

プログラミング言語

2011/05/28

- アドバイザ
 - 中島浩、佐藤三久、米澤明憲*
- 取りまとめ
 - 丸山直也(東工大)、滝沢寛之(東北大)*、中尾昌広(筑波大)、千葉滋(東工大)*、田浦健次郎(東大)*、八杉昌宏(京大)
- 参加者
 - 平石拓(京大)、田中 良夫(産総研)、窪田昌史(広島市立大)

*印：第4回ワークショップ欠席者

現状認識

- OpenMP, MPI
- 自動並列化コンパイラ
- Pthreads (Java)
- HPF, Chapel
- PGAS (X10, XcalableMP, CAF, UPC)
- Charm++, ABCL, RPC
- Cilk, TBB
- CUDA, OpenCL, GLSL, Brook
- SIMD
- Tascell
- DSL (ArBB, Google Graph, ...), MapReduce
- 関数型言語 (Fortress, 並列Haskell, Id), MATLAB
- (GHC, KL1, Freng)
- Sequoia, SWIFT (task parallel)
- Xcrypt, GXP Make

俯瞰図の軸

- 各言語がどう使われていて、どういう問題があるか。
 - 使われていないのはなんで使われてないか
 - 逆に使われているものはなんで使われているのか？
- 特徴付けるもの
 - 共有メモリ vs 分散メモリ (global/local view)
 - ノード内外
 - Explicit (低水準) vs Implicit (高水準)
 - ヘテロ (GPU) をユーザに見せる/見せない
 - Domain specific?
 - データ並列 vs タスク並列

10年後のプログラミング言語・モデル に必要な機能

- 10年後のアーキテクチャ?
 - BlueGene like or scalar-vector heterogeneous
- 要件
 - Performance, Scalability
 - Locality, Load balancing, Adaptability (動的に使用するリソース量を変える)
 - Productivity
 - 透過性, 俯瞰性, 高水準、debugging, performance tuning
 - Fault tolerance/resilience
 - → 全体はムリ, 部分CheckPointing
 - Interoperability (特にOpenMP, MPIとの)
 - 低教育コスト(Fortran-like?)
 - Safety, Reliability, (consistency)
 - Low power consumption

その他議論されたこと

- そもそもOpenMP/MPI(+CUDA)でなぜダメ？
 - そのうち書けなくなるはず
 - 性能面でも
 - Fault tolerant自体はMPIに入る？
- OpenMP+ (MPI→RPC)で, 階層並列化→高scalability, fault tolerance の実績
 - 現状では, gridでしか動かないので普及の問題
- CUDAがあったからGPUが普及した. (なぜCUDAは普及したか? 簡単に書けそうだったから?)
- PGASは普及するか? (MPIに対する優位性をどう主張?)
- アプリ屋はOpenMP/MPIでやってしまうのではないか?
 - 新しいの
- ESでHPFを宣伝したけどあまり使ってもらえなかった
 - 理由の分析: 他でつかえるかの心配, 性能でないときにどうすればいいかがわからない
- from scratch でない人をどうするか(教育コスト問題)
- Exaでどんなアプリ, プログラムが書かれるか

- DSLだとして, その実装はどの言語で誰が書く?
- requirementは共有したいけど, 実現方法は必ずしもそうでなくていい

ロードマップ作成に向けて

- 要件を満たすプログラミング言語・モデルはなにか？
- 案1: アプリケーションドメイン専用モデル・フレームワーク・言語
- 案2: 汎用言語で担保

ドメイン特化型モデル

- なぜ？
 - 要件をすべて満たす+汎用→破綻
- しかし・・・
 - ドメインはたくさん→だれが作るのか？
 - ウェブ・エンタープライズソフトウェアでは →ドメイン特化型言語・フレームワークをつくるためのメタフレームワーク