Join Strategies on Multi-Dimensional Clustered Relations

7Q-7

Lilian Harada, Masaru Kitsuregawa, Mikio Takagi

University of Tokyo

1. Introduction

For the last years we have developed relational database systems like GRACE and FDS-R (Functional Disk with Relational database engine) in which the relational algebraic operations were based on dynamic clustering algorithms. Performance evaluations of FDS-R [1] and other works [2] showed that the dynamic clustering algorithm was a good solution for very large unstructured relations. Concerning the relational algebraic operations on structured relations, recent researches exemplified by the Predicate-tree [3], the Superjoin Algorithm [4] and the DYOP [5] are based on multidimensional clustered relations. The first two are based on hashing, and the last, on a grid-partitioned relation. Anyway, all them are based on structures that partition the relation based on the relation space, not the attribute values. These kind of partitioning are obviously easy to be handled for join operations because the generated partitions are naturally disjoint concerning the discriminate attributes. However they are not conceivable in real applications since the methods based on hashing are not order-preserving and the ones based on grid-partitions show low load-factor for reasonable number of discriminate attributes.

Here we present some join strategies for KD-tree indexed relations [6]. As a multidimensional clustering method, KD-tree presents the desired characteristics of order preserving and high load-factor. However, the KD-tree generates partitions that are not disjoint concerning the value of discriminate attributes, which makes the operations on them difficult. In what follows we recall some strategies we have proposed to deal with this difficulty and previously introduced in [7]. Some analysis and evaluations demonstrate the effectiveness of the join strategies for KD-tree indexed relations.

2. Analytical Evaluation of the Proposed Join Algorithms

We consider the join of two relations R and S much larger than the main memory. The basic idea of the following algorithms is to divide relation R into sets of pages which are loaded into the main memory, and for each of these sets, determine a range of the join attribute domain in which to execute the join. Given this range the join is executed with the corresponding set of pages of relation S. Here the set of tuples of R which are loaded into the memory and determine the join range is denoted wave. The proposed strategies differ in the determination of the waves and the way theses waves propagate over relation R. We propose 5 basic algorithms and extended versions of them (algorithms 1-5, 1m-5m). For details, refer [7].

For a relation of size $N = 2^p$ pages and indexed by a KD-tree, the minimum size of a wave is $cs = 2^{p(k-1)/k}$ pages. Because the details of the join algorithms are presented in [7], here we only present their analytical evaluation equations.The analysis will be restricted for perfectly balanced KD-tree indexed relations in which at least one cluster fits into the main memory, whose size is M pages. For k = 2, the memory size constraint is identical to those of dynamic clustering algorithms [2]. The following expressions estimate the number of page accesses π for the proposed algorithms 1-4 and their extended versions 1m-4m.

Algorithm 1 :

$$\pi = N\left(5 - \frac{M}{cs}\right)$$

Algorithm 2 :

$$\pi = N\left(\frac{M-1}{cs}\frac{cs}{M-1} + 2\right)$$

Algorithm 3 :

$$\pi = N + \sum_{i=1}^{\frac{p}{k}-1} 2^{ki}\left(2\,cs - M + 2^{ki} + 1\right) + 2\,cs$$

Algorithm 4 :

$$\pi = N\left(1 + \frac{M-1}{M - cs - 1 + 2^{ki}}\right)$$

Algorithm 1m :

$$\pi = \begin{cases} N\left(4 + \frac{\sigma}{\mu} - \frac{M}{cs}\right) & \text{when } M - cs >= \frac{\sigma}{\mu} cs \\ N\left(4 - \frac{1}{cs}\left(2^{k-1} + \sqrt{2^{2(k-1)} + \frac{4\mu}{\sigma}(M - cs)(cs - 2^{k-1})}\right)\right) & \text{otherwise} \end{cases}$$

Algorithm 3m : Since its expression is too long and complex, we omit it here and just show the results in the next section.

Algorithm 4m :

$$\pi = N\left(\frac{(cs - 2^{k-1})^2\,\mu}{(\mu - 3\sigma)\,cs^2 + (2\sigma(M-1) - 2^{k-1}(\mu - \sigma))\,cs} + 2\right)$$

Join Strategies on Multi-Dimensional Clustered Relations

Lilian Harada, Masaru Kitsuregawa, Mikio Takagi

University of Tokyo

3. Simulation Results

Fig. 1(a)(b) and 2(a)(b) show the simulation and respective analytical estimation of number of page accesses for the proposed join algorithms, using random data distributions for k = 2 and k = 3, respectively. As can be seen, the prediction was quite good.

