非対称ネットワークを隠蔽する
高速通信インフラストラクチャの設計と実装

濱野智行† 中田秀基‡ 松岡 聡†§

グリッド環境で問題となっている非対称ネットワークを扱う研究は数多く存在するが、十分な接続性とグリッド環境に適したセキュリティ・サイトポリシ非依存性・高通信性能を達成するものは存在しない。そこで、非対称ネットワークを隠蔽し、それを意識せず通信可能であり、グリッド環境に適した通信インフラストラクチャを提案する。また、そのプロトタイプ JRouter を実装し、それを用いて実際のグリッド環境でその性能評価を行った。その結果、接続性・セキュリティ・サイトポリシ非依存性において十分な性能であるが、通信性能において十分とは言えないという結論に至ったため、更なる性能向上のための施策について議論を行う。

Towards a high-performance overlay network infrastructure for Grid computing

TOMOYUKI HAMANO† , HIDEMOTO NAKADA ‡ and SATOSHI MATSUOKA †§

Many research and development are being carried out for overlay networks in the presence of firewalls and private addresses. But most of them are not suitable for Grid environment in terms of connectivity, security, independency of site policies, and most importantly, high performance. We propose a overlay network infrastructure for Grid environments that addresses these problems, and a prototype implementation JRouter. Performance results shows that the system achieves all of the requirements of a Grid environment, except for high-performance in the presence of encryption. We propose additional measures to address this problem.

1. はじめに

グリッドコンピューティングによる広域分散計算が現実的になってきており、従来の科学技術計算の分野だけでなく、ビジネスの分野でも広く認知されてきている。グリッドとは、OS やアーキテクチャが異なり、複数の管理ドメインが存在するリソース(計算機、ストレージ、実験装置など) を、複数の動的に構成される仮想組織で安全に共有するための技術である[1]。大規模な解析を必要とする高エネルギー物理学や天文学などの研究分野で利用されており、それらで生成される膨大なデータを効率良く処理可能なシステムが求められている。

その要請に応えるシステムを構築する際、複数のサイトが参加するグリッド環境では、セキュリティ

† 東京工業大学
Tokyo Institute of Technology
‡ 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
§ 国立情報学研究所
National Institute of Informatics

機能や、サイトポリシ非依存性もまた要求される。それらを満たし、グリッド上での仮想組織の設立を可能にするグリッドミドルウェアが多数開発されている。[2][3][4]

だが、現在もグリッド上のミドルウェアやアプリケーションが共通して抱えている問題の一つに、非対称ネットワークが存在する。非対称ネットワークとは、NAT/NAPT やファイアウォールなどの存在により、それらの外側から内側への接続が制限されるようなネットワーク環境を指し、この存在により各サイト間の協調が妨げられるという問題が生じている。

それを解決するための非対称ネットワークを扱う研究は多数存在するが、対応する非対称ネットワーク環境が限定的である、OS 依存である、セキュアでないなど、グリッド環境に適した非対称ネットワーク隠蔽技術は存在しないため、それを要望する声は高まっている。

そこで、我々は非対称ネットワークを隠蔽し、あたかもそれが存在しないかのように通信が可能となる通信インフラストラクチャを提案する。それはグリッド上の通信インフラとして適切なセキュリティ・サイトポリシ非依存を持ち、かつ他の研究分野からの要請である高通信性能を持ち合わせるような設計である。

また、そのプロトタイプ実装を行い、それを用いて複数サイト間で性能評価を行った。その評価結果からさらに高通信性能を向上させるために追加すべき事項の考察を行った。

2. 関連研究

本研究の対象であるネットワークの非対称性を扱う際、「非対称性を生じない技術による代替」と「非対称性を隠蔽する技術の適用」の 2 つの解決法が考えられる。前者は、ネットワークに非対称性を生じる技術に替えて、その技術の代替として十分な機能を持ちながら対称性を維持できるような方法を適用するもので、後者は、非対称性を生じる技術を用いながら、それを意識することなく対称的に通信が行える方法を適用するものである。

