
 汎用分散遺伝的アルゴリズムの実装とグリッドミドルウェアへの対応
 澤田 淳二

1 はじめに

構造物の設計, 化学反応シミュレータのパラメータ設定など, 様々な分野において最適化が必要とされている。そのため, 特定の分野に特化した最適化ソフトウェアではなく, 汎用的に利用可能な最適化ソフトウェアが有用となる。また, 大規模な実問題の最適化には, 非常に長い時間を必要とする。そのため, 最適化ソフトウェアでは, 計算時間短縮のために, 処理の並列化が可能であることが望ましい。

広範囲の問題に適用可能な最適化手法に遺伝的アルゴリズム (Genetic Algorithm: GA) がある。一方, 大規模な計算を行うための環境として, グリッドが注目されている。グリッドとは, 広い地域に配置された計算資源や情報資源を結びつけて, 広域的に分散/並列計算を行うための基盤のことである。

本研究では, 汎用最適化ソフトウェアである ga2k のグリッド環境での使用を想定し, グリッドのミドルウェアとして, Condor を利用する場合と NetSolve を利用する場合を比較する。

2 遺伝的アルゴリズム

GA は, 生物の進化を模倣した確率的な最適化手法である。GA では, 探索空間上の探索点を生物の個体とみなし, 個体の集合体である母集団に, 選択, 交叉, 突然変異という遺伝的操作を繰り返し加えることにより, 解探索が行われる。GA は, 目的関数の勾配情報を使用せずに解探索を行うため, 連続問題にも離散問題にも適用できるという利点がある。また, 多峰性の関数に対しても大域的最適解を得ることが可能である。

3 グリッドとそのミドルウェア

ネットワーク技術が進歩を遂げることに伴い, 高い通信性能を持つネットワークが利用できるようになった。これに伴い, 広い地域に配置された計算資源や情報資源を結びつけて, 広域的に分散/並列計算を行うための基盤となるグリッドと呼ばれる計算環境が研究されるようになった。グリッド環境では, いくつかの問題点が存在する。例えば, 異なるアーキテクチャをもつ計算機間でのデータ形式の変換や計算資源の管理などである。通常, これらの問題を解決するために, ミドルウェアを使用する。ミドルウェアの例として, ジョブスケジューリングを行う Condor¹⁾ やグリッド上に存在する計算資源を

利用するための RPC システムである NetSolve²⁾ が挙げられる。

4 汎用最適化ソフトウェア ga2k

最適化ソフトウェアに求められる機能は, 次の三点である。

- 任意の対象問題に適用可能であること
- 並列化が可能であること
- 高精度の解を探索可能であること

本研究では, 汎用最適化ソフトウェアとして ga2k を作成した。ga2k には次のような特徴がある。

- (1) 最適化手法として分散 GA を使用
- (2) 任意の対象問題のためのインターフェース
- (3) MPI を用いたサブ母集団ベースの並列モデル
- (4) MPI を用いたマスタースレーブモデル

Condor では, MPI プログラムの実行をサポートしている。Condor を用いてジョブを実行することにより, MPI マスタースレーブモデルをグリッド環境で使用することが可能となる。この場合, マスタースレーブも Condor Pool 中の計算機が用いられる。

- (5) NetSolve を用いたマスタースレーブモデル

NetSolve マスタースレーブモデルでは, 評価計算処理が NetSolve Server に分配され, それ以外の処理は NetSolve Client が行う。NetSolve により, 自動的に評価サーバの負荷状況に応じて評価計算処理の分配が行われる。

5 Condor と NetSolve を用いた ga2k マスタースレーブモデルの比較

ga2k のグリッド環境での使用を想定し, MPI マスタースレーブモデル ga2k を Condor を利用して実行する場合と, NetSolve マスタースレーブモデル ga2k を実行する場合を比較する。

実験には PC クラスタを使用した。実際のグリッド環境とは異なり, 通信速度や各計算機の処理能力は均一である。各ノードの CPU は Pentium III 800MHz であ

