

円整列問題に於ける通信の 同期・非同期による性能評価

4N-8

水野裕識 村岡洋一
早稲田大学理工学部

【1】 目的

分散された計算機上に機能的に分けられたプロセスを複数生成し、それらの相互作用の計算形態を明らかにすることが求められている。本稿では、その具体例として複数の移動体(MO:Moving Object)が自律分散的に円形成を行うアルゴリズム^[1]を計算機実装し、各移動のタイミングを同期させた場合と非同期に行わせた場合の円収束の比較について論じる。

【2】 円整列問題の同期、非同期について

まず円整列アルゴリズム^[1]を説明する。この問題における相手情報は、2次元仮想グリッドの位置(x,y)として表す。

前提) 各移動体は円の直径を獲得してある。

1) 各移動体は他の移動体との距離を求め、最遠、最近移動体を知る。

2.1) 最遠移動体距離 > 直径の場合

最遠移動体に少し接近する。 (式1)

2.2) 最遠移動体距離 < 直径の場合

最遠移動体から少し遠のく。 (式2)

2.3) 最遠移動体距離 = 直径の場合

最近移動体から少し遠のく。 (式3)

$$P_{new} = P - \alpha \frac{(P - P_{far})}{|P - P_{far}|} \quad (1)$$

α : 基準移動量 (=1.0)

$$P_{new} = P + \alpha \frac{(P - P_{far})}{|P - P_{far}|} \quad (2)$$

$$P_{new} = P + \alpha \frac{(P - P_{near})}{|P - P_{near}|} \quad (3)$$

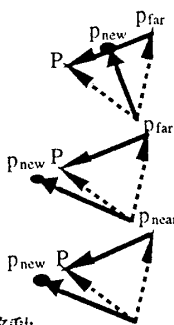


図1 MOの3種類の移動

各移動体は他移動体の状況を監視しながら移動するので、各移動体は相手情報を獲得する通信とその情報による位置判断の2つの処理機構が必要となる。

通信により世代の古い相手位置を獲得し、世代の古いデータの作用を受けて新世代の位置移動が行わ

Comparison of convergence of sync and async communication concernd with the problem which each MO forms up in circle
Hironori MIZUNO Yoichi MURAOKA

School of Science and Engineering ,Waseda University

れる点から、円整列は関数収束問題と捉えられる。

(1)同期通信の場合 --- 位置の更新を一斉に行うという意味での同期である。この場合収束時間は通信時間の最も遅いMOに影響をされる可能性を持つ。一方で、非同期通信の場合に比べて通信量が減り、収束時間が減少する可能性を持つ。

(2)非同期通信の場合 --- 通信、位置の処理はどこにも待合せが存在せず、MOは次々と処理を進行する。従って、相手の位置は現時刻から見て常に最新位置として獲得しうるが、非同期では待合せがないため、冗長な通信が増える可能性を持つ。

関数収束を効率良く行わせる為に、同期通信・非同期通信のどちらで行わせるべきかを確認する。

【3】 PVM^[2]による円整列システム実装環境

各移動体はEthernet 接続されたOpenSystemな計算機環境にPVMを用いて実装した。1つの移動体は、同一ホストに割り当て、位置の問い合わせを行う返答プロセスと位置を決定する2つのループプロセスで構成する(図2)。また、それぞれのMOは位置をメッセージデータとして駆動する方式を取る。

1) 自己位置決定プロセス

- a) 利己的判断に基づく位置の獲得を行う。
- b) 他の移動体位置に基づく自己位置決定、更新
- c) 自己位置の更新を返答プロセスに伝達する。

a) 通信相手は相手との距離に応じて選択する。また、円収束を速く行う為に、返答の遅いプロセスに対しては通信を打ち切り、過去の履歴に基づいて相手を決める。

b) 円整列アルゴリズムは位置に注目する。通信時間に対して1度の移動に対する判断時間は無視しうる。通信の同期は図2中、位置更新の直前でとる。

2) 返答プロセス

- a) 自分以外の問い合わせを受理し位置を返す。
- b) 自分の更新を受理し自らの位置を更新する。
- a) 問い合わせを行いたいMOに対して自分のHost番号と位置請求のタグを付して投げる。問い合わせのタグに応じてHost番号に現在位置を返答する。