ここではこれらの既存の研究について言及する。

2.1 非対称性を生じない技術による代替

IPv6 を導入することで IP アドレス空間を拡大することが可能である。そのゆえ IPv6 を用いれば、割り当てられたグローバル IP 枯渇に備えたプライベート NAT 空間を一掃できるため、その意味では対称性を回復することが可能である。だが IPv6 化の必要性や、IPv4 と NAT との組み合わせにより内部ネットワークトポロジ隠蔽するポリシの存在から、全てのサイトに IPv6 を適用するのは困難である。また、ファイアウォールによる非対称性については解決法を別に用意する必要があるため、IPv6 だけでは十分な接続性を確立できない

表 1 非対称ネットワークを隠蔽する技術の比較

	接続性	ポリシ非依存	通信性能
中間ノードが通信リレー	○	○	△
NAT テーブル動的変更	○	×	○
UDP Hole Punching	△	○	△

2.2 非対称を隠蔽する技術の適用

中間ノードが通信をリレーする手法 これは、SOCKS[5],GCB[6] のように、各サイトから接続可能な中間ノードにあらかじめ接続を確立しておき、各サイトからの通信を中間ノードがリレーすることでもう一方への接続性を確立する手法である。各サイトから中間ノードへの接続は、通常の接続と何ら変わりは無いため、接続性を損なうことは無い。また、サイトポリシによって制限される可能性は小さい。だが、この中間ホストによるリレーのコストが発生するため、通信性能に悪影響が出る。

NAT テーブルを動的に変更する手法 これは、RSIP[7],DPF[6] のように、セッション確立時に内部ホストに外部アドレスを対応付け、それを NAT ルールに動的に反映させ、そのルールを用いて外部からの接続性を確立する手法である。NAT テーブルを動的に変更するために NAT ゲートウェイ上に存在するサーバにリクエストを出す際、多少のコストが発生するが、その後は通常の NAT と同様のマッピング処理を行うため、通信性能に影響はほとんど無い。また、NAT を導入しているサイトは通常 NAT 自身がファイアウォールの役割を果たしており、NAT テーブルで許可されるこの手法の接続性に問題は無い。だが、この手法では NAT テーブルにアクセス可能な権限でサーバを起動する必要があるため、それを他サイトの管理者に認めてもらう必要がある。また、NAT テーブルを勝手に変更することを認めてもらう必要もある。これは通常管理

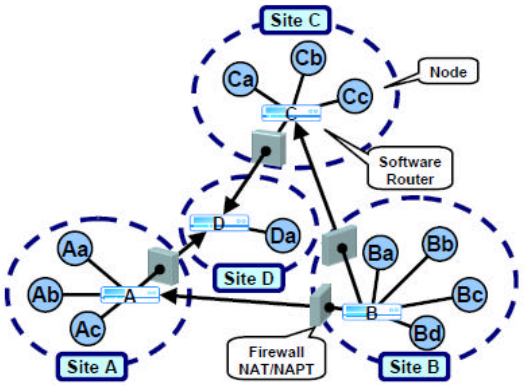

図 1　提案手法の概要

上認められることではないため、非常にサイトポリシに依存する問題である。

UDP Hole Punching　これは、TURN[8]のように、プライベート空間から定期的にUDP パケットを送信することでNAT ルールの有効期限を維持し、そのルールを用いて外部からの接続性を確立する手法である。プライベート空間から定期的に送るパケットは、通常のUDP パケットであるため、接続性を損なうことは無い。また、サイトポリシによって制限されることも無い。だが、UDP Hole Punchingで接続性を確立できない種類のNAT が存在し、また接続性を一般的に確立するには TURN の様にリレーホストを用意する必要があるため、通信性能に影響が出る。

3.　非対称ネットワークを隠蔽する高速通信インフラストラクチャの提案

グリッド環境において非対称ネットワークが問題視されているが、グリッドの要件を満たしながらネットワークの接続性を回復する機構が求められながらも、存在していない。

そこで我々はグリッドの要求を満たし、非対称ネットワークを隠蔽する通信インフラストラクチャを提案する。これをグリッドコンピューティングに適用することで、仮想組織内でネットワークの非対称性を意識しないシームレスな通信が可能になる。

