解説

京速コンピュータ「京」におけるアプリケーション高性能化

Performance Improvement of Application on the “K computer”

南　一生

Abstract

現在，理化学研究所において，2012年の完成を目指し京速コンピュータ「京」が開発されている．本システムは，8万個以上のCPU（ノード），64万個以上のプロセッサコアから構成される超並列計算機であり，各プロセッサコアには，高速計算のための多くの新しい機構が導入されている．理化学研究所ではプロジェクトを進めつつ，多様な応用分野のアプリケーションにおいて，「京」の性能を十分に生かし切るために，高並列化とCPU単体性能向上の両面からアプリケーションプログラムを高性能化する研究開発を実施している．本稿では，その概要について紹介する．

キーワード：スーパコンピュータ，並列化，チューニング，高性能化

1. はじめに

2011年6月20日（月），ドイツ・ハンブルクで開催された第26回国際スーパコンピューティング会議ISC '11において，日本製のスーパコンピュータ「京（けい）」(注1)が第37回TOP500リスト第1位を獲得した．TOP500リストとは，LINPACK（リンパック）と呼ばれるベンチマークプログラムの実行速度を指標として，世界で最も高速なコンピュータシステムの上位500位までを定期的にランク付けするプロジェクトである．今回の快挙は，日本製のスーパコンピュータとしては，初代地球シミュレータが2004年11月に第1位を明け渡してから7年ぶりの返り咲きだったこと，第2位となった中国製スーパコンピュータ「天河1A」の性能と比べて，約3倍の8.162PFLOPS（1秒間に8,162兆回の浮動小数点演算能力）のLINPACK性能を達成したことなどをはじめとして，世界に大きなインパクトを与えた．

応用面から見ると，「京」は，極微細な量子の世界から膨大な数の銀河を内包する宇宙まで，広大なスケールの中に存在する様々な自然現象を解明し社会に貢献する革新的な手段として期待されている．例えば，ナノスケールでは，次世代エレクトロニクス革新の加速，また，数十mから数百kmという人間社会のスケールでは，地震波動伝搬と構造物の応答を組み合わせた耐震シミュレーションによる詳細な防災計画への貢献，地球規模の数千から数万kmのスケールでは，正確な台風の進路・強度の予測や気候変動研究への貢献等が期待されている．更に大きな10^{20}m以上の宇宙現象の解明も期待されている．

2. プロジェクト概要

日本では地球シミュレータの開発プロジェクト以降，大規模なスーパコンピュータ開発プロジェクトが立ち上がらなかった．そのため，スーパコンピューティング技術及び計算資源の量において，米国などの後じんを拝することになった．そんな中，将来にわたり，日本の高度な研究開発能力や産業競争力を維持，発展させるためには，日本もスーパコンピューティング技術の強化が必要不可欠であるとの認識から，スーパコンピューティング技術は，第三期科学技術基本計画の「国家基幹技術」に位置付けられた．この結果，平成18年（2006年）度に文部科学省の「最先端・高性能スーパーコンピュータの開発利用」プロジェクト（通称，次世代スーパーコンピュータプロジェクト）が7か年計画で開始され，独立

南　一生　独立行政法人理化学研究所次世代スーパーコンピュータ開発実施本部
E-mail　minami_kaz@riken.jp
Kazuo MINAMI, Nonmember (Next-Generation Supercomputer R & D Center, RIKEN, Kobe-shi, 650-0047 Japan).
電子情報通信学会誌　Vol.95 No.2 pp.125-130 2012年2月

(注1)「京」（英語表記はK computer）は，開発当初「次世代スーパーコンピュータ」と呼ばれていたシステムの愛称．2010年7月に約1,500通の応募の中から選ばれたもの．（参考URL：http://www.nsc.riken.jp/aisho/kekkahoukoku.html）

行政法人理化学研究所（以下，理研）がその主たる開発を行うことになった．2010 年 9 月末に最初の「京」のきょう体が搬入され，2011 年 9 月までに全システムの設置が完了した．今後「京」は，システムソフトウェア等の調整・改良を経て，2012 年 6 月に完成し，2012 年 11 月には供用を開始する予定である．

