システム改変なしの GRID 相互連携技術の開発

（独）日本原子力研究開発機構

■ 執筆者Ｐｒｏｆｉｌｅ ■

2005 年　科学技術振興機構　博士研究員
2006 年　（独）日本原子力研究開発機構
システム計算科学センター（現職）

櫛田　慶幸

■ 論文要旨 ■

地理的に分散した資源を共有するシステムとして、GRID が使われるようになった。しかしながら、GRID の研究開発が各研究機関で独立に行われたため、数々の GRID を構成するソフトウェア（GRID ミドルウェア）が互換性を持たずに開発されてきた。このような現状を打破すべく、日本原子力研究開発機構では異なる二つ GRID システムを改変することなしに接続する技術を開発し、ドイツ・シュツットガルト・ハイパフォーマンスコンピューティングセンターにて運用されている UNICORE との接続に成功した。開発した相互接続システムでは、計算機資源共有に必要な機能である、ファイル管理、資源情報探査、ジョブ実行を両システムからシームレスに（互いの資源の所属を意識することなく）実行できる。本稿においては、ファイル管理機能に特に重点を置き、ファイルの転送性能の測定を行い、増加分が転送時間と同程度であることが確認できた。

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

GRID 環境とはネットワークを介して広域に散在する計算資源, 情報資源, 人などのリソースを仮想的に統合して利用するための基盤技術である。究極的な GRID の目標は、全世界に渡るシームレスな資源共有環境を提供することにある。しかしながら、Globus プロジェクト[１]に端を発し、様々なプロジェクトが開かれ各々独自の GRID ミドルウェアを開発したため、初志と相反するように互換性のないミドルウェアが乱立することとなった。このような状況を打開すべく標準化が進められているが[２]、それに対応するためにはミドルウェアの修正が必要であり、現在運用されている GRID に対しては管理者・ユーザーの双方に負担が大きい。そのため、本研究では、実運用中の GRID を停止することなく相互連携を行う方法を開発することを目的とし、IT Based Laboratory (以下 ITBL という)と Uniform Interface to Computing Resources(以下 UNICORE という)という二つの GRID 間で通訳を行う、連携サーバーの開発を行った。結果、運用中の GRID を停止させることなく相互連携を行うことができ、二つの GRID に接続された計算機を利用することが可能となった。

近年、GRID が注目を集めるに至った背景は、大きく分けて次の２種類に分けることができる。すなわち、

- 扱うべき問題の大規模・複雑化
- ネットワークの性能向上

である。まず、前者について詳しく述べる。

近年, コンピュータシミュレーションなどにおいて扱う対象問題は大規模, 複雑化の一途を辿っている。流体解析の直接シミュレーション,量子力学に基づく第一原理的シミュレーション、タンパク質折り畳み、である。これらのシミュレーションが要求するコンピュータ資源は、莫大なものとなっている。他方、ITER（現在、フランスに建設中の核融合実験装置）や、LHC（スイス・ジュネーブに建設された大規模陽子加速器）などの大規模実験施設から出力されるデータも取り扱いが困難になりつつある。また、これらのような、世界に唯一の実験設備から出力されるデータは、世界中の研究機関に所属する研究者間でシームレスに情報共有が行われる必要がある。

続いて後者について述べる。ネットワーク性能はプロセッサーの性能向上を上回る勢いで向上しているといわれている。ギルダーの法則[３]にしたがえば、６ヶ月で２倍高速化することが知られている。これは、プロセッサーの高速化法則でよく知られるムーアの法則が 18 ヶ月で２倍のであることを考えれば、驚異的な速度であるといえる。このような進歩は、ネットワークを介した資源利用の可能性を大きく開いた。例えば、パーソナルコンピュータのハードディスク接続規格である SATA 規格は最大値 1,200Mbps であるのに対し、ネットワークの標準的接続規格である Gigabit Ethernet の転送速度は 1,000Mbps となっており、ほぼ同じといって差し支えない水準である。このような、ネットワークの高速な進歩のため、すくなくともデータの保管・解析に関する限り、高級な大型コンピュータハードウェアを一箇所に設置し利用するよりも、分散した環境に設置された比較的小型で安価なコンピュータを接続することがコスト的に優位になる可能性が示唆されるようになった。

