

科学技術計算用並行記述言語 NCC/i の プログラミング環境

1 K-3

國松 敦 木村 哲郎 天野 英晴

慶応義塾大学

1 はじめに

近年、多くのリンク結合による大規模な並列計算機が研究・開発されている。これらの計算機に適した言語の一つとして、メッセージパッシングに基づく並行記述言語が考えられる。これは処理系の実装が容易な反面、交信に伴うオーバーヘッドが大きいために、交信量の計算速度への影響が大きな問題になる。これを解決する方法として、複数のプロセスを一つのプロセスに融合するという方法が考えられる。ここで言う融合とは、複数のプロセスを一つのプロセスとして実現し、それらのプロセス間のメッセージの交信を単なる変数への代入に置き換えることである。融合されたプロセス同士ではメッセージによる交信を必要としないので、全体として交信するメッセージの量が減少し、実行速度の向上が得られる。

メッセージパッシング型の並行記述言語 NCC/i[1]ではプロセスを融合するための記述が可能になっている。しかし、プログラミング言語にもある程度の制約が加えられているため、プログラムの記述し易さを犠牲にしている。そこで、NCC/iによる実行時の速度の向上を得て、しかもプログラムを記述し易い環境が望まれる。ここではNCC/iのプログラミングを支援するNCC/iブラウザを提案する。

2 NCC/i

NCモデル(Node and Connecting-line Model)[2]では一般の物理現象を比較的容易に表現することができる。これは静的に存在するプロセスとネットワークにより問題を表現するモデルである。このNCモデルに基づいたメッセージパッシング型の並行記述言語NCC[2]に、プロセスを融合するための記述を可能にした言語がNCC/iである。プロセスの融合は、融合の対象となるそれぞれのプロセスの振舞い(制御の流れや交信回数等)の解析が必要となる。しかし、NCCのプログラムでこの解析を行なうのは非常に困難である。そこで、NCC/iでは複数のプロセス間で共通な制御構造(共通制御構造と呼ぶ)を明示的に指示することでこの解析を容易にしている。

NCC/iは基本的な構文はCに基づいており、メッセージパッシングのための命令を備えている。そしてプログラムは次の三つのパートにより記述される。

- プロセス宣言部:
プロセスが使用するポート(交信路の識別子)や

変数を宣言する。

- プログラム記述部:
プロセスのプログラムを記述する。
- ネットワーク記述部:
交信路(プロセス間のポートの結合)を記述する。

また、NCC/iでは共通制御構造を持つ場合、それらを融合して一つのプロセスにするような指示を記述することができる。NCC/iのプログラム記述部の例をリスト1に示す。このリスト中で

```
local プロセスタイプ名 {
    プログラム記述
}
```

と記述されている部分がそれぞれのプロセスタイプに固有のプログラムを記述する部分である。

この例で説明するとtype_Aとtype_Bの二つのプロセスタイプの記述があり、par_forがこの二つのプロセスタイプの共通制御構造である。

```
/* program description part */
group{
    int    i;

    local type_A {
        x = 0;
        send( out, x );
        receive( in, &x );
    }
    local type_B {
        receive( in, &y );
        y++;
        send( out, y );
    }
    /* common control structure */
    par_for ( i = 0; i < 100; i++ ) {
        local type_A {
            send( out, x );
            receive( in, &x );
        }
        local type_B {
            receive( in, &y );
            y++;
            send( out, y );
        }
    }
}
```

リスト1: NCC/i プログラム記述部の例

3 NCC/iのプログラムを組む上での問題点

リスト1に示されるように複数のプロセスの記述を一つのプログラムに混ぜて書かなければならず、プロ

プログラムの記述し易さを損ねている。つまり、NCC/i ではプロセスを融合することによるプログラム実行速度の向上を得るために、ある程度プログラムの記述し易さが犠牲にされているのである。このように NCC/i ではプログラムを組む上で問題となる点がいくつかある。ここに NCC/i がプログラムを組む上で問題点を整理する。

1. プログラムを、プロセス宣言部、コネクション記述部、プログラム記述部に分けて記述しなければならないためプログラムの編集作業が複雑になってしまう。また、変数名やポート名等は各部で整合を取る必要がある。
2. 融合可能なプロセスを記述する際には、それぞれのプログラムを混合して一つのプログラムに記述しなければならない。
3. 共通制御構造を意識してプログラムを記述しなければならない。

