

共有配列を用いたオブジェクト指向自動分散計算システム¹

2 J-4

畠山 正行, 上原 均, 齊藤 賢二

茨城大学工学部情報工学科

{masayuki, uehara, st94042}@cis.ibaraki.ac.jp

1 はじめに

我々の研究グループでは, データ構造とメソッドを強固な1セットとして扱うオブジェクトをモデル単位としたオブジェクトベースに基づいたシミュレーションおよびその支援環境について研究している。

その支援環境の一環として複数の計算機をあたかも1つの計算機のようにアクセスでき, 計算を自動化する試みが行なわれている。現在, 複数の計算機を小さなエージェントの働きにより1つの仮想計算機としてユーザに提供するシステム Gaea[1] が実現されている。

Gaeaでは分散計算を実現するために, そのデータやメッセージの交換に黑板モデル[1]を用いている。黑板モデルは理解と実装が簡単であるが, そのアクセススピードに問題があり, またユーザはそのフォーマットを理解する必要もある。

しかし我々は将来, 複数計算機で分散共有できるメモリと同等な機能を提供できるシステム²を目指している。そこで現状の黑板モデルに代わり, ユーザに複数計算機間で共有されたメモリの代替として仮想的にそう見える・そう使える仮想的な記憶システムを配列の形式で提供し, 簡単に分散計算が行なえるようなシステムを構想し, その基盤となるライブラリを既に開発した[2]。

我々はこのような配列を共有配列(共有データクラス[2])と呼び, その有効性と応用性を確かめるため, 本研究のシステムを試作した。

2 共有配列

共有配列とは各計算機に共有されて存在し, ユーザが自由に読み書きすることができる仮想的な次元配列を指す(図1)。この配列は各計算機間の配列の総和で構成されている事が望ましく, 各計

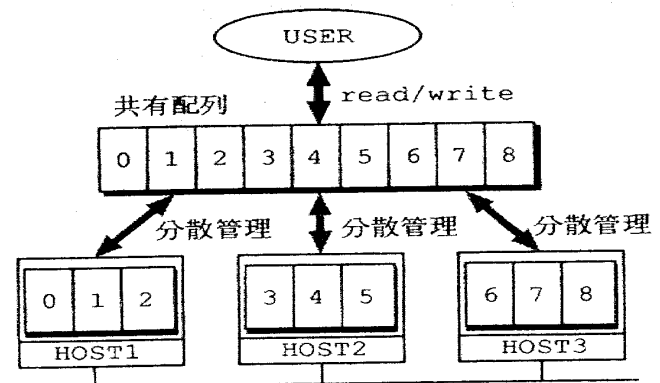


図1 共有配列の概念図

算機内部で通信を行ない, データに矛盾が起きないように管理されなければならない(図1)。ユーザは配列が分散されていることを意識する必要はなく, 計算のプログラムを書くことができる。

共有配列の実現は, 通常の配列の様にメモリ上に配列を作成するのではない。NFS(Network File System)など透過なファイルシステムを利用し, 各計算機で共有されているディスク領域に共有配列に相当する役割を果たせるファイルを作成することで簡単に実現できる。しかしNFSでは実行速度に問題がある。データに高速にアクセスできる分散共有メモリ[3]は未だ実用的ではない。

そこで分散された計算機毎にバイナリファイルを作成し, メモリ上にマップする³ことで高速な読み書きに対応し, 通信をしながらそのバイナリファイルを管理することで共有配列を実現した。

本システムでは異機種間のデータを交換する分散数値計算のような要求にも対応できるように設計された。ただしステップ毎の同期制御を実現するためには, ユーザがどの場所で同期を取りたいのかわからなくてはならない。そこでその同期制御を実現するためにライブラリを作成してユーザに提供し, ユーザ自身が数値計算プログラム中に埋め込めるようにした。また共有配列の提供や, 計算機間での通信を行なうためのライブラリも用意した。

¹Object-oriented, distributed computing system using the "Shared Array"

Masayuki Hatakeyama, Hitosi Uehara, Kenji Saito
Department of Computer and Information Sciences,
Ibaraki University, 4-12-1, Nakanarusawa, Hitachi-city,
316-8511 Japan.

