広域分散ファイルシステムMUFSとその適用

鈴木重治*　小泉一郎**　内田昭宏*

*NECオープンシステム基盤開発研究所
**NEC技術情報システム開発

分散ファイルシステムを，大規模・広域分散環境にも適用することが望まれるようになったが，現在広く使われているNFSを，そのまま大規模・広域分散環境に適用することにはいくつかの問題がある．我々は，既存NFSにディスク上キャッシュ機能，耐障害性能，異種プロトコル接続機能を追加する，広域分散ファイルシステムMUFSを開発した．本論文では，MUFSの実現方式と，その効果について述べる．

1　はじめに

近年，ダウンサイジング化とネットワーク利用の普及によって，従来，比較的小規模のLANが主体だった分散環境が，複数のLANを広域ネットワーク(WAN)で接続した大規模・広域分散環境へと発展してきた．

そして，このような分散環境の拡大に伴い，主にLAN環境で使われていた分散ファイルシステムを大規模・広域分散環境に適用することが求められるようになった．

現在，LAN環境での分散ファイルシステムとしては，NFSが広く用いられているが，NFSをそのまま大規模・広域環境に適用するにはいくつかの問題がある．広域環境専用に新たなファイルシステムを開発することも考えられるが，この方法では，既存のNFS資源をそのまま利用することはできない．

我々が開発したMUFS(interMediate Universal File System)は，NFSファイルサーバやNFSクライアントなどの資源はそのままに，ディスク上キャッシュを利用した効率的な広域ファイル共有の実現や，異種プロトコルアクセスなど，NFSの機能を拡張するパッケージである．

本論文では，MUFSの実現方法と，適用結果について述べる．

"Distributed File System MUFS and It's Application"
by Shigeharu SUSUKI*,Ichiro KOIZUMI** and Akihiro UCHIDA*
* NEC Open Systems Development Laboratories.
** NEC Scientific Information System Development Ltd.

2　分散ファイルシステムの現状

LAN環境では，分散ファイルシステムの業界標準として，Sun Microsystems社のNFSが広く用いられている．しかし，NFSは比較的小規模のLAN環境で最大の性能を発揮するように設計されているため，大規模・広域分散環境で利用するには，次に挙げる問題がある．

- ステートレスプロトコル

  ステートレスプロトコルは，サーバの負荷を低減し，サーバ障害からの回復も容易とする効果がある．しかし，ステートレスであるために，厳密なキャッシュコンシステンシを維持することができない．

- RPCのセマンティックス

  NFSに使われているRPC(ONC/RPC)は，タイムアウトによる再送のみをサポートしている．また，NFSのRPCの中には，同じリクエストを何回繰り返し実行しても同じ結果を返さない(create, removeなどの)リクエストが存在するため，ネットワークの遅延が大きくかつ不安定なWANや，サーバが高負荷の状態では，しばしばエラーを生じる．

大規模・広域分散環境でも，分散ファイルシステムを利用可能とするため，次ぎに挙げるように，いくつかのアプローチが試みられている．

- 新たなファイルシステム，ファイル共有プロトコルを作る

  CMUの開発したAFS (Andrew File System)[1]が代表的なものである．AFSはロー

カルディスク上キャッシュの導入により，広域ファイル共有でのパフォーマンスを向上させた．同時に，サーバの複製機能の実現，セキュリティの強化も行なわれている．

AFS に続いて CMU で開発された Coda [2][3] は，AFS の広域ファイル共有に加えて，ネットワーク切断時にもファイルアクセスの継続を可能とするディスコネクティッド・オペレーションを導入した．

- 既存の NFS 資源を生かして機能強化を行う

  Sun Microsystems 社自身によって，RPC のタイムアウト時間をネットワーク / サーバの特性に応じて，自動的に調整する機能が導入された．NFS プロトコルを拡張する提案も行われている．

  InterStream 社の eNFS/Cache は，NFS クライアントにディスク上キャッシュの機能を導入して，大規模 LAN 環境での NFS クライアントの性能向上を実現している．

  また，NFS プロトコルを介して，FTP などNFS 以外のファイルアクセスプロトコルの利用を可能にする，WWFS (World Wide File System) [4], ALEX [5] などもある．

