

グリッドファイルシステム GSI-SFS

武田 伸悟*1, 伊達 進*1, 下條 真司*2

{stakeda,date}@ais.cmc.osaka-u.ac.jp, shimojo@cmc.osaka-u.ac.jp

概要 グリッド技術はインターネット上のストレージやプロセッサ資源などの様々なリソースを多数のユーザで共有することを可能にし、ユーザが膨大なグリッド上のリソースにどこからでもアクセス可能にする。グリッドでは信頼性のないネットワークに分散して配置されたストレージを利用するため、安全にデータを共有する手段が必要とされている。しかし、既存のファイルアクセスサービスで利便性とセキュリティの両方の要件を満たすものは少ない。本研究では、グリッドで利用できる利便性とセキュリティを備えたファイルシステム GSI-SFS を提案、実装し、グリッド上で容易かつ安全にストレージを共有することを可能とした。

GSI-SFS: A Grid File System

Shingo Takeda*3, Susumu Date*3, Shinji Shimojo*4

{stakeda,date}@ais.cmc.osaka-u.ac.jp, shimojo@cmc.osaka-u.ac.jp

Abstract Grid technology enables us to share a diversity of computational resources on the Internet. In the grid environment, users often demand a convenient method for sharing data safely among the storages, each of which is separately located across the untrusted public network. At present, few file transfer services which balance the trade-off between convenience and security exist, which leads to the difficulties of dealing with confidential data on the grid. To satisfy such users' demand, we have developed a user convenient and secure file system seamlessly integrated to the grid environment.

1 序論

近年、ネットワーク技術の発達やネットワークインフラストラクチャの整備が進んだことにより、世界中に分散したコンピュータのストレージやプロセッサ資源をグリッド [1] によって多数のユーザで共有することが行われつつある。グリッドとは、ネットワーク上のコンピュータのストレージやプロセッサ資源、さらには計測機器やセンサといったデバイスへのシームレスなアクセスを可能にすることで、ユーザに巨大な仮想コンピュータを提供する分散コンピューティング技術である。グリッドは物理、生物、医学などあらゆる分野で応用が期待されている。

しかし、現在のグリッド技術はこのような大規模なリソースの共有を可能にするまでに至っていない。最も大きな問題の一つはグリッドが非常に複雑で、利用するにはグリッドに関する専門知識が必要であり、操作性や利便性にも欠けているという点である。本研究では、特にグリッド上のストレージを容易に利用する手段が少ないことから、グリッドにおけるファイルアクセスの利便性に着目する。

グリッドはインターネット上に構築されるが、インターネットは多数のユーザが共有しており、盗聴や改竄の危険性がある信頼性のないネットワークである。しかし、企業など高いセキュリティを必要とする組織も多い。グリッド上でストレージの共有を行うためには、このような組織でも安心して利用できる高いセキュリティを提供しなければならない。

このようにグリッドのユーザは利便性とセキュリティの両方を必要としているが、セキュリティを高

*1 大阪大学 大学院情報科学研究科

*2 大阪大学 サイバーメディアセンター

*3 Graduate School of Information Science and Technology, Osaka University

*4 Cybermedia Center, Osaka University

めるために厳しいユーザ認証を行うと利便性が低下し、利便性とセキュリティはトレードオフの関係にある。本研究ではセキュリティを犠牲にすることなくデータアクセスを容易にすることを目的とし、グリッドで利用できる安全なファイルシステムの開発を行った。本研究で開発したグリッドファイルシステム GSI-SFS はユーザの利便性を改善し、グリッド上のストレージを安全に共有することを可能にする。

本論文では、第2部で GSI-SFS の開発に利用した技術について概説した後、第3部で GSI-SFS の設計について述べる。そして、第4部で実装を評価し、最後に第5部で結論を述べる。

2 基盤技術

本研究ではグリッドファイルシステム GSI-SFS を開発するために Grid Security Infrastructure (GSI) [2] と Self-certifying File System (SFS) [3] を基盤技術として用いた。本章では GSI と SFS の概要について述べる。

2.1 Grid Security Infrastructure (GSI)

GSI は Globus Toolkit (以下 Globus) [4] に含まれるコンポーネントの一つで、認証や暗号化といったセキュリティ機能を提供する。ここでは Globus と、GSI の特徴である PKI とシングルサインオンについて概説する。

