様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ５ 月 ２６ 日現在

機関番号：１７５０１
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５００３４４

研究課題名（和文）共有・分散メモリ型並列処理の融合による統計計算アルゴリズムの効率化

研究課題名（英文）Improving Statistical Calculation via Hybrid Parallel Processing with Shared and Distributed Memory Based Parallelization

研究代表者

越智　義道（OCHI, Yoshimichi）

大分大学・工学部・教授

研究者番号：６０１８５６１８

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：マルチコアCPUを持つ複数の計算機群からなる計算環境下で，統計計算上の並列処理のアルゴリズムにおける処理の効率化の検討を行った．並列化にあたっては共有メモリ型スレッド並列処理と分散メモリ型プロセス並列処理の特性に配慮し，それらを融合して総合的にアルゴリズムの処理効率を向上することを考えた．まず，モンテカルロ積分にこの融合した並列処理を適用し，そのタスク分配の方法について検討した．この検討をもとに，現実的な統計解析上のアルゴリズムの並列化について，離散反応データにおける確率変動モデル，ロジスティック回帰における正確分析に関する検討を行った．

研究成果の概要（英文）：It is investigated to improve statistical calculation with parallel processing algorithms under an environment with a group of computers having multicore CPUs. In order to implementing parallel algorithms, characteristics of shared memory parallelism and distributed memory parallelism are taken into account and hybrid type processing with both approaches is devised to achieve overall efficiency of algorithms. First, general methods of task management are examined to parallelize Monte Carlo quadrature as a hybrid parallel processing and it is followed by investigating ways to parallelize specific algorithms of actual statistical analyses, such as probability variation model for discrete response data and exact inference of logistic regression analysis.

研究分野：計算機統計学

キーワード：統計科学　計算機統計　並列計算　共有メモリ型並列計算　分散メモリ型並列計算

１．研究開始当初の背景

計算機を支える技術の急速な進歩によりPCの処理能力は大幅に向上している．研究者が日常的に使用する PC 単体も，すでにハードウエアでは 64 ビット対応の CPU が基本となり，さらにコアを複数持つものが広く普及している．コアの複数化はCPUクロックの高周期化の限界に対応するものであったが，汎用的な計算機でもこれらの仕様のものがその基本となってきている．ハードウエアのこれらの進展に伴うように OS やアプリケーションの 64 ビット化やマルチタスク化といったソフトウエアでの対応も進んできている．

計算機の技術的な進展は，統計科学における計算の側面を大きく変貌させ，従来ではその実行時間あるいは計算機のメモリの制約のため使用をためらわれた方法であっても，実用的な分析方法として考えられるようになってきている．統計的な手法開発の観点からは，そのような方法が現実的な手法として実装可能であれば，その詳細な特性評価を試みる必要が生じ，さらなる計算処理能力の向上への期待を生み出すこととなる．

ところが，これらの計算機のハードウエアにおける進展に対して，ソフトウエアにおける対応やアプリケーションの適応の状況は，必ずしもこれに十分に追従できていない現実がある．また，アプリケーション開発環境においても並列計算を念頭においた環境が用意されつつあるが，それらを活用したアプリケーションが容易に構築できるまでに至っていない状況にある．

一方で，近年マルチコア化の動きはスピードを増し，汎用CPUでも6コア，12スレッドが可能なものが現れ，2011 年 11 月からは 8 コアを持つCPUも市販されている．この様に，計算機単体内で複数コアを持ち並列処理が実行可能な場合でも，プロセス並列の考え方を適用することはできる．

ところが，計算機をまたぐプロセス間通信と同一計算機内でのプロセス間通信では速度が大きく異なり，ある意味で，計算処理のためのプロセスの群化の考え方を導入する必要があり，マルチコアCPUを持つ単一計算機内ではスレッド並列処理が望ましいと言える．そこでは，メモリを処理単位ごとに分割して管理する必要はなく，そのプロセスで確保したメモリを処理単位（ここではスレッド）が共有して活用することが可能となり，処理で使用可能なメモリ空間の範囲を広く保持することもできる．また，同一計算機上のメモリを共有することの利点を生かすことにより，通信によらずにタスク処理単位（ここではスレッド）間で処理すべきデータの受け渡しが可能である．近年ではこのようなスレッド並列を可能にするために環境 Windows スレッド API，Pthreads，Solaris スレッド，Java スレッド・クラスや共有メモリ型並列のための規格 OpenMP(V4.0：http://www.openmp.org/)も整備されてきている．さらに，共有・分散メモリ型並列処理の融合処理に関する研究も計算機科学の分野では進んできており(Rabenseifner ら，2008)，統計計算でもこれらの技術の活用が望まれる．