このような背景を踏まえ現状を鑑みれば、全世界的に分散した資源を統一的に扱う基盤技術が求められ、かつ、ハードウェアの観点からは普及帯にあるもので構成可能である。

ソフトウェアの観点では、研究機関単位という比較的小規模な単位では技術が確立したと言って差し支えないが、それらをまとめる技術が欠乏しているということが言える。本研究は、GRID 技術によって統合する範囲をより広範にするものである。

２．概要

２．１ GRID 環境

GRID とは，ネットワークを介して広域に散在する計算資源，情報資源，人などのリソースを仮想的に統合して利用するための基盤技術であり，Power Grid（電力網）を語源としていると言われている。 電力網は社会の基盤を形成し，利用者はそれがどのようにして作られたかを意識することなく利用している。これと同様の状況をコンピュータの利用においても実現することを目指し GRID（本稿ではこのような計算機環境を特に識別するため、GRID という表記を用いる）と呼称し，仮想的な統合計算環境、大規模データ処理環境を構築することが目標となった。GRID と電力網のアナロジーはケーブルを接続するだけで利用できるといった簡便さだけではなく、資源（電力で言うところの発電所）が地理的に分散した箇所から、様々な手段（火力・水力・原子力など）によって供給されることにもある。すなわち、GRID と既存の計算環境との違いは、以下のようにまとめる。

- 各種リソースへの簡単なアクセス。すなわち、一度 GRID へ接続すればあらゆるリソースにシームレスに接続できる。
- ヘテロな環境であること。すなわち、GRID を構成する様々な資源は，ネットワークを介して広域に分散している。
- リソース構成の動的な変化。すなわち、あらゆる資源はネットワークを介して接続されている。また、各資源は、それを管理する団体が独立して管理をおこなうため、すべての資源が独立に刻一刻と変化する。

リソース供給源の違いを吸収し、ユーザーからは一つであるかのような環境の提供を可能とするため、さまざまな GRID ミドルウェアと呼ばれるソフトウェアが開発されてきた。GRID ミドルウェアが実用段階へと入ったのは近年のことであり、これまで各研究機関が独立して開発してきたこともあり、多種多様な GRID ミドルウェアが公開されている。ここですべてを紹介することはできないため、日本原子力研究開発機構が開発を行っている GRID ミドルウェアである ITBL(Atomic Energy Grid Infra-Stracture)ミドルウェアおよび、共同研究先であり、また本研究において GRID 相互連携を行ったシュツットガルド大学ハイパフォーマンスコンピューティングセンター(Höchstleistungsrechenzentrum Stuttgart: 以下 HLRS という)が利用している UNICORE の概略を示す。

