

Title	並列ハッシュ結合における実行時のデータの偏りの扱いに関する研究
Author(s)	土屋, 由美子
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1073
Rights	
Description	Supervisor:横田 治夫, 情報科学研究科, 修士

並列ハッシュ結合における実行時のデータの偏りの扱いに関する研究

土屋 由美子

北陸先端科学技術大学院大学 情報科学研究科

1997年2月14日

キーワード: 並列データベース, ハッシュ結合, 結合偏り, 負荷均衡.

研究の背景 近年、並列データベースは多くの研究が行われている分野であり、その成果も多くの商用システムとして世の中に出回っている。並列データベースに関する課題の1つにデータ偏りの制御がある。データ偏りに関する研究の中に、データ偏りを扱う並列ハッシュ結合アルゴリズムの研究がある。並列ハッシュ結合では、リレーションをハッシュにより複数のパーティション（バケット）に分割し、そのそれぞれの結合を各プロセッサで独立に実行させる事ができるアルゴリズムである。並列ハッシュ結合は、結合属性値の分散が均一である場合、各プロセッサへ均等な負荷を与える事ができ、スケールアップ、スピードアップ共に優れるアルゴリズムとなっている。しかし、並列ハッシュ結合はハッシュ関数を用いて部分処理を生成しているので、特定のハッシュ値に属性値が集中するようなリレーションに対する結合の場合は、属性値が集中したバケットの処理により全体の性能が低下してしまう。本論文内では並列ハッシュ結合アルゴリズムおよび、並列ハッシュ結合に対するデータ偏りの結合処理時間への影響について実験した結果を述べている。

既存の偏り制御アルゴリズム データ偏りによる著しい並列ハッシュ結合の性能低下を防ぐため、様々なデータの偏りに対応できる並列ハッシュ結合アルゴリズムが提案された。初期のデータ偏りへの対応は、バケット数をプロセッサ数よりも多くし、バケットサイズを調べ、各プロセッサの総処理タプル数が均衡するようにプロセッサへバケットを割り当てていく、再分散偏り制御であった。この再分散偏り制御を行う並列ハッシュ結合はZipf-like 分散を用いたシミュレーションで、データ偏りがある場合に通常の並列ハッシュ結合に比べより良い性能を示した。本論文内では、[KIT90], [HUA95] で提案された再分

散偏り制御アルゴリズムおよび、プロセッサへのバケットの割り当て方による結合処理の応答時間を調べる実験を行った結果について述べている。

この様なアルゴリズムにより、各プロセッサへ同数のタプルを割り当てる事ができたとしても、各バケットの結合選択率の違いにより、バケット結合の結果として生成されるマッチタプル数がプロセッサにより均一でない場合、マッチタプルのライト処理によりプロセッサの処理時間がばらつき、著しい性能低下を引き起こしてしまう。これに対応するために、各バケットの生成マッチタプル数を見積り、この情報も用いてプロセッサへのバケット割当を決めるような静的な結合偏り制御アルゴリズムが提案された。本論文では、[WOL91] で使われているマッチタプル数をバケットサイズから静的に見積もる方法について述べ、その見積り能力の検討を行っている。

しかし、この様な方法を用いても静的な結果見積りが誤る場合や 1 つの属性値が高い偏りを持つような場合には十分な効果が得られない。このため、バケット結合を監視し、見積り通りに処理が行われているか調べ、もし見積りが誤っている場合でも、そのバケットを処理しているプロセッサが他のプロセッサに比べて、重い負荷を負わないようにその処理の一部を移送する事で動的に結合生成偏りに対応するアルゴリズムが現れた [HAR95]。本論文ではこのアルゴリズムの詳細について説明している。また、このアルゴリズムで用いられている過負荷の検出法や実行時再見積りに関する実験結果についても述べている。

