム、CAOSS を提案してきた。以下では CAOSS の構成要素および複製データの管理方式について説明する。

3.1 構成要素

CAOSS はネットワーク上のサーバやユーザ端末など任意の端末をストレージとして利用するオンラインストレージシステムである。以下に CAOSS の構成要素を示す (図2)。

デポ (Depot): 意味的に依存関係を持つ複数のデータを収容した更新の単位。デポはネットワーク内で一意な識別子であるデポ ID (D-ID) で識別される。

コアデポ (Core Depot): 更新の権限を持ったデポ。一つのデポ ID に対して、(接続されている) ネットワーク内で唯一存在する。コアデポは後述の複製デポのリストを管理し、更新があった時には把

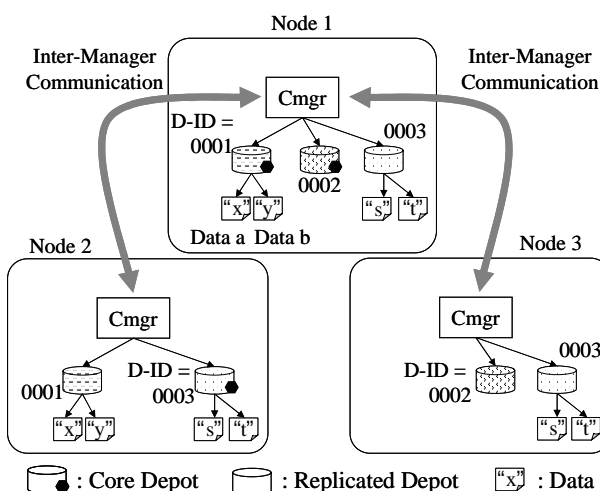


図2 CAOSS の構成要素

握している複製デポに更新内容を通知する。

複製デポ (Replicated Depot): 読み出しのみ可能なデポ。一つのデポ ID に対してネットワーク内で0個以上複数存在し得る。複製デポはコアデポと協調して、お互いの役割を入れ替えることが出来る。つまりマスターデータであるコアデポは、CAOSS を構成する任意の端末に移動することが出来る。

また、複製デポはコアデポとの情報の整合性の度合いとして、「複製レベル」を選択することができる。複製レベルは以下の三つのうちのどれかであり、また複製デポは任意のタイミングで複製レベルを変更できる。

- **強整合:**
コアデポと常に同期がとれていることが保証されている複製デポ。
- **弱整合:**
コアデポと同期がとれていることが期待される複製デポ。
- **キャッシュ:**
以前参照したコアデポのデータを記憶している複製デポ。分散ファイルシステムなどで言うキャッシュに相当する。

CAOSS マネージャ (CAOSS Manager): CAOSS のストレージを構成する端末上に必ず存在するプロセスで、デポの操作の受け付けや分散管理を行う。以下 Cmgr と表記する。

また、上に挙げた構成要素以外にも CAOSS を実際に利用するに当たっては、デポを識別するデポ ID を、実際に存在するコアデポあるいは複製デポのネットワークアドレスに変換する名前解決サービスの存在が必要となる。しかし CAOSS の前提としているトポロジの動的なネットワーク環境では、DNS のような固定の端末の存在する名前解

決サービスを利用することが出来ない．このため、現在盛んに研究が進められている Chord [8] や CAN [9] などの P2P 系の名前解決サービスの利用を想定している．

