1J-04

並列 VC における冗長的な通信グループの動的再構成法

富田 大喜 †　　黒川 陽太 †　　福士 将 †
山口大学大学院創成科学研究科 †

1　はじめに

ボランティアコンピューティング（VC ）は，有志から提供されたコンピュータ（ノード ）をインターネット経由で接続し，1 つの大規模な計算機として利用する仕組みである．VC では，ノードが計算中に離脱する問題があるため，個々のジョブが独立した分散計算のみを対象としており，ジョブ間で通信が必要になる並列計算は対象外とされてきた [1]．VC で並列計算を実行させようとする場合，通信相手のノードの離脱により，通信が行えず，処理が完了しない問題がある．

本報告では，並列 VC の問題の解決策として，並列計算を行うノードの集合（通信グループ ）を冗長化し，動的に再構成する手法を提案する．

2　並列 VC における冗長的な通信グループの動的再構成法

本章では，提案手法の説明の為に，想定する並列 VC の構成，および離脱検知の方法を説明し，その後，提案手法の具体的な手続きを説明する．

2.1　並列 VC の構成

対象とする並列計算のモデルとして，1 つの計算処理（ジョブ ）が P 個のサブジョブに分割され，P 台のノードで通信を行いながら並列に実行されるモデルを想定する．このとき，P 台のノードの集合を通信グループと呼ぶ．

提案手法では，図 1 に示すように，通信グループ G^i を R 個に冗長化した構成をとる $(0 \leq i < R)$．この構成では，各 G^i の中の j 番目のノード n_j^i は，同一のサブジョブを実行することになる．このノードの集合をクラスタと呼び，C_j で表す $(0 \leq j < P)$．

ノード n_j^i が離脱した場合，G^i の各ノードは通信相手が不在になる．この場合，C_j には同じサブジョブを担当するノード n_j^k が存在することに着目し（$k \neq i$ ），n_j^k に n_j^i の通信の役割を代替させることにより，G^i を動的に再構成する．

A Dynamic Reconfiguration Method of Redundant Communication Groups for Parallel Volunteer Computing
† Graduate School of Sciences and Technology for Innovation, Yamaguchi University

図 1 に示すノード以外にも，遊休ノード n_{idle} が存在し，マスタ $master$ によって管理されているものとする．

以降で，ノードの生存確認と動的再構成の詳細について説明する．

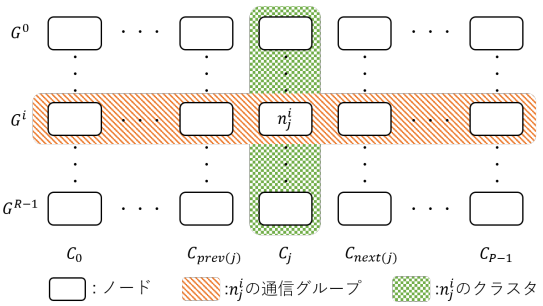

図 1　想定する並列 VC の構成

2.2　ノードの生存・離脱の確認

ノードの生存・離脱の確認は，定期的にハートビート信号（HB）を送り合うことで行う．n_j^i は，$n_{neighbor} \in (G^i \cup C_{prev(j)} \cup C_j \cup C_{next(j)})$ の範囲の $n_{neighbor}$ に対して，HB を送信する．ここで，$prev(j) = j - 1$，$next(j) = j + 1$ であり，この加減算は R を法とするものである．すなわち，各ノードは，自身が所属する通信グループとクラスタに加えて，隣接クラスタのノードに対して HB を送信する．HB を一定時間受信しない場合，そのノードが離脱したものを判断する．

n_j^i は,$n_{neighbor}$ と直接通信を行うためにアドレスをテーブル $n_j^i.table$ に保存している．$n_j^i.table$ は通信グループ番号とクラスタ番号の組 (roleID) とアドレスを対応付けるテーブルである．

2.3　動的再構成のための手続き

通信グループの動的再構成は，代理手続き，ノード補充手続き，クラスタ補充手続きの 3 つの主手続きで構成される．また，これらの手続きを実行するために，ASK 手続き，INFORM 手続き，ADVERTISE 手続き，INHERIT 手続きの 4 つの副手続きがある．