２．１．１ ITBL 環境

原子力機構では、原子力分野に求められる計算科学基盤の構築に向け、原子力グリッド基盤 ITBL の開発を行っている。図１に ITBL の概念図を示す。

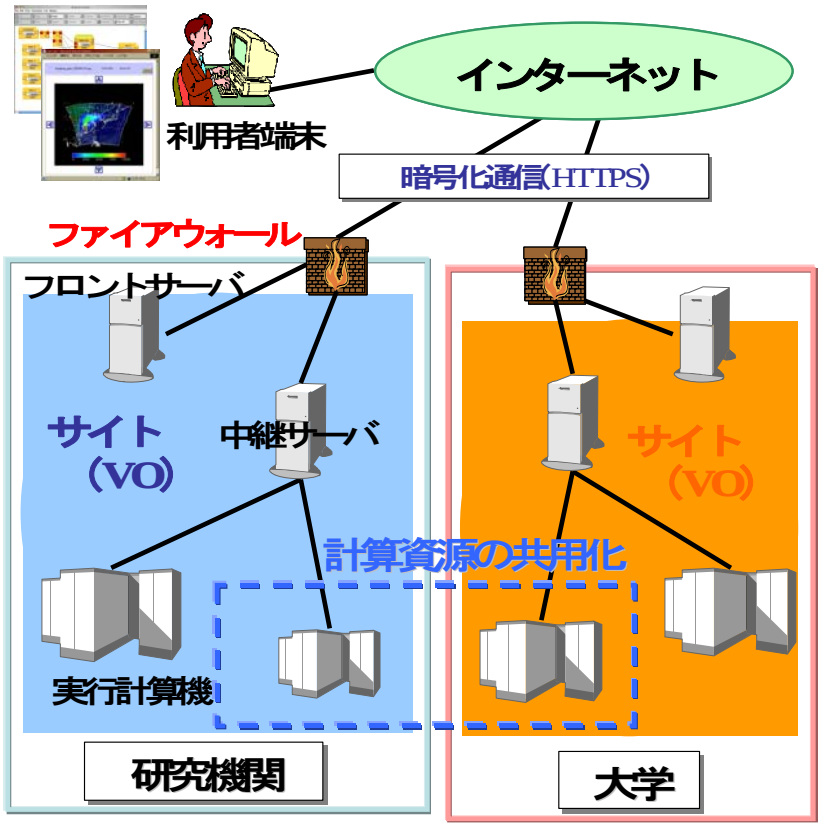

図１　ITBL のイメージ

図に示されるように、ITBL は異なる複数の計算機サイトを、暗号化された通信で、セキュアに接続し、ユーザーからは利用者端末（クライアント）を通して一つの統合化された環境として機能しているかのように振舞う。ここで、計算機サイトとは、ファイアーウォールにより外部から保護されたローカルエリアネットワークのことである。また、一般に使われている GRID の単語としては VO(Virtual Organization:仮想組織)が近い。各計算機サイトには、スーパーコンピュータのほか、ITBL を実現するためのサーバー群(フロントサーバー、中継サーバ)が設置される。ITBL では、全体を統括するサーバーが設置されず、各計算機サイトで独立に計算機群の運用ができるように設計がなされている。すなわち、ITBL においては各計算機サイトに設置されるサーバーを通してのみ他の計算機サイトへの通信が成立する。このような、分散的な処理制御系統を ITBL が採用したため、ITBL－UNICORE 相互運用を達成するにあたり、運用中の GRID 環境を全く停止する必要がなかった。詳しくは、後述する。

２．１．２ UNICORE 環境

本節では、相互運用の相手である UNICORE の紹介を行う。図２に UNICORE の概要を示す。UNICORE は ITBL と同じく利用者は暗号化されたセキュアな SSL 通信経路を通じて、計算機サイト（UNICORE では USITE とよばれる）にアクセスすることができる。ファイアーウォールを越えたゲートウェイ以降の環境がローカルエリアネットワークであることも ITBL と同様であり、自 USITE 以外とのすべての通信はゲートウェイを介して行われるのも ITBL との類似をみることができる。USITE 内部の Network Job Superviser (以下

NJS という)、Target System Interface(以下 TSI という)などは計算資源管理などを受け持つ UNICORE 特有の機能のため、ここでは詳しく触れない。詳細は、文献[４]が参考になる。

