グリッドにおける大規模ファイル共有システムの構築

武田 伸悟[†]，伊達 進[†]，下條 真司[‡]
{takeda,sdate}@ist.osaka-u.ac.jp, shimojo@cmc.osaka-u.ac.jp

概要　ネットワークインフラストラクチャの普及と高速化に伴って，グリッドは様々な分野に導入されつつある．しかし，グリッドではパフォーマンスを重視するあまりセキュリティが犠牲にされることが多く，安全にグリッド上のデータにアクセスする手段は少ない．このことが，製薬，医療などに代表される機密データを扱う分野にもグリッドを導入し，大規模な組織間の連携を実現することの障害となっている．そこで，我々はGSI-SFS を開発し，安全かつ容易にグリッド上のファイルにアクセスすることを可能とした．本論文では主にGSI-SFS の適用形態について議論する．

Construction of Large-scale File Sharing Systems on the Grid

Shingo Takeda[†], Susumu Date[†], Shinji Shimojo[‡]

Abstract As faster network infrastructures spread, the grids are getting adopted in variety of more fields. People in the fields such as drug manufacturing and medical treatment also desire to realize large-scale interorganizatinoal collaboration applying the grid technology. However, in the grid, security is likely to be sacrificed to achieve high performance, and which makes it difficult for the people handling confidential data to share them. To solve the problem, we have developed the GSI-SFS which facilitates secure access to files on the grid. Applications with the GSI-SFS are mainly described in the paper.

1　はじめに

グリッド [1] では共通の目的を持った組織や個人が集まって仮想組織 (Virtual Organization: VO) を形成し，組織の枠を越えてリソースの共有を行うことができる．近年，グリッドの概念は物理学や天文学などで必要となる大規模計算のためだけでなく，大規模なデータの共有，さらには大規模なサービスの共有にも導入されるようになった．ネットワークインフラストラクチャの普及と高速化に伴い，様々な分野でグリッドを導入し，組織間の連携を深めてゆくことが検討されている．

これらの分野のうち，製薬，医療など企業秘密や個人のプライバシーをグリッドで扱おうとするところではデータの機密保護が不可欠である．しかしながら，現在のグリッドでは高いパフォーマンスを求めるあまりセキュリティが犠牲とされる場合が多い．グリッドではインターネットが利用されるが，インターネットは公共ネットワークであるため信頼性がなく，暗号化や署名を行わなければ盗聴や改竄，成りすましによる被害に逢う危険性がある．一方，データに対して暗号化や署名といった CPU に大きな負荷のかかる処理を行うとパフォーマンスが低下するため，グリッドで利用でき，データを暗号化して転送するソフトウェアは少ない．

我々はインターネットを介して構築されるグリッドにおいてもデータの機密性，整合性を保護することができるデータ共有手段が必要であると考え，グリッドファイルシステム GSI-SFS [2] を開発した．GSI-SFS は UNIX の標準ファイル入出力を利用する既存のアプリケーションを使用して，グリッド上のファイルに安全かつ容易にアクセスすることを可能

† 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
‡ 大阪大学 サイバーメディアセンター
Cybermedia Center, Osaka University

とする．

本論文では，まず第 2 章でグリッドにおけるデータ共有の現状と問題点ついて述べ，第 3 章ではその問題点を解決する GSI-SFS の概要について述べる．第 4 章では 分散計算と機密データ共有の 2 つのケースを想定し GSI-SFS がどのように用途に適用できるか，またはできないかを分析する．最後に第 5 章でまとめと今後の課題について述べる．

2 グリッドにおけるデータ共有の現状

現在, グリッド環境を構築するミドルウェアとしては Globus Project[*1] が開発している Globus Toolkit [3] (以下 Globus と略記する) が最も普及している．Globus はファイル転送機能として, Grid File Transfer Protocol (GridFTP) [4] を実装している．GridFTP は FTP に Globus の認証手法を付加し，TCP のコネクションを複数確立することでスループットの向上が可能なプロトコルである．