最後の点について補足説明をする。融合可能なプロセスにはプロセス間に共通制御構造が存在しなければならない。そのため、一般的なプログラム作成の流れから、融合可能なプロセス間の共通制御構造を決定してから、各プロセスを記述することになるものと思われる。すなわち、NCC/i のプログラマは融合可能なプロセスの共通制御構造を意識してプログラムを記述しなければならないのである。

4 NCC/i ブラウザ

NCC/i のプログラミング環境の改善を行なうための一案として、ユーザーインターフェースを改善するという方法が考えられる。そこで、どのようなユーザーインターフェースが要求されるのかを、先に述べた問題点をもとに考えてみる。

1. の問題に対しては、各部が必要に応じてすぐに表示・編集できるような環境が望まれる。それには、マルチウィンドウタイプの環境が適している。また、ブラウザが各部の整合をチェックする機能も望まれる。
 2. の問題に対しては、各プロセスタイプごとに編集できるような環境が望まれる。つまり、プログラマは融合可能なプロセスを各々のプロセスタイプごとに編集し、ブラウザが自動的に各プロセスタイプの記述を混合して NCC/i のプログラムリストを出力するような環境である。このような環境であればプログラマへの負担はかなり軽減されるものと思われる。
 3. の問題に対しては、共通制御構造には専用の編集画面を用意するという方法が考えられる。例えば、共通制御構造は通常のプログラム記述部とは分けて編集し、共通制御構造はここでのみ編集できるものとする。そして共通制御構造が決定次第、それを各プロセスのプログラム記述部に挿入する。この方法では自然に共通制御構造を意識したプログラミングが可能となる。
- 以上の考察を基に構築されたのが図 1 に示される NCC/i ブラウザである。このブラウザは OSF/Motif[3] を用いて実装されている。

また、NCC/i で科学技術計算をプログラミングする場合には、典型的な通信パターン及びプログラムパターンというものが存在する。このブラウザでは、そのようなパターンをテンプレート集として提供することもできる。

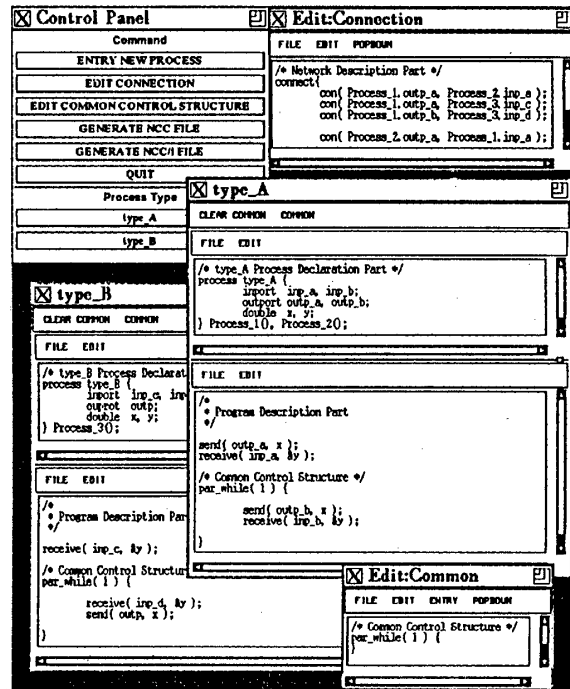


図 1: NCC/i ブラウザ

5 終りに

NCC/i のプログラミング環境を改善するために NCC/i ブラウザを作成した。よりよい環境を構築するためには、まだ、様々な機能が考えられる。今後、多くのユーザーの意見を取り入れてよりよい環境に仕上げていく予定である。

参考文献

- [1] Tetsuro Kimura, Taisuke Boku, Tomohiro Kudoh, and Hideharu Amono, *A Concurrent Programming Restructuring System for Scientific Calculations*, Hawaii International Conference on System Sciences; 1991
- [2] Hideharu Amano, Taisuke Boku, Tomohiro Kudoh, *(SM)²-II: A Large-Scale Multiprocessor for Sparse Matrix Calculations*, IEEE Transactions on Computers Vol.39, No.7, July 1990
- [3] Open Software Foundation, *OSF/Motif Style Guide*, 1990