

並列言語 OPA における一貫性制御に対応した差分プログラミング

3Z-7

馬谷 誠二, 湯浅 太一, 八杉 昌宏, 小宮 常康

京都大学大学院情報学研究科

通信情報システム専攻

1. はじめに

OPA[1]は、Java言語に似たシンタックスに加え、構造化された並列構文を持ったオブジェクト指向並列言語である。

OPAでは、オブジェクトの一貫性制御のために、オブジェクトの多重状態モデルを採用することで、これらの排他制御を行っている。オブジェクトの多重状態モデルとは、メソッドが呼び出される度に、メソッドの処理用にオブジェクトのコピーが作成されるモデルのことである。

Java言語にはクラス継承機能の一つとしてsuperメソッド呼出しによる差分プログラミングがあるが、OPAでもこの機能を利用できるようにしたい。しかし、メソッド単位のきめ細かな排他制御方式により、superメソッドをそのまま呼び出すことは許されない。本研究ではOPAにおけるオブジェクトの一貫性制御に対応したsuperメソッド呼出し機構を実現する。

2. OPAにおけるデータの一貫性制御

OPAでは、instantメソッドというメソッドを用いてオブジェクトの一貫性制御のための排他制御を最低限に抑える。instantメソッドを、インスタンス変数に対して行う読み書きを解析することにより読み出し専用(RO)メソッドと読み書き両用(RW)メソッドに分類し、また、instantメソッドによるオブジェクトに関する読み出しや書き込みをそれぞれメソッド開始点や更新の可能性が消失する点で一括して行う。これにより、ROメソッドの並列実行を許すとともに、ROメソッドのnon-blockingな実行が可能となる。

オブジェクトに対しては、次のような意味において一貫性を保持する。

- ROメソッドがアクセスするインスタンス変数の状態は、当該ROメソッド呼出しの開始以前に一括更新を完了したRWメソッド呼出しの結果が反映されているものとし、当該ROメソッド呼出しが完了するまでの間、他のメソッド呼出しの影響は受けず一定とする。
- RWメソッドがアクセスするインスタンス変数の状態は、当該RWメソッド呼出しの開始以前に開始されたRWメソッド呼出しの一括更新結果が反映されているものを初期値とし、当該RWメソッド呼出しが完了するまでの間、他のメソッド呼出しの影響は受けず、当該RWメソッドによる代入によってのみ変化する。

このinstantメソッドの動作は、メソッド起動の度に各メソッドの関数フレーム内にオブジェクト本体のコピーを作成し(RWメソッドについては)一括更新点において変更された可能性のあるデータを本体へ書き戻すことにより実現されている。

3. 差分プログラミング

superによる差分プログラミングとは、あるクラスのメソッド定義をその親クラスのメソッドを再利用してその差分のみを記述して行うものである。Java言語でもともとサポートされており、「super. メソッド名(...)」でオブジェクトの実際のクラスに関係なく、つまり遅延束縛せずに親クラスで定義されているメソッドを、直接選択して起動する。

OPA では、キーワード instant を用いた排他制御がメソッド起動のたびに行われ、親クラスのメソッドを super で呼び出す際にもその機構が働くことにより、super メソッドを呼び出す側と呼び出される側の両方が RW メソッドであればデッドロックがおこってしまう。

この問題に限定的であれ対処するために、super による差分プログラミングに限り他の RW メソッドを使って RW メソッドを定義することを次のようにして許すこととした。

super では遅延束縛せずに呼び出されるため、実行するメソッドをコンパイル時に確定でき、インライン展開したコードを生成することも可能である。そこで、「super. メソッド名 (...)」で確定されるメソッドと、それを利用するメソッドの両方が RW メソッドであってもよい。すなわち、再利用される親クラスのメソッドの実行を含めた子クラスのメソッドの実行を単位として、オブジェクト本体への一括アクセスや相互排除などを実現すればよい。

4. 実装

OPA では各 instant メソッドに対し、実際にコピーしたり書き戻したりするデータを最低限にとどめるため、読み書きするインスタンス変数の集合の解析を行っているが、本体で super 呼出しを行っているメソッド (super 呼出し側メソッドと呼ぶ) については、super で呼び出される側のメソッド (親クラスのメソッドと呼ぶ) が読み書きするインスタンス変数も解析結果に加えなおすことで、親クラスのメソッドで用いられるオブジェクトのデータは、super 呼出し側メソッドでも用いられるものとして扱われることにする。そうしておけば、親クラスのメソッドはオブジェクトのデータを本体から直接複製するのではなく、super 呼出し側メソッドがすでに持っているコピーから複製すればよい。親クラスのメソッドは排他制御の必要なしに呼ばれ、一括更新は super 呼出し側メソッドのコピーに対して行われる。

5.まとめ

プログラミング言語 OPA では、オブジェクトの一貫性制御のためにきめ細かな排他制御方式を採用することで並列処理の高速化を図っている。

本研究では super による差分プログラミングを OPA 上で実現するにあたり、別のメソッド呼出し機構を用意することで対処した。

6. 謝辞

最後に、本論文の作成にあたりいろいろとお世話になりました西村祥治氏をはじめとする湯浅研究室の方々に深く感謝いたします。

参考文献

- [1] 八杉 昌宏, 瀧 和夫: 並列処理のためのオブジェクト指向言語 OPA の設計とその実装, 情報処理学会研究報告, Vol.96, No.82, pp.157-162 (1996).