並列Schemeにおける即時タスク生成法と遅延タスク生成法の融合

3Z－6

窪田 貴志，小宮 常康，湯淺 太一，八杉 昌宏
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

並列Scheme[1]におけるタスク生成アルゴリズムとして，即時タスク生成法と遅延タスク生成法[2，3]が知られている．即時タスク生成法は，並列化のための式が評価された時点で必ず新たなタスクを生成するため，並行処理(concurrent processing)を行う場合に向いている．一方，遅延タスク生成法は，並列化のための式が評価された時点における継続を生成する．他のプロセッサがアイドル状態になった場合には，その継続を盗みに行き，並列に処理が行われる．よって，余分なタスク生成のオーバヘッドを抑えることができる．これら二つのアルゴリズムには，用途，適正に違いがあるにもかかわらず，既存のScheme処理系ではどちらか一方を実装したものしか見られなかった．本研究では，両者を組み合わせた形のタスク生成機構を提案し，Javaスレッドによる実装を行った．本稿ではその概要を述べる．

2．タスク生成機構の提案

並列Schemeにおけるタスク生成には一般的にfuture式が使用される．future式を多用するプログラムを即時タスク生成法によって実行すると，タスクの生成，切り換えのオーバヘッドが性能低下の要因となることがある．このオーバヘッドを抑制できるという点において，遅延タスク生成法は優れている．しかし，並行処理を目的とするようなタスク生成を行う場合，遅延タスク生成法では不都合な点が指摘できる．

一例として，アニメーションを並列計算によって処理しつつ，その途中結果を表示するためのタスクを生成する場合を考える．遅延タスク生成法を用いると，最初に一定数のプロセッサがすべて描画イメージの計算に割り当てられ，描画を行うための継続がすぐには実行されないという状況が起こり得る．また，もう一つの例として，ある計算を実行するタスクを生成し，同時にその計算を停止するための入力待ちを行うような場合を考える．future式を評価したプロセッサは，入力待ちの処理を行うための継続を生成して，計算の処理に移る．ここでアイドル状態となるプロセッサが現れない場合，継続はすぐには盗まれない．そのため，ユーザ入力があっても計算を停止するという処理を行うことができない．

本稿で提案するタスク生成法では，2つのタスク生成アルゴリズムのための並列記述式をそれぞれ用意する．OSの提供するスレッドを使用することを前提とし，プロセッサへのスレッド割当てはOSが行うものとする．遅延タスク生成法によるタスク生成には従来通りfuture式

```
(future 《式》)
```

を用いる．一方，即時タスク生成法によるタスク生成には，新たにspawn式

```
(spawn 《式》)
```

を導入する．spawn式を評価すると，処理系は直ちにspawnオブジェクトと呼ばれるオブジェクトを生成する．そして，spawnオブジェクトに対して，即座にスレッドを生成する．《式》の計算途中でfuture式を使用する場合は

```
(spawn 《スレッド数》 《式》)
```

という形でスレッド数を指定して，spawnオブジェクトに対して複数のスレッドを割り当てることができる．また，spawnオブジェクトという単位にタスクをグループ化することによって，spawnオブジェクト毎に優先度を定義することが可能になる．よって，遅延タスク生成法だけでは実現が難しいような優先度を利用したスケジューリングも，極めて容易に行うことができる．上で述べた例題を解決している様子を図1に示す．

Integration of Eager Task Creation Method and Lazy Task Creation Method for Parallel Scheme
Takashi Kubota, Tsuneyasu Komiya, Taiichi Yuasa, Masahiro Yasugi
Department of Communications and Computer Engineering,
Graduate School of Infomatics, Kyoto University
kubotan@kuis.kyoto-u.ac.jp