一方で，この共有メモリ型による並列計算では，スレッド内でのみ有効となるメモリと共有されるメモリとの間での区別を厳格にし，十分な配慮を行わなければ，デッドロックと呼ばれる処理停止やデータ・レースと呼ばれるデータの不整合の問題を生じることになる．また，並列化の方法によってはそのスループット（単位時間当たりの処理能力）の低下を招くことになる．実際，当該研究代表者でも統計計算で多く用いられるモンテカルロ積分に関わるスレッド並列での処理能力の低下を経験している．

２．研究の目的

本研究では，マルチコア CPU を持つ複数の計算機群からなる計算環境下で，統計計算上のアルゴリズムの並列処理による効率化の検討を行う．上で述べたモンテカルロ積分の例では，疑似乱数生成の際に利用される状態変数の共有メモリ上での配置に問題があり，スレッド処理を形成する際の処理同期にもとづく遅延が原因であり，この問題を回避することにより処理効率の向上を図れることが確認された．このように統計計算上のアルゴリズムについて，並列処理，特に共有メモリ型のスレッド並列計算を活用する際の，アルゴリズムの並列化について検討を加え，さらに複数のマルチコアCPUを持つ計算機を用いた環境下でのプロセス間通信との融合に関わるアルゴリズム上の問題点について検討を行うことを目的とする．

３．研究の方法

マルチコアCPUを持つ複数の計算機群からなる計算環境下で，統計計算上の並列処理のアルゴリズムにおける処理の効率化の検討を行う．このために，

・共有メモリ型スレッド並列処理として OpenMP を用いる
・分散メモリ型プロセス並列処理として MPI 準拠の環境 OpenMPI を用いる

こととし，共有・分散メモリ型並列計算の融合化法について検討を加える．

具体的な統計計算として，まず．モンテカルロ積分に関わるアルゴリズムについて，そのタスク分配の方法について基礎的な検討を加える．その後，現実的な統計解析上のアルゴリズムの並列化について，離散データにおける確率変動モデル，ロジスティック回帰における正確分析に関する検討を行った．前者は各処理の独立性が高い状況，後者は処理間の依存性の高い状況を検討することを意味する．

４．研究成果

（１）共有・分散メモリ型並列計算の融合におけるタスク分散の検討

モンテカルロ法による円周率πの求積を題材として，共有・分散メモリ型の計算のタスク分配について検討を加えた．ここでの計算アルゴリズムは，正方形内に一様乱数により点(x,y)を配置して，正方形内の扇形内の点$(x^2+y^2\leqq 1)$の計数値をもとに，円周率πを求める方法を採用した．乱数生成については，共有メモリ環境での生成に配慮して，GNU Scientific Library (GSL) ライブラリを用いることとして，乱数生成器としてはメルセンヌ・ツイスター法(gsl_rng_mt19937)を利用し，プロセス・スレッドごとに固有のシードを与えて計算を行った．この GSL ライブラリにおける乱数生成器は，Gcc ライブラリでの rand 関数等の乱数生成器の様に，メモリ共有メモリ型並列環境下で用いる際に副作用を起こすことが無く，スレッド・セーフな環境として利用することができる．

ちなみに，100 億個の点を生成して，1 台の計算機（PhenomII X6 1090T）上で，OpenMPI，OpenMP を用いてそれぞれ分散メモリ型並列ならびに共有メモリ型並列の処理時間を比較した結果は

プロセス・スレッド数	MPI による処理時間（秒）	OpenMP による処理時間（秒）
2	135.44	135.50
3	91.497	91.438
4	76.216	74.959
5	61.020	60.329
6	50.927	51.240