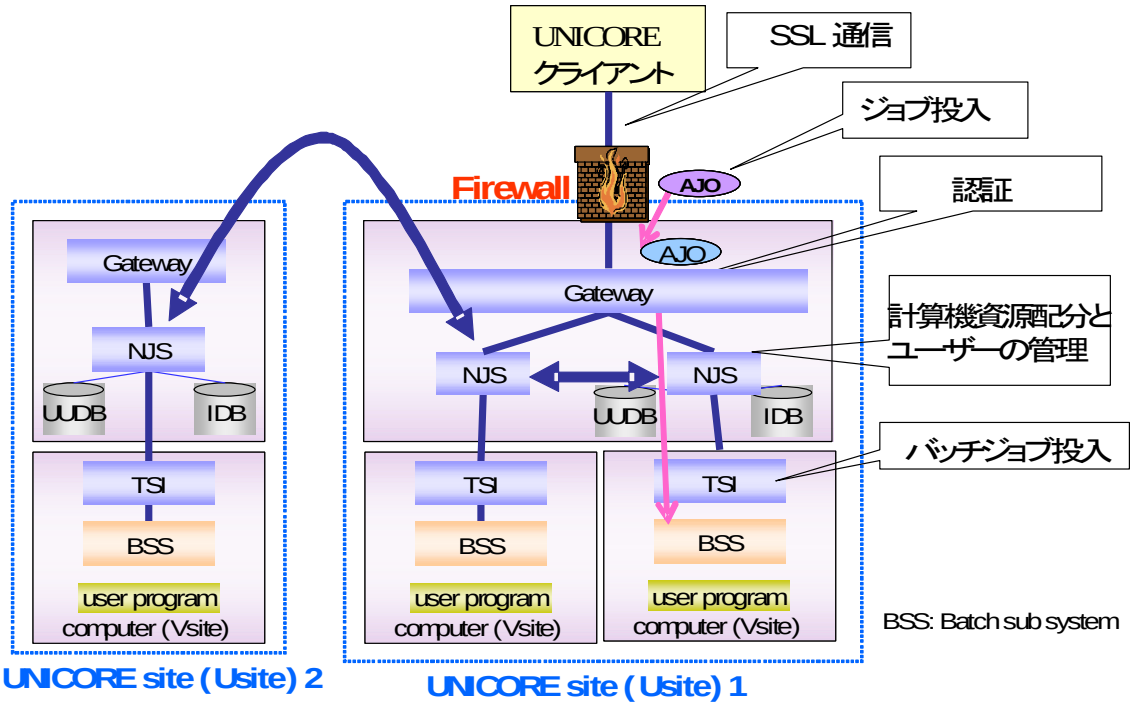

図２　UNICORE の概要

２．２ 相互連携システム

「はじめに」、で言及したとおり、GRID はそもそも世界規模での資源共有を目指して研究が進められてきた。Globus プロジェクトに端を発し、様々なプロジェクトが開かれ各々独自の GRID ミドルウェアを開発したため、初志と相反するように互換性のないミドルウェアが乱立することとなった。このような状況を打開すべく標準化が進められているが、それに対応するためにはミドルウェアの修正が必要であり、現在運用されている GRID に対しては管理者・ユーザーの双方に負担が大きい。そのため、本研究では、ITBL と UNICORE の間に立ち通訳を行う、連携サーバーの開発を行うことで、運用中の GRID を停止させることなく相互連携した。連携サーバーシステムの概略図を図３に示す。

2.1 節で触れたとおり、GRID の計算機管理単位（計算機サイト）などの点で類似点が多く見られる。本研究では試験的に ITBL から UNICORE への接続を可能とすることを目標とし、連携サーバーの実装を行った。但し、連携サーバーの実現方法を考えれば大局的な概念としては UNICORE から ITBL への逆向きの接続も容易に可能であることを強調しておく。連携サーバーの設計に際しては、以下の２点が満たされるよう留意した。

- ITBL および UNICORE に設置されている計算機について計算機の所属を意識することなく、ファイル管理、計算機資源情報の取得、計算ジョブの管理を行うことができる。
- ユーザー証明書は、手元の計算機以外のいかなる計算機にも設置されない。

前者に関して詳細に述べる。GRID ミドルウェアが満たすべき要件として挙げられるもののうち、比較的優先度が高く、プリミティブであるものを選んだ。これ以外にも複数の計算機の状況を考慮したジョブスケジューリングなど現在までに開発されている GRID ミドルウェアが提供する機能は様々あるが、ここに挙げた機能が実装されることで、順次高度化していくことが可能である。続いて、後者に関して述べる。前者で挙げた機能を持つように連携サーバーを実装することを考えた場合、実現方法の大きな方針は、連携サーバーが、接続先の GRID のユーザー端末として振舞うよう実装することである。このとき、最も単純な方法は、GRID の正式なユーザーであることを証明するユーザー証明書を連携サーバー内に配置し、適宜接続先の GRID に提出することである[５]。しかしながら、このような方法では、連携サーバーの脆弱性がそのまま両 GRID を脅かすため、セキュリティの観点からは容認できない。このため、本研究では有効期間の短い代理証明書を用いることでセキュリティを維持しつつ、相互連携機能を実現することとした。但し、2007 年８月現在はファイル管理等の実装の都合上、UNICORE のユーザー証明書を連携サーバー上に設置するという仮処置を採っている。

