同期機構付き分散共有メモリの提案

3T－1

屋代寛、後藤努、林剛久、吉澤聡
(株)日立製作所　中央研究所

１はじめに

近年、ワークステーションの普及にともない、複数台のワークステーションをネットワークで結合したクラスタシステムが並列処理に利用されるようになり、並列処理がより身近なものになってきた。

クラスタシステムは共有メモリ型と分散メモリ型のアーキテクチャのうち、分散メモリ型に分類される。分散メモリ型は共有メモリ型に比べ、スケーラビリティの面では優れているが、プログラミングが困難であると言われている。この問題を軽減するため、分散共有メモリ方式がいくつか提案されてきているが[1]、メッセージ通信と比較すると並列処理による性能向上を得にくいという問題があった。

本報告では、分散共有メモリのプログラミングの容易さを保ちつつ、並列処理における性能向上の得られる API(Application Program Interface)として同期機構付き共有メモリを提案する。本提案のAPIを実現するために、信頼性のあるブロードキャストを利用した一貫性保証方式を採用し、実現方式の検討を行なった結果について述べる。

２.同期機構付き分散共有メモリ

分散共有メモリによるプログラミングの容易さは次の点であると考える。

位置透過性：どの計算機に最新データが存在するかをプログラマが意識しない。

参照による共有：アドレスを指定することによって共有データにアクセスすることができる。

一方で、クラスタシステムにおける分散共有メモリでは仮想記憶(ページング)とメッセージ通信を用いるが、この方式には次の問題がある。

(1) データ転送回数：false sharing が発生した場合には，メモリ書き込み毎にページングが発生する可能性がある．例えばfalse sharingを避けるためにページを意識する等，分散メモリ型アーキテクチャ向けのチューニングが必要である．

(2) 同期オーバヘッド：共有するデータの依存関係を保証する必要があるため、バリア同期などを用いて明示的な同期を記述する必要がある。このため、メッセージ通信では不要な同期のための通信がデータ転送とは別に発生してしまう。

前述第１の問題に対しては、データ転送のためのチューニングをするには、ページという物理的な単位を意識するのではなく、より論理的な単位（例えば、変数レベル)で意識できるのが望ましいという観点から、論理的なデータオブジェクト単位で共有データの管理を行なう方式[2]を提案した。

筆者等は、第２の同期オーバヘッドの問題を解決するために、前回報告した方式を改良した同期機構付き分散共有メモリを提案する。インタフェースの特徴は次の通りである(表１参照)。

(1) ユーザが共有データの粒度を意識し、共有データ生成時にデータサイズを宣言する。(nsmget)

(2) 共有データの番号とメモリ空間を対応付けて管理する(表中の共有メモリ識別子)。(nsmat)

(3) ユーザが一貫性保証に関するタイミングを指定可能とする。

・共有データ獲得(nsmacq)

・共有データ解放(nsmrel)

(4) 共有データ獲得(nsmacq)時に指定する待ち合わせ数と優先度により、共有データの依存関係による同期をとり、余分な同期のための通信を削減可能。

・待ち合わせ指定：同期をとる計算機の数を設定し、データ獲得要求の数が設定した計算機の数に等しくなるまでデータへのアクセスを禁止する。

・優先度指定：共有データにアクセスする計算機の順序を指定するために、優先度を設定する。

A study of distributed shared memory with embedded synchronization mechanism

Hiroshi YASHIRO, Tsutomu GOTO,
Takehisa HAYASHI, Satoshi YOSHIZAWA
Central Research Laboratory, HITACHI Ltd.
1-280 Higashi-koigakubo, Kokubunji, Tokyo 185, JAPAN

表1 同期機構付き分散共有メモリのインタフェース

機能	引き数
領域の確保	nsmget(キー値,共有データの大きさ)
アドレスの参照	nsmat(共有メモリ識別子)
アドレスの解放	nsmdt(共有メモリ識別子)
一貫性制御(獲得)	nsmacq(共有メモリ識別子、待ち数、優先度)
一貫性制御(解放)	nsmrel(共有メモリ識別子)