であった．ちなみに１プロセスでの実行時間は 266.60 秒であったのでそれぞれタスク分配の効果を確認できている．

次に，６コアの CPU（PhenomII X6 1090T）３台と８コアの CPU（FX8350E）２台の計５台を用いて 500 億個の乱数を用いて，各計算機間の並列処理に MPI を用い，並列計算機内での処理に OpenMP を用いて，共有・分散メモリ並列融合よる計算について検討した．この際，各スレッドに均等にタスクを分配する方法を取ると，

スレッド数	総スレッド数	処理時間（秒）
2	10	138.67
3	15	94.544
4	20	106.76
5	25	120.68
6	30	110.55

となり，逐次プログラムで要した処理時間 1328.51 秒に比べれば，はるかに短い時間で処理を終えることができたものの，スレッドを増すことによる明確な効率向上の効果は得られなかった．これは，処理能力の低い CPU でのスレッドでの処理遅延による影響に限定されるものではなく，コア数に対して使用するスレッドが増加し，CPU の処理能力にゆとりがなくなった場合に，処理時間のばらつきが大きくなることに起因することが確認された．このため，スレッド段階でのタスク分配について，OpenMP での dynamic スケジュール，つまり，タスクを細かなチャンクに分割し，処理済みのスレッドが順に次のチャンクの処理を行う方法，を採用して計算を行うことを検討した．その結果が次の通りである．

スレッド数	総スレッド数	処理時間（秒）
2	10	135.24
3	15	92.686
4	20	102.48
5	25	78.099
6	30	66.202

先の結果とは異なり，基本的には使うスレッドの増加によって処理時間が短縮される効果が確認できるようになったが，計算機ごとのスレッド数が 4 の場合に，期待される効果を得ることができていない．この原因としては，計算アルゴリズムの性質の観点から MPI レベルでの処理においてサーバーとなる計算機の処理能力の問題が示唆された．ちなみに，この計算では，PhenomII X6 1090T をサーバー計算機として利用していた．一方で，処理能力に余裕のある FX8350E をサーバーとして実行した結果は以下のとおりである．

スレッド数	総スレッド数	処理時間（秒）
2	10	138.90
3	15	92.280
4	20	83.341
5	25	75.268
6	30	63.172

この結果，先の結果で見られたようなサーバー用計算機での大きな処理遅延を回避することが可能になり，相応の処理効率の向上を確認することができるようになった．

このタスクスケジューリングの検討に用いている問題においては，その全ての並列処理を MPI による分散メモリ型処理として計算を行うことも可能である．そこで，５台の計算機でそれぞれ複数のプロセスを起動して実行した結果が次のとおりである．

プロセス数	総プロセス数	処理時間（秒）
2	10	135.87
3	15	91.838
4	20	76.401
5	25	63.793
6	30	53.944

今回のこの問題においては MPI を用いて分散メモリ型のプロセス処理にもとづいて計算を行った方が最大で 10 秒程度高速に処理

を終了することが可能であった．これは，MPI 環境下での処理時間のばらつきが，OpenMP でのスレッドでの処理時間のばらつきに比して小さいことに起因していることが計算機ごとの処理時間の分析によって確認された．

ただし，上記の計算は，計算処理に比してプロセス間通信の負荷は軽いものであり，並列計算の際にその比率が異なる場合には注意が必要である．つまり，生成する点の個数を少なくした場合には，この結果が逆転することとなる．例えば，生成する一様乱数を 100 万個にした場合，

プロセス・スレッド数	MPI による処理時間（秒）	OpenMP による処理時間（秒）
2	0.0227	0.0212
3	0.0212	0.0168
4	0.0215	0.0152
5	0.0220	0.0132
6	0.0227	0.0115

となり，MPI の方はそのプロセス数を増すとかえって所要時間が増す一方で，OpenMP ではスレッドを増やすと徐々に時間を短縮する効果を得ていることに注意が必要である．

共有・分散メモリ型の並列計算の融合においてはこのように処理の特性に留意してその並列化のアルゴリズムを検討することが重要である．

（２）離散データの確率変動モデルにおけるアルゴリズムの並列化

２値反応データに関わる確率変動モデルの分析について考察した． 観測 $(y_i, \mathbf{x}_i)$ i=1,2, …, N における反応 y_i について，その反応確率 p_i が固定された時の確率分布を２項分布 Binom(n_i, p_i) とし，確率 p_i のロジスティック変換が次のように，共変量 $\mathbf{x}_i$ を用いた回帰式

$$\log\frac{p_i}{1-p_i}=\eta_i=\alpha+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+e_i,\quad e_i\sim N(0,\sigma^2)$$

