

WEB ベースのグリッドミドルウェア『Friendly Grid』の開発 —グリッド計算をより身近に—

小矢 英毅[†] 董 芳艶[†] 廣田 薫[†]

東京工業大学 大学院総合理工学研究科[†]

1. はじめに

インターネットをはじめとする広域ネットワークの進歩により、広域ネットワーク上のデータ共有や、並列分散処理を可能にするグリッド技術が注目されている。既存のグリッド環境には、GridRPC システム [1] の Ninf-G [2] や OmniRPC [3] などが存在する。これらのシステムを用いた並列分散処理の実現には、多数の計算機の確保と各計算機をリソースとして機能させるためのソフトウェアのインストールや設定を必要とする。このようなグリッド環境構築作業は、並列分散処理を実現する開発者にとっての妨げとなる。

本研究では、ネットワークに接続された不特定多数のパーソナルコンピュータ (PC) をリソースとして利用可能かつグリッド環境の構築作業を必要としない Web ベースのグリッド計算用ミドルウェア Friendly Grid を提案する。そして、モンテカルロ法による円周率算出問題を用いて Friendly Grid を性能評価する。

2. Friendly Grid の構成

Friendly Grid は Web ベースのミドルウェアとして設計され、Web ブラウザ上で動作する Friendly Grid Server (FGS) と、オペレーティングシステム上で動作する Friendly Grid Client (FGC) の 2 つで構成される。

2.1. Friendly Grid Server (FGS)

FGS は Web ブラウザとブラウザ用のプラグインである Silverlight を基盤とし、IronPython の実行環境と TCP/IP による通信環境を備える。これらの環境を利用し、リモートから Web ブラウザ上で Python のプログラムを実行する Friendly Grid Remote Procedure Call (FGridRPC) と、プログラムや演算に必要なファ

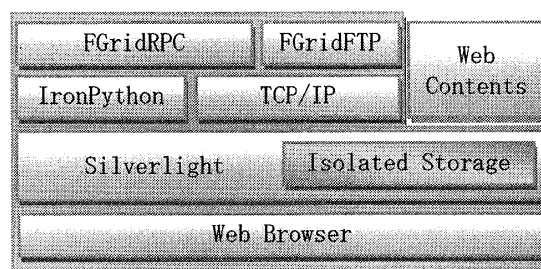


図1. Friendly Grid Server の構成

イルを FGS に転送する Friendly Grid File Transfer Protocol (FGridFTP) を実装する。FGS の構成を図1に示す。

2.2. Friendly Grid Client (FGC)

FGC は FGS が提供している FGridFTP および FGridRPC を用いて、全てのリソースに対するプログラムの配置や、演算に必要なファイルのアップロード、リモートからのプログラム呼び出しなどリソースを管理する役割を担う。

3. Friendly Grid を用いたグリッド環境の構築

Friendly Grid を用いて並列分散処理を実装する開発者は、Web ページを1つと、ユニークな IP アドレスまたはドメイン名を持つクライアント PC を1つ必要とする。そして、Web ページの一部に FGS を埋め込んで公開し、クライアント PC 上で FGC を実行させる。Friendly Grid が構築するグリッド環境を表したのが図2である。ネットワークに接続された PC を分散処理のためのリソースとするには、Web ブラウザで該当 Web ページにアクセスするだけでよく (図2①)、専用ソフトウェアのインストールや設定は必要としない。Web ページにアクセスすると、Web ページの HTML ドキュメントや画像データと同様に FGS がダウンロードされ Web ブラウザ上で実行される (図2②)。そして、クライアント PC 上の FGC にリソースとして登録され (図2③)、その後は FGC からの FGridRPC や FGridFTP の要求に応じて処理を実行する (図2④)。

Development of web-based grid middleware "Friendly Grid"
—making grid computing more friendly—

[†]Hidetaka Koya, Fangyan DONG, Kaoru Hirota:
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology

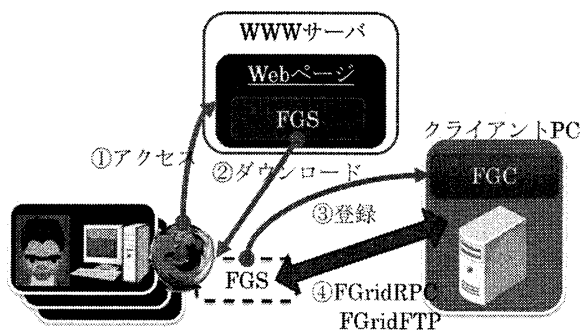


図 2. Friendly Grid が構築するグリッド環境

Friendly Grid の特徴のひとつとして既存の Web 技術や Web コンテンツとの連携が挙げられる。Web ページをインターネット上で公開すれば 不特定多数の Web コンテンツ利用者から協力を得られ、Web ページを認証制にすれば協力するユーザを制限することもできる。例えば、既存の動画配信等のコンテンツと組み合わせて、Web ブラウザで動画を視聴中にバックグラウンドで Friendly Grid が分散処理を実行するという事も可能になる。

