並列処理におけるＰＥ間に渡るゴールの重みつき参照カウントを用いた管理方式

六沢 一昭＊　吉吉 信行＊　瀧　和男＊
吉田かおる＊　稲村　雄＊　中島　浩＊＊

＊（財）新世代コンピュータ技術開発機構　＊＊三菱電機(株)

1．はじめに

ゴールの終了の検出と強制終了は，論理型プログラムをマルチプロセッサで並列実行する際の主要課題である。これらは密結合マルチプロセッサでは簡単であるが疎結合マルチプロセッサでは難しい。これは，送信はなされたが宛先プロセッサにはまだ到着していないようなゴールがネットワーク上に存在しうるためである。

本稿では，疎結合マルチプロセッサであるマルチＰＳＩ[1]における Weighted Throw Counting(WTC)方式によるゴールの終了検出と強制終了について述べる。

WTC方式はルートとリーフに重みを持たすことによって管理を行なうものである。この重み付けをＧＣに適用したものに Weighted Reference Counting方式[3,4]がある。

2．計算モデル

マルチＰＳＩは最大６４台のＰＳＩをネットワークで結合した並列マシンである。共有メモリはなく各ＰＳＩ(以下ＰＥと略す)はメッセージ通信によってＰＥ間処理を行なう。同一ＰＥ間ではメッセージは送出した順序に到着するが，一定時間内に到着する保証はない。

すべてのゴールは荘園の下で管理される[2]。システムには有限個の荘園があり，荘園ＩＤで識別される。ゴールは新しい荘園を生成することができる。このため荘園はネストすることがあるが，ネストした荘園はゴールと同じように扱うことができるので，本稿では省略する。

負荷分散などのためにゴールが他のＰＥへ投げ出されることがある。投げ出されてから到着するまで時間がかかるため，ある与えられた時刻においてネットワーク上にゴールが存在する可能性がある(図１)。

ゴールのあるＰＥには里親がありゴールはその下に属する。里親は１つの荘園に対して１ＰＥに１つだけ存在する。受信したゴールは対応する(同じ荘園ＩＤを持つ)里親に加えられる。対応する里親がない場合は新しく作る。里親は自分の下のゴールの強制終了を行なうことができ，自分の下のすべてのゴールが終了すると消滅する。

3．問題点

荘園は全ゴールの終了の検出及び強制終了ができなければならない。以下にそれらを行なう際の問題点を述べる。

3－1．終了の検出

里親は消滅する際その旨を伝えるメッセージ(%terminated)を荘園へ送ることができる。しかしネットワーク上にもゴールがあるかもしれないので，たとえ荘園がすべてのＰＥから %terminated を受信してもすべてのゴールが終了したとは限らない。あるＰＥは %terminated を送信した後，ゴールを再び受信するかもしれない(図２)。

ゴール受信に対して必ず応答を返すならば，ゴールの投げ出しと応答をカウントすることによってゴールの終了を正しく検出することができる[5]。しかしこの方式ではゴール投げ出しと同数の制御(応答)メッセージが必要になってしまう。

3－2．強制終了

荘園が強制終了を通知するメッセージ(%abort)をブロードキャストすると，その時点で里親の下にあったゴールの強制終了は可能である。しかしネットワーク上にあったゴールは強制終了できない。

ゴールの強制終了を行なった後も里親が消滅せず，その後受信したゴールを棄ててしまうのならば，ブロードキャストによる強制終了は可能である。

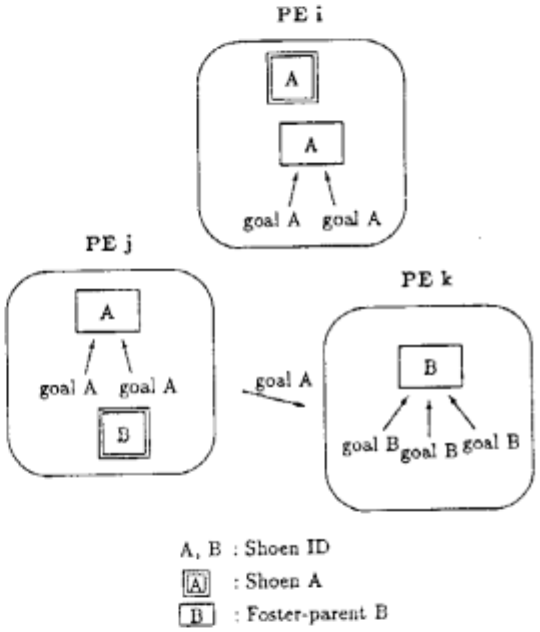

図１．荘園と里親(Foster-parent)，ゴール

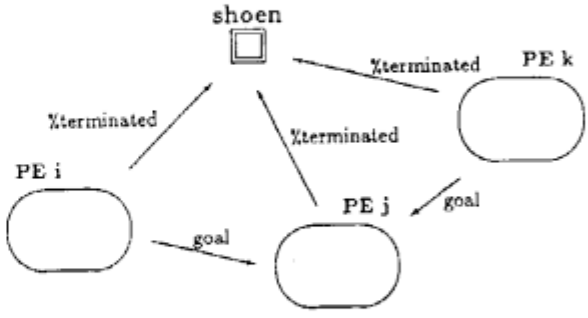

図２．まだゴールがネットワーク上にある。

A Termination Detection and Abortion Scheme for Distributed Processing Systems Using Weighted Throw Counting

Kazuaki HOKUSAWA, Nobuyuki ICHIYOSHI, Kazuo TAKI, Kaoru YOSHIDA, Yu INAMURA (ICOT), Hiroshi NAKASHIMA (MELCO)

