

ユーザー支援報告

寺崎順

筑波大数理物質系
計算科学研究センター

1. やること、メンバー、依頼者にとっての利点
2. 平成26年度の依頼と回答11件中の抜粋
3. 今年度の進歩と成果
4. 依頼のしかた
5. まとめ

2015年 3月

1. やること

物性・素粒子・原子核・宇宙研究の数値計算に関する可能な限りあらゆる問題を対象に、ユーザー支援チームメンバーが個々の支援要請に応じて検討し、助言・提案を行う。集まった計算技術情報を公開し分野の人々の間で共有する。

メンバー：総勢27名、うち戦略プログラムによる被雇用人員18名。自分の研究を行いながら、ユーザー支援を行っている。

専門分野	人数
素粒子	12名
計算システム	5名
宇宙	5名
原子核	4名
技術支援	1名

所属	人数
KEK	10名
筑波大	8名
京大	3名
千葉大	2名
ほか	4名

依頼者にとっての利点

いろいろな専門の豊富な経験をもつ支援員が結集している

- Fortran, C, C++, MPI, OpenMP, Cuda, IDLなどをカバー、
- 物理はもとより応用数学、可視化もカバー



ひとりまたは限られたグループでの議論では解けないプログラミング上の問題が多くの場合解決できる。もし解けなくても何らかの情報が得られる。

- ・ 大規模計算や並列化に限らず、どんなに些細な問題でも扱う。
- ・ 最先端のプログラムを開発中の研究者の方々はもとより院生の皆さんからの依頼も歓迎。



依頼しやすい。

2. 平成26年度の依頼と回答7件の要約(平成27年1月20日現在)

1	問題	銀河衝撃波の再現において数値振動が発生し計算が破綻してしまう。どうすればよいか。
	解決法・参考情報の要約	残念ながら、最先端の研究レベルで根本的解決方法が確立していない。
2	問題	時間変化に乏しい関数 $F(t)$ と球ベッセル関数 $j_l(t-t_0)$ との積に関する定積分を、倍精度実数の範囲で精度よく行うためのアルゴリズムを知りたい。
	解決法・参考情報の要約	本件に関わる文章として、京都大学数理解析研究所の講究録にある【Bessel 関数を含む振動積分に対する数値積分公式】【部分積分法による数値積分法】を紹介した。また、実在する類似のライブラリとして、関数 $F(t)$ と三角関数のとの積の定積分を精度よく行う qawo ルーチン (QUADPACK, GSL) があることを伝えた。

3	問題	QCDのシミュレーションで大きなゲージ配位($10^3 \times 64$)をファイルへ書き出すのに時間がかかっている。MPIを用いて高速化したい。データの中身は変えないようにする。
	解決法・参考情報の要約	書き出しのためのループをひとつ減らし、今まで一つのプロセスが書き出しを行っていたのを、複数のプロセスが異なるファイルへ並列に書き出すような構造のプログラムにした。プロセス番号をある数で割った余りによって書き出しプロセスが定義される。3時間10分かかっていた書き出しが6分で終わるようになった。
4	問題	磁気流体シミュレーションコードへ非一様メッシュを導入する際、数値補完と磁場の発散の処理の方法を知りたい。
	解決法・参考情報の要約	依頼者の方法で期待される空間精度は、非一様メッシュ幅の場合、隣同士のメッシュ幅の差の2乗程度で一様メッシュの場合と比べて低くなる。メッシュ幅の変化率はあまり大きくならないように、注意が必要である。また、数値的な磁場の発散誤差の処理では、依頼者が使用するアルゴリズムでは、シミュレーション空間の最大／最小メッシュ幅の比が大きいと効率的に磁場の数値的発散誤差をとりのぞくことができない。これらの点に注意して、比較的単純な次元の波の伝搬テストから始めて、コードが正しく実装できているかどうか試してはどうかと提案した。

5	問題	十万次元くらいのエルミート行列を用いた連立方程式を解くのに適した方法またはサブルーチンを知りたい。
	解決法・参考情報の要約	担当者が高性能計算の扉に掲載された線形方程式解法の情報参照しながら検討中。
6	問題	コード開発をチームで行っている。バージョン管理など共同作業体制づくりのノウハウを知りたい。
	解決法・参考情報の要約	バージョン管理については、最近では「Git」や「Mercurial」と呼ばれる分散管理システムを利用して管理することが主流になっている。 プロジェクトの運用に関しては、Web上で柔軟なプロジェクトの運用管理ができるシステムに「Redmine」がある。 共にプロジェクトの運用方針によって具体的な使用方法が変わってくるので、今後、依頼者と相談の上適切な方法を提案する予定である。
7	問題	ユニタリ行列 U を $U=ADB$ とする分解(A, B は実直交行列と D は対角に位相が並ぶ対角行列)は可能か。可能ならばどのようにして数値的に準備すれば良いのか。
	解決法・参考情報の要約	まず、素粒子標準模型の小林-益川行列を双ユニタリ行列で対角化するときと同様のテクニックを用いることで、このような分解ができることを示した。さらに、その証明で用いた手法を数値的に準備する方法について解説した。