3.「京」のシステム概要とプロセッサの特徴

「京」は全体としては 8 万個以上の CPU から構成される超大規模システムであるが，最も小さな構成単位はノードと呼ばれている．ノードは，8 コアから構成される CPU，通信用 LSI 及びメモリから構成され，1 秒間に 1,280 億回の演算性能と 16 GByte のメモリ容量を持っている．このノード四つが一つのシステムボードの上に実装され，更にシステムボードは計算機ラックに 24 枚搭載されている．「京」全体としては，計算機ラック 864 台から構成され，1 秒間に 1 京回（10 PFLOPS）以上の演算性能と，1 PByte 以上のメモリ容量を持っている．

また「京」では，優れた省電力性能と高い信頼性を実現している．「京」で採用されている SPARC64™ Ⅷfx は，チップ当りで 128 GFLOPS という高い演算性能を持ちながら消費電力は 58 W に抑えられており，他のスーパコンピュータで使用されている CPU と比較してトップクラスの省電力性能を持っている．これは，2 GHz という低い動作周波数や，使わない回路への電力供給をカットする機構など，省電力のための様々な技術によって実現されている．この CPU の駆動温度は約 30℃ と，他のスーパコンピュータに搭載されている CPU と比較しても，極めて低い温度になっており，故障率の低減に寄与している．また，1. に示した LINPACK ベンチマークプログラムの実行時間として約 28 時間かかったが，これはすなわち，28 時間連続でシステム全体を高い負荷をかけながら運転しても故障が起きなかったということを意味する．これは，他のスーパコンピュータと比較して，傑出して大きな値になっており，「京」の故障率の低さと信頼性の高さを端的に示しているといえる．

このように，「京」には世界最高レベルの演算性能だけでなく，省電力性能，信頼性，可用性，運用性など，共用システムとしての利活用を見据えた様々な機能を備えている．

4.「京」利用上の留意点

1. に示したように幅広い応用が期待される「京」であるが，その性能を十分生かし切るために，使い方において特有の留意点が存在する．計算機は，単体プロセッサの時代から，プログラムに内在する並列性をコンパイラで解釈することにより，高速処理を実現してきた．その後，スーパコンピュータが並列アーキテクチャへと変化してからは，数千から数万に及ぶプロセス間の並列性をプログラム上で明示して利用することで超高速計算を実現してきた．言い換えればプログラマは，プロセス間の並列性を意識して並列化し，またプロセスごとのデータの分散を意識してプログラミングすることが必要となった．

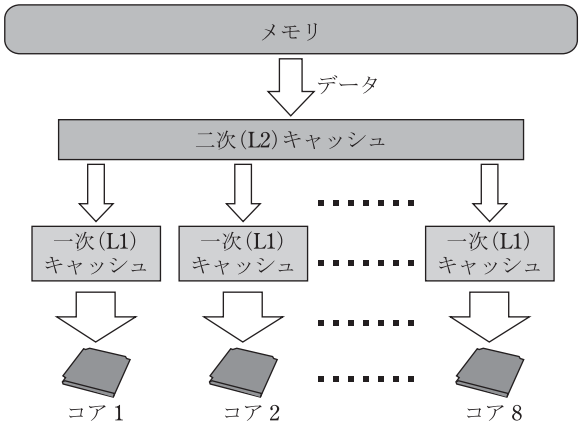

図 1 「京」の CPU 概要　CPU は 8 コアから構成される．各コアごとに一次キャッシュを持ち，8 コア共有の二次キャッシュを持つ．

一方，計算科学の初期において高級言語とコンパイラが整備されてからは，研究者やプログラマは，定式化・離散化された理論モデル式に忠実に，かつ物理現象に沿った素直なプログラミングを行うことが一般的であった．しかし，現在では，計算機自体の変化によるプログラミングの変化も生じている．メモリのデータ供給能力と演算器の計算能力が良くバランスしていた時代もあったが，現代の計算機は，演算器の計算能力が高くなる一方でメモリのデータ供給能力が相対的に不足している．この問題をメモリウォール問題という．この問題に対処するために，データ供給能力の高いキャッシュメモリ（一次／二次）を設け，キャッシュに置いたデータを何回も再利用し演算を行う方法が取られている（図 1 に「京」のキャッシュ構成を含む CPU の概要を示す）．こうすることで演算器の能力を十分使い切ることができる．このように，以前の計算機上でのプログラミングに比べ，キャッシュといった多階層メモリ構造を意識してプログラミングを行う必要が顕在化した．また，ここで述べたようなデータの再利用ができないプログラムも多く存在し，このような場合は，演算器の能力を十分使い切ることができない．そのような場合は，メモリのデータ転送能力を使い切るようにプログラミングを工夫する必要がある．