しかしこの方式には重大な欠点がある。少数のＰＥにのみ里親が存在した場合多くの %abort が無駄になってしまう。また荘園ＩＤの再利用ができない。荘園ＩＤの再利用を可能にするには，すべてのゴールが強制終了されたことを検出することが必要である。

4．解決方法

MTC方式による終了の検出と強制終了について述べる。この方式は，制御メッセージの送信は少なく，荘園ＩＤの再利用も可能である。

4－1．終了の検出

荘園と里親及びネットワーク上のゴールに"重み"を持たせる。荘園の重みは負の整数，里親及びゴールの重みは正の整数である。そして重みの合計がゼロになるように制御する。この結果，すべてのゴールが終了した時のみ荘園の重みがゼロになる(図３)。

里親はゴールに重みを付けて投げ出す。自分の重みはその分だけ減らす。ゴールを受信するとその重みを対応する里親に加える。対応する里親が存在せず新たに生成した場合は重みの初期値をゴールの重みにセットする。

自分の下にあるゴールがすべて終了すると里親は消滅し荘園へ %terminated を送る。%terminated は消滅した里親が保持していた重みを運ぶ。荘園が %terminated を受信すると，運ばれてきた重みを自分の重みに加える。この操作により重みがゼロになったならば全ゴールの終了が検出される。

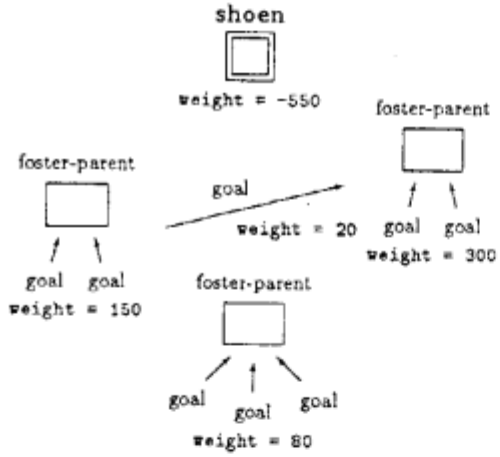

図３．Weighted Throw Counting

4－2．強制終了

4-1 で述べた方法により終了の検出は正しく行われる。従ってすべての里親に %abort が届けばよい。3-2 で述べたように %abort をブロードキャストするのでは不十分である。%abort 受信後に生成された里親には %abort が届かないためである。

ここで里親の生成を荘園へ伝えるメッセージ(%ready)を導入する。新たに里親を生成した時 %ready を荘園へ送信する。荘園は %ready を受信すると送信元ＰＥを記憶する。この記憶は %terminated を受信した際削除する。

強制終了は以下のように %abort を送信することによって行なう。尚 %abort にもゴールと同様重みを付ける。

① 記憶しているＰＥへ %abort を送信する。

② ①の後 %ready を受信した場合，送信元ＰＥへ %abort を送信する。

強制終了開始時に荘園が把握していた里親は①によって，①の後に検出された里親は②によって消滅する(図４)。

%abort を受信すると里親は自分の下のゴールをすべて終了させ消滅し %terminated を荘園へ送信する。%terminated は消滅した里親の重みと %abort の重みの合計を運ぶ。里親が既に消滅していた場合は，%abort の重みを荘園へ返却するメッセージ(%return)を荘園へ送る。荘園は %return を受信すると %terminated 受信時と同様に運ばれてきた重みを自分の重みに加える。この操作により重みがゼロになったならば全ゴールの強制終了が検出される。

%abort を１回送信すると１つの里親が消滅しその重みが荘園へ戻る。里親とネットワーク中のゴールが持つ重みの合計は有限なので，有限回の %abort送信によって強制終了は完了する。

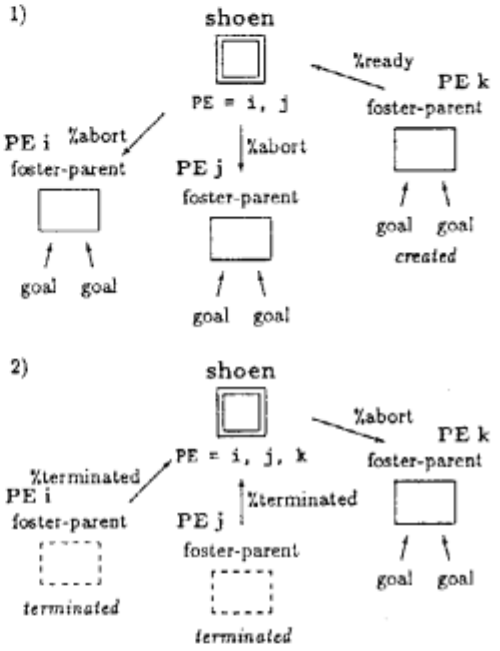

図４．強制終了

4－3．重みが１である時の処理

重みが１であるとゴールを投げ出すことができない。この時里親は荘園へ重みの割り付けを要求するメッセージを送信する。このメッセージを受信すると荘園は重みを割り付けるメッセージを返信する。

5．おわりに

MTC方式によるゴールの終了検出と強制終了について述べた。この方式の利点は，「制御メッセージの送信が少なくてすむ。」，「荘園ＩＤの再利用が可能。」である。

<参考文献>

[1] 瀧 和男 他，"Multi-PSI システムの概要，" 第32回情処全大 5Q-8，1986.

[2] 佐藤 裕幸 他，"並列論理型ＯＳ－ＰＩＭＯＳ(1)，" 第35回情処全大 4D-3，1987.

[3] D.I.Bevan，"Distributed garbage collection using reference counting，" PARLE，1987.

[4] P.Watson and I.Watson，"An efficient garbage collection scheme for parallel computer architectures，" PARLE，1987.

[5] N.Ichiyoshi et al，"A Distributed Implementation of Flat GHC on the Multi-PSI，" ICLP，1987.