複数OS利用のためのユーザアカウントデータベース統合

倉前宏行† 島野顕継† 木村彰徳†
松本政秀† 古野良樹†† 亀島鉱二†

OSが混在した教育用PCクラスタシステムにおいて，ユーザアカウントデータベースを統合するため，ディレクトリサービスを導入した．このシステムの性能を評価するため，ユーザ情報への参照経路に基づいた性能測定方法を開発した．評価の結果，ユーザアカウントデータベースの統合によって性能が低下することが明らかとなった．特にUNIX環境におけるネームサービス性能は実用に耐えがたいことから，NISを用いたバイパスシステムを開発し問題を解決した．

Integration of User Account Database on Multiple OS Environment

Hiroyuki Kuramae,† Akitsugu Shimano,† Akinori Kimura,†
Masahide Matsumoto,† Yoshiki Furuno†† and Kohji Kamejima†

In a PC cluster system for education, directory service has been introduced for integration of user account database on multiple OS platforms. In order to evaluate the system, a performance measurement method is developed based on lookup user information paths. As a result of the evaluation, the performance is deteriorated by the integration of the user account database. Since name service performance of the UNIX environment is hard to resist practical use, a bypass system using NIS is developed and solved for the performance problem.

1. 緒　言

工学教育では，一般的なコンピュータリテラシとともに，コンピュータ上に実現象を表現する技術を習得させることが重要である．このためには，情報処理教育と科学計算教育を有機的に結びつける必要がある．実際，対象のモデル化，プログラミング，データ解析から，文書作成，プレゼンテーションまでをカバーする科学計算-情報処理プロセスを一貫して実行するスキルを見に付けておくことは，研究者・技術者としての知的生産性を高める上で不可欠である．著者らは，このような視点から，Windows NTとPC-UNIXのデュアルブート環境を提供する教育用PCクラスタシステムを構築してきた[1)~3)]．このシステムは，コンピュータリテラシ教育からプログラミング，数値実験などの講義・演習，さらには研究利用や授業時間外のオープン利用まで，さまざまな授業演習等で利用する．よって，ユーザはコンピュータに初めて触れる者から，研究のためのシステムソフトウェア開発を行う者まで多様であり，そのスキルや利用形態も多岐にわたる．教育用システムには，このようなユーザにフレキシブルに対処するような管理・運用が求められる．

一方でインターネットへの接続が普及するにつれ，セキュリティ保全の要求も発生してきた．こうした教育用システムにおいては，ユーザを特定・認証するためのユーザアカウント(ユーザ名とパスワード)の管理が重要となる．本学をはじめ多くの大学では，入学時にアカウントを発行することから，これを用いて学内の全てのネットワークおよびコンピュータシステムをシームレスに利用できることが，利便性上，望ましい．

情報センターの運用におけるセキュリティの確保やシステム管理の省力化を目的とした研究成果[4)~9)]が報告されている．しかし，WindowsとUNIXのユーザアカウントを統合するとともに，フレキシビリティとセキュリティを両立させる運用方式は容易ではなく，このための技術も確立されていない．

本研究では，ユーザアカウントを完全に統合するとともに十分なセキュリティレベルを保つため，ディレクトリサービスNDS (Novell Directory Services)を導入した[2),3)]．しかしNDSを導入したシステムは，実用には耐えがたい性能であり，商用ソフトウェアソリューションであるため，その原因を究明することは困難である．そこで，ディレクトリサービスによってアカウント統合されたシステムで必要とされる性能を明らかにするため，ユーザアカウント情報のデータフローに基づいたシステムの性能測定方法を開発し，実際に測定を行なった．本稿では，ディレクトリサービスを用いたアカウント統合システムの性能評価について述べる．

† 大阪工業大学
Osaka Institute of Technology
kuramae@dim.oit.ac.jp
†† シンフォニーインターナショナル
Symphony Intl., LTD

