様式 C-19

科学研究費補助金研究成果報告書

平成 22 年 6 月 18 日現在

研究種目： 若手研究（B）
研究期間： 2008～2009
課題番号： 20700095
研究課題名（和文） マルチコア計算機クラスタにおける頻出系列パターン抽出処理の並列化に関する研究
研究課題名（英文） Research on Parallel Extraction of Frequent Sequence Pattern on a Multi-Core Computer Cluster.
研究代表者
田村 慶一（Tamura Keiichi）
広島市立大学・情報科学研究科・講師
研究者番号：80347616

研究成果の概要（和文）：

本研究では，CPU の主流となりつつあるマルチコア CPU を搭載した計算機で構成される計算機クラスタにおける頻出系列パターン抽出処理の並列化のための並列化モデルの開発を行った．開発したマルチコア計算機クラスタのための並列化モデルは，マルチコア計算機クラスタの計算機資源を効率的に使用するために，(1) 共有資源の効率的な利用方式，(2) コア間と計算機間の効率的な連携手法，(3) マルチコア計算機クラスタのための動的負荷分散手法の 3 つの手法を兼ね備えている．開発した並列化モデルを用いて典型的な頻出系列パターン抽出処理の並列化を行った．実験結果により，開発した並列化モデルの有効性を示すことができた．

研究成果の概要（英文）：

This study has developed a novel parallelization model for extraction of frequent sequence pattern on a computer cluster which is constructed by computers with multi-core processors, which is becoming mainstream of CPU. In order to use computer resources efficiently on a multi-core computing cluster, the developed parallelization model has three characteristics, (1) efficient use method of shared resource, (2) efficient coordinated technique between cores and between computers, (3) dynamic load-balancing technique for a multi-core computer cluster. A typical frequent pattern extraction algorithm was made parallel by using this model. The experimental results shows good performance.

交付決定額

（金額単位：円）

	直接経費	間接経費	合 計
2008 年度	1,700,000	510,000	2,210,000
2009 年度	1,000,000	300,000	1,300,000
年度			
年度			
年度			
総 計	2,700,000	810,000	3,510,000

研究分野：並列処理，データマイニング
科研費の分科・細目：情報学・メディア情報学・データベース

キーワード：並列処理，マルチコア，データマイニング

１．研究開始当初の背景

近年，データベースに蓄積されたデータから有用な情報を抽出するためのデータマイニング技術が注目されている．記憶装置の大容量化と低価格化により，データベースに蓄積されたデータは大規模化しており，高性能データマイニング技術の開発が重要な課題となっている．データマイニングの中で最も重要な問題のひとつに頻出系列パターンを抽出する問題がある．頻出系列パターンとは，系列データ中に頻出する順序関係を持ったアイテムの組合せパターンであり，抽出した頻出系列パターンは様々な解析のための基礎データとなっている．頻出系列パターン抽出処理は，CPU コストと I/O コストとが非常に高いが，多くの場合，高い並列性を持つ．そこで，様々な並列計算機において，その並列化に関する研究が盛んに行われてきた．

本研究では，CPU の主流となりつつあるマルチコア CPU を搭載した計算機（PC もしくはワークステーション）で構成される計算機クラスタにおける頻出系列パターン抽出処理の並列化に関する研究を行う．複数のプロセッサコア（以下，コアと呼ぶ）を１個のパッケージに集積した CPU をマルチコア CPU という．CPU のマルチコア化にともない，計算機クラスタは，マルチコア CPU を搭載した計算機によって構成されるようになっている．マルチコア CPU を搭載した計算機で構成される計算機クラスタのことを，ここでは，マルチコア計算機クラスタと呼ぶこととする．

