核言語ＫＬ１分散実行方式*
=ソフトウェアシミュレータによる評価=

6V-2

大原 有理[1] 鳥居 悟[1] 小野 越夫[1] 神田 陽治[2]
富士通㈱[1] 富士通国際研[2]

はじめに

第五世代コンピュータプロジェクトにおいて，核言語第一版（ＫＬ１）の中核をなす部分として，並列論理型言語ＧＨＣ（Guarded Horn Clauses）〔上田 85a〕が採用されている．

我々は，このＧＨＣを簡単化したFlatＧＨＣを分散環境下で実行する分散処理系ソフトウェアシミュレータを〔上田 85b〕をもとに逐次計算機上に作成した．

本論文では，我々が作成したソフトウェアシミュレータについて述べ，このシミュレータを用いて行ったプラグマ〔Shapiro 84〕付ＦＧＨＣプログラムのシミュレーション結果を示し，その結果をもとにプラグマによる分散方式の評価を行う．

1. 分散処理系ソフトウェアシミュレータ

ＫＬ１分散処理系ソフトウェアシミュレータにおいては，複数のプロセッサ（ＰＥ）の（擬似）並列実行と，各ＰＥ内のゴールの逐次実行の両方をシミュレートしなければならない．

ＰＥ内のゴールの逐次実行は，既に上田処理系〔上田 85b〕に実現されている．一方，ＰＥの（擬似）並列実行のシミュレーションは，ＰＥの逐次実行に他ならない．そこで我々は，上田処理系を二階建てにした方式を採用した．上田方式は，キューの先頭の要素を取り出して実行を行い，あるタイミングでその実行を中止し，キューの最後部にキューイングし直す，そして，次の要素を再び取り出して実行するものである．我々の方式は，上田方式を二段構造に拡張したものである．一段目はＰＥの実行を表し，二段目はその各ＰＥ内でのゴールの実行を表す．これを詳細に示したのが図１である．

このように，本ソフトウェアシミュレータは，上田方式におけるキューを二本管理しており，いくつかのサンプルプログラムにおいて，上田処理系と同等の実行性能を得ることができた．

2. シミュレーション

今回作成したソフトウェアシミュレータを用いて，本論文の最後に示したＱｕｅｅｎプログラムを対象に入力データが４～６Ｑｕｅｅｎ，ＲＣが100 と10のときの逐次実行とプラグマによる二種類の分散実行（分散方式①，②）とのシミュレーションを行った．ここでＲＣとは，ＰＥのスイッチを行うタイミングを示すもので，ＲＣが100 とは，各ＰＥで100 回のゴールリダクションが行われると，ＰＥのスイッチが行われることを表す．

また，入力データを固定し，設置ＰＥ台数を変えたシミュレーションを行った．

＊本研究は，第五世代コンピュータプロジェクトの一環として行われたものである．

Distributed Execution of the KL1
- Evaluation with the Software Simulator -
1 Yuri OHARA, Satoru TORII, Etsuo ONO FUJITSU LIMITED
2 Youji KONDA IIAS-SIS, FUJITSU LIMITED

図２，３は，それぞれＲＣが100 と10のときの入力データの大きさとサイクル数の関係を表す．ここで分散方式①では，使用するＰＥが４台に固定される．それに対して分散方式②では，設置ＰＥ台数によって，使用するＰＥ台数が変わる．また，サイクル数とは，実行が終わるまでのシステムキューの周回数を示す．