2. OS 混在環境におけるユーザアカウント管理の問題点

Windows と UNIX のユーザアカウントを統合化もしくは一元化する際，実現しなければならない主な機能として

(1) Windows へログオンする際の SAM (Security Account Manager) 認証

(2) UNIX のログイン認証，および ftp, telnet, rsh, rlogin, POP などさまざまなアプリケーションにおけるユーザ認証

(3) UNIX におけるユーザ情報提供のためのネームサービス

が挙げられる．このうち，(1) は，通常，ドメインコントローラを用いてユーザ情報を管理するが，図 1 に示すように，Samba[10] や Solaris PC Netlink[11] といったソフトウェアを導入することにより，UNIX サーバ上に構築することもできる．(2), (3) については，通常，NIS (Network Information Service) が用いられる．

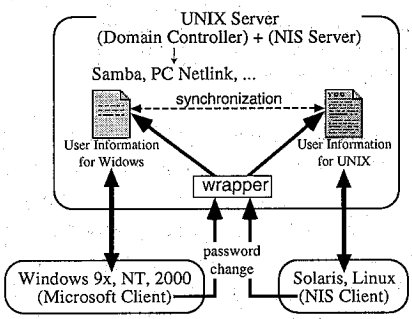


図 1　パスワード情報の同期によるアカウント一元化

新規ユーザの登録時に Windows および UNIX のパスワード情報を一致させておき，ユーザ自身によるパスワード変更がそれぞれのデータベースに反映されると同期をとることができる．よって，パスワード変更を Wrapper を経由させることにより，2 つのユーザアカウントデータベースの一元化が実現される．しかしながら，パスワード同期のための Wrapper プログラムを悪意を持って攻撃されることや，NIS のセキュリティの甘さ，2 つのデータベースの同期のタイミングなど，全学規模の大規模・大量ユーザの一元管理には問題がある．

別の方法として，Microsoft Windows Services for UNIX[12] を用いると，Windows ドメインコントローラを NIS サーバとして動作させることで，UNIX アカウントを Windows へ統合することができる．しかし，この方法では，パスワードとして使用できる文字種や文字数に制限があるほか，安定的なシステム運用の面で問題がある．

3. NDS の導入によるユーザアカウント統合

3.1 システム構成

2 つの OS のユーザ情報を完全に統合するため，ディレクトリサービスを利用する．ディレクトリサービスは LDAP (Light-weight Directory Access Protocol) などオープンな規格に基づいて実装され，OS 環境によらないユーザ情報管理への期待が高まっている．そこで，全学の学生ユーザアカウントを完全に統合し，ユーザアカウントデータベースの階層管理を行なうため，図 2 に示すように，大宮 (大阪市旭区) および枚方 (大阪府枚方市) の 2 つのキャンパスの情報センターを中心に NDS Corporate Edition 8.1-1 を導入した．

クライアントは RedHat 6.2J (Linux 2.2.16) と Windows NT もしくは 2000 のデュアルブート環境を構築し，UNIX サーバは全て Solaris 7 とした．これら全てのサーバとクライアントを各キャンパスおよび学科のサブシステムの階層に合わせて，図 2 (b) のようにパーティション分割した．この階層構造に合わせて，各キャンパスの情報センターに NDS マスターサーバを置き，各パーティションごとにドメインコントローラを置いた．各演習室および学科サブシステムには，それぞれ NDS レプリカサーバおよび BDC (Back-up Domain Controller) を設置した．ユーザアカウントは，教員・学生の所属学部・学科ごとに階層管理した．

3.2 Windows アカウントの NDS 統合

Windows 環境においては，Windows 9x/NT/2000 のクライアントは，ドメインコントローラを NDS レプリカサーバとして設定するのみで，ユーザアカウントを NDS へ統合できる．図 3 に示すように，Windows クライアントからの NT ドメインログオン要求はドメインコントローラ上で NDS へ自動的にリダイレクトされ，NDS による SAM 認証を受ける．