Globus はセキュリティ、データアクセス、リソース管理、システムの監視といった基本的な技術を提供するミドルウェアである。グリッドを利用するためにはグリッドに共通する機能を提供するミドルウェアと、その上で動作するアプリケーションが必要である。グリッド構築のためのミドルウェアは多数あるが、その中でも Globus Project の提供する Globus Toolkit は活発に開発が進められており、多くの研究機関や企業のコンピュータに導入されている。

GSI は Globus の提供するコンポーネントの一つで、グリッドコンピューティングのためのユーザ-ホスト間の相互認証と通信のセキュリティを提供する。GSI は公開鍵基盤 (Public Key Infrastructure: PKI) の一種である。PKI では公開鍵の信頼性は認証局 (Certificate Authority: CA) によって保証されるため複雑な鍵交換をユーザや管理者が行う必要がな

/C=JP/ST=Osaka/O=Osaka University/OU=Shimojo Laboratory/
CN=Shingo TAKEDA/Email=stakeda@ais.cmc.osaka-u.ac.jp

図1: X.509 電子鍵証明書の DN の例

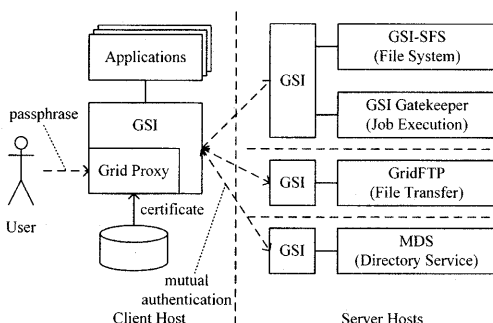


図2: GSI のシングルサインオン

く、大規模なシステムでは PKI は必要不可欠の技術となっている。PKI の標準プロトコルに X.509 がある。GSI は X.509 の電子鍵証明書を使用してユーザとホストの相互認証を提供し、Distinguished Name (DN) (図1) でユーザやホストを識別する。

グリッドではユーザは多数のホスト上の複数のサービスを利用するため、サービスを利用する度にユーザが対話的に認証を行うと利便性が大きく損われる。そのため、GSI はシングルサインオン機能を提供し、ユーザが一度パスフレーズを入力した後はグリッドプロキシが代理で認証を行うため、ユーザは透過的に様々なグリッドサービス (ジョブの投入、ファイル転送、情報検索など) を利用できる (図2)。

2.2 Self-certifying File System (SFS)

ファイルシステムはオペレーティングシステムに深く関わっているため、ファイルシステムの新規開発は非常にコストがかかる。そこで開発コストを抑えるため既存のネットワークファイルシステムを基盤としてグリッドファイルシステムを実現することを考えた。本研究では利便性、セキュリティ、柔軟性、移植性が高いことから SFS を基盤ネットワークファイルシステムに選択した。

SFS は UNIX で一般的に利用されている Network File System (NFS) の転送データを暗号化し、認証データを付加して転送するサーバ (SFS サーバ)、クライアント (SFS クライアント) および各種ツール

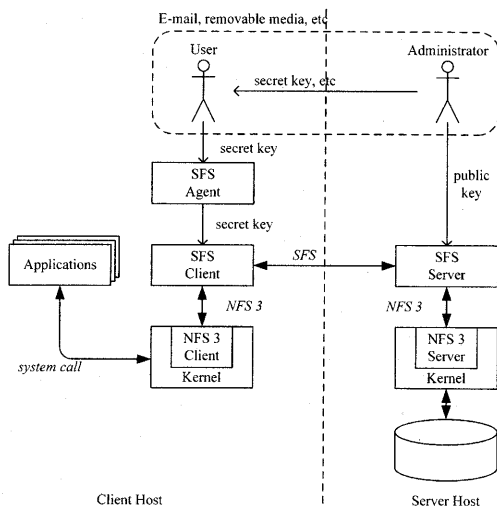


図 3: SFS の構成

(SFS エージェントなど) からなる (図 3). SFS は NFS を利用するため、ユーザはネットワーク上のファイルをローカルファイルと同様に (ネットワーク上にあることを意識せずに) 利用することができる。利便性が高い。

また、NFS がホスト-ホスト間の認証を行い転送データの機密性は保証しないのに対して、SFS ではユーザ-ホスト間の相互認証と転送データの暗号化を行い、高いセキュリティを提供する。多くのユーザがクライアントを共有することが多いグリッドにおいてはユーザ-ホスト間の認証が必要不可欠である。

