

オブジェクト指向分散環境OZ++におけるガベージコレクション

1H-3

大西 雅夫*(東洋情報システム) 濱崎 陽一(電子技術総合研究所)

音川 英之*(シャープ) 塚本 享治(電子技術総合研究所)

* 情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」研究員

1 はじめに

オブジェクト指向分散処理環境OZ++では、OZ++言語によるクラスの記述に従って、オブジェクトの生成やオブジェクト間でのメソッド呼び出しが行なわれることにより計算が進む。OZ++言語システムはガベージコレクタを持つ。また、クラスの供給は、ネットワークサービスとして実現されており、オブジェクトの実行/交換メカニズムであるエグゼキュータはクラスの情報を必要に応じて取り寄せる。これらの情報は、効率を上げるため、エグゼキュータ内部にキャッシュされる。

本稿では、ガベージオブジェクトの回収とクラス情報のキャッシュの破棄とをガベージコレクションと総称し、エグゼキュータにおけるその実装を報告する。

2 OZ++の構成とメモリ管理

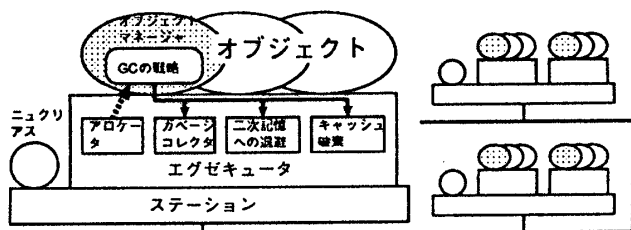


図 1: OZ++システムの構成とメモリ管理

図1に示した通り、OZ++は、ネットワークで接続された複数のステーション(計算機)上に実装される。エグゼキュータは、オブジェクトを実装し、メソッドのマルチスレッドによる実行やオブジェクト間の通信を行う。エグゼキュータ上にはオブジェクトマネージャ(以下OMと略記)と呼ばれる特殊なオブジェクトが存在しており、これが、エグゼキュータ上のオブジェクトを管理するためのポリシーを実装している[2]。

Garbage Collection of OZ++: An Object-Oriented Distributed Systems Environment

Masao Onishi* (Toyo Information Systems, Co., Ltd.),

Yoichi Hamazaki (Electrotechnical Laboratory),

Hideyuki Otokawa* (Sharp Corporation),

Michiharu Tsukamoto (Electrotechnical Laboratory)

*: Research Fellow of Open Fundamental Software Technology Project in Information-technology Promotion Agency, Japan

メモリ管理においても我々は、エグゼキュータとOMの役割分担を次のように明確にした。すなわち、エグゼキュータがメモリの再利用のための様々な基本メカニズムを提供するのに対し、OMは、それらの基本メカニズムを組み合わせて、必要なガベージコレクションを行なうための上位の戦略を実装するのである。このことにより、メモリ管理においても拡張性の高いシステムが実現できる。なお、本稿ではエグゼキュータが提供する下位のメカニズムについて述べる。

3 エグゼキュータのメモリ管理の方針

エグゼキュータ内でのメモリの消費には、エグゼキュータ本体のコードやデータ領域といった固定的なものや、スレッドスタック、通信バッファといった利用後ただちに解放されるもののほかに、次のものがある。

(1)オブジェクト: アプリケーションのモジュール性を高めるために、我々はガベージコレクタの存在を前提としてOZ++言語システムを設計した。従って、プログラムが実行されるに従って、ガベージが蓄積する。

(2)クラス情報のキャッシュ: オブジェクトの交換を効率良く行なうために、OZ++では、オブジェクトのメモリレイアウト、実行可能コードなどはネットワークサービスを介してオンデマンドで供給され、エグゼキュータ内部にキャッシュされる。

上の2項目に関し我々は、メモリ使用の上限値をユーザが指定することとした。その上限値を越えそうになった時点で、OZ++システムは、ガベージの回収やキャッシュの破棄を自動的に行なう。

4 ガベージコレクションの実装

以下ではエグゼキュータにおけるガベージコレクションの実装を説明する。また、オブジェクトのメモリ領域の有効利用に関連して、オブジェクトの二次記憶への一時退避のメカニズムにも触れる。

4.1 オブジェクトのメモリ領域の構造

OZ++におけるオブジェクトモデルを図2に示す。オブジェクトにはシステム全体でユニークなIDを持っていてどこからでもアクセス可能なグローバルオブジェクトと、グローバルオブジェクトの部品として使われるローカルオブジェクトとがある。一つのグローバルオブ