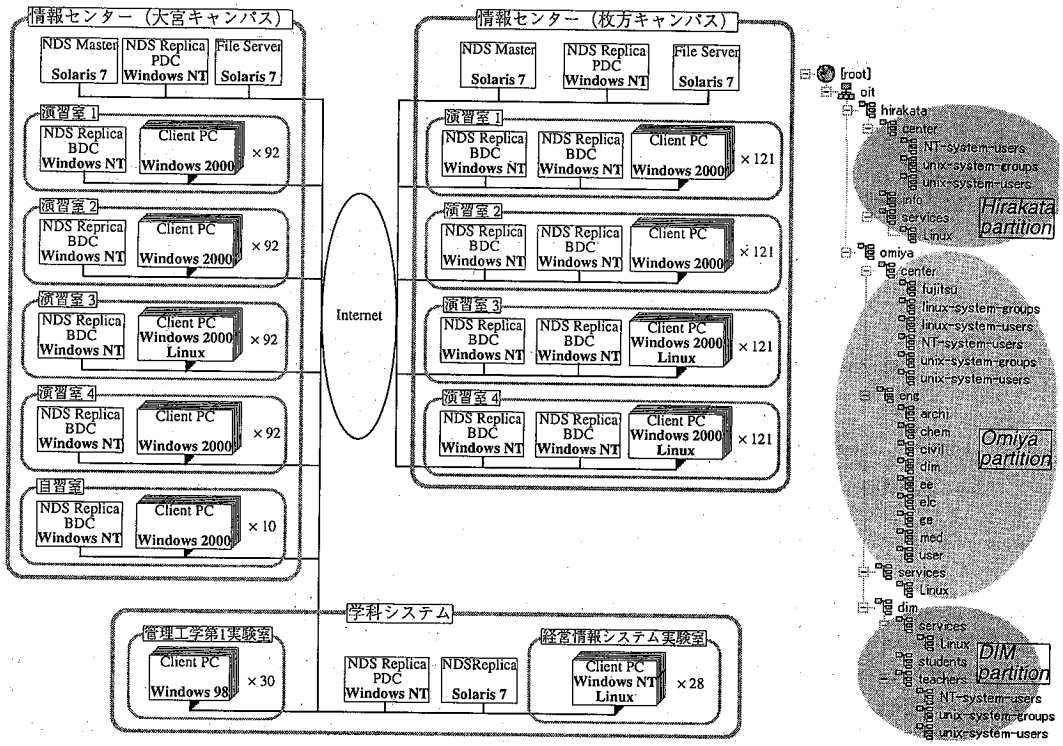
3.3 UNIX アカウントの NDS 統合

Linux や Solaris など UNIX 環境には，NDS の PAM (Pluggable Authentication Modules) 認証モジュール (`pam_nds.so.0`) が提供され，ログインをはじめ ftp, telnet, rlogin などのアプリケーションにおけるユーザ認証を NDS へ統合できる．この設定は，PAM 設定ファイルにおいて，図 4 のように NDS PAM モジュールの指定を行なう．

```
auth       sufficient   /lib/security/pam_nds.so.0
account    sufficient   /lib/security/pam_nds.so.0
password   sufficient   /lib/security/pam_nds.so.0
session    sufficient   /lib/security/pam_nds.so.0
```

図 4　NDS 認証のための PAM 設定ファイルの記述

NDS に登録されているユーザ情報は，NSS (Name Service Switch) フレームを通じて `/etc/passwd` ファイ



(a) システム構成　　(b) パーティション分割

図2　NDS の全学的導入

ルの内容に相当する情報を配布することができる．設定は，/etc/nsswitch.conf ファイル中に図5のように記述することにより，NDS の NSS モジュールを指定する．ただし，NDS パスワードは RSA 方式により暗号化されていることから，getpwuid() や getpwnam() などのシステムコールによってもアプリケーションには開示できないため，これらによって得られるパスワードフィールドは，* となる．

```
passwd:     files nds
group:      files nds
```

図5　/etc/nsswitch.conf ファイルの設定

4. システムの応答性能測定

4.1 測定の目的

本システムにおいては，合計 900 台ものサーバ/クライアント機に NDS を導入し，のべ 10,000 名分のユーザアカウントを管理する．このような大規模システムを運用する上で，実用に耐えうる性能が得られているかどうかを調査するため，上述したユーザアカウントデータベースへの情報参照経路に基づいて性能測定を行なった．