SFS を用いてサーバ上のファイルシステムにアクセスするには Self-certifying pathname (自己認証パス名) と呼ばれる `/sfs/` で始まるパス名を使用する (図 4)。自己認証パス名にはホスト名とホスト ID (ホスト公開鍵などのハッシュ値) が含まれており、他のホストと重複することはない。NFS や Andrew File System (AFS) など、他の多くのネットワークファイルシステムではシステム管理者のみがマウントによってディレクトリ構造を定義することができるが、SFS では自己認証パスにシンボリックリンクを作成することで管理権限を持たないユーザでも仮想のディレクトリ構造を構築することができる。図 5 の例ではクライアントの

Location
`/sfs/@host.osaka-u.ac.jp,`
`wmdq53f44i39irfhvbsq8zmyazbf2m4g/home`
 Host ID

図 4: SFS の自己認証パス名の例

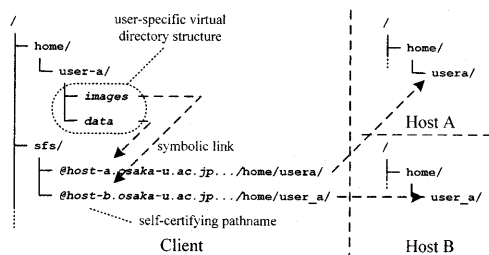


図 5: SFS による仮想ディレクトリ構造

`/home/user-a/images/` が実際にはホスト B の `/home/user_a/` であり、`/home/user-a/data` がホスト A の `/home/user_a/` であることをユーザは意識する必要がない。この柔軟性の高さも多くのユーザがクライアントを共有するグリッドにおいては重要である。

一般的にファイルシステムは強くオペレーティングシステムに依存するが、SFS は NFS を転送する仕組みであるため NFS が利用できるほとんどの UNIX システムで利用でき、移植性にも優れている。移植性の高さも多様なコンピュータが使用されるグリッドでは重要である。

3 設計アプローチ

グリッドでは利便性とセキュリティを兼ね備えたファイルアクセスサービスが必要とされている。ここではユーザの要求を分析し、既存のファイルアクセスサービスがその要件を満たさないことを示した後、利便性とセキュリティの両立を実現する GSI-SFS の設計について述べる。

3.1 ユーザの要求

グリッドを用いたストレージの共有を行なうにはセキュリティを犠牲にすることなく利便性を高める必要がある。ここでは利便性とセキュリティについてユーザの要求を分析する。ここで、ユーザはグリッ

ドや情報技術に関する深い知識を持たないことを仮定している。

グリッドの特徴として、ユーザは異なる管理ドメインに属する多数のホストを利用することが挙げられる。そのためユーザがホストごとに自分のアカウント情報を管理し、ホストを利用する度に対話的に認証を行うことは困難である。利便性を高めるためには認証を可能な限り自動化し、認証によるユーザの負担を小さくしなければならない。またファイルにアクセスする度に、そのファイルがどのホストのどのディレクトリに存在するかをユーザが意識しては効率的に作業を行うことができない。そして、グリッドはインターネット上に構築されるためネットワークの速度はLANに比べて低速であり、効率の良い転送手法が必要である。以上をまとめると、利便性を高めるためには認証の透過性、位置の透過性、効率的な転送が必要である。

グリッドはインターネット上に構築されるが、インターネットは多数のユーザが共有しており、盗聴や改竄の危険性がある信頼性のないネットワークである。しかし、企業など高いセキュリティを必要とする組織も多く、暗号技術を用いて安全な認証とデータ転送を行う必要がある。また、多くの組織のネットワークにはセキュリティを高めるためにファイアウォールが設置されており、ファイアウォール内にホストを設置することが容易であることも必要である。

3.2 既存ファイルアクセスサービス

ここではGSIを使用して利用できる既存のファイルアクセスサービスの概要とその問題点について述べる。これらのサービスはすべてGSIを使用して実装されており、認証の安全性は高く、シングルサインオンで透過的に利用できる。

GridFTP [5] は、File Transfer Protocol (FTP) の認証をGSIに置き換えたものである。GridFTPは認証のみをGSIで置き換えたものであり、データのセキュリティはFTPと同様に保護されず、データ転送のためのTCPポートが固定されておらずファイアウォールの使用が困難である。

