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.

広域・広帯域ネットワークのための GridFTP 自動パラメータ設定機構の実装と評価

井上　史斗† 　伊藤　建志†† 　大崎　博之†† 　今瀬　真††

† 大阪大学 基礎工学部情報科学科
〒 565-0871 大阪府吹田市山田丘 1-5
†† 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: †f-inoue@ics.es.osaka-u.ac.jp, ††{t-itou,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。GridFTP は、複数の TCP コネクションを並列に確立することによりスループットの向上を図る「並列データ転送」と呼ばれる機能をサポートしている。しかし、GridFTP が高いスループットを実現するためには、転送に利用する並列 TCP コネクション数を、ネットワーク環境に応じて適切に設定しなければならない。我々はこれまで、グリッドのミドルウエア層から計測できる情報のみを用いて、並列 TCP コネクション数を調整する機構 GridFTP-APT (GridFTP with Automatic Parallelism Tuning) を提案した。本稿では、グリッドコンピューティングの標準ミドルウェアである Globus Toolkit に含まれる GridFTP クライアント (globus-url-copy プログラム) に GridFTP-APT をどのように実装したかを説明する。また、ネットワークエミュレータを用いた実験により、現実的なネットワーク環境における、GridFTP-APT の性能を明らかにする。その結果、GridFTP-APT がさまざまなネットワーク環境で有効に動作することを示す。
キーワード　グリッドコンピューティング、GridFTP、並列 TCP コネクション、自動パラメータ設定機構、Globus Toolkit

Implementation and Evaluation of GridFTP Automatic Parallelism Tuning Mechanism for Long-Fat Networks

Fumito INOUE†, Takeshi ITO††, Hiroyuki OHSAKI††, and Makoto IMASE††

† School of Engineering Science, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
†† Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
E-mail: †f-inoue@ics.es.osaka-u.ac.jp, ††{t-itou,oosaki,imase}@ist.osaka-u.ac.jp

Abstract　GridFTP has been used as a data transfer protocol to effectively transfer a large volume of data in Grid computing. GridFTP supports a feature called *parallel data transfer* that improves throughput by establishing multiple TCP connections in parallel. However, in order to achieve high GridFTP throughput, the number of TCP connections should be optimized based on the network status. We have proposed an automatic parallelism tuning mechanism called *GridFTP-APT (GridFTP with Automatic Parallelism Tuning)* that adjusts the number of parallel TCP connections only using information measurable at Grid middleware. In this paper, we first explain our implementation of GridFTP-APT in the GridFTP client (globus-url-copy program) included in the Globus Toolkit. We then investigate the effectiveness of GridFTP-APT in realistic network environment through several experiments using our GridFTP-APT implementation and a network emulator. Consequently, we demonstrate the effectiveness of GridFTP-APT in various network environments.
Key words　Grid Computing, GridFTP, Parallel TCP Connections, APT (Automatic Parallelism Tuning) Mechanism, Globus Toolkit

1 はじめに

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のプロトコルとして、GridFTP が使用されている [1, 2]。GridFTP は、TCP の問題点を解消することを目的として設計されており、既存の FTP にさまざまな機能が追加されている。例えば GridFTP は、複数の TCP コネクションによる並列データ転送や、TCP ソケットバッファサイズの自動交渉といった機能を持つ。

GridFTP の有効性は、並列 TCP コネクション数などの制御パラメータの設定に大きく依存する、また、GridFTP が良好に動作する制御パラメータの値は、ネットワークの状況に応じて変化することが知られている [3–5]。しかし、ネットワークの状況に応じて、GridFTP の制御パラメータを適切に設定することは容易ではない。GridFTP のプロトコルの仕様には、GridFTP のサーバ–クライアント間で、並列 TCP コネクション数を指定するコマンドが含まれている。しかし並列 TCP コネクション数をどのように決定すれば良いかは、GridFTP のプロトコルでは規定されていない。

我々はこれまで、GridFTP の並列 TCP コネクション数調整機構 GridFTP-APT (GridFTP with Automatic Parallelism Tuning) を提案した [6]。GridFTP-APT は、GridFTP クライアント上で動作し、GridFTP のグッドプットを最大化するように並列 TCP コネクション数を自動的に調整する。GridFTP-APT は、GridFTP のグッドプットが、並列 TCP コネクション数に関して上に凸の関数であるという性質を利用している。GridFTP-APT は、最大化問題の数値計算アルゴリズムを用いて、最適な並列 TCP コネクション数を探索する。

