並列 FP-growth における処理負荷予測及び負荷分散への適用

Processing Load Prediction for Parallel FP-growth and Its Application for Load Balancing

イコ プラムディオノ♦ 高橋 克巳♦
アンソニー K．H．テゥング♥
喜連川 優♠

Pramudiono IKO Katsumi TAKAHASHI
Anthony K.H. Tung Masaru KITSUREGAWA

並列相関ルールマイニング処理において、負荷分散がスケラビリティーの重要な要素となる．本論文では並列 FP-growth における負荷予測手法を検討する．まず、マイニング処理実行前に収集可能な path depth のような統計に基づいた負荷予測関数及びアイテム処理順の提案と共に、マイニング処理後にしか収集できない処理回数のサンプリング手法も提案する．その上、処理単位である条件つきパターンベースの初期配分及び条件つきパターンベース処理中の負荷分散に提案手法を実装し、有効性を検証する．性能評価より、サンプリング手法及びアイテム処理順の工夫が初期配分に有効であることが示される．

Load balancing is a dominant factor to achieve scalable parallel frequent pattern mining. In this paper, we examine some methods to predict processing load for parallel FP-growth algorithm. We propose item processing order based heuristic and load prediction function based on the path depth and other statistics which can be collected before the execution of mining process. We also propose sampling to predict statistics such as the number of iterations. Finally, we implement those methods to improve the initial distribution of processing units i.e. conditional pattern bases as well as the load balancing during the execution of those conditional pattern bases. The performance evaluation shows that sampling based load prediction and item ordering heuristics perform well for the initial distribution.

1. はじめに

頻出パターンマイニング分野では、Apriori というアルゴリズム[1]が主流だったが、FP-growth が新たなパラダイムを切り開いた[4].トランザクションデータベースを FP-tree というメモリ上データ構造に圧縮することで、Apriori のような従来アルゴリズムよりはるかに高い性能が達成できた.

さらなる高性能なマイニング処理は並列化により期待できるが、FP-growth のような主メモリ上に処理が必要なため、データの分割や負荷分散が困難であるとされている.そのため、最初の並列化実装は共有メモリ計算機上である[6]．無共有環境上の最初の実装である並列FP-growth は path depth というパラメータに基づいた負荷分散機構を備えたが、負荷予測モデルとして十分ではなかった[5].

本論文では、PC クラスタのような無共有環境上の並列アルゴリズムをスケールアップするために、並列化のボットルネックとなる負荷の偏り及びその対策を論ずる．本研究は並列FP-growth を基に、処理単位である条件つきパターンベース（conditional pattern base、CPB）処理に関する負荷指標を詳細に評価し、それらに基づいて、負荷予測モデルを提案する．条件つきパターンベース処理前に収集できる負荷の統計的指標を検討した結果、負荷予測関数を用いる負荷分散機構及びアイテムの処理順による最適化手法を提案する．また、処理後の負荷統計である処理回数を少量のデータセットよりサンプリングし、負荷を予測する手法も提案する.

提案手法を並列FP-growth の二つの処理ボトルネックに適用し、PC クラスタ上の実験により、その有効性を検証する．最初のボトルネックは処理ノードに分散された条件つきパターンベースを集約し、実行するノードに初期配分するフェーズであり、もう一つは条件つきパターンベース処理中のノード間偏りを均等化する負荷分散フェーズである.

本論文の構成は第２章で基となる並列FP-growth を簡略に説明する．第３章では、条件つきパターンベース処理に関する統計を説明し、最も時間がかかる条件つきパターンベース処理より収集した結果を比較し、負荷予測手法としての有効性を検討する．第4章では、条件つきパターンベース処理の負荷に関する統計を用いた負荷分散機構を提案する．特に、初期配分フェーズ及び負荷分散フェーズへ適用する．第５章では、PC クラスタ上で我々の手法の実装及び性能評価を報告する.

2. 並列 FP-growth

FP-growth の実行は二つのフェーズに分けることができる：FP-tree の構築とそのFP-tree からすべての頻出パターンを抽出する[4]．並列FP-growth の基本的な考え方はそれぞれの処理ノードが完全な条件つきパターンベースを受けてから、独立にその条件つきパターンベースの処理を完成させる．単純並列化の例は図１に示される．基本的に SEND プロセスと RECV プロセスという二つのプロセスが必要である.

2.1 FP-tree の構築

FP-tree は頻出アイテムだけが含まれる一種の trie であり、頻出なアイテムが共有されやすいように、頻度順に並べる．頻出パターンを求めるのに、必要な情報がFP-tree にすべて含まれている.

