社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

グリッドコンピューティングにおける データ転送プロトコル GridFTP の性能評価

吉村　安彦† 　大崎　博之† 　今瀬　眞†

† 大阪大学 大学院情報科学研究科
〒 565-0871 大阪府吹田市山田丘 1-5
E-mail: †{y-yosimr,oosaki,imase}@ist.osaka-u.ac.jp

あらまし　グリッドコンピューティングでは、大容量のデータを効率的に転送するために、GridFTP と呼ばれるデータ転送プロトコルが用いられている。現在、GridFTP バージョン 1 (GridFTP v1)、および GridFTP v1 を拡張した GridFTP バージョン 2 (GridFTP v2) が提案されている。GridFTP v2 では、X-ブロックモードと呼ばれる転送モードにより、データストリーミング、動的資源割当、チェックサム転送などをサポートしている。しかし、GridFTP v2 の有効性は、これまで十分に明らかにされていない。そこで本稿では、数学的解析およびシミュレーションにより、GridFTP v1 および GridFTP v2 の性能を評価し、GridFTP v1 の性能限界を明らかにするとともに、GridFTP v2 の有効性を定量的に明らかにする。

キーワード　グリッドコンピューティング、データ転送プロトコル、GridFTP、X-ブロックモード、データストリーミング、動的資源割当

Performance Evaluation of Data Transfer Protocol GridFTP for Grid Computing

Yasuhiko YOSHIMURA†, Hiroyuki OHSAKI†, and Makoto IMASE†

† Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
E-mail: †{y-yosimr,oosaki,imase}@ist.osaka-u.ac.jp

Abstract　In Grid computing, a data transfer protocol called *GridFTP* has been widely used for efficiently transferring a large amount of data. Currently, two versions of GridFTP protocols, GridFTP version 1 (GridFTP v1) and GridFTP version 2 (GridFTP v2), have been proposed in the GGF. GridFTP v2 supports several advanced features such as data streaming, dynamic resource allocation, and checksum transfer, by defining a transfer mode called *X-block* mode. However, in the literature, effectiveness of GridFTP v2 has not been fully investigated. In this paper, we therefore quantitatively evaluate performance of GridFTP v1 and GridFTP v2 using mathematical analysis and simulation experiments. We reveal the performance limitation of GridFTP v1, and quantitatively show effectiveness of GridFTP v2.

Key words　Grid Computing, Data Transfer Protocol, GridFTP, X-Block Mode, Data Streaming, Dynamic Resource Allocation

1　はじめに

広域ネットワーク上で、効率的なグリッドコンピューティングを実現するためには、さまざまな問題を解決しなければならない。グリッドコンピューティングでは、分散した計算機間で転送されるデータ量が膨大であり、なおかつネットワークのラウンドトリップ時間が非常に大きい。例えば、グリッドコンピューティングを利用して、高エネルギー物理現象の解析を行う場合、単一の測定データは数 TB にも及ぶ [1]。また、広域ネットワークのラウンドトリップ時間は、場合によっては数百 ms にも及ぶこともある。

このため、広域ネットワーク上で、大容量のデータ転送を効率的に行うことができる、データ転送プロトコルが要求されている [2]。現在、インターネットで広く用いられている転送プロトコルである、FTP (File Transfer Protocol) [3] や HTTP (Hyper Text Transfer Protocol) [4] は、このような大容量のデータ転送を想定して設計されていない。FTP や HTTP の処理オーバヘッドや、トランスポート層プロトコル TCP (Transmission Control Protocol) の限界のため、FTP や HTTP など既存のデータ転送プロトコルを用いた場合、大容量のデータ転送に膨大な時間がかかってしまう [2]。

このような問題を解決するため、グリッドコンピューティン

グ上で、大容量のデータを効率的に転送するプロトコルとして、GridFTP が提案されている [5, 6]。GridFTP は、現在インターネット上で広く用いられている FTP を拡張したプロトコルである。GridFTP は、トランスポート層の通信プロトコルとして使用される TCP の問題点を解消するように設計されている。例えば、GridFTP は、GridFTP のサーバ–クライアント間で、使用する TCP ソケットバッファの大きさを交渉する機能 (TCP ソケットバッファサイズの自動交渉) や、単一のファイルを複数の TCP コネクションを用いて並列に転送する機能 (並列データ転送) をサポートしている。これらの機能により、特に帯域遅延積の大きなネットワークにおいて、GridFTP は従来の FTP よりも効率的なデータ転送が可能であることが知られている [7]。

これまで、GGF において、GridFTP バージョン 1 (GridFTP v1) の仕様策定が完了 [5] し、GridFTP バージョン 2 (GridFTP v2) の標準化が活発に行われてきた [6]。