を通じて，正規分布に従うと仮定する．この時，尤度は

$$L(\boldsymbol{\beta},\mu,\sigma^2)$$
$$=\prod_{i=1}^{N}\int\binom{n_i}{y_i}p(\eta_i)^{y_i}(1-p(\eta_i))^{n_i-y_i}\,d\Phi(\eta_i;\mu,\sigma^2)$$

となる．この積分計算については，より柔軟な分布族への拡張に対応することを前提として，モンンテカルロ積分を用い，４節（１）で検討した共有・分散メモリ型並列処理の融合による並列計算を採用することとした．計算機間では MPI によるプロセス間通信を基礎とする分散メモリ型並列，計算機内では OpenMP を用いた共有メモリ型並列（スレッド並列）を用い並列計算を行うこととした．計算に利用した計算機（CPU）は Xeon x5570（４コア）１台，PhenomII X6 1090T（６コア）３台の４台を用い，各計算機での CPU コア数分のスレッドを起動し，総計 22 スレッドを利用した．

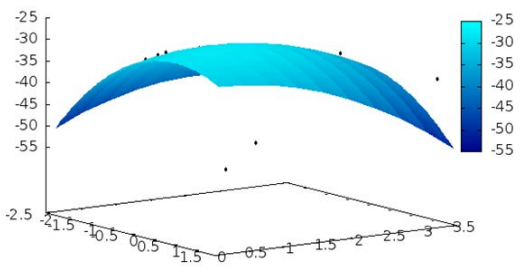
ここでは，Wilson(1989)の住宅満足度調査のデータを用い，その調査目標であったメトロポリタンエリア居住か否かによる満足度の違いについては尤度比検定を行うことを目標とした．データは各調査単位で５件の近隣世帯を調査し，それぞれの地域ごとに 18 組，17 組，総計 35 組のデータからなる．

尤度比検定では，まず帰無仮説と対立仮説の２つの設定の下で，最尤法を行う必要があり，その尤度の最大化について繰り返し計算が必要である．その際に，各計算でこの 35 組の観測に関わる尤度の計算が必要となる．各組におけるモンテカルロ計算で十分な精度を保って結果を得るためには少なくとも１億個の乱数生成が必要となり，逐次計算を基礎とする計算法は，現実的な分析法ではない．

そこで，ここで先の共有・分散メモリ融合型の並列処理を用い，22 スレッドでの並列計算を行い，尤度関数の最適化法にネルダー・ミード（NM）法を用いて計算したところ，4210.4 秒で計算を終えることができた．このとき，尤度関数の評価回数は 128 回であった，22 スレッドを使っても１回の尤度関数の評価に 30 秒以上を要していることが分かる．

ネルダー・ミード法を始め，多くの既存の最適化手法は目的関数の値が正確に計算されることを前提に考案されている．ところが，この例のように，モンテカルロ積分の場合，十分な精度を保つためには多くの乱数生成を必要とし，分析のための基礎計算として用いるには，並列計算を用いるとしても，計算コストは高いものになる．このため，この応用例では，初期段階では計算精度を抑えることにより処理速度を高め，段階的に計算精度を高める方法について検討した．

モデルに用いられるパラメータ空間において最適解が含まれると想定される領域上で仮想最適解を中心として多変量正規分布にもとづいて評価点を選定し，その評価点においてモンテカルロ積分を用いた目的関数値を計算し，得られた評価値に２次曲面方程式を適合する．



得られた２次曲面の方程式から，次ステップの最適解候補を選定し，評価点選定のための正規分布の分散を段階ごとに縮小し，探索領域の絞り込みを行う方法を検討した．この探索の段階に従ってモンテカルロ積分の評価を 10 万個の乱数生成から徐々に 1 億個まで増加させることとした．その結果，処理時間を 2601.8 秒まで減少させることが可能になった．