ここに述べた 2 点，すなわち「並列性を意識したプロ

グラミング」と「実行性能を意識したプログラミング」は，現代のスーパコンピュータ利用，特に8万個余に及ぶプロセッサを備え，数々の機能強化・新機能が導入されている「京」においては，ユーザ，研究者，プログラマが認識すべき非常に重要な留意点である．

5. 高並列化プログラミング

流体や構造解析のシミュレーションにおいては，空間方向にメッシュという格子を構成し，メッシュごとに計算を行う．このような例を用いて簡単に並列化について説明する．流体や構造解析のシミュレーションにおいて，並列化前の逐次計算では，メッシュごとに逐次に計算を進める．並列化するためには，メッシュを複数の領域に分割し，分割された領域を各々のプロセッサに分担させ，並列に計算を実施する．このような並列化手法を領域分割という．領域分割による並列計算においては，ある計算単位ごとに隣接するプロセッサと領域の一部のデータを交換するために隣接通信を行う．また全ての領域について内積計算を行う場合は，全プロセッサのデータの和を取るために大域通信を実施する．高並列化において重要な点は，ここに挙げた隣接通信と大域通信の時間をできるだけ小さくすることである．

次に並列化率と並列化効率について説明する．逐次計算において，並列計算できる部分が99%，並列計算できない非並列計算部が1%あったとすると，この計算の並列化率は99%である．並列化率が99%の計算で計算時間全体が100秒であったとして，100プロセッサを用いて並列計算した場合を考える．並列計算部は100プロセッサで並列計算できるため，99秒/100＝0.99秒となる．それに非並列計算部の1秒が加わるため，100プロセッサでの計算時間は，1.99秒となる．元が100秒の計算であるので，100プロセッサで約1/50の計算時間になる．この場合，100プロセッサで1/100の実行時間短縮が期待されるのに対し，1/50の実行時間短縮にとどまるため，並列化効率は50%であるという．同様に計算すると1,000プロセッサでは，約1/91の計算時間になる．つまり1%の非並列部が残っていると数十プロセッサ以上用いても効率の良い並列計算はできない．このことから，通信時間の短縮と同様に，並列計算上重要な点は，非並列計算部をできるだけ小さくすることである．

6. CPU性能を使い切るプログラミング

6.1 高い性能を得るための要素

CPU単体で高い性能を実現するためには，アプリケーションが図1に示した八つのコアを用いて高い並列化効率を実現していることが重要である．八つのコアで高い並列化効率が実現できていることを前提として，CPU単体性能向上のために重要と考えられる要素を以下に示す．

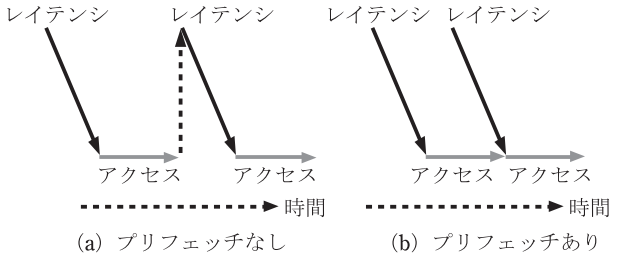

図2 プリフェッチの概要 プリフェッチによりレイテンシ（メモリ及びキャッシュへのアクセスの立ち上がり）が隠れる．

① プリフェッチの有効利用

ブリフェッチとは，データのロード命令を先んじて発行する機能である．レイテンシとは，データアクセスを始めてから実際にデータアクセスが開始されるまでの初期処理の時間のことである．図1のメモリからL2キャッシュ，L2キャッシュからL1キャッシュへのアクセスは，共にプリフェッチを有効に動作させることが高い性能を実現するために重要な要素である．プリフェッチは，ロード命令におけるデータアクセスのレイテンシを隠すものであり，プリフェッチが効かない状態でロード命令が発効されると，メモリ，L2キャッシュへのアクセスで，レイテンシ分の大きなペナルティが発生する（図2）．L1キャッシュへのレイテンシを1とするとL2キャッシュのレイテンシは10程度，メモリへのレイテンシは100程度である．

