

並列論理型OS - PIMOS (2)

開発支援KL1処理系

4D-4

宮崎 敏彦

木村 康則

(財) 新世代コンピュータ技術開発機構

1. はじめに

我々は並列推論マシンの開発と並行して、そのOSであるPIMOS(1),(2)の開発を行うために、汎用計算機上にPDS S (PIMOS Development Support System) と呼ぶKL1システムを開発中である。PDS SはC言語で記述されたバイト・コード・インタプリタ部とKL1自身で記述されたランタイム・サポート・ルーティン及びインタプリタ等からなるシステムである。本論文ではこのうちPDS SのOS記述サポート機能を中心に述べる。

KL1には、その母体言語であるFlatGHC(4)に、OSの記述を支援する為の幾つかの機能が付加されている。そのうちで最も重要な機能として荘園(3)が上げられる。荘園とは言語が規定している資源管理及び例外処理の最小単位である。荘園にはコントロール・ストリームとレポート・ストリームと呼ぶ2本のストリームが接続されている。コントロール・ストリームは荘園を制御するためのストリームであり、後述する種々のコマンドを流すことができる。レポート・ストリームは例外情報など荘園内からの情報/要請が流れ出てくるストリームである。荘園のユーザはこのレポート・ストリームの各種情報を解釈するプログラムを記述することによって例外事項等を適当に操作することができる。PIMOSではこのような荘園の機能を用いてタスク管理等を記述するわけである。

2. 資源管理

PDS Sで管理している資源とは、ゴールのリダクション数である。(これは実行時間やメモリ消費量の非常におおまかな近似と考えれば良い。Multi-PSI や PIM では実行時間やメモリの消費量をちゃんと管理すると思われる。) 全てのゴールは必ず何れかの荘園に属しており、各ゴールのリダクションはそのゴールが属している荘園の下で管理される。荘園にはその荘園内で実行できるリダクションの最大許容量を与え、これによって資源の使い過ぎを抑えることができる。またこの最大許容量とは別に各荘園にはその荘園がとりあえず消費してよい割り当て量が与えられている。荘園がネストした場合、親荘園は必要に応じて自分自身の割り当て量を子荘園達への割り当て量として少しずつ分け与えなければならない。

ゴールの実行中にこの割り当て量を消費してしまった場

合は、自動的に、システムで規定している分割量が実行中の荘園の親荘園が持つ割り当て量(の残り)から引かれ、不足した荘園に与えられる。このとき、もし与えた割り当て量の累計が当該荘園の最大許容量を超えるようであれば、新たな割り当ては行われず、「資源の不足」という例外情報が当該荘園のレポート・ストリームに流され、その荘園(及びゴール)は中断状態になる。また、分割要求を受けた親荘園の割り当て量が分割量より少ない場合はそのまた親の荘園に同じ要求を出す。もし、どの祖先荘園も割り当てることができなかった場合は、許容量を超えた場合と同様、「資源の不足」例外をレポート・ストリームに流し荘園は中断状態になる。

「資源の不足」例外がレポート・ストリームから流れ出て来た場合の処理はソフトで記述されなければならない。例えば、実行をそのまま放棄したい場合は abort 命令をレポート・ストリームに流せば良いし、新たな資源を割り当ててその荘園の実行を継続したい場合はコントロール・ストリームに次のような命令を流せばよい。

```
more_reduction(ReductionNumber)
```

処理系はこの命令を受け取ると、ReductionNumber で与えられた数だけ最大許容リダクション数を増やす。

3. プライオリティ管理

荘園のもう一つの機能としてゴールのプライオリティ管理がある。各荘園レコードにはその荘園内で実行可能なゴールのプライオリティの上限と下限がセットされておりこれを超えるプライオリティでゴールを実行することはできない。荘園にセットされる上限値と下限値は、コントロール・ストリームに次のようなコマンドを流すことにより決定される。

```
priority(Min, Max)
```

ここで Min 及び Max はこの荘園の親荘園の下限値とこの荘園を生成したゴールのプライオリティとの差に対する割合である。すなわち、荘園の下限値は親荘園の下限値に Min で計算される値を加えた値、上限値は生成したゴールのプライオリティから Max で計算される値を引いた値となる。

PIMOS Development Support System

T.Miyazaki and Y.Kimura

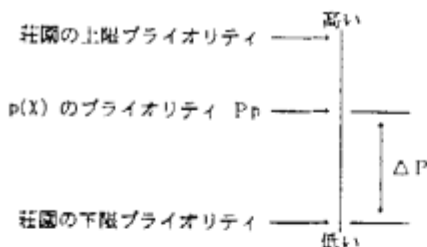
Institute for New Generation Computer Technology