表 1 に既存の非対称ネットワーク隠蔽手法の得失を示す。この比較結果を基に、我々は非対称性隠蔽手法として「中間ノードが通信をリレーする技術」を提案システムに採用する。そうすることで、この手法の持つ接続性・サイトポリシ非依存性を提案システムに備えることができる。高通信性能は別の手段を用いて達成し、さらにセキュリティ機構を追加することでグリッド環境に適したシステムにする。

提案システムの概要を図1 に示す。この図では、4つの異なる管理ポリシを持ったサイトが参加する仮想組織を想定している。システムを構成するコンポーネントは以下の 3 つである。

ソフトウェアルータ　リレーの役割を果たす中間ノード。

通信ノード　他サイトの通信ノードとデータ通信を行うノード。

障壁　ネットワークの非対称性を生み出し、一方向の接続性を失わせる。具体的にはファイアウォールとNAT/NAPT。

3.1 オーバーレイネットワークの構築

ネーミング　オーバレイネットワークに参加するノードを識別するために、ノードには一意な名前をつける必要がある。これに既存の IP アドレスを用いることはできない。なぜなら、本システムは各サイトのプライベート空間を接続するため、プライベートアドレスが競合する恐れがあるためである。本システムではソフトウェアルータ、通信ノードにそれぞれ任意の名前を付けられる。ソフトウェアルータはオーバレイネットワークのトポロジを把握しており、その情報を基に接続してくる他のソフトウェアルータや通信ノードの名前の一意性を保証する。

ルーティング　データを適切な目的地に配送するためのルーティングを、本システムではソフトウェアルータが、自身の知るネットワークトポロジ情報を隣接するソフトウェアルータに対して定期的に通知・収集することでトポロジ全体を把握し、それを基に行う。トポロジ情報はルーティングテーブルに蓄積され、新しいノードの参加が認められると参加を認めたソフトウェアルータのテーブルが即時に更新される。

3.2 セキュリティ

オーバレイネットワーク上で仮想組織をセキュアに設立するために、第三者の参加や詐称を妨げる認証・認可機構と、悪意のある参加者による盗聴を防ぐ暗号化機構が必要である。これを同時に解決するために本システムでは、PKI (Public Key Infrastructure) を用いる。

PKI を用いて以下のコンポーネント間で相互認証を行うことで第三者の参加や詐称を未然に防ぐ。

- ソフトウェアルータ―ソフトウェアルータ
- ソフトウェアルータ―通信ノード
- 送信側通信ノード―受信側通信ノード

また、通信ノード間で暗号機構を用いて暗号化・復号化することで中間のソフトウェアルータによる盗聴を防ぐことが可能となる。

3.3 サイトポリシ非依存

サイトポリシ非依存性の一つに、通常権限で動作するシステムであることが挙げられる。もしシステムが Super User 権限を強いるものであると、他サイトでの動作はその管理ポリシに依存してしまう。だが、通常権限で動作するシステムであれば、自身の判断で動作可能であるため、サイトポリシに依存しない。

サイトポリシに依存しないシステムにするために、可能な限り通常権限で動作する機構で実現する。

3.4 高速通信性能

各ソフトウェアルータはルーティングのためにネットワークトポロジ全体を把握している。そのため、最短ホップで到達可能なパスを選択することが可能であり、そうすることでリレーコストが最小に抑えられるため、高い通信性能が見込める。

4. プロトタイプ実装

提案システムのソフトウェアルータのプロトタイプである JRouter を実装した。また、通信ノードが JRouter に接続するためのソケット JRServer-Socket/JRSocket、通信に使用する入出力ストリーム JROutputStream/JRInputStream を用意し、管理クライアント JRMonitor も実装した。

これら(JRMonitor 以外) を Pure Java で実装することで、プラットフォームに依らない動作を可能にした。

以下では JRServerSocket で接続する通信ノードを JRouter Server、JRSocket で接続する通信ノードを Router Client と呼ぶ。

4.1 ソフトウェアルータ― JRouter

JRouter は他の JRouter や JRouter Server/Client との接続を管理し、必要な通信を収集したトポロジ情報を基にリレーする。各接続は 2 本の TCP コネクションを用いており、一方は制御パケット通信用、もう一方はリレーデータ通信用である。前者は接続時の登録・切断時の登録抹消・keepalive・