3 MUFS

我々が開発した MUFS は，NFS ファイルサーバや NFS クライアントなどの資源はそのままに，ディスク上キャッシュを利用した効率的な広域ファイル共有の実現や，異種プロトコルアクセスなど，NFS の機能を拡張するパッケージである．

MUFS は，NFS クライアントと NFS サーバの間に位置し，NFS クライアントから NFS プロトコルを受け取り，ディスク上キャッシュを通して，マスタファイルサーバへのアクセスを行なう (図 1参照)．

MUFS の主な機能を以下に挙げる．

- ディスク上キャッシュ

  大容量のキャッシュをディスク上に持つことで，遠隔地との効率的なファイル共有を可能にする．サーバ負荷の大きい大規模 LAN 環境においても，クライアントのパフォーマンスを向上させる効果がある．

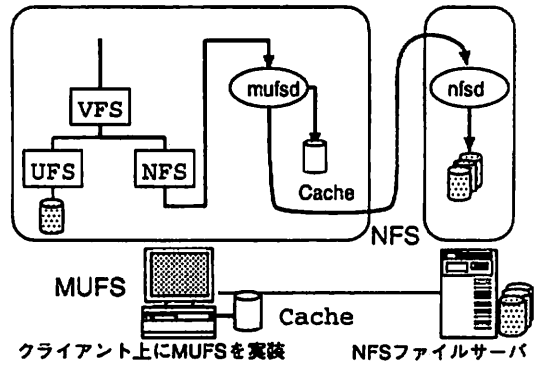

図 1: MUFS

  また，ネットワークやファイルサーバに対しての負荷を低減させることで，システム全体の性能を向上させる効果もある．

- 耐障害性能

  ネットワークやサーバ障害時に動作継続を可能にする，ディスコネクティッド・オペレーションを実現している．

- 異機種 / 異種プロトコル接続

  FTP サーバやメインフレーム (ACOS4/6) [CoupleFS[1]] [7] [8] 上のファイルを，UNIX ローカルファイル空間でアクセス可能とする．

  さらに，FTP 接続機能を利用して，任意の anonymous FTP サイトをローカルファイル空間にマッピングしてアクセス可能にする機能も実現している．

- 既存の NFS 資源との互換性

  既存の NFS ファイルサーバを，そのまま，広域ファイル共有のためのファイルサーバとして利用することができる．

MUFS は，NFS プロトコルを受け取るユーザプロセスとして実現されている．これは，実現の容易性と，移植性を考慮したためであるが，ユーザプロセスとして実現されているため，パフォーマンス的に限界がある．また，NFS クライアントとの間の通信もステートレスプロトコルであるため，クライアント側でオープン / クローズを認識できないことなどのデメリットもある．

[1] CoupleFS は，MUFS に ACOS の名前空間と接続する機能を拡張して，ACOS とのファイル共有を実現している．

3.1 ローカルディスク上キャッシュ

MUFS は，ローカルディスク上にキャッシュを持つ．キャッシュの対象は，データ，アトリビュート，ディレクトリである．MUFSは，ファイル全体をキャッシュの単位としている．これは以下の理由による．

- ブロック単位のキャッシュを導入した場合，ディスコネクティッド・オペレーション時の，ファイルアクセス制御が複雑になりすぎる．
- 過去のファイルアクセスパターンの研究 [9] [10] によりファイルの一部のリード要求に対して，ファイル全体をキャッシュしてしまっても，キャッシュの利用効率が極端に低下することはないことが予想された．
  - ファイルアクセスのうち 70% は，ファイル全体をアクセスする．
  - ファイルアクセスのうち 90% 以上は，シーケンシャルにアクセスする．
  - リードアクセスのうち，78% は，ファイル全体をシーケンシャルにアクセスする．

キャッシュ容量の上限は，共有するファイルシステムごとに設定することができる．キャッシュ容量が上限値を越えた場合には，デーモンプロセスによって自動的にキャッシュを消去する．キャッシュの消去は，基本的に最も使われていないものから順に (LRU = Least Recently Used) 消去を行う．

しかし，不意のディスコネクティッド・オペレーションに備え，作業継続に最低限必要なファイルは，最近アクセスされていなくともキャッシュ上に残す必要がある．MUFS は，キャッシュ消去に対して優先度を付ける機能によって，これに備えている．

