

高速フィルタリングプロセッサ実験システムの開発 (4)

3Q-7

性能評価

酒井 直文、河村 信男、土田 正士、武藤 英男
(株)日立製作所 システム開発研究所

1. はじめに

既に実用面でも主流になっているRDBは、使い易い反面、性能上の問題が有る。特にRDB本来の特徴である柔軟な問合せを処理するためには、インデクス付けされていないカラムについても効率的に検索する必要が有り、これを実現するために外付けのプロセッサを用いるアプローチがある〔1〕〔2〕。我々は、高速フィルタリングプロセッサを利用して検索処理を実行する実験用DBMSを開発した。本稿では、この実験システムの性能をシミュレータで評価した結果を報告する。

2. 評価モデル

本実験システムの目的は、同一DBに対して複数のユーザが並列にアクセスするマルチユーザ環境で、集合処理のCPUステップ数を削減し、チャンネル負荷を低減することにより、RDBの特徴である柔軟な問合せの処理を効率化することに有る。このため、一件検索(本プロセッサを用いずにインデクス経由でカラムにアクセスする検索)と集計検索(あるカラムに注目して平均や合計などを求める検索)とが混在するモデル(TM1)、及び一件検索と集合検索(条件を満足するタプルをユーザに出力する検索)とが混在するモデル(TM2)の2つのモデルを設定し、集計検索と集合検索に本プロセッサを用いた場合(実験システムの方式)と、用いない場合(従来方式)との性能評価を行った。評価すべき性能としては、応答時間、CPU利用率、チャンネル(CH)利用率を選んだ。さらに、システムパラメタとしてCPU性能、ディスク性能、全検索要求の中の一件検索の比率、本プロセッサ利用時の一括入力ページ数などを可変にして、様々な利用形態を想定している。

3. 評価結果

図1～図3に本実験システムの性能評価結果の一例を示す。(集合処理の応答時間、CPU利用率、CH利用

率、及び一件検索の応答時間をプロットしてある。)

図1、図2はCPUの相対性能1の場合のTM1、TM2の性能をそれぞれ示したものであり、図3はCPUの相対性能4の場合のTM1の性能を示したものである。また、図1、図3は長さ50バイトのタプルが20万件あるDB(約2500ページ)を検索対象とし、図2は長さ500バイトのタプルが5万件あるDB(約6250ページ)を検索対象としている。

3.1 CPU性能による違い

図1ではCPU性能が低いため、CPU利用率は本プロセッサを利用した場合でもかなり高くなっているが、それでもCPUで処理すべき演算の絶対量が減っているために、応答時間は1/5以下になっている。一方、図3ではCPU性能に余裕があるため、本プロセッサ利用の効果が応答時間に関しては図1ほどは見えないが、CPUステップ数が大幅に削減できている。このように、負荷に対してCPU性能が低い場合は応答時間の短縮という形で、負荷に対してCPU性能が高い場合はCPUステップ数の削減という形で本プロセッサの効果が外に現われる。(ただし、応答時間の短縮も、当然CPUステップ数の低減によって可能になったわけである。)

3.2 処理内容、DB構造による違い

処理内容やDB構造の違いが性能に与える影響は、図1と図2とを比較すればわかる。TM1とTM2ではとCPUの処理量はTM1の方が多く、またDB内のタプル件数も図1の場合の方が多いため、本プロセッサを使わない場合のCPUでの処理量のうち、本プロセッサ利用によって削減できる処理量は図1の方が多いため、CPU利用率、応答時間ともに図1の方が本プロセッサの効果が大きくなっている。ただし、DBの全ページ数は図2のDBの方が多いため、CH利用率に関しては、図2の方が効果が大きくなっている。

3.3 平均性能と最大性能

以上、図1～図3に示した例を用いて実験システムの性能評価結果を簡単に紹介したが、各種システムパラメータや問合せパターンを変化させたシミュレーションの結果、平均的には、CPUステップ数が1/3程度、応答時間が1/2程度になるという結果が得られた。ただし、ここでの評価はDBや問合せパターンを現在実際に使われている典型的な環境に合わせた結果である。本プロセッサの効果が最も顕著な例——複数の文字列検索指定があり、CPU負荷が非常に高いような問合せ——では、CPUステップ数が1/100程度になることを確認している。

3.4 その他

このように、程度の差はあるが、ここで示したどの例でも本プロセッサを利用した場合の集合検索の性能が良くなっているが、それ以外に、特に図1のような場合に一件検索の応答時間が大幅に短縮されていることは注目に値する。これは集合検索に本プロセッサを適用することによってCPUステップ数が削減できたため、空いた時間を一件検索の処理にあてることが可能になったことを示している。図2、図3の場合にこの効果が見えないのは、もともとのCPU利用率があまり高くないためである。