3.2 データ管理方式

CAOSS ではデポの生成，更新，読み出し処理を以下のように実行する．

- ある端末の Cmgr に新規デポの生成が要求されると、その Cmgr は自端末にコアデポを生成する (図3 (a))．デポ生成後そのデポ内に任意のデータを保存することができる．
- コアデポの存在しない端末でそのデポ内のデータに対する読み出し要求を受けた Cmgr は、コアデポを管理する Cmgr からデータを読み出し、複製デポを生成して要求に応える (図3 (b))．このとき生成される複製デポの複製レベルはデポ規定の値を用いて設定される．また、各複製デポには有効期間が設定される．
- コアデポの存在する端末でそのデポ内のデータに対する更新要求を受けた Cmgr は、全ての強整合複製デポを管理する Cmgr と交渉して更新を実行し、複製にその更新を反映する (図3 (c))．このとき、コアデポから通信が不可能な有効期間内の強整合な複製デポが存在する間は、デポの更新操作は失敗する．デポの更新操作の成功後、弱整合な複製データに対して更新を反映させ、同時にキャッシュ複製データに対して更新を通知し、キャッシュの無効化を行う．弱整合およびキャッシュ複製データに対する通信は一方方向のメッセージとして送信するため、ネットワーク切断などの理由によりメッセージが失われた場合は複製データの最新性を担保できない．
- CAOSS を利用するクライアントから、有効期間内の強整合な複製デポへと参照要求があった場合、複製デポはコアデポへの通信なしに最新性の保証されたデータを読み出すことが可能である．弱整合な複製デポもコアデポへの通信なしに最新のデータを読み出すことが可能であるが、この場合データの最新性について厳密には保証されない．キャッシュ複製デポは自身のキャッシュが無効化されていなければ、(おそらく)最新のデータを読み出すことが出来る．いずれの場合もコアデータと通信すれば自身のデータの最新性を確認できるため、常に最新性の保証されたデータだけの参照を許すことも可能である．
- 複製デポの存在する端末でそのデポ内のデータに対する更新要求を受けた Cmgr は、コアデポを管理する Cmgr と交渉して、自端末上にコアデポを移動させてから更新を行う (図3 (d))．なお、ネットワークの状況や要求パターンによって

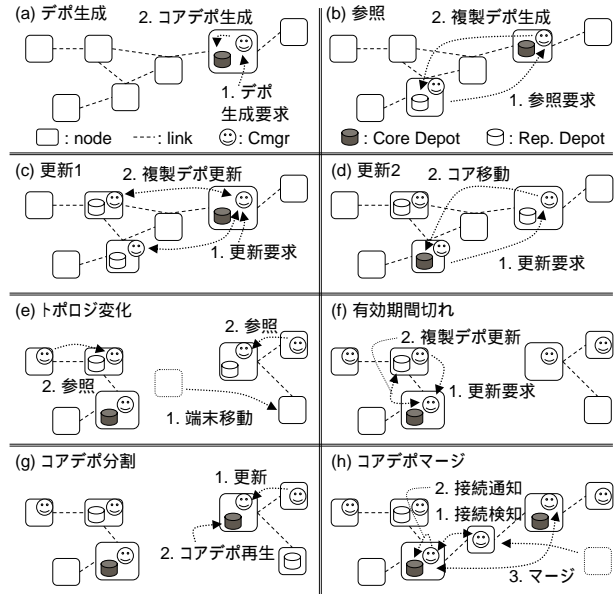


図3 CAOSS の基本動作

は、移動を行わずリモートに更新を行う場合もある．

- ネットワークトポロジが変化してコアデポと複製デポを管理する Cmgr 同士が通信できなくなった場合でも、それぞれのデータを読み出すことは可能である．この時、有効期間内の強整合複製デポであればデータの最新性は保証できる (図3 (e))．
- 通信不可能な複製デポの有効期間が過ぎると、複製デポは無効となる．その後、更新要求が発生しても通信可能な複製デポ間で協調して更新可能となる (図3 (f))．各複製デポは自身の過去の利用履歴などにもとづいて、コアデポに対して有効期間の延長を要求するかどうか決定する．

このように、更新要求の発生した端末にコアデポを移動し、複製デポには有効期間を設けることにより、トポロジが変化し易いユビキタスネットワーク環境で、データの更新と読み出しの可用性の低下を防ぐことができる．また厳密な同期保証と、分散ファイルシステムなど同様のキャッシュ動作が同時に可能であり、複製データ管理の柔軟性が高い．

さらに、CAOSS ではトポロジの変化などの理由によりネットワークが複数のサブネットワークに分割されてしまった場合、コアデポを分割してそれぞれのネットワーク内にコアデポが存在するようにして対処する．これによって分断されたそれぞれのネットワーク内でも整合性を保ったままのデータアクセスが可能である．コアデポの分割と統合は以下のように行われる．