²仮想的な DSM(Distributed Shared Memory)。

³mmap(UNIX System Call) を使用した。

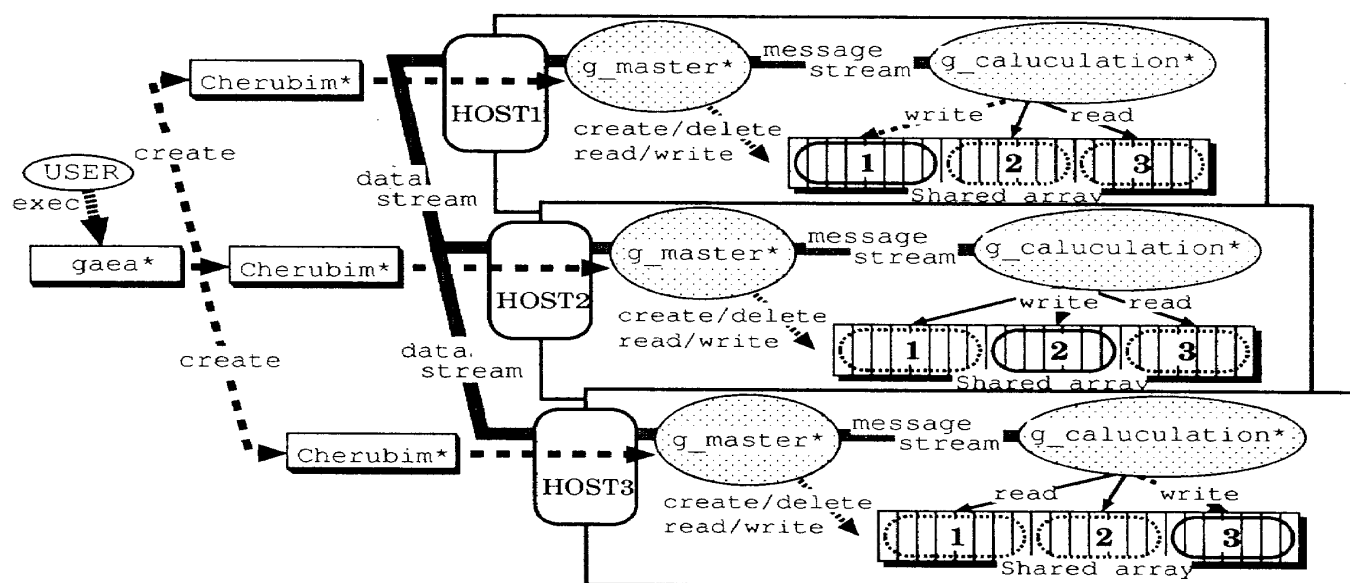


図2 システムの全体構成

3 システムの全体構成

3 台の計算機で構成されたシステムを図2に示す。各計算機毎に共有配列を生成し、それぞれの配列には計算機全てのデータが格納されている。

ユーザが書くべき計算のソースコードを g_calculation と表してある。その中でユーザは共有配列に計算したデータの書き込みを行ない、他の計算機で計算されたデータをこの共有配列から読み出すことでデータ交換が可能となる。この配列を管理しているのが g_master である。g_master 間では配列データの送受信を行なっている。ユーザはこの部分に手を加える必要はないが、ライブラリとして通信部もカスタマイズすることは可能になっている。

ユーザにはライブラリの数個のコマンドを g_calculation のソースコードに埋め込むことで、g_master と通信し、共有配列にアクセスできる。

4 実装と評価

実装はC++言語を使用し、ネットワークで接続されたUNIX環境内で行なった。(1+2)*(3+4)という簡単な計算を、共有配列を使用して分散させて行ないその動作を確認した。さらに使用条件や計算プログラム g_calculation をテスト的に種々変更して動作させた結果、巨大配列を必要とするような計算にはデータ通信時に時間がかかるので利用に注意が必要である事が判明した。しかし一方、ユーザはあくまで一元的な配列が存在することを意識してプログラムするだけで共有配列を使う記述ができるので、非常に簡単に分散計算のソース

コードを書くことが可能となることも分かった。

また全ての計算機に全てのデータを持つ配列が存在するので、分散環境内のある計算機が異常終了した場合にも計算を途中で諦めることなく計算を再開したり、データを安全にセーブできるといった対応をすることができる。

更にこの方式は分散計算を行なう際の計算領域の再分配も簡単に行なうことができ、これは動的負荷分散を実現するのに有効である。

5 実装中及び次の段階の計画

現在、本システム上で差分法の1種であるRoeスキームを用いた流体の数値計算を行なおうと実装中である。計算内に存在する各ノードが持つ大量の情報を、共有配列上で読み書きすることでノード間のデータ交換が可能となる。

参考文献

- [1] 直井 稔: エージェントを用いたオブジェクトベースシミュレーション仮想計算機システム, 平成8年度茨城大学大学院理工学研究科情報工学専攻修士学位論文。
- [2] 上原 均, 畠山正行: UNIXプロセスの分散オブジェクト化 —プロセスオブジェクトモデルの提案—, オブジェクト指向最前線, 情報処理学会・ソフトウェア工学研究会, 朝倉書店, pp.15-22, (1996)。
- [3] Tanenbaum, A.S.: *Distributed Operating Systems*, Prentice-Hall, Inc.(1995).(水野 ほか 訳: 分散オペレーティングシステム, プレンティスホール出版, (1996))。