コーディネータの分散配置による動的な偏り制御 [HAR95] で提案された動的な結合偏り制御アルゴリズムでは、単一のコーディネータが各バケット処理の監視、負荷移送の決定、処理プロセッサの状態の管理を行い、それらから得られる情報を基に、効果的な結合偏り制御を行う。しかし、並列データベースの規模が拡大を続けるとすると、大量の情報および偏り制御処理が単一のコーディネータに集中してしまい、その拡大の規模によっては単一のプロセッサの能力を越えてしまう事が考えられる。このため、本研究では、コーディネータの行う処理を各プロセッサがローカルに実行するための方法について検討した。

コーディネータを分散配置し、各プロセッサが独自の判断で動的に過負荷を検出し、必要であればその移送を行うために、次の方法で動的な偏り制御を行う。

過負荷の検出 処理を終えたプローブタプル数とそのプローブタプルにより生成されたマッチタプル数をカウントし、これらから、1 タプルあたりの生成マッチタプル数（これを爆発率という）を計算し、これをあらかじめ見積もられた爆発率と比較する事により、予測以上の生成マッチタプルが発生する場合を検出する。予想以上の生成マッチタプルは予想外の負荷に相当するので、これを検出する事で過負荷の検出が行える。

移送する処理 移送する処理の内容は生成マッチタプルのライト処理である。大量のマッチタプルが生成される場合、単一のディスクへこれらタプルのライト処理が集中してしまう事により著しい負荷の増加が発生する。この処理を他ノードで実行する事により過負荷ノードの処理時間の短縮とディスク容量の偏りの緩和が可能となる。

過負荷の移送量 過負荷が検出されると、それ以降にそのバケットの結合により生成されたマッチタuppルの一部は他ノードへ移送される。この移送量は、各プローブタuppルにより生成されたマッチタuppルの $(N - 1)/N$ となる。ここで N はプロセッサ数とする。すなわち、過負荷が検出されたノードのライト処理は $1/N$ ずつ全ノードが処理する事になる。ただし、誤って過負荷を検出した場合を想定し、各プローブタuppル処理により生成されたマッチタuppル数を調べ、その数があらかじめ見積もられた爆発率よりも小さい場合には過負荷の移送を行わない。

このように、コーディネータの分散配置のために本研究では移送の方式を移送先の状態の管理を省略し、生成マッチタuppルのライト処理という単一の負荷移送を行う事にした。この場合、移送先プロセッサの状態を考慮した移送を行う事が困難となり、十分な性能を示せない可能性がある。

本研究ではこのような方法による偏り制御の能力に関する実験を nCUBE/2 上で行い、どの程度の性能を示すか調べた。実験では、各プロセッサのライトタuppル数、各プロセッサの結合処理時間について通常の並列ハッシュ結合と比較した結果を示す。この結果より、各プロセッサのライトタuppル数のばらつきが通常の並列ハッシュ結合アルゴリズムに比べ $1/3$ の標準偏差を持ち、高偏りバケット処理を通常の並列ハッシュ結合アルゴリズムに比べ最良の場合で 0.67 倍の処理時間で行える事が分かった。

参考文献

- [DWT92] D.J. DeWitt, J.F. Naughton, D.A. Schneider, S. Seshadri “Practical Skew Handling in Parallel Joins” Proc. 18th VLDB Conf. 1992
- [HUA95] K.A. Hua, C. Lee, C.M. Hua “Dynamic Load Balancing in Multicomputer Database Systems Using Partition Tuning” IEEE Trans. Knowledge and Data Engineering. vol. 7, No. 6, Dec. 1995
- [KIT90] M. Kitsuregawa, Y. Ogawa “Bucket Spreading Parallel Hash: A New, Robust, Parallel Hash Join Method for Data Skew in the Super Database Computer” Proc. 16th Conf. VLDB 1990
- [HAR95] L. Harada, M. Kitsuregawa “Dynamic Join Product Skew Handling for Hash-Joins in Shared-Noting Database Systems” Proc. 4th Inter. Conf. on DAS-FAA’95, Apr. 1995
- [WOL91] J.L. Wolf, D.M. Dias, P.S. Yu, J. Turek “An Effective Algorithm for Parallelizing Hash Joins in the Presence of Data Skew” Proc. 7th Int. Conf. DE 1991