3 実現方式

分散共有メモリ型計算機において、共有データの一貫性保証をメッセージ通信によって行なう必要がある。また、メッセージ通信の方式には、さらに、1)ブロードキャスト使用, 2)ユニキャスト (1対1通信) のみの使用という選択肢がある。前者1)では、データを必要としていない計算機でも通信オーバヘッドが発生してしまう。しかし、共有している計算機数に対して、一貫性保証の手続きを一定コストで実行でき、かつそのアルゴリズムが単純になるという利点がある。ただし、ここで用いるブロードキャストはイーサネットやFDDIが提供するブロードキャスト機能では不十分であり，信頼性を付加する[3]必要がある[注1]。

この信頼性のあるブロードキャストに基づいて、同期機構を制御し、かつ、共有メモリの一貫性保証をするための制御メッセージを次のように設計した。

ACQUIRE:データ獲得要求を指示するnsmacq関数によって、自計算機に最新の共有データが無い場合には、他計算機に対してブロードキャストを発行する。このメッセージには、(a)共有メモリ識別子、(b)共有メモリ待ち合わせ数、(c)優先度の情報を含む。

ACQUIRE_ACK: 最新の共有データを保有する計算機がACQUIREプロトコルを受信した場合に、最新の共有データを保有する計算機が発行する。このメッセージには、(a)共有メモリ識別子、(b)最新データを保有する計算機、(c)最新の共有データ、(d)待ち計算機リスト待ち計算機リストは、優先度指定によって獲得要求を保留されている他計算機のリストである。

[注1]ここでいう「信頼性のあるブロードキャスト」とは、ブロードキャストしたパケットが消失することなく全計算機に到着し、かつ、ブロードキャストしたパケットを受信する順序が全計算機で必ず同じになることを保証する。

図1では、上述のプロトコルを用いて、同期制御と共有メモリの一貫性制御が行われる様子を示している。この例では、計算機j→計算機k→計算機iという順番で共有データを処理する。計算機kよりも計算機iが先にメモリの獲得要求を出しても、要求優先度の低い計算機iは待たされ、先に計算機kで共有メモリの獲得をすることができる。また、同期情報はデータ転送にのみ転送してしまうため、同期のための余分な通信は発生しない。

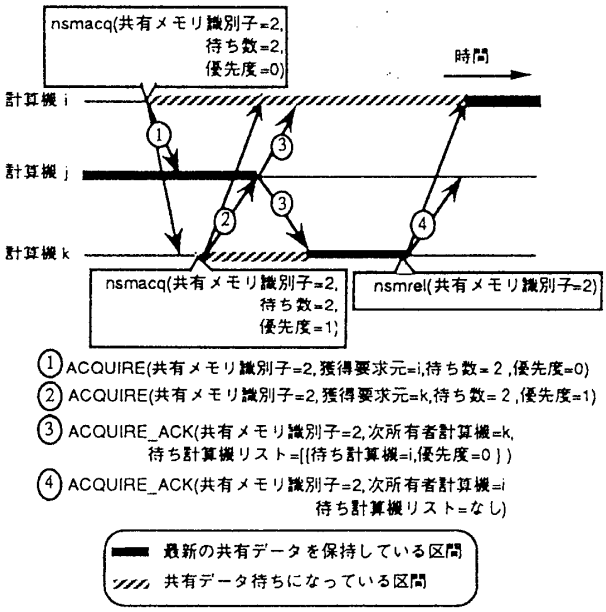

図1 共有メモリ一貫性保証のためのメッセージ

4 まとめと今後の課題

同期機構付き分散共有メモリのAPIの提案と実現方式を示した。検討した実現方式においては、共有データ一貫性保証のためのデータ転送と合わせて同期情報も転送するため、同期のための転送が発生しない。今後は、今回提案した方式に関する評価を行なう予定である。

参考文献

[1] B.Notzner *et.al.*:"Distributed Shared Memory: A Survey of Issues and Algorithms", IEEE Computer, 1991.

[2] 山内 他:"分散共有オブジェクトの提案",第48回情処全大,1994

[3] A.S.Tanenbaum *et al.*:"Parallel Programming Using Shared Objects and Broadcasting", IEEE Computer, 1992