本稿では、グリッドコンピューティングの標準ミドルウェアである Globus Toolkit に含まれる GridFTP クライアント (globus-url-copy プログラム) に GridFTP-APT を実装する。また、ネットワークエミュレータを用いた実験により、現実的なネットワーク環境における、GridFTP-APT の性能を明らかにする。ネットワークエミュレータを用いて、ボトルネックリンクの帯域および伝搬遅延を変化させた実験を行う。このようなネットワーク環境下で、GridFTP-APT の性能にどのような影響を与えるかを定量的に明らかにする。また、バックグラウンドトラヒックが存在する場合や、複数の GridFTP セッションが競合する場合に、GridFTP-APT がどのような特性を示すかについても調査を行う。その結果、GridFTP-APT がさまざまなネットワーク環境で有効に動作することを示す。

本稿の構成は以下の通りである。2 章では、GridFTP の並列 TCP コネクション数調整機構 GridFTP-APT の概要を説明する。3 章では、Globus Toolkit に含まれる GridFTP クライアントに、GridFTP-APT をどのように実装したのかを説明する。4. 1 章では、ネットワークエミュレータを用いた実験により、さまざまなネットワーク環境下における GridFTP-APT の性能を定量的に評価する。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 並列 TCP コネクション数調整機構 GridFTP-APT

本章では、GridFTP-APT の概要を説明する。GridFTP-APT の詳細については、文献 [6] を参照されたい。

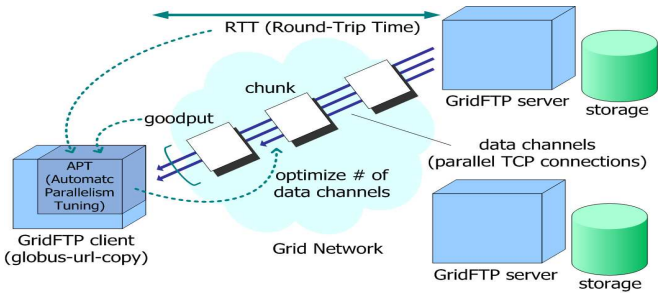

図 1: GridFTP の並列 TCP コネクション数調整機構 GridFTP-APT の概要
Fig. 1 GridFTP-APT (GridFTP with Automatic Parallelism Tuning) overview

GridFTP-APT は、GridFTP クライアント上で動作する機構である (図 1)。GridFTP-APT は、転送したいファイルを「チャンク」と呼ばれるブロックに分割して転送する。チャンク転送ごとに、GridFTP-APT は並列 TCP コネクション数を自動的に調整する。

GridFTP-APT では、最大化問題の数値計算アルゴリズムを用いて、チャンク転送ごとに GridFTP のグッドプットを計測する。その結果に応じて、GridFTP のグッドプットが最大化されるように、並列 TCP コネクション数を調整する。GridFTP-APT では、最大化問題の数値計算アルゴリズムとして、黄金分割探索法 (Golden Section Search Method) を用いる [7]。

2.1 並列 TCP コネクション数調整アルゴリズム

まず、黄金分割探索法を適用するために、GridFTP のグッドプットが上に凸となる、並列 TCP コネクション数の範囲 (以下、ブラケットと呼ぶ) を探索する。

並列 TCP コネクション数を小さな値から開始し、GridFTP のグッドプットが減少し始めるまで、各チャンク転送ごとに並列 TCP コネクション数を乗算的に増加させる。これにより、GridFTP のグッドプットが最大となる点を含む、並列 TCP コネクション数の範囲 (ブラケット) を決定する。

以下では、チャンク転送に用いる並列 TCP コネクション数を N、その時に計測された GridFTP のグッドプットを $G(N)$ と表記する。また、k 回前のチャンク転送時に用いた並列 TCP コネクション数を N_{-k} と表記する。