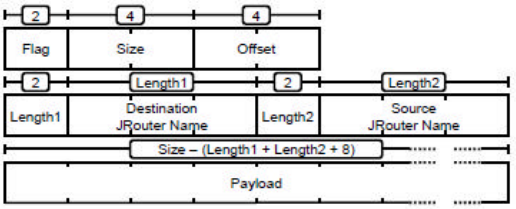

図 2 リレーデータのフォーマット

トポロジ情報送受信に用いられ、後者はリレーされるデータの送受信・認証トークンの送受信などリレーが必要になる通信に用いられる。リレーデータのフォーマットを図 2 に示す。

JRouter ではこれらの TCP コネクションを Java New I/O を用いて管理している。そうすることで単一スレッドでの動作が可能になり、スレッドコンテキストの切替コストが削減した。

セキュリティ機構には GSI(Grid Security Infrastructure)[9]を使用し、接続を受け入れると、接続元と相互認証を行う。JRouter 側ではホスト証明書と鍵を用意する必要があり、その保存位置は指定可能である。

4.2 接続用ソケット ― JRServerSocket, JR-Socket

JRServerSocket は JRSocket からの接続を待ち、接続要求を受けると、対応する JRSocket を生成する。その後、接続要求を送信した JRSocket と、生成された JRSocket とで、通信ピア間の相互認証を行う。これは、通信ピア間に存在する JRouter が認証トークンをリレーすることで実現した。相互認証に成功すると、それぞれの間で接続が確立され、通信が可能になる。

JRServerSocket と JRSocket は、JRouter 間・通信ピア間と 2 回相互認証を行うため、認証コンテキストを 2 つ用意する必要がある。そのどちらにも、ホスト証明書・ユーザ証明書のいずれかを使用可能である。

JRSocket は、通信モードを指定することができる。通信モードを指定することで、SSL 暗号化の有無を指定することが可能である。

JRServerSocket と JRSocket はそれぞれ、java.net.ServerSocket と java.net.Socket と継承こそしていないが、同様のインタフェースを持つ。そのため直感的な使用が可能である。

4.3 入出力ストリーム ― JROutputStream, JRInputStream

JROutputStream は、JRouter でのリレーの際

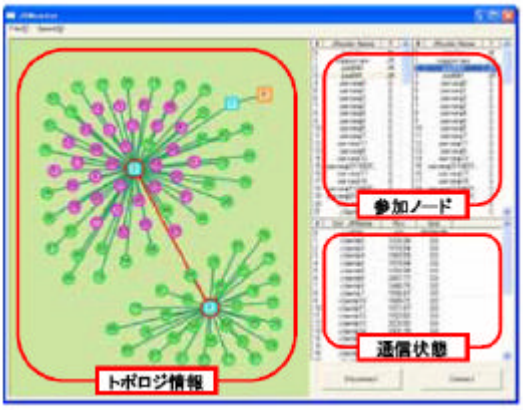

図 3　JRMonitor のスナップショット

に必要なヘッダを付加し、必要であれば SSL 暗号化を行い、出力を行う。この出力ストリームに書き込まれたデータは指定されたサイズまでバッファリングされる。

JRInputStream はそれを受信すると、ヘッダを解析・除去し、SSL 暗号化されていれば復号化を行い、アプリケーションへとデータを渡す。入力ストリームの終端は、JROutputStream が閉じられた時、終端を通知するリレーデータを送信することで検知可能にしている。JRInputStream はこのリレーデータを受信すると、入力ストリームが終端に至ったと判断する。

4.4 管理用クライアント― JRMonitor

JRMonitor はオーバレイネットワークの状況を視覚化し、把握を容易にする。また、リモートでノード間を接続・切断を行う管理機構を持つ。JRMonitor のスナップショットを図 3 に示す。

5.　評価

5.1　基礎評価

2 サイト間を JRouter で接続し、通信を行うモデルを図 4 に示す。JRouter は各サイトのゲートウェイと定めたホストに設置し、ホスト k とホスト $n+k$(k = 1, 2, …, n) に存在する JRouter Client がそれを介して通信を行う。全ての通信が JRouter 間の 1 つの接続を通して行われるため、サイト全体のスループットと JRouter 間の接続の実効バンド幅とを比較することで JRouter のリレーコストによる性能低下が見積もれる。また、通信ペア数を変化させることでスループットの飽和が CPU パワー由来か否かを判断することができる。