これまで、GridFTP v1 に関しては、さまざまな性能評価が行われている。文献 [7] では、TCP の流体近似モデルを多重化することにより、GridFTP v1 を連続時間システムとしてモデル化している。これにより、GridFTP のスループットおよびパケット棄却率を導出している。さらに、GridFTP のスループットを最大化できる、最適な並列 TCP コネクション数および TCP ソケットバッファサイズを明らかにしている。また、文献 [8] では、GridFTP の並列データ転送に着目し、GridFTP の自動パラメータ設定機構を提案している。計測した GridFTP のグッドプットとラウンドトリップ時間をもとに、並列 TCP コネクション数を調整することにより、GridFTP のグッドプットが向上することを示している。しかし、GridFTP v2 の有効性に関しては、これまでに十分な検討が行われていない。

そこで本稿では、数学的解析およびシミュレーションにより、GridFTP v1 および GridFTP v2 の性能を比較し、GridFTP v1 の性能限界を明らかにするとともに、GridFTP v2 の有効性を定量的に明らかにする。まず、GridFTP サーバおよびクライアントがそれぞれ単一の場合に着目し、GridFTP v2 のデータストリーミングの有効性を数学的解析手法によって明らかにする。具体的には、GridFTP v1 を M/G/1 待ち行列として、また、GridFTP v2 を M/PS/1 待ち行列としてモデル化することにより、GridFTP における平均ファイル転送時間およびファイル転送時間の標準偏差を導出する。次に、シミュレーションにより、GridFTP サーバが複数の環境における、GridFTP v2 の動的資源割当の有効性を評価する。具体的には、GridFTP v2 において、擬似的にネットワークの帯域割当を行う制御機構を提案する。シミュレーションにより、平均ファイル転送時間を測定する。

本稿の構成は以下の通りである。まず、2 章において、GridFTP の概要を説明する。特に、GridFTP v2 に新たに導入された機能である、データストリーミングおよび動的資源割当を説明する。3 章では、数学的解析手法により、GridFTP v2 のデータストリーミングの有効性を評価する。4 章では、シミュレーションにより、GridFTP サーバが複数の環境における、GridFTP v2 の動的資源割当の有効性を評価する。最後に、5 章において、本稿のまとめと今後の課題を述べる。

2 データ転送プロトコル GridFTP

GridFTP は、グリッドコンピューティングにおいて、大容量のデータを効率的に転送するために設計されたプロトコルである [5, 6]。GridFTP は、現在インターネット上で広く用いられている FTP を拡張したプロトコルである。

これまで、GGF において、GridFTP バージョン 1 (GridFTP v1) の仕様策定が完了 [5] し、GridFTP バージョン 2 (GridFTP v2) の標準化が活発に行われてきた [6]。GridFTP v1 および GridFTP v2 は、表 1 のような機能をサポートしている。

GridFTP v1 は、既存の FTP への変更を最小限に抑えながら、効率的なデータ転送を実現している。このため、GridFTP v1 は、元となった FTP の制約から生じる、さまざまな問題点を持っている [9]。例えば、GridFTP v1 は、並列データ転送をサポートしているが、FTP のセマンティクスを継承しているため、同一のデータチャンネル上で、複数のファイルをパイプライン転送することができない。他にも、並列データ転送をサポートするように FTP を拡張したために、ファイル転送中に使用するデータチャンネル数を増減できない、といった制約も存在する。

表 1　GridFTP v1 および GridFTP v2 の機能比較
Table 1　Comparison of GridFTP v1 and GridFTP v2 features

	GridFTP v1	GridFTP v2
TCP ソケットバッファサイズの自動交渉	○	○
並列データ転送	○	○
データ転送の第三者制御	○	○
部分ファイル転送	○	○
セキュリティ	○	○
信頼性のあるデータ転送	○	○
データストリーミング	×	○
動的資源割当	×	○
GET/PUT コマンド	×	○
ストリームモードにおける EOF 通信	×	○
チェックサム転送	×	○

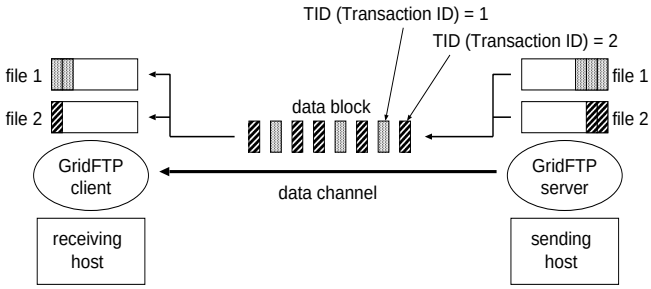

図 1　データストリーミングを用いた複数ファイルのパイプライン転送
Fig. 1　Pipelied file transfer using the data streamng

GridFTP v2 では、GridFTP v1 の拡張ブロックモード (E-ブロックモード) をさらに拡張した、X-ブロックモードを定義している [6]。これにより、同一のデータチャンネル上で複数のファイルをパイプライン転送する機能 (データストリーミング) や、ファイル転送中にデータチャンネル数 (つまり、並列データ転送に使用する TCP コネクション数) を動的に増減する機能 (動的資源割当) をサポートしている。さらに、X-ブロックモードのデータブロック単位やファイル単位で、データの破損を検出する機能 (チェックサム転送) もサポートしている。