3.2 キャッシュコンシステンシ

MUFS は，NFS プロトコルを使っているため，本質的にキャッシュとサーバの間で完全なコンシステンシを提供することができない．(通常の NFS インプリメンテーションでも，パフォーマンス向上のためにメモリ上のキャッシュを持つ．このため，数秒から数十秒の間メモリ上のキャッシュと，ファイルサーバ上のデータが一致しない危険性を持っている)

キャッシュされたファイルアトリビュート，データに対してタイムアウト時間を設けることで，キャッシュのコンシステンシを維持する方法を採用している．

標準的な NFS のクライアントと同等の数秒から数十秒のタイムアウト時間で利用することも，数時間から一日というように長いタイムアウト時間を設定して，ルーズなコンシステンシで利用することも可能である．

ファイル更新の衝突が起こった場合，NFS では後からサーバを更新した方が優先されるが，MUFS はサーバのデータを更新する前に，衝突の検査を行う．タイムアウトを長くとると，更新が衝突する危険性は増大するが，ユーザが意識しないで，不容易にサーバのデータが更新されることはない．

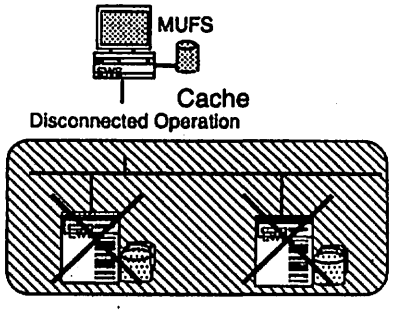

図 2: ディスコネクティッド・オペレーション

3.3 耐障害性

MUFS は，予め複数のサーバを登録しておいて，利用可能なサーバを動的に選択する，サーバ切替機能を実現している．

同様のサーバ切替機能を実現する，automounter[2]，や amd では，既にオープンされているファイルに対して，動的にサーバを切り替えることはできないが，MUFS ではオープン中にサーバが切り替わっても，ユーザは意識することなくファイルアクセスの継続が可能である．

ただし，MUFS 自身はサーバの複製を生成する機能を持たないので，複製間の整合性を保つ作業はユーザに任されている．

[2]Sun OS の機能

更に，ファイルサーバの障害や，ネットワークの分断などによって，アクセス可能なファイルサーバが存在しない場合にも，ディスク上にキャッシュされたデータをアクセス可能とする，ディスコネクティッド・オペレーションの機能を実現している (図2参照).

ディスコネクティッド状態で動作している場合は，キャッシュされていないファイルへのアクセスはエラーとなる．ディスコネクティッド状態でもキャッシュ上のファイルの更新が可能である．更新されたデータは，サーバ接続が回復した際にまとめてファイルサーバにライトバックされる．

4 MUFS の適用

LAN 環境では，多数のワークステーションが，一台のファイルサーバのファイルを共有するシステム構成が，広く用いられている．しかし，接続するワークステーションの数が増大した大規模 LAN 環境では，サーバへの負荷集中や，ネットワークの負荷増大によって，ファイルアクセスのレスポンスが低下するという，広域環境と同様の問題が生じるようになってきた．

このような大規模 LAN 環境では，広域ファイル共有と同様のディスク上キャッシュをクライアント側に導入することによって，ファイルアクセスのスループットを向上させることが可能である．さらには，ファイルサーバ切替機能とディスコネクティッド・オペレーション機能によって，耐障害性を高めることが可能である．そこで，MUFS を大規模 LAN 環境に適用し，ネットワークやサーバの負荷低減の効果を検証する．また，低速の回線を介したファイル共有についても検証する．

4.1 コマンドファイルシステムの共有

コマンドやコマンドに関係したデータベース，ライブラリ (以後コマンドファイルシステムと呼ぶ) を利用して検証を行なう．コマンドファイルシステムは，典型的な LAN 環境においてワークステーションが共通に利用するファイルであり，アクセスできなくなると，作業の継続に支障をきたす，重要なファイルである．

共有対象のコマンドファイルシステムを，ファイルサーバ上で静的に観測した結果を以下に示す．

- コマンドファイルシステムのファイル数は 1 万 9 千個，サイズは 485MB
- 更新は稀れで，ほとんどのファイルは 1ヵ月以上更新されていない (図3参照).
- read アクセスは一部のファイルに集中している．一週間以内にアクセスされたのは，全体の 13% 程度．一ヵ月以内にアクセスされたのは，全体の 38% 程度 (図4参照).