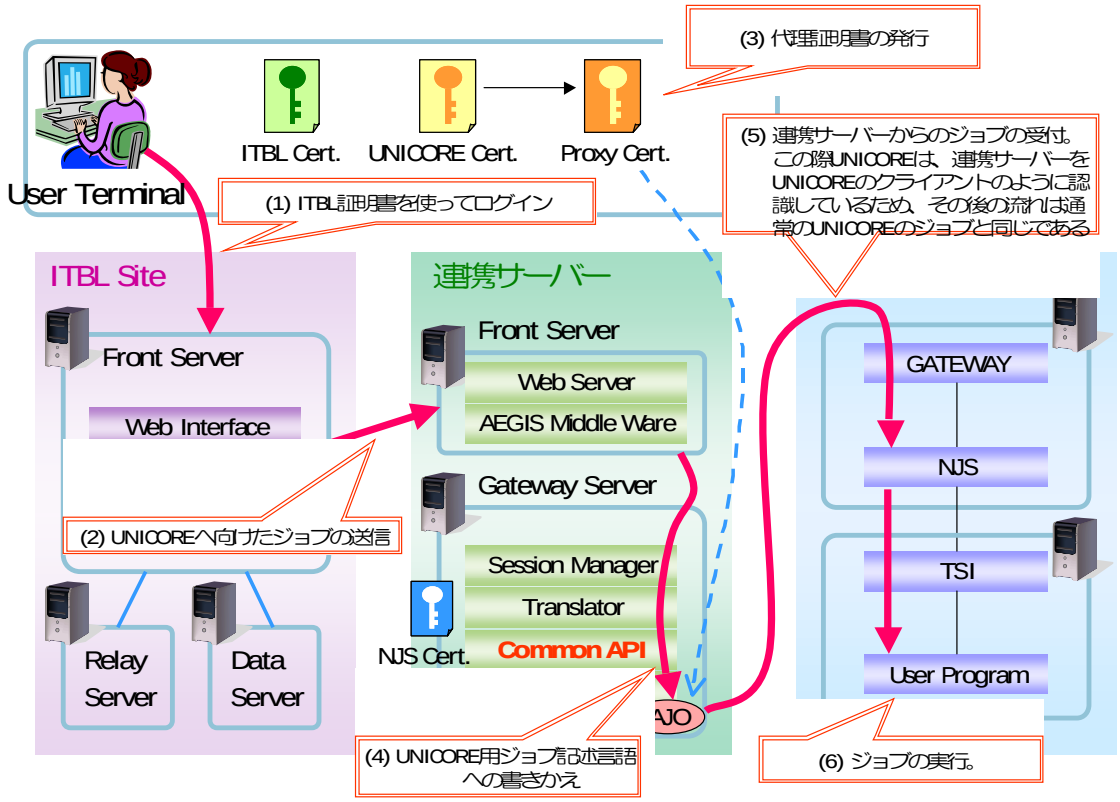

図３ 連携サーバーを介したジョブリクエストフロー

ここで、図３で示した ITBL ユーザーが UNICORE に接続された計算機を利用する際の流れの概略を示す。まず、ユーザーは自身の前にあるターミナル(一般的に Windows)から、ITBL 用のユーザー証明書を使い ITBL にログインする。このとき、UNICORE の計算機を利用するユーザーは、UNICORE 用のユーザー証明書を同時に所持している必要がある。ユーザーが ITBL から UNICORE の計算機を利用しようと試みた場合、ITBL は ITBL と

UNICORE との間に設置された連携サーバーにジョブ実行のリクエストを送信する。このとき、連携サーバーはユーザーの持つ UNICORE 用のユーザー証明書から代理証明書を発行する。連携サーバーは、ITBL から送られたジョブリクエストを、AJO と呼ばれる UNICORE のジョブリクエストに変換し、自身がもつ NJS 証明書（UNICORE のサイトであることを証明する証明書）と、代理証明書を用いて、UNICORE にジョブを依頼する。以降、UNICORE 内部では、あたかも UNICORE の通常のユーザーが利用しているかのように処理がすすみ、ジョブの終了はおおむね実行とは逆の径路をたどってユーザーに戻ることとなる。図中で Common API と表記しているものは、現在、原子力機構で開発を進めている異なる GRID に向けて開発されているアプリケーションの移植性を高めるための、共通的な Application Program Interface (API)のことである。

３．評価

本章では、実装した連携サーバーの性能を評価するために、ファイルの転送速度を指標として用いた。ファイルの転送を行う場合、すべてのデータは連携サーバを通過してゆくため、連携サーバーによるオーバーヘッドの確認には最適である。ここでは、東京上野に設置された Windows 端末（以降、本章の中では単にターミナル）、ITBL に接続された計算機（以降、本章の中では単に ITBL。東京上野に設置）および、UNICORE に設置された計算機（以降、本章の中では単に UNICORE。ドイツ　シュツットガルトに設置）の間、すなわち、ターミナルーITBL、ターミナルーUNICORE、および ITBL－UNICORE の３パターンについてファイルの転送速度を計測した。データは１バイトから 50 メガバイトまでの、９種類行った。