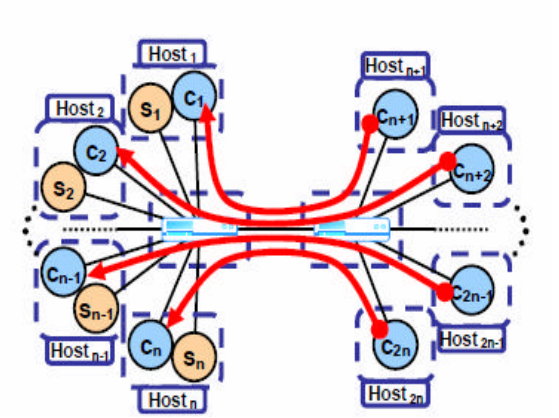

図 4　2 サイト間通信モデル

	東工大 Presto III		産総研 Koume		筑波大 Alice	
	Gateway	Others	Gateway	Others	Gateway	Others
CPU	Opteron242 x2	Opteron242 x2	Pentium III 1.4GHzx2	Pentium III 1.4GHzx2	Xeon 2.4GHzx2	Athlon 1800+x2
Mem	2GB	2GB	2GB	1GB	1GB	1.5GB
NIC	1000BASE-T	1000BASE-T	1000BASE-T	1000BASE-T	1000BASE-T	100BASE-TX
OS	Linux 2.4.27	Linux 2.4.27	Linux 2.4.20	Linux 2.4.20	Linux 2.4.20	Linux 2.4.19
	徳島大 Protein		東京電機大 sdpa			
	Gateway	Others	Gateway	Others		
CPU	Athlon MP 2000+x2	Athlon MP 2000+x2	Athlon MP 1.33GHz	Athlon MP 2400+x2		
Mem	512MB	512MB	768MB	1GB		
NIC	100BASE-TX	100BASE-TX	1000BASE-T	1000BASE-T		
OS	Linux 2.4.18	Linux 2.4.19	Linux 2.4.22	Linux 2.4.21		

図 5　評価環境

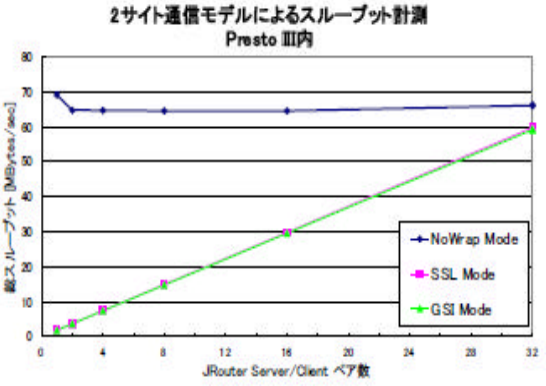

図 6　PrestoIII 内での評価

サイト全体の総スループットは、図 4 に示すように一方のサイトから送信をし続け、総スループットが定常状態になったときの値を用いた。

このモデルを以下の 3 つの条件で評価を行った。評価環境は図 5 の通りである。NoWrap Mode は SSL 暗号化なし、SSL/GSI Mode は SSL 暗号化ありのモードで、SSL Mode と GSI Mode との違いは権限委譲機能の有無で、スループットにはほとんど差は出ない。また、Bandwidth は Iperf[10]によって計