表１は，入力データを５Ｑｕｅｅｎ，ＲＣを10，分散方式を②とし，設置ＰＥ台数を変えてシミュレーションを行ったときのサイクル数と使用ＰＥ台数を示したものである．

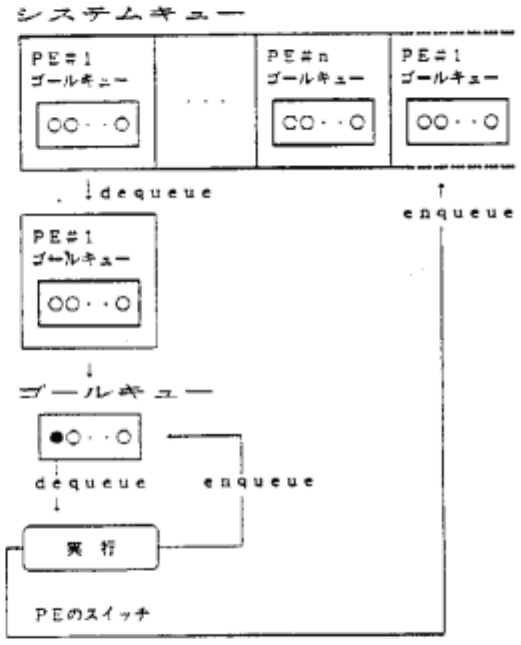

図１ ソフトウェアシミュレータの構成

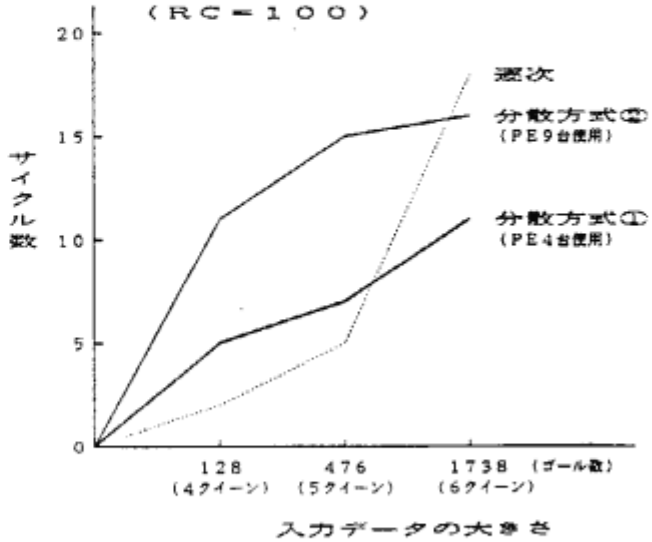

図２ 入力データの大きさとサイクル数の関係

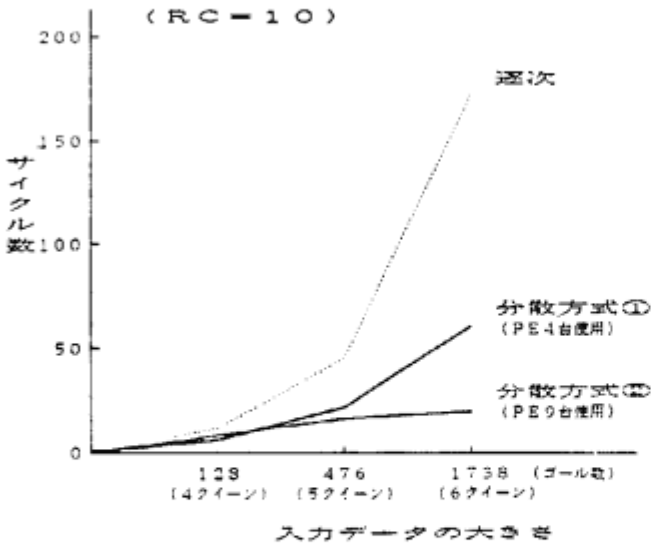

図３　入力データの大きさとサイクル数の関係

表１　設置ＰＥ台数を変えたシミュレーション結果

設置ＰＥ台数	４台	９台	１６台	２５台	100 台
サイクル数	２２	１９	１８	１８	１８
使用ＰＥ台数	４台	９台	１６台	１９台	２４台

3.　入力データと分散効果

図２より，入力データが小さいときは，ＰＥ間通信のため，分散実行の方が逐次実行よりもサイクル数が多いが，入力データが大きくなると，逆転している．また，使用ＰＥ台数が４台に固定される分散方式①とＰＥ台数が固定されない分散方式②とを比較すると，今回の結果では分散方式②の方がサイクル数が多いが，さらに入力データが大きくなると，分散方式①と②は逆転すると予想される．図３では，分散方式①，②ともに逐次実行よりサイクル数が少ない．また，使用ＰＥ台数が多い分散方式②の方が分散方式①よりもサイクル数が少ない．つまり，入力データが大きくなるに伴い分散の効果は顕著になり，さらに，多くのＰＥを使用した方が分散の効果は高められる．

