

SCIMAコンパイラへの導入

— 春の物思い —

東京大学大学院

工学系研究科電気工学専攻

坂井修一

sakai@mtl.t.u-tokyo.ac.jp

<http://www.mtl.t.u-tokyo.ac.jp/~sakai/>

背景

■ 計算機システムにおける収束と発散

– HPCから情報家電・携帯電話まで :収束と発散が進行中

– 収束

- 半導体デバイス : CMOS
- CPU基本アーキテクチャ : Intel、RISC
- 通信基本プロトコル : IP
- OS : Windows, UNIX
- 言語 : FORTRAN, C(++), JAVA
- ハードウェアやアーキテクチャのライブラリ : IP
- I/O系
- ソフトウェアのコンポーネント: 枠組の作り方を含む
- ハイエンド側のレスポンス重視、ローエンドのスループット重視

– 発散

- (とはいってもの) HPCはスループット指向、ローエンドはレスポンス指向
- 実装 : コスト、計算速度、メモリ量、実装規模、消費電力などで対極
- アプリケーションの流動性 : HPC、ビジネス、WS以下で異なる

HPCコンパイラを考えるにあたって

SCIMAはアーキテクチャ透過性を犠牲にして性能をかせく方式

■ 選択肢

- ベース言語：性能、プログラマの嗜好と慣れ、互換性
 - FORTRAN
 - C, C++
 - JAVA
 - FLENG, Id
- ベースコンパイラ: ソースが使えることが必須
 - (from scratch)
 - SUIF, GNU, NEWCC
- (並列化)ライブラリ
- ターゲットユーザ：パワーユーザから初心者まで
 - ソースの記述レベルが異なる

NEWCC on SCIMA (1/2)

■ NEWCC

- 富士通研による第1版作成
- 田中・坂井研との共同研究開発(1998-)
- SCIMA部分は未来開拓による開発
- 国際産学共同研究(2001-)

■ 選択の理由

- Cの汎用性
- ベースシステムあり、ソース利用可能
- コンパイラチームの存在
 - HPC以外の最適化は共通化
- 開発のしやすさ
 - 日本語での即時的なやりとりなど

NEWCC on SCIMA (2/2)

■ ターゲットユーザ

中級:

- 上級は完全にアセンブラや詳細なプラグマの形で書く
- アーキテクチャを全く意識しないコードは最適化に限界

■ 中身 中村君発表

– 上流

page load/store最適化など

– 下流

MIPS用 / SCIMAシミュレータ用バックエンド
コードスケジューリング

おわりに

- NEWCC on SCIMA 開発の経緯
- NEWCC on SCIMA (2001)
 - コードスケジューリングなどの充実
- 一般論
 - ユーザレベルの細分化
 - 自動化とツールのバランス
 - マルチプロセッサ化
 - I/O系：レスポンスも重視
 - ネットワークコンピューティングとの関係