以下では、GridFTP v2 で新たに定義された X-ブロックモードを説明する。また、X-ブロックモードによって実現される、データストリーミングおよび動的資源割当についても説明する。GridFTP v2 には、これらの機能以外にも、転送すべきファイルの指定およびデータチャンネルの確立を同時に行うことができる機能 (GET/PUT コマンド)、ストリームモードにおいてファイル転送の終了を明示的に受信側ホストへ通知する機能 (ストリームモードにおける EOF 通信)、ファイル単位もしくはブロック単位でデータの破損を検出する機能 (チェックサム転送) がサポートされている (表 1)。これらの機能の詳細については、文献 [6] を参照されたい。

GridFTP v2 では、X-ブロックモードを用いることによって、以下のような機能が実現されている。

- データストリーミング

GridFTP v2 は、X-ブロックモードのトランザクション識別子 TID (Transaction ID) を用いることにより、同一のデータチャンネル上で複数のファイルをパイプライン転送するという、データストリーミングと呼ばれる機能 [6] を実現している (図 1)。

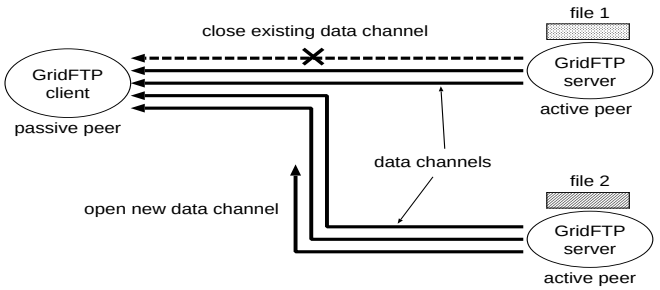

図 2　動的資源割当を用いたデータチャンネル数の動的な増減
Fig. 2　Dynamic data channel allocation using the dynamic resource allocation

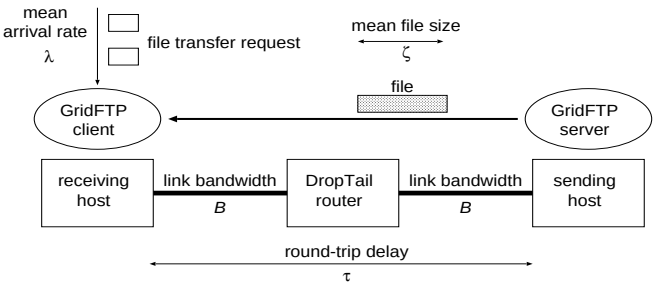

図 3　データストリーミングの解析モデル
Fig. 3　Analytic model for the data streaming

GridFTP v1 は、FTP のセマンティクスを継承しているため、同一のデータチャンネル上で、複数のファイルをパイプライン転送することができないという制限を持っていた [5, 6]。つまり、GridFTP v1 では、複数のファイルを同一のチャンネル上で同時に転送することができない。このため、あるファイルの転送が終了してから、次のファイルの転送を開始する必要がある。

一方、GridFTP v2 の X-ブロックモードでは、データブロックのヘッダにトランザクション識別子を追加することにより、GridFTP v1 の問題を解消している。GridFTP サーバ / クライアントは、異なるファイルの転送に対して、それぞれ異なるトランザクション識別子を割り当てることにより、同一のデータチャンネル上で、複数のファイルをパイプライン転送することが可能である。これにより、GridFTP v2 では、GridFTP v1 において、連続してファイルを転送する時に生じるオーバヘッドを削減することが可能となり、ファイル転送効率の向上が期待できる。

- 動的資源割当

GridFTP v2 は、READY/CLOSE/BYE コマンドを用いることにより、ファイル転送中に動的にデータチャンネル数を増減するという、動的資源割当を実現している (図 2)。

GridFTP v1 は、並列データ転送をサポートするように FTP を拡張したために、ファイル転送中に使用するデータチャンネル数を増減できないという制限を持っていた。また、データチャンネルの競合状態を防ぐために、ファイルを送信する GridFTP サーバ / クライアントが、データチャンネルを確立しなければならないという制限も持っていた [6]。

一方、GridFTP v2 では、READY/CLOSE/BYE コマンドを新たに追加することにより、ファイル転送中にデータチャンネル数を動的に増減することができる。また、ファイルを受信する GridFTP サーバ / クライアントが、データチャンネルを確立することも可能である。

ただし、データチャンネルの追加は、最初にデータチャンネルを確立した GridFTP サーバ / クライアント (アクティブピア) のみが実行することができる。逆に言えば、最初にデータチャンネルを確立しなかった GridFTP サーバ / クライアント (パッシブピア) は、データチャンネルの切断のみが可能である。

3　データストリーミングの性能評価

本章では、GridFTP サーバおよびクライアントがそれぞれ単一の場合に着目し、GridFTP v2 のデータストリーミングの有効性を数学的解析手法によって明らかにする。具体的には、GridFTP v1 および GridFTP v2 を、それぞれ待ち行列システムとしてモデル化することにより、GridFTP における平均ファイル転送時間およびファイル転送時間の標準偏差を導出する。