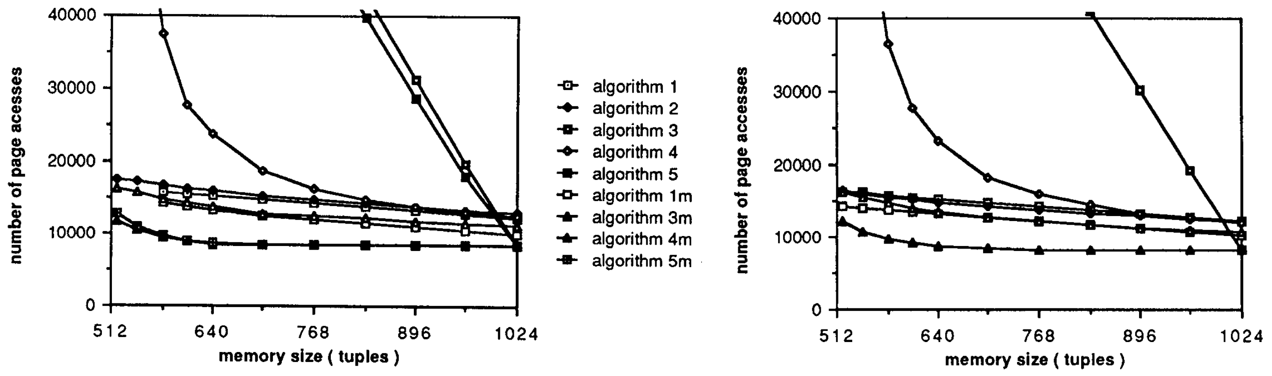

Fig.1. k = 2 (a) Simulation (b) Analytical Estimation

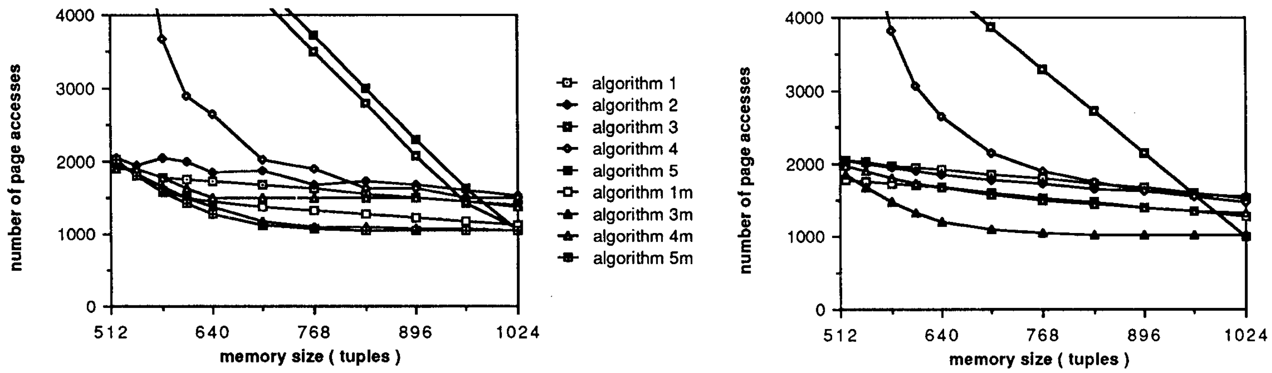

Fig.2. k = 3 (a) Simulation (b) Analytical Estimation

4. Comparison with Dynamic Clustering Strategies

Fig.3 compares the number of pages accesses of the proposed algorithm 3m for 2, 3, and 4 dimensions, with some well known dynamic clustering strategies as simple hashing, grace hash and hybrid hash. As can be seen from the figure, when the memory size is smaller than the cluster size, the performance of our proposed algorithm 3m degrades abruptly. Increasing the dimension of the clustering, the corresponding cluster size increases and the memory size required for a good performance of algorithm 3m increases. Thus, the physical organization of a database must be carefully designed to achieve a good performance.

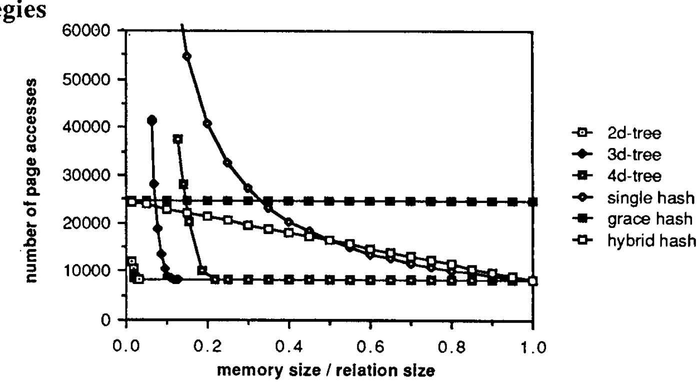

Fig.3. Join Strategies

5. Conclusion

For operations like join, structuring the relation, in order to reduce the address space into partitions and organize data in a way that would present a coupling to the requirements of the operation, avoids the costly partitioning of the relation via hashing presented by the dynamic clustering strategies. Here we use KD-tree indexed relations and present some analytical and simulation results for some join strategies on them, clarifying their effectiveness. Their implementation details are under design now.

References

[1] 原田他：機能ディスクシステム(FDS-R)の性能評価，電子情報通信学会研究会報告DE86-26(1987/3/27)

[2] De Witt, et.al, "Implementation Techniques for Main Memory Database Systems", Proc.of SIGMOD Conf., pp.1-8, June 1984

[3] Cheiney,et.al.,"A Reliable Parallel Backend Using Multiattribute Clustering and Select-Join Operator", Proc.of 12th. Int. Conf. VLDB, pp.220-227, Aug. 1986

[4] Thom,J.A.,et.al., "A Superjoin Algorithm dor Deductive Databases," Proc.of VLDB Conf., pp.189-196, 1986

[5] Ozkarahan,E.A. et.al.,"Join Strategies Using Data Space Partitioning", New Generation Computing, Vol.6 No.1, pp. 19-39, 1988

[6] Bentley, "Multidimensional Binary Search Tree in Database Applications", Trans. on SE, Vol.SE-5, No.4, pp.333-340, 1979

[7] 原田他：KD－木のもとづくFDS-Rのファイル編制-Join演算の処理方式-，電子情報通信学会研究会報告DE87-20 (1988.3.18)