

PC-Clusterにおける並列分散GAの実装

廣安 知之^{*}, 三木 光範^{*}, 谷村 勇輔^{**}

* 同志社大学工学部 ** 同志社大学大学院

並列遺伝的アルゴリズム（PGA）には、代表的なモデルとして粗粒度モデルと細粒度モデルがある。本論文では、10ノード以下のPC-Cluster型並列計算機を用いて、速度向上率の観点から2つのPGAのモデルについて比較検討を行う。さらに、PC-ClusterのネットワークにFastEthernetを用いた場合とMyrinetを用いた場合の比較も行う。GAの対象問題としては、数学的関数であるRastrigin関数を用いている。実験により得られた結果は、PC-Clusterではネットワーク性能があまり良くないために、細粒度モデルが圧倒的に不利であった。粗粒度モデルでは、ある程度の規模の問題であれば、十分に速度向上が望めることが明らかとなった。

Implementation of Parallel and Distributed Genetic Algorithms Using a PC-Cluster

Tomoyuki HIROYASU, Mitsunori MIKI, and Yusuke TANIMURA

Doshisha University Graduate School of Doshisha University

In this paper, the characteristics of the typical two models of parallel genetic algorithms are compared. Those models are the coarse grained model and the micro grained model. Especially, the parallel efficiency on PC clusters that have not more than 10 nodes are discussed. The cluster used in this study have two kinds of network; FastEthernet and Myrinet. The characteristics are examined through the numerical examples; the Rastrigin function. In result, the micro grained model shows bad performance because of having not so high performance network on a PC-Cluster. The micro grained models shows good performance if the problems are properly big.

1 はじめに

PC-Clusterは、いくつもの汎用的なPCをネットワーク接続して構築される並列計算機である。現在のPCは数年前に比べて格段に性能が向上したため、高性能であるが高価な専用並列計算機に比べて、PC-Clusterは非常に高いコストパフォーマンスを有している。これまでにPC-Clusterを用いた応用研究としては、重力の多体シミュレーションやタンパク質構造の解析などがあるが、今後ますます広く利用されるようになるとPC-Cluster上で高いパフォーマンスを有するアプリケーションモデルが重要となる。そこで本研究では、広く使用されている最適化アプリケーションの1つである遺伝的アルゴリズム(GA: Genetic Algorithms)を並列化し、PC-Cluster上に実装するための並列モデルの検討を行う。GAは多くの並列性を有したアルゴリズムであると言われており[1]、これまでも並列計算の分野において研究が行われている。代表的な並列遺伝的アルゴリズム(PGA: Parallel Genetic Algorithms)には、粗粒度モデルと細粒

度モデルの2つがある。本論文ではこれら2つの並列モデルの比較として、通信性能の異なる2つのPC-Clusterを用いて速度向上率を検証する。

2 並列遺伝的アルゴリズムのモデル

2.1 遺伝的アルゴリズム

GAは自然界の進化と淘汰の仕組みを模倣したアルゴリズムである。GAは評価・選択・交叉・突然変異と呼ばれる遺伝的操作を繰り返すことで最適な解を見つけ出す最適化アルゴリズムである。本論文におけるGAの遺伝的操作は、交叉には一点交叉、選択にはルーレット選択を用いている。実用的なGAの並列モデルを考えると、通信によるオーバーヘッドやモデルの単純さの点から、粗粒度モデルと細粒度モデルを考えることができる。次節では、これらのモデルについて詳細を述べる。

2.2 粗粒度並列モデル

粗粒度モデルは一般に島モデルとも呼ばれる。このモデルでは、母集団を複数のサブ母集団に分割し、各母集団ごとに遺伝的操作を実行する。そし

て数世代の反復計算ごとに、あるサブ母集団のいくつかの個体を他のサブ母集団に移動させる。この遺伝的操作は移住と呼ばれる。本論文での移住は、全てのサブ母集団において同一の世代で適用し、ランダムリング型の移住を行う。これは移住世代ごとにランダムにリングを形成し、そのリングに基づいて移住先が決定される。つまり母集団全体でみるとある世代における移住は、図 1 に示すようにリング型の個体交換モデルとなる。移住個体数は初期パラメータで与えるが、移住個体や移住先などは、移住世代ごとにランダムに選ばれることになる。粗粒度モデル GA の処理の流れを図 2 に示す。