3.1　解　　析

解析モデルを図 3 に示す。GridFTP サーバから GridFTP クライアントへのファイル転送を考え、GridFTP サーバ–クライアント間のリンク帯域を B、ラウンドトリップ時間を τ とする。GridFTP クライアントへのファイル転送要求の発生はポアソンに従うと仮定し、その平均発生レートを λ とする。また、GridFTP サーバから GridFTP クライアントへ転送されるファイルサイズは、平均 ζ の指数分布に従うものとする。

まず、GridFTP v1 の場合を考える。GridFTP v1 では、ファイル転送要求が GridFTP クライアントに発生すると、RETR コマンドや ERET コマンドを用いて GridFTP サーバから GridFTP クライアントへ順次ファイルが転送される。ただし、GridFTP v1 はコマンドのパイプライン処理を行えないため、あるファイルの転送が終了した後、次のファイル転送が開始されるまでに、少なくともラウンドトリップ時間 τ だけの遅延が生じる。ファイル転送間の遅延を τ (つまり、ファイル転送要求があれば即座に次のファイル転送を開始できる) とすれば、GridFTP v1 におけるファイル転送は M/G/1 待ち行列によってモデル化できる。

サービス時間 $\tilde{x}$ (キューでの待ち時間を除き、単一のファイル転送に要する時間) の確率分布関数を $F_x(x)$ とし、$\mu \equiv B/\zeta$ と定義する。GridFTP サーバ–クライアント間の帯域 B が一定であると仮定すれば、$F_x(x)$ は次式で与えられる。

$$F_x(x) = 1 - e^{-\mu\,(x-\tau)} \tag{1}$$

これより、確率密度関数 $f_x(x)$ は

$$f_x(x) \equiv \frac{dF_x(x)}{dx} = \mu e^{-\mu\,(x-\tau)} \tag{2}$$

であるから、サービス時間 $\tilde{x}$ の n 次モーメント $\overline{x^n}$ は

$$\overline{x^1} \equiv \int_0^\infty x f_x(x) dx = \frac{e^{\mu\,\tau}}{\mu} \tag{3}$$

$$\overline{x^2} \equiv \int_0^\infty x^2 f_x(x) dx = \frac{2\,e^{\mu\,\tau}}{\mu^2} \tag{4}$$

となる。

$M/G/1$ 待ち行列の平均系内待ち時間 T は

$$T = \overline{x^1} + \frac{\lambda\,\overline{x^2}}{2\,(1-\lambda/\mu)} \tag{5}$$

であるから、式 (3)、(4) より、GridFTP v1 における平均ファイル転送時間 T は次式で与えられる。

$$T = \frac{e^{\mu\,\tau}\,\lambda}{(1-\lambda/\mu)\,\mu^2} + \frac{e^{\mu\,\tau}}{\mu} \tag{6}$$

一方、$M/G/1$ 待ち行列の系内待ち時間の n 次モーメント $\overline{s^n}$ は

$$\overline{s^n} = \sum_{i=0}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) \overline{w^{n-i}}\,\overline{x^i} \tag{7}$$

ただし、

$$\overline{w^1} = \frac{\lambda\,\mu\,\overline{x^2}}{2(\mu-\lambda)} \tag{8}$$

$$\overline{w^2} = \frac{\lambda\,\mu\,\left(3\,\lambda\,\mu\,\overline{x^2}^2 - 2\,\overline{x^3}(\mu-\lambda)\right)}{6\,(\mu-\lambda)^2} \tag{9}$$

である。これより、GridFTP v1 におけるファイル転送時間の標準偏差 σ は次式で与えられる。

$$\sigma \equiv \sqrt{\overline{s^2} - \overline{s^1}^2} = \sqrt{\frac{e^{\mu\,\tau}\,(2(\mu-\lambda) - e^{\mu\,\tau}\,(\mu-2\,\lambda))}{(\mu-\lambda)^2\,\mu}} \tag{10}$$

次に、GridFTP v2 の場合を考える。GridFTP v2 では、データブロックのヘッダにファイル固有のトランザクション識別子を指定することにより、複数のファイルを同時に GridFTP サーバからクライアントへ転送することが可能である。このため、GridFTP v2 におけるファイル転送は M/PS/1 待ち行列によってモデル化できる。

GridFTP サーバ–クライアント間のリンク帯域 B が一定であると仮定すれば、平均サービス時間 x (単一のファイル転送に要する時間) は次式で与えられる。

$$x = \frac{\zeta}{B} \tag{11}$$

$M/PS/1$ 待ち行列において、サービス時間が x である客の平均系内待ち時間 $T(x)$ は

$$T(x) = \frac{\mu\,x}{\mu-\lambda} \tag{12}$$

であるから、GridFTP v2 における平均ファイル転送時間 T は次式で与えられる。

$$T = \int_0^\infty T(x) f_x(x) dx \tag{13}$$

ここで、$f_x(x)$ はファイル転送時間 $\tilde{x}$ の確率密度関数である。ファイルサイズが平均 ζ の指数分布に従うことから、

$$f_x(x) = \mu e^{-\mu\,x} \tag{14}$$

となる。

一方、$M/PS/1$ 待ち行列では、サービス時間が x である客に対する、系内待ち時間の分散は次式で与えられる。

$$\sigma^2(x) = \frac{2\rho x}{\mu(1-\rho)^3} - \frac{2\rho}{1-\rho}\left[1 - e^{-\mu(1-\rho)x}\right] \tag{15}$$