例として図１の Node1 と Node2 の SEND プロセスがローカルデータベースからそれぞれの FP-tree を構築する．一回目のデータベーススキャンの後、各ノードの SEND プロセスがすべてのアイテムのカウントをお互いに交換し、全

♦正会員 NTT情報流通プラットフォーム研究所
iko.pramudiono@lab.ntt.co.jp
♦正会員 NTT情報流通プラットフォーム研究所
takahashi.katsumi@lab.ntt.co.jp
♥シンガポール国立大学 atung@comp.nus.edu.sg
♠正会員 東京大学生産技術研究所
kitsure@tkl.iis.u-tokyo.ac.jp

体の頻出アイテムを決定する．交換後、各ノードの SEND プロセスが全アイテムのカウントを保有するので、サポートの降順にソートし、グローバルのF-listを生成できる．二回目のデータベーススキャンで、各々の SEND プロセスがノードのローカルデータベースからローカルFP-treeを構築する．

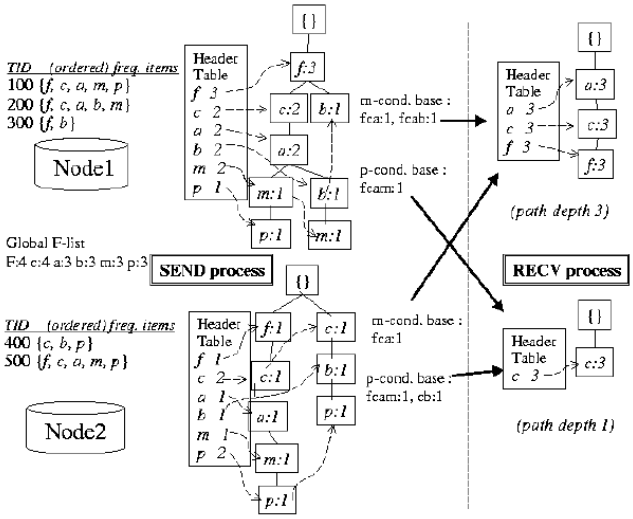

図１ FP-growth 単純並列化の例

Fig.1 Illustration of FP-growth Trivial Parallel Execution

2.2 FP-growth

並列FP-growthはローカルのFP-treeより条件つきパターンベースを生成する．m条件つきパターンベースはmと共起するすべてのアイテム集合の部分データベースとなっている．

その条件つきパターンベースから条件つきFP-treeを構築し、新たな条件つきパターンベースが生成できないまでFP-growthが再帰的に繰り返され,パターン片を成長させていく．自ら処理するより、SENDプロセスがハッシュ関数を用いて、条件つきパターンベースを他のノードのRECVプロセスに送る．RECVプロセスがすべてのノードのSENDプロセスから条件つきパターンベースを収集し、グローバル条件つきパターンベースを再構築する．グローバル条件つきパターンベースを再現したRECVプロセスは独自にFP-growthを実行する．

3. 処理負荷の統計

並列Fp-growthは条件つきパターンベース処理を処理単位としたため、条件つきパターンベースの処理負荷に関する統計的指標を検討する．

表１では、15位までの最も負荷が重い条件つきパターンベース処理に関する統計が示される．T25.I20.D100K.i10KというデータセットはApriori論文のデータ生成手法に基づき、次のパラメータを用いた：トランザクション数１００K、アイテム数１０K、平均トランザクション長２５、平均頻出アイテム長２０である[1]．最小支持度は0.1%に設定された．その最小支持度を満たす頻出アイテムは6689であるので、10Kアイテムのうち、1000を超える支持度を持つアイテムは6689であることが分かる．

表１ 負荷が重い処理の統計

Table 1. Statistics of the most time consuming conditional pattern bases

負荷ランク	アイテム順	処理回数	パターン長	要素数	ノード数
1	5507	16036864	23	107	241
2	6126	8650752	19	73	297
3	4792	7920128	22	132	236
4	4350	5161984	18	132	285
5	5577	4174752	21	101	236
6	4164	4072448	21	169	207
7	4119	2580992	17	155	261
8	5444	2096640	21	116	286
9	4066	2083072	21	176	217
10	3780	1290496	16	168	241
11	5423	1048576	21	109	253
12	3526	1041536	20	201	199
13	4568	659456	16	122	285
14	4961	524288	20	119	192
15	3345	524288	20	219	195