	ステップ	尤度関数の評価回数　乱数生成数別				
		10万	100万	1000万	1億	時間(秒)
NM	20				128	4210.4
曲面	5	120	120	120	60	2601.8

ちなみに，この両者で計算された p 値はそれぞれ 0.3170(NM)，0.3168(曲面)であり，小数点以下３桁の精度を保つ結果が得られている．

（３）ロジスティック回帰における正確分析に関する並列化

（１），（２）の検討で扱ったモンテカルロ積分の様な処理はその処理単位間の独立性が高く，比較的に並列化が容易である．本研究では，次に，処理間の依存関係の高い処理に関わる並列化の問題として，ロジスティック回帰分析の条件付き正確推測に関わる計算アルゴリズムの並列化について検討を加えた．

対象個体 i (i=1,2,…,N)の 2 値反応を Y_i，その反応確率を p_i，共変量 $\mathbf{x}_i$ とするとき，ロジスティック回帰モデルは

$$\log\frac{p_i}{1-p_i}=\alpha+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\qquad(i=1,2,\ldots,N)$$

と書ける．ここで，$\boldsymbol{\beta}$ は説明変数に対応する回帰パラメータである．このとき，このモデルの尤度は

$$L(\boldsymbol{\beta})=\frac{\exp(\mathbf{t}^{*\mathrm{T}}\boldsymbol{\beta})}{\prod(1+\exp(\mathbf{x}_i\boldsymbol{\beta})}$$

と書ける．$\mathbf{t}^*$は $\boldsymbol{\beta}$ の十分統計量

$$\mathbf{T}=\sum Y_i\mathbf{x}_i$$

の実現値である．

ここで，パラメータ $\boldsymbol{\beta}$ について $\boldsymbol{\beta}^{\mathrm{T}}=(\boldsymbol{\beta}_1^{\mathrm{T}},\boldsymbol{\beta}_2^{\mathrm{T}})$ とし，$\boldsymbol{\beta}_1$ を局外パラメータ，$\boldsymbol{\beta}_2$ を目的パラメータとする．ここで，$\boldsymbol{\beta}_2$ に関する仮説検定 H_0：$\boldsymbol{\beta}_2=\boldsymbol{\beta}_2{}^*$を考える．　このとき，$\boldsymbol{\beta}_1$ に対応する十分統計量 $\mathbf{T}_1$ について条件付けを行って条件付分布を考えると，

$$\Pr(\mathbf{T}_2=\mathbf{t}_2\mid\mathbf{T}_1=\mathbf{t}_1^*)=\frac{c(\mathbf{t}_1^*,\mathbf{t}_2)\exp(\mathbf{t}_2^{\mathrm{T}}\boldsymbol{\beta}_2)}{\sum_{\mathbf{u}_2}c(\mathbf{t}_1^*,\mathbf{u}_2)\exp(\mathbf{u}_2^{\mathrm{T}}\boldsymbol{\beta}_2)}$$

となる．ここで，$c(\mathbf{t})$は十分統計量の実現値が $\mathbf{t}$ であるような反応 y_i からなる N ビット２値系列の個数である．正確推測ではこの条件付き分布にもとづいて p 値の計算を行うことになる．

このとき，正確推測では，条件 $\mathbf{T}_1=\mathbf{t}_1{}^*$のもとで，$\mathbf{t}_2$ の可能な全てのパターンと，その時の２値系列の個数 $c(\mathbf{t}_1{}^*,\mathbf{t}_2)$の数え上げの処理を行うことが必要である．この計算については MSA (Hirji ら，1987)と呼ばれるアルゴリズムが存在する．このアルゴリズムはデータの個数だけの段階を踏んで計算することになるが，第 n 番目のデータまでを用いて計算される十分統計量を $\mathbf{T}_{(n)}$とすると，n+1 番目の十分統計量 $\mathbf{T}_{(n+1)}$が