これより、GridFTP v2 におけるファイル転送時間の標準偏差 σ は次式で与えられる。

$$\sigma^2 \equiv \int_0^\infty \sigma(x) f_x(x) dx \tag{16}$$

$$= \int_0^\infty \frac{1}{\mu e^{\mu\,x}} \frac{2\left\{\lambda(e^{(\lambda-\mu)x} - 1)(\lambda-\mu)^2 + \mu^2\,x\right\}}{(1-\lambda/\mu)^3} dx \tag{17}$$

3.2 数 値 例

以下では、いくつかの数値例により、GridFTP v2 のデータストリーミングの有効性を明らかにする。数値例で用いたパラメータ設定を表 2 に示す。数値例ではトラヒック負荷 $\rho(\equiv \lambda/\mu)$ を与え、ここからファイル転送要求の平均発生レート λ を決定した。以降の数値例では、特に断りのない限り、表 2 のパラメータを用いた。

まず、リンク帯域が、GridFTP v1 および GridFTP v2 の平均ファイル転送時間に与える影響に着目する。図 4 に、リンク帯域 B を 1–1000 [packet/ms] と変化させた時の、GridFTP v1 および GridFTP v2 の平均ファイル転送時間 T を示す。図 4 より、GridFTP v2 の平均ファイル転送時間が、GridFTP v1 の平均ファイル転送時間よりも小さな値となっていることが分かる。特に、リンク帯域が大きい領域において、GridFTP v1 および GridFTP v2 の平均ファイル転送時間の差が大きくなっている。これは、以下のように説明できる。リンク帯域が大きくなるにつれ、ファイル転送に要する時間が短かくなる。しかし、GridFTP v1 におけるオーバヘッド (ファイルを連続して転送する時の遅延) は、ラウンドトリップ時間によって決まる。このため、リンク帯域が大きくなるにつれ、GridFTP v1 におけるオーバヘッドが相対的に大きくなっているからと考えられる。

図 5 に、ラウンドトリップ時間 τ を 1–90 [ms] と変化させた時の、GridFTP v1 および GridFTP v2 の平均ファイル転送時間 T を示す。図 5 より、ラウンドトリップ時間 τ が大きくなるにつれて、GridFTP v1 の平均ファイル転送時間が増加していることが分かる。一方、GridFTP v2 の平均ファイル転送時間は、ラウンドトリップ時間 τ によらず一定である。このため、GridFTP v2 のデータストリーミングは、特にラウンドトリップ時間の大きな広域ネットワークにおいて有効であると言える。

最後に、図 6 に、トラヒック負荷 ρ を 0.8–1.0 と変化させた時の、GridFTP v1 および GridFTP v2 の平均ファイル転送時間 T を示す。図 6 より、トラヒック負荷 ρ が大きくなるにつれて、GridFTP v1 および GridFTP v2 の平均ファイル転送時間が急激に増加していることが分かる。特に、トラヒック負荷 ρ が大きい領域において、GridFTP v1 および GridFTP v2 の平均ファイル転送時間の差が大きくなっている。しかし、GridFTP v1 および GridFTP v2 の平均ファイル転送時間の比は、トラヒック負荷 ρ によらずほぼ一定である。このことから、GridFTP v2 のデータストリーミングは、平均ファイル転送時間を小さくする

表 2　数値例で用いたパラメータ設定
Table 2　Parameter configuration used in numerical examples

リンク帯域	B	100	[packet/ms]
ラウンドトリップ時間	τ	10	[ms]
トラヒック負荷	ρ	0.9	
平均ファイルサイズ	ζ	5000	[packet]

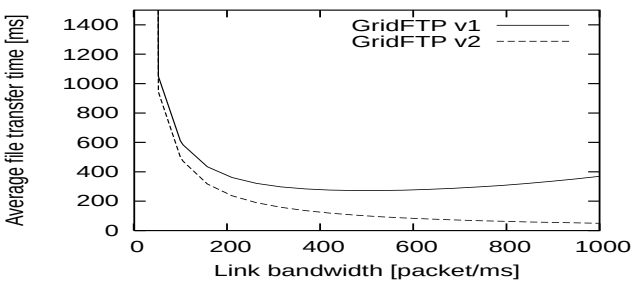

図 4　リンク帯域 B と平均ファイル転送時間 T の関係
Fig. 4　Link bandwidth B vs. average file transfer time T

