

HPC-phys 勉強会

SX-Aurora を使ってみた

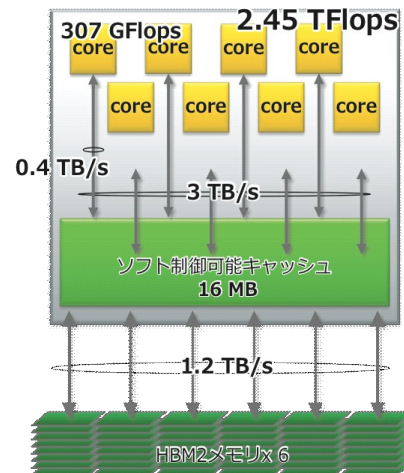
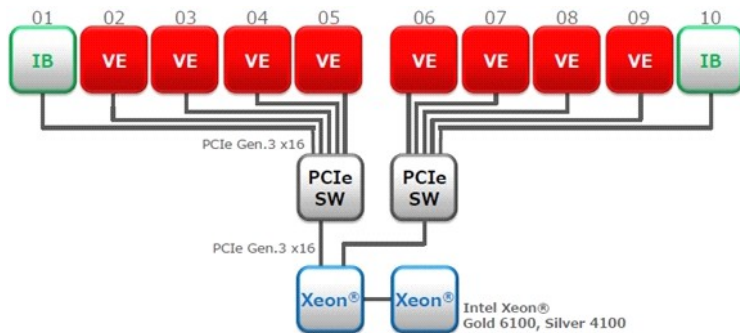
Hideo Matsufuru (KEK) 

31 January 2020



SX-Aurora@KEK

- NEC SX-Aurora TSUBASA A500-64 1 rack (64 VE/8 VH)
 - NEC ベクトルアーキテクチャの最新版
 - ノード構成：Vector Host (VH) + 8 x Vector Engine (VE)
 - プログラムは VE上で直接実行、システムコールをVHにオフロード
 - Peak performance: 156.8 TFlops, 3 TB memory
 - 2020年3月に2倍に増強
- Vector Engine
 - 8 cores, 2.46 TFlops (DP)
 - Memory B/W: 1228 GB/s (B/F=0.5)
 - Vector length = 256, #vector register = 64





プログラミング

- “VE (Vector Engine) execution model”
 - プログラムはVEで実行 (明示的なオフロードは不要)
→ 通常のノードと同じように使用
 - I/O などのシステムコールをVH (Vector Host) へ “オフロード”
 - 通常のコードを再コンパイルして利用可
 - 性能が不十分なら指示文の挿入などで細かく制御
 - プロファイラによる分析
- 開発環境
 - Fortran, C/C++ コンパイラ (自動ベクトル化)
 - MPI, OpenMP, 最適化ライブラリ (BLAS, LAPACK など)
 - プロファイラ、デバッガ



素粒子宇宙原子核シミュレーションプログラム

青山龍美、日本物理学会2019年秋季大会スライドより改編

- 素粒子・原子核・宇宙物理学分野の大規模シミュレーションに基づく理論的研究を推進
 - これまでのKEK大型シミュレーション研究に続く共同利用プログラム
 - 計算基礎化学連携拠点(JICFuS)によるサポートの下で共同利用を実施し、計算資源を提供
 - 大規模実験プロジェクトとの連携：実験データを精密に予言する理論計算により実験の成果の最大化を図る
- 研究課題を公募

<http://research.kek.jp/group/pna-sp/>

 - 研究テーマを設定しグループを組織、代表者が申請
 - 公募要項についてはKEK共同利用実験を参照
 - 一般利用の課題は随時受付
 - 大型利用の申請：間もなく2020年度の公募を予定



使ってみた

- 球対称性(1D)の下での超新星爆発
 - 住吉さん(沼津高専)との共同研究
 - 2019年秋の学会で報告
 - ブロック三重行列の線形方程式+衝突項の計算 (ニュートリノ反応過程~積分計算)
 - ベースコードはFortran, 一部 C (GPU利用のための変更) + MPI
- 格子QCD
 - Bridge++プロジェクトでの研究
 - 2019年秋の学会で青山さんが報告
 - 4次元格子上のステンシル計算：疎行列の線型方程式
 - C++, MPI + OpenMP



