

課題名 (タイトル) :

ポスト「京」アプリケーション開発

利用者氏名 :

○石川 裕*、中村 宜文*、辻 美和子*、八代 尚*、三上 和徳*、石川 健一**、南 一生*、黒田 明義*、熊畑 清*、北澤 好人*、鈴木 惣一朗*、藤田 航平*、西澤 誠也*、似鳥 啓吾*、河東田 道夫*、児玉 祐悦*、村井 均*、畑中 正行*、小山 洋一 a、藤田 憲正 a、渡辺 尊紀 b、中島 浩*

所属 :

* 計算科学研究機構フラッグシップ 2020 プロジェクト

** 計算科学研究機構連続系場の理論研究チーム

a 株式会社トランス・ニュー・テクノロジー

b 株式会社アックス

<p>1. 本課題の研究の背景、目的、関係するプロジェクトとの関係</p>	<p>(1) GENESIS ① 計算内容:創薬候補スクリーニング ② 計算手法:分子動力学</p> <p>(2) Genomon ① 計算内容:個人ゲノム解析 ② 計算手法:大容量データ解析</p> <p>(3) GAMERA ① 計算内容:地震シミュレーション ② 計算手法:非構造・構造格子ステンシル複合の有限要素法</p> <p>(4) NICAM+LETKF ① 計算内容:気象予測 ② 計算手法:構造格子有限体積法+局所アンサンブル変換カルマンフィルター法</p> <p>(5) NTChem ① 計算内容:光化学エネルギー変換シミュレーション ② 計算手法:高精度分子軌道法</p> <p>(6) FFB (FrontFlow/blue) ① 計算内容:ターボ機械の熱流動解析など ② 計算手法:非構造格子・有限要素法</p> <p>(7) RSDFT ① 計算内容:異種物質ナノ界面などの解析 ② 計算手法:密度汎関数法</p> <p>(8) Adventure ① 計算内容:例)ターボ機械全体シミュレーション ② 計算手法:非構造格子・有限要素法</p> <p>(9) 格子 QCD</p>
<p>理化学研究所計算科学研究機構は、社会的・科学的に重要な課題群を解決するべく、文部科学省のポスト「京」開発プロジェクト(フラッグシップ 2020)のもと、2020年の完成を目指し、ポスト「京」スーパーコンピュータの設計開発及びアプリケーションソフトウェア(アプリ)の開発を行っている。本プロジェクトの特徴的な点として、ハードウェアとソフトウェアの開発設計を両者の綿密な連携によって進める協調設計(コデザイン)を行っていることが上げられる。アプリケーションの研究開発をハードウェアの設計と同時期に行い、互いにフィードバックすることで、課題解決のための最適な設計を可能とし、最大限の成果を早期に創出することを目指している。本課題は、上記プロジェクトのコデザインにおける社会的・科学的に重要な課題群を解決するためのアプリケーション群の研究開発を行うものである。</p>	
<p>2. 具体的な利用内容、計算方法</p> <p>ポスト「京」で重点的に取り組むべき社会的・科学的課題が有識者会議において選定され、さらに各課題からターゲットアプリケーションが選定されている。計算内容は素粒子、物性、化学、生命、気象、地震、ものづくり等幅広く、その計算手法は多岐にわたる。以下がターゲットアプリケーションと計算内容及び計算手法である。</p>	

- ① 計算内容:素粒子シミュレーション
- ② 計算手法:構造格子経路積分モンテカルロ法

これらの計算性能を測定し、その詳細な分析を行い、律速部分の最適化・アルゴリズムの改良を行う。これにより、上記課題を実現するために必要なポスト「京」のハードウェア・システムソフトウェアスペックが明らかになる。ハードウェア・システムソフトウェアの調整・改良でも実現困難な場合は、アプリケーションのアルゴリズムのさらなる改良を行う。

3. 結果

(1) GENESIS

本課題での報告はない。

(2) Genomon

Genomon は、入力データを加工して出力する小規模なプログラムをパイプライン的につなげるによりゲノム解析を行うワークフロー型アプリケーションである。アプリケーションチューニングを FX100 システム (MPC) で行った。MPC では、「京」と比較して set jmp 関数の実行オーバーヘッドが増加していたため、該当プログラムのエラー処理を書き換えて対応した。また、ポスト「京」及びポスト「京」時代の Intel 社製 CPU 搭載の同規模スーパーコンピュータ上での Genomon の性能予測を目的として、MPC 及び Intel Xeon 搭載機 (ACS) 上での性能測定を行った。I/O の比重が比較的大きなパイプライン構成プログラムでは、特に MPC において、ストレージシステムの構成に由来して実行時間のばらつきが多かったため、これらのパイプライン構成プログラムについては、ローカルストレージを持つ「京」での性能実測値を基に性能予測を行った。

(3) GAMERA

ポスト「京」における GAMERA の性能予測のため、MPC 及び ACS 上で性能評価を行った。また、「京」用にチューニングされた主要演算カーネルに対し、複数のチューニング手法を適用することによって、両システム上でも性能が改善することが分かるなど、マシン特性にあわせたアルゴリズム・実装開発に対