GridFTP を含む全ての Globus のアプリケーションはユーザとホスト間の相互認証を行うために Grid Security Infrastructure (GSI) [5] と呼ぶセキュリティ機構を使用する．GSI は Globus の主要コンポーネントの一つで， X. 509 身分証明書を使用して PKI に基づいた安全な認証を提供する．また，GSI ではプロキシと呼ばれる仕組みを通して，ユーザは一度パスフレーズを入力するだけで全ての GSI を使用したアプリケーションを利用でき，シングルサインオンを実現している．

GSI を使用した広域分散データ管理システムとして，Storage Resource Broker (SRB) [6] と Grid Datafarm (Gfarm) [7] が挙げられる．SRB はグリッド上のファイルやデータベースといった異なる種類のデータリソースを統一的に検索したり，アクセスしたりすることを可能にする．一方，Gfarm は TB/s オーダの帯域で PB オーダのデータを扱うために設計された広域分散ファイルシステムである．

これら GridFTP，SRB，Gfarm は GSI を使用しているため，X. 509 身分証明書により安全な認証が可能である．しかしながら，これらはパフォーマンスを重視しており転送データの暗号化は行わないため，インターネットのような公共ネットワークを介して使用した場合データの機密性は保証されない．我々は, データ機密性の保護が不可欠な分野にグリッドを導入してゆくためには, 安全な認証だけでなく，データの機密性と整合性の保護も行えるファイル転送手段が必要であると考え，グリッドファイルシステム GSI-SFS を開発した．次の第 3 章では GSI-SFS の概要について述べる．

[*1] http://www.globus.org/

3 ファイルシステム GSI-SFS の概要

GSI-SFS は GSI の認証で Self-certifying File System (SFS) [8] を利用できるように拡張したものである．SFS は NFS によって送信されるデータを暗号化，署名して中継する (カーネルレベルではなく) ユーザレベルのファイルシステムである．同質なコンピュータが高速で信頼性のあるネットワークで接続されたクラスタとは異なり，グリッド上のコンピュータは異質で, 比較的低速な信頼性のないネットワークで接続されている．このような特性から, 我々はセキュリティと移植性に優れた SFS を，グリッドで利用できるように拡張した GSI-SFS を開発した．GSI-SFS と SFS，NFS の関係を図 1 に示す．SFS は NFS をインターネットで使用できるように拡張し，GSI-SFS は SFS をグリッドで使用できるように拡張したものである．GSI-SFS の詳細な内部構造については文献 [2] を参考にされたい．GSI-SFS は GSI の特長である，