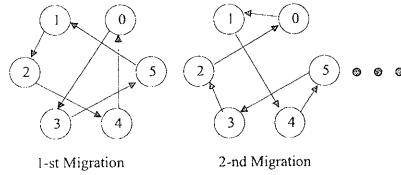


図 1: ある世代における移住の様子

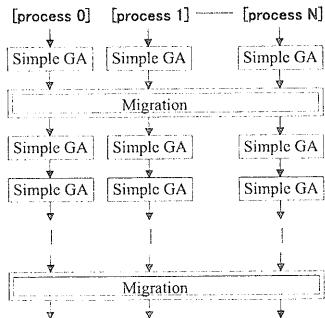


図 2: 粗粒度モデル GA の処理の流れ

2.3 細粒度並列モデル

GAにおいては評価の操作に要する時間が、全体の計算時間の大部分を占めることが多く、対象とする問題が複雑になるほどその傾向が強くなる。そこで非常に単純な並列モデルとして評価計算の部分を分散し、並列に計算を行う細粒度並列モデルが考えられる。これはマスター・スレーブ型をとり、評価を除く全ての遺伝的操作はマスターとなる 1 つのプロセスにおいて行う。評価の操作は、マスターから複数のスレーブに評価すべき個体のデータを送信し、スレーブにおいて実際の計算を行

い、結果をマスターに返すという手順となる。ロードバランスを考慮して、マスターからスレーブへの個体データの送信は 1 個体ずつとするため、実際に図 3 に示すようなモデルとなる。

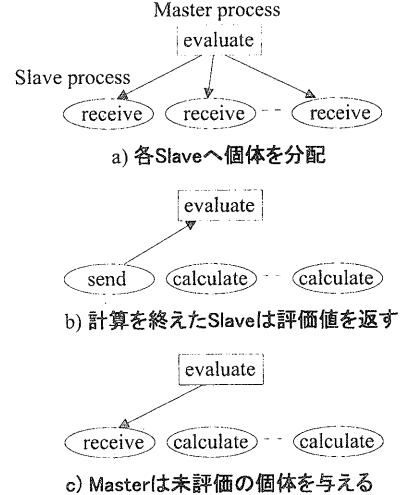


図 3: 個体の分散計算

3 粗粒度モデルと細粒度モデルの比較

前章では、2 つの PGA のモデルについて紹介した。本章では、それぞれのモデルにおいて計算時間を測定し、結果からわかる特徴について述べる。

3.1 使用する計算機環境

実験で用いた PC-Cluster 型の並列計算機は 8 台の PC から構成される。各ノードの接続には、100Mbps 対応のスイッチングハブと Myrinet の 2 種類のネットワークを用いる。Myrinet は Myricom 社により開発されたギガビットネットワークである[2]。Myrinet のホストインターフェースには LANai と呼ばれる RISC プロセッサが搭載され、Myrinet 上の通信プロトコルを制御し、高速な通信を実現している。表 1 にこの PC-Cluster のスペックを示す。また並列プログラムの記述には MPICH[3] と呼ばれる通信ライブラリを用いて、メッセージ・パッシング型の並列プログラムで PGA を記述している。

3.2 実験方法

計算速度の測定を行う実験において、テスト関数として Rastrigin 関数を用いる。Rastrigin 関数は式(1) のように表される。ここで f は適合度値であり f を最大にする x_i を求める。実際には f の最大値は 0.0 であることがわかっている。