頻出系列パターン抽出処理の並列化に関する研究は，共有メモリ型/分散メモリ型並列計算機，PC/ワークステーションクラスタやグリッドなど様々なプラットフォームで盛んに行われている．これまでに提案されている様々な並列化手法は，その負荷分散手法もあわせて非常にすぐれたものが存在するが，マルチコア計算機クラスタの計算機資源を有効に活かすには，その特質に着目した新しい並列化技術が必要である．本研究では，共有メモリ型/分散メモリ型並列計算機，PC/ワークステーションクラスタやグリッドなど様々なプラットフォームで行われてきた頻出系列パターン抽出処理の並列化に関する研究を継承しつつ，マルチコア計算機クラスタの計算機資源を有効に活かす並列化モデルの開発を目指す．

２．研究の目的

シングルコアの CPU を搭載した計算機によって構成される計算機クラスタにおける頻出系列パターン抽出処理の並列化に関する研究や，単一のマルチコア CPU 上での頻出系列パターン抽出処理の効率化に関する研究は様々行われているが，マルチコア計算機クラスタにおける頻出系列パターン抽出処理の並列化に関する研究は十分に行われていない．マルチコア計算機クラスタの計算機資源を有効に使用するためには，マルチコア計算機クラスタの特徴を考慮した，頻出系列パターン抽出処理の並列化技術の開発が不可欠である．

そこで，本研究では，マルチコア計算機クラスタの計算機資源を効率的に使用するために，

（1）共有資源の効率的な利用方式，

（2）コア間と計算機間の効率的な連携手法，

（3）マルチコア計算機クラスタのための動的負荷分散手法，

を持った並列化モデルの開発を目指す．開発をする並列化モデルは，マルチコア計算機クラスタの計算機資源の特徴を考慮したものであり，この並列化モデルを用いることで，マルチコア計算機クラスタの十分な性能を引き出すことができると期待できる．

３．研究の方法

本研究では，マルチコアの中でもホモジニアスマルチコア CPU に研究の的を絞り研究を進める．ホモジニアスマルチコア CPU を搭載する計算機で構成されるマルチコア計算機クラスタにおける頻出系列パターン抽出処理の並列化では，次に示す大きな３つの課題があり，従来の計算機クラスタを対象に提案されてきた並列化手法をそのまま使用することができない．

(a) キャッシュ，メモリや I/O など共有資源へのアクセス競合：研究対象としているホモジニアスマルチコア CPU は，中央演算装置や１次キャッシュは個々に独立しているが，その他のレベルのキャッシュ，メモリバスや I/O バスを共有しているものが多い．つまり，複数のコアで共有資源を持つことになる．このとき，共有資源へのアクセスが競合すると，処理の効率が落ちてしまう．

(b) コア間と計算機間の通信コストの

差：マルチコア計算機では，コア間と計算機間という 2 つの階層的な通信レベルが存在する．コア間はセマフォ，共有メモリやマルチスレッドにより計算機間と比較して高速な通信が可能であるが，計算機間ではコア間と比較して通信コストが高くなる．この通信コストの差を考慮しないと十分な性能が得られないことが考えられる．

(c) 不均一な計算機構成：CPU の多コア化は，年々進んでいくと考えられる．そこで，計算機クラスタを構成する計算機はこれまで以上に様々な CPU 構成の計算機が同時につながることになる．このとき，各計算機の構成を考慮しないと，負荷の偏りが生じ，全体の性能が低下する可能性がある．

本研究では，(a) に関しては，「共有資源の効率的な利用方式」により解決を行う．(b) に関しては，「コア間と計算機間の効率的な連携手法」により解決を行う．(c) に関しては，「マルチコア計算機クラスタのための動的負荷分散手法」により解決を行う．また，研究期間内に基本的なアルゴリズムを設計するとともに，プロトタイプを実装し，性能評価まで行う．

４．研究成果

本研究では，３．で示したように，(1) 共有資源の効率的な利用方式，(2) コア間と計算機間の効率的な連携手法，(3) マルチコア計算機クラスタのための動的負荷分散手法の 3 つ手法を持つマルチコア計算機クラスタのための並列化モデルの開発を行った．また，開発を行ったマルチコア計算機クラスタのための並列化モデルを使って，典型的な頻出系列パターン抽出処理の 1 つである段階的一般化法の並列化を行い，その有効性を確認することができた．