測定は，図2 (a) 中の学科サブシステムである「経営情報システム実験室」の環境で行なった．クライアントは富士通 FMV-6450DX3 (Pentium III 450MHz CPU, 96MB RAM, Windows NT Workstation 4 SP 3/RedHat 6.2J OSs)，UNIX サーバは富士通 GP400S5 (UltraSPARC IIi 360MHz CPU, 128MB RAM, Solaris 7 OS)，NT サーバは富士通 FMV-6550TX3 (Pentium III 550MHz CPU, 256MB RAM, Windows NT Server 4 SP 6 OS) である．これらは，Fast Ethernet スイッチ (100 Mbps) によって接続されている．本実験室サブシステムは，DIM パーティションとして独立しているが，Omiya パーティション (大宮キャンパス情報センター) の読みだしレプリカとして設定しており，8,000 名分のユーザ情報を参照することができる．

なお，NDS の性能測定結果を比較検討するため，同じハードウェア構成で NIS マスターサーバおよび NT サーバ (PDC) を用意し，NDS を利用せずにユーザアカウントを NIS と PDC とで別々に管理した場合においても同様に性能を測定した．

4.2 Windows におけるユーザ情報参照性能

4.2.1 測定方法

Windows NT には，ユーザ情報を参照するための Win32API ライブラリ関数として，ユーザ名から SID

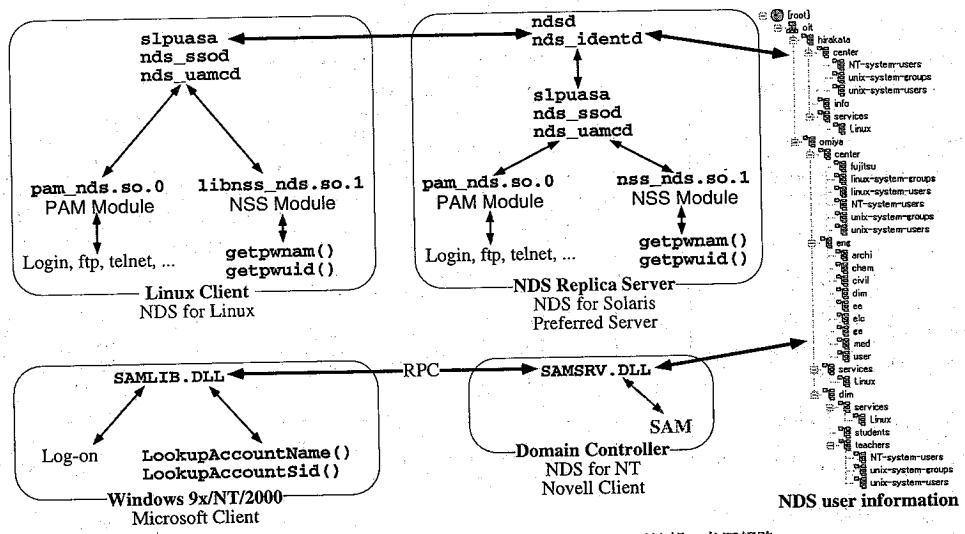


図3　ユーザアカウントの統合システムにおけるユーザ情報の参照経路

(security identifier) を取り出す LookupAccountName() およびSIDからユーザ名などを取り出す LookupAccountSid() が用意されている．そこで，これらの関数の応答性能について測定した．

このプログラムは大きく分けて3つの処理からなる．

0. 測定用ユーザ名入力　検索対象とするユーザ名をあらかじめ用意しておいたファイルから読み込む．
1. **LookupAccountName() 性能測定**　ユーザ名を LookupAccountName() 関数に与え，SID を取得するのに要した時間を測定する．
2. **LookupAccountSid() 性能測定**　1. において取得したSIDを LookupAccountSid() 関数に与え，ユーザ名，所属ドメインを取得するのに要した時間を測定する．

時間測定は，それぞれの処理を現在時刻を取得する関数 time() で挟み込み，処理に要した時間 (実時間) を計測した．

4.2.2 測定結果

測定結果から1ユーザ分の情報を取得するのに要した平均時間を表1に示す．これより，NDS 環境は PDC の SAM データベース参照に比べ2倍ほど性能が低いことがわかる．