表 1: クラスタのスペック

CPU	Pentium2 400MHz	
Memory	128MB	
OS	Linux2.2.10	
Network	Ethernet TCP/IP	Myrinet GM1.02
通信ライブラリ	MPICH1.1.2	

$$f = -10n - \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x)) \quad (1)$$

Rastrigin 関数は比較的大きな局所解をもつが、設計変数間に依存関係がないために GA では容易に最適解を見つけることができる。本実験においては、5 次元の Rastrigin 関数と 15 次元の Rastrigin 関数を用いる。各設計変数は 10 ビットで表現されるため、それぞれの問題において遺伝子長は 50 ビットと 150 ビットになる。それぞれの問題において、交叉率は 0.6、突然変異率は遺伝子長分の 1、移住間隔は 5、移住率はサブ母集団の個体数の 0.15 とした。個体数は総個体で 5 次元の場合に 960 個体、15 次元の場合に 1920 個体とした。つまり、粗粒度モデルにおいてはサブ母集団数はプロセッサ数と同等とし、各サブ母集団の個体数は総個体数/サブ母集団数となる。

実験では、さらにこの評価関数に一定の待ち時間を埋め込んで、評価計算に要する時間と通信時間の関係から、全体の計算時間がどのように変化するかを調査する。

3.3 粗粒度並列モデル GA の実験結果

粗粒度並列モデル GA を Ethernet-Cluster と Myrinet-Cluster で実行した時の速度向上率（総計算時間／1 プロセッサ時の総計算時間）を図 4 に示す。グラフ中のパラメータは、評価計算部分の待ち時間で単位は [sec] である。これより Ethernet では、評価計算時間が小さい時にはプロセッサ数を増やしてもあまり速度向上が得られないのに対し、Myrinet では十分な速度向上率が得られており、かつ理想値に近い。理想値以上の計算速度が得られているのは、粗粒度モデルを用いたことにより GA の解の探索能力が向上したためであると思われる。図 5 に示すグラフは、遺伝子長が異なる場合の問題を解いた結果である。遺伝子長が 50bit から 150bit と 3 倍になることで、通信負荷の影響を大きく受けているのがわかる。

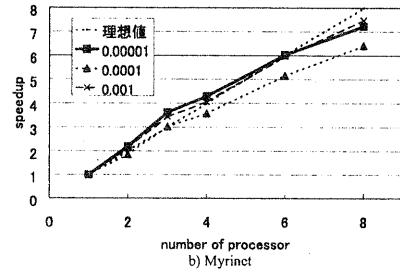
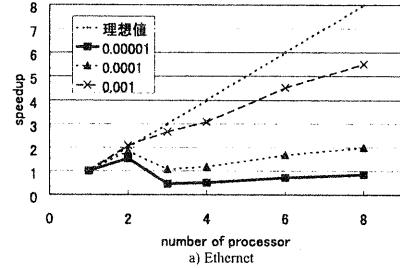


図 4: 粗粒度並列モデル GA の速度向上率 (50bit)

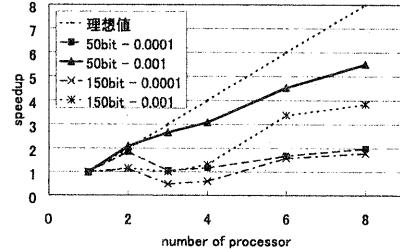


図 5: 50bit と 150bit の問題の比較 (Ethernet)

3.4 細粒度並列モデルの実験結果

細粒度並列モデル GA を Ethernet-Cluster と Myrinet-Cluster で実行した時の計算時間の測定結果を図 6 に示す。細粒度並列モデルでは Ethernet でも Myrinet でも、評価計算部分に入る待ち時間が非常に大きい場合でないと、十分な速度向上が得られなかった。これは粗粒度並列モデル GA に比べて、通信量および通信回数が大きくなるために、その部分のオーバーヘッドがかなり生じてしまうからであると考えることができる。さらに図中では、ある数以上のプロセッサを用いても実行時間が変化していないことがわかる。これは個体の分配を行う通信に時間がかかり、ほとんど未評価の個体を与えられないプロセッサが存在するためである。