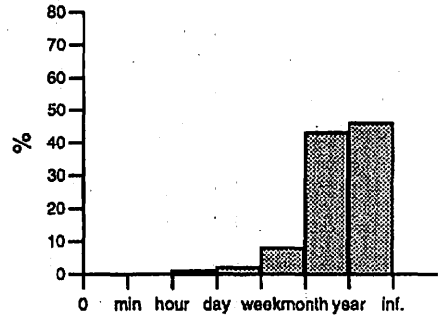

図 3: 最終更新時間

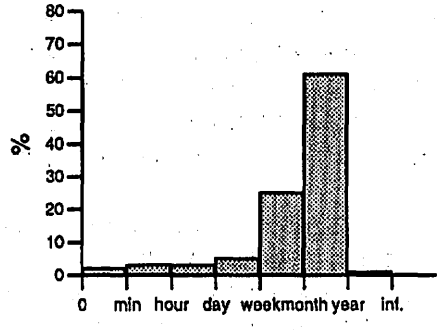

図 4: 最終アクセス時間

一週間以内にアクセスされたファイルは，全体の 13% (= 63MB) である．しかし，これは複数のワークステーションによって共有されているファイルサーバ上での測定値のため，個々のワークステーション上では，これよりかなり少ないキャッシュ容量で，日常の作業で利用するコマンドのワークセットを納めることが可能と予測される．

このコマンドファイルシステムを NFS を利用して共有した場合の，NFS リクエストの使用分布

を表1に示す．write，create，setattr 等のファイル内容を変更するリクエストは少なく，lookup，getattr，read 等のファイル内容を読み出すリクエストがほとんどである．

表 1: NFS リクエストの分布

NFS リクエスト	%
getattr	48.9
lookup	23.0
readlink	15.7
read	4.9
setattr	2.3
write	1.2
create	1.2
remove	1.2
readdir	1.6
statfs	0.0
null	0.0
root	0.0
writecache	0.0
rename	0.0
link	0.0
symlink	0.0
mkdir	0.0
rmdir	0.0

4.2 効果

実際に我々自身の作業環境でのコマンドファイルシステム共有に MUFS を適用し，キャッシュが空の状態から約 260 時間後までのデータを採取した．図 5に，MUFS の運用開始直後のキャッシュ増加を，図 6には同じキャッシュの長時間にわたる増加傾向を示す．また，それぞれのグラフには同時系列でキャッシュヒット率も示す．この測定を行っている間，キャッシュの消去は行っていない．

運用開始直後は，X-window や Nemacs などの作業環境立ち上げのため，短時間に大量のファイルがアクセスされるが，キャッシュが空の状態なので，全てのアクセスはキャッシュミスヒット[3]となり，キャッシュ容量も急速に増加する．キャッシュヒット率が 100% になる場合があるが，これは夜間や休日でマシンを利用していないか，ディスコネクティッド・オペレーション運用である．

キャッシュサイズは，運用開始後 5 時間で約 16MB に達し，一週間の作業を行った後には，約 33MB となった．これから，運用開始後時間がたてば，キャッシュ増加量が低くなり，ファイルアクセスの多くをキャッシュで解決し，サーバへアクセスしていないことが分かる．また，485MB のコマンドファイルシステムを共有していても，一週間で 33MB しか利用しておらず，コマンドファイルシステム共有ならキャッシュ容量が数十 MB で十分である事が分かる．

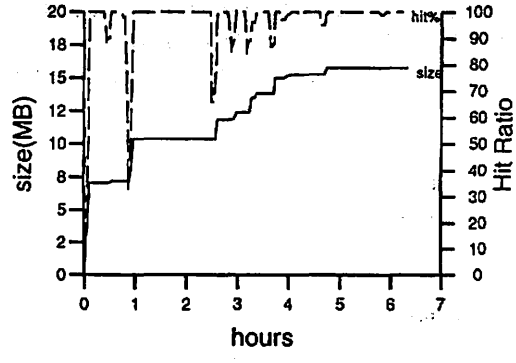

図 5: キャッシュ増加 (短期間)