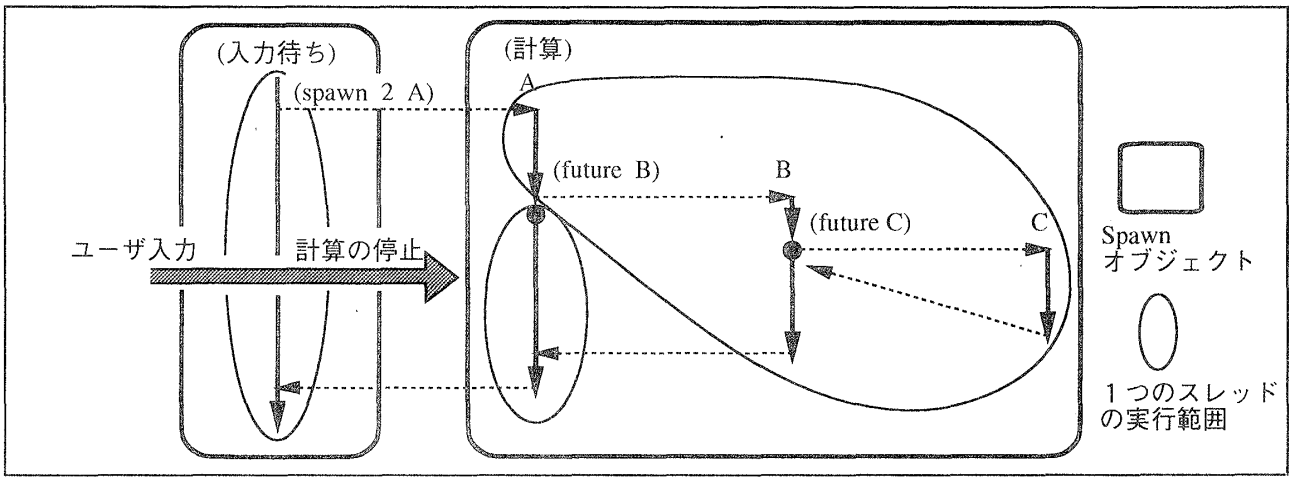

図1 ： spawn式とfuture式を組み合わせたタスク生成

3．Java上のScheme処理系「ぶぶ」への実装

本稿で提案したタスク生成機構は，京都大学が開発したScheme処理系「ぶぶ」[4] 上に実装した.「ぶぶ」はJava言語で記述されており，Java Virtual Machine上で動作する．従来のScheme処理系の機能に加えてJavaをベースとしたオブジェクト指向機能 [5] が拡張されている．具体的には，Javaのクラスライブラリをロードして実行することが可能で，さらにロードしたクラスを継承してSchemeのクラスとして定義することも可能である．これによって，Schemeの上からJavaが提供するGUIを利用したり，分散環境を利用することが可能である．

「ぶぶ」におけるプロセッサ，すなわちバイトコードインタプリタは，java.lang.Threadクラスのサブクラスとして実現されている．spawn式の処理として，まずspawnオブジェクトに対してjava.lang.ThreadGroupクラスを定義する．そして，そのスレッドグループに属するプロセッサのインスタンスを指定数だけ生成する．spawnオブジェクトに属するプロセッサは，Javaスレッドのスレッドグループの機能を利用して，優先度の変更，一時停止，再開，終了といったメソッドを呼び出すことで，一括に操作できる．よって，それらのメソッドを呼び出すための組み込み関数を用意する．遅延タスク生成法は，TUTScheme[1, 3] の実装を参考にして「ぶぶ」上に移植した．

4．まとめ

本稿では，並列Scheme上での，即時タスク生成法と遅延タスク生成法を組み合わせた形のタスク生成機構について提案し，Javaスレッドによる実装法について述べた．

これによって，プログラマは，spawn式とfuture式のどちらを使用すればよいか，及び，spawn式を使用する場合はspawnオブジェクトに対していくつのスレッドを割り当てるかという2点にのみ注意すれば，一方のタスク生成アルゴリズムのみを用いた場合に生じる問題点を解決することができる．

参考文献

[1] 小宮常康: Schemeの機能拡張に関する研究，博士論文，豊橋技術科学大学 (1992).

[2] E.Mohr, D.Kranz and R.Halstead: Lazy Task Creation: A Technique for Increasing the Granuatiry of Parallen Programs, *IEEE Trans. Parallel and Distributed Systems*, Vol. 2, No. 3, pp. 264-280 (1991).

[3] 高橋雅彦: 並列Schemeにおける遅延タスク生成法とその実現，特別研究報告書，京都大学工学部情報学科 (1997).

[4] 山中政宣: Java上のScheme処理系「ぶぶ」の実装，特別研究報告書，京都大学工学部情報学科 (1997).

[5] 木俣功: Javaをベースとしたオブジェクト指向拡張Scheme，修士論文，京都大学大学院工学研究科 (1999).