- Cmgr は周囲の端末上の Cmgr と協調して、ネットワークの接続・切断を定期的に調査し、自

端末上のコアデポに通知する。

- 複製デポがデータの更新要求を受けた場合で、更新処理のシーケンスの中でコアデポとのネットワーク切断を発見した場合、複製デポは周囲の端末上の複製デポの中から最も新しいものを選んでコアデポとして再生させる (図3 (g))。
- Cmgr から新しいネットワークの接続を通知されたコアデポは、同じデポ ID を持つコアデポが自分以外に存在しないかどうかを検索する。存在した場合は CVS などの既存のアルゴリズムを用いて両者のデータをマージし、いずれか一つのコアデポを残して複製デポへと降格する (図3 (h))。データの更新衝突によってマージが失敗した場合、ユーザがアクセスしてきたタイミングでユーザに通知し、人手による衝突の回復を促す。

4 CAOSS の効果と応用例

2節で説明した既存のストレージシステムが抱える問題点について、CAOSS による解決方法を以下でそれぞれ説明する。4.4節では CAOSS を使ったアプリケーションサービスの一例として、アドホックネットワークに適した掲示板サービスを挙げる。

4.1 キャッシュデータの同期

CAOSS ではマスタとなるコアデポを移動可能にすることによって、コアデポにアクセス可能な範囲では常に複製デポ間の一貫性制御が可能となる。このためネットワーク内でのデータのバージョンは常に一つとなり、2.1節で述べたようにクライアント端末毎に多数のバージョンが発生してしまう問題は解決される。

また、外部との接続帯域が狭いネットワークや、接続リンクの信頼性が低いネットワーク内においても、コアデポをネットワーク内に移動することによって低コストで複製デポ間の同期を取ることが可能になる。我々は、コアデポをデータ更新端末まで常に移動させることによって、一般的なデータ利用パターンにおいてデータアクセスの性能が向上することをシミュレーションによって確認している [5]。

さらに、ネットワークの分断が起きた場合にはコアデポが自動的に分割されるため、分割されたそれぞれのネットワーク内でも複製データ間の整合性は維持される。結果として同時に存在するデータのバージョン数は最大でも分割されたネットワーク数程度に限られ、最終的なデータの同期時においても更新衝突の回復のコストは小さくてすむ。我々は大まかな評価として、分割されたネットワーク数に対して2倍以上の更新回数があれば、衝突回復のコストは Coda などのファイルシステムに比べて半分以下になると見積もっている [4]。

4.2 複製同期の柔軟性

CAOSS では複製デポの複製レベルを選択することによって、端末が必要とする同期レベルに柔軟に対応することが出来る。例えば、頻繁に最新情報を参照する端末には強整合の複製デポをおくことによって、参照操作の遅延を小さくすることが可能である。また、Cmgr を実行していない複製データを作らない端末でも、別の端末上の Cmgr に接続することによってストレージアクセスが可能である。このため、PDA などのリソースの少ない端末にも適している。

一方、強整合な複製デポは更新操作の際に確実な同期を要求するため、このレベルの複製デポの数が増えると更新操作にかかる遅延が大きくなり、更新が失敗する可能性も高くなってしまふ。このため、端末の要求に適した複製レベルのデポを最適に配置し、平均的な更新遅延などが最小になるようにすることが重要である。このアルゴリズムについては今後検討を進めていく。

また、強整合あるいは弱整合の複製デポが多数存在する場合、データ更新の度に同期のための通信が行われ、ネットワーク帯域を消費することが予想される。しかし CAOSS ではコアデポがデータ更新端末付近に移動すること、必要のない複製データは自動的に有効期間切れ (expire) することによって、同期通信が行われるネットワーク範囲をある程度限定することが可能である。このため消費されるネットワーク帯域の大きさもそれ程問題にならない可能性がある。

4.3 ファイル単位の同期

CAOSS はデポという単位で情報の同期を行う。デポは CVS のモジュールに相当し、ユーザは作業環境に必要な複数のデータをデポの中に登録する。情報の同期はこのデポの単位で行われる。

例えばスケジュールアプリケーションにおいて、ファイル a にユーザ A の予定、ファイル b にユーザ B の予定が書かれていたとして、ある会議に出席する予定の人物がユーザ A からユーザ B に交代したとする。このとき、ファイル a, b は同時に更新されるが、ファイル単位のキャッシュ手法では両方のファイルのキャッシュが同時に同期される保証はない。片方のファイルだけが同期された場合、ファイルを利用するアプリケーションには矛盾したデータが登録されてしまふ。デポ単位による同期ではアプリケーションにとって必要なデータを全てデポに登録しておく限り、このような矛盾は生じない。