表１　ファイル転送に要した時間および転送速度

Data (byte)	ターミナルーUNICORE		ターミナルーITBL		ITBL-UNICORE	
	time (ms)	rate (kbps)	time (ms)	rate (kbps)	time (ms)	rate (kbps)
1	8781	0.00	1953	0.00	26062	0.00
10	13297	0.01	3188	0.03	36453	0.00
100	12485	0.06	2797	0.29	44500	0.02
1K	13313	0.62	3578	2.29	57453	0.14
10K	12672	6.46	4000	20.48	63844	1.28
100K	16750	48.91	5406	151.54	76140	10.76
1M	57891	144.90	13656	614.28	86547	96.93
10M	473719	177.08	94766	885.19	3789453	22.14
50M	2333110	179.77	484719	865.31	-	-

表１に、ターミナルーUNICORE、ターミナル―ITBL、および ITBL－UNICORE の各ケースについてファイルサイズを変化させた場合の、ファイルの転送に掛かった時間とその時の転送速度をまとめた。表中で K はキロを表し 1,000 倍することを意味するし、M はメガを表し、1,000,000 倍を意味する。また、kbps は kilo byte per second の略であり一秒あたりのデータ転送量を示している。ターミナルーITBL 間は同じ Local Area Network（以降 LAN という）に設置されていることもあり、どのデータ量に対しても最も良い転送速度を示し

ている。次に良い転送速度であったのは、ターミナルーUNICORE 間である。この場合、インターネットを介して接続されるため、ターミナルーITBL 間の転送速度に比較して性能が悪くなっている。ITBLーUNICORE 間は、ターミナルーUNICORE 間に比較して更に３倍から４倍程度遅い結果となった。物理的なネットワークを考えれば、このような性能の劣化は連携サーバーの処理に掛かるオーバーヘッドであると考えることができる。また、10 メガバイトで極端に性能の劣化が見られるが、これは連携サーバーのデータ送受信処理の問題であることが分かっている。一定量を越えるデータを送受信する場合、連携サーバーのデータ処理単位（数メガバイト）以内で収まっていれば、そのまま処理が続けられるが、処理単位を越えた場合、データを複数の処理単位に分割し送信する。この際、実装の関係上、すでに送信が行われたデータについても重複して送信している。この問題は、現在改善中である。