Global Access to Secondary Storage (GASS) [6] はGlobusの提供するUNIXやC言語の標準入出力を使用してグリッド上でデータ転送を行うための

ライブラリである。GlobusにはGASSを使用してファイル転送を行うサーバとクライアントが付属しているが、GridFTPと同様にデータのセキュリティは保護されず、ファイアウォールの使用も難しい。

GSI-Enabled OpenSSH (以下GSI-SSH) [7] はUniversity of IllinoisのNational Center for Supercomputing Applications (NCSA) で保守されている、OpenSSH [8] を拡張してGSIに対応させたものである。OpenSSHはOpenSSL [9] を用いて実装されており、転送データは暗号化される。そのため、GSI-SSHでファイルの転送をする際のセキュリティは高い。また、固定されたTCPポートのみを利用するためファイアウォールの使用は容易である。

これら3つのサービスを利用するとき、ユーザはファイルにアクセスする度にファイルの存在するホストとディレクトリを意識する必要があり利便性に欠ける。また、ファイル単位の転送を行うため、ファイルの一部分しか必要としない場合でもファイル全体を転送する必要があり効率的でない。

一方、第2.2部で述べたSFSはネットワークファイルシステムであるためファイルの任意の位置にアクセスすることができる。そのため、ファイルの必要な部分だけを転送することができ効率的である。また、固定されたTCPポートのみを利用するためファイアウォールの使用は容易である。しかし、SFSを利用するにはあらかじめSFSサーバにユーザのSFS公開鍵を登録しておく必要がある。このような鍵の管理は小規模なファイル共有に適しているが、多数のホストと多数のユーザからなるグリッドで行うことは困難である。以上をまとめると表1のようになり、第3.1部で述べた全ての要件を満たすものはない。

3.3 GSI-SFS

本研究では第3.1部で述べた要件を満たすためにGSIとSFSを用い、グリッドファイルシステムGSI-SFSを開発した。表1に示したようにSFSの鍵管理の問題を解決し、透過的な認証を提供すればSFSで全ての要件を満たすことができるため、本研究ではGSIを用いてこの問題を解決する。

SFSのソースコードを変更して認証部分をGSIで置き換えることも考えられるが、SFSには外部認証

表 1: ファイルアクセスサービスの比較

	GridFTP	GASS	GSI-SSH	SFS
利便性				
認証の透過性	○	○	○	×
位置の透過性	×	×	×	○
転送の効率	×	×	×	○
セキュリティ				
認証の安全性	○	○	○	○
データの安全性	×	×	○	○
ファイアウォールの使用	×	×	○	○

プログラムを呼び出して動的に認証を行う機能があるため、これを利用する。この機能はユーザがホストを認証する目的で実装されているものであるが、GSIを使用したサーバとクライアントを作成することでユーザとホストの相互認証を実現できる。本研究では、このサーバとクライアントをそれぞれ GSI-SFS サーバ、GSI-SFS クライアントと呼ぶ。また、GSI, SFS, GSI-SFS サーバ、GSI-SFS クライアントからなるシステム全体を GSI-SFS と呼ぶ。GSI-SFS の構成を図 6 に示す。図では一つの SFS サーバのみを表記しているが、SFS クライアントは複数の SFS サーバに接続可能である。GSI-SFS では、管理者がユーザの公開鍵を登録し、秘密鍵をユーザに転送するという作業を GSI を利用して自動化する。

GSI-SFS では、SFS エージェントに GSI-SFS クライアントを外部認証プログラムとして登録しておくことで動的に鍵の転送を行う。SFS エージェントは '/sfs/gsi/ホスト名/' で始まるパスへのアクセスがあると、GSI-SFS クライアントを呼び出して鍵の取得を行う。また、'/sfs/gsi-home/ホスト名/' で始まるパスへのアクセスがあると、GSI-SFS はサーバ上のホームディレクトリの位置をクライアントに提供する。GSI-SFS での認証のおおまかな流れを以下に示す(図 6 中の番号と対応している)。