それぞれの統計について検討する．

１．アイテム支持度

アイテム支持度は一回目のデータベーススキャンで収集可能である．その支持度に基づき、F-listに表されるアイテム順が決定され、FP-treeに挿入される順番も決まる．

しかし表１より、アイテム順が処理負荷の順番に比例しないことが分かる．6689頻出アイテムのうち、最も処理負荷が重いのがアイテム5507、その次は6126、4792等である．また、アイテム順の最適化の可能性も示唆される．この最適化が初期配分フェーズに詳細に検討される．

２．FP-tree内のノード数

FP-tree内のノード数はFP-tree構築後に簡単に収集可能な統計である．ノード数が多ければ、FP-treeを探索する時間が増加する．FP-tree探索が負荷の主な要因であると考えられるが、負荷ランクとの相関が見られない．

３．条件つきパターンベース内の要素数

条件つきパターンベース内の要素数も処理前に収集可能な統計である．それはもう一つの処理負荷の主な要因であるFP-treeへの挿入回数を決定するが、ノード数同様、相関が見られない．

４．再帰的処理の回数

条件つきパターンベース処理を処理単位とするので、その再帰的な処理回数を負荷の指標にするのが当然である．しかし、その正確な回数は処理が終了するまで記録できない．また、ある条件つきパターンベース処理回数はそれから再帰的に生成される条件つきパターンベースの処理回数の総数である．

５．パターン長

頻出パターンのパターン長も処理負荷の良い指標である．処理負荷はパターン長に指数的に比例すると考えられるが、正確なパターン長は条件つきパターンベース処理終了後しか測れない．

６．Path depth

Path depthの定義は次の通りである：

Path depthの定義：条件つきパターンベース内の最小サポート値を満たす最長のパターンの長さである．

Path depthの例：図１では、mの条件つきパターンベース内で最も長いパターンは<acf>のため、そのpath depthは３である．

そのため、path depthはパターン長を処理前に近似することが可能だと考えられる．Path depthはFP-treeの構築時に収集可能だが、FP-treeのいくつかのパスが頻出パターンにならないことがあるため、あくまでもパターン長の近似とする．

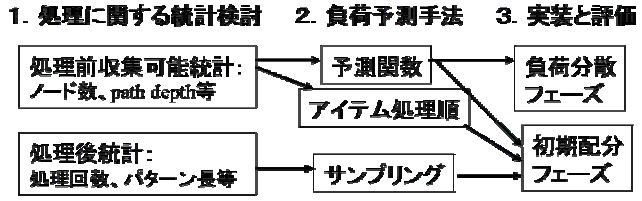

図２　負荷予測の提案手法の分類

Fig.2　Proposed methods of load prediction

4. 負荷予測手法

いくつか頻出パターンマイニングのアルゴリズムが提案されてきたが、頻出パターンマイニング処理に関する負荷モデルの議論がほとんどされていない．

本研究では、前章で検討した統計を用いて、負荷予測手法を提案する．図２に負荷予測手法の分類及びその適用を示す．処理前に収集可能な統計を基に、予測関数及びアイテム処理順という負荷予測手法を提案する．処理後に収集可能な統計よりサンプリング手法を提案する．それらの手法を初期配分フェーズに実装し、負荷分散フェーズには負荷予測関数を適用する．

4.1　処理前統計の負荷予測手法

条件つきパターンベース処理前に収集される統計を用いて、以下の二つの手法を提案する：．

１．負荷予測関数

支持度suppをもつ長さ|X|の頻出パターンの全サブセットが頻出パターンであれば、処理負荷は$(2^{|X|}-1)$*suppで推測される[2]．しかし、処理前には正確なパターン長が得られないので、path depth αによる近似を用いた負荷予測関数を提案する．また、パターンの支持度の近似として、条件つきパターンベースのアイテムセット支持度$supp_I$を用いる．アイテムセットＩ条件つきパターンベース処理負荷の予測関数として(1)を用いる

処理負荷 = $(2^{\alpha}-1)$*$supp_I$　　(1)

初期配分フェーズでは、予測関数に基づき、条件つきパターンベースの送り先ノードを決定する．また、負荷分散フェーズでは、処理ノードに蓄積される条件つきパターンベースの処理負荷を予測関数で推測し、ある一定の閾値を超える場合、その条件つきパターンベースを分割する．それ以外は、分割オーバヘッドを低減させるため、その条件つきパターンベース処理を最後まで完了させる．

