

テーマ：5年後にセンター運用可能な高性能並列計算機システムは、どのようなシステム仕様が考えられ、そのために、今後どのような研究開発をしていくべきなのか

大島聡史（東京大学 情報基盤センター 助教）

5年後のプログラミング言語（処理系）
について／GPGPUを含めた視点から

個人的な 主な興味の方角性

- × ヘテロジニアス環境における全プロセッサの最大活用
 - + 具体例：CPUとGPUがあるときにどう仕事を振り分けるのが最良か？
 - + 独自の超プログラミング言語が作りたい、という立場ではない
 - × (ので、「真新しい言語を作きましょう」という話はしません)

現在のセンターマシンとプログラミング環境

ノード間並列化：
MPI

計算ノード× β
：～数千台

ノード内並列化：
MPI、OpenMP



言語：C、Fortran

マルチコアCPU× α
：4コア×4ソケットetc.

ShellScript、Perlなど



スクリプト処理

CUDA

アクセラレータ(GPU)× γ
：～数百並列×～数台/node

5年後のセンターマシン想像図

× 超大規模クラスタ型 with アクセラレータ

計算ノード× β
: ~数千台、数万台



アクセラレータ× γ
(GPU or GPU後継機?)
: ~数千並列×~10台



マルチコア/メニーコアCPU× α
: ~100コア×~50ソケット
(ノードあたり~1000コア程度)

- 想定: CPUのみで100PFlopsを目標とすると、CPU 2.0GHz*4IPC*1000コア=8TFlops, 12500ノードで達成
- 個人的には、CPUのコア数は著しくは増えないと考えている。シンプル超多数コアはGPUが担当すればよい。

5年後のプログラミング環境

× 現在のプログラミング環境は引き継げるか？

+ CPU

× flat MPI

- * 1000コア×10000ノード=1000万プロセス
→非現実的

× OpenMP+MPI

- * 1000コアOpenMP×10000ノードMPI
→幾分、現実的な範囲

+ GPU

× CUDA、OpenCL

- * 数千並列×～10台
→規模的には現実的、GPU間の通信が課題か

「個別では」
なんとかなりそう
(5年後も使えそう)

5年後に向けた課題

- × マルチコア・メニーコアCPUとアクセラレータ（GPU）に適切に処理を割り当てて最大性能を達成できるか？
 - + ノード内の「選択」
 - × CPUもGPUも数千並列、どちらで実行する？
 - × 実行時に（データにより）決まる可能性
 - + 適切な並列度
 - × 常に数千の並列度を使い切れるか？
 - × ノードを占有しきれない処理ばかりではない（が、ノード内で分割すると主記憶が足りない）
 - × アプリケーション全体にまたがる並列度管理（データとタスク管理）
 - + どうやって記述するか？
 - × 新言語：普及、教育、……

解決策 = 5年後に向けた研究

- × タスク（関数など）の特性を反映した最適化（スケジューリングなど）が可能な、既存言語ベースの処理系
- × タスクの依存関係・並列度・メモリアクセス特性
 - + ユーザ（プログラマ）による特性記述が必要
 - + タスク並列言語：ユーザの負担が大
 - + 指示子、セマンティクス（OpenMP、シェーダ言語）
 - + データと変数の対応付け（データがどう移動するか）
 - × △ 変数がどのメモリに確保されるか
- × 実行時（実行直前）に決まるパラメタ（変数）も利用（言語だけではなくランタイムの研究）
 - + with 自動チューニング