- X. 509 身分証明書を使用した PKI に基づく安全な認証
- プロキシを使用したシングルサインオン

および SFS の特長である，

- ネットワーク上を流れるデータの暗号化, 署名
- 標準的な NFS を中継することによる高い移植性
- 非特権ユーザによるオンデマンドなマウント

を併わせ持つ．さらに異質なグリッド環境のために，

- 統一的なホームディレクトリへのアクセス

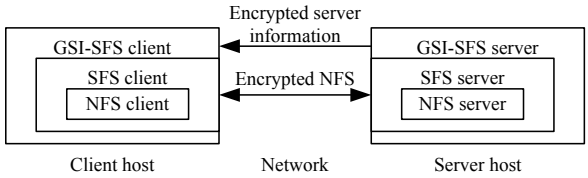

図 1: GSI-SFS と SFS，NFS の関係

を提供する．

GSI-SFS を使用して, クライアントにログインしているユーザがグリッド上のファイルにアクセスする例を図 2 に示す．‘`gsisfskey --initiate`’ コマンドでは, 必要に応じてユーザにパスフレーズを問い合わせて GSI のプロキシを有効にし, ‘`/sfs/gsi`’ で始まる GSI-SFS のパスを SFS のクライアントに登録する．この後はユーザは認証を意識することなく，シングルサインオンで GSI-SFS サーバのファイルにアクセスできる．GSI-SFS では明示的なマウント，アンマウント操作は必要なく，ユーザがアクセスを行ったときにオンデマンドでサーバに接続し，認証が行われる．また，切断はタイムアウトによって自動的に行われる．ユーザは UNIX の標準ファイル入出力を使用する既存のアプリケーションでファイルの読み書きが行え，標準の UNIX コマンド (`ls`, `cp`, `mv` 等) でファイルを操作し，シンボリックリンクを利用することもできる．GSI-SFS のパスにシンボリックリンクを張ることがマウントに相当すると言え，この操作は非特権ユーザでも行うことができる．最後の例では ‘`/sfs/gsi-home/`’ で始まるパス名を使用している．これはサーバ上のユーザのホームディレクトリを指しており，ユーザが全てのサーバ上のホームディレクトリを把握しておく必要がない．このように，GSI-SFS はグリッド上のファイルへの安全かつ利便性の高いアクセス手段を提供する．

4 GSI-SFS によるファイル共有システム

グリッドでは CPU，ストレージ，センサ，測定デバイスなど様々なリソースが共有される．ここでは CPU を共有した分散計算環境，およびディスクを共有した機密データの共有の 2 つの典型的なケースにおいて，GSI-SFS がどのように利用できるかについて考察する．

```
$ gsisfskey --initiate
Enter GRID pass phrase for this identity:
$ less /sfs/gsi/host.domain/pub/data.txt
$ ln -s /sfs/gsi/host.domain/pub storage
$ cp storage/data.txt .
$ ./mycalc \
   < /sfs/gsi-home/host.domain/data.txt \
   > /sfs/gsi-home/host.domain/result.txt
```

図 2: GSI-SFS の使用例

表 1: 実験に用いた PC の構成

CPU	Intel Xeon 2.8GHz (Hyper-threading) × 2
Memory	2GB
Disk	Maxtor 4A250J0
Network	Intel 82545EM (64-bit PCI)
OS	RedHat Linux 9
Globus	2.4.0
SFS	0.7.2
GSI-SFS	0.0.6a

4.1 分散計算環境への応用

現在，最も一般的なグリッドの利用形態は CPU を共有した広域分散計算であろう．計算プログラムを作成する上で最も手軽なデータ保存方法はファイルとして出力することである．そして，その出力ファイルを別のプログラムの入力ファイルとして使用することがしばしば行われる．また, プログラム自身もファイルである．このように，CPU を共有する場合においても遠隔ファイルアクセスは非常に重要である．ここでは GSI-SFS のパフォーマンスを評価し，GSI-SFS がどのような用途に適用可能かを考察する．

今回のパフォーマンスの測定では表 1 に示す同じ構成の 2 台の PC をサーバとクライアントとして用いた．これら 2 台の PC はクロスケーブルで直結されているためネットワーク帯域を独占することができ，信号伝達に起因する遅延は非常に小さい．

GSI-SFS には複数のサーバに分散してデータを保存し，パフォーマンスを向上させたり，信頼性を高めたりする機能は備わっていない．そのため，GSI-SFS サーバのスループットでは (a) ローカルファイルシステムのスループット，(b) CPU がファイルアクセス，暗号化処理，プロトコル処理などを行うスループット，(c) ネットワークのスループットのうち最小のものがボトルネックとなる．これらのうちボトル

表 2: ベンチマーク結果 (1 [MB] = 2^{20} [B])

	Read [MB/s]	Write [MB/s]
(a) ローカル FS	39.5	26.3
(c) 実効ネットワーク帯域	110.1	111.1
(d) GSI-SFS スループット	16.8	8.0