表1　Windows 環境において1ユーザ分の情報を取得するのに要した時間

	NDS	PDC	性能差
LookupAccountName()	0.054 秒	0.024 秒	2.25
LookupAccountSid()	0.027 秒	0.013 秒	2.08

4.3 PAM 認証性能

4.3.1 測定方法

UNIX 環境における NDS PAM 認証モジュールの性能を測定するため，独自の PAM アプリケーションを作成しユーザ認証に要する時間を測定した．この性能測定プログラムは，PAM ライブラリ (Linux-PAM 0.75) を用いて，1ユーザずつ PAM 認証を行い，これに要した時間 (実時間) を計測する．なお，ユーザ認証を行なう際，現実のユーザパスワードは不明であるので，デタラメな文字列をパスワードとして与え，認証に失敗させた．すなわち，PAM 認証のための API のうち，pam_start()，pam_authenticate()，pam_end() を順に呼び出すが，このうち pam_authenticate() はエラーを返す．したがって，本来ユーザ認証に必要な pam_acct_mgmt() などを引続き呼び出すことはできないため，本測定は本来のユーザ認証に要する時間よりも甘い性能評価となる．

測定においては，

- まず NDS 認証を行ない，これに失敗したら UNIX 認証 (NIS による認証) を行なう設定 (NDS が推奨している設定)
- NDS 認証のみを行なう設定
- UNIX 認証のみを行なう設定

の3つの PAM 設定ファイルを準備し，それぞれの性能を測定した．なお，時間の測定は，測定する処理の前後に現在の時刻を取得する関数 gettimeofday() で挟み，処理に要した時間 (実時間) を計測した．

4.3.2 測定結果

307名分のユーザに対して測定用 PAM アプリケーションを実行して得られた結果を表2に示す．これより，NDS によるユーザ認証は UNIX 認証 (NIS による認証) に比べ2倍以上性能が低いことがわかる．また，NDS が想定

している PAM 認証モジュールの選択順序では，UNIX 認証のみを行なうのに比べ，3 倍以上の時間を要することがわかる．

4.4 NSS 参照性能

4.4.1 測定方法

UNIX 環境においてユーザ情報を参照するためのライブラリ関数のうち，UID (User ID) からパスワードファイルの登録項目を取り出す関数 `getpwuid()` および，ユーザのログイン名からパスワードファイルの登録項目を取り出す関数 `getpwnam()` の応答性能について測定した．

このプログラムは大きく分けて 3 つの処理からなる．

1. **ユーザ全検索** UID を 0〜19,999 までの 20,000 ユーザ分 (本システムにおいてユーザとして登録されている UID の範囲) についてそれぞれ `getpwuid()` を実行するのに要する時間を測定する．このうち，ユーザとして登録されている 8,000 名分についてはユーザ情報を取得できる．
2. **getpwuid() 性能測定** 登録されているユーザの UID を連続して `getpwuid()` 関数に与え，ユーザのログイン名を取得するのに要する時間を測定する．
3. **getpwnam() 性能測定** 登録されているユーザのログイン名を連続して `getpwnam()` 関数に与え，UID を取得するのに要する時間を測定する．

4.4.2 測定結果

NSS 設定ファイル (`/etc/nsswitch.conf`) の `passwd` エントリを切替え，NDS を参照した場合と NIS を参照した場合のそれぞれにおいて性能を測定した．

1 ユーザ分の情報を取得するのに要した平均時間を表 3 に示す．これより，NDS 環境は NIS 環境に比べ 100 倍以上も性能が低いことがわかる．この性能差は，たとえば `ls -l` コマンドを実行した際に，ファイルの所有者を表示するためファイルの i ノードに記録されている UID からユーザ名を取得するような場合に多大な影響を及ぼす．

表 3 Linux 環境において 1 ユーザ分の情報を取得するのに要した時間

	NDS	NIS	性能差
ユーザ全検索性能	1.204 秒	0.00064 秒	1881 倍
`getpwuid()`	0.071 秒	0.00069 秒	103 倍
`getpwnam()`	0.132 秒	0.00070 秒	189 倍