1. SFS エージェントは GSI-SFS クライアントを呼び出す
2. GSI を用いた GSI-SFS クライアントとサーバの相互認証
3. GSI-SFS サーバがユーザの SFS 鍵ペアを生成、公開鍵を登録

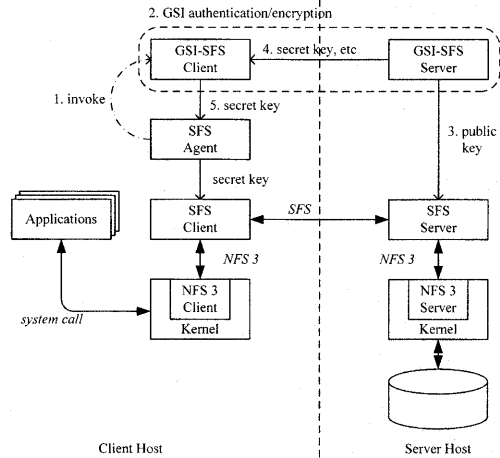


図 6: GSI-SFS の構成

4. GSI-SFS サーバがクライアントへ、生成した秘密鍵を GSI の機能で暗号化して送信
5. GSI-SFS クライアントは受信した秘密鍵を SFS エージェントに登録

以上の流れは自動的に、かつ動的に行われユーザが意識する必要はない。ユーザは既存の GSI のユーザ証明書と秘密鍵で SFS サーバに透過的にアクセスできるようになる。

GSI-SFS では動的に SFS 鍵の生成と登録を行うが、ユーザがアクセスする度にこの処理を行うことは効率的でない。鍵の生成では予測が困難な疑似乱数を生成する必要があり、鍵の生成は GSI-SFS の認証の過程で最も時間を要する処理である。そこで、GSI-SFS サーバは生成した鍵を一定時間キャッシュしておき、同じユーザからアクセスがあった場合はキャッシュにある鍵をクライアントに送信する。これにより同じホストに複数回アクセスするときの認証時間を短縮することができる。

4 評価

GSI-SFS は第 3.1 部で述べた要件を全て満たす。以下では利便性、セキュリティ、パフォーマンスについて GSI-SFS を評価する。

4.1 利便性

GSI-SFS によって、ユーザは認証やファイルの実際の位置を意識せずに、グリッド上の巨大なストレージを利用することができる。このため、既存のファイルアクセスサービスよりも GSI-SFS が利便性において優れていると考えられる。

GSI-SFS クライアントは、Globus を利用した他のアプリケーションと同様に、GSI のグリッドプロキシから証明書を取得して認証を行う。そのためユーザはシングルサインオンで透過的に利用でき、グリッド上の SFS サーバにアクセスするために SFS の鍵を登録する必要はない。また、既存の GSI の証明書を再利用でき、他のグリッドサービスと認証が二重化することがない。

GSI-SFS をストレージとして利用する場合、ユーザが自由に読み書きできるホームディレクトリの位置が重要であるが、グリッド上の多数のホストにある自分のホームディレクトリの位置をユーザが全て把握することは難しい。そこで GSI-SFS はホームディレクトリの位置情報をクライアントに提供し、ユーザは '/sfs/gsi-home/ホスト名/' で始まるパスを使用してホームディレクトリの位置を意識せずにアクセスすることができる。そして、グリッド上のファイルにアクセスするために特別なアプリケーションを必要とせず、既存のアプリケーションを利用できる。

GridFTP などの従来のファイルアクセスサービスではファイルの単位の転送しか行えなかった。一方、GSI-SFS はファイルシステムとして利用できるためファイルの任意の位置にアクセスでき、効率の良い転送が行える。

4.2 セキュリティ

GSI-SFS では GSI と SFS の両方の認証が行われる。GSI は X.509 電子鍵証明書を使用して相互認証を行い、SFS は公開鍵暗号で相互認証を行う。GSI と SFS は OpenSSL でも利用されている代表的な暗号アルゴリズムを使用しており、これらのアルゴリズムは一般的な用途に十分な暗号強度を持っている。

SFS は改竄を防止するために転送データに暗号ハッシュ関数による認証データを付加し、盗聴を防止するために暗号化する。ここでも代表的なアルゴ

表 2: 測定環境

	サーバ	クライアント
Linux	2.4.20	2.4.18
Globus	2.2.4	2.2.4
SFS	0.7.2	0.7.2
CPU	Pentium4 2.4GHz	Pentium4 1.6GHz
メモリ	1GB DDR	1GB SDR
ネットワーク	Intel 82554GC	Intel 82554GC
ハードディスク	Western Digital WD200EB	Maxtor 4K040H2