以下，開発を行った並列化モデルの特徴と，性能評価の結果を示す．

（1）開発を行った並列化モデル

開発したマルチコア計算機クラスタのための並列化モデルの特徴は以下のとおりである．

(1) 各計算機で起動するワーカはひとつで，コア数に応じたスレッド（ワーカスレッドと呼ぶ）をワーカ内に複数の起動し，ワーカスレッドにおいてタスクを実行する．各コア上でワーカを起動する場合と比較して，ワーカ内でコア数に応じたワーカスレッドを起動した方が，共有資源を効率的に利用することができ，計算機内で負荷の偏りを効率的に解消できる．

(2) ワーカは 2 種類のタスクプールを持つ．ワーカは 1 つのグローバルタスクプールを持ち，各ワーカスレッドが 1 つのローカルタスクプールを持つ．各ワーカスレッドがローカルタスクプールを持つことで，ワーカスレッド間の競合が少なくなる．ワーカスレッド間の競合を少なくすることで共有資源を有効に活用することができる．

(3) ワーカスレッド間とワーカ間との 2 つの階層に分けて，キャッシュベースのランダムタスク・ステイル法を用いて負荷の偏りを解消する．キャッシュベースのランダムタスク・ステイル法を階層的に用いることで，計算機内と計算機間の負荷の偏りを効率的に解消できる．また，キャッシュベースのランダムタスク・ステイル法を用いることで，不均一な計算機構成の場合も，効率的に負荷の偏りを解消することができる．

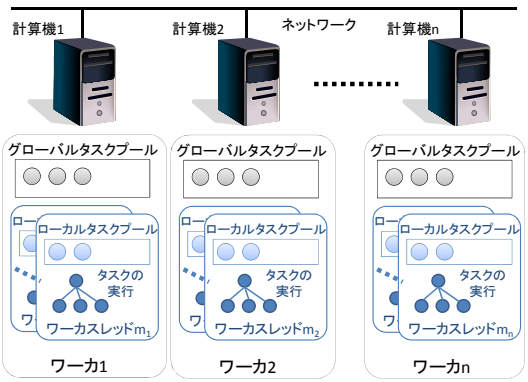

図 1 ：並列化モデル

提案する並列化モデルを図 1 に示す．各計算機にはワーカを 1 つ起動する．ワーカは複数のワーカスレッドを持つ．ワーカは他のすべてのワーカと通信を行うことができ，ワーカスレッドは他のすべてのワーカスレッドとやり取りができる．

ワーカは，2 種類のタスクプールを持つ．1 つ目がグローバルタスクプールで，2 つ目がローカルタスクプールである．グローバルタスクプールはワーカで 1 つ設置する．ローカルタスクプールはワーカスレッドごとに 1 つ設置する．

ワーカスレッドは，ローカルタスクプールからタスクを取り出し，タスクを実行する．ワーカスレッドは，ローカルタスクプールに

タスクがなくなった場合のみ，グローバルタスクプールからタスクを取り出す．ワーカスレッドごとにローカルタスクプールを持つことで，タスク処理に関わる競合がワーカスレッド間で発生することを回避できる．

また，ワーカスレッド間と計算機間の負荷分散はキャッシュベースのランダムタスク・ステイル法を階層的に適用することで動的に実施する．

キャッシュベースのランダムタスク・ステイル法は，ランダムにタスク要求を出すのではなく，タスクを獲得できたワーカを記録しておき，記録がある場合，記録されたワーカに最初に優先的にタスク要求を出す手法である．列挙木の探索では負荷の偏りが大きくなり，特定のワーカ群にタスクが集中する可能性がある．このような場合，タスク要求をランダムに出すと無駄なタスク要求が増えるため，全体の性能が低下する可能性が高い．キャッシュベースのランダムタスク・ステイル法では，タスクを獲得できたワーカにはタスクが存在する可能性が高いため，そのワーカに優先的にタスク要求を出すことにより，この問題が発生することを防いでいる．