２．アイテム処理順

初期配分フェーズでは、ネットワークのレイテンシーのため、時間のかかる条件つきパターンベースを先に配信されることが望まれる．表１よりF-listのアイテム順に従う処理順は条件つきパターンベースの実際の処理負荷の順番と一致しないことが判明した．図３はパターン長の近似であるpath depthのT25.I20.D100K.i10Kにおける分布を示す．X軸はアイテムを１０分割したセグメントを表し、y軸が条件つきパターンベース内のpath depthの割合を表す．図３よりF-listのアイテム順であるセグメント１０からの処理順の場合、短いpath depthの条件つきパターンベースが最初に処理されることが分かる．

そのため、アイテム処理順による最適化された初期配分フェーズでは、セグメント７よりアイテム処理することで、path depthが長い条件つきパターンベースが先に処理される可能性が高くなる．

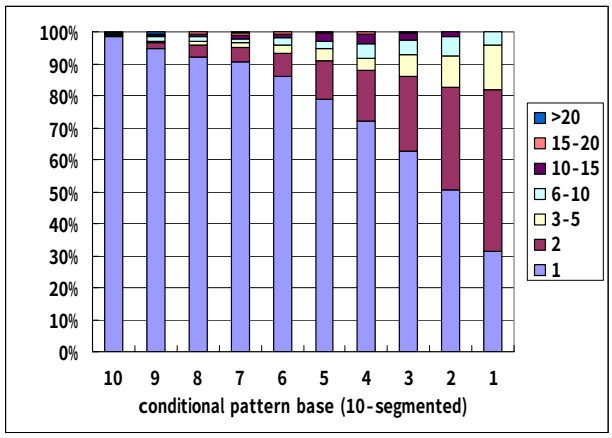

図３　path depth の分布

Fig.3　Path depth distribution

4.2　処理後統計の負荷予測手法

条件つきパターンベースの処理回数のような統計を得るために、マイニング処理を完了させなければならない．データセットをサンプリングすることで、少量のデータをマイニング処理し、事前に統計を収集ことが可能になる．

サンプリング手法は相関ルールマイニング初期段階から提案された手法[6]だが、今までは頻出パターンの出力が目的だった．提案手法では、頻出パターン自体ではなく、条件つきパターンベース処理に関する統計の推計を目的とする．

初期配分フェーズへの適用では、FP-treeの構築時にトランザクションのサンプリングを行い、一つの処理ノードに集約し、マイニング処理を行う．その結果、処理回数を収集し、それに基づき、条件つきパターンベースの送り先ノードを決定する．

5. 実装と性能評価

提案した方式の性能を評価するために、ＰＣクラスタ上で実装を行った．実行環境として、100Base-TX Ethernet Switchに相互接続される３２ノードからなるＰＣクラスタである．各々のノードはPentiumIII 800Mhz及び128MBメモリを搭載している．

5.1　データセット

性能評価のために、二つのデータセットを用いた．一つ3章で記述されたT25.I20.D100K.i10Kという人工データである．もう一つは約３４万のトランザクション及び５７２アイテムのベルギー交通事故の実データセットである[3]．最小支持度は20%とする．

5.2 初期配分フェーズ

人工データ及び実データを用いたノード数４及び８の実行時間の結果を表２と表３に示す．人工データの場合、いずれもアイテム処理順最適化が最も速いが、アイテム数が少ない実データの場合、ノード数が増加するとサンプリング手法がアイテム処理順最適化よりも高速する．精度の良い予測がサンプリングオーバヘッドを十分に補うと考えられる．

表2 負荷分散手法比較（初期配分、人工データ）
Table 2. Comparison of load balancing methods (Initial distribution, synthetic data)

ノード数	処理順	予測関数	サンプリング
4	92.9s	103.3s	96.6s
8	52.3s	57.3s	53.4s

表3 負荷分散手法比較（初期配分、実データ）
Table 3. Comparison of load balancing methods (Initial distribution, real data)

ノード数	処理順	予測関数	サンプリング
4	193.9s	204.9s	194.8s
8	112.6s	112.6s	96.4s

5.3 負荷分散フェーズ

従来のpath depthのみの負荷分散機構と比較した性能評価を図４に示す[5]．8ノード以上では、絶対性能がmin pdepth閾値の12の従来手法に及ばないものの、負荷予測関数手法の閾値thを1000から100000に変更しても性能への影響が小さく、パラメータ依存度が低いと言える．

