

# Globus における GridFTP の性能評価

河合祐貴 後藤滋樹

早稲田大学大学院 理工学研究科 情報・ネットワーク専攻

## 1 研究概要

Grid コンピューティングは広域ネットワーク上の計算、データ、実験装置、センサー、人間などの資源を仮想化・統合し、必要に応じて仮想計算機 (Virtual Computer) や仮想組織 (Virtual Organization) を動的に形成するためのシステムである。現在、Grid コンピューティングは高エネルギー物理学や宇宙科学のような科学技術分野で利用されている。こうした分野においてはテラバイトもしくはペタバイト単位の膨大な量のデータ転送をすることがある。本研究では Grid コンピューティングにおけるデータ転送技術を詳しく分析する。ここで対象とするのは、Grid コンピューティングのミドルウェア Globus のデータ転送プロトコル GridFTP である。本論文では、データ転送を行う際に意図的に遅延を発生させてスループットの変化を測定する。この測定によって GridFTP の性能を明らかにする。転送するデータ量が少ない場合には GridFTP の方が通常の FTP よりも性能が低くなる場合がある。

## 2 測定方法

本研究では、通常の FTP と GridFTP でファイル転送した際の RTT とスループットを測定し、それらを比較して考察する。GridFTP クライアントは、Globus API バンドルに含まれている globus-url-copy を使用した。早稲田大学大久保キャンパスと喜久井町キャンパスにそれぞれ 1 台ずつ Globus をインストールした計算機を用意し、この 2 地点間でファイル転送を行った。大久保キャンパス内の計算機とルータの間には、dummynet を利用できる FreeBSD をインストールした計算機をブリッジとして動作させ、意図的に遅延を発生させた。転送するファイルサイズは 1M、10M、100M、500M バイトである。それぞれのファイルに対し、GET と PUT を行った。GridFTP の並列データ転送では、ストリーム数を変えて転送を行い、ファイルサイズに対する最適なストリーム数を求め、この値を用いて測定した。RTT の測定には ping を利用してパケットを

送信し平均値を求め、スループットの測定は tcpdump を用いてパケットを収集して計算により求めた。最適ストリーム数を表 1 に示す。

表 1: 最適ストリーム数

	1M	10M	100M	500M
GET	1	2	2	16
PUT	8	64	32	64

## 3 測定結果と考察

測定結果を図 1～8 に示す。1M バイトの結果を見ると、GET、PUT 共に RTT が大きくなるとその差はほとんど見られない。RTT が小さいと、GET は FTP、PUT は GridFTP が優れた値を示した。10M バイトと 100M バイトの GET では、良く似たグラフとなっていて、わずかながら GridFTP が優っている。10M バイトの PUT では、FTP と GridFTP の差は大きく、FTP の PUT は 10M バイト以上のデータを扱った場合、低スループットに留まる。10M GET、100M PUT、500M GET、500M PUT では圧倒的に GridFTP の方が高スループットを示す。また PUT に関しては RTT が大きくなってもそれほどスループットが低下しない。FTP 全般について言えることは、RTT が 50ms を越えるあたりからスループットの向上は全く見られず横ばいとなる。GridFTP は、GET よりも PUT が、またデータサイズが大きくなるほど通常の FTP との差が大きくなることがわかった。