ネックとなる部分を調べるために，先に述べた実験環境でベンチマークを行った．結果を表 2 に示す．まず，ファイルシステムのベンチマークツールである Bonnie++ 1.03a[*2]をサーバ上で実行し，表中 (a) に示すブロック読み込みとブロック書き込みのスループットが得られた．次に，(b) CPU のスループットを測定することは困難であるため，(c) ネットワークのスループットを Netperf 2.2pl2[*3]を使用して測定し，表中 (c) に示す TCP での実効ネットワーク帯域が得られた．そして，Bonnie++ をクライアントで実行し GSI-SFS サーバのスループットを測定し，表中 (d) に示すスループットが得られた．これらの結果より，(a)，(b)，(c) のうち最小のものがボトルネックとなり (d) の上限となるため，この実験環境においては (c) > (a) > (b) となっており，CPU がボトルネックとなっていることが分かる．ここではクロスケーブルで直結した帯域 1000Mb/s の 1000BASE-T という高速なネットワークで実験を行ったためこのような結果となったが，負荷の高いネットワークや 100Mb/s 以下のネットワークではネットワークの帯域がボトルネックとなる．

大きなファイルを転送する場合はスループットが重要であるが，小さなファイルを多数転送する場合はスループットよりも認証などによるオーバヘッドが問題となる．GSI-SFS では，アクセスが発生し以下の流れで認証が行われる場合に最もオーバヘッドが大きくなる．

1. GSI を使用した X.509 の相互認証
2. サーバによる SFS ユーザ鍵の生成
3. 生成した鍵のクラアントへの転送，登録
4. SFS の鍵を使用した相互認証
5. ファイルアクセス

[*2] http://www.coker.com.au/bonnie++/
[*3] http://www.netperf.org/netperf/NetperfPage.html

表 3: 0 B のファイルのダウンロードに要した平均時間

実行された手順	平均所要時間 [s]
1, 2, 3, 4, 5	1.21
1, 3, 4, 5	0.16
4, 5	0.07
5	0.005

特に 2 の処理では，セキュリティ上，推測されにくい擬似乱数の生成が必要となり他の処理と比べて長い時間を要する．そこで，GSI-SFS では一度生成した SFS 鍵を一定時間キャッシュする．これにより，同じサーバに繰り返しアクセスする場合は 2 の処理を省略することができる．また，SFS クライアントは鍵の登録時に指定した期限が切れるか，終了されるまで鍵を保持するため，SFS クライアントにすでに鍵が登録されていれば 1，2，3 の処理を省略できる．さらに，SFS ではファイルアクセス終了後も一定時間コネクションを保持するので，すでにコネクションが張られている場合は 1，2，3，4 の処理を省略できる．このオーバヘッドを測定するために，GSI-SFS を使用して先述した実験環境において 0 B のファイルをダウンロードするときに要した平均時間を測定した．結果を表 3 に示す．この結果より，同じサーバ上のファイルに繰り返しアクセスする場合はオーバヘッドが非常に小さいが，異なる多数のサーバにアクセスする場合はオーバヘッドは最大 1 秒以上と大きくなることが分かる．

これまでに述べてきたように，GSI-SFS では高速なネットワークを使用しても CPU がボトルネックとなり高いスループットが得られず，多数のサーバに分散した小さなファイルにアクセスするとオーバヘッドが大きい．そのため，分散計算において GSI-SFS が最も有効であるのは，多くの場合 100Mb/s 以下のネットワークで接続されているユーザの端末からグリッド上にあるサーバのストレージにアクセスする場合であると考える．この場合，転送されるファイルとしてまずプログラムやスクリプト，設定ファイルなどが挙げられる．これらのファイルは測定データや実験データとは異なり，改竄された場合には悪意を持った第三者にシステムへの侵入を許してしまう危険性がある．GSI-SFS では転送側がデータに署名し，受信側で検証して改竄を防止することができる．

