
新しい NetSolve Farming の実装
下坂久司

1 isdl_netsl_farm の作成

1.1 NetSolve System

NetSolve システム¹⁾ は, Jack Dongarra 等によって開発された RPC ベースの Grid ミドルウェアである。NetSolve システムはネットワークを介して, 遠隔にある科学技術計算用ライブラリを利用するためのクライアント・サーバモデルに基づくシステムであり, クライアントは NetSolve システムの提供する API を利用することで, ネットワーク上にあるハードウェア, ソフトウェア (科学技術用計算ライブラリ) の両方の計算資源に簡単にリモートアクセスを行うことができる。

1.2 NetSolve Farming

NetSolve システムでは, 同じ種類の連続した複数の RPC 要求を行いたい場合, Farming という機能を利用することができる。Farming は, 内部でノンブロッキング通信の RPC を利用し, 連続して RPC 要求を実行する。この機能を利用することで, クライアントは複雑なノンブロッキング通信の RPC を連続して行うプログラムを作成する必要なく, 連続した RPC 要求を行うことができる。

1.3 NetSolve Farming の問題点

NetSolve Farming は非常に便利な機能であるが, 現在までに実験によって次のような問題点が確認されている。

1. 大量の RPC 要求を繰り返し Farming で実行した場合, CLOSE_WAIT 状態のポートが大量に発生し, 処理が中断する。
2. Farming のオーバーヘッド時間は, 通常の RPC の場合と比較し長い。これは, 解析が比較的長いものであれば問題にならないが, 短い解析の場合, 実行時間に大きく影響すると考えられる。

特に一つ目の問題点は, Farming がノンブロッキング通信の RPC を利用していることが理由となっている。ブロッキング通信とノンブロッキング通信の RPC の違いは, 処理要求の送信と結果の受信の間に, 受信すべきデータが相手から送られているかどうかの受信状態のチェックを行うか, 行わないかである。CLOSE_WAIT のポートが発生するのは, この受信状態をチェックする部分である。

1.4 isdl_netsl_farm

NetSolve Farming の問題点を解決するため, Farming を NetSolve とは異なる方法で実装した。新たな Farming は次のようなものである。

1. 名前は isdl_netsl_farm とする。
2. NetSolve の Farmin と同じ引数 (API) をとる。これは既存の Farming を利用するプログラムからの移行をスムーズに行うためである。
3. ノンブロッキングではなくブロッキング通信による RPC を利用する。
4. ブロッキング通信による RPC では, 1 プロセス (スレッド) につき, 一つの RPC しか同時に実行できない。そのため, 複数のスレッドを立ち上げ, それぞれのスレッドで RPC 要求を行う。
5. 全体として, 各 RPC は一定時間の間隔を空け, 連続して RPC 要求を行う。

特に 5 つ目の特徴に関して, NetSolve Farming では 1 秒程度の間隔で RPC 要求を連続して処理している。サーバでの計算時間が短く, 大量の RPC 要求を連続して実行する場合, TIME_WAIT 状態のポートが増えるという問題がある。isdl_netsl_farm では RPC の間隔を 0.5 秒程度にしている。これは, ポートの制限を超えず, かつ Farming のオーバーヘッドを縮小できる間隔である。

2 今後の課題

- SC のポスターの作成
- SC のデモの作成
- Cambria の再構築
- 制約条件外の個体の引き戻し手法の改良
- MPS シンポジウムのポスター講演の論文原稿作成
- 理工研の執筆

参考文献

- 1) H. Casanova and J. Dongarra, NetSolve : A Network Server for Solving Computational Science Problems, *Proc. of Supercomputing '96 Conference*, 1996