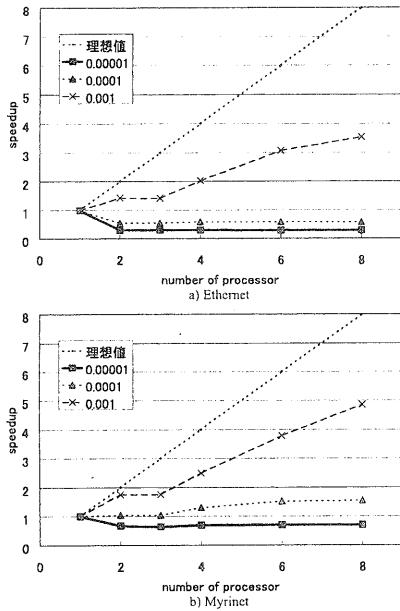


図 6: 細粒度並列モデル GA の速度向上率 (50bit)

4 考察

モデルの検討と実験の結果をまとめると、表 2 のようになる。

粗粒度並列モデル GA では、移住のための通信はある世代間隔で行われるためその通信負荷は小さい。高速なネットワークを用いたり、評価計算に十分に時間のかかる問題であれば、かなりの速度向上が期待できる。ただし本論文で用いたリング型の移住モデルでは、巡回的な依存関係が生じるために通信のデッドロックが発生する可能性がある。本論文では、MPI_Sendrecv() を用いることでこの問題を回避しているが、それでもサブ母集団数が奇数の場合には通信のステップが 1 つ多くなってしまうのである。

細粒度並列モデル GA では評価計算に時間がかかるような場合でも、マスター プロセスを走らせるノードが 1 つ必要となるため、例えば 8 ノードを用いた時の並列化効率の上限は $7/8$ となる。一方、評価計算に時間がかかるない場合は GA は毎世代、個体数分だけの通信を行う必要があるため、通信負荷は非常に高いものになる。そのような場合には、図 7 に示すようにユーザが意図したノード数を全て有効に利用することはできない。図 7 は片道の通信時間を 1、1 個体の評価計算時間を

n として、 n が十分に小さいと仮定してノード数が有効に利用できない時の予想される計算時間であり、実際の結果もこのようになった。

表 2: 並列モデルの比較

モデル	粗粒度並列 モデル GA	細粒度並列 モデル GA
通信負荷	低い	高い
最高並列化 効率	100%	100%未満
Simple GA と の必要計算量 の比較	減少	変化なし
GA のパラメ ータの数	増加	変化なし

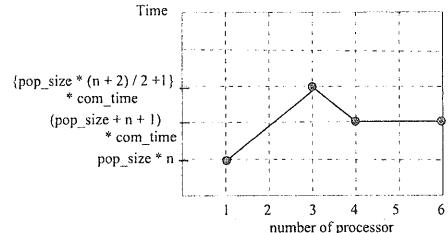


図 7: 全ノードが有効利用されない時の計算時間

5 結論

本論文では、粗粒度並列モデル GA と細粒度並列モデル GA のモデル比較を行った。計算速度は、対象問題の複雑さから生じる個体評価の計算時間と、並列に処理を行う時に必要な通信のオーバーヘッド時間との関係によって決定される。実験結果より、今回用いたような小規模な PC-Cluster では特に通信に気を配る必要があり、細粒度並列モデル GA は通信コストが高い場合低い場合とも PC-Cluster に適したモデルではないといえる。それに対して、粗粒度並列モデル GA では通信量がそれほど多くないので、並列化によって速度向上が実現できる。ただし移住のためのパラメータ設定が必要となる。

参考文献

- [1] Goldberg,D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989
- [2] <http://www.myri.com/>
- [3] <http://www.mcs.anl.gov/mpi/index.html>