超新星爆発シミュレーション

Elapsed time [sec]/iteration

$$N_r = 256, N_{E_\nu} = 14, N_{\text{ang}} = 6$$

| System | IBM POWER8 | POWER8+Pascal | SX-Aurora |
|--------------------|--------------|----------------|------------------|
| Accelerator | | NVIDIA P100 x4 | Vector Engine x1 |
| #core/device (DP) | 20 (16 used) | 1792 | 8 |
| Peak/device (DP) | 0.46 TFlops | 4.7 TFlops | 2.46 TFlops |
| Memory Bandwidth | | 720 GB/s | 1228 GB/s |
| Elapsed time [sec] | | | |
| Collision term | 0.441 | 0.30 | 2.4 (C-code) |
| Iterative solver | 3.21 | 1.59 | 0.76 (Fortran) |

- SX-Aurora 1 VE (8 core) を 8 MPI process で使用 (flat-MPI)
 - Fortran は original code、C は OpenACC version を適用
 - ベクトル演算器に合わせたチューニングは未実施
 - 線形ソルバーは Fortran コードで十分な効率 (Cf. C-code: 2.6 sec)
 - Collision term は C-code の方が高速 (Fortran: 13.5 sec)
→ もう少し速くしたい：ベクトルレジスタに合わせた構造変更が必要



格子QCD



- **Bridge++**

- 格子ゲージ理論のための汎用コード
- オブジェクト指向、C++
- 設計方針：可読性、拡張性、移植性、高性能
- 2009年開発開始、2012年公開、現在 ver.1.5.3 (2019.12)
- 現在の開発メンバー：

Y. Akahoshi, S. Aoki (YITP), T. Aoyama, Y. Namekawa, H. Matsufuru (KEK), I. Kanamori (RIKEN), K. Kanaya, Y. Taniguchi (Tsukuba), H. Nemura (RCNP), and contributors

- **最新アーキテクチャへの対応**

- **Bridge++ “core library” + Extensions (alternative code)**
→ SIMD (KNL, Xeon Skylake), GPU (OpenACC), etc.
- **今回のターゲット：Vector architecture**



格子QCD

- 実装方針

- サイトループでベクトル化
- AoS (Array of Structure) から SoA にデータレイアウトを変更
→ サイトループを最内側へ
- ループ長を SX のベクトル長 256 に合わせる
- コンパイラの出力を見て必要なら指示文を入れる
- プロファイラ (Ftrace)

- 現状

- Wilson, staggered フェルミオンの線型ソルバーが実装済
- MPI 並列化はOK, OpenMP マルチスレッド化を実装中



格子QCD

- 結果

- Cf. 青山龍美、日本物理学会 2019年秋季大会

| | Single core 16x16x8x8 | 1 VE (8 cores) 16x16x16x32 / [1,1,2,4] |
|--------------------------|--------------------------|---|
| Wilson mult | 41.7 GFlops | 81.8 GFlops |
| Staggered mult | 31.4 GFlops | 80.0 GFlops |
| Wilson solver (BiCGStab) | 29.1 GFlops | 79.1 GFlops |
| peak performance | 308 GFlops | 2.42 TFlops |

- 1 core では ~10% の性能、しかし 8 process では並列性能が不十分
- ベクトル化率 99.9, 平均ベクトル長 256: 十分なベクトル化
- 並列性能が悪い原因は現在調査中
- マルチスレッド化の方が良い? 現在実装中



まとめ

- SX-Aurora はベクトルアーキテクチャ
 - Byte/Flop が大きい (~ 0.5) のが特長
 - ベクトル長が長い計算に有効
 - Vector Engine でコードを実行 (通常のノードのように使える)
- プログラミング
 - 再コンパイルしてそこそこの性能は出る
 - ベクトルエンジンの特長を活かすにはチューニングが必要
 - 高速化、特に並列性能向上のテクニックを策定中
- 共同利用： KEK 素粒子原子核宇宙シミュレーションプログラムを通して利用可能



THE INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ITS APPLICATIONS

University of Cagliari, Cagliari, Italy July 1-4, 2020

Log in

only in current section

Search Site

Search

Home

Themes

Committees

Workshops

Call for Papers

Submission Site

Home

Quick Links

[Instructions for Authors](#)

[Registration Info](#)

[ORCID](#)

[Publishing Open Access in LNCS](#)

The 20th International Conference on Computational Science and its Applications

The 20th International Conference on Computational Science and Applications (ICCSA 2020) will be held on July 1 - 4, 2020 in Cagliari, Italy in collaboration with the University of Cagliari, Italy.



ICCSA 2020

- 1-4 July 2020 at Cagliari, Italy
- Workshop "Large Scale Simulation Science"
- Paper submission: deadline 15 Mar 2020

