

科学研究費助成事業 研究成果報告書

平成 27 年 6 月 8 日現在

機関番号：12601

研究種目：基盤研究(C)

研究期間：2012～2014

課題番号：24500030

研究課題名(和文) 科学技術計算に有用な超高速4倍長演算に関する研究

研究課題名(英文) Research on High Speed Quadruple Precision Operations for Science and Engineering

研究代表者

金田 康正 (KANADA, Yasumasa)

東京大学・情報基盤センター・教授

研究者番号：90115551

交付決定額(研究期間全体)：(直接経費) 4,100,000円

研究成果の概要(和文)：倍精度数よりも有効桁数が多くなる4倍長数、あるいはそれ以上の制度を有する高精度演算の必要性が益々高まり更なる計算精度の向上が必要となっている。本研究は、倍精度演算を越える「4倍精度演算のさらなる高速化と実用化」そして「フリーソフトウェアとしての成果物(ソフトウェア)の公開」を目的としての研究である。

3年間に渡る研究で、(1) 4倍精度(以上)の演算を簡単に記述できるとともに、(2) 高精度数値計算ソフトウェア開発に適した新しい計算機言語の細かな修正と最終設計を終えた。(3) 実際に使いものになるトランスレーターや最適化ソフトの作成を行ったが、機能、性能の点から問題が残っている。

研究成果の概要(英文)：Double precision operations in the HPC are the major computing mode for the numerical computations in the science and engineering. This research focus on the higher precision computing more than double precision operations, i.e. quadruple precision operations. One of final targets of this research is free distribution of such software obtained through the research. Under three years of research period, new computer language specification with features of (1) easy description of more than quadruple precision operations, (2) good for high performance computing, is fixed. (3) Now is the stage of creating translator of such new computer language. However the translator is insufficient for the real usage in the functionality and performance point of views.

研究分野：高性能計算

キーワード：数値計算 高速計算 トランスレーター 数値計算ライブラリー コンパイラー 4倍長演算 大規模計算 並列数値計算

1. 研究開始当初の背景

数値計算法や並列計算機の性能向上の進展は大規模科学技術計算の発展における重要な鍵となっている。特に数値計算分野での浮動小数点数演算は計算機の中で有限桁に丸められることから、よほど特殊な演算で無い限り演算毎に誤差が発生する。

さて計算機の中の浮動小数点数の有効桁は、現在広く用いられている倍精度数の場合 10 進で約 16 桁である。演算毎に誤差が発生していると考え、大規模計算では最終結果にある程度 (少なからず) の誤差が含まれることが容易に想像できる。その計算誤差がどれくらい含まれるかは問題並びに計算手法依存であり、理論的に求まる計算誤差の値と大きくかけ離れていることが現実にはしばしば生じる。その結果、計算規模が増大するに従って多くの計算量、即ち長時間に渡る計算時間を必要とし、結果として計算誤差も増大することになる。

実際浮動小数点数に基づく数値計算法の中には、計算規模が増すに従って計算誤差が増大し、より多くの計算を必要としてしまうものもある。例えば線形方程式 $Ax=b$ の解法であり、クリロフ部分空間を用いる CG 法などは誤差の影響を大きく受ける。また計算機における浮動小数点数演算では実数を一定の有効桁で近似してしまうため、丸め誤差が発生する。そのために、倍精度データ型から仮数部を増やした 4 倍精度データ型を用いた 4 倍精度演算の必要性が高まってきており注目されている。

現実に京を代表とする高性能並列計算機の性能向上に対する要求は衰えを知らず、毎年演算性能は上昇する一方である。しかし残念ながら、浮動小数点数演算ハードウェアは、どのような理由か分からないが、旧態依然の倍精度演算しかサポートしていないのが現実である。その様な「いびつ」と言って良い状況にあるために、倍精度数よりも有効桁数が多

い 4 倍精度数、あるいはそれ以上の精度を有する高精度演算の必要性また重要性が近年益々高まってきており、計算規模が大規模になるに従って更なる計算性能向上が必要となっている。

2. 研究の目的

当初本研究は、上記研究背景に基づき、それまでの科学研究費に基づく研究成果の上に、(1) Intel や AMD による x86 アーキテクチャーまた IBM POWER7 アーキテクチャーにおける超高速な 4 倍精度演算の実現、(2) GPGPU 上の超高速な 4 倍精度演算の実現、(3) 4 倍精度三角関数や指数関数等の 4 倍精度算術基本関数の超高速化、(4) FFT、連立方程式の解法、固有値の求解等の基本ライブラリーの超高速 4 倍精度化、(5) 実アプリケーションへの適用、(6) 4 倍精度を超える高精度演算の実現、(7) 得られたソフトの Web 公開、のそれぞれを実現することを目的としていた。