2.3.1　ASK 手続き

ノード n からノード m に，G^i に属する全ノードのアドレスを問い合わせる手続きを $ASK(G^i)$ 手続きと呼び，

$$n \xrightarrow{ASK(G^i)} m,$$

と表記する．

m がこの問い合わせを受けた時，$m.table$ の情報が，返答のために参照される．

2.3.2 INFORM 手続き

ノード n からノード m に，G^i，C_j に属するノード n_{new} のアドレスが $addr(n_{new})$ であることを伝える手続きを $INFORM(i,j,addr(n_{new}))$ 手続きと呼び，

$$n \xrightarrow{INFORM(i,j,addr(n_{new}))} m,$$

と表記する．

ノード m がこの情報を受けた時，テーブル $m.table$ の roleID(i,j) と紐づくアドレス情報が $addr(n_{new})$ で更新される．

2.3.3 ADVERTISE 手続き

ノード n からあるノードの集合 N に属する全ノードに対して n が G^i，C_j に属することを伝える手続きを $ADVERTISE(i,j)$ 手続きと呼び，

$$n \xrightarrow{ADVERTISE(i,j)} N,$$

と表記する．

この手続きは，N の全ノードに対して $INFORM(i,j,addr(n))$ を行うことで実現される．

2.3.4 INHERIT 手続き

ノード n からあるノード m に対して，あるノードの集合 N に属する全ノードについて，アドレス $addr(n)(n \in N)$ と roleID(i,j) を伝える手続きを $INHERIT(N)$ 手続きと呼び，

$$n \xrightarrow{INHERIT(N)} m,$$

と表記する．

2.3.5 代理手続き

ノード n_j^i が離脱したとき，C_j の各ノードが離脱を検知することで，代理手続きが始まる．その手順は以下の通りである．

1. C_j のノード間で代理を行うノード n_j^{proxy} を決定する
2. $n_j^{proxy} \xrightarrow{ASK(G^i)} n_{next(j)}^i$
3. $n_j^{proxy} \xrightarrow{ADVERTISE(i,j)} (G^i \cup C_{prev(j)} \cup C_j \cup C_{next(j)})$

2.3.6 ノード補充手続き

C_j のノードが少なくなってきたことを，n_j^i を代理しているノード n_j^{proxy} が検知したとき，ノード補充手続きが始まる．その手順は以下の通りである．

1. n_j^{proxy} は roleID(i,j) を遊休ノード n_{idle} に割り当てるように，$master$ に要請する
2. $master \xrightarrow{INFORM(i,j,addr(n_{idle}))} n_j^{proxy}$
3. $n_j^{proxy} \xrightarrow{INHERIT(G^i \cup C_{prev(j)} \cup C_j \cup C_{next(j)})} n_{idle}$
4. $n_{idle} \xrightarrow{ADVERTISE(i,j)} (G^i \cup C_{prev(j)} \cup C_j \cup C_{next(j)})$

2.3.7 クラスタ補充手続き

C_j の全ノードが離脱したことを，$C_{prev(j)}, C_{next(j)}$ が検知することでクラスタ補充手続きが始まる．その手順は以下の通りである．ただし，各ステップは全ての $i(0 \leq i < R)$ に対して同時に行われる．

1. $n_{prev(j)}^i, n_{next(j)}^i$ は，C_j を補充するように $master$ に要請する
2. $master \xrightarrow{INHERIT(C_{new})} n_{new}^i$
3. $master \xrightarrow{INHERIT(C_{new})} n_{prev(j)}^i, n_{next(j)}^i$
4. $n_{prev(j)}^i \xrightarrow{INHERIT(C_{prev(j)})} n_{new}^i$
5. $n_{next(j)}^i \xrightarrow{INHERIT(C_{next(j)})} n_{new}^i$
6. $n_{new}^i \xrightarrow{ASK(G^i)} n_{next(j)}^i$
7. $n_{new}^i \xrightarrow{ADVERTISE(i,j)} G^i$

3 通信量の見積もり

各手続きに必要な通信量をパケット数単位で計算した．まず，副手続きの通信量を計算する．できるだけ情報を 1 パケットにまとめて送ると仮定する．INFORM 手続きと ASK 手続きに用いる通信量は，$O(1)$ パケットである．ノード集合 N に対する $ADVERTISE(N)$ の通信量は，$O(|N|)$ パケットである．$INHERIT(N)$ の通信量は，$O(1)$ パケットである．次に，主手続きの通信量を計算する．代理手続きとノード補充手続きの通信量は，どちらも $O(R+3P)$ である．クラスタ補充手続きの通信量は，$O(RP+9P)$ である．

4 結論

並列 VC の実現に向けて，通信グループの動的再構成方法を提案し，通信量を計算した．今後の課題は，提案手法を実装し，動的再構成動作を検証するとともに，再構成時間などの性能評価を行うことである．

参考文献

[1] Folding@home, foldingathome.org (2022)