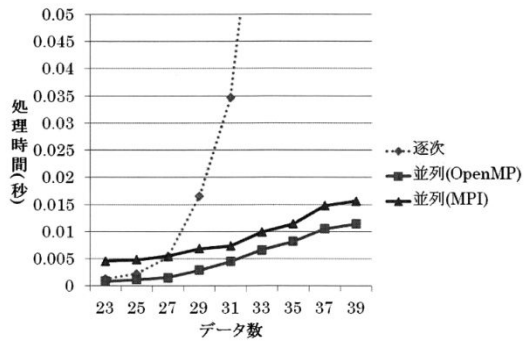
$$\mathbf{T}_{(n+1)}=\mathbf{T}_{(n)}+y_{n+1}\mathbf{x}_{n+1}$$

となることを利用した再帰計算を基礎としている．このアルゴリズムのままでは並列化は難しいが，これを２段階に拡張して，

$$\mathbf{T}_{(n+2)}=\mathbf{T}_{(n)}+y_{n+1}\mathbf{x}_{n+1}+y_{n+2}\mathbf{x}_{n+2}$$

とすることで，加算パターンとして，組(y_{n+1}，y_{n+2})についての可能な４つの結果に対応する加算とすることが可能であり，処理を並列化できる．

さらに，計算アルゴリズム上これらから計算される可能な十分統計量の４つのリストを１つのリストに整列する必要があるが，この部分は，パイプライン処理にとって並列実装が可能である．前者の加算処理ついては MPI による実装が可能であるが，後者のパイプライン処理について分散メモリ型の並列処理である MPI 実装では細かなプロセス間通信を数多く必要とするため，必ずしも逐次処理に優る効果が得られない場合があることを確認した．従ってこの部分については共有メモリ型並列処理を考え OpenMP を用いて実装することとした．



２値離散共変量を３つ，連続量共変量を１つ，目的パラメータとして離散共変量の１つ対応するパラメータを選んで，実装したプログラムで実行した結果は上図のとおりである．使用した CPU は PhenomII X4 965(４コア)である．逐次プログラムの場合，データ数が増えるに従い急速に実行時間が増大しており，上図には示していないが，データ数 39 の場合，約 0.25 秒かかっていた．マージ実装に MPI を利用した場合，データ数 27 以下では逐次計算に劣る結果となっている．一方で，OpenMP ではその様子は見られず，一貫して逐次処理より高速に処理できることが確認された．このように，アルゴリズムの特性に合わせて分散メモリ並列と共有メモリ並

列を融合することによって効率的な並列処理が行えることが確認できた.

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）

〔学会発表〕（計　７　件）

①大場紀彦・越智義道，分散・共有メモリ型ハイブリッド並列処理の統計計算への応用，大分統計談話会第 46 回大会，2012 年 10 月 11 日，富士通大分システムラボラトリ（大分県・大分市）

②田中大介・越智義道，離散データにおける正確推測のための並列化計算アルゴリズム，大分統計談話会第 47 回大会，2013 年 2 月 14 日，富士通大分システムラボラトリ（大分県・大分市）

③越智義道，統計計算アルゴリズムと並列処理方式について，医学統計研究会　秋季セミナー鹿児島 2013，2013 年 9 月 21 日，宝山ホール（鹿児島県・鹿児島市）

④田中大介・越智義道，ロジスティック回帰モデルにおける条件付き正確推測のための並列処理，日本計算機統計学会・第 27 回シンポジウム，2013 年 11 月 16 日，市民会館崇城大学ホール（熊本県，熊本市）

⑤大場紀彦・越智義道，モンテカルロ積分評価を含む場合の最尤法の適用について，日本計算機統計学会・第 27 回シンポジウム 2013 年 11 月 16 日，市民会館崇城大学ホール（熊本県，熊本市）

⑥高野　滉・越智義道，離散データ解析における条件無し正確推測での完全分離型データの処理について，大分統計談話会・第 51 回大会，2015 年 2 月 12 日，富士通大分システムラボラトリ（大分県・大分市）

⑦越智義道，統計計算における並列計算環境の活用と課題，医学統計研究会冬季セミナー鹿児島 2015，2015 年 1 月 24 日，鹿児島県民交流センター（鹿児島県・鹿児島市）

〔その他〕
ホームページ等
http://bunsyo1.ad.oita-u.ac.jp:8080/kobetu.asp?id=227

６．研究組織
(1)研究代表者
越智　義道（OCHI Yoshimichi）
大分大学・工学部・教授
研究者番号：60185618