プログラムを計算サーバに転送し実行すると，サー

バ上に結果がファイルとして保存されることが多い．GSI-SFS ではこのような結果ファイルや共有ファイルに端末上のアプリケーションを使用してアクセスでき，例えば可視化アプリケーションなどに有効であると考える．端末上のアプリケーションはローカルファイルと全く同様にグリッド上のファイルにアクセスすることができ，GSI-SFS のパスにアクセスすると，オンデマンドで相互認証が行われる．そしてファイルにアクセスすると，アクセスされた部分がオンデマンドで転送され，一度読み出したデータは OS によってキャッシュされる．これらにより，アプリケーションは効率的にグリッド上のファイルにアクセスすることができる．また，分散計算ではデータの保存領域としてユーザのホームディレクトリを使用することがしばしば行われるが，GSI-SFS では ‘`/sfs/gsi-home/`’ で始まるパス名を使用することでサーバ上のホームディレクトリにアクセスでき，サーバによってホームディレクトリの位置が異なるグリッド環境においても統一的に扱うことができる．そして，クライアントにグローバルアドレスが不要であるため，グローバルアドレスを持たない端末やクラスタのノードからも NAT 越しにアクセスが可能である．これからは NAT に代わり，IPv6 が採用されることも多くなると予測される．GSI-SFS では SFS から拡張した部分については IPv6 に対応しているが，SFS 本体は IPv6 に未対応でるため，SFS のソースコードを変更して IPv6 に対応させることを検討している．

4.2 機密データ共有への応用

GSI-SFS はネットワークを介して転送される全てのデータを暗号化，認証することで盗聴，改竄を防止する．現在の SFS ではシンプルな ARC4 をベースとしたアルゴリズムでデータの暗号化を行っているが，より暗号強度が高いと考えられている AES への移行作業もなされている．

GSI-SFS では，認証に GSI を使用する．GSI は PKI に基づいており，各ユーザと各ホストは認証局 (Certificate authority: CA) から発行された X.509 身分証明書，および信頼する全ての認証局の証明書をもっている必要がある．組織の枠を越えてリソースを共有するには，図 3 に示すように (a) 全ての組織

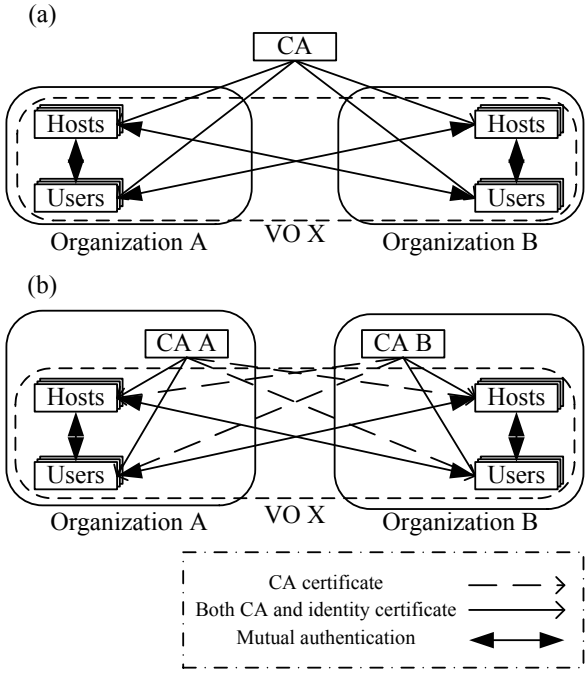

図 3: 仮想組織の形成と認証局の運営

が信頼する一つの組織が認証局を運営するか，または (b) 各組織が認証局を運営し認証局の証明書を全ての組織に配布する．これにより仮想組織を形成し，組織の枠を越えた相互認証が可能となる．