GridFTP-APT は、以下のようにブラケットを探索する。

(1) 並列 TCP コネクション数 N を初期化する。

$$N \leftarrow N_0 \tag{1}$$

ここで、N_0 は並列 TCP コネクション数の初期値である。

(2) チャンクを転送し、その時の GridFTP のグッドプット $G(N)$ を計測する。

(3) 以下の不等式が成立すれば、ブラケットを (N_{-2}, N_{-1}, N) と決定し、アルゴリズムを終了する。成立しなければステップ (4) へ進む。

$$G(N) < G(N_{-1}) \tag{2}$$

(4) 並列 TCP コネクション数 N を次式のように増加させ、ステップ (2) に戻る。

$$N \leftarrow \alpha \times N \tag{3}$$

ここで $\alpha(>1)$ は制御パラメータである。

次に、上記のように決定されたブラケット (l, m, r) に対して、チャンク転送中に黄金分割探索法を適用し、GridFTP のグッドプットが最大となる並列 TCP コネクション数を探索する。

GridFTP-APT は、以下のように最適な並列 TCP コネクション数 N を探索する。

(1) 並列 TCP コネクション数 N を以下のように決定する。

$$N \leftarrow \begin{cases} l + (m-l)\nu & \text{if } m-l > r-m \\ m + (r-m)\nu & \text{otherwise} \end{cases} \quad (4)$$

ここで ν は黄金分割比 $(= (3-\sqrt{5})/2)$ である。

(2) チャンクを転送し、その時の GridFTP のグッドプット $G(N)$ を計測する。

(3) 以下の不等式が成立すれば、ステップ (4) へ進む。

$$G(N) > G(m) \quad (5)$$

成立しなければ、ブラケットを以下のように変更し、ステップ (1) に戻る。

$$(l, m, r) \leftarrow \begin{cases} (l, m, N) & \text{if } m < N \\ (N, m, r) & \text{otherwise} \end{cases} \quad (6)$$

(4) ブラケットを以下のように変更し、ステップ (1) に戻る。

$$(l, m, r) \leftarrow \begin{cases} (m, N, r) & \text{if } m < N \\ (l, N, m) & \text{otherwise} \end{cases} \quad (7)$$

2.2 チャンクサイズ決定アルゴリズム

GridFTP-APT では、次回のチャンク転送時の GridFTP のグッドプットを予測することにより、チャンク転送に要する時間ができるだけ一定となるよう、チャンクサイズを動的に変更する。具体的には、GridFTP-APT は、以下のようにチャンクサイズ X を決定する。

ブラケットを探索する時は、次回のチャンク転送時の GridFTP のグッドプットを、前回および前々回の GridFTP グッドプットの比から $G(N_{-1}) \times G(N_{-1})/G(N_{-2})$ と予測し、以下のようにチャンクサイズを決定する。

$$X \leftarrow G(N_{-1}) \frac{G(N_{-1})}{G(N_{-2})} \Delta \quad (8)$$

ここで、Δ は制御パラメータであり、チャンク転送時間の目標値を意味している。

なお、最初のチャンク転送時には、前回および前々回の GridFTP のグッドプット $G(N_{-1})$、$G(N_{-2})$ が不明であるため、以下のようにチャンクサイズ X を決定する。

$$X \leftarrow \frac{N_0 W}{R} \Delta \quad (9)$$

ここで、W は、TCP ソケットバッファサイズであり、R はラウンドトリップ時間である。

また、2 回目のチャンク転送時には、前々回の GridFTP のグッドプット $G(N_{-2})$ が不明であるため、以下のようにチャンクサイズ X を決定する。

$$X \leftarrow \alpha\, G(N_{-1}) \Delta \quad (10)$$

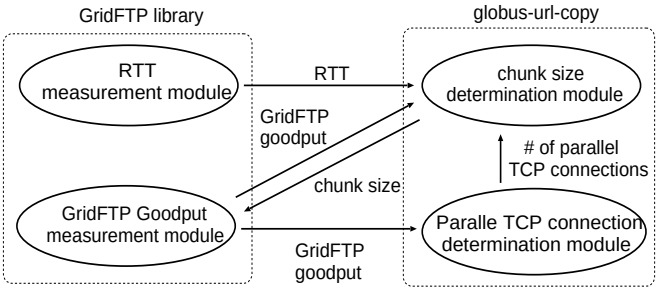

図 2: GridFTP-APT のモジュール構成
Fig. 2 Module composition of GridFTP-APT

黄金分割法を適用し、最適な並列 TCP コネクション数を探索する時は、次回のチャンク転送時の GridFTP のグッドプットを、ブラケット (l, m, r) 内の 2 点の GridFTP グッドプットの内挿によって予測し、以下のようにチャンクサイズを決定する。