FTP と GridFTP のスループットを比較すると、概ね GridFTP の方が優れている。しかし、1M バイトの

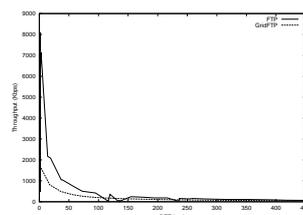


図 1: 1M GET

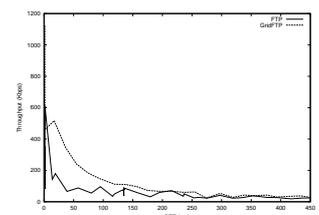


図 2: 1M PUT

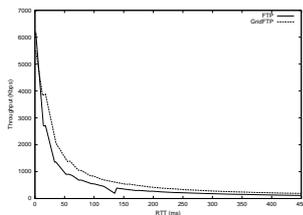


図 3: 10M GET

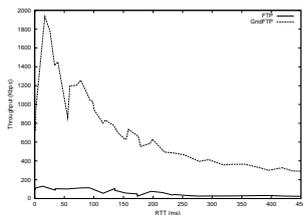


図 4: 10M PUT

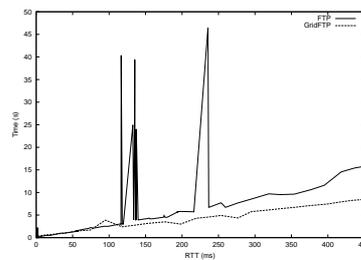


図 9: FTP のデータ転送時間と GridFTP のデータ転送が開始されるまでの時間

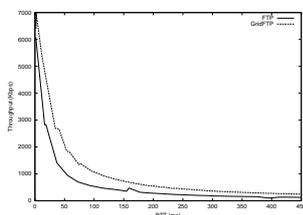


図 5: 100M GET

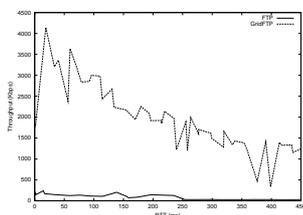


図 6: 100M PUT

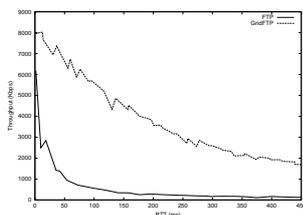


図 7: 500M GET

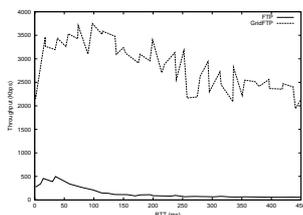


図 8: 500M PUT

ファイルを GET した時のスループットは通常の FTPの方が優れた値を示している。これは、Grid コンピューティングを実現するメカニズムがもたらすオーバーヘッドによるものと考えられる。

クライアントが globus-url-copy を実行すると、ユーザプロキシ証明書とローカル ID を含めたプロセスを生成し、リモートファイルへのアクセス要求を出す。クライアントからの要求が届くと、GridFTP サーバは GSI と呼ばれるセキュリティ機構によって、クライアントから渡されたユーザプロキシ証明書を元に権限を確認し、ローカル ID との対応付けを行いファイルへのアクセス許可を確認し、実際にデータ転送を開始する。プロセスを実行するまでのこうした一連の動作がオーバーヘッドとなり、1M バイトのファイルの GET では、GridFTP の能力が発揮できなかったものと考えられる。これを確認するために、クライアントがジョブの要求を出してから実際にデータ転送が開始されるまでに要した時間を tcpdump でパケットを解析することにより割り出した。これを FTP のデータ転送時間と重ね合わせてみると図 9 に示す結果を得た。

RTT が小さな値の時は、FTP のデータ転送時間を示す線と GridFTP 特有の処理にかかる時間を表す線が、ほぼ重なりあう。つまり、GridFTP でクライアントがユーザプロキシ証明書と共にアクセス要求を出し、

GSI によって認証している間に、複雑な仕組みのない FTP はデータ転送を完了してしまう。RTT が大きい場合でも、ファイルのサイズが小さい時には、FTP のデータ転送時間に占める GridFTP のデータ転送が開始されるまでの時間の割合は大きい。このように、Grid のセキュリティを実現するための動作がオーバーヘッドとなり 1M バイトのデータの GET では FTP の方が優れた値を示す。

## 4 まとめ

本研究は、GridFTP は通常の FTP と比べて高スループットを得ることができるが、扱うデータサイズが小さい時は Grid 特有の処理がオーバーヘッドとなり優れたパフォーマンスを得ることができないということを明らかにした。Grid が現在利用されているのは主に科学技術分野であると述べたが、この場合、扱うデータサイズは比較的大きい。これらオーバーヘッドがとりわけ問題になることはないと思われる。しかし、Grid コンピューティングは今もなお進化し続け、今後我々の生活にますます身近なものになると考えられる。そうなるに粒度が小さい計算を多数実行したり、比較的小さいサイズの小さなデータを複数の計算機に同時に転送するといった機会が増えてくると予想される。そのような場合には、本実験で明らかにしたオーバーヘッドが問題となる。この点は今後の展開において考慮すべき問題である。

## 参考文献

- [1] The Globus Alliance. <http://www.globus.org/>.
- [2] Foster, I. and Kesselman, C.: “The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann, 1999.
- [3] Grid Datafarm. <http://datafarm.apgrid.org/>.