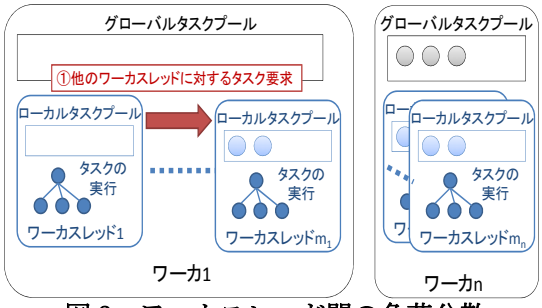

図 2：ワーカスレッド間の負荷分散

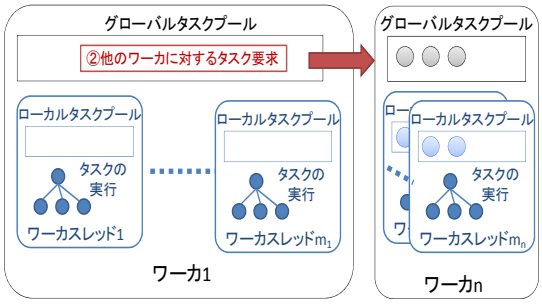

図 3：ワーカ間の負荷分散

以下に負荷分散の手順を示す．

(1) ワーカスレッドは，ローカルタスクプールおよび，ローカルのグローバルタスクプールにタスクがなくなった場合，他のワーカスレッドにタスク要求のメッセージを送信する（図 2）．メッセージ送信の方法としてキャッシュベースのランダムタスク・ステイル法を用いる．タスク要求を受け取った他のワーカスレッドは，ローカルタスクプールにタスクがある場合はそのタスクを返事として返す．

(2) もし，他のワーカスレッドからタスクを取得できた場合は，負荷分散の処理を終了する．他のどのワーカスレッドからもタスクを受け取ることができなかった場合，(3) へ進む．

(3) ワーカは他のワーカへタスク要求を出す（図 3）．メッセージ送信の方法としてキャッシュベースのランダムタスク・ステイル法を用いる．タスク要求を受け取ったワーカは，グローバルタスクプールにタスクが存在する場合はそのタスクを，存在しない場合はワーカ内のローカルタスクプールに存在するタスクがあれば奪い取り，要求を出したワーカに返す．

（２）性能評価

開発を行ったマルチコア計算機クラスタのための並列化モデルを用いて，テキストデータベースから頻出系列パターンを取り出す典型的な手法の１つである段階的一般化法の並列化を行った．段階的一般化法では最小汎化集合と呼ばれる系列パターンを頻度付きで取り出すことができる．

頻出系列パターンを抽出する処理は，多くの手法が系列を列挙するためにメインの処理が列挙木の探索となる．段階的一般化法では列挙木の深さの偏りが非常に大きく，単純に並列化しただけだと負荷の偏りが大きくなる．そこで，開発を行った並列化モデルを用いて並列化を行うことで負荷の偏りを動的に解消することができると期待される．

実験環境は，表１に示すように４台の計算機から構成されるマルチコア計算機クラスタを使用した．また，各計算機で 1.0Gbs のイーサネットに接続されている．各計算機に 1 つのワーカを配置し，各ワーカで起動するワーカスレッドの数は均一であるものとする．

表１：計算機環境

CPU	AMD PhenomX4 9350e(2.00GHz
コア数	4
メモリ	DDR2-800 2GB
ディスク	500GB
OS	Fedora Core 12

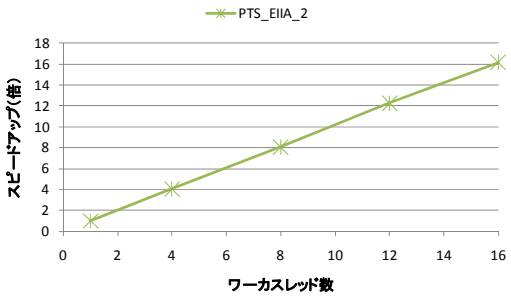

図 4：スピードアップ
(PTS_EIIA_2 データセット)

