

スケーラブルプログラミングシステム：CC++ コンパイラと使用経験

2U-2

藤田 昭平

fujita@cs.titech.ac.jp

東京工大 大学院情報理工学研究科

1 はじめに

大規模システムの解析および設計では、これを要素に分割し、その後個々の要素を結合する手段がしばしば用いられる。この際、個々の要素が有している性質が、各要素の結合により構成されたシステム (Composite Systems) で継承されるかが問題になる [1]。

それぞれの要素プログラムを並列に構成することにより構築されるプログラムを、構成型並列プログラム (Compositional Parallel Program) という。ただし、この構成型並列プログラムは各要素プログラムの性質を継承している [3]。

本報告では、オブジェクト指向言語に関数型言語 [2] の機能を組み込んだマルチパラダイムの構成型並列プログラミング言語 CC++ の特徴と使用経験について述べる。

2 CC++ とは

2.1 構成型並列プログラミング言語

構成型並列プログラミング言語 CC++¹ は、逐次言語 C++ に幾つかの簡単な拡張を加えたものである。CC++ は C, C++ のスーパーセットであり、正しい C あるいは C++ プログラムは、また正しい CC++ プログラムである。C および C++ との整合性により、C, C++ のユーザは CC++ へ容易に移行できる。

CC++ は、以下に示す 8 個の機能を追加することにより、C++ を拡張する。

- **par**

並列に実行される“文”を囲むブロック。

- **parfor**

並列に実行される“繰り返しループ”。

- **spawn**

並列に実行する新しい制御の“スレッド”を生成する。

- **atomic**

並列に構成された動作の“実行順序”を制御する。

- **sync**

“同期”のために使用されるデータ。

- **processor objects**

計算の分散 (“分散オブジェクト”) を記述する。

- **global**

分散オブジェクトをリンクする“ポインタ”であり、分散オブジェクト間の“相互作用”を記述する。

- **data transfer**

“分離アドレス空間”の間で、どのような“データを転送”するかを記述する。

このような僅かな拡張と単純さにもかかわらず、C++ の記述能力と併せて、CC++ は並列分散プログラミングのための豊富で強力な記述能力を有している。

2.2 CC++ の適用分野

CC++ は、以下に示すような幅広い分野に適している。

- **共有メモリモデルおよび分散メモリモデル**

CC++ は、共有メモリマルチプロセッサ、分散メモリマルチコンピュータ、さらにヘテロジニアスなネットワークベースの (分散メタコンピュータ) システムで使用できる。

¹Caltech Compositional Systems Research Group

- 粒度

CC++ は細粒度 (fine-grain) から疎粒度 (coarse-grain) まで幅広い並列分散計算を記述できる。細粒度の並列性は、小さなスレッド間の“頻繁な同期”を必要とする。疎粒度の並列性は、大きな分散オブジェクト間の“疎な同期”を必要とする。

- データ並列およびタスク並列プログラミング
データ並列およびタスク並列アプリケーションの両方 (およびその組合せ) を記述できる。

- 同期メカニズム

CC++ が提供する同期メカニズムは非常に強力であり、伝統的な命令的 (imperative) 同期と通信パラダイムを全て記述できる。

- 逐次プログラミングおよび並列プログラミング
並列ブロックと逐次ブロックの表記法は似ている。

C++ におけるオブジェクト指向の特徴は全て、CC++ でも保存されている。例えば、汎用クラスおよび継承メカニズムは、コードの再利用に便利である。したがって、

- ライブラリパッケージの開発
- プログラム開発、保守のための仕様記述

等に、CC++ は優れた“構成型の”記述法 (“Compositional” Notation) を提供している。

3 CC++ の処理系と使用経験

当日、OHP 等にて説明する。[4] も参照。

4 おわりに

CC++ は MIMD アーキテクチャの MPP および分散メタコンピュータのためのプログラミング言語である。

CC++ の基本的な特徴は次のようになる。

- CC++ は、オブジェクト指向言語 C++ を拡張し、関数型言語機能を組み込んだマルチパラダイムプログラミング言語である。

- CC++ は、一般的な共有メモリ型プログラミングモデルを提供している。分散オブジェクトへのアクセスは、グローバルポインタによる。

- CC++ のプリミティブ操作として、RPC を使用する。

- CC++ では、決定的プログラムおよび非決定的プログラムの両方を記述できる。

- CC++ では、全ての代入操作を値の単一代入として扱う。

- CC++ は形式的手法を支援することを目的としている。これにより、正しいプログラムの開発が促進され、テスト、デバックの“手間”が減少する。

CC++ は

- ♣ 大規模科学技術計算

- ♡ グラフィカルユーザインタフェース (GUI)

- ♣ リアクティブ システム (分散制御)

等の幅広い分野に適している。

参考文献

- [1] S. Fujita et al., “Controllability and Observability of Composite Time-Varying Linear Systems,” (in Japanese), 計測自動制御学会 論文集, 4 巻, 3 号, 234 - 241, (1968).

- [2] B. Randell et al., *Advanced Course on Functional Programming and its Applications*, 20 - 31 July 1981.

- [3] K.M. Chandy et al., “Compositional C++ : Compositional Parallel Programming,” Caltech CS-TR-92-13, (1992).

- [4] 藤田 他(訳)、並行分散システム : オペレーティングシステム、データベース、分散マルチメディアシステムへの統合的アプローチ、トッパン (1994、秋)。

Scalable Programming Systems : CC++ -
Its Compiler & Experience -
Shohei FUJITA

TITech, Graduate School of Computer Science
Meguro-ku, Tokyo 152, Japan.