大域的ゴミ集め機能を持った分散ヒープ*

4D－3

藤原克則　青柳龍也　有山正孝†‡
電気通信大学

1　はじめに

従来の実装においては自身の局所的ヒープ領域のみを対象としたゴミ集めのみで済んでいたものが、分散実装においては、分散されたシステム全体に渡った大域的ゴミ集めを行なわなければならない。

これらの局所的並びに大域的ゴミ集めに関しては、多くのアルゴリズムが提唱されているが、共通の枠組においてそれらを実装し、比較、評価するといったことは、あまり行なわれていない。

本研究では、局所的ならびに大域的ゴミ集めの各種アルゴリズムの実装において、共通の枠組となるような分散ヒープの設計、実装を行なう。

このヒープは、局所的ならびに大域的ゴミ集めに関して考案されている各種のアルゴリズム [1] が有している共通点に着目し、ヒープの構成要素、関連要素を抽象化してそのインタフェイスを統一することで、各種ゴミ集めアルゴリズムの容易な実装、ヒープ内へのゴミ集め機能の隠蔽およびヒープ外への統一された操作方法を提供しようというものである。

2　ゴミ集めのアルゴリズム

ここでは、各種の局所的並びに大域的ゴミ集めアルゴリズムの実装において、必要とされる共通の機能について考察する。

2.1　局所的ゴミ集め

以下、ヒープ領域の管理単位をセル（cell）、現コンテキストにおいて必要なセルを特定する処理系のレジスタや実行時スタックなどから成る集合を、ルートセット（root set）と呼ぶ。

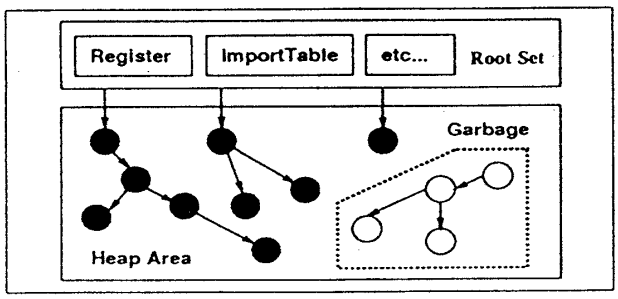

図 1: ヒープ領域摸式図

局所的なゴミ集めを行なう際に、外部からの統一的な操作をするために必要な機能は、以下のようなものである。

- ルートセットからそれが参照しているセルの情報を得る
- あるセルが、“生きた（live）”- もしくは “死んだ（dead）”- セルであることを通達する

また、ヒープ領域を実際には複数の記憶領域の集合の抽象概念ととらえるなら、それらの記憶領域間のゴミ集めの同期をとる機能も必要でる。

2.2　大域的ゴミ集め

処理系を分散実装することによりヒープ領域も分散されるため、遠隔参照（remote reference）が必要になる。

一般に、他のノード（node）からの遠隔参照および他のノードへの遠隔参照に関する情報を持つものをそれぞれ、インポートテーブル（import table）、イクスポートテーブル（export table）と呼ぶ。

以下、これら2つのテーブルにおいて個々の参照に関する情報を持つものをエントリ（entry）と呼ぶ。

インポートテーブルが参照しているセルは他のノードから参照されているため、局所ヒープ領域内で全く参照されていなくてもゴミとなることはない。このためインポートテーブルはルートセットに属する。

大域的ゴミ集めは、おおむね以下のような手順で行なわれる。

1. 局所的ゴミ集めを行なう
2. イクスポートテーブルのゴミ集めを行なう
   必要なら他のノードへゴミとなったエントリの情報を送る
3. 他のノードからのエントリ削除の情報にしたがってインポートテーブルのゴミ集めを行なう

以上のことから、大域的なゴミ集めを行なう際に外部からの統一的な操作をするためには、以下のような機能が必要である。

- あるエントリに対して、その” 生死” を通達する機能
- ノード間通信の統一的なインタフェイス
  大域的ゴミ集めに関してノード間通信で交わされるメッセージはアルゴリズム依存となるため、依存部分を隠蔽しつつ統一して取り扱うことのできる機構が必要とされる

3　分散ヒープの設計

実装には C++ を用いた。これは継承を用いることにより、外部に対するインタフェイスを容易に統一できるためである。

ヒープ領域は、以下のクラスを継承したクラスのオブジェクトにより構成される。

- class LocalRoot
  ヒープ領域に対する参照を取り出すためのインタフェイスを提供する抽象クラス

*Distributed heap with global garbage collection
†Katsunori FUJIWARA, Tatsuya AOYAGI, Masataka ARIYAMA
‡The University of Electro-Communications