4.4 応用例：アドホック掲示板サービス

グループ内の意見交換や議論追跡のため、掲示板や Weblog のような情報共有サービスを提供することは協調作業支援として非常に重要である。

CAOSS で想定するようなネットワーク環境を考えた場合、ネットワークトポロジの変化に伴って接続する掲示板を逐一変更しなければならないのは利便性の大幅な低下を招く。掲示板のデータを CAOSS 上にデポとして参照するようにすれば、そのようなネットワークにおいても常に単一の掲示板を利用することができる。このサービスではネットワークの分断が起きた場合にも、それぞれのネットワーク内の議論の履歴を整合性のとれた形で保存することができるため、協調作業支援としての可用性が高い。また、議論に参加するユーザの活発度によってその端末上の複製デポの複製レベルを変化させるなど、柔軟な運用が可能である。

5 関連研究

5.1 CVS

現在最も普及しているバージョン管理システムである CVS [5] は、インターネットという広域分散環境での協調的なファイルの更新や同期を可能としているため、一種のオンラインストレージサービスとして捉えることができる。ファイルサーバにあたるレポジトリは基本的に単一であるが、CVSup [10] などのツールを使うことによりレポジトリを複数分散配置することも可能である。しかしこの場合でもマスタとなるレポジトリは固定であり、複製レポジトリの生成も手動で行わなければならない。

CAOSS では複製レポジトリに当たる複製デポの自動的な生成を行うことによってストレージとしての利便性を高めている。また、マスタとなるデータの位置を適宜変更することが可能なため、トポロジの変化する動的なネットワーク環境での可用性が高い。

5.2 P2P ファイル共有サービス

最近注目を集めている P2P 型のファイル共有サービス [11] は広域分散環境でのデータ共有を可能としている。これらの技術ではサーバのファイル転送負荷を多数のクライアントに分散させることによって非常に規模の大きいファイル共有サービスを成立させた。しかし多くのサービスでは Read Only なファイルの配布を行っているのみであり、分散配布されたデータの更新を行う操作は基本的に考慮されていない。また複製されたファイルの管理はユーザに委ねられるため、データの永続性を保証することもできない。

CAOSS ではデポ ID から実際にデポを検索する際に利用する外部の名前解決サービスとして、Chord [8] など P2P 系の名前解決サービスを想定している。CAOSS は P2P 系広域データ分散サービスに対して、可用性の高いデータ更新方法を提供するシステムとしてとらえることも可能である。

6 おわりに

本稿では、将来のユビキタスネットワークにおいて、特定の端末に依存しないサービス提供を可能とするために必要な適応型オンラインストレージシステム CAOSS を提案し、既存のストレージシステムが抱える課題と対比して有効性を説明した。

CAOSS の定量的な有効性については、現在強整合の複製デポのみを含む系についてシミュレーション環境を構築し、評価を行っている。今後、全ての種類の複製デポを含んだ評価を行い、有効性の総合的な評価を行っていく。またプロトタイプシステムを構築して、無線マルチホップネットワークなど実環境での性能評価を行い、更に有効性を示していく予定である。

参考文献

- [1] <http://my.yahoo.co.jp/>
- [2] "Network Mobility Support Terminology", Internet Draft, IETF, July 2002.
- [3] 中村 元紀, 井上 知洋, 久保田 稔, "アドホックネットワーク環境のためのデータ管理方式", 信学総大 B-15-6, 2002年3月.
- [4] 井上知洋, 中村元紀, 久保田稔, "適応型オンラインストレージにおける切断時データ同期方式", FIT2002 M-88, 2002年9月.
- [5] 中村 元紀, 井上 知洋, 久保田 稔, "環境適応型オンラインストレージにおける複製管理方式の評価", 信学会 MoMuC 研究会論文集, 2002年11月 (掲載予定).
- [6] <http://www.cvshome.org/>
- [7] <http://www.coda.cs.cmu.edu/>
- [8] Ion Stoica, et al., "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", *Proceedings of ACM SIGCOMM 2001*.
- [9] Sylvia Ratnasamy, et al., "A Scalable Content-Addressable Network", *Proceedings of ACM SIGCOMM 2001*.
- [10] <http://www.polstra.com/projects/freeware/CVSup/>
- [11] Ian Clarke, et al., "Protecting Free Expression Online with Freenet," *IEEE Internet Computing* 6(1), pp. 40-49 (2002).