また，演算と並行してプリフェッチを行うことにより，レイテンシのみならず，メモリへのアクセスそのものを隠せる場合もある．

② ラインアクセスの有効利用

「京」のCPUでは，メモリとL2キャッシュについては，データは1ライン（128 Byte）ごとにアクセスされる．高い性能を得るためには，ロードしてきた1ラインのデータをなるべく多く使用した演算を行うことが重要である．1ラインのデータのうち例えば8 Byteのデータ1個しか使用できない場合は，1ライン分の16個の要素をロードするために16ラインのデータをアクセスする必要が生じるため大きなペナルティとなり，見かけ上のメモリアクセス性能は，1/16となる．

③ キャッシュの有効利用

$(n \times n)$ 行列同士の行列・行列積の計算を考える．二つの行列の要素は，合わせて $2n^2$ 個である．演算の数は，積と和を合わせて $2n^3$ 個である．$2n^2$ 個の要素を幾

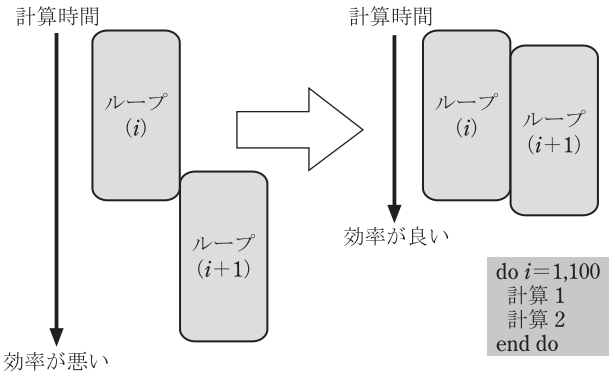

図 3　効率の良い命令スケジューリング　効率の良い命令スケジューリングでループ（i）とループ（$i+1$）の処理を重ねることができる．

つかの小さな $2n'^2$ 個の要素の組に分割し，それぞれの要素の組をキャッシュに載せて演算を実行すれば，$2n'^2$ 個のデータを再利用して $2n'^3$ 回の演算が実行できる．このような計算は，キャッシュを有効利用でき高速な計算が可能である．原理的に n 個のデータを使って n^2 回の演算ができるような計算（または n^2 個のデータを使って n^3 回の演算可能な計算等）は，ここに示した行列・行列積のような方法を使うことにより，高速な計算が可能である．

④　効率の良い命令スケジューリング

「京」は 256 本という多数の浮動小数点レジスタが装備されている．この浮動小数点レジスタをコンパイラが有効利用することにより，ループのインデックス方向の演算を重ねる演算スケジューリングが性能向上のために重要である（図 3）．コンパイラがうまくスケジューリングできない場合，手動でループ分割やループ展開をすることで性能が向上する場合がある．

⑤　SIMD 演算器の有効利用

「京」の CPU コアは，積和演算器 2 本を 2 セット備えている．積和演算（2 演算）×積和演算器 2 本×2 セットとなり，1 クロックで合計 8 演算が可能である．CPU の動作クロックが 2 GHz であるため 8 演算 ×2 GHz＝16 G 演算（16 GFLOPS）が 1 コアでの 1 秒間の演算ピーク性能である．それぞれの 2 本の積和演算器は，ベクトル長 2 の SIMD 演算器として動作する．したがって高い演算性能を実現するためには，積和演算が SIMD 化され，更に 2 セットの SIMD 演算器が同時に動作する状態になることが重要である．

6.2　高い性能を得るための要素と要求 B/F 値の関係

CPU 単体性能の面から見るとアプリケーションは，大きく二つのタイプに分類できる．一つは，アプリケーションの浮動小数点演算数（FLOP）に比べ，データ転送要求（Byte）が小さいタイプである．このタイプの計算を要求 B/F 値が低い計算という．これは，4. で述べた，キャッシュに置いたデータを何回も再利用して演算を行うことで，高い CPU 単体性能が得やすい計算である．もう一方は，逆にアプリケーションの浮動小数点演算数（FLOP）に比べ，メモリとのデータ転送要求（Byte）が大きいタイプである．このタイプの計算を要求 B/F 値が高い計算という．このタイプの計算は，原理的にキャッシュの有効利用が難しく高い CPU 単体性能が得にくい．これら二つのタイプのそれぞれについて，6.1 に示した高い性能を得るための要素のうち重要な項目について関連を示す．