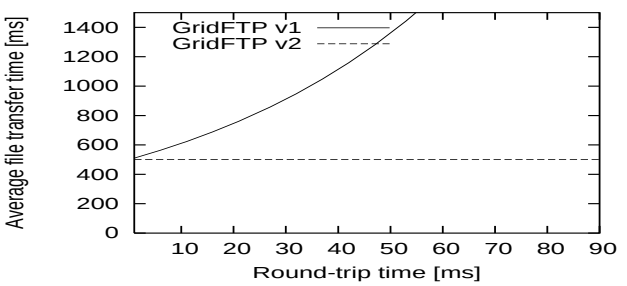

図 5　ラウンドトリップ時間 τ と平均ファイル転送時間 T の関係
Fig. 5　Round-trip time τ vs. average file transfer time T

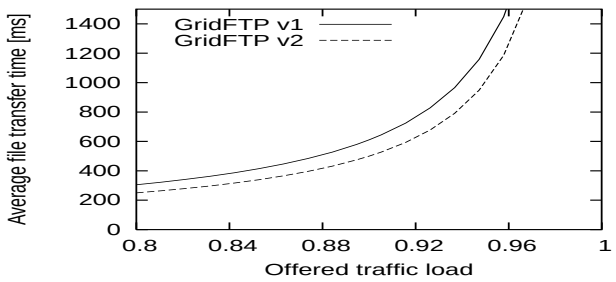

図 6　トラヒック負荷 ρ と平均ファイル転送時間 T の関係
Fig. 6　Offered traffic load ρ vs. average file transfer time T

効果があるが、トラヒック負荷に対する耐性を向上させることには効果がないことが分かる。

4　動的資源割当の性能評価

本章では、シミュレーションにより、GridFTP サーバが複数の環境における、GridFTP v2 の動的資源割当の有効性を評価する。

2 章で述べたように、GridFTP v2 の動的資源割当を用いることにより、ファイル転送中にデータチャンネル数 (並列データ転送に使用する TCP コネクション数) を動的に増減することができる。GridFTP v2 の動的資源割当それ自体は、ファイル転送中にデータチャンネル数を増減するという単純な機能であるが、これを利用することにより、帯域や CPU 資源などの割当制御を擬似的に行うことができる [6]。

本稿では、GridFTP v2 において、アクティブな GridFTP セッション数に応じて、データチャンネル数を動的に調整することにより、擬似的にネットワークの帯域割当を行う、帯域割当制御を提案する。帯域割当制御は、GridFTP クライアント上で動作する制御である。提案する帯域割当制御の基本的なアイディアは、(1) TCP ソケットバッファサイズを帯域遅延積よりも小さな値に設定し、各 TCP コネクションが利用可能な帯域を意図的に低く抑える、(2) 各 GridFTP セッションが使用するデータチャンネル数を一定時間ごとに変更することにより、擬似的に各 GridFTP セッションに帯域割当を行うというものである。

帯域割当制御の動作は以下の通りである。ネットワークの利用可能帯域を B、ラウンドトリップ時間を R、データチャンネル数の最大値を N_{max} と表記する。ネットワークの利用可能帯域 B およびラウンドトリップ時間 R は、帯域割当制御によって計測される。一方、データチャンネル数の最大値 N_{max} は、GridFTP クライアントの利用者によって設定される。まず、帯域割当制御は、TCP ソケットバッファサイズを W を以下のように設定する。

$$W \leftarrow \frac{B\,R}{N_{max}}\alpha \tag{18}$$

ここで、α は制御パラメータである。これにより、各 TCP コネクションが利用可能な帯域の最大値が、$B\,\alpha/N_{max}$ に制限される。次に、帯域割当制御は、一定時間 Δ ごとに、アクティブな GridFTP セッション数 M を計測し、GridFTP セッション i が使用するデータチャンネル数 N_i を以下のように更新する。

$$N_i \leftarrow \frac{N_{max}}{M} \tag{19}$$

これにより、各 GridFTP セッションが利用できる帯域が、$B\,\alpha/M$ によって制限されることになる。なお、各 GridFTP セッションは、ファイル転送開始時には、N_0 本のデータチャンネルを用いて並列データ転送を行う。

シミュレーションに用いたネットワークのトポロジを図 **??** に示す。単一の GridFTP クライアントが、4 台の GridFTP サーバに接続されている。このネットワークでは、2 台の DropTail ルータ間のリンクがボトルネックリンクとなる。GridFTP クライアントへのファイル転送要求は、ポアソンに従って発生させた。ファイル転送要求を受け取った GridFTP クライアントは、4 台からランダムに選択された GridFTP サーバに、ファイル転送を要求する。GridFTP サーバから、GridFTP クライアントに転送されるファイルサイズは指数分布によって与えた。

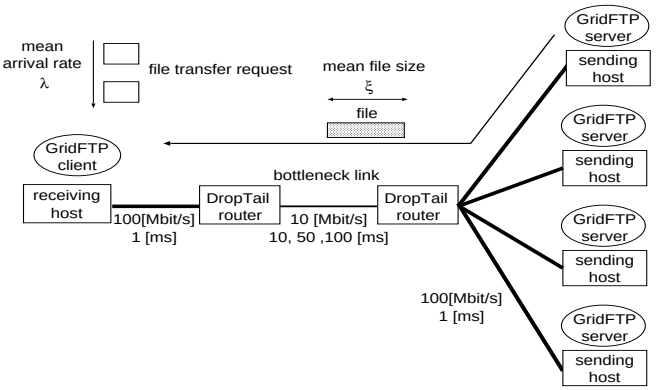

図 7　動的資源割当のシミュレーションで用いたネットワークトポロジ
Fig. 7　Network topology used in simulation for the dynamic resource allocation