$$X \leftarrow \begin{cases} ((1-\xi)\, G(l) + \xi\, G(m))\, \Delta & \text{if } N < m \\ ((1-\xi)\, G(m) + \xi\, G(r))\, \Delta & \text{otherwise} \end{cases}$$

$$\xi \leftarrow \begin{cases} \frac{N-l}{m-l} & \text{if } N < m \\ \frac{N-m}{r-m} & \text{otherwise} \end{cases} \quad (11)$$

3 GridFTP-APT の実装

3.1 概　　要

Globus Toolkit に含まれる GridFTP クライアント (globus-url-copy プログラム) を修正することにより GridFTP-APT を実装した。

globus-url-copy プログラムはコマンドラインで動作する GridFTP クライアントであり、以下のように起動される。

```
globus-url-copy [option] srcURL dstURL
```

ここで srcURL は転送元ファイルの URL、dstURL は転送先ファイルの URL である。例えば、GridFTP サーバ host のファイル /foo/bar を取得するためには、

```
globus-url-copy gsiftp://host/foo/bar \
  file:///tmp/bar
```

のように起動すればよい。

GridFTP-APT の制御パラメータ (並列 TCP コネクション数の初期値 N_0、並列 TCP コネクション数の乗算増加量 α、チャンク転送時間の目標値 Δ) は、globus-url-copy プログラムのオプションとして指定することができる。なお、GridFTP-APT の機能はデフォルトで有効になっているが、globus-url-copy プログラムに -noapt オプションを指定した場合には無効となる。

3.2 モジュール構成

GridFTP-APT を実装するために、globus-url-copy プログラムに 4 種類のモジュール (ラウンドトリップ時間測定モジュール、グッドプット計測モジュール、チャンクサイズ決定モジュール、並列 TCP コネクション数決定モジュール) を追加した (図 2)。以下、これら 4 つのモジュールを説明する。

ラウンドトリップ時間測定モジュールは、GridFTP サーバ–クライアント間の平均ラウンドトリップ時間を計測するためのモジュールである。制御チャンネル上を転送される各 GridFTP コマンドの応答時間を計測し、移動指数平均によって平均ラウンドトリップ時間を求める。

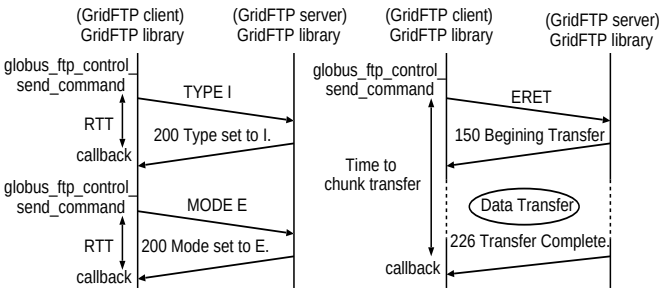

図 3: ラウンドトリップ時間および GridFTP グッドプットの計測法
Fig. 3 Measurement method of round-trip time and GridFTP goodput

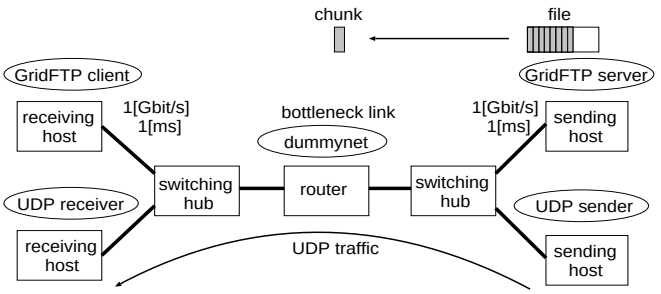

図 4: 実験に用いたネットワークトポロジ
Fig. 4 Network topology used in experiments

ラウンドトリップ時間計測モジュールは、Globus Toolkit に含まれる GridFTP ライブラリを修正することにより実装した。`globus_ftp_control_send_command` 関数は GridFTP コマンドを送信するための関数であり、GridFTP コマンドの応答受信時にコールバック関数が起動される。そこで、`globus_ftp_control_send_command` が起動された時刻と、コールバック関数が起動された時刻を gettimeofday(2) によって取得することにより、ラウンドトリップ時間を計測する (図 3)。

