

今後の並列プログラミングモデル ～レガシーモデルから関数型まで～

富士通研究所 岩下英俊

2013/8/29

HPC向け並列プログラミングモデル

■ アプローチは二分化すると考える。どちらも重要

	レガシーモデル	新しいモデル
狙い	<ul style="list-style-type: none">既存アプリの維持、連続的発展ユーザの維持、拡大	<ul style="list-style-type: none">従来にない新しいアプリ領域新しいユーザ層の獲得
価値	<ul style="list-style-type: none">性能 (FLOPS値)、HWピーク性能比	<ul style="list-style-type: none">定性的に新しいものが書ける
要件	<ul style="list-style-type: none">絶対性能が出ること過去の資産との連続性	<ul style="list-style-type: none">領域を特化し、それに適した文法書式単体性能よりも、スケールアウトすること
例	通信ライブラリ: MPI, ARMCI, GM 共有メモリ向け言語: OpenMP, TBB(Intel) アクセラレータ向け言語: OpenACC データ並列: HPF, XPF(富士通), XMP PGAS: CAF, UPC	データ並列: Chapel(Gray) 関数並列: X10(IBM) 関数型言語: Erlang, Haskell, Scala 関数型フレームワーク: MapReduce

■ EXA時代でも、レガシーモデルは続くと思われる

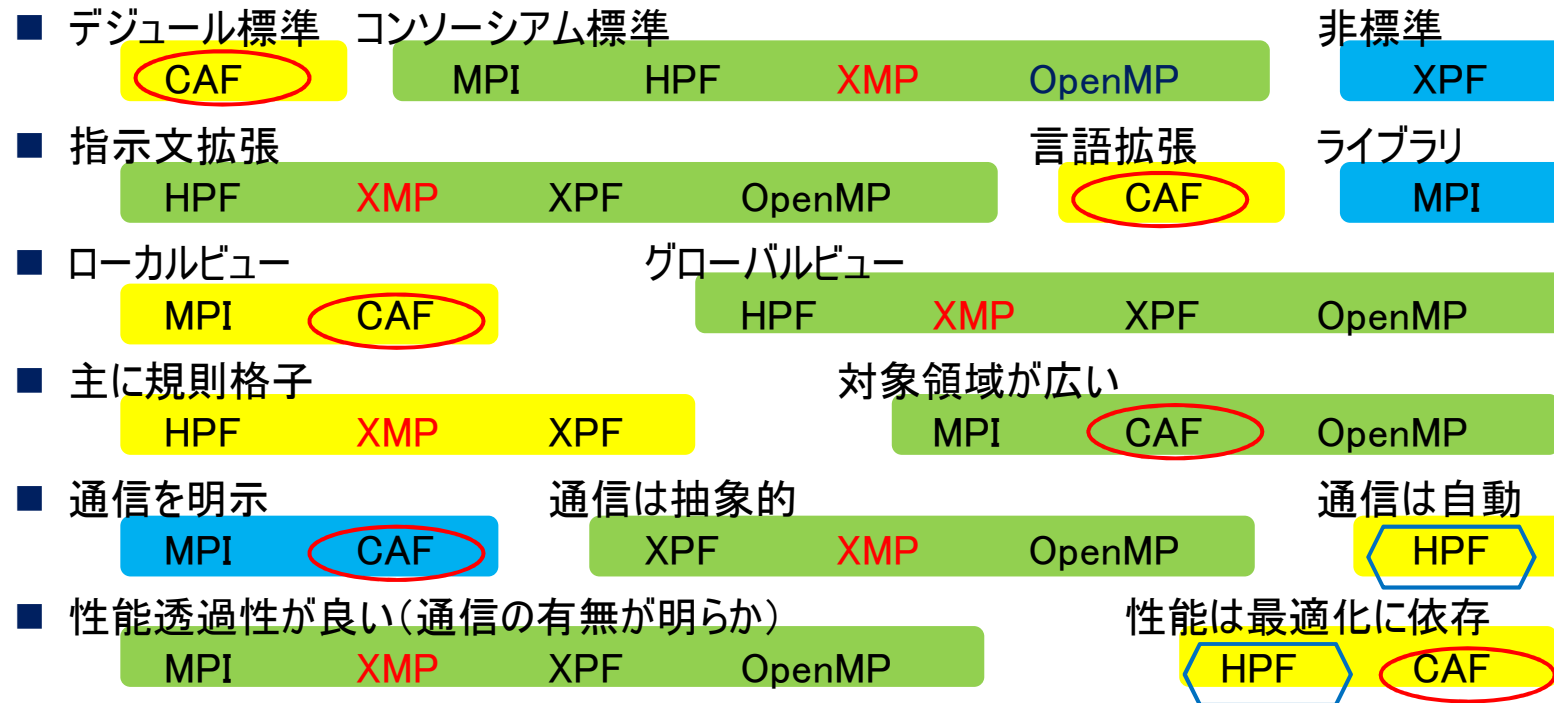
- ハードは数年、アプリは10年

- 要件から、ベース言語はFortranとC。この20年大きく変わっていないので、この先も。

■ 新しいモデルは爆発するか？

- キーは関数型。安価な計算パワーをざぶざぶ使う考え方で高並列を狙う。

■ 分散メモリ向けモデル + OpenMP を分類



■ XMP 言語仕様の特徴

- OpenMP (成功した言語) に似ている。課題は対象領域が限定されること。
- CAF と相補的。V1.1 の CAF (coarray 機能) 対応は、XMP のカバレッジを大きく広げた。
- HPF の自動化のメリットを捨てて、性能透過性を採った。ここは賛否あるかも。

- すべてを「関数の定義」で記述する言語
 - LISP、LISP風言語Scheme、Closure