表 3　動的資源割当のシミュレーションで用いたパラメータ設定
Table 3　Parameter configuration used in simulation for the dynamic resource allocation

ボトルネックリンクの帯域	10	[Mbit/s]
ボトルネックリンクの伝搬遅延	10, 50, 100	[ms]
ルータのバッファサイズ	100	[packet]
GridFTP サーバ数	4	
平均ファイルサイズ	1500	[packet]
パケットサイズ	1000	[byte]
TCP ソケットバッファサイズ	64	[packet]
ファイル転送要求の平均発生レート	1	[request/s]
データチャンネル更新間隔	1	[s]
データチャンネル数の最大値 N_{max}	50	
制御パラメータ α	1.5	

GridFTP v2 では、GridFTP クライアントは、提案する帯域割当制御によって、アクティブな GridFTP セッション数 M に応じて、各 GridFTP セッションが使用するデータチャンネル数を動的に変更させた。比較のため、GridFTP v1 のシミュレーションもあわせて実行した。GridFTP v1 のシミュレーションでは、データチャンネル数を N_0 と固定した。

シミュレーションでは、データチャンネル数の初期値 N_0 を変化させて、平均ファイル転送時間を計測した。1 回のシミュレーションでは、200 個のファイル転送要求を発生させ、101 ~ 200 個目のファイル転送に要した時間の平均を計測した。シミュレーションを繰り返して実行し、これらの値の平均および 95% 信頼区間を求めた。

なお、シミュレーションには、ns-2 (version 2.28) [10] を修正して使用した。シミュレーションで用いたパラメータ設定を、表 3 に示す。以降のシミュレーションでは、特に断りのない限り、表 3 のパラメータを用いた。

まず、図 8 に、データチャンネル数の初期値 N_0 を変化させた時の、GridFTP v1 の平均ファイル転送時間を示す。ボトルネックリンクの伝搬遅延を 10, 50, 100 [ms] に、データチャンネル数の初期値 N_0 を 1–50 と変化させてミュレーションを行った。この結果より、ボトルネックリンクの伝搬遅延が変化するにつれ、平均ファイル転送時間が最小となる、データチャンネル数の初期値 N_0 が変化していることが分かる。これはつまり、GridFTP v1 では、ボトルネックリンクの伝搬遅延が変化すると、つまり、ネットワークの状況が変化すると、それに応じて

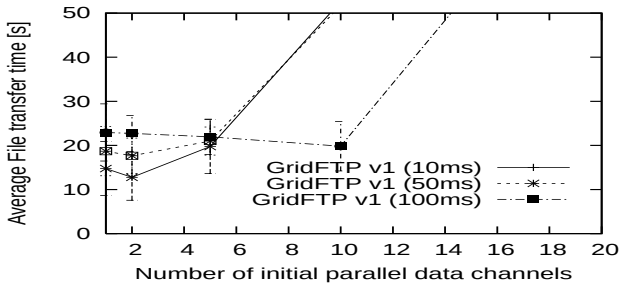

図 8　GridFTP v1 の平均ファイル転送時間 (データチャンネル数の初期値 N_0 を変化させた場合)

Fig. 8　Average file transfer time of GridFTP v1 (effect of the number of initial data channels N_0)

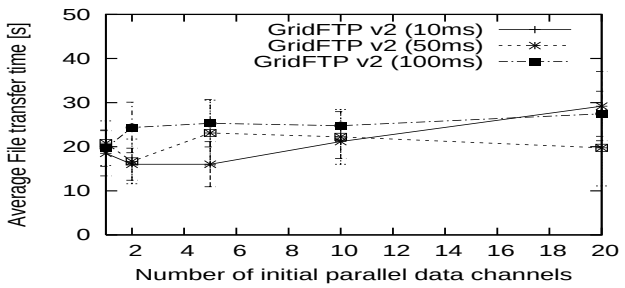

図 9　GridFTP v2 の平均ファイル転送時間 (データチャンネル数の初期値 N_0 を変化させた場合)

Fig. 9　Average file transfer time of GridFTP v2 (effect of the number of initial data channels N_0)

データチャンネル数の初期値 N_0 を、適切に設定する必要があることを意味している。

次に図 9 に、データチャンネル数の初期値 N_0 を変化させた時の、GridFTP v2 の平均ファイル転送時間を示す。ここでは、提案する帯域割当制御を使用しているが、式 (18) による TCP ソケットバッファサイズの設定は行っていない。TCP ソケットバッファサイズは、TCP コネクション 1 本当たりのネットワークの帯域遅延積よりも、十分大きな値である 64 [packet] に固定している。この結果より、GridFTP v2 では、ボトルネックリンクの伝搬遅延が変化しても、平均ファイル転送時間がほとんど変化してないことが分かる。これは、式 (19) を用いて、アクティブな GridFTP セッション数 M に応じて、データチャンネル数を調整しているため、適切な数のデータチャンネルを用いてファイルを転送できているためと考えられる。