実際の研究推進にあたっては、京を代表とする計算機の性能向上が著しい事を考慮し、まず優先すべき目標として、倍精度演算を越える「4 倍精度数演算のさらなる高速化と実用化」、「4 倍長精度数が記述し易い計算機言語の設計とトランスレーターの作成」そして完成したソフトウェアを「フリーソフトウェアとしての公開」へと目的を少し変更した。

3. 研究の方法

本研究で使用する 4 倍精度数の表現として、完成させるソフトウェアの汎用性を高める観点から、当初の科学研究費申請書にそれまでに研究成果として記載の研究発表論文に示すように、これまでの事前研究で得られた成果である倍精度数を 2 個用いて表される 128 ビットの浮動小数点データ型を採用する。この方法に基づく 4 倍精度演算はソフトウェアによって 2 つの倍精度数を使用して実現 (エミュレート) するため、倍精度演算に比べ多くの計算回数を必要とする。(加減

乗算で約 10 倍、除算で 2 ～ 3 倍の時間を必要とする。) このことは言い換えると、計算精度を向上させると計算時間が増大してしまうという問題点がある。しかしこの実現方法は、既存の各種計算機アーキテクチャー上でも実現が容易な、汎用性が高い方法である。

このことは精度と計算時間がトレードオフの関係にあり、基本的な計算性能向上はハードウェアによる 4 倍長演算の実現を待つ必要がある。過去にはハードウェアによる 4 倍長演算を実現していた計算機は存在していたが、残念ながら現時点ではそのようなハードウェアを有する高性能計算機の開発計画は存在せず、当面はソフトウェアによるエミュレーションを利用せざるを得ない状況になっているのが現状である。その為本研究では、ソフトウェアによるエミュレーションを中心技術として採用することとする。

さて CPU コア数を増やして並列処理することで演算性能を上げる現状の並列計算機の 4 倍精度演算は、データアクセス回数より演算回数の割合が多いという性質がある。そこで本研究では、4 倍精度演算アルゴリズムの定量的な解析を行なうことで 4 倍精度演算が抱える本質的問題点を明らかにする。その後、高速化手法として最近のプロセッサで実現する積和演算命令(FMA, Fused Multiply-Add)を用いる方法で 4 倍精度演算の演算回数を削減するとともに、丸め誤差の低減を実現する。その上で、東京大学情報基盤センターがサービスを行っている大規模計算で先端的に利用されていた並列計算機、SR11000 モデルJ2(以下SR11000/J2)上のCPUである POWER 5 系アーキテクチャーに特化したさらなる最適化を行う。その後広く利用されている、Intel あるいは AMD の両社が製造している汎用マイクロプロセッサ向けの移植と最適化を行う。また最近注目を浴びているグラフィックプロセッサ (GPGPU) による 4 倍長演算の実現および高速化を試みる

と同時に、高性能なランタイムルーチンに基づき、三角関数、指数関数、対数関数等の基本的な数学関数の 4 倍精度化、高速フーリエ変換 (FFT) や連立方程式の解法、固有値の求解等のライブラリーの 4 倍精度化を試みる。いずれにおいても、研究で得られたソフトを一日平均 30 件以上 (10年間の長期間に渡る平均値) のアクセス頻度を誇る研究室のホームページ経由で公開することが最終目標である。

これまでに実施されている 4 倍精度演算に関する研究として、Bailey や Hida らは倍精度浮動小数点数を組み合わせた

「double-double」精度のアルゴリズムを Fortran や C++ などで実現し、四則演算だけではなく、三角関数や対数関数、指数関数なども提供している。またそれらを QD(Quad Double) ライブラリーとして公開している。QD ライブラリーには、「double-double」精度の 4 倍精度数を用いて、4 倍精度と倍精度の演算や、倍精度浮動小数点数を 4 個用いた「quad-double」精度で 8 倍精度演算を実現している。しかし倍精度数をさらに上位と下位部分に分割して計算するライブラリーで用いられているアルゴリズムは、積和演算命令を使用しておらず、事前研究で成果を得ている 4 倍精度乗算の約 2 倍の演算回数を必要としてしまう。積和演算を用いた 4 倍精度乗算もあるが、演算回数は多くなってしまっているという欠点がある。さらに日本で使われているスーパーコンピューターである、POWER 7 アーキテクチャー向けに最適化されていない。また Intel や AMD のプロセッサが実現している SSE2、SSE3 等の SIMD 命令を用いた高速化も実現していない。

また 4 倍精度演算を線形計算や反復解法に応用した研究にいくつかの研究がある。まず 4 倍精度演算をサポートしているライブラリーとして、GEMM++(Generic Matrix Methods)がある。GEMM++ は C++ で記述され