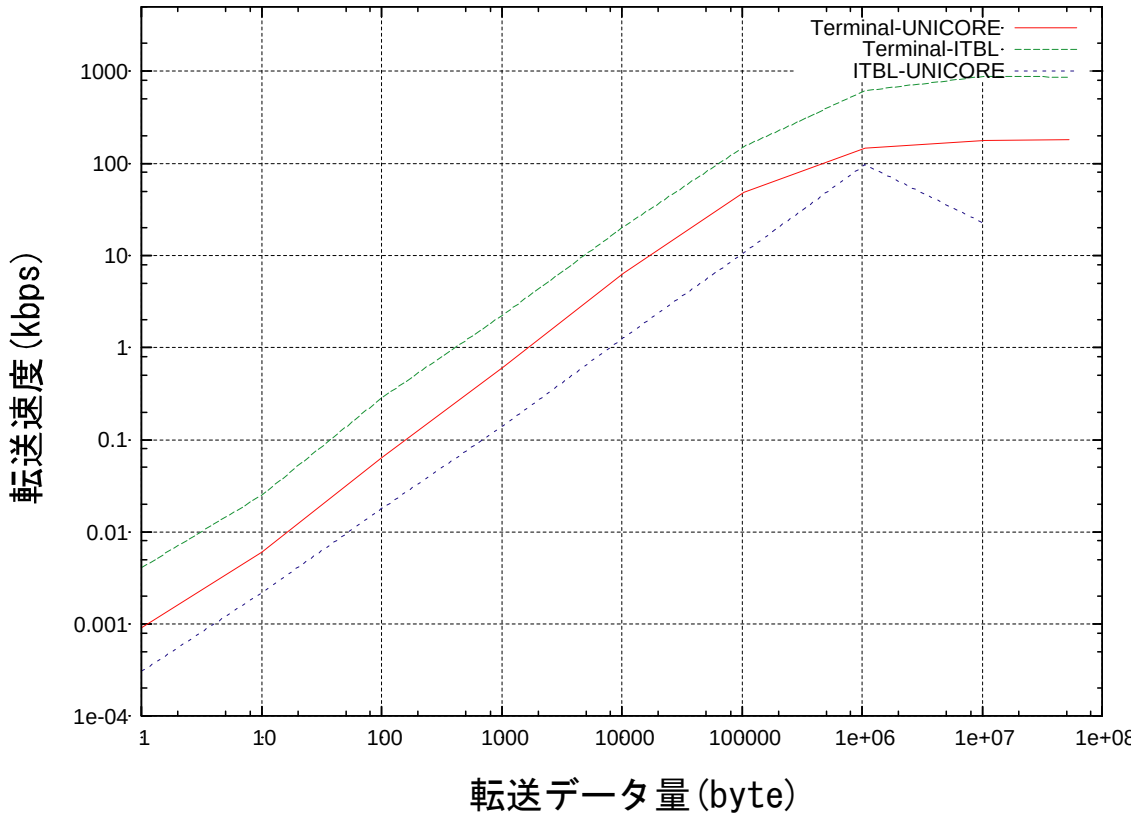

図４　データ転送量と転送速度の関係

図４にデータ転送量と転送速度の関係をプロットした。プロットは両対数プロットである。表１からも読み取れるように、ターミナルーITBL が最も高性能であり、ターミナルーUNICORE、ITBLーUNICORE の順に性能が悪く、この関係はすべての転送量について同じである。データ転送量が増えるにしたがって、転送速度が改善されることも読み取ることができる。ターミナルーITBL およびターミナルーUNICORE はデータ量が１メガバイトを越えたところで、転送速度の改善が見られなくなっており、実効的な転送速度は最大で、ターミナルーITBL の場合、約 870kbps、ターミナルーUNICORE の場合、約 180kbps とな

っている。ITBL－UNIUCORE では前述の通とおり、実装の問題があり 10 メガバイト以降性能が極端に劣化してしまうが、今回測定できた範囲内では、最大で約 100kbps の転送速度であった。物理的な転送径路を考えば、ITBL－UNICORE 間の転送速度とターミナルーUNICORE 間での転送速度を比較することによって、連携サーバーのオーバーヘッドと考えることができる。すなわち、今回計測した範囲の中では、転送速度と同程度のオーバーヘッドが確認された。また、ジョブの実行や他の命令は、極めて小規模なデータの転送であると見なすことができ（連携サーバーはデータを中継するのみであり、解釈実行は UNICORE が行う）、その場合のオーバーヘッドは大きく見積もっても 1 分程度である。

４．むすび

本稿では、日本原子力研究開発機構で開発を行っている ITBL および、UNICORE という、二つのことなる GRID の接続法について報告した。現在、異なる GRID の接続は GRID Interoperability と称され標準化する方向で進んでいる。これらの方法に対し、われわれの開発した手法である連携サーバーを用いる手法では、a) 両 GRID はいかなる改変も必要とせず、b）運用中の GRID の停止も必要としない、という二点において、優位である。実際、今回開発した連携サーバーは UNICORE に対してクライアントであるかのように振舞うため、UNICORE システムに対してはいかなる改変を求めることがなかった。また、連携サーバーを用いることによるオーバーヘッドも、最大で 1 分程度であることが確認できた。数値計算などでは、実際の処理時間が数時間以上かかることは珍しくなく、比較して軽微である。また、本稿では GRID の特徴である高いセキュリティを維持しつつ、異なる GRID を連携するための方策を提示した。必要最小限の有効期限を持つ代理証明書を用いることが、その方策の要である。本稿で示した、連携サーバーを設置し異なるシステムを接続する手法は GRID に限らず適用可能であるため、システムの停止ができない、かつ、異なるシステムを接続しなければならない場合に有効な手段となり得る。

参考文献

[１]Foster I. ; Grid Technologies & Applications: Architecture & Achievements, International Conference on Computing in High Energy and Nuclear Physics, 2001.

[２]Open Grid Forum; http://www.ogf.org

[３] Guilder G. : Telecsm: How Infinite Bandwidth Will Revolutionize Our World, Simon & Schuster, 2000.

[４]Sneling D., Berghe S. : “Grid Technology and Roadmap from UNICORE to OGSA”, Magazine FUJITSU 2004-3, vol. 55, 2004. (URL: http://img.jp.fujitsu.com/downloads/jp/jmag/vol55-2/paper08.pdf)

[５]櫛田　慶幸，鈴木　喜雄，青柳　哲男, 中島　憲宏 ： 異なるグリッド環境の相互接続法 ,平成１９年度　全NEC　C&Cシステムユーザー会　ユーザー事例論文集 掲載予定.