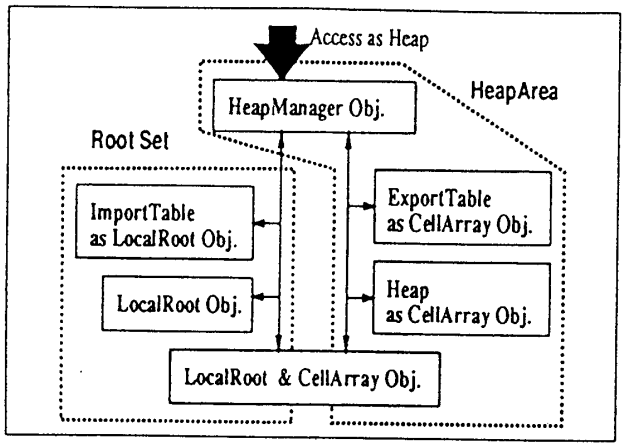

図 2: ヒープ構成例

- class CellArray
  ヒープ領域の一部ととしてアクセスならびにゴミ集めのインタフェイスを提供する抽象クラス

- class Heap
  class CellArray のサブクラスで、領域の割り当てなどのインタフェイスを提供する抽象クラス

- class ExportTable、class ImportTable
  それぞれ class CellArray、class LocalRoot のサブクラスで、遠隔参照のエントリに関するインタフェイスを提供する抽象クラス

- class HeapManager
  class LocalRoot、class CellArray のオブジェクトを管理し、外部からのヒープ領域へのアクセス、ならびにゴミ集めにおける各オブジェクト間の仲介をするクラス

class LocalRoot もしくは class CellArray を継承したクラスを作成することで、class HeapManager オブジェクトの管理下、相互に統一的なインタフェイスを通してゴミ集めをおこなうことができる。

class CellArray オブジェクトは、他の class CellArray オブジェクトを参照している生きた - もしくは死んだ - セルを持つ時、そのオブジェクトに対して参照しているセルの生死を通達することで相互にゴミ集めを進める。

局所的ゴミ集めにおける各 class CellArray オブジェクト（以下 CellArrayObj.）間の同期は、class HeapManager オブジェクト（以下 HeapManagerObj.）が現在進行中のゴミ集めを終了させるよう通達を受けて以降、以下のように進められる。

1. 管理下の CellArrayObj. にゴミ集め終了処理をさせる。各 CellArrayObj. はわかっている限りでセルの生死を識別する。ただしこの時点で回収処理は行なわれない。

2. まだ通達していない他の CellArrayObj. 内のセルへの参照は HeapManagerObj. を通して通達する。HeapManagerObj. は通達を受けた CellArrayObj. に対して、再び終了処理を行なわせる。

3. HeapManagerObj. は終了処理が必要な CellArrayObj. がなくなるまで以上の動作を繰り返す。

4. 各 CellArrayObj. にゴミ集め終了を通達する。この時点でゴミであるセルは各 CellArrayObj. によって回収される。

5. HeapManagerObj. は RootSet の参照しているセルが生きていることを各 CellArrayObj. に通達する。この時点から新たなゴミ集めが開始される。

大域的ゴミ集めは、ノード間通信で遠隔参照エントリ削除のメッセージを受信した時、インポートテーブルに対してエントリの削除を通達することで行なわれる。

以上の動作にともなう他のノードへのエントリ削除要求や、セルのイクスポートといったアルゴリズム依存部分は、class ImportTable、class ExportTable、メッセージ解析クラスのサブクラスを作成することで実装する。

実際のアルゴリズム実装の例として局所的ゴミ集めを [2] で行なうには、class Heap で定義されている領域の割り当て関数を、割当量に応じたセルのフォワードおよびスキャンを行なうように、また、class CellArray で定義されているアクセス用関数を、フォワードされたセルへのアクセスの整合性がとれるようにオーバロード (over load) することで実装できる。

4 おわりに

本研究において設計した分散ヒープをベースにすることで、現在までのところ [2][3][4] などの実装が、個別に実装を行なった時に比べて容易に行なうことができた。

今後、さらに多くのアルゴリズムの実装を行ない、今回の基本設計の妥当性ならびに問題点について明らかにするとともに、各種アルゴリズム自体の評価も行ない、その長所、短所を明確にすることで、大域的ゴミ集めに対する新しいアプローチを求めていこうと思う。

参考文献

[1] Abdullahi, S. E , Miranda, E. E & Ringwood, G. "Collection Schemes for Distributed Garbage." International Workshop IWMM 92 Proceeding, September 1992

[2] Baker, H. G "List Processing in Real Time on a Serial Computer." *Communications of the ACM*,Vol. 21, No. 4, 1978

[3] Baker, H.G "The Treadmill: Real-Time Garbage Collection Without Motion Sickness." *ACM SIGPLAN Notices*,Vol. 27, No. 3, 1992

[4] Shapiro, M. , Plainfosse, D. & Gruber, O. "A garbage detection protocol for a realistic distributed object-support system" Rapport de Recherche INRIA 1320, November 1990