2. 計算高速化実践事例6件の要約(平成27年1月20日 現在)

1	項目・計算機・使用言語・課題	「京」コンピュータで計算を行う上でのTips。 Fortran, MPI, OpenMP。 超新星爆発。
	内容の要約	「京」のコンパイラは最内ループを見つけてそこを最適化しようとする。SIMD最適化のため最内ループを明示することが望ましい。コンパイルオプションやプロファイラの活用も推奨される。「京」上のallreduceは当初不安定であったが、リダクションのブロック化で安定して実行できるようになった。
2	項目・計算機・使用言語・課題	「京」コンピュータでの計算情報。 Fortran, OpenMP。 原子核構造。
	内容の要約	行ったチューニングの要点は以下のようなものである。アルゴリズムをできる限り密行列積で記述しBLASを使う。OpenMPでのスレッド中Fortranの「自動配列」を排除する。計算の大規模化のため並列階層を増やす。(並列計算で求めるものを増やす。) プロファイラの活用。

3	項目・計算機・使用言語・課題	「京」コンピュータ上での高速化についてのTips。C、格子QCD。
	内容の要約	レジスタの効率的利用。このためのスカラー変数の有効利用、ループアンローリング、レジスタあふれの防止、京のSIMD用拡張変数型とそれを引用する組み込み関数の利用。富士通の低レベル通信関数の利用。
4	項目・計算機・使用言語・課題	「京」/FX10を利用する際の注意点。C/C++。格子QCDによる核力。
	内容の要約	可変長配列のメモリ確保/解放のタイミングがプログラムの期待と異なる振る舞いをするため、スタック領域を食い潰してプログラムがクラッシュするという問題があった。対策：たとえサイズが小さくても、可変長配列を自動変数としてスタックに取るのはやめ、malloc/free, new/delete 等でヒープ領域に明示的に確保・解放する。あるいは、可能であれば、自動変数のままでもループの外側で定義すれば問題は軽減される。

	項目・計算機・使用言語・課題	CUDAでの高速化についてのTips。格子QCD。
5	内容の要約	<p>レジスタの効率的利用に関して、事例3のTipsがあてはまる。1ブロック内で多くのスレッドを生成する場合1スレッドあたりで使用できるレジスタ数は少なくなる。スレッド長さとかernel内部の複雑さをコーディングの際に調整する必要がある。</p> <p>tex1Dfetch などの関数を通じてデータを読み出すと読み出し専用キャッシュであるTextureキャッシュにデータが残り計算速度の向上に有効である(2-3割速くなる場合有り)。</p>
	項目・計算機・使用言語・課題	Hitachi SR16000を利用する際の注意点。C/C++。格子QCD。
6	内容の要約	<p>事例4の可変長配列の問題がこの計算システムでも起きる。対処法は同じ。スレーブスレッド用にシステムがデフォルトで確保するスタック領域が小さすぎる為、スタックオーバーフローでプログラムがクラッシュするという問題もある。対策:実行前にスレーブスレッドのスタックサイズを拡張しておく。</p>

以上の計算高速化実践事例の情報提供は以下の方々からいただきました。
厚く御礼申し上げます。

1. 滝脇和也様 (国立天文台)
2. 清水則孝様 (東京大)
3. 石川健一様 (広島大)
4. 土井琢身様 (理研)
5. 石川健一様 (広島大)
6. 土井琢身様 (理研)

3. JICFuSレポート 各種計算機性能比較調査

(KEK、濱口)

多倍長精度計算、多次元積分、線形方程式解法、N体問題、量子モンテカルロ法への応用等の計算性能を、SR16000, AMD GPU, Xeon Phi 等様々なアーキテクチャの計算機上で実行し、高速化の方法とその効果について議論している。- 各種計算機基本性能調査 - 各種計算機アプリケーション性能比較としてそれぞれ毎年度4回のレポートとしてまとめられ、公開されている。

3. 今年度の進歩と成果

- 本年度は8件の支援依頼を受け付けた。3時間余りかかっていた処理が6分に短縮されたという事例は本活動を審査する作業部会において高く評価された。
- 計算高速化事例を広く募集し、公表した。
- 各種計算機およびアプリケーションの性能調査は膨大な資料に発展した。