た密行列と疎行列に対するテンプレートライブラリーで、4 倍精度演算については QD ライブラリーを外部ライブラリーとして呼び出している。Langou らは倍精度演算の直接解法で得られた解を 4 倍精度演算で反復することで、4 倍精度演算の直接解法より高速に解を求められることを示している。さらに Kotakemori らは直接解法ではなく、反復解法を用いた倍精度と 4 倍精度の混合型反復解法を提案しており、その混合型反復法は反復法ライブラリー Lis で実装されている。また一組の倍精度数を SSE2 命令を用いて高速化している。それら研究に対し、本研究はこれまでに実施してきた基本となる4倍精度演算の高速化の上に研究を行うという点でこれまでの研究と異なる。

4. 研究成果

これまでの研究に基づく提案手法で、まず 4 倍精度演算アルゴリズムを定量的に解析し四則演算などの演算回数を削減している。POWER 5 アーキテクチャーにおいては x86 アーキテクチャーに実装されていない積和演算命令を実行できることから、誤差を発生させずに高い有効桁を保ったまま演算できるという特徴を利用した演算回数の削減である。またそれに加え POWER 5 アーキテクチャーに特化した浮動小数点演算演算の最適化を実現した。そもそも 4 倍精度演算は一回の演算に大きな計算コストを必要とするため、本質的にデータ転送がボトルネックになりやすく、データ転送遅延を考慮せずに演算のみを最適化することができるという特性がある。その特性を利用した主な最適化手法として、浮動小数点演算命令のレイテンシー隠蔽と 64 ビット浮動小数点レジスタ数の考慮という 2 点がある。レジスタ数に収まる範囲内でループ展開を行い、命令の順序を入れ替えて浮動小数点演算命令のレイテンシーを隠蔽するわけである。レイテンシーを隠蔽すること

により CPU はストールせずに実行できるので高速化が可能となる。

これらに示す最適化を行った結果、並列計算機 SR11000 モデル J2 上の gcc4.1.1 に対して提案手法が最大で約 5 倍の速度向上、また SR11000/J2 最適化コンパイラーを上回る性能をこれまでに得ている。ループ展開また浮動小数点演算命令のレイテンシー隠蔽には、連続データの同一パターンによる演算が必須であることから、これはベクトルデータに対する最適化されたランタイムの実現が本質的であることが容易に推察できる。また逆に言うならば、ベクトルデータに対する4 倍精度演算の高速化は十分な効果が期待できることが判明している。

本研究が目指す最終目的の一つとして、4 倍精度以上の数値データに対する計算プログラムを簡単に記述できる新しい計算機言語の設計と実際に使い物になるソフトウェアシステム（コンパイラーあるいは別言語へのトランスレーター）の作成がある。4 倍精度数を高速にサポートできるランタイムルーチンが完成したとしても、複雑な 4 倍長演算の全てをサブルーチン呼び出しで記述するのは、プログラミングの立場からすると非常に障壁が高く、とても使いにくいものとなることは必須である。それで 4 倍長（以上）の精度の数値演算を必要とするプログラムの記述が容易な、真に実用となる新しい計算機言語の設計が必要であり、実際に実用となることを目指し、これまでの 3 年間の研究期間の大半をこの新言語の設計と、実際のトランスレーターの（プロトタイプの）作成に充てた。その為に、研究成果の対外発表件数はごく少数となってしまった。しかし実際に即した十分な検討の結果、新言語のおおよそのスペックを決定し、不十分ではあるが最適化したソースを生成するトランスレーターのプロトタイプを作成し、そのシステムの動作確認を行っている段階で 3 年間の研究期間は終了してし

まった。研究期間の終了である 2015 年 3 月末をもって勤務先の東京大学を定年退職となるが、年退職後も引き続き新言語のトランスレーターの作成を継続し、十分な安定性と高度な最適化能力を獲得した後に、ホームページ等で広く公開を行う予定としている。

なホームページや他のデータサーバーはそれまでの勤務先のサーバーとは異なるサーバーへデータや環境の移動が必要となり、現在その環境整備中である。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 0 件)

[学会発表] (計 1 件)

・ Yasumasa KANADA, Computing pi : the story of my pi computation from 1980 onwards, 2013年 12 月 17 日、Univ. of Loughborough (Loughborough, England) (招待講演)

[図書] (計 1 件)

金田 康正、ウエッジ、スパコンとは何か - 1 位か 2 位か、それが問題か? -, 2012 年、220p.

[産業財産権]

○出願状況 (計 0 件)

名称 :
発明者 :
権利者 :
種類 :
番号 :
出願年月日 :
国内外の別 :

○取得状況 (計 件)

名称 :
発明者 :
権利者 :
種類 :
番号 :
出願年月日 :
取得年月日 :
国内外の別 :

[その他]

ホームページ等

www.super-computing.org

6. 研究組織

(1) 研究代表者

金田 康正 (KANADA, Yasumasa)

東京大学・情報基盤センター 教授

研究者番号 : 90115551

(2) 研究分担者

()

研究者番号 :

(3) 連携研究者

()

研究者番号 :