

課題名(タイトル):

FX100 を用いたチェックスイート性能評価

利用者氏名:

○南 一生(1,2), 黒田 明義(1,2), 熊畑 清(2,1)

理研における所属研究室名:

(1) 計算科学研究センター 運用技術部門 チューニング技術ユニット

(2) 計算科学研究センター フラッグシップ 2020 プロジェクト アプリケーション開発チーム

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

本課題では、「京」並びにポスト京での利活用を促進するために、スーパーコンピュータ「京」(以下「京」と記す)開発時に作成したチェックスイートやサンプルアプリケーションコードを用いて、FX100 や GPU での性能解析を行い、システム性能の評価、チューニング効果の評価、チェックスイートの問題点の洗い出し、新たなチェックスイートの試作などを行うことを目的に利用を行った。

ここで用いたチェックスイートとは、HPC で利用される計算科学アプリケーションから、主要な計算箇所をループ単位の粒度で切り出したものである。切り出しに用いたアプリケーションは、各分野から、アルゴリズム特性に応じて幅広く選択されている。また用途に応じて、単体チェックスイート、並列チェックスイートに分類出来る。

チェックスイートは、「京」開発当初、システム評価並びにアプリケーションのチューニング指針を得ることを目的に構築された。「京」の運用では、システム(OS, ミドルウェア等)のエンハンス/メンテナンスに伴うアプリケーションの性能劣化を「未然に防止する」「検知する」ことを目的とし、以下の機能整備を行った。

- (1) 複数システム(マシン, コンパイラ)動作
- (2) 計算結果のバリデーションチェック機能
- (3) 性能検証の比較結果レポート機能
- (4) 一括実行機能
- (5) ジョブ多重度管理機能
- (6) 評価カーネル選択機能

現在開発が進められているポスト「京」のスーパーコンピュータ開発では、最新システムで測定された性能をもとに次世代システムの性能評価を行う。このため、現段階で最

新のシステムである FX100 並びに GPU を代表とするその他のアーキテクチャを使用した性能評価が重要である。本課題では、現状のチェックスイートを用いて、性能評価を行い、そこから問題点を洗い出し、チェックスイートの再整備を行った。更に実運用で発生した問題の解決に向けたテスト計算を実施した。

2. 具体的な利用内容、計算方法

(1) FrontFlow/blue カーネルの評価

FrontFlow/blue(以降 FFB)は東京大学革新的シミュレーション研究センターで開発された有限要素法による流体シミュレーションコードである。本アプリケーションはものづくりの現場で実用に供されており、今年度も引き続き幾つかの主要演算部分(カーネルと呼称)に着眼して「京」と比しての性能評価ならびに高度化の検討を行った。

(2) CUBE カーネルの評価

CUBE は理化学研究所計算科学センターで開発している有限体積法による流体シミュレーションコードである。今年度から新たに本アプリについてもカーネルに着目して「京」と比して性能評価ならびに高度化の検討を行った。

(3) cuDNN の機能並びに性能調査

cuDNN は、NVIDIA が提供している Deep Learning による機械学習フレームワークであり、主要な演算を高速化するためのライブラリである。畳み込み計算や活性化関数の計算機能が提供されている。今年度は、「京」やポスト京での Deep Learning の利活用促進に向け、本ライブラリで提供されている機能や性能について調査した。

(4) 汎用 convolution カーネルの試作と調査

Deep Learning による機械学習で利用される計算環境の

多くは、高速化のための専用ライブラリが用意されているが、「京」や FX100、ポスト京などの特殊な環境では、汎用的な手法を用いた計算が必要である。今年度は、「京」やポスト京での Deep Learning の利活用促進に向け、専用ライブラリの機能や性能調査結果をもとに、FX100 でも動作する汎用高速化ライブラリを試作し、その性能を調査した。

3. 結果

(1) FrontFlow/blue カーネルの評価

本コードでは運動方程式の計算、連続の式の計算、修正という 3 つのステージを繰り返し処理することで流体場の時間変動を計算している。本年度は、運動方程式計算カーネルの改善を行った。

このカーネルでは要素が持つ値に係数を乗じ、その要素が頂点として参照している複数個の節点に足し込むという処理を行っている。各節点には、自身を参照している複数の要素からの値が足し込まれるのであるが、元々の実装では要素番号順のループを回して、各要素が持っている、頂点として参照している節点のリストを参照して値の足し込み先の節点を決め足し込んでいた。このためメモリアクセスの観点ではストア先の配列やランダムアクセスとなり効率が悪いという特長があった。

このようなアクセスパターンに対しては、節点番号順のループを回すようにループ構造を変更することが効果的であることが過去に行った、連続の式の計算で用いられるカーネルの改善により判っている。そこで本年度は運動方程式カーネルに対してもループ構造を変更する改善を行った。これによりストア先配列へのアクセスが連続アクセスとなり、また、ストア命令を発効する回数も削減でき、実行時間を半減できた。