リズムが使用されている。

GridFTP や GASS ではデータの転送に TCP の不特定のポートを利用するためファイアウォールを使用することが難しいが、GSI-SFS では特定のポートのみを使用するため GSI-SSH と同様にファイアウォールを使用してセキュリティを高めることができる。以上より、GSI-SFS のセキュリティ強度は GSI-SSH と同等で、他の転送データの暗号化を行わないサービスよりも高いと考えられる。

4.3 パフォーマンス

SFS はセキュリティや移植性を重視して設計されており、パフォーマンスはあまり重視されていない。しかし、グリッドでは大きなファイルを転送することが多く、特にスループットが重要である。ここではパフォーマンスに関する実験結果を示し、最後に考察を述べる。

今回パフォーマンスの測定に用いた 2 台の PC の構成を表 2 に示す。サーバ-クライアント間はクロスケーブルで接続されており、これら 2 台のホストが回線を占有することができる。

4.3.1 スループット

SFS では暗号アルゴリズムを用いて転送データの暗号化を行っておりスループットの低下が予測される。そこで、ここではファイルをサーバからクライアントへコピーするのに要する時間を測定することによって GSI-SFS のスループットを測定し、他のファイルアクセスサービスと比較する。測定用のファイルは内容が疑似乱数で構成されるファイルである。ファイルサイズはメインメモリ上のキャッシュの影響を避けるため物理メモリの 2 倍である 2GB とした。回線速度 1000 M bit/s での測定結果を図 7 に、100 M bit/s での測定結果を図 8 に示す。スループットは 2GB を、time コマンドで測定した所要時間で

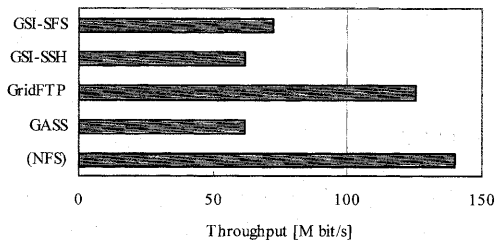


図 7: 1000 Mbit/s LAN でのスループット

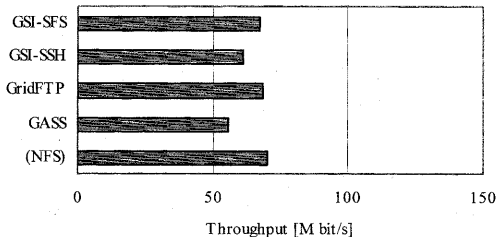


図 8: 100 Mbit/s LAN でのスループット

割ったものとし、それぞれ 3 回ずつ測定して相加平均をとった。

これらのサービスのうち、転送データの暗号化を行わないものは GridFTP, GASS および NFS である。NFS が最も高速であるが、NFS はグリッドの認証で利用できない。GridFTP は高速で、機密性と整合性を必要としないファイルをファイル単位で転送するのに適していると言える。しかし、機密性や整合性を必要とするファイルの転送には GridFTP は利用できず、GSI-SFS や GSI-SSH を利用する必要がある。回線速度が 100 Mbit/s のときは回線速度がボトルネックとなり、NFS と GridFTP のスループットが大幅に低下している。そのため GSI-SFS や GSI-SSH との差が縮まり、暗号化によるスループットへの影響がほとんどなくなっている。

4.3.2 認証に要する時間

GSI-SFS は SFS に変更を加えずに、GSI の認証で SFS サーバにアクセスすることを可能にする。そのため GSI と SFS の両方で認証が行われ、SFS の認証を GSI で置き換える場合に比べて認証に多くの時間を要する。そこで認証によるオーバーヘッドを調べるために、GSI-SFS, GSI-SSH, GridFTP を

表 3: 0 B のファイルのコピーに要した平均時間

	平均所要時間 [s]
GSI-SFS	
鍵生成あり	1.66
キャッシュの鍵を使用	0.23
エージェントに登録済み	0.005
GSI-SSH	0.33
GridFTP	0.36

使用して 0 B のファイルをサーバからクライアントへコピーするのに要する時間を測定した。測定では time コマンドを使用してそれぞれ 10 回 (SFS のみ 100 回) 繰り返す時間を測定し、10 (SFS のみ 100) で割った結果を平均所要時間とした。測定結果を表 3 に示す。