グッドプット測定モジュールは、各チャンク転送ごとのグッドプットを計測するためのモジュールである。データチャンネル上を転送されるチャンクの大きさと、チャンク転送に要した時間からグッドプットを計算する。

グッドプット計測モジュールは、 Globus Toolkit に含まれる GridFTP ライブラリを修正することにより実装した。GridFTP クライアントが `ERET` コマンドや `ESTO` コマンドを送信した (`globus_ftp_control_send_command` 関数が起動された) 時刻と、チャンク転送が完了した (コールバック関数が起動された) 時刻を gettimeofday(2) によって取得することにより、チャンク転送に要した時間を計測する (図 3)。

並列 TCP コネクション数決定モジュールは、GridFTP のグッドプットが最大となるように、GridFTP-APT のアルゴリズムに基づいて並列 TCP コネクション数を決定するモジュールである。具体的には、式 (1)、(3)、(4) によって、次回のチャンク転送に用いる並列 TCP コネクション数を決定する。

並列 TCP コネクション数決定モジュールは、Globus Toolkit に含まれる globus-url-copy プログラムを修正することによって実装した。並列 TCP コネクション数決定モジュールは、各チャンク転送ごとに、globus-url-copy プログラム中の `globus_l_guc_transfer_files` 関数から自動的に起動される。

チャンクサイズ決定モジュールは、各チャンク転送に要する時間ができるだけ一定となるように、次回に転送されるチャンクの大きさを決定するモジュールである。具体的には、式 (8)–(11) によって、次回のチャンクサイズを計算する。

チャンクサイズ決定モジュールは、Globus Toolkit に含まれる globus-url-copy プログラムを修正することによって実装した。チャンクサイズ決定モジュールは、各チャンク転送ごとに、globus-url-copy プログラム中の `globus_l_guc_transfer_files` 関数から自動的に起動される。

表 1 実験で用いたパラメータ設定
Table 1 Parameter configuration used in experiments

ネットワークエミュレータの帯域	1	[Gbit/s]
ネットワークエミュレータの遅延	10	[ms]
ルータのバッファサイズ	500	[packet]
ルータのキュー管理方式	DropTail	
TCP ソケットバッファサイズ	64	[Kbyte]
TCP のパケット長	1,500	[byte]
並列 TCP コネクション数の初期値 N_0	4	
並列 TCP コネクション数の乗算増加量 α	2	
チャンク転送時間の目標値 Δ	3	[s]
GridFTP で転送するファイルサイズ	50	[Gbyte]

4 性能評価

4.1 実験環境

図 4 に実験に用いたネットワークのトポロジを示す。GridFTP サーバと GridFTP クライアントが 2 台のイーサーネットスイッチおよび PC ルータを介して接続されている。ここで PC ルータは、ボトルネックリンクの帯域および遅延の変化を模擬するためのネットワークエミュレータ [8] である。すべてのリンクの帯域は 1 [Gbit/s] であるため、ネットワークエミュレータが必ずボトルネックとなる。また、バックグラウンドトラヒックとして、固定レートの UDP トラヒックを送信するための計算機も接続されている。

GridFTP サーバ、GridFTP クライアント、ネットワークエミュレータには、Xeon 3.06 [GHz]、2 [Gbyte] メモリ搭載の計算機を使用した。GridFTP サーバおよび GridFTP クライアントには Debian GNU/Linux (sarge) (Linux カーネル 2.6.8)、Globus Toolkit 4.0.3 を用いた。ネットワークエミュレータには FreeBSD 6.1、 dummynet 1.3.14.1 を用いた。実験で用いたパラメータ設定を表 1 に示す。以下の実験では、特に断りのない限り表 1 のパラメータを用いている。