要求する B/F 値が低いアプリケーションについては，原理的に高いメモリデータ転送性能（メモリバンド幅）は必要ないので，まずデータをオンキャッシュにするコーディング（③）が重要となる．次に L2 キャッシュについてはラインごとのアクセスとなるため，L2 キャッシュのライン上のデータを有効に利用するコーディング（②）が重要となる（L1 キャッシュにオンキャッシュの場合は必要でない）．それが実現できた上で④，⑤が重要となる．

要求する B/F 値が高いアプリケーションについては，CPU 演算性能を最大限使用することよりも，使用するメモリバンド幅が実効メモリバンド幅にできるだけ近いこと，つまりメモリバンド幅を使い切ることが大事であり，上記の五つの項目のうち最も重要なのは①，②である．次に一部のデータについては，データの再利用性を生かせる場合があり，その場合は③が重要となってくる．これら①～③が満たされ，計算に必要なデータが演算器に供給された状態で，それらのデータを十分使える程度に④のスケジューリングができて，更に⑤の SIMD 演算器が有効に活用できる状態であることが必要である．

7.「京」性能実証のためのアプリケーション

理研では，「京」の共用開始に先立ち，システム性能を実証するためのアプリケーション群の整備を進めている．整備の中ではアプリケーション群に対して，超並列性を最大限に引き出すとともに，プロセッサに導入された新機能や強化された機能を十分活用するためのプログラムの書換え作業を，アプリケーションの開発者と協力して進めている．これらのアプリケーション群は，「京」の汎用性を生かし，様々な応用分野のアプリケーションが幅広く高い性能を発揮できることを実証できるように選択されている．また今後の計算機開発に役立つように，計算機科学的特性の観点を含め選択した．ここでいう計算機科学的特性とは，一つは並列化の面から見て，

比較的シンプルな並列化手法で良好な並列性能が得やすいものと，複雑な並列化手法を採用しないと高い並列性能が得られないものの両極端なものから選択することである．また二つ目は，CPU 単体性能において，「京」のようなスカラ計算機では高い単体性能が得にくい傾向にあるものと，比較的に高い単体性能が得やすい傾向にあるものから選択することである．これら二つの観点を基に，地球科学分野のアプリケーションを 2 本（NICAM[1]，Seism3D[2]），ナノ分野のアプリケーションを 2 本（PHASE[3]，RSDFT[4]），工学分野のアプリケーションを 1 本（FrontFlow/Blue[5]，以下 FFB と略す），物理分野のアプリケーション 1 本（Lattice QCD[6]）の合計 6 本のアプリケーションを選択した．

（A） 地球科学分野（NICAM, Seism3D）
（a） 高並列化の観点では，比較的シンプルな領域分割の手法が用いられ，隣接通信が主であるため，高並列においても並列性能は比較的実現しやすい傾向を持つ．
（b） 単体性能から見ると，高い B/F 性能が必要とされる傾向があり，「京」にとっては高性能を得るのが難しいアプリケーションと予想された．

（B） ナノ分野（PHASE, RSDFT）
（a） 高並列化の観点では，現状の並列化手法を用いると，数万のノードを備える「京」での並列化には適応し難く，抜本的な並列化手法の見直しが鍵となる．
（b） 単体性能から見ると，主要処理の行列・行列積化により高性能は実現可能と予想された．

（C） 工学分野（FrontFlow/Blue）
（a） 高並列化の観点では，比較的並列性を識別しやすい領域分割の手法を用いるが，地球科学分野のアプリケーションよりも並列化手法は多少複雑になる．隣接通信は高並列時でも大きな負荷とならないが，大域通信負荷は高並列時に増大する傾向がある．
（b） 単体性能から見ると，高い B/F 値が要求され，更に，一部のデータは，キャッシュラインの有効利用を妨げるため，「京」では高い性能を得ることは難しいアプリケーションと予想された．

（D） 物理分野（LatticeQCD）
（a） 高並列化の観点では，（C）同様に領域分割手法を用いるが，並列特性分析の結果，通信回数（頻度）が多くなることが明らかになった．このため，通信トポロジーの活用を意識した高度な並列化が必要である．
（b） 単体性能の面から見ると，比較的高い B/F 値を要求するアプリケーションであるが，分割の大きさを調節することにより，キャッシュの有効利用が見込める．ただし，効率の良い命令スケジューリングが難しいアプリケーションである．