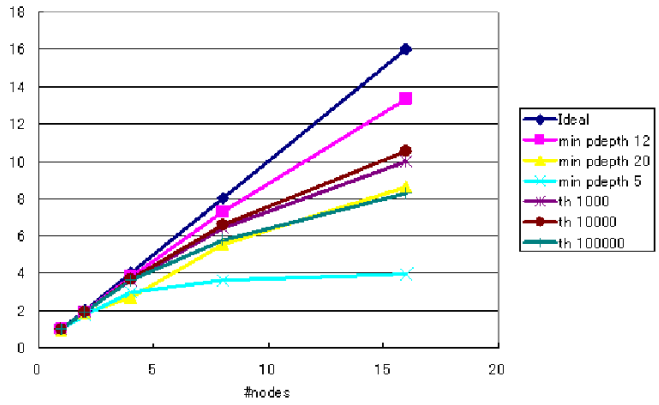

図４ 台数効果（人工データ）

Fig.4 Speedup ratio(synthetic data)

6. まとめと今後の課題

ＰＣクラスタで実装される並列 FP-growth の最適化を報告した．マイニング処理前後の統計を検討した結果、負荷予測関数、アイテム処理順及びサンプリングという三つの負荷予測手法を提案し、実験を通して、その有効性を比較した．初期配分フェーズでは、アイテム処理順最適化及びサンプリング手法が最も良い性能を示すが、負荷予測関数の手法がまだ改良が必要である．

これからより高精度の負荷モデルを構築する他、他のアルゴリズムへの適用も検討する

[文献]

[1] R. Agrawal and R. Srikant."Fast Algorithms for Mining Association Rules". In Proceedings of the 20th Int. Conf. on VLDB, pp.487-499, September 1994.

[2] G. Cong, B. C. Ooi, K.-L. Tan, A. K. H. Tung." Go Green: Recycle and Reuse Frequent Patterns". In Proc. of Int. Conf. on Data Engineering (ICDE'2004), 2004.

[3] Geurts, K. Wets, G. and Brijs, T.`` Profiling high frequency accident locations using association rules". In Electronic Proc. of the 82th Annual Meeting of the Transportation Research Board, 2003.

[4]J. Han, J. Pei and Y. Yin "Mining Frequent Pattern without Candidate Generation"In Proc. of the ACM SIGMOD Conf. on Management of Data, 2000

[5] I. Pramudiono and M. Kitsuregawa ``Tree Structure based Parallel Frequent Pattern Mining on PC Cluster". In Proc of 14th Int. Conf. on Database and Expert Systems Applications (DEXA'03), 2003.

[6] H. Toivonen. ``Sampling Large Databases for Association Rules" In Proc. of the 22th Int. Conf. on Very Large Data Bases(VLDB), 1996

[7] O. R. Zaiane, M. El-Hajj, and P. Lu. `` Fast Parallel Association Rule Mining Without Candidacy Generation" In Proc. of the IEEE 2001 Int. Conf. on Data Mining (ICDM'2001), 2001.

イコ プラムディオノ Pramudiono IKO
2004 年東京大学大学院工学系研究科博士課程卒業．2000 同大学大学院工学系研究科修士課程修了．工学博士．現在ＮＴＴ情報流通プラットフォーム研究所社員．並列データマイニング・ウェブマイニングの研究・開発に従事．

高橋 克己 Katsumi TAKAHASHI
東京大学大学院工学系研究科博士課程在学中．1988 年東京工業大学大学院修士課程修了．現在ＮＴＴ情報流通プラットフォーム研究所主幹員．モバイル情報サービス、セキュリティーとプライバシー、データマイニング、データストレージシステムの研究・開発に従事．

アンソニーＫ．Ｈ．テゥング Anthony K.H. Tung
2001 年シモンフレイサー大学博士課程卒業．シンガポール国立大学助教授．

喜連川 優 Masaru KITSUREGAWA
1978 年東京大学工学部電子工学科卒．1983 年同大大学院工学系研究科情報工学博士課程了．工学博士．同年同大生産技術研究所講師．現在，同教授．平成 15 年 4 月より、同所 戦略情報融合国際研究センター長．データベース工学，並列処理，Ｗｅｂマイニングに関する研究に従事． 情報処理学会理事，SNIA-Japan 顧問，ACM SIGMOD Japan Chapter Chair．平成 9，10 年本学会データ工学研究専門委員会委員長． VLDB Trustee，IEEE ICDE，PAKDD，WAIM ステアリングコミティメンバ．