り、ノード間は100MbpsのFastEthernetで接続されている。

計算負荷の高い問題としてHIDECSを用いたディーゼルエンジン燃料噴射スケジュール最適化問題を使用した。HIDECSは、ディーゼル燃焼のシミュレートに高い計算負荷を必要とする。使用した実験環境では、1回の評価計算に約20秒の時間を要した。

5.1 実行時間

30回試行での実行時間の中央値をFig. 1に示す。NetSolveは、Condorを利用してMPIプログラムを実行する場合と比較して、長い実行時間が必要になることがわかる。

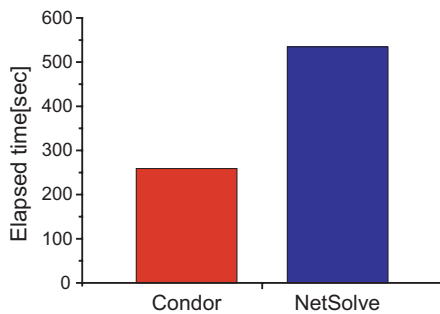


Fig. 1 Elapsed time

5.2 オーバーヘッド

1個体あたりの通信時間の平均値をFig. 2に示す。MPIでは、通信に要する時間は少ないことがわかる。一方、NetSolveでは、大きな通信オーバーヘッドがかかることがわかる。

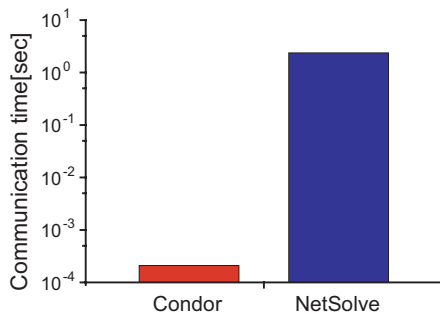


Fig. 2 Overhead

5.3 障害耐性

MPIプログラムをCondorを用いて実行する場合は、障害が発生した場合に常に実行が中断されてしまうが、NetSolveでは、エラーメッセージに応じて障害への対処が可能である。1回のジョブの実行に長い時間を必要とする場合、実行の途中でジョブが中断されてしまうとその損失は大きい。よって、障害耐性の点では、NetSolveが有利であると言える。ただし、NetSolveの障害対策を実現するためには、プログラマが適切な処理を記述する必要がある。

5.4 導入コスト

Condorは、NetSolveと比較して、導入時に必要となる設定項目が多い。また、インストールに必要なディスク容量も、Condorの方が大きくなる。

5.5 ジョブ投入時のユーザビリティ

Condorでは、ジョブを実行する際に、ジョブに関する情報を記述する必要があり、敷居の高いものとなる。NetSolveは、通常のプログラムと同じようにコマンドラインから実行するだけでよい。

6 まとめ

実行時間、システムのオーバーヘッドという観点で比較すると、MPIマスタースレーブモデルをCondorを利用して実行する方がNetSolveよりも有利であると言える。一方、障害耐性を比較すると、NetSolveでは障害の検出が行えるため、障害が発生した場合でもプログラムが継続可能であり、NetSolveマスタースレーブモデルの方が有利であると言える。

以上のことから、使用する環境が安定している(障害の発生が少ない、計算資源の追加・削除が頻繁に行われない)場合は、MPIマスタースレーブモデルをCondorを利用して実行するのが適していると言える。それに対し、使用する環境が安定していない場合は、NetSolveマスタースレーブモデルが適していると言える。

7 今後の課題

まもなく、NetSolve 1.5 がリリースされる。NetSolve 1.5では、処理の高速化が行われている。今後の課題として、NetSolve 1.5でのオーバーヘッドを計測することが挙げられる。

また、Condor上でマスタースレーブモデルGAを実装する方法として、MWライブラリというもの提供されている。このライブラリ上にga2kを実装し、NetSolveとの比較を行う必要がある。

参考文献

- 1) Jim Basney and Miron Livny. Deploying a High Throughput Computing Cluster. In Rajkumar Buyya, editor, *High Performance Cluster Computing*, Vol. 1, chapter 5. Prentice Hall PTR, May 1999.
- 2) Henri Casanova and Jack Dongarra. NetSolve: A Network Enabled Server for Solving Computational Science Problems. *The International Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 3, pp. 212-223, Fall 1997.