NDS 環境においては，ユーザ全検索性能が非常な値を示している．これは，ユーザとして登録されていない UID を `getpwuid()` に与えた際，複数存在するレプリカサーバに問い合わせながらユーザ情報ツリー上を検索するため，戻り値として `NULL` が得られるまでに秒オーダの時間を要することを表している．

以上の測定結果について，NIS あるいは PDC の性能を 1 とした場合の NDS の性能比較をまとめて図 6 に示す．これより，PAM 認証および Windows の情報参照性能は，NIS あるいは PDC を用いた場合の約半分しかない．さらに，NSS 参照性能は桁違いに劣悪であることがわかる．

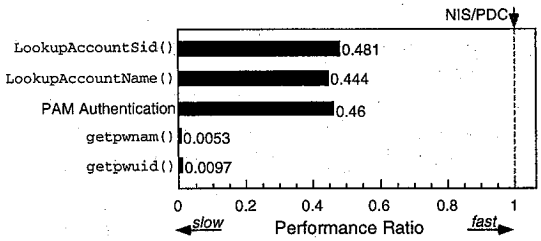


図 6 NDS の性能

4.5 負荷状態における性能測定

上述のユーザ情報参照性能の測定は，いずれも 1 台のクライアント (他のクライアントは電源 OFF 状態) から参照要求を行った際の応答性能である．したがってこれらの測定は，多人数の学生が同時に利用するという実際の講義・演習における利用形態を模擬していない．そこで，多数のクライアントから一斉にユーザ情報を参照した場合の性能を測定した．

図 7 に示すように，25 台の Linux クライアントを負荷ジェネレータとして，`getpwuid()` 呼び出しを無限に繰り返すプログラムを実行しておき，システム全体を負荷状態とする．この負荷環境において，応答性能測定用プログラムを利用し，`getpwuid()` および `getpwnam()` の応答性能を測定する．

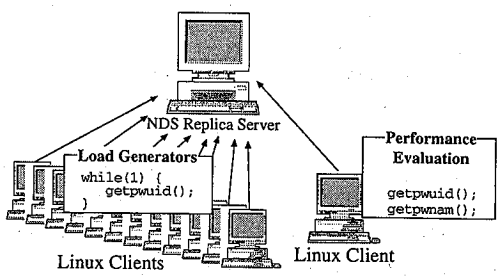


図 7 負荷状態における応答性能の測定方法

測定の結果得られた，1 ユーザ分の情報を取得するのに要した時間を表 4 に示す．表 3 と 4 を比較すると，NDS 環境，NIS 環境のいずれにおいても，ユーザ情報の参照を同時に行うと，応答性能は低下しているが，NIS 環境においては 10 倍程度であるのに対し，NDS 環境においては数千倍も低下していることがわかる．したがって，NDS 環境において多数のクライアントから同時にユーザ情報を参照する場合には，大幅な性能低下となる．

通常の Linux 環境の使用においては，ログインなどのユーザ認証の回数にくらべ，`ls -l` や `ps` コマンドの実

表2　PAM 認証に要した時間

	NDS+UNIX 認証	NDS 認証のみ	UNIX 認証のみ	性能差
実行時間	1037.74 秒	668.03 秒	305.81 秒	
認証 1 回当たりの平均時間	3.38 秒	2.17 秒	0.99 秒	2.19

表4　負荷状態において 1 ユーザ分の情報を取得するのに要した時間

	NDS	NIS	性能差
getpwuid()	137.744 秒	0.00750 秒	18,366 倍
getpwnam()	103.724 秒	0.00778 秒	13,332 倍

行，ファイルマネージャの操作といった NSS を経由したユーザ情報の参照回数は格段に多い．したがって，NDS の NSS 参照性能の劣悪さはシステム性能の大幅な低下をまねく．

5. ネームサービスのバイパスシステムの構築

性能評価の結果より，Windows のユーザ情報参照性能および UNIX の PAM 認証性能については，1 サブセット数十台規模のシステム構成では，ほぼ問題がないといえる．しかし，UNIX におけるユーザ情報参照のためのネームサービス性能の低下は，利用上多大な影響を及ぼすこととなり，実用に耐えられない．