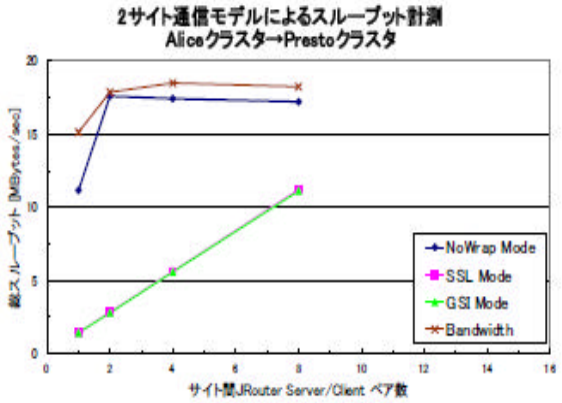

図 7　Alice - PrestoIII 間での評価

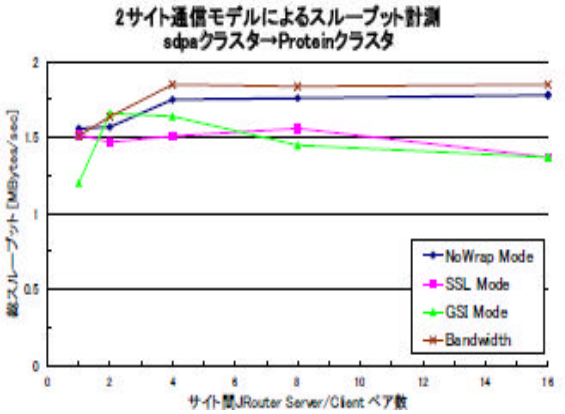

図 8　sdpa - Protein 間での評価

測した JRouter 間の実効バンド幅である。

1 サイト内での評価　Presto III クラスタ内で仮想的に 2 サイトを構築し、評価した結果を図 6 に示す。この時の JRouter 間の実効バンド幅は約 110MBytes/sec であったが、NoWrap Mode のスループットはそれに遠く及ばない値で飽和している。これはリレーコストが原因であると考えられる。SSL/GSI Mode は CPU パワーが原因で十分なスループットが出ておらず、ペア数を増加すると線形にパフォーマンスが向上する様子が見られる。

実効バンド幅が大きい 2 サイト間での評価　実効バンド幅の大きい専用線で接続された東工大-筑波大間での評価結果を図 7 に示す。NoWrap Mode のペア 1 のときは Alice クラスタ内の LAN 環境が

100BASE-TX であるために実行バンド幅を大きく下回っているが、それ以降は実行バンド幅程度の性能を示している。SSL/GSI Mode は 1 サイト内での結果と同様である。

実効バンド幅が小さい 2 サイト間での評価　実

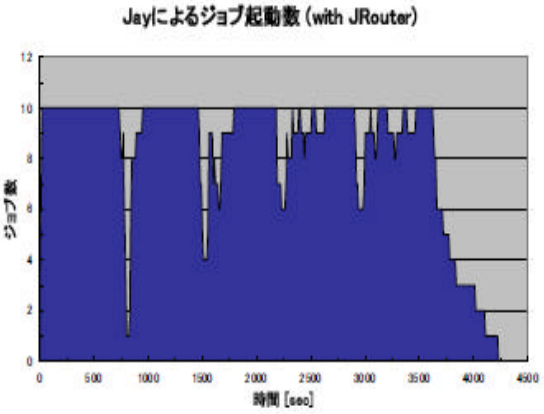

図 9　Jay を用いたジョブ実行のジョブ起動数の変化

効バンド幅の小さい WAN 環境で接続された徳島大-東京電機大間での評価結果を図 8 に示す。NoWrapMode は常に実効バンド幅同等の性能であり、SSL/GSI Mode も実効バンド幅に比べ CPU パワーが大きいため、ペア 1 からスループットが飽和している様子が見られる。

5.2　実アプリケーションを用いた評価

実際に非対称ネットワークが問題となる典型例である、ジョブスケジューリングシステム Jay[11] を用いて評価を行った。ジョブとして blast[12]を用いた。評価には Presto III と Koume を用いた。

Presto III の 1 ホストにサブミットマシンとセントラルマネージャを配置し、そこに 50 ジョブ投入した。ジョブは Presto III・Koume それぞれのサイトに配置された JRouter を介して、それぞれの実行マシンに到達し、起動される。起動されたジョブ数を時系列に示したのが図 9 である。実行マシン数は Presto III6 ホスト、Koume 4 ホストの計 10 ホストである。