KL1にはゴールのプライオリティをゴールごとに指定できる機能がある。プライオリティには論理プライオリティと物理プライオリティがあり、ゴールはそれぞれ自身の論理プライオリティを持つ。(以下単にプライオリティと言った場合は論理プライオリティを意味する。)処理系内にはあらかじめ指定されたレベルの物理プライオリティがあり、スケジューラはゴールをゴール・キューに繋げる際論理プライオリティを物理プライオリティに変換する。

(物理プライオリティは論理プライオリティよりも精度が悪いので、ユーザはプライオリティを変えたからといって必ず実際のスケジュールがその通りになることを期待してはいけなない。)

荘園が持つプライオリティの上/下限値も論理プライオリティである。ユーザはゴールのプライオリティを指定する場合その親ゴールのプライオリティとそのゴールが所属する荘園の上/下限値との相対値で指定する。具体的には、あるゴールを親ゴールよりも高い/低いプライオリティに設定したい場合は、親ゴールのプライオリティと荘園の上/下限プライオリティとの差の何割を親ゴールのプライオリティに加える/差し引くか指定するわけである。例を示そう。

```
親ゴール: p(X).
クローズ: p(A) :- true |
            q(A,B), r(B)@priority(N).
```



上記のプログラムでは、ボディ部のゴール $q(A,B)$ は親ゴール $p(X)$ と同じプライオリティで実行され、 $r(B)$ はプログラムによって計算されるプライオリティで実行される。簡単のため、与えられた値 N が負の整数の場合を考える。ここで、親ゴール $p(X)$ のプライオリティを Pp とし、荘園の下限プライオリティと Pp の差を ΔP とすると、プログラムとして与えられた N は ΔP に対する割合であり、 $r(B)$ のプライオリティは Pp からその割合分を引いた値になる。

4. デバイス・ストリーム

PIMOSではあらゆる機能を、論理の枠組みからはずれないように実現するよう努力している。これはKL1のような並列論理型言語でOSを記述する場合非常に自然でありかつ必要なアプローチであろう。このためPDSで入出力デバイス等をKL1におけるプロセスとして捉え、

そのプロセスとユニフィケーションによるメッセージ交換をすることによってそれぞれのデバイスの機能を利用できるようにになっている。デバイス・ストリームとはこのプロセスに繋がるストリームのことである。PDSで現在実現されているデバイス・ストリームには以下のものがある。

(1) ターミナル

ディスプレイとキーボードを意味する。このストリームには文字や行単位の入出力、トークンの入力、タームの出力命令等を流すことができる。

(2) ファイル

指定した名前を持つファイルへのストリームを返すオープン命令と、ファイルを閉じるクローズ命令を流すことができる。取ってこられたファイルへのストリームにはターミナルに流すことができるコマンドと同様のコマンドを流すことができる。

(3) コード領域

主に実行コードを管理するデバイスであり、コードの生成(アセンブル)、セーブ、ロード、コード形式の変更等のコマンドを流すことができる。コードの生成をこのようにデバイス・ストリームとして実現することにより処理系やOSで使う特権組込み言語の隠蔽を容易に行うことができる。

5. その他の機能

その他の主な機能として、例外処理機能とデッドロックの検出機能がある。KL1では、ユニフィケーションの失敗も例外として扱われておりデバッグ等の記述に見えている。また荘園内で起きた部分的なデッドロックもGCを起動することによって検出できる。これによって、PIMOSの下で実行されるユーザ・タスクのデッドロックを検出し、ユーザに例外情報として提供することが出来るようになっている。

謝辞

ICOT Multi-PS1(V2) 処理方式検討会のメンバには多くの重要な助言をいただいた。特に第四研究室近山氏、中島氏、三菱電気 中島氏からは多くの実現方式案を提案していただいた。ここに感謝する。また富士通S.S.L.の平野氏、中越氏には処理系の作成を手伝って頂いた。

参考文献

- (1) 佐藤他: 並列論理型OS-PIMOS(1)-資源管理方式-, 本大会予稿集 4D-3
- (2) 佐藤他: PIMOSの概要-並列推論マシン用オペレーティング・システムの構築-, 本学会第34回全国大会予稿集 2P-8, 1987
- (3) 近山, 瀧他: PIMOS第1版概念仕様書, ICOT TM-290, 1987
- (4) 上田: Guarded Horn Clauses, LPC'85, pp.285 1986-6