4.2 GridFTP グッドプットの時間的変動

まず、GridFTP が実際のネットワーク上で有効に動作するかどうかを調査する。図 5 に、GridFTP-APT を用いた場合の GridFTP グッドプットの時間的変動を示す。比較のため、並列 TCP コネクション数を 1、4、8、16、32、64 と固定した時の GridFTP グッドプットもあわせて示している。並列 TCP コネクション数を固定した時の GridFTP グッドプットより、最適な並列 TCP コネクション数は 16 ~ 64 の間に存在することが分かる。またこの図より、40 [s] 程度で並列 TCP コネクション数が

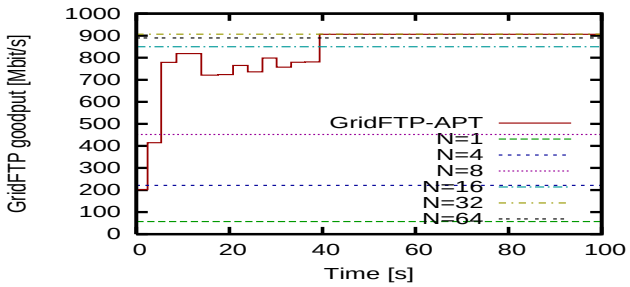

図 5: GridFTP グッドプットの時間的変動
Fig. 5　Evolution of GridFTP goodput

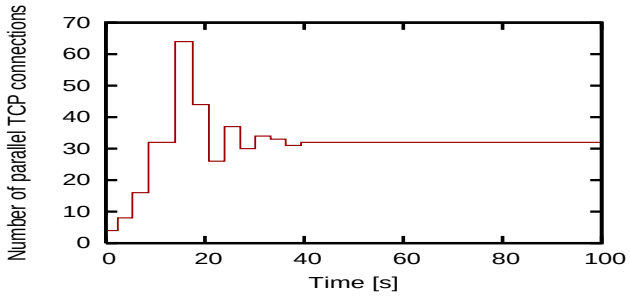

図 6: 並列 TCP コネクション数の時間的変動
Fig. 6　Evolution of the number of parallel TCP connections

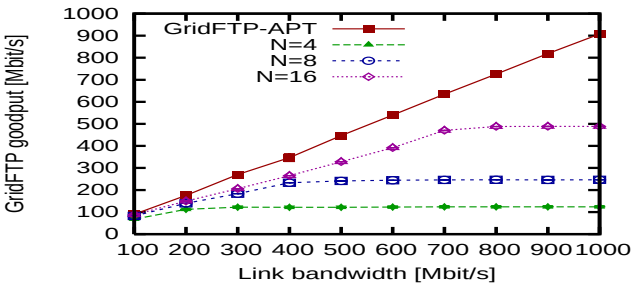

図 7: ボトルネックリンクの帯域と GridFTP グッドプットの関係
Fig. 7　Bandwidth of the bottleneck link vs. GridFTP goodput

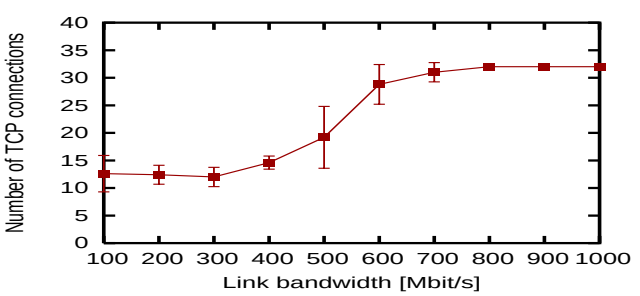

図 8: ボトルネックリンクの帯域と GridFTP-APT によって決定された並列 TCP コネクション数の関係
Fig. 8　Bandwidth of the bottleneck link vs. the optimal number of parallel TCP connections

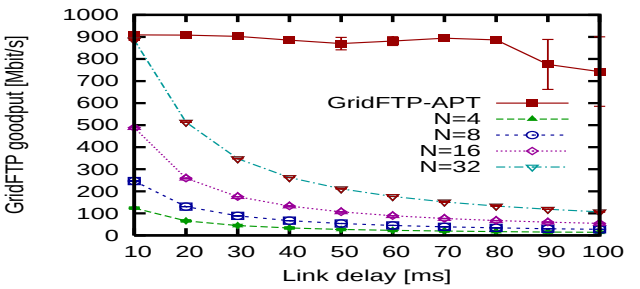

図 9: ボトルネックリンクの遅延と GridFTP グッドプットの関係
Fig. 9　Bottleneck link delay vs. GridFTP goodput

