ga2k/NetSolve の作成と実問題への適用

澤田　淳二

1　前回からの課題

前回からの課題としては，

- NetSolve を利用した ga2k の作成
- ga2k の実問題 (エンジン最適化問題) への適用

があった．

2　課題の達成状況

2.1　ga2k への Evaluation クラスの追加

NetSolve を利用した ga2k(以下，ga2k/NetSolve) を作成するために，まず，従来の ga2k から評価計算に関する部分を切り離し，Evaluation クラスという評価計算専用のクラスを作成した．このクラスを作成したのは，

- 全個体の評価計算を最後に一括して行いたい
- 実際の評価計算がどのように行われているかを意識せずに評価値を得たい

という理由からである．Evaluation クラスの fitness() メソッドを呼び出すことで，評価値を得るようにプログラムを変更した．詳しくは，第 14 回研究報告[1]を参照されたい．

2.2　ga2k/NetSolve

Evaluation クラスを継承し，評価計算に，NetSolve を用いた Grid RPC[2]を利用するクラスを作成した．並列化を行うために，Farming を用いたが，評価計算が一定数以上に達するとそれ以降，RPC が実行できずに異常終了してしまうという問題が発生した．

原因を調べるために，ノンブロッキング通信を用いて，N 回の RPC を n 並列で実行するプログラムを作成した．このプログラムでは，

1. n 並列で RPC を実行した後，すぐに結果の受信体勢に入る (netslwt() を使用)．なお，RPC が終了するまでは待機状態になる．結果の受信が終了したら，次の RPC を実行する

2. n 並列で RPC を実行した後，RPC が完了するのを確認 (netslpr() を使用) してから結果を受信する (netslwt() を使用)．結果の受信が終了したら，次の RPC を実行する

という 2 パターンを作成した．これらは，いずれも，1 つの RPC が終了したら，すぐに次の RPC を実行するというように，常に n 個の RPC が並列で実行される．これは，Farming の実装と同じ方式である．

この 2 パターンについて，n の値を，1, 2, 5, 10, 20, 50 と変化させて，最後の RPC を実行してから 1 分後の CLOSE_WAIT 状態のソケット数を計測した．RPC の総数は 100 である．

パターン 1 の結果を Fig. 1 に，パターン 2 の結果を Fig. 2 に示す．

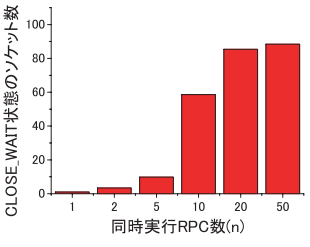

Fig. 1　すぐに受信体勢になった場合

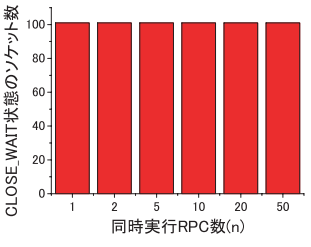

Fig. 2　完了を確認してから結果を受信した場合

なお，パターン 1 で n = 1 の場合は，ブロッキング通信と同じ処理を行うことになる．Fig. 1, 2 より，NetSolve では，ノンブロッキング通信において，ソケットが閉じられないことがあることがわかった．

2.3　ga2k の実問題への適用

ga2k を利用して，エンジン最適化問題を解くために，ga2k で実問題を解くための拡張を行った．実問題を解く際には，次の二点が問題となる．

- 遺伝子のデコード
- 評価計算

この問題を解決するために，オプションにより，デコード処理，評価計算処理を行う外部プログラムを指定できるように変更を加えた．

3　今後の課題

- ga2k/NetSolve の完成
- ga2k の実問題への適用

[1] http://mikilab.doshisha.ac.jp/dia/research/report/2002/0613/014/report20020613014.html

[2] Remote Procedure Call