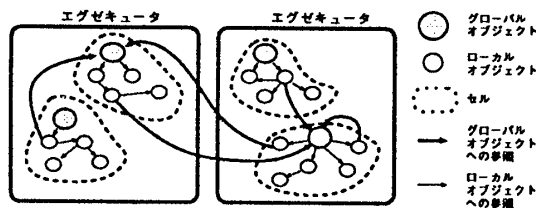


図 2: オブジェクトモデル

ジェクトを起点としてローカルオブジェクトへの参照の連鎖を辿って得られるオブジェクトの集合をセルと呼ぶ[1]。ローカルオブジェクトはセル間では共有され得ない。エグゼキュータ内のメモリ管理モジュールは、オブジェクトのためのメモリ領域を、セルを単位に分割して管理している。セルに対応する各々の領域をオブジェクトヒープと呼ぶ。回収の対象は、オブジェクトヒープ内で参照されないローカルオブジェクトである。グローバルオブジェクトへの参照を追跡して、ガベージとなったセルを自動回収することも考えられるが、OZ++ではグローバルオブジェクトに対する参照が広範囲に及び、このような回収は非効率的となるので採用しなかった。

4.2 ガベージオブジェクトの回収

OZ++は、計算機をまたがってオブジェクトの交換を行なうシステムである。そのため、高速なマーシャリング/アンマーシャリングを実現しやすいように、言語処理系は、ポインタの個数や位置に関する情報をオブジェクトに埋め込んでいる。従って、回収の際にもその情報を利用して、オブジェクトの参照関係の追跡を高速に実現できる。

他方、スタック中のデータについては実装の容易さを考慮して、OZ++言語処理系はガベージコレクタを支援しない方針を採っている。我々は、処理系の支援を必要とせず、プログラミングの煩雑さも招かない有効な手法として、スレッドスタックからポイントされるオブジェクトのマーキングに限り、保守的回収[3]を用いた。これは、対象領域をアドレスポインタの羅列であると仮定して、疑わしきはマークする、というアプローチである。従って、この手法でマークされたオブジェクトを移動あるいは複写することは困難となる。マークするかどうかの判定を高速化するために、オブジェクトヒープはページサイズの整数倍のチャンクから成り、各チャンクには同一サイズのオブジェクトのみが収容されている。このような、ふたつの異なる手法の融合が我々のコレクタの特徴のひとつである。

また、我々のコレクタはstop-and-collect方式であるが、エグゼキュータは、回収の対象となったグローバルオブジェクトのスレッド(デーモンやメソッド実行)の

みをサスペンドする。従って、他のオブジェクトは、回収期間中も並行動作が可能である。

4.3 クラス情報のキャッシュの破棄

オブジェクトのメモリ割り付けやメソッド実行など、クラス情報のキャッシュを利用する際には、キャッシュ管理モジュールが介在してキャッシュの参照カウンタをメンテナンスする。エグゼキュータはこのカウンタが0のキャッシュを破棄するメカニズムをOMに対して提供している。

4.4 オブジェクトの二次記憶への退避

エグゼキュータはセルを単位としてオブジェクトを二次記憶に一時的に退避させる機能を持つ。エグゼキュータからの要求に従って、OMがメモリの再利用を行う際、ガベージコレクションの他にこの機能も利用できる。この機能の実行は対象オブジェクトのスレッドが存在しない場合に限られるので、セルを主記憶に復帰させる際には最大限の回収/圧縮が実現できる。

5 まとめ

エグゼキュータはメモリの再利用のための様々なメカニズムをOMに提供している。OZ++のコレクタでは、処理系がオブジェクト内部に埋め込むポインタに関する情報を利用する方法と、そのような情報なしに行なわれる保守的回収の手法とが融合されている。また、セルという単位が存在するために、ガベージの回収とオブジェクトの実行とを並行して行なうことが可能である。

現在エグゼキュータに本稿で述べたメカニズムを組み込み終えたところである。今後、本格的な稼働試験を行ない、並行性や回収率、性能の向上を検討する。

本研究は、情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。

参考文献

- [1] 塚本他: 「オブジェクト指向分散環境OZ++の基本設計」、SWoPP93, Aug. 1993
- [2] 大西他: 「オブジェクト指向分散環境OZ++のオブジェクトマネージャの設計」、情報処理学会第48回全国大会, Mar. 1994
- [3] H.-J. Boehm, "Garbage Collection in an Uncooperative Environment", Software — Practice and Experience, Vol. 18(9), 807-820 (Sept. 1988)