4. Friendly Grid の基本性能の評価

Friendly Grid の性能評価を目的とし、モンテカルロ法による円周率算出を Friendly Grid を用いて行った。

4.1. 実験環境

WWW サーバ上に実験用の Web ページを立ち上げ、Web ページ上に FGS を配置し、LAN 内の 8 台の PC で実験用 Web ページにアクセスすることで、FGS を起動する。8 台の PC はそれぞれ性能が異なり、利用する Web ブラウザの種類も異なる。そして、別の 1 台の PC 上で FGC を起動し、モンテカルロ法による円周率算出を分散処理させ、リソース数に対する演算時間と、リソース数 1 での演算時間に対する処理速度向上率を測定した。

4.2. 実験結果・考察

リソース数に対する演算時間の測定結果が図 3 である。リソース数 1 のとき 39.8 秒で、リソース数 8 のとき 8.2 秒であった。リソース数に対する処理速度向上率の測定結果が図 4 である。リソース数 8 のとき処理速度向上率は 4.8 倍であった。完全な分散化が可能であるモンテカルロ法による円周率算出問題で処理速度向上率が 4.8 倍となってしまったのは、タスクの分配作業や通信によるオーバーヘッドが大きい。原因としては、Friendly Grid のシステムが未完成であることも挙げられるが、統一的な実験環境を準備できなかったため、一部の低スペック PC がボトルネックになっている可能性もある。

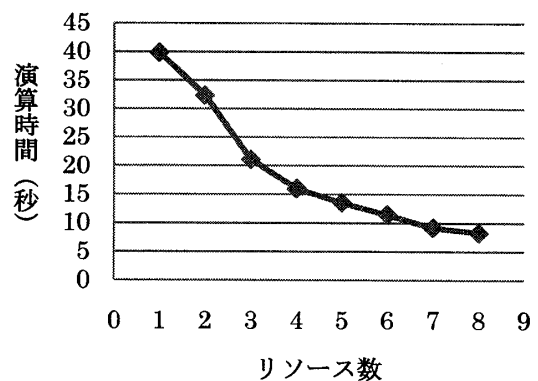


図 3. リソース数に対する演算時間 (秒)

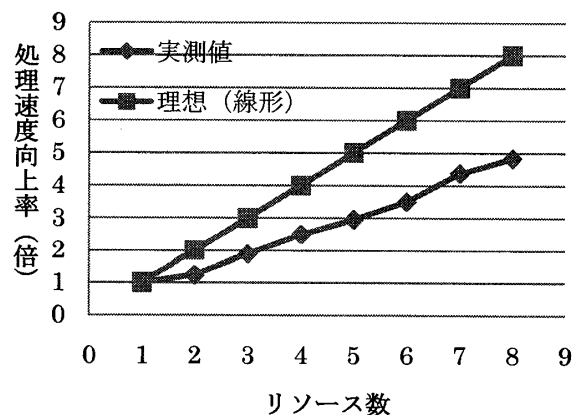


図 4. リソース数に対する処理速度向上率

5. おわりに

本研究では Web ベースのグリッド計算用ミドルウェア Friendly Grid を提案した。そして、モンテカルロ法による円周率算出問題を用いて Friendly Grid による分散処理実験を行った。実験の結果、リソース数 8 に対する処理速度向上率 4.8 倍となることを確認した。しかしながら、得られた処理性能向上率は十分なものではないため、今後は Friendly Grid の完成・改善はもちろんのこと、PC の性能による負荷分散が可能なシステムの構築が必要である。

参考文献

- [1] 谷村勇輔, 中田秀基, 田中良夫, 関口智嗣: GridRPC システムの比較—アプリケーション開発における違い, 情報処理学会研究報告 2004-HPC-101, pp. 91-96, 2005.
- [2] 中田秀基, 朝生正人, 谷村勇輔, 田中良夫, 関口智嗣: グリッド RPC システム Ninf-G の可搬性および適応性の改善, 情報処理学会研究報告 2007-HPC-112, pp. 37-42, 2007.
- [3] 佐藤三久, 朴泰祐, 高橋大介: OmniRPC: グリッド環境での並列プログラミングのための Grid RPC システム, 情報処理学会誌 コンピューティングシステム, Vol. 44, No. SIG11, pp. 34-45, 2003.