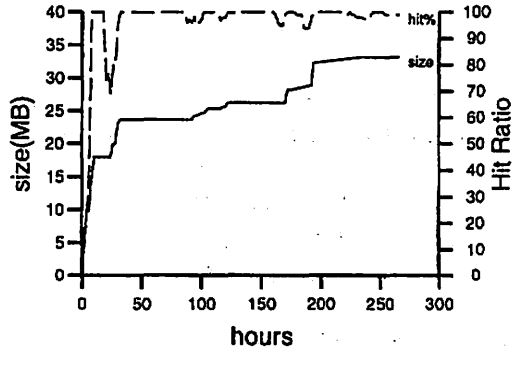

図 6: キャッシュ増加 (長期間)

[3] キャッシュヒット率は，ある観測点から 6 時間経過した時のキャッシュの増加分を，その期間に行なわれた NFS read リクエストの総バイト数で割った値を 1 から引いた値である．

4.3 低速回線への適用

典型的な広域ネットワークは，LANと比べて速度が遅く，容量の低い回線を用いている．そこで，大規模LAN環境の場合と同様，コマンドファイルシステムを低速回線を介して共有した場合を考え，MUFSの効果を検証する．

図6のグラフから，急激にキャッシュ容量が増加する最初の部分を除くと，キャッシュミスヒットによって生じるファイル転送は，1時間あたり約35KB程度である．この程度の通信量であれば，低速回線を介したファイル共有も十分実用的なものとなる．

稀に，MByteオーダのファイルをアクセスしたときに，10秒単位の待ちが生じるが，図7のデータから全ファイルの50%は2.5KB以下である．64Kbps程度の通信速度が得られれば，2.5KBのデータは0.5秒程度で転送出来るため，キャッシュミスヒットが生じても，転送待ち時間が意識されることはない．

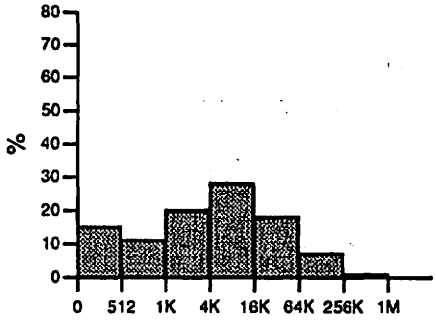

図7: ファイルサイズの分布

現在のインプリメンテーションでは，ファイル全体が転送されるまで待ってから，ファイルアクセスの要求に応えているが，キャッシュをファイル単位とすることとは別に，部分的にファイル転送が終わった段階で，ファイルアクセス要求に応えるような機能拡張が必要である．

4.4 低速回線での性能

サーバとの通信速度を制限した環境で，MUFSを利用した場合の評価を行った．

ここでは，合計10MBのファイル群の転送(cp)と，MUFSのソースファイル(合計360KB，約16,000ライン)のmake時間，そして，Nemacsの起動時間それぞれについて，全てキャッシュミスヒットする場合と，全てキャッシュヒットする場合の2通りについて測定することで低速回線でのMUFSの性能を評価する．

測定結果を表2に示す(参考にLAN環境でNFSを利用したファイル共有の場合のデータも付ける)．

表2: 64Kbps回線シミュレーション

処理内容	LAN NFS	低速回線 cacheあり	低速回線 cacheなし
10MB file 転送	42s	40s	300s
make	164s	300s	355s
Nemacs	7s	7s	23s

測定結果より，MUFSのようなディスク上キャッシュ機能を使えば，64Kbps程度の回線でも現実的な性能でコマンド共有が行なえることが確認できた．

稀れに大きなファイルに対して，キャッシュのミスヒットが生じると，ファイルアクセスが完了するまでに幾らか待たされるという問題は残るが，ある程度の長期間MUFSを運用することによって，キャッシュのヒット率が上がり，また，利用者が広域回線経由のファイル共有であることを認識していれば，十分に実用に耐えるものであることを確認した．

4.5 耐障害性

図6に示すように，キャッシュ容量が急激に増加するは最初の数日だけであり，これ以降，キャッシュは僅かな増加しかしない．

これ以降も，キャッシュは一定量の増加を続け，キャッシュヒット率も100%に近づくことはないが，日常の作業に必要なファイルは，この段階で，ほとんどキャッシュされている．

図6に示したキャッシュ増加のグラフ中でも，50時間経過した前後においてネットワーク障害によってディスコネクティッド状態となり，一時的にキャッシュの増加が止まっている部分がある．