4.　ＲＣと分散方式

ＲＣを各ＰＥの処理能力と考える．図２のＲＣが100 の場合には，各ＰＥの処理能力が高いので，ＰＥを４台しか使わない分散方式①の方が分散の効果が高い．一方，図３のＲＣが10の場合には，ＰＥを９台使用する分散方式②の方が各ＰＥへの負荷集中が少ないので，分散の効果が高い．つまり，ＰＥの処理能力によって，効果のある分散方式は異なる．

5.　設置ＰＥ台数と分散効果

表１より，設置ＰＥ台数を増やすと使用ＰＥ台数も増え，分散の効果は高まる．しかし，設置ＰＥ台数を１６台以上に増やしてもそれ以上の効果はない．これは，Shapiro 流のプラグマではＰＥ台数の効果をうまく利用できない例である．

おわりに

今回，ＫＬ１分散処理ソフトウェアシミュレータを用いて，プラグマ付ＦＧＨＣプログラムのシミュレーションを行い，分散方式についての評価を行った．

その結果，上田処理系を二段構造に拡張することによって，大きな例題による分散シミュレーションが可能になった．また，今回のシミュレーションでは，Shapiro 流のプラグマによる分散実行の効果が認められた．しかし，ＰＥ台数の効果がうまく利用できないケースもあり，Shapiro 流のプラグマが常に良い性能を引き出すわけではない．入力データの大きさやＰＥの処理能力によって，効果のある分散方式が異なる．これらの点を考慮して，より大きな入力データや多くのＰＥ台数のもとで効果のあるプラグマ方式に改良していく必要がある．

参考文献

〔上田 85a〕　Ueda,K. "Guarded Horn Clauses"
ICOT Technical Report TR-103.

〔上田 85b〕　Ueda,K. and Chikayama,T.: Concurrent
Prolog Compiler on top of Prolog,
In Proc. of Symp. on Logic Prolog.,
1985. pp.119-126.

〔Shapiro 84〕 Shapiro,E.Y. SYSTOLIC PROGRAMMING
in FGCS,1984

サンプルプログラム

```
queen(A,I,Q) :- true | gen(A,A,I,nil,nil,nil,Q,nil).

gen(N,A,IL,IR,PR,PM,Q,QT) :- N=$=0,P1:=N+A,M1:=N-A |
  test(N,IL,IR,A,PR,P1,M1,PM,PM,Q,QS),
  gen(N,IL,A,IR,PR,PM,QS,QT).    %分散２
gen(N,nil,__, __, __,Q,QT) :- N=$=0 | Q=QT.
gen(0,__, __,R, __,Q,QT) :- true | Q=R,QT.

test(N, __, __, __,P,M,(P1,M1),PML, __,Q,QT) :-
  P1=:=P | Q=QT.
test(N, __, __, __,P,M,(P1,M1),PML, __,Q,QT) :-
  M1=:=M | Q=QT.
test(N,IL,IR,PR,P,M,(P1,M1),PML,PM,Q,QT) :-
  P1=$=P, M1=$=M | test(N,IL,IR,PR,P,M,PML,PM,Q,QT).
test(N,IL,IR,PR,P,M,nil,PM,Q,QT) :- N1:=N-1 |
  app(IL,IR,IN),
  gen(N1,IN,nil,PR,(P,M),PM,Q,QT).   %分散１　分散２

app(A,X,Y,Z) :- true | Z=A,W, app(X,Y,W).
app(nil,Y,Z) :- true | Y=Z.
```

分散１は，分散方式①でプラグマを付加したゴールを示す．
分散２は，分散方式②でプラグマを付加したゴールを示す．