GSI-SFS サーバはユーザの認証に成功すると，Globus で設定されている許可情報と照合し，ユーザの証明書に記述されたグリッドでの名前をローカルシステムのユーザ ID にマッピングする．アクセスコントロールはマッピングされたユーザ ID に基づき，OS によって行わる．サーバでは (1) どのディレクトリをエクスポートするか，(2) グリッドのユーザをどのユーザ ID にマッピングするか，の二点でアクセスコントロールを設定する．しかし，これだけでは柔軟なアクセスコントロールを行うことが難しい．AFS で採用されているような Access Control List (ACL) を SFS に実装する研究も行われており [9]，GSI-SFS においても ACL に対応可能であるか調査中である．

GSI-SFS ではネットワークを流れるデータは暗号化されるが，ディスクに保存されるデータは暗号化されない．これは内容を見られてはならない組織が管理するサーバには機密データを保存できないことを意味する．しかし非常に大きなデータを，大きなストレージを保有する組織に安全に預けたいという

要求もあり，GSI-SFS でもクライアント側で暗号化して保存できるようにすることが可能か調査中である．製薬会社などでは，公共データベースなど，一般に公開されているデータにアクセスする場合でも，そのアクセス情報，例えばどの企業がどのタンパク質のデータにアクセスしているか，ということも機密となる．GSI-SFS ではネットワークを流れるデータは全て暗号化されるため，アクセス情報が盗聴されることを防止できる．さらに，同じクライアントにログインしているユーザでも，他のユーザがどこのサーバのどのファイルにアクセスしているかを知ることはできない．しかし，クライアントの特権ユーザからは無防備であり，信頼できるクライアントを使用することが前提となる．

GSI-SFS は SRB のような情報を検索するための機能は有していない．しかしながら GSI-SFS のパス名は URL のように世界的に一意であり，この性質を利用して WWW やデータベースなどを使用して GSI-SFS とは独立した検索システムを構築することは可能である．

5　まとめと今後の課題

製薬，医療に代表される機密性の高いデータを処理する分野にもグリッドを導入し，大規模な組織間の連携を実現するためには，暗号技術を使用しインターネットを介しても機密性と整合性を保護できるデータ転送手段が必要である．そこで我々は GSI-SFS を開発し，既存のアプリケーションでグリッド上のファイルに安全かつ容易にアクセスすることを可能とした．

本論文では広域分散計算において, GSI-SFS はユーザの端末からグリッド上のサーバにあるファイルへアクセスすることに適していることを述べた．また機密データ共有においては，信頼できる組織間で信頼できないネットワークを介してファイルを共有することに適していることを述べた．

今後の課題としては，まず SFS を IPv6 へ対応させ，IPv6 ネットワークで GSI-SFS を利用可能にすることが挙げられる．そして，GSI-SFS で ACL を使用した柔軟性の高いアクセスコントロールと，クライアント側で暗号化してサーバに保存する機能を実現する方法を検討していく予定である．

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] 武田伸悟, 伊達進, 下條真司. グリッドファイルシステム GSI-SFS. 情報処理学会研究報告 2003-OS-93, pp. 97–104, May 2003.

[3] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The Internatinoal Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 2, pp. 115–128, Summer 1997.

[4] 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.

[5] 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.

[6] A. Rajasekar, M. Wan, and R. Moore. MySRB & SRB - Components of a data grid. In *The 11th International Symposium on High Performance Distributed Computing (HPDC-11)*, July 2002.

[7] 建部修見, 森田洋平, 松岡聡, 関口智嗣, 曽田哲之. ペタバイトスケールデータインテンシブコンピューティングのための Grid Datafarm アーキテクチャ. 情報処理学会論文誌: ハイパフォーマンスコンピューティングシステム Vol. 43, No. SIG 6 (HPS 5), pp. 184–195, September 2002.

[8] D. Mazières. *Self-certifying File System*. PhD thesis, Massachusetts Institute of Technology, May 2000.

[9] G. Savvides. *Access Control List for the Self-Certifying Filesystem*. Master thesis, Massachusetts Institute of Technology, May 2000.