

1 はじめに

1.1 想定するハードウェア

XMP は、PC クラスタ、ブレードサーバや、分散メモリを持つスーパーコンピュータで利用できる、並列プログラミング言語です。プログラミングは、図 1 のように、メモリが分散している計算機を想定して行います¹。

XMP では、メモリと CPU をもつ「ノード」を並列実行の単位と考えます。データ（変数）を複数のノードに分散させ、同時にデータに対応する計算処理も分散させることで、並列効果を狙います。

CPU は同じノード内のメモリに対して通常通りアクセスできます。ノード間のデータのやりとりは、メモリ間の通信を行う構文を使って行います。

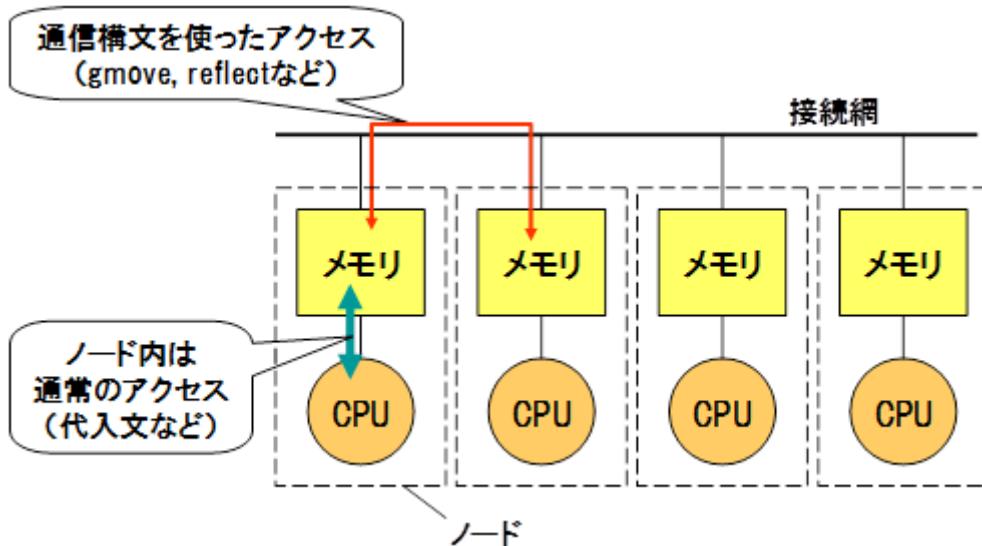


図 1: 想定するハードウェア

¹ 実際のハードウェアは、共有メモリであってもかまいません。チップ内の各コアを CPU と考えることもできます。

1.2 XMP プログラミング

XMP は、Fortran をベースとする仕様と、C をベースとする仕様があります。どちらも、ベース言語を指示文で拡張した言語仕様です。ここでは、Fortran をベースにする XMP を解説しています。

XMP は、2通りのプログラミングスタイルをサポートしています。

- グローバルビュー：問題の全体をプログラムし、それを N ノードに分けて実行させる。
- ローカルビュー：1 ノードの動作をプログラムし、それを N 個並列に実行させる。

グローバルビューのプログラムでは、データや計算はまずグローバルなイメージで表現され、それを N 個に分割すると表現されます。実際に各ノードに配置されるデータと各ノードで行われる計算や通信は、コンパイラが生成します。ローカルビューのプログラムでは、各ノードに配置されるデータと各ノードで行われる計算や通信が表現されます。

ここでは、グローバルビューによるプログラミングを解説しています。グローバルビューのプログラミングは、逐次プログラムを並列化することです。XMP における並列化とは、

- 1) データの分散
- 2) 計算の負荷分散
- 3) 通信・同期

を記述することです。