結果を見ると、GSI-SFS で最も時間がかかる場合、すなわちサーバに鍵がキャッシュされておらず新規に生成する場合、ファイルのコピーが完了するまでに 1.66 秒の時間を要することが分かる。これは GSI-SSH, GridFTP などと比べて 5 倍程度遅い。しかしサーバに鍵がキャッシュされている場合、GSI-SFS の方が高速である。同じホストに複数回アクセスする場合、2 回目以降は SFS エージェントに鍵が登録された状態であるため、GSI-SFS クライアントは呼び出されず非常に早い応答が得られる。

4.3.3 パフォーマンスに関する考察

GSI-SFS では暗号化によるスループットの低下が生じ、1000 Mbit/s の回線速度では GridFTP の半分程度のスループットとなった。しかし、GSI-SFS では認証のオーバーヘッドが小さい場合が多く、小さなファイルを多数転送する場合は GSI-SFS のほうが有利である。また、SFS はファイルシステムであるため、ファイルの任意の位置にアクセスすることができ、ファイル全体の転送が終了するのを待たずに得られたデータから処理することができる。

このように GSI-SFS では暗号化によるスループットの低下はあるが、応答が早く、効率が高い。実験では 2 台の PC が帯域を占有できる LAN の環境で測定を行ったが、グリッドは多数のユーザが共有するインターネット上に構築されるため回線速度は実験環境よりも遅い場合がほとんどである。低帯域になればサービスによるスループットの差は小さくな

り、効率の高さが重要となる。効率の高い GSI-SFS はグリッドに適したファイルアクセスサービスであると考えられる。

5 結論

本研究では、GSI と SFS を用いて利便性とセキュリティを備えたグリッドファイルシステム GSI-SFS を設計し、GSI-SFS サーバとクライアントを実装した。GSI-SFS サーバとクライアントは GSI を用いて安全に SFS の鍵管理を自動化する。GSI-SFS はユーザの利便性を改善し、グリッド上でストレージを安全に共有することを可能にする。

GSI-SFS により、ユーザはグリッド上の巨大なストレージへの透過的なアクセスが可能となる。ユーザはシングルサインオンで認証を意識することなくグリッド上のファイルへアクセスできる。

また、ユーザは管理権限なしにユーザ独自の仮想のディレクトリ構成を構築することができ、ファイルの実際の位置をアクセスする度に意識する必要がなくなる。そして、グリッド上のファイルにアクセスするために特別なアプリケーションを必要とせず、既存のアプリケーションでグリッド上のファイルにアクセスできる。

GSI-SFS では転送データの暗号化を行うため、暗号化しない場合に比べてスループットが低下する。しかし、GSI-SFS はネットワークファイルシステムとして利用できるためファイル単位の転送サービスに比べて効率が良く、頻繁にアクセスする場合は認証のオーバーヘッドが小さい。グリッドは低速なインターネット上に構築されるため、多くの場合スループットよりも効率の方が重要である。

今後の課題としては、本ファイルシステムをより有用なものにできるよう、システムを実際に利用するユーザの観点からシステム改善をおこなっていくことを考えている。

謝辞

本研究は科学研究費補助金特定領域研究 (C)「Grid 技術を適応した新しい研究手法とデータ管理技術の研究」(13224059)の助成を受けて行われた。また、本研究は文部科学省科学技術振興費主要 5 分野の

研究開発委託事業の IT プログラム「スーパーコンピュータネットワークの構築」の一環として実施された研究成果の一部である。

参考文献

- [1] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal Supercomputer Applications*, Vol. 15, No. 3, January 2001.
- [2] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. A national-scale authentication infrastructure. *IEEE Computer*, Vol. 33, No. 12, pp. 60–66, December 2000.
- [3] David Mazières. *Self-certifying File System*. PhD thesis, Massachusetts Institute of Technology, May 2000.
- [4] I. Foster and C. Kesselman. The globus project: A status report. In *Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop*, pp. 4–18, 1998.
- [5] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Meder, and S. Tuecke. GridFTP protocol specification. In *GGF GridFTP Working Group Document*, September 2002.
- [6] J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke. GASS: A data movement and access service for wide area computing systems. In *The Sixth Workshop on Input/Output in Parallel and Distributed Systems*, pp. 78–88, Atlanta, GA, May 1999. ACM Press.
- [7] NCSA. GSI-Enabled OpenSSH. <http://www.ncsa.uiuc.edu/Divisions/ACES/GSI/openssh/>.
- [8] OpenBSD. OpenSSH. <http://www.openssh.com/>.
- [9] The OpenSSL Project. OpenSSL: The open source toolkit for SSL/TLS. <http://www.openssl.org/>.