する知見を得た。

(4) NICAM+LETKF

ポスト「京」及びポスト「京」時代の Intel 社製 CPU 上での気象シミュレーションコードおよびデータ同化ツールの性能を評価するため、抽出した演算カーネルとアプリケーション全体についてそれぞれ性能評価を行った。計測には MPC 及び ACS を利用した。MPC を用いた計測では hybrid 実行形態での詳細な演算量、データ移動量、命令数およびキャッシュミス率を取得し性能特性分析を行った。ACS では flat MPI 実行形態での実効性能および経過時間を計測した。これらの情報を元に、ポスト「京」およびポスト「京」時代のスーパーコンピュータ上での性能を補外によって推定した。構造格子を用いた気象アプリケーションでは、どちらのシステムにおいてもメモリ・キャッシュ性能が実効性能を大きく左右することがわかった。

(5) NTChem

NTChem による重点課題ターゲット問題をポスト「京」の構成で実行する場合を想定して設定した計算モデルの 1 プロセスあたりの計算データを作成し、MPC の計算ノードおよび ACS の計算ノードで実行し、ハードウェアパフォーマンスカウンタをベースにして得られるプロセス性能統計情報を収集した。経過時間と計算の内容をもとにしたサブルーチン分類を行い、負荷の高いサブルーチンから実効性能、メモリ・キャッシュを経由するデータ移動量、浮動小数点演算量、整数演算量、待ち時間などの性能特性分析を行った。計算負荷の大きな部分を二電子積分計算が占めることが MPC・ACS とともに確認され、MPC では計算時間全体のボトルネックが L1/L2 キャッシュのデータロード待ち (ビジー状態) 時間として現れること、複雑な間接参照のため SIMD 化ができない区間ではキャッシュの構成の違いにより ACS と MPC との間で計算性能に有意な差がでることなどが明らかとなり、今後の開発検討の指針となる重要な情報が得られた。

(6) FFB

次の 2 点について利用した。①ポスト「京」完成時点において利用可能な汎用のスーパーコンピュータ上での本アプリの性能予測を目的として、ACS を用いてベンチマーク問題を実行した(実行形態は flat MPI)。その結果から、ポスト「京」完成時点でのインテル製 CPU 採用スーパーコンピュータ上での本アプリの性能を、ソケット当たりの演算能力、メモリバンド幅、利用可能プロセス数などを補外することで推測した。②ポスト「京」上における本アプリの性能予測を目的として、MPC を用いてベンチマーク問題を実行した(実行形態は hybrid)。詳細プロファイラを用いて計測した演算命令数、キャッシュミス回数などのハードウェアカウンタ値から、想定されるポスト「京」のハードウェアスペック上での本アプリの性能を推定した。

(7) RSDFT

Intel 社製 CPU を使用したポスト「京」時代の一般的なスーパーコンピュータ及びポスト「京」での RSDFT アプリケーションの性能を予測するため、ACS 及び MPC の 1 ノードを用いて、RSDFT のカーネルの主要部分として、行列行列積ライブラリの計算区間の性能を測定した。各測定区間のピーク性能の比率は、ACS と MPC のシステムの違いでは大きく変わらず、どちらもライブラリの性能に依存することがわかった。この結果、RSDFT に関しては、行列行列積ライブラリの性能が出やすいパラメータやシステム環境の設定が必要であることがわかった。

(8) Adventure

Adventure による重点課題ターゲット問題をポスト「京」の構成で実行する場合を想定して設定した計算モデルの、主要計算部 2 区間のそれぞれの、1 プロセス当たりの計算に掛かる実行時間を、ACS の 1 ノードを用いて測定し、Intel 社製 CPU を使用したポスト「京」時代の一般的なスーパーコンピュータでの Adventure のアプリケーション性能を推定した。

(9) 格子 QCD

Intel 社製 CPU を使用したポスト「京」時代の一般的なスーパーコンピュータ及びポスト「京」での格子 QCD アプリケーションの性能を予測するため、ACS 及び MPC の 1 ノードを用いて、格子 QCD の主要計算部 5 区間の性能を測定した。各測定区間の実行性能の比率は、ACS と MPC のシステムの違いでは大きく変わらず、どちらもノード当たりの問題規模を大きくすれば性能が良くなり、小さくすれば性能が悪くなるということがわかった。この結果、格子 QCD に関しては、ストロングスケールリングが良いシステム及びアルゴリズムが必要であることがわかった。

4. まとめ

以上の結果より、ポスト「京」及びポスト「京」時代のスーパーコンピュータにおけるターゲットアプリケーションの性能予測ができ、アルゴリズムを改良するための手がかりを得た。

5. 今後の計画・展望

来年度以降も継続して本課題に取り組んでいくが、計算科学研究機構に設置する FX100 等を利用して行う予定である。

6. 利用がなかった場合の理由

該当しない。