8. アプリケーションの性能例

7. に示した 6 本のアプリケーションのうち 2 本について，高性能化の結果を述べる．

8.1 RSDFT

ナノスケールでの量子論的諸現象をシミュレーションするアプリケーションである．新機能を有するナノ物質の構造を予測することで，次世代半導体テクノロジーの加速が期待されている．RSDFT は，並列化手法を見直し，空間の領域分割から空間＋エネルギーバンド並列（2 軸並列）へとプログラムの書換えを実施した．並列軸を増やすことで 10 万並列レベルに対応可能となった．空間並列のみの場合は，全プロセッサ間の大域通信が必要であったが，このプログラムの書換えにより，通信時間の増大を招く空間に対する大域通信を一部のプロセッサ間での通信とすることで，通信時間の短縮が図れた．CPU 単体性能についても行列・行列積の形に書き換えてあり，キャッシュの有効利用，効率の良いスケジューリング，SIMD 演算器の有効利用ができており高い性能が得られている．総合すると，「京」の 96 きょう体，9,216 プロセッサ，73,728 コアを使用し，ピーク性能 1.18 PFLOPS に対し 31.4%（370 TFLOPS）という高い実行性能を実現した．

8.2 Seism3D

Seism3D は，有限差分法により数値的に粘弾性方程式を時間発展させることにより，地震伝搬と津波を連動して解く大規模な並列化に対応しているアプリケーションである．Seism3D は，単体性能の向上が課題であったが，プリフェッチの有効利用，ラインアクセスの有効利用，キャッシュの有効利用のための書換えを実施することで，16 プロセッサ，128 コアを使用して，ピーク性能比：10.3% から 15.3% まで向上し約 5 割の性能向上が得られた．現在，高並列での性能測定を実施中であるが，高並列においても同程度のピーク性能比を予想している．

9. ま と め

現代のスーパコンピュータでは，高並列化と CPU 単体性能向上の両面からアプリケーションプログラムを高性能化することが重要であり，科学技術計算用に機能拡

張された64万以上の計算コアから構成される「京」については，なおさらである．本稿では，まず，高並列化の面では通信時間の短縮と非並列計算部の縮小が重要であること，またCPU単体性能向上の面では，五つの重要点を挙げ，アプリケーションが要求するB/F値と五つの重要点の関係について示した．次に理研で実施しているアプリケーションの高性能化の概要を紹介し，RSDFTとSeism3Dについては，高性能化の現状について報告した．

謝辞　本性能最適化に際し御討論頂き貴重な助言を頂いた，東京大学地震研究所の古村孝志教授，並びに理化学研究所次世代スーパーコンピュータ開発実施本部の諸氏に感謝します．本稿の結果は，理化学研究所計算科学研究機構が保有する京速コンピュータ「京」の試験利用によるものです．

文　　献

（1） M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno, and S. Iga, "Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloudresolving simulations," J. Comput. Phys., vol. 227, no. 7, pp. 3486-3514, 2008.
（2） 古村孝志，"差分法による3次元不均質場での地震波伝播の大規模計算，" 地震2, vol. 61, pp. S83-S92, 2009.
（3） http://www.ciss.iis.u-tokyo.ac.jp/riss/project/device/
（4） J. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi, S. Okada, and K. Yabana, "A massively-parallel electronic-structure calculations based on real-space density functional theory," J. Comput. Phys., vol. 229, no. 6, pp. 2339-2363, 2010.
（5） 乱流音場解析ソフトウェア　FrontFlow/Blue. http://www.ciss.iis.u-tokyo.ac.jp/rss21/theme/multi/fluid/fluid_softwareinfo.html
（6） S. Aoki, K. -I. Ishikawa, N. Ishizuka, T. Izubuchi, D. Kadoh, K. Kanaya, Y. Kuramashi, Y. Namekawa, M. Okawa, Y. Taniguchi, A. Ukawa, N. Ukita, and T. Yoshie, "2 + 1 flavor lattice QCD toward the physical point," phys. Rev. D, vol. 79, 034503, 2009.

（平成23年10月15日受付　平成23年11月2日最終受付）

南（みなみ）　一生（かずお）

昭56日大・理工・物理卒．同年，富士通株式会社入社．平12(財)高度情報科学技術研究機構入社．平20-03から独立行政法人理化学研究所次世代スーパーコンピュータ開発実施本部開発グループアプリケーション開発チームチームリーダー．