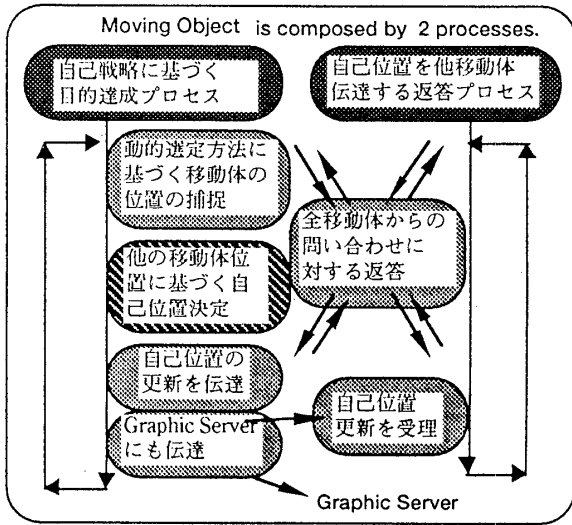


図2. 1 MOのプロセス構成

b) 1.c)による自分の位置変更要求をこの部分で受理する。これにより常に最新の自己位置が相手に返答されるようになる。

【4】同期と非同期による収束比較

2つの基準値（収束時間と、通信回数）を設定し、この2つの基準値により評価する。通信回数は終了迄の全MO通信回数の総和とする。

MOの移動アルゴリズムでは、最遠個体と最近個体との通信が重要なので、MOは距離テーブルに応じた問い合わせの優先順位を持つ。しかし、MOの移動により優先順位は変化するために、全体通信をできるだけ避けつつ、最遠および最近の2個体との通信を続けることが重要になる。図3に、全体への問い合わせを頻度(何度一度全体へ問い合わせるか)を横軸に取り、収束時間及び通信回数に関する同期、非同

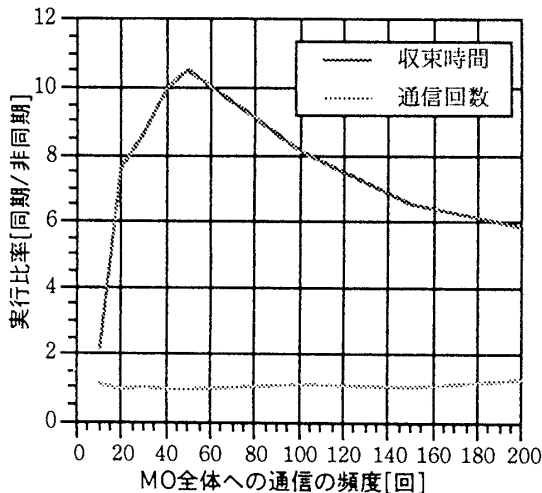


図3. 同期通信・非同期通信の比較

期で行った場合の相対比を示す。

【同期通信、非同期通信の収束の評価】

本実装に於ける円収束アルゴリズムでは、非同期通信が同期通信に比較して収束が早い結果を得た。

3章で述べたように各MOの1回の位置移動の更新時間は通信時間に等しいと見なせる。同期の場合、他のMOは最も通信時間が遅いMOに待たされる。逆に、非同期の場合、通信時間の空きを詰めることが可能な為、時間当たりの参照し合あう比率が高まり、更新された位置データの波及が速くなる結果、収束時間が短縮すると考えられる。

一方通信回数について非同期の場合、図3を見ると2割程度同期に比べて増える。非同期の場合には、通信回数の増加にもかかわらず、収束時間が短い。

しかし、今回は10MOの構成のため通信路が飽和しないで済んだが、非同期は通信回数が増えるという不利な点を持つ。通信が飽和しない範囲において収束時間が結果として短くなる非同期システムが有効であると考えられる。

一方で同期非同期共に、全体として整合性を取れない状態を生ずる可能性がある。これは1つのMOの移動が全体の円収束状態に対して寄与しない場合で、全MOが対峙してしまう場合である。また、近傍の相互作用だけでは、円でないだましの形態(円に近い三角形)が生ずる場合がある。これはMOの移動が近傍の相互作用による局所的な位置を更新するだけでなく、全体の状態を監視するといった大局的に見た場合の位置移動制御が必要となる。本稿における大局的に見た位置更新は任意のMO位置を確率的に揺らすことで収束しない場合を回避した。

【5】むすび

本稿では、円整列問題に於ける通信を同期させた場合と非同期で行わせた場合の収束時間による比較を行い、非同期で行わせた方が収束は早くなる結果を報告した。今後は実ロボット等の動的に変動する環境における相互作用の計算形態を明らかにする。

参考文献

[1] 水野,村岡,"円整列問題に於ける利己的判断に基づいた通信手法の提案",第3回マルチエージェントと協調計算ワークショップ
 [2] Jack Kongarra,"PVM3.0 User's Guide and Reference Manual",Oak Ridge National Laboratory ORNL/TM-12187
 [3] 澤田,田辺,吉田,"自律分散的に秩序形成を行うロボット群の並列計算機上でのシミュレーション",情処第46回全大,3A-1