さらに、式 (18) を用いて、TCP ソケットバッファサイズを意図的に小さな値に抑えることにより、GridFTP v2 の平均ファイル転送時間がさらに減少することを示す。図 10 に、データチャンネル数の初期値 N_0 を変化させた時の、平均ファイル転送時間を示す。ここでは、データチャンネル数の最大値を $N_{max} = 50$ とし、帯域割当制御の制御パラメータを $\alpha = 1.5$ としている。この結果より、帯域割当制御を用いない場合 (図 9) と比較して、帯域割当制御を用いることにより、平均ファイル転送時間が大幅に減少していることが分かる。紙面の都合上、結果は省略するが、帯域割当制御を用いることにより、ネットワーク中でのパケット棄却率が大幅に減少していた。つまり、帯域割当制御を用いることにより、ネットワークの輻輳がある程度解消され、その結果、平均ファイル転送時間が減少したものと考えられる。

5　まとめと今後の課題

本稿では、数学的解析およびシミュレーションにより、GridFTP v1 および GridFTP v2 の性能を比較し、GridFTP v1 の性能限界を明らかにするとともに、GridFTP v2 の有効性を定量的に明らかにした。まず、GridFTP v2 のデータストリーミングの有効性を、数学的解析手法によって明らかにした。また、GridFTP v2 の動的資源割当の有効性を、シミュレーションによって、明らかにした。

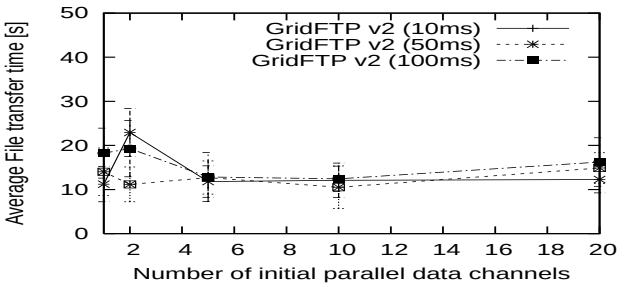

図 10　TCP ソケットバッファサイズを制限した時の GridFTP v2 の平均ファイル転送時間 (データチャンネル数の初期値 N_0 を変化させた場合)

Fig. 10　Average file transfer time of GridFTP v2 with limited TCP socket buffer size (effect of the number of initial data channels N_0)

今後の課題として、GridFTP v2 でサポートされている、他の機能の有効性を明らかにすることが挙げられる。具体的には、GET/PUT コマンド、ストリームモードにおける EOF 通信、およびチェックサム転送の有効性を評価する必要がある。

謝　　辞

本研究を実施するにあたり、 有意義な議論をしていただいた、大阪大学大学院情報科学研究科の村田正幸氏、伊藤建志氏、山本英之氏に感謝いたします。

文　　献

[1] H. B. Newman, “Data intensive Grids and networks for high energy and nuclear physics,” *InterAct Magazine*, Sept. 2002. Also available as `http://netlab.caltech.edu/FAST/references/InterAct_NetworksGridsforHEP_%hbn091502.doc`.

[2] V. Sander *et al.*, “Networking issues for Grid infrastructure,” *GGF Document Series GFD.37*, Nov. 2004. Also available as `http://www.gridforum.org/documents/GFD.37.pdf`.

[3] J. Postel and J. Reynolds, “File transfer protocol (FTP),” *Request for Comments (RFC) 959*, Oct. 1985.

[4] R. Fielding *et al.*, “Hypertext transfer protocol – HTTP/1.1,” *Request for Comments (RFC) 2616*, June 1999.

[5] W. Allcock *et al.*, “GridFTP: Protocol extensions to FTP for the Grid,” *GGF Document Series GFD.20*, Apr. 2003. Also available as `http://www.gridforum.org/GFD.20.pdf`.

[6] I. Mandrichenko, W. Allcock, and T. Perelmutov, “GridFTP v2 protocol description,” *GGF Document Series GFD.47*, May 2005. Also available as `http://www.gridforum.org/GFD.47.pdf`.

[7] 伊藤 建志, 大崎 博之, 今瀬 真, “広域グリッドコンピューティングにおけるデータ転送プロトコル GridFTP のパラメータ設定方法に関する検討,” 電子情報通信学会技術研究報告 (IN2004-39), pp. 19–24, July 2004.

[8] 伊藤 建志, 大崎 博之, 今瀬 真, “データ転送プロトコル GridFTP の自動パラメータ設定機 構の提案,” 電子情報通信学会技術研究報告 (IN2004-182), pp. 49–54, Feb. 2005.

[9] I. Mandrichenko *et al.*, “GridFTP protocol improvements,” *GGF Document Series GFD.21*, July 2003. Also available as `http://www.gridforum.org/GFD.21.pdf`.

[10] “The network simulator – ns2.” available at `http://www.isi.edu/nsnam/ns/`.