まだキャッシュ量が伸び続けている段階ではあるが，障害が短期間であれば，作業に支障がでることはない．

5 その他の運用例

- FTP サーバの複製

  距離の離れた二つの地域に対して，同一内容の anonymous FTP サービスを行うために MUFS を利用している．

  以前は，片方のサーバにマスタ側のサブセットを作り，夜間にマスタの FTP サーバからサブ FTP サーバに更新分を転送していた．

  MUFS を導入することによって，各地区間のネットワーク負荷と，マスタ FTP サーバの負荷を低減できた．また，各地区の FTP サーバの占有ディスク容量も削減することができた．

- anonymous FTP ゲートウェイ

  社外インタネットとのゲートウェイマシンで，任意の anonymous FTP サイトを ".../mufs/aftp.site.name/...." という名前でアクセス可能にするゲートウェイ機能の運用を行っている．

  anonymous FTP サイトを UNIX ファイル空間でアクセスできるというメリットは大きいが，anonymous FTP の空間の広さ (1000 以上のサイト，TByte オーダのトータルサイズ) のため，100KB 程度のキャッシュサイズだとキャッシュヒットによる性能向上が期待できないという問題もある．

6 おわりに

本論文では，広域分散ファイルシステム MUFS を，LAN 環境のコマンドファイルシステムに適用し，ネットワーク・サーバの負荷低減と耐障害性に効果を発揮することを確認した．また，MUFS が WAN 環境でも効果があることを，WAN シミュレーションより確認できた．

我々は，広域分散開発環境を構築する基盤技術という位置付けで MUFS を開発している．MUFS の耐障害性の機能であるディスコネクティッドオペレーションは，モバイルコンピューティング (mobile computing) を実現する基盤技術として用いることができる．

MUFS の今後の課題として，以下のものがある．

- 広域認証の実現

  MUFS は，ユーザの認証として，UNIX ファイルシステムにおけるユーザ ID，グループ ID を用いている．広域分散環境では，ユーザ ID だけで認証することはセキュアではない．Kerberos[11] のような認証システムを組み込む必要がある．

- コンシステンシ

  現バージョンでも，ファイル更新の衝突は検出されているが，更新衝突時の調停 (Reconciliation) 方法には検討の余地がある．

  より厳密なセマンティックによるコンシステンシチェックが必要とされる場合には，ファイル共有プロトコル自体を変更する必要がある．

- サーバの複製生成の実現

  MUFS は，サーバの切替を実現しているが，サーバの複製を生成していない．耐障害性能を一層強化するためには，サーバの複製を生成する機能を実現する必要がある．

今後さらに，MUFS を WAN 環境へ適用し，評価し，広域分散環境対応へと強化していく予定である．

参考文献

[1] Alfred Z. Spector 他, "Wide Area File Service and the AFS Experimental System (Uniting File Systems)", Unix Review, Vol 7, No 3, March 1989

[2] M. Satyanarayanan 他, "Coda: A Highly Available File System for a Distributed Workstation Environment", CMU-CS-89-165, 1989

[3] James J. Kistler 他, "Disconnecte Operation in the Coda File System", ACM Symposium on Operating System Principle, Oct 1991

[4] 山口英 他, "IP Meeting '92", Nov 1992, JEPG/IP, pp 62-64

[5] Vincent Cate, "Alex - a Global File System", USENIXWorkshop on File System Program, May 92

[6] 小泉一郎 他, "ディスク上キャッシュを持つ分散ファイルシステムのLAN環境への適用", 情報処理学会第43回全国大会, 1991/10

[7] 伊波通啓 他, "異機種分散ファイルシステム - CoupeFSの設計 -", 情報処理学会第45回全国大会, 1992/10

[8] 仁科雅子 他, "CoupleFSの名前管理機構", 情報処理学会第45回全国大会, 1992/10

[9] Ousterhout,J.K. 他, "A Trace-Driven Analysis of the Unix 4.2 BSD File System", Proc 10th ACM Symbosium on Operating System Principles, Orcas Island, Dec, 1985

[10] Mary G.Baker 他, "Measurements of a Distributed File System", PROC of the Thirteenth SOSP, Pacific Grove, CA, 14-16 October 1991,pp 198-212

[11] Jennifer G. Steiner 他, "Kerberos: An Authentication Server for Open Network Systems", In Proceedings of the USENIX Winter Conference, 1988, pp 191-202