(2) CUBE カーネルの評価

開発者による調査によって本コードでは非粘性流束計算、粘性流束計算、時間発展計算の 3 つのカーネルが計算時間に支配的であることが判っており、これらのカーネルの改善が課題である。

今年度は最も計算時間を占めている非粘性流束カーネルの改善を試みた。本カーネルは直方体セルの 6 つの面それぞれでの 5 種の値の数値流束を計算する処理を行っており、演算量が大きい、それと比してメモリアクセス量は小さく、キャッシュに載っているという特長がある。プロファイラによる特性分析では演算スケジュールネックに陥っている

ことが明らかとなった。これは最内ループ内の命令数が多いことでソフトウェアパイプラインが適応されなかったことが主因であると考えられた。

そこで長い最内ループを複数のより小さなループに分割することで、ソフトウェアパイプラインを適応させるを試みた。一連の計算処理を途中で分割するため、一時配列に構造のループで必要となる値を保存する必要がある、適切な位置で分割しないとより一層計算時間を要してしまう。ここでは数パターンの分割について検討を行ったが、有意な改善結果は得られておらず現在調査中である。

(3) cuDNN の機能並びに性能調査

Deep Learning の主要な演算は各レイヤーの畳み込み計算である。この畳み込みの大きさなど特徴的なパラメータは、学習・推論で使われるネットワーク固有のものであり、多岐に渡る。ここでは多様な畳み込み計算として、Baidu Research が提供している Deep Bench を使用して、いくつか画像解析で使われる ResNet のパラメータを追加し、100 余ほどの畳み込みパラメータを用いて cuDNN の機能や性能の調査を実施した。

cuDNN では畳み込み計算に対して、gemm、FFT、Winograd といったいくつかの計算手法を提供し、その中で最速のもので計算するようになっている。Deep Bench などのパラメータを用いて自動で選択されたアルゴリズム一覧を表に示す。多くのパラメータで gemm が適用され、フィルターサイズが大きいものなどで Winograd が、一部のパラメータで FFT が使われていることが分かった。またバッチサイズを大きくすると FFT が、より多く選択されることが分かった。

プロファイラ nvprof を用いて詳細な解析を行うと、ここで使われるパラメータは事前に 1 回計算することで、速いものを選択していることが分かった。また FFT などの手法は、全体的に高速であるのではなく、いくつかの特徴的なパラメータで高速であることが分かった。

(4) 汎用畳み込み計算カーネルの試作と調査

「京」や FX100 でも計算することができる汎用畳み込み計算カーネルを作成した。作成したカーネルは cuDNN や Intel MKL DNN での調査結果をもとに、direct 法、gemm 法、FFT 法、DFT 法、Winograd 法、Winograd tiling 法などで、それぞれについて何パターンかチューニングを施した。また学習用の backward propagation にも対応し、活性化関数

についてもカーネルを試作した。本カーネル集は、ScaLAPACK や FFTW など汎用的な数値ライブラリのみ使用しており、「京」、ポスト京以外にも Windows 上の cygwin などでも動作することを確認している。

表に HOKUSAI FX100 での効率の結果を示す。gemm を用いた畳み込み計算では、効率最大 51%、平均 21%程度と「京」での性能最大 78%、平均 42%と比べて幾分低めであるといえる。FFT の選択率は「京」での汎用畳み込みカーネル並びに cuDNN での測定と同程度であった。

表: cuDNN の選択アルゴリズム(表左)と、汎用畳み込みカーネルの gemm 性能(表右)

Forward algorithm name	Kepler K20X			peak ratio [%]	
	cuDNN6.0	cuDNN7.3	N=32,64		gemm
GEMM	2	0	1		
IMPLICIT_GEMM	33	43	37	max.	51.01
IMPLICIT_PRECOMP_GEMM	41	30	33	ave.	20.91
FFT	1	2	2	min.	3.42
FFT_TILING	0	0	6		
WINOGRAD	0	0	0		
WINOGRAD_NONFUSED	31	34	29		
sum	108	109	108		

4. まとめ

「京」で構築されたチェックスイートやテストコード、テストサンプルを用いて、HOKUSAI GWMPG や ACSG での評価を行った。これらの解析を詳細に進めることで、カーネルの更なるチューニングの知見が得られた。FrontFlow/blue や CUBE のカーネルでの結果はその代表である。また Deep Learning の利活用を目指した調査により、「京」やポスト京でも先行している GPU などに対して、十分対抗できる性能を達成でき、実用的な計算が可能である見込みであることが分かった。

5. 今後の計画・展望

今回の課題実施によって得られた知見をもとに、更なる詳細の解析を行うとともに、より汎用かつ有用なチェックスイートの整備に力を入れ、京、FX100 並びにポスト京の利活用促進に邁進する予定である。

6. 利用がなかった場合の理由

該当なし