この結果から実際に非対称ネットワークが問題となっているアプリケーションがシームレスに動作可能になることが確認できた。

6.　考察

評価結果から接続性・セキュリティ・サイトポリシ非依存・通信性能の観点から考察を行う。

接続性について、5 サイトの異なるプライベート空間のホスト間で通信が可能であったことから、十分な性能を持ち合わせていると考えられる。

セキュリティについて、各通信ピアが接続する JRouter と通信相手と 2 段階で相互認証を行い、ま

た通信ピア間で暗号化を行う機構が確認できた。これより十分なセキュリティ機構であると考えられる。

サイトポリシ非依存について、異なるサイトポリシを持った 5 サイトで問題なく動作し、また Super User 権限を持たないサイトでも動作することを確認できた。これより十分なサイトポリシ非依存性を持つと考えられる。

通信性能について、実効バンド幅が小さいサイト間ではリレーコストが隠されるため、十分な性能であることが確認できたが、実効バンド幅の大きい GbE などで接続された 1 サイト内ではリレーコストにより十分な性能とは言えない結果が見られた。そこで更なる通信性能向上を実現する手段について以下で述べる。

受信バッファサイズ変更 JRouter のバッファ溢れが起きないよう受信バッファを大きく取る。

送信データの圧縮 CPU パワーの余剰を利用して、送信データを圧縮する。[13]

通信プロトコルの見直し 現状は TCP を使用しているが、グリッド環境によっては他のプロトコルを使用する。

マルチパス転送 リアルタイムにスループットを計測し、それに基づき経路策定・マルチパス転送を行う。

7. まとめと今後の課題

グリッド環境に適した非対称ネットワークを隠蔽する通信インフラストラクチャの提案と、そのプロトタイプとして JRouter を実装した。また、JRouter によって構成されるオーバレイネットワークへの参加を容易にするソケット JRServerSocket/JRSocket 、入出力ストリーム JRInputStream/JROutputStream、オーバレイネットワークの管理をグラフィカルに行うことのできる管理クライアント JRMonitor も実装した。

さらに、JRouter を用いて複数サイトで性能評価を行うことで、接続性・セキュリティ・サイトポリシ非依存について十分な性能を持ち、通信性能に関して十分な性能でないことが分かった。その上で更に通信性能を向上させる施策について考察した。

今後の課題として、プロトタイプ実装への通信性能向上施策の適用、リアルタイム経路コスト計測手法の検討、UDP のサポート、オーバレイネットワークへのドメイン概念導入などが挙げられる。

参 考 文 献

[1] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. International J. Supercomputer Applications, 2001..

[2] Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. In Intl J. Supercomputer Applications, pp. 115-128, 1997.

[3] The Condor Project Homepage. http://www.cs.wisc.edu/condor/.

[4] Dietmar W. Erwin and Davind F. Snelling. Unicore – a grid computing environment. Euro-Par 2001, 2001.

[5] M. Leech, M. Ganis, Y.Lee, R.Kuris, D.Koblas, and L.Jones. Socks protocol version 5. IETF RFC1928, 1996.

[6] Sechang Son and Miron Livny. Recovering internet symmetry in distributed computing. Proceedings of the 3rd International Symposium on Cluster Computing and the Grid, 2003.

[7] Michael Borella and Gabriel Montenegro.Rsip: Address sharing with end-to-end security. Proceedings of the Special Workshop on Intelligence at the Network Edge, 2000.

[8] J. Rosenberg, R. Mahy, and C. Huitema. Traversal Using Relay NAT (TURN), 2003. http://www.jdrosen.net/papers/draft-rosenbergmidcom-turn-03.html.

[9] GSI Documentation. http://www-unix.globus.org/toolkit/docs/3.2/gsi/index.html.

[10] Iperf - The TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/Projects/Iperf/.

[11] 町田悠哉, 中田秀基, 松岡聡. ポータビリティの高いジョブスケジューリングシステムの設計と実装. 情報処理学会研究報告 2004-HPC-99 (SWoPP 2004), July 2004.

[12] NCBI BLAST. http://www.ncbi.nlm.nih.gov/BLAST/.

[13] Alexandre Denix, Olivier Aumage, Rutger Hofman, Kees Verstoep, Thilo Kielmann, and Henri E. Bal. Wide-area communication for grids: An integrated solution to connectivity, performance and security problems. HPDC-13, January 2004.