5G-7

関数を future で評価する言語

柳瀬 龍郎† Yanase Tatsurou‡
田村 信介† Tamura Sinnsuke‡
谷口 秀次† Taniguchi Syuji‡
福井大学工学部† Univ. of Fukui‡

まえがき　この報告は，並列処理を容易におこなう言語の提案である．厳密には C 言語プログラムの記述において，並列処理を簡便に記述するための，システムの提案であり，そのためのプリプロセッサの提案でもある．

処理の(部分処理への)分割の戦略と作り出された部分処理の分散実行のための手段についての報告．とくに分散処理の負荷分散について古くから研究がなされている

本研究では多重呼び出し可能な“future”による並列処理を提案する．

MULTI-LISP において提案された[1]future 概念による関数評価の方法は，近年のマルチコアプロセッサには最も適切な並列処理記述でかつ，処理システムと考えられる．

future による並列処理の戦略　全体の処理はマスターワーカシステムとして動作する．

計算モデルは
１）マルチコア
２）SMP
３）ネットワーク結合型
のいずれも計算モデルとして想定可能である．

この future による並列処理のアーキテクチャは，プログラムのディレクティブとして future 処理を宣言された関数に限って並列処理を行う対象とする．

プログラムにおいて定義され future 宣言された関数の評価を，呼び出し元の計算資源（仮想プロセッサ：マスタ）とは別の計算資源（仮想プロセッサ：ワーカ）においてマスタにおける実行文と並列に評価処理をおこなう．この呼び出されたワーカにおいて，再度関数の並列評価を別のワーカに依頼することを許す．ワーカが抱え込む処理単位（SubJob）の個数は複数個可能とする．また仮想プロセッサは一個の PC に複数個の存在が可能である．仮想プロセッサは，プロセスあるいはスレッドによって実装される．おおよそネットワーク計算モデルの場合はプロセスによって、マルチコアや SMP の場合はスレッドによって仮想プロセッサは実装されるのが適切と思われる．

future 宣言された関数（Subjob）を呼び出したプロセッサ（この時点で，マスタとなる）では，その Subjob の評価の完了を待たず（ブロックされることなく）実行制御が返ってくる．この時返値は”future という仮の値”となっている．したがってマスタはそのまま実行を継続することが出来る．関数の評価は他の仮想プロセス：ワーカによって，マスタの実行制御と並列に処理されるが，future 宣言された Subjob が大量に生成された場合，動的にワーカ群に分配される．

このようにして並列処理により全体の高速処理をはかる．

マスタにはマネージャが存在し，ワーカに動的に Subjob を割り当てる機能を持つ．他のワーカから Subjob 評価の依頼が来た場合，自分が忙しければ他のワーカに Subjob の依頼を行い，またその返値が来たことを管理するキューを持つ．

返値の管理　評価がワーカからマスタに返されて来たら、ワーカのマネージャは評価依頼された Subjob のキューを参照し評価完了をマークし，それまで future であった変数の値を実際とする．この時点でキューから Subjob の名前やデータなどの項目を削除する．

仮の値 future のままで制御の進行ができない場合には，評価の完了を待たなければならない．

評価の完了を待つ必要があるのは；
(1) future が四則演算式の項目として現れる実行文に至った場合。
(2) future が関数の引数として用いられている文に至った場合.
などである．

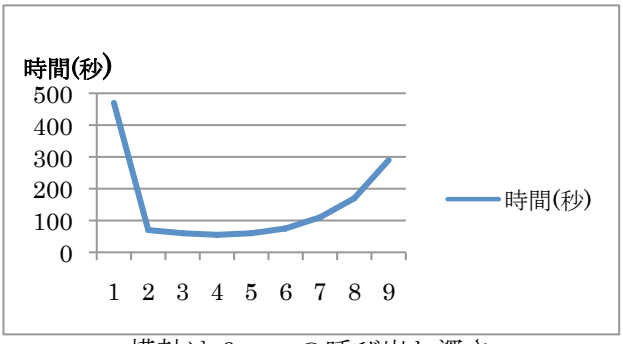
プリプロセッサ

“future”　ディレクティブによって future 宣言された関数を引数として API“future”を CALL するステートメントを生成し，また返値の確認を行う touch 関数を，必要な実行文の直前に挿入する．future の引数は；
返値が代入される変数のアドレス
その変数の型
評価関数名
関数の型
関数引数の個数
引数 1,引数 2,…
である．

例えばステートメントが

v = Sub(a1,a2,a3,..);
であれば、v, Sub が float 型の場合，プリプロセッサによって，一行全体が；
future(&v,"f",Sub,"f",a1,a2,a3,…)
に変換される．
“touch” 将来値 future がそのままでは実行ができないステートメントがでてきた場合，future の評価値が確定するまで実行をブロックするために使用される API であり，本研究では四則演算項，関数の引数．．．．等として future が使用されるとき，評価値が確定する迄，実行文の実行を停止する．先のシテートメントでは，変数 a の future 値が確定するのを待つために次のような文が挿入される；
touch(&v);

評価実験 13 クイーンの問題で，future の多重呼び出しを行った結果を示す、詳細は発表にて結果を示す．



横軸は futur の呼び出し深さ

```
/**_**FUTURE sub1 **_**/
main(){
 int a,b;
 a= sub(3,2,1);
…
…
…
 b = a + 10;
printf(“b=%f¥n”, b);
}
```

c 言語表記

＝＞
変換

```
main(argc,argv**){
int a,b;
F_init(argc,argv**);
future(&a,”f”,”sub”,”f”,3,2,1);
…
…
…
touch(&a);
b = a + 10;
printf(“b=%f¥n”, b);
F_end()
}
```

CLIP 表記

C 言語表記から future を使った CLIP 表記に，プリプロセッサによって変換される

図 1 プリプロセッサでの変換

参考文献：
[1]Robert H. Halstead,Jr:ParallelSymbolic Computing, Computer, IEEE, vol. 19, No. 8, 35-43(Aug.1986)