 - Scala (object指向 + 関数型)、Erlang (Prolog風)、Haskell (独特)
- 純粹関数型の場合、副作用を持たない。
 - 同じ入力には同じ出力を返す。関数の評価が他に影響を及ぼさない。
→ 並列に呼び出すことができる。動的な並列化 (タスク並列)
- HPCで特に注目されているもの
 - Erlang
 - エリクソン社で商用。分散環境向け。軽量プロセスによる並行処理
 - 耐障害性 (関数の応答と前提としない実行) → OR並列、探索問題
 - Haskell
 - 完全な純粹関数型
 - 先進的機能多数。理解困難との批判も
- 内側を従来の並列化、外側に関数型並列を使う
 - 動的なタスク生成、連成シミュレーション、アンサンブル

最後に (Open Discussion)

■ (再掲)アプローチは二分化すると考える。

	レガシーモデル	新しいモデル
狙い	<ul style="list-style-type: none">既存アプリの維持、連続的発展ユーザの維持、拡大	<ul style="list-style-type: none">従来にない新しいアプリ領域新しいユーザ層の獲得
価値	<ul style="list-style-type: none">性能 (FLOPS値)、HWピーク性能比	<ul style="list-style-type: none">定性的に新しいものが書ける
要件	<ul style="list-style-type: none">絶対性能が出ること過去の資産との連続性	ベース言語は FortranまたはC <ul style="list-style-type: none">狙った領域に特化した書き易い文法書式実アプリに対応できるスケーラビリティがある

- レガシーモデルは今後どう変わるか？ 変わらないのか？
 - 新しいモデルとの中間に解はないか？
- XMPは、レガシーモデルの最後の(最善の)アプローチになれるか
 - ミッションクリティカルなプロジェクトは信頼できるMPI ← ここを破壊できるか？
 - 市販/フリーのパッケージライブラリ/フレームワークで十分 ← どう対応するか？
- XMPをレガシーユーザにアピールしたい。何が足りないか
 - 使えるコンパイラを早く出すこと。そのためにどうするか
 - アプリが足りない。アプリがないとコンパイラが迷走する
 - 開発環境(プログラム、デバッグ、プロファイル等)の充実はこれから
- 新しいモデルは、HPCユーザを集められるか
 - 静観でよいのか。何ができるか
 - レガシーと新モデルのハイブリッドに行くか