ところで、ここではインデックス付けされていないカラムの検索に限って本プロセッサの適用を考えたが、インデックス付けされているカラムを検索する場合でも、条件を満足するタプルが多ければCPUステップ数やI/O回数が多くなるので、本プロセッサを用いた方が効率的である。また、本プロセッサ利用に伴う一括入力ページ数を多くし過ぎると、排他制御の点でシステム性能が悪くなることもある。このような点を考慮して、いつ本プロセッサを使うべきかを判断する最適化処理も組み込んでいる([3])。

4. おわりに

本プロセッサを利用した実験システムの性能をシミュレータで評価した結果、平均的なマルチユーザ環境においてCPUステップ数が1/3程度、応答時間が1/2程度ということを確認できた。

今後は、本プロセッサ実機を用いてさらに広範囲にわたって性能評価を行い、その有用性を確認する予定である。

参考文献

1. 北嶋他：高速フィルタリングプロセッサ実験システムの開発(1) —全体構想—、情報処理学会第38回全国大会。
2. 大曾根他：高速フィルタリングプロセッサ実験システムの開発(2) —ハードウェア方式—、情報処理学会第38回全国大会。
3. 河村他：高速フィルタリングプロセッサ実験システムの開発(3) —ソフトウェア方式—、情報処理学会第38回全国大会。

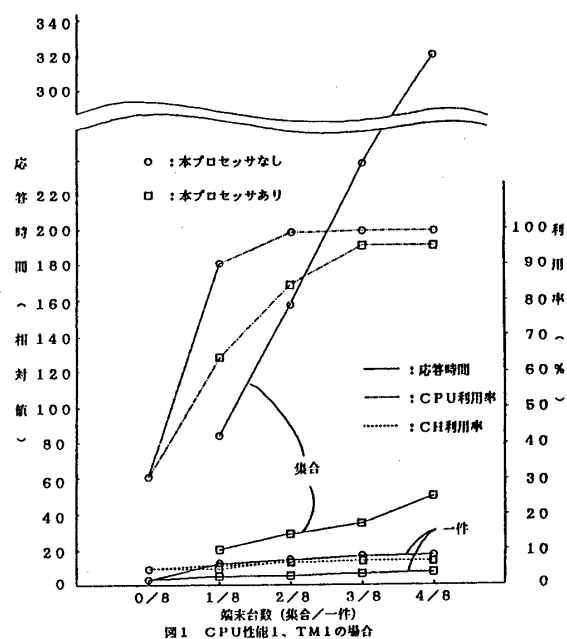


図1 CPU性能1, TM1の場合

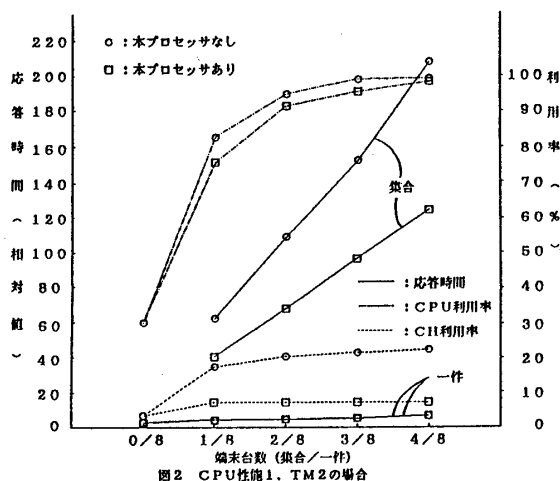


図2 CPU性能1, TM2の場合

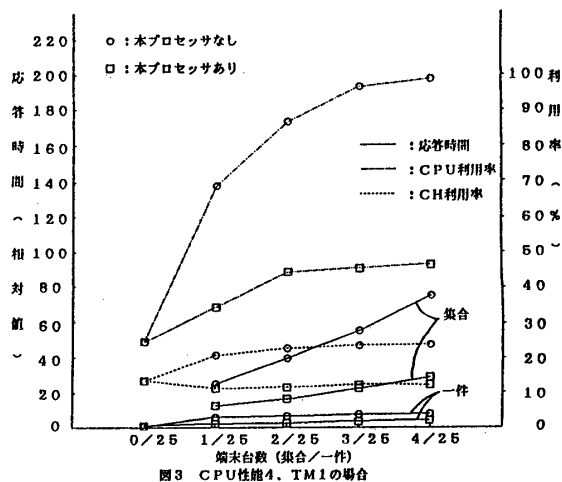


図3 CPU性能4, TM1の場合