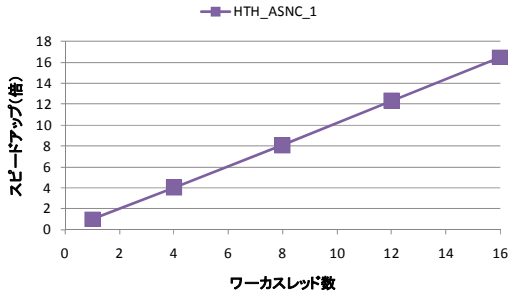

図 5：スピードアップ
(HTH_ASNC_1 データセット)

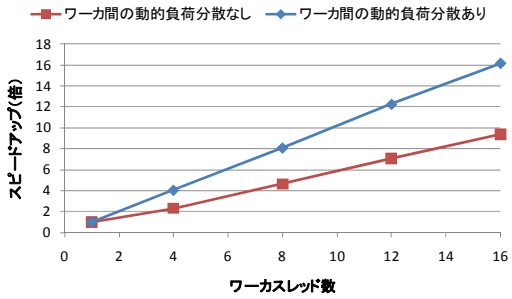

図 6：ワーカ間の負荷分散の効果
(PTS_EIIA_2 データセット)

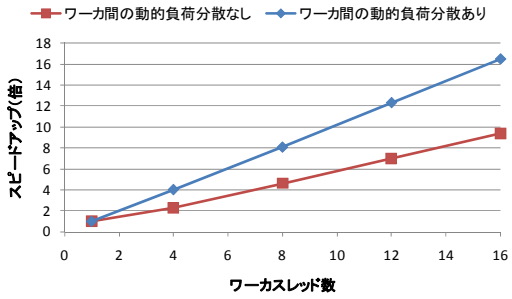

図 7：ワーカ間の負荷分散の効果
(HTH_ASNC_1 データセット)

図 4 と図 5 に実験結果を示す．図 4 と図 5 とは横軸はワーカスレッド数，縦軸はスピードアップ（速度向上比）を示す．スピードアップとは，ワーカスレッド数が 1 のときの処理時間を複数のワーカスレッドを使用したときの処理時間で割った値を示している．このグラフから分かるようにワーカスレッドが 16 のときにほぼ 16 倍のスピードアップが得られている．

また，ワーカ間において負荷分散をしなかった場合の実験結果を図 6 と図 7 とに示す．グラフから分かるように，ワーカ間において負荷分散をしなかった場合は，性能が出ていないことが分かる．負荷分散を行わなかった場合は，スピードアップは 9 倍で，負荷分散を行った場合と比較して，性能が悪い．

マルチコア化は急速に進んでおり，マルチコアの計算機資源を活用するためにはマルチコアの特徴を考慮した並列化モデルの開発が不可欠である．開発を行った並列化モデルは頻出系列パターン抽出だけでなく他のアプリケーションにも応用可能である．今後の方向性としては，開発を行った並列化モデルを他のアプリケーションの並列化に応用し，今回開発を行った並列化モデルの一般的な有効性を示していきたい．

５．主な発表論文等

〔学会発表〕（計３件）

[1] 澤田 祐介，田村 慶一，荒木 康太郎，北上 始：ミスマッチクラスタを表現する最小汎化集合の高速抽出，第 72 回情報処理学会・数理モデル化と問題解決研究会，pp. 73-76，2008 年 12 月．

[2] 宮原 和也，田村 慶一，北上 始：アライメントに基づくミスマッチクラスタからの最小汎化集合の抽出，データ工学と情報マネジメントに関するフォーラム DEIM2010，電子情報通信学会データ工学研究専門委員会，Online Proceedings，2010 年 3 月．

[3] 田村 慶一，北上 始：マルチコア計算機クラスタ上における段階的一般化法の並列処理，FIT2010，2010 年 9 月発表予定．

６．研究組織

(1)研究代表者

田村　慶一（Tamura Keiichi）
広島市立大学・情報科学研究科・講師
研究者番号：80347616

(2)研究分担者

なし

(3)連携研究者

なし