最適化され、GridFTP グッドプットが 900 [Mbit/s] に到達していることが分かる。このときの並列 TCP コネクション数の時間的変動を図 6 に示す。図 6 より、40 [s] 程度で、並列 TCP コネクション数が 32 本に収束していることが分かる。これは、最適な並列 TCP コネクション数が 16 ~ 64 の間に存在するという結果 (図 5) とも合致している。以上の考察より、GridFTP-APT は転送開始から 40 [s] 程度で、コネクション数を最適化し、ネットワークの資源を有効に利用できていることが分かる。

4.3　帯域と GridFTP グッドプットの関係

ボトルネックリンクの帯域が GridFTP-APT に与える影響を調査するために、ネットワークエミュレータの帯域を変化させて実験を行なった。図 7 および図 8 に、ネットワークエミュレータの帯域を 100 ~ 1000 [Mbit/s] に変化させた場合の、(並列 TCP コネクション数の最適化が終了した後の) GridFTP グッドプットおよび並列 TCP コネクション数を示す。5 回の実験を行ない、これらの平均および 95% 信頼区間を計測した。比較のため、並列 TCP コネクション数を 4、8、16 と固定した時の GridFTP グッドプットもあわせて示している。

図 7 より、GridFTP-APT はネットワーク資源を有効に利用できていることが分かる。また図 8 より、ネットワークの帯域が大きくなるにつれ、最適な並列 TCP コネクション数が増加していることが分かる。これは、ネットワークの帯域が大きくなるにつれ帯域遅延積が増加し、ネットワーク資源を十分に利用するために必要となる並列 TCP コネクション数が増加しているためと考えられる。

4.4　伝搬遅延と GridFTP グッドプットの関係

ボトルネックリンクの伝搬遅延が GridFTP-APT に与える影響を調査するために、ネットワークエミュレータの遅延を変化させて実験を行なった。図 9 および図 10 に、ネットワークエミュレータの遅延を 10 ~ 100 [ms] に変化させた時の、(並列 TCP コネクション数の最適化が終了した後の) GridFTP グッドプットおよび並列 TCP コネクション数を示す。比較のため、並列 TCP コネクション数を 4、8、16、32 と固定した時の GridFTP グッドプットもあわせて示している。

図 9 より、ネットワークエミュレータの遅延が 80 [ms] 以下の場合、GridFTP-APT はネットワーク資源を有効に利用できていることが分かる。ネットワークエミュレータの遅延が 90 [ms] 以上の領域では、GridFTP グッドプットが若干低下しているが、これはデフォルトの制御パラメータ Δ の値が小さいためと考えられる。ネットワークの遅延に応じて制御パラメータを設定することにより、GridFTP グッドプットが改善できると考えられる。また、図 10 より、ネットワークエミュレータの遅延にあわせて、並列 TCP コネクション数を最適化できていることが分かる。

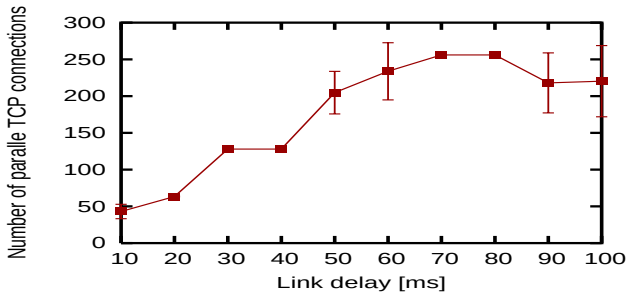

図 10: ボトルネックリンクの遅延と GridFTP-APT によって決定された並列 TCP コネクション数の関係
Fig. 10 Bottleneck link delay vs. the optimal number of parallel TCP connections

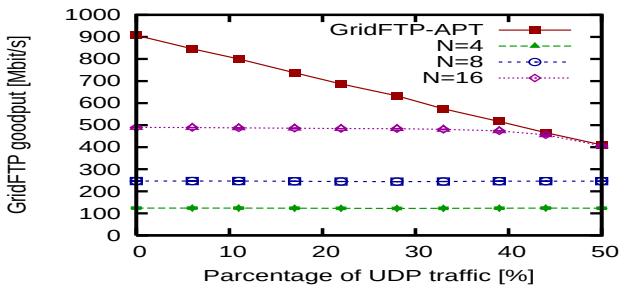

図 11: バックグラウンドトラヒックの割合と GridFTP グッドプットの関係
Fig. 11 Amount of background traffic vs. GridFTP goodput