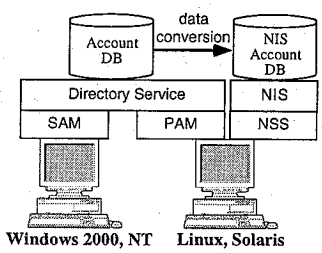


図8　ネームサービスのバイパス提供システム

そこで，図8 に示すように，ユーザ情報に関するネームサービスを NIS によってバイパス提供するシステムを設計構築した．本バイパスシステムは，PAM 認証のみに NDS を利用し，パスワード照合以外の NSS フレームによるユーザ情報は NIS を参照する．このとき，パスワードの参照は PAM 経由で NDS へ行われ不要であることから，NIS データベースにはパスワード情報を含めずセキュリティを確保した．

この仕組みにより，パスワード照合以外のネームサービスの応答性能は，NIS の性能そのものとなり，同時に，NDS レプリカサーバやネットワークの負荷も下げることができた．

6. 結　　言

ディレクトリサービスを用いた OS 混在環境下のユーザアカウント統合について，ユーザ情報を参照する際のデータフローに基づいて性能評価を行ない，

(1) アカウントデータベースの統合により，Windows および UNIX の認証のためのユーザ情報の参照性能は，2 倍程度低下することがわかった．

(2) 多数のユーザが同時に利用する高負荷時においては，UNIX のネームサービスは実用に耐えられないことがわかった．

(3) NIS を用いたネームサービスのバイパスシステムは，大きな効果があることを示した．

参 考 文 献

1) Shimano, A. and Kuramae, H.: Design and Construction of Educational Computer System Using Self-maintenance System for Files and User Identification Agent, Proc. of 9th IEEE International Workshop on Robot and Human Interactive Communication, pp. 23–28 (2000).

2) 倉前宏行，島野顕継，木村彰徳，松本政秀，亀島紘二: ディレクトリサービスを用いた教育用 PC クラスタシステムの学生ユーザアカウント管理，情報処理学会分散システム/インターネット運用技術シンポジウム 2001 論文集，pp. 93–98 (2001).

3) Kuramae, H., Shimano, A., Kimura, A., Matsumoto, M., Furuno Y. and Kamejima, K.: User Account Management of PC Cluster System for Education Using Directory Service, Proc. World Multiconference on Systemics, Cybernetics and Informatics, Vol. VIII, pp. 68-73, (2001).

4) 中山仁，大西淑雅，末永正，有田五次郎: 工学系学生のための情報処理集合教育環境の設計と構築，情報処理学会論文誌，Vol. 35, No. 11, pp. 2225–2238 (1994).

5) 安東孝二，吉岡顕，田中哲朗: 大規模計算機センターのセキュリティ対策事例，情報処理学会分散システム/インターネット運用技術研究会報告，99-DSM-16, pp. 43–47 (1999).

6) 田中哲朗，安東孝二，吉岡顕: 複数 OS 環境におけるユーザ管理，情報処理学会分散システム/インターネット運用技術研究会報告，99-DSM-16, pp. 49–54 (1999).

7) 中山仁，大西淑雅，望月雅光，山之上卓，甲斐郷子: Linux thin client を端末とする集合教育用計算機環境の構築，情報処理学会分散システム/インターネット運用技術研究会報告，2000-DSM-18, pp. 31–36 (2000).

8) 斎藤明紀: 教育用大型計算機システムにおける管理の省力化手法，情報処理学会論文誌，Vo. 41, No. 12, pp. 3198–3207 (2000).

9) 石原進: 集合型情報処理教育施設のネットワーク設計，教育システム情報学会誌，Vol. 17, No. 4, pp. 606–608 (2000).

10) Eckstein, R., C-Brown, D. and Kelly, P.: Using Samba, O'Reilly & Associates Inc. (2000).

11) http://www.sun.co.jp/software/interoperability/netlink/

12) http://www.microsoft.com/japan/products/ntserver/sfu/