4.5 バックグラウンドトラヒックの影響

現実のネットワークでは、さまざまなトラヒックがボトルネックリンクを共有する。このため、バックグラウンドトラヒックが GridFTP-APT の性能にどのような影響を与えるかを調査する。図 11 に、バックグラウンドトラヒックの転送レートを、ボトルネックリンク帯域の 10 ~ 50 % に変化させた時の、(並列 TCP コネクション数の最適化が完了した後の) GridFTP グッドプットを示す。比較のため、並列 TCP コネクション数を 4、8、16 と固定した時の GridFTP グッドプットもあわせて示している。この結果より、GridFTP-APT は、バックグラウンドトラヒックの転送レートによらず、ネットワークの資源を十分に利用できていることが分かる。

4.6 複数の GridFTP セッション間の公平性

ただし、バックグラウンドトラヒックとして、複数の GridFTP セッションがボトルネックリンクを共有することも考えられる。この場合、各 GridFTP セッションがボトルネックリンクの帯域を公平に共有できるかが重要となる。そこで、複数の GridFTP セッション間の公平性を調査するために、4 本の GridFTP セッションが 100 [s] ごとに転送を開始した時の、各 GridFTP セッションのスループットの時間的変動を計測した (図 12)。図 12 より、GridFTP セッション数が増加するにつれ、GridFTP セッション間の公平性が低下していることが分かる。この問題については今後さらなる検討が必要であろう。

5 まとめと今後の課題

本稿では、グリッドコンピューティングの標準ミドルウェアである Globus Toolkit に含まれる GridFTP クライアント (globus-url-copy プログラム) に GridFTP-APT をどのように実装したかを説明した。また、ネットワークエミュレータを用いた実験により、現実的なネットワーク環境における、GridFTP-APT の性能を明らかにした。その結果、GridFTP-APT がさまざまなネットワーク環境で有効に動作することを示した。

今後の課題として、GridFTP-APT のパラメータチューニングや、複数の GridFTP セッション間の公平性の改善などが挙げられる。

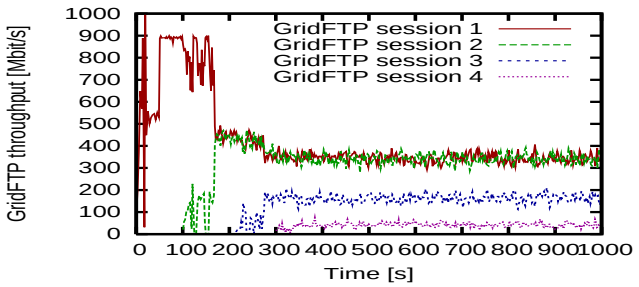

図 12: GridFTP セッションのスループットの時間的変動
Fig. 12 Evolution of throughput of four GridFTP sessions

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に感謝する。また、本研究の一部は、文部科学省における超高速コンピューター網形成プロジェクト (NAREGI) の支援を受けている。ここに記して感謝の意を表する。

文　　献

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

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

[3] T. Ito, H. Ohsaki, and M. Imase, "On parameter tuning of data transfer protocol GridFTP in wide-area Grid computing," in *Proceedings of Second International Workshop on Networks for Grid Applications (GridNets 2005)*, pp. 415–421, Oct. 2005.

[4] T. J. Hacker, B. D. Athey, and B. Noble, "The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network," in *Proceedings of the 16th IEEE-CS/ACM International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 434–443, Apr. 2002.

[5] H. Sivakumar, S. Bailey, and R. L. Grossman, "PSockets: The case for application-level network striping for data intensive applications using high speed wide area networks," in *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing*, pp. 4–10, Nov. 2000.

[6] T. Ito, H. Ohsaki, and M. Imase, "GridFTP-APT: Automatic parallelism tuning mechanism for data transfer protocol GridFTP," in *Proceedings of 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid2006)*, pp. 454–461, May 2006.

[7